DRA78x
SoC for Automotive Infotainment
Silicon Revision 2.0A, 2.0

Texas Instruments Jacinto6 RSP (Radio Sound Processor) Family of Products

Technical Reference Manual

Literature Number: SPRUIF4D
June 2017–Revised July 2019

Contents

List of Figures

List of Tables

Revision History

Changes from September 4, 2018 to July 12, 2019 (from C Revision (September 2018) to D Revision) **Page**

- Updated Speed Grade in Table DRA78x Part Number Identifier 256
- Updated EffectiveResolution Parameter Value 2514
- Updated Table EMIF Configuration Sequence 2755
- Updated GPMC XIP Timing Parameters 2886
- Updated GPMC Subsection Chip-Select Base Address and Region Size 2886
- Updated Bit-Field Names of CTRL_CORE_SMA_SW_7 Register 3737
- Updated Section Write Nonposting Synchronization Mode and Reset Value of TSICR[2] POSTED Bit 3915
- Removed Caution from I2C Registers 3986
- Updated Entire Subsection McSPI Programming Guide 4113
- Updated Description about the WDCNT Bit Field Behavior 4160
- Updated DLC Coding Description In Subsection CAN FD Operation 4431
- Updated Note for MMCHS_HCTL[2] HSPE Bit and Added Note for MMCHS_CAPA[21] HSS Bit 4867
- Updated Figures in Chapter ADC 5114
- Updated GPMC XIP Timing Parameters 5208

Read This First

Community Resources

The following link connects to TI community resources. Linked contents are provided "AS IS" by the respective contributors. They do not constitute TI specifications and do not necessarily reflect TI's views; see TI's Terms of Use.

TI Embedded Processors Wiki —Texas Instruments Embedded Processors Wiki

Established to assist developers using the many Embedded Processors from Texas Instruments to get started, help each other innovate, and foster the growth of general knowledge about the hardware and software surrounding these devices.

About This Manual

Information About Cautions and Warnings

This book may contain cautions and warnings.

CAUTION

This is an example of a caution statement.

A caution statement describes a situation that could potentially damage your software or equipment.

WARNING

This is an example of a warning statement.

A warning statement describes a situation that could potentially cause harm to you.

The information in a caution or a warning is provided for your protection. Please read each caution and warning carefully.

Register, Field, and Bit Calls

The naming convention applied for a call consists of:

- For a register call: <*Module name*>.<*Register name*>; for example: UART.UASR
- For a bit field call:
  - <*Module name*>.<*Register name*>*[End:Start]* <*Field name*> *field*; for example, UART.UASR[4:0] SPEED bit field
  - <*Field name*> *field* <*Module name*>.<*Register name*>*[End:Start]*; for example, SPEED bit field UART.UASR[4:0]
- For a bit call:
  - <*Module name*>.<*Register name*>*[pos]* <*Bit name*> *bit*; for example, UART.UASR[5] BIT_BY_CHAR bit
  - <*Bit name*> *bit* <*Module name*>.<*Register name*>*[pos]*; for example, BIT_BY_CHAR bit UART.UASR[5]

To help the reader navigate the document, each register call is hyperlinked to its register description in the register manual section. After each register description, a table summarizes all hyperlinked register calls.

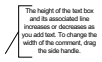
Coding Rules

The programming models or code listings follow the rules:

Type	Definition	Example
File	Starts with the module name	PRCM_test1.c MCBSP1_init.h
Variable	Global variables are prefixed by "g_" Pointers are prefixed by "p" Global pointers are prefixed by "g_p"	g_SDMA_LogicalChan pAddrCounter g_pSDMA_LogicalChan
Function	Starts with the module name	PRCM_SetupClocks() ArmIntC_MaskInterrupts()
Typedef	Ends with "_t"	PRCM_Struct_t
Definition	Starts with the module name and is followed by the register name	#define SMS_ERR_TYPE *((volatileUint32*)0x680080F4) #define MCBSP2_RCR1_REG *((volatile Uint32*)0x4807401C)
Enumeration	Starts with the module name	Typedef enum DMA_Mode_Label { INPUT_MODE OUTPUT_MODE } DMA_Mode_t;

Flow Chart Rules

Flow charts follow the following rules:

Shape	Name	Definition
	Process	Any computational steps or processing function of a program; defined operation(s) causing change in value, form, or location of information
	Decision	A decision or switching-type operation that determines which of a number of alternate paths is followed
	Predefined process or sub-process	One or more named operations or program steps specified in a subroutine or another set of flow charts
	Data or I/O	General I/O function; information available for processing (input) or recording of processed information (output)
	Terminator	Terminal point in a flow chart: start, stop, halt, delay, or interrupt; may show exit from a closed subroutine
	Annotation	Additional descriptive clarification, comment
	On page connector (reference)	Exit to, or entry from, another part of chart in the same page
	Off page connector (reference)	The flow continues on a different page.
	Summing Junction	Logical AND
	Or	Logical OR
	Parallel mode (ISO)	Beginning or end of two or more simultaneous operations
	Flow Line	Lines indicate the sequence of steps and the direction of flow.

DRA78x MIPI® Disclaimer

The material contained herein is not a license, either expressly or impliedly, to any IPR owned or controlled by any of the authors or developers of this material or MIPI. The material contained herein is provided on an "AS IS" basis and to the maximum extent permitted by applicable law, this material is provided AS IS AND WITH ALL FAULTS, and the authors and developers of this material and MIPI hereby disclaim all other warranties and conditions, either express, implied or statutory, including, but not limited to, any (if any) implied warranties, duties or conditions of merchantability, of fitness for a particular purpose, of accuracy or completeness of responses, of results, of workmanlike effort, of lack of viruses, and of lack of negligence.

ALSO, THERE IS NO WARRANTY OF CONDITION OF TITLE, QUIET ENJOYMENT, QUIET POSSESSION, CORRESPONDENCE TO DESCRIPTION OR NON-INFRINGEMENT WITH REGARD TO THIS MATERIAL OR THE CONTENTS OF THIS DOCUMENT. IN NO EVENT WILL ANY AUTHOR OR DEVELOPER OF THIS MATERIAL OR THE CONTENTS OF THIS DOCUMENT OR MIPI BE LIABLE TO ANY OTHER PARTY FOR THE COST OF PROCURING SUBSTITUTE GOODS OR SERVICES, LOST PROFITS, LOSS OF USE, LOSS OF DATA, OR ANY INCIDENTAL, CONSEQUENTIAL, DIRECT, INDIRECT, OR SPECIAL DAMAGES WHETHER UNDER CONTRACT, TORT, WARRANTY, OR OTHERWISE, ARISING IN ANY WAY OUT OF THIS OR ANY OTHER AGREEMENT, SPECIFICATION OR DOCUMENT RELATING TO THIS MATERIAL, WHETHER OR NOT SUCH PARTY HAD ADVANCE NOTICE OF THE POSSIBILITY OF SUCH DAMAGES.

Without limiting the generality of this Disclaimer stated above, the user of the contents of this Document is further notified that MIPI: (a) does not evaluate, test or verify the accuracy, soundness or credibility of the contents of this Document; (b) does not monitor or enforce compliance with the contents of this Document; and (c) does not certify, test, or in any manner investigate products or services or any claims of compliance with the contents of this Document. The use or implementation of the contents of this Document may involve or require the use of intellectual property rights ("IPR") including (but not limited to) patents, patent applications, or copyrights owned by one or more parties, whether or not Members of MIPI. MIPI does not make any search or investigation for IPR, nor does MIPI require or request the disclosure of any IPR or claims of IPR as respects the contents of this Document or otherwise. Questions pertaining to this document, or the terms or conditions of its provision, should be addressed to:

MIPI Alliance, Inc. c/o IEEE-ISTO 445 Hoes Lane Piscataway, NJ 08854 Attn: Board Secretary

Trademarks

OMAP, ICECrusher, and SmartReflex are trademarks of Texas Instruments.

Arm, Thumb, AMBA, and Cortex are registered trademarks of Arm Limited (or its subsidiaries) in the US and/or elsewhere.

Arm9, CoreSight, and Neon are trademarks of Arm Limited (or its subsidiaries) in the US and/or elsewhere.

Java is registered trademark of Oracle and/or its affiliates.

Bluetooth is a registered trademark of Bluetooth SIG, Inc. and is licensed to Texas Instruments.

Arteris is a trademark of Arteris, Inc.

SonicsLX, Sonics3220 are trademarks or registered trademarks of Sonics, Inc.

HD Radio is a trademark of iBiquity Digital Corporation.

MIPI is a registered trademark of the MIPI Alliance, Inc.

OneNAND is a trademark of SAMSUNG Electronics, Corporation.

All other trademarks are the property of their respective owners.

Introduction

This chapter introduces the features, subsystems, and architecture of the DRA78x (Jacinto6 RSP) family of devices.

Topic **Page**

1.1 DRA78x Overview

DRA78x is a high-performance, infotainment applications device, based on enhanced OMAP™ architecture, integrated on a 28-nm technology. The architecture is designed to deliver high-performance concurrencies for automotive co-processor, hybrid radio, and amplifier applications in a cost-effective solution, providing full scalability from the DRA75x (Jacinto6 EP and Jacinto6 Ex), DRA74x (Jacinto6), DRA72x (Jacinto6 Eco), and DRA71x (Jacinto6 Entry) families of infotainment processors.

NOTE: The supported set of features and peripherals is device part number dependent. Refer to device Data Manual, for more information.

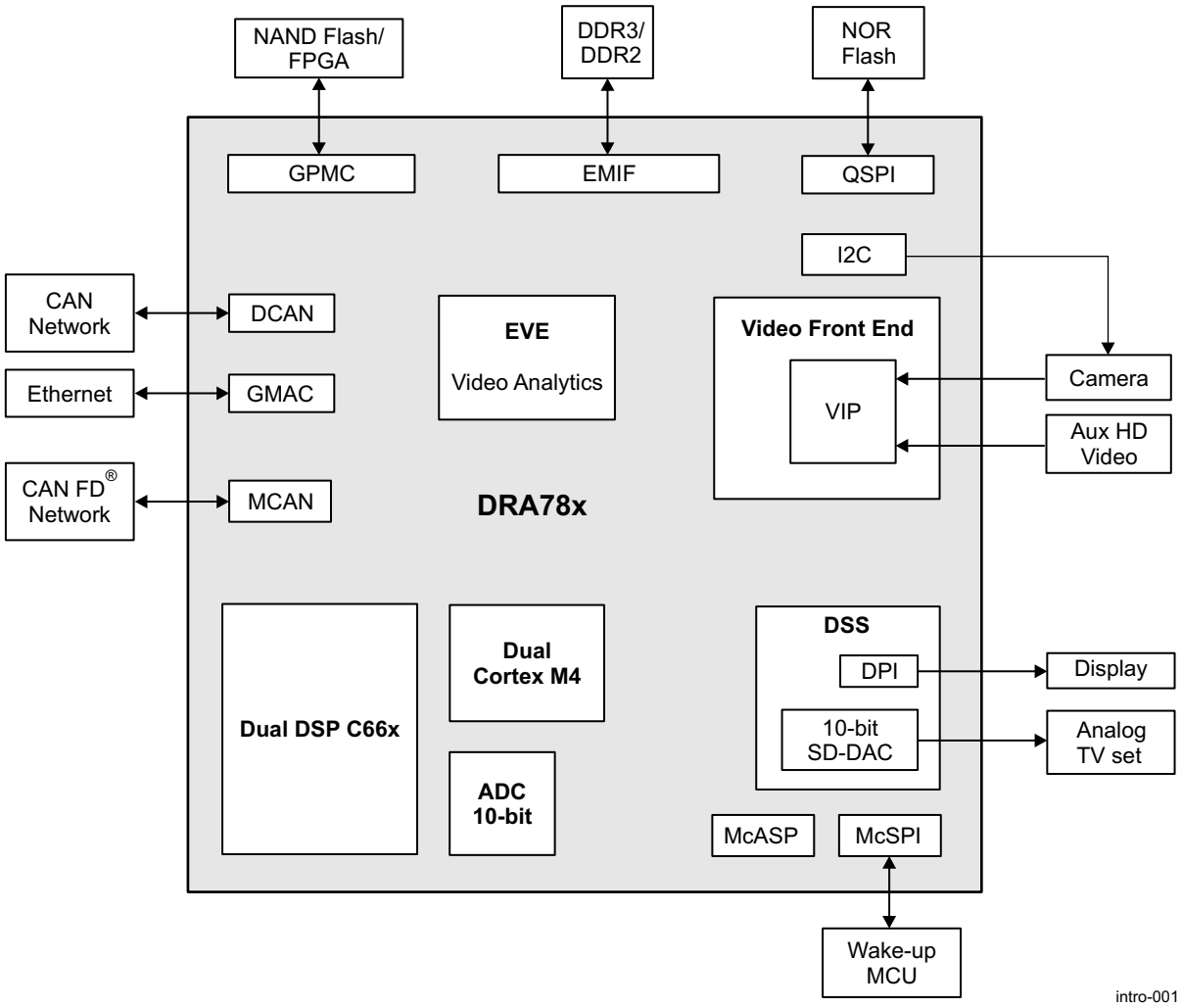
- The device is composed of the following main subsystems:
  - Up to two Digital Signal Processor (DSP) C66x subsystems
  - Embedded Vision Engine (EVE) accelerator subsystem
  - Dual Cortex®-M4 microprocessor unit (MCU) subsystem
  - Video Input capture Port (VIP)
  - Display Subsystem (DSS)
  - Debug Subsystem
- The device provides a rich set of connectivity peripherals, including among others:
  - Display parallel 24-bit output (for MIPI DPI 2.0, or BT.656/BT.1120 support), and one NTSC/PAL Standard Definition DAC (for composite video support)
  - Gigabit Ethernet (GEMAC) subsystem
  - One DCAN subsystem
  - One MCAN subsystem
  - One SDIO controller
  - One QSPI module
  - Three McASP modules
  - One PWMSS module
- The device also integrates:
  - On-chip memory
  - External memory interfaces
  - Memory management
  - Level 3 (L3_MAIN) and level 4 (L4) interconnects
  - System and serial control peripherals
  - PLLs (Phase Lock Loop) for internal clock generation
  - Audio Tracking Logic (ATL)
- The device includes comprehensive support for functional safety system requirements:
  - Dual core ECC-protected M4
  - ECC protected 32-bit DDR interface
  - Error Detection and Correction:
    - Parity bit per byte on C66x DSP L1 program cache and Single-Error Correction Dual-Error Detection (SECDED) on L2 memories on the DSP
    - SECDED on Large L3 on-chip RAM
  - Dedicated Memory management units per CPU (Cortex-M4 MCU, C66x DSP, EDMA) to implement freedom from interference
  - Memory protection units inside Cortex-M4 MCU subsystem to allow freedom from interference
  - Memory protection units internal to DSP cache controller to allow freedom from interference
  - Firewalls to enable isolation and in-advertent accesses
  - Real Time Interrupt (RTI) module supporting a windowed watchdog feature

- Two C66x DSP subsystems for redundant calculations
- MISR test schemes for EVE
- Parity protected EVE internal memories on minimum access size granularity
- Temperature monitoring sensors
- Embedded 8-channel ADC for system monitoring

1.2 DRA78x Environment

Figure 1-1 is an example for a non-exhaustive environment diagram for the DRA78x device.

Figure 1-1. DRA78x Environment Diagram

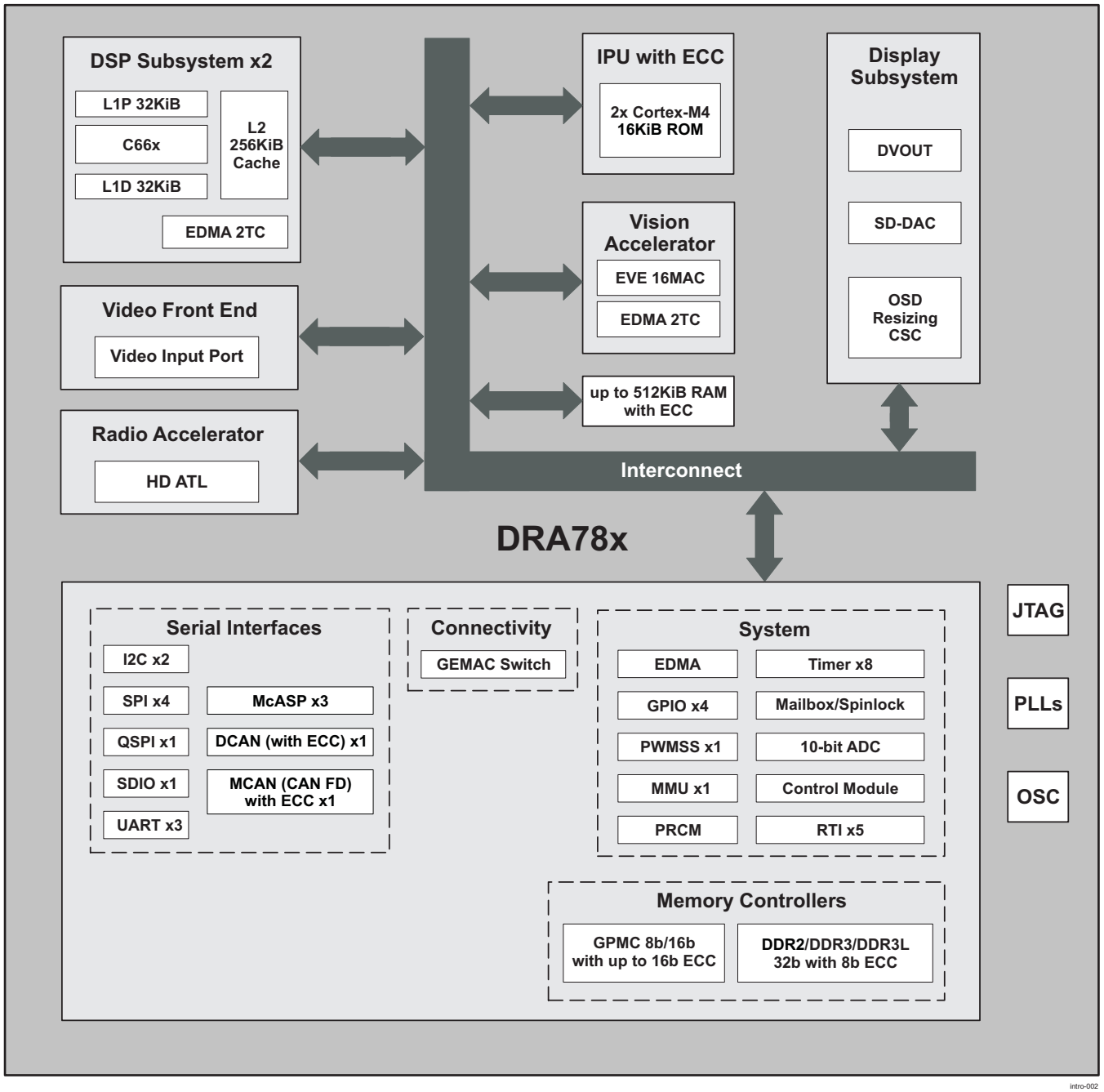


1.3 DRA78x Description

1.3.1 Block Diagram

Figure 1-2 shows the block diagram of the DRA78x device.

Figure 1-2. DRA78x Block Diagram



NOTE: The supported set of features and peripherals is device part number dependent. Refer to device Data Manual, for more information.

1.3.2 MCU Subsystem

The Dual Cortex-M4 MCU subsystem includes the following components:

- Two Cortex-M4 CPUs
- ARMv7E-M and Thumb-2 instruction set architectures
- Hardware division and single-cycle multiplication acceleration
- Dedicated INTC with up to 63 physical interrupt events with 16-level priority
- Two-level memory subsystem hierarchy
  - L1 Unicache RAM
    32-KiB shared cache memory
  - L2 ROM + RAM
    - 64-KiB RAM
    - 16-KiB bootable ROM
- ECC on all internal memories
- MMU with protection attributes for address translation
- Integrated power management
- Emulation feature embedded in the Cortex-M4

1.3.3 DSP Subsystem

There can be up to two DSP subsystems in the device. Each DSP subsystem contains the following submodules:

- TMS320C66x VLIW DSP core for audio processing, and general-purpose imaging and video processing. It extends the performance of existing C64x+ and C647x DSPs through enhancements and new features.
  - 32-KiB L1D and 32-KiB L1P cache or addressable SRAM
  - 288-KiB L2 cache
    - 256-KiB configurable as cache or SRAM
    - 32-KiB SRAM
- Enhanced direct memory access (EDMA) engine for video and audio data transfer
- Memory management units (MMU) for address management.
- Interrupt controller (INTC)
- Emulation capabilities

1.3.4 EVE Subsystem

The Embedded Vision Engine (EVE) module is a programmable imaging and vision processing engine allowing late in-development or post-silicon processing requirements to be met by addition of differentiating features in imaging and vision products. EVE also provides full featured debug support.

The EVE subsystem consists of the following main units:

- ARP32 scalar core with 32-KiB direct mapped program cache, and 32-KiB data memory
- VCOP vector core with 16 processing elements, 32-KiB working buffer, and four 16-KiB image buffers
- Integrated EDMA controller and memory management units

1.3.5 Video Input Capture

There is one Video Input Port (VIP) module in the device, providing video capture functions. The VIP module supports:

- Two separate 16-bit video ports for parallel RGB/YUV/RAW (or BT656/1120) data, up to 165 MHz
- Up to two separate 8-bit video ports for YUV/RAW (or BT656) data, up to 165 MHz
- Embedded Sync (multiplexed sources) and Discrete Sync (single source) data interface modes

- Color space conversion and scaling:
  - Up to 2047 pixels wide input with scaling
  - Up to 3840 pixels wide input - when chroma up/down sampling without scaling
  - Up to 4095 pixels wide input - without scaling and chroma up/down sampling
  - Maximum input resolution support is further limited by pixel clock and feature-dependent constraints
- Embedded DMA engine, supporting tiled (2D) and raster addressing

1.3.6 Display Subsystem

The display subsystem provides the logic and control signals required to interface the device system memory frame buffer (SDRAM) directly to a display or central applications processor with display compatible inputs. Various pixel processing capabilities are supported, such as: color space conversion, filtering, scaling, blending, color keying, etc. The display subsystem allows low-power display refresh and arbitration between normal and low-priority pipelines. The display subsystem consists of the following components:

- Display controller (DISPC): Reads and displays the encoded pixel data stored in memory and writes the output of one of the overlays or one of the pipelines into the system memory. The display controller supports the following components:
  - Two video pipelines, one graphic pipeline, and one write-back pipeline.
  - Graphic pipeline supports pixel formats such as: ARGB16-4444, RGB16-565, ARGB16-1555, ARGB32-8888, RGBA32-8888, RGB24-888, and BITMAP (1, 2, 4, or 8 bits per pixel). It implements also color-depth expansion.
  - Video pipelines support pixel formats such as: YUV (4:2:0/4:2:2), RGB565, ARGB16-4444, RGB16-565, ARGB16-1555, ARGB32-8888, RGBA32-8888, RGB24-888 and BITMAP (1, 2, 4, or 8 bits per pixel). They provide color conversion from YUV to RGB, and scaling up to 1080p (5-tap).
  - CLUT (1, 2, 4, or 8-bit BITMAP input, 24-bit RGB output) on video and graphics pipelines.
  - Write-back pipeline: YUV or RGB input from either one of video pipeline outputs, graphics pipeline output, or overlay managers outputs. Uses poly-phase filtering for independent horizontal and vertical resampling (upsampling and downsampling). It allows programmable color space conversion of RGB24 into YUV4:2:2-UYVY, YUV4:2:2-YUV2, or YUV4:2:0-NV12 or NV21, and selection of color-depth reduction from RGB24 to RGB16.
  - Own direct memory access (DMA) engine
  - Two overlay managers, receiving the outputs of the video and graphic pipelines, and applying overlay function using rules based on z-order, transparency color keys, and alpha blending values.
- NTSC/PAL Video Encoder (VENC) with the following main features:
  - Output to 10-bit on-chip standard definition video digital-to-analog converter (SD_DAC), supporting sampling rates up to 60 MSPS, and full-scale output voltage (1.2 Vpp with a 75Ohm load)
  - Support for square pixel sampling
  - Programmable horizontal synchronization, vertical timing, and waveforms
- Display outputs:
  - One parallel 24-bit RGB888 output, used for BT.656 or BT.1120, or MIPI® DPI 2.0 interface, supporting up to 165 MHz pixel clock video formats defined in CEA-861-E and VESA DMT standards.
  - One composite analog video output (DC or AC coupled), provided by the SD_DAC.
  - Parallel and SD_DAC interfaces can output the same video (composed by a single overlay manager in the Display Controller), but not simultaneously.

1.3.7 On-Chip Debug Support

The on-chip debug support has the following features:

- Multiprocessor debugging lets users control multiple CPU cores embedded in the device, such as:
  - Global starting and stopping of individual or multiple processors

  - Each processor can generate triggers that can be used to alter the execution flow of other processors
  - System clocking and power down
  - Interconnection of multiple devices
  - Channel triggering
- The following device cores can be debugged via Code Composer Studio (CCS):
  - Cortex-M4
  - DSP
  - EVE
- Target debugging, using IEEE1149.1 (JTAG®)
- Reduction of power consumption in normal operating mode

The debug subsystem includes:

- Generic TAP for emulation and test control (ICEPick-D)
- Debug access port (DAP)
- Embedded Trace Macro (ETM)
- Trace Port Interface Unit (TPIU)
- Embedded Trace Buffer (ETB)
- Emulation Pin Manager (EPM)
- Cross triggering (XTRIG)

The debug subsystem provides also:

- ICEMelter, for controlling the wake-up and power-down of the emulation power domain
- L3_INSTR CORE instrumentation interconnect
- OCP watch-point (OCP-WP), for monitoring L3 interconnect transaction when target transaction attributes match the user-defined attributes or trigger on external debug event
- Power-management events profiler (PM instrumentation)
- Clock-management events profiler (CM instrumentation)
- Statistics collector (performance probes)

1.3.8 On-Chip Memory

- The device includes one On-Chip Memory Controller (OCMC) with associated RAM with ECC, with total size of up to 512 KiB.

1.3.9 Memory Management

The memory management is performed from:

- Enhanced DMA (EDMA) controller supporting two simultaneous read and two simultaneous write physical channels, and up to 64 programmable logical channels.
- One memory management unit (MMU), with 4KiB, 64KiB, 1MiB, 16MiB programmable page sizes, and 32 entries TLB.

1.3.10 External Memory Interfaces

- One 32-bit DDR memory controller (EMIF):
  - Supporting DDR2/DDR3/DDR3L
  - 32-bit data path, one chip-select
  - The EMIF controller supports Single bit Error Correction and Dual Error Detection (SECDED)
    - SECDED supported in both 32-bit and 16-bit/Narrow mode
    - Programmable address ranges to define SECDED protected region
    - Parity bits calculated and stored on all writes to SECDED protected address region

    - Parity bits verified on all reads from SECDED protected address region
    - Statistics for 1-bit and 2-bit errors
- General-purpose memory controller (GPMC) supporting connection with:
  - Asynchronous SRAM memories
  - Asynchronous and synchronous NOR flash memories
  - NAND flash memories, with up to 16-bit ECC via the Error Location Module (ELM)
  - Pseudo-SRAM devices
- Quad SPI module, supporting 1 to 4 address bytes for SPI NOR flash, and up to 100 MHz single data rate

1.3.11 Power, Reset, and Clock Management

The PRCM module allows efficient control of clocks and power according to the required performance, and reduction of power consumption. The PRCM module is divided into:

- Power and reset management (PRM), based on the SmartReflex™ Class-0 framework with the following features:
  - Adaptive Voltage Scaling (AVS)
  - Dynamic clock gating
  - Dynamic power switching (DPS)
  - Dynamic frequency scaling
- Clock management (CM) for clock generation and distribution, allowing reduction of dynamic consumption.
- Temperature sensor

1.3.12 System and Connectivity Peripherals

The device supports a rich set of peripherals to provide flexible and high-speed interfacing and on-chip programming resources.

1.3.12.1 System Peripherals

- Eight general-purpose timers (two timer modules supporting 1-ms tick generation)
- One 32-kHz synchronization timer
- System control module, which contains registers for the following functions:
  - Static device configuration
  - Debug and observability
  - Status
  - Pad configuration
  - I/O configuration
  - eFuse logic
  - Analog function control
  - System boot decoding logic
- System Mailbox with 2 mailbox message queues
- SpinLock module for hardware semaphore between the MCU and DSP subsystems
- Inter-processor Communication Register
- Multiple general-purpose input/output (GPIO) modules
- One Audio Tracking Logic (ATL) module, containing four ATL instances for HD-Radio support and asynchronous sample rate conversion assistance. Each ATL instance supports error tracking between two reference signals, and generation of modulated clock (using software controlled cycle stealing).

1.3.12.2 Connectivity Peripherals

- One two-port switch (GEMAC), supporting 10, 100, or 1000 Mbps. It provides up to two external Ethernet ports and one internal CPPI interface port with Industrial Ethernet and 802.1ae support. Included 1.8- or 3.3-V RGMII.
- One DCAN controller (DCAN), supporting bitrates up to 1 Mbit/s, and compliant to the Controller Area Network (CAN) 2.0B protocol specification.
- One MCAN controller, supporting CAN FD® (Flexible Data rate) with up to 64 data bytes per frame and bitrates up to 5 Mbit/s. Compliant to ISO 11898-1:2015.
  - The availability of CAN FD feature is device part number dependent. Refer to device Data Manual for more information.

1.3.12.3 Serial Control Peripherals

- Three universal asynchronous receiver/transmitter (UART) modules as serial-communication interfaces, 16C750 compatible
- Four general-purpose multichannel serial peripheral interface (McSPI) modules
- Two I^2C controller modules with rates up to 400kbps
- One SDIO controller supporting 4-bit data bus width and SD3.0 physical layer with SDA3.00 standards, with embedded dual voltage I/Os (1.8 or 3V).
- Three Multichannel Audio Serial Ports (McASP), each supporting DIT for S/PDIF output, and providing connectivity to HD Radio tuners and audio ADC/DAC. In addition:
  - McASP1 supports up to 16 channels and independent TX/RX clock/sync domains.
  - McASP2 and McASP3 support up to 6 channels each, and independent TX/RX clock/sync domains.

1.4 DRA78x Family

The supported set of features and peripherals is device part number dependent. Refer to device Data Manual, for more information.

1.5 DRA78x Device Identification

Table 1-1 describes the identification registers.

The identification registers include the data registers listed in Table 1-2 and Table 1-4. These registers are read-only accessed ports that are programmed into eFuses FARM FROM.

Table 1-1. Device Identification Register Fields

Register Field	Alias Name	Physical Address	Address Offset
CTRL_CORE_STATUS[8:6] DEVICE_TYPE	DEVICE_TYPE	0x4A00 2134	0x134
CTRL_WKUP_STD_FUSE_DIE_ID_0[31:0] STD_FUSE_DIE_ID_0	DIE_ID[31:0]	0x4AE0 C200	0x200
CTRL_WKUP_ID_CODE[31:0] STD_FUSE_IDCODE	ID_CODE	0x4AE0 C204	0x204
CTRL_WKUP_STD_FUSE_DIE_ID_1[31:0] STD_FUSE_DIE_ID_1	DIE_ID[63:32]	0x4AE0 C208	0x208
CTRL_WKUP_STD_FUSE_DIE_ID_2[31:0] STD_FUSE_DIE_ID_2	DIE_ID[95:64]	0x4AE0 C20C	0x20C
CTRL_WKUP_STD_FUSE_DIE_ID_3[31:0] STD_FUSE_DIE_ID_3	DIE_ID[127:96]	0x4AE0 C210	0x210
CTRL_WKUP_STD_FUSE_PROD_ID_0[31:0] STD_FUSE_PROD_ID	PROD_ID	0x4AE0 C214	0x214

Table 1-2. DIE_ID

Register Field	Alias Name	Value
CTRL_WKUP_STD_FUSE_DIE_ID_0[31:0] STD_FUSE_DIE_ID_0	DIE_ID[31:0]	This register is for internal-use only.
CTRL_WKUP_STD_FUSE_DIE_ID_1[31:0] STD_FUSE_DIE_ID_1	DIE_ID[63:32]	This register is for internal-use only.
CTRL_WKUP_STD_FUSE_DIE_ID_2[31:0] STD_FUSE_DIE_ID_2	DIE_ID[95:64]	Part number identifier. See Table 1-3 for more information.
CTRL_WKUP_STD_FUSE_DIE_ID_3[31:0] STD_FUSE_DIE_ID_3	DIE_ID[127:96]	Reserved.

The part number identification data can be read in the [31:0] STD_FUSE_DIE_ID_2 bit-field of the CTRL_WKUP_STD_FUSE_DIE_ID_2 register. See Table 1-3 for more information.

Table 1-3. DRA78x Part Number Identifier

CTRL_WKUP_STD_FUSE_DIE_ID_2[31:0] STD_FUSE_DIE_ID_2	Value and Description	Comment
[31:24] Base PN	Refer to the Device Comparison section of a device-specific Data Manual (DM) for the Base PN value of a given part number.	Refer to device DM for details on the supported features for a given part number.
[23:19] Speed	1 (0x00) = A speed designator 2 (0x01) = B speed designator ... 25 (0x18) = Y speed designator 26 (0x19) = Z speed designator	Refer to device DM for supported speed grades for a given device, and the definition for supported speed grades.
[18] Temperature	0 = Reserved 1 = Automotive, -40°C to 125°C	Junction temperature.
[17:16] Package	1 = ABF 15×15 Others = Reserved	Refer to device DM for details on packaging.
[15:0] Reserved	Reserved	Reserved

The product type can be read in the value of the RAMP_SYSTEM bit field of the ID_CODE register (see Table 1-4). The silicon revision can be read in the value of bit field [4:0] of CTRL_WKUP_STD_FUSE_DIE_ID_3 register (see Table 1-5).

Table 1-4. ID_CODE

Register Field	Value	Comment
CTRL_WKUP_ID_CODE[31:28] RESERVED	N/A	Reserved
CTRL_WKUP_ID_CODE[27:12] RAMP_SYSTEM	See Table 1-5	Ramp system number
CTRL_WKUP_ID_CODE[11:1] TI_IDM	0x17	Manufacturer identity (TI)
CTRL_WKUP_ID_CODE[0] ONE	0x1	Always set to 1

Table 1-5, lists the device ramp system, ID_CODE, and silicon revision values.

Table 1-5. Device Identification Values

Silicon Type	Silicon Revision(1)	RAMP_SYSTEM	ID_CODE
DRA78x SR2.0	0x2	0xBB4C	0x2BB4C02F
DRA78x SR2.0A	0x3	0xBB4C	0x3BB4C02F

(1) Can be read in the CTRL_WKUP_STD_FUSE_DIE_ID_3 [4:0] register bit field.

The device type can be read in the PROD_ID register (see Table 1-6).

Table 1-6. PROD_ID

Register Field	Value	Comment
CTRL_WKUP_STD_FUSE_PROD_ID_0[7:0] DEVICE_TYPE	0xF0	Reads 0xF0 when device is a general-purpose (GP) device

The device type can be read also in the CTRL_CORE_STATUS register (see Table 1-7).

Table 1-7. DEVICE_TYPE

Register Field	Value	Comment
CTRL_CORE_STATUS[8:6] DEVICE_TYPE	0x3	Reads 0x3 when device is a general-purpose (GP) device

1.6 DRA78x Package Characteristics Overview

The package option for the device is: 15x15 FCBGA Lidded - ABF.

- Package Type: Lidded
  - Body: 15 × 15 mm
  - Technology: Ball Grid Array (BGA) package
  - Ball pitch: 0.65mm
  - Pins: 367

For more information on packaging refer to device Data Manual.

Memory Mapping

This chapter describes the memory mapping in the device.

Topic **Page**

2.1 Introduction

The memory map has the following features that are shared among the initiators:

- Memory space: General-Purpose Memory Controller (GPMC)
- Register spaces: Level 3 (L3_MAIN) and level 4 (L4) interconnects
- Dedicated spaces: EVE/IPU/DSP subsystems

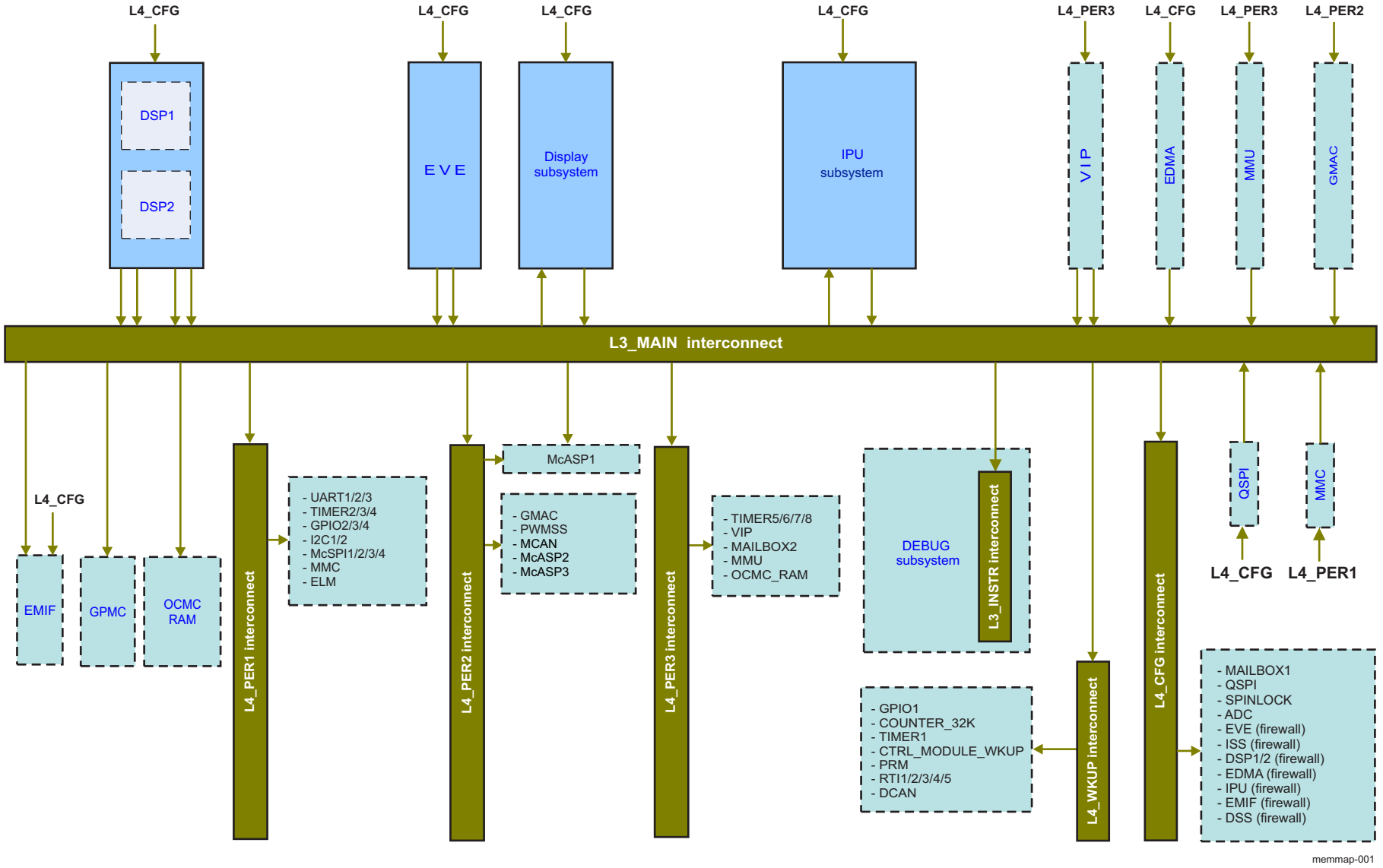
The GPMC and EMIF are dedicated to memory connection. The GPMC is used for NOR and NAND flash, and Static Random Access Memories (SRAMs). The EMIF is used for Synchronous Dynamic Random Access Memories (SDRAMs), such as DDR. For more information, see Section 10.2, *EMIF Controller*.

The L3_MAIN interconnect allows the sharing of resources, such as peripherals and external or on-chip memories, among all the initiators of the platform. The L4 interconnects control access to the peripherals.

Transfers across the platform between initiators and targets are physically conditioned by the chip interconnect and can be logically conditioned by firewalls. For more information about the intercommunication (L3_MAIN and L4 interconnects) and protection mechanisms implemented in the device, see Section 9.2, *L3_MAIN Interconnect*, and Section 9.3, *L4 Interconnects*.

Figure 2-1 shows the interconnect of the device, and the main modules and subsystems in the platform.

Figure 2-1. Interconnect Overview



2.2 L3_MAIN Memory Map

The memory space system is hierarchical: level 1 (L1), level 2 (L2), level 3 (L3_MAIN), and level 4 (L4). L1 and L2 are memories in the IPU and Digital Signal Processor (DSP) subsystems. L3_MAIN handles many types of data transfers, including data exchange with system on-chip/external memories. The chip-level interconnect, which consists of one L3_MAIN and number of L4 interconnects, enables communication among all modules and subsystems.

This section provides a global view of the memory mapping of the device at the L3_MAIN interconnect and describes the boot, GPMC, and SDRAM controller (SDRC) (EMIF) spaces.

The system memory mapping is flexible, with two levels of granularity for target address space allocation:

- L1: The four quarters are labeled Q0, Q1, Q2, and Q3. Each quarter corresponds to a 1-GiB address space (the total low-address space is 4 GiB, 32-bit).
- L2: Each quarter is divided into eight blocks of 32 MiB, with target spaces mapped in the blocks.

This organization allows the decoding of all target spaces based on the 7 most-significant bits (MSBs) of the 32-bit address, that is bits [31:25].

- Boot space:

  When booting from the on-chip ROM with the appropriate external sys_boot pin configuration, the lowest 1-MiB memory space [0x0000 0000–0x000F FFFF] is redirected to the on-chip boot ROM address space [0x4000 0000–0x400F FFFF].

  When booting from the GPMC, the memory space is part of the GPMC address space. At reset, the 0x0000 0000 address is available on chip-select 0 (CS0) for a memory size of 16 MiB.

  For more information about the sys_boot pins configuration, see Section 10.3, *General-Purpose Memory Controller*, and Chapter 25, *Initialization*.

- GPMC space:

  Eight independent GPMC chip-selects (CS0 to CS7) are available in the first quarter (Q0) of the addressing space to access NOR/NAND flash and SRAM. The chip-selects have a programmable start address and programmable size (up to 128 MiB) in a total memory space of (Q0) 1GiB, but limited now to 512 MiB.

- DDR/SDRAM (EMIF) space:

  Q2 addressing space is 1GiB and is dedicated to EMIF.

  Q3 addressing space is 1GiB and is dedicated to EMIF.

Table 2-1 describes the global memory map.

Table 2-1. L3_MAIN Memory Map

Quarter	Region Name	Start_Address (hex)	End_Address (hex)	Size	Description
Q0 (1GiB)	GPMC	0x0000_0000	0x1FFF_FFFF	512MiB	8/16-bit Ex(1)/R/W
	Reserved	0x2000_0000	0x3FFF_FFFF	512MiB	Reserved
Q1 (1GiB)	Reserved	0x4000_0000	0x402F_FFFF	3MiB	Reserved
	OCMC_RAM	0x4030_0000	0x4037_FFFF	512KiB	32-bit Ex(1)/R/W
	Reserved	0x4038_0000	0x407F_FFFF	4 512KiB	Reserved
	DSP1_L2_SRAM	0x4080_0000	0x4084_7FFF	288KiB	DSP1 L2 SRAM and cache. See Table 2-9.
	Reserved	0x4084_8000	0x40CF_FFFF	4832KiB	Reserved
	DSP1_SYSTEM	0x40D0_0000	0x40D0_0FFF	4KiB	DSP1 System MMR block
	DSP1_MMU0CFG	0x40D0_1000	0x40D0_1FFF	4KiB	DSP1 MMU0 configuration
	DSP1_MMU1CFG	0x40D0_2000	0x40D0_2FFF	4KiB	DSP1 MMU1 configuration
	DSP1_FW0CFG	0x40D0_3000	0x40D0_3FFF	4KiB	DSP1 Firewall 0 config
	DSP1_FW1CFG	0x40D0_4000	0x40D0_4FFF	4KiB	DSP1 Firewall 1 config

(1) Ex = Executable

Table 2-1. L3_MAIN Memory Map (continued)

Quarter	Region Name	Start_Address (hex)	End_Address (hex)	Size	Description
	DSP1_EDMA_TC0	0x40D0_5000	0x40D0_5FFF	4KiB	DSP1 EDMA Transfer Controller 0
	DSP1_EDMA_TC1	0x40D0_6000	0x40D0_6FFF	4KiB	DSP1 EDMA Transfer Controller 1
	DSP1_NoC	0x40D0_7000	0x40D0_7FFF	4KiB	DSP1 interconnect registers
	Reserved	0x40D0_8000	0x40D0_FFFF	32KiB	Reserved
	DSP1_EDMA_CC	0x40D1_0000	0x40D1_7FFF	32KiB	DSP1 EDMA Channel Controller
	Reserved	0x40D1_8000	0x40DF_FFFF	928KiB	Reserved
	DSP1_L1P_SRAM	0x40E0_0000	0x40E0_7FFF	32KiB	DSP1 L1P Cache/RAM
	Reserved	0x40E0_8000	0x40EF_FFFF	992KiB	Reserved
	DSP1_L1D_SRAM	0x40F0_0000	0x40F0_7FFF	32KiB	DSP1 L1D Cache/RAM
	Reserved	0x40F0_8000	0x40FF_FFFF	992KiB	Reserved
	DSP2_L2_SRAM	0x4100_0000	0x4104_7FFF	288KiB	DSP2 L2 SRAM and cache. See Table 2-9.
	Reserved	0x4104_8000	0x414F_FFFF	4832KiB	Reserved
	DSP2_SYSTEM	0x4150_0000	0x4150_0FFF	4KiB	DSP2 System MMR block
	DSP2_MMU0CFG	0x4150_1000	0x4150_1FFF	4KiB	DSP2 MMU0 configuration
	DSP2_MMU1CFG	0x4150_2000	0x4150_2FFF	4KiB	DSP2 MMU1 configuration
	DSP2_FW0CFG	0x4150_3000	0x4150_3FFF	4KiB	DSP2 Firewall 0 config
	DSP2_FW1CFG	0x4150_4000	0x4150_4FFF	4KiB	DSP2 Firewall 1 config
	DSP2_EDMA_TC0	0x4150_5000	0x4150_5FFF	4KiB	DSP2 EDMA Transfer Controller 0
	DSP2_EDMA_TC1	0x4150_6000	0x4150_6FFF	4KiB	DSP2 EDMA Transfer Controller 1
	DSP2_NoC	0x4150_7000	0x4150_7FFF	4KiB	DSP2 interconnect registers
	Reserved	0x4150_8000	0x4150_FFFF	32KiB	Reserved
	DSP2_EDMA_CC	0x4151_0000	0x4151_7FFF	32KiB	DSP2 EDMA Channel Controller
	Reserved	0x4151_8000	0x415F_FFFF	928KiB	Reserved
	DSP2_L1P_SRAM	0x4160_0000	0x4160_7FFF	32KiB	DSP2 L1P Cache/RAM
	Reserved	0x4160_8000	0x416F_FFFF	992KiB	Reserved
	DSP2_L1D_SRAM	0x4170_0000	0x4170_7FFF	32KiB	DSP2 L1D Cache/RAM
	Reserved	0x4170_8000	0x417F_FFFF	992KiB	Reserved
	OCMC_RAM_CBUF	0x4180_0000	0x41FF_FFFF	8MiB	OCMC RAM CBUF virtual address space (Bit 31 needs to be set on the OCMC data interface)
	EVE	0x4200_0000	0x420F_FFFF	1MiB	EVE configuration space
	Reserved	0x4210_0000	0x432F_FFFF	18MiB	Reserved
	EDMA_TPCC	0x4330_0000	0x433F_FFFF	1MiB	EDMA TPCC configuration space
	EDMA_TC0	0x4340_0000	0x434F_FFFF	1MiB	EDMA TPTC1 configuration space
	EDMA_TC1	0x4350_0000	0x435F_FFFF	1MiB	EDMA TPTC2 configuration space
	Reserved	0x4360_0000	0x43FF_FFFF	10MiB	Reserved
	L3_MAIN_SN	0x4400_0000	0x457F_FFFF	24MiB	L3 configuration registers (Service Network)
	McASP1	0x4580_0000	0x45BF_FFFF	4MiB	McASP1 configuration space
	Reserved	0x45C0_0000	0x47FF_FFFF	36MiB	Reserved
	L4_PER1	0x4800_0000	0x481F_FFFF	2MiB	L4_PER1 domain. See Table 2-5
	Reserved	0x4820_0000	0x483F_FFFF	2MiB	Reserved
	L4_PER2	0x4840_0000	0x485F_FFFF	2MiB	L4_PER2 domain. See Table 2-6
	Reserved	0x4860_0000	0x487F_FFFF	2MiB	Reserved
	L4_PER3	0x4880_0000	0x48BF_FFFF	4MiB	L4_PER3 domain. See Table 2-7
	Reserved	0x48C0_0000	0x49FF_FFFF	20MiB	Reserved
	L4_CFG	0x4A00_0000	0x4ADF_FFFF	14MiB	L4_CFG domain. See Table 2-3
	L4_WKUP	0x4AE0_0000	0x4AFF_FFFF	2MiB	L4_WKUP domain. See Table 2-4
	Reserved	0x4B00_0000	0x4B2F_FFFF	3MiB	Reserved

Table 2-1. L3_MAIN Memory Map (continued)

Quarter	Region Name	Start_Address (hex)	End_Address (hex)	Size	Description
	QSPI_ADDRSP0	0x4B30_0000	0x4B3F_FFFF	1MiB	QSPI MMR space (Maddrspace 0)
	Reserved	0x4B40_0000	0x4B7F_FFFF	4MiB	Reserved
	Reserved	0x4B80_0000	0x4BAF_FFFF	3MiB	Reserved
	Reserved	0x4BB0_0000	0x4BBF_FFFF	1MiB	Reserved
	Reserved	0x4BC0_0000	0x4BCF_FFFF	1MiB	Reserved
	Reserved	0x4BD0_0000	0x4BDF_FFFF	3MiB	Reserved
	EMIF	0x4C00_0000	0x4CFF_FFFF	16MiB	EMIF configuration registers
	Reserved	0x4D00_0000	0x4FFF_FFFF	48MiB	Reserved
	GPMC	0x5000_0000	0x50FF_FFFF	16MiB	GPMC configuration registers
	Reserved	0x5100_0000	0x51FF_FFFF	16MiB	Reserved
	ISS	0x5200_0000	0x53FF_FFFF	32MiB	ISS
	L3_INSTR	0x5400_0000	0x547F_FFFF	8MiB	Emulation domain. See Section 2.2.1
	CT_TBR	0x5480_0000	0x54FF_FFFF	8MiB	Emulation domain. See Section 2.2.1
	Reserved	0x5500_0000	0x57FF_FFFF	49MiB	Reserved
	DSS	0x5800_0000	0x587F_FFFF	8MiB	DSS domain
	IPU_ROM	0x5880_0000	0x58FF_FFFF	16KiB	IPU_ROM
	Reserved	0x5880_4000	0x5881_FFFF	112KiB	Reserved
	IPU_RAM	0x5882_0000	0x5882_FFFF	64KiB	IPU_RAM
	Reserved	0x5883_0000	0x5887_FFFF	320KiB	Reserved
	Reserved	0x5888_0000	0x5888_07FF	2KiB	Reserved
	IPU_UNICACHE_MMU	0x5888_0800	0x5888_0FFF	2KiB	IPU_UNICACHE_MMU config registers
	Reserved	0x5888_1000	0x5888_1FFF	4KiB	Reserved
	IPU_MMU	0x5888_2000	0x5888_2FFF	4KiB	IPU_MMU config registers
	Reserved	0x5888_3000	0x5BFF_FFFF	56MiB	Reserved
	QSPI_ADDRSP1	0x5C00_0000	0x5FFF_FFFF	64MiB	QSPI CS0/CS1/CS2/CS3 space (Maddrspace 1)
	Reserved	0x6000_0000	0x7FFF_FFFF	512MiB	Reserved
Q2 (1GiB)	DDR-SDRAM address space				
	EMIF_SDRAM	0x8000_0000	0xBFFF_FFFF	1GiB	EMIF: Access to DDR
Q3 (1GiB)	EMIF_SDRAM	0xC000_0000	0xFFFF_FFFF	1GiB	EMIF: Access to DDR

2.2.1 L3_INSTR Memory Map

The L3_INSTR interconnect is a 8-MiB space composed of the L3_INSTR interconnect configuration registers and module registers.

Table 2-2 describes the mapping of the registers for the L3_INSTR interconnect.

Table 2-2. L3_INSTR Memory Map

Region name	Start_address (hex)	End_address (hex)	Size	Description
CT_STM_ADD_SP_0	0x5400_0000	0x540F_FFFF	1MiB	MIPI_STM - System Trace (address space 0)
CT_STM_ADD_SP_1	0x5410_0000	0x5413_FFFF	256KiB	MIPI_STM - System Trace (address space 1)
Reserved	0x5414_0000	0x5415_FFFF	132KiB	Reserved
DRM	0x5416_0000	0x5416_0FFF	4KiB	DRM (OCP) - Debug Register Mapping
CT_STM_CONF_PORT	0x5416_1000	0x5416_1FFF	4KiB	MIPI_STM(OCP) configuration port - System Trace

Table 2-2. L3_INSTR Memory Map (continued)

Region name	Start_address (hex)	End_address (hex)	Size	Description
Reserved	0x5416_2000	0x5416_2FFF	4KiB	Reserved
CS_TPIU	0x5416_3000	0x5416_3FFF	4KiB	CS_TPIU (APBv3) - Trace Port Interface Unit
DEBUGSS_CS_TF_1	0x5416_4000	0x5416_4FFF	4KiB	CS_TF (APBv3) - Trace Funnel for DEBUGSS
Reserved	0x5416_5000	0x5416_6FFF	8KiB	Reserved
CT_TBR	0x5416_7000	0x5416_7FFF	4KiB	C-Tools Trace Buffer
CT_UART	0x5416_8000	0x5416_8FFF	4KiB	C-Tools UART
DEBUGSS_CS_CTI	0x5416_9000	0x5416_9FFF	4KiB	Cross Triggering Interface
Reserved	0x5416_C000	0x5417_1FFF	32KiB	Reserved
L4_CFG_EMU	0x5417_2000	0x5417_2FFF	4KiB	Interconnect registers
Reserved	0x5417_3000	0x5417_FFFF	52KiB	Reserved
L3_INSTR_EMU	0x5418_0000	0x5418_0FFF	4KiB	Interconnect registers
Reserved	0x5418_1000	0x547F_FFFF	6652KiB	Reserved

2.3 L4 Memory Map

The L4 interconnects handle transfers with peripherals. The L4 interconnect comprises the following interconnects:

- L4_CFG
- L4_WKUP
- L4_PER1
- L4_PER2
- L4_PER3

The L4 interconnect can be configured to tune the access according to the characteristics of each module.

The following sections describe the register mapping of the L4 interconnect. Software configures these registers.

2.3.1 L4_CFG Memory Map

The L4_CFG interconnect is a 12-MiB space composed of the L4_CFG interconnect configuration registers and the module registers.

Table 2-3 describes the mapping of the registers for the L4_CFG interconnect.

NOTE: All memory spaces described as modules provide direct access to module registers outside the L4_CFG interconnect. All other accesses are internal to the L4_CFG interconnect.

Table 2-3. L4_CFG Memory Map

Module name	Region Name	Start_address (hex)	End_address (hex)	Size	Description
L4_CFG	L4_CFG_AP	0x4A00_0000	0x4A00_07FF	2KiB	Address protection
	L4_CFG_LA	0x4A00_0800	0x4A00_0FFF	2KiB	Link agent
	L4_CFG_IA_IP0	0x4A00_1000	0x4A00_1FFF	4KiB	Initiator port
CTRL_MODULE_CORE	TP_CTRL_MODULE_CORE_TARG	0x4A00_2000	0x4A00_3FFF	8KiB	Module target port
	TA_CTRL_MODULE_CORE_TARG	0x4A00_4000	0x4A00_4FFF	4KiB	L4 target agent
CM_CORE_AON	TP_CM_CORE_AON_TARG	0x4A00_5000	0x4A00_5FFF	4KiB	Module target port
	TA_CM_CORE_AON_TARG	0x4A00_6000	0x4A00_6FFF	4KiB	L4 target agent
Reserved	Reserved	0x4A00_7000	0x4A00_7FFF	4KiB	Reserved
CM_CORE	TP_CM_CORE_TARG	0x4A00_8000	0x4A00_9FFF	8KiB	Module target port
	TA_CM_CORE_TARG	0x4A00_A000	0x4A00_AFFF	4KiB	L4 target agent
Reserved	Reserved	0x4A00_B000	0x4A0D_CFFF	860KiB	Reserved
SMARTREFLEX_CORE	TP_SMARTREFLEX_CORE_TARG	0x4A0D_D000	0x4A0D_DFFF	4KiB	Module target port
	TA_SMARTREFLEX_CORE_TARG	0x4A0D_E000	0x4A0D_EFFF	4KiB	L4 target agent
Reserved	Reserved	0x4A0D_F000	0x4A0D_FFFF	300KiB	Reserved
EFUSE_CTRL_CUST	TP_EFUSE_CTRL_CUST_TARG	0x4A0E_0000	0x4A0E_0FFF	4KiB	Module target port
	TA_EFUSE_CTRL_CUST_TARG	0x4A0E_1000	0x4A0E_1FFF	4KiB	L4 target agent
Reserved	Reserved	0x4A0E_2000	0x4A0F_3FFF	8KiB	Reserved
MAILBOX1	TP_MAILBOX1_TARG	0x4A0F_4000	0x4A0F_4FFF	4KiB	Module target port
	TA_MAILBOX1_TARG	0x4A0F_5000	0x4A0F_5FFF	4KiB	L4 target agent

Table 2-3. L4_CFG Memory Map (continued)

Module name	Region Name	Start_address (hex)	End_address (hex)	Size	Description
SPINLOCK	TP_SPINLOCK_TARG	0x4A0F_6000	0x4A0F_6FFF	4KiB	Module target port
	TA_SPINLOCK_TARG	0x4A0F_7000	0x4A0F_7FFF	4KiB	L4 target agent
Reserved	Reserved	0x4A0F_8000	0x4A10_1FFF	40KiB	Reserved
OCP_WP_NOC	TP_OCP_WP_NOC_TARG	0x4A10_2000	0x4A10_2FFF	4KiB	Module target port
	TA_OCP_WP_NOC_TARG	0x4A10_3000	0x4A10_3FFF	4KiB	L4 target agent
Reserved	Reserved	0x4A10_4000	0x4A10_7FFF	16KiB	Reserved
IEEE1500	TP_IEEE1500_2_OCP_TARG	0x4A10_8000	0x4A10_8FFF	4KiB	Module target port
	TA_IEEE1500_2_OCP_TARG	0x4A10_9000	0x4A10_9FFF	4KiB	L4 target agent
Reserved	Reserved	0x4A10_A000	0x4A15_0FFF	290KiB	Reserved
EVE	TP_EVE_FW_CFG_TARG	0x4A15_1000	0x4A15_1FFF	4KiB	Module target port
	TA_EVE_FW_CFG_TARG	0x4A15_2000	0x4A15_2FFF	4KiB	L4 target agent
Reserved	Reserved	0x4A15_3000	0x4A15_AFFF	32KiB	Reserved
IPU	TP_IPU_FW_CFG_TARG	0x4A15_B000	0x4A15_BFFF	4KiB	Module target port
	TA_IPU_FW_CFG_TARG	0x4A15_C000	0x4A15_CFFF	4KiB	L4 target agent
Reserved	Reserved	0x4A15_D000	0x4A16_0FFF	16KiB	Reserved
EDMA_TPCC	TP_EDMA_TPCC_FW_CFG_TARG	0x4A16_1000	0x4A16_1FFF	4KiB	Module target port
	TA_EDMA_TPCC_FW_CFG_TARG	0x4A16_2000	0x4A16_2FFF	4KiB	L4 target agent
EDMA_TC0	TP_EDMA_TC0_FW_CFG_TARG	0x4A16_3000	0x4A16_3FFF	4KiB	Module target port
	TA_EDMA_TC0_FW_CFG_TARG	0x4A16_4000	0x4A16_4FFF	4KiB	L4 target agent
Reserved	Reserved	0x4A16_5000	0x4A16_6FFF	8KiB	Reserved
McASP	TP_MCASP_FW_CFG_TARG	0x4A16_7000	0x4A16_7FFF	4KiB	Module target port (applies to all McASP modules)
	TA_MCASP_FW_CFG_TARG	0x4A16_8000	0x4A16_8FFF	4KiB	L4 target agent (applies to all McASP modules)
Reserved	Reserved	0x4A16_9000	0x4A16_CFFF	16KiB	Reserved
TSC_ADC	TP_TSC_ADC_FW_CFG_TARG	0x4A16_D000	0x4A16_DFFF	4KiB	Module target port
	TA_TSC_ADC_FW_CFG_TARG	0x4A16_E000	0x4A16_EFFF	4KiB	L4 target agent
Reserved	Reserved	0x4A16_F000	0x4A17_0FFF	8KiB	Reserved
DSP1	TP_DSP1_FW_CFG_TARG	0x4A17_1000	0x4A17_1FFF	4KiB	Module target port
	TA_DSP1_FW_CFG_TARG	0x4A17_2000	0x4A17_2FFF	4KiB	L4 target agent
DSP2	TP_DSP2_FW_CFG_TARG	0x4A17_3000	0x4A17_3FFF	4KiB	Module target port
	TA_DSP2_FW_CFG_TARG	0x4A17_4000	0x4A17_4FFF	4KiB	L4 target agent
Reserved	Reserved	0x4A17_5000	0x4A17_8FFF	16KiB	Reserved
QSPI	TP_QSPI_FW_CFG_TARG	0x4A17_9000	0x4A17_9FFF	4KiB	Module target port
	TA_QSPI_FW_CFG_TARG	0x4A17_A000	0x4A17_AFFF	4KiB	L4 target agent
Reserved	Reserved	0x4A17_B000	0x4A18_2FFF	32KiB	Reserved
SMARTREFLEX_DSPEVE	TP_SMARTREFLEX_DSPEVE_TARG	0x4A18_3000	0x4A18_3FFF	4KiB	Module target port
	TA_SMARTREFLEX_DSPEVE_TARG	0x4A18_4000	0x4A18_4FFF	4KiB	L4 target agent
Reserved	Reserved	0x4A18_5000	0x4A20_BFFF	553KiB	Reserved

Table 2-3. L4_CFG Memory Map (continued)

Module name	Region Name	Start_address (hex)	End_address (hex)	Size	Description
EMIF	TP_EMIF_FW _CFG_TARG	0x4A20_C000	0x4A20_CFFF	4KiB	Module target port
	TA_EMIF_FW _CFG_TARG	0x4A20_D000	0x4A20_DFFF	4KiB	L4 target agent
Reserved	Reserved	0x4A20_E000	0x4A20_EFFF	8KiB	Reserved
GPMC	TP_GPMC_FW _CFG_TARG	0x4A21_0000	0x4A21_0FFF	4KiB	Module target port
	TA_GPMC_FW _CFG_TARG	0x4A21_1000	0x4A21_1FFF	4KiB	L4 target agent
OCMC_RAM	TP_OCMC_RAM_FW_CFG_TA RG	0x4A21_2000	0x4A21_2FFF	4KiB	Module target port
	TA_OCMC_RAM_FW_CFG_TA RG	0x4A21_3000	0x4A21_3FFF	4KiB	L4 target agent
Reserved	Reserved	0x4A21_4000	0x4A21_BFFF	32KiB	Reserved
DSS	TP_DSS_FW_CFG_TARG	0x4A21_C000	0x4A21_CFFF	4KiB	Module target port
	TA_DSS_FW_CFG_TARG	0x4A21_D000	0x4A21_DFFF	4KiB	L4 target agent
Reserved	Reserved	0x4A21_E000	0x4A22_3FFF	24KiB	Reserved
DEBUGSS	TP_DEBUGSS_CT_TBR _FW_CFG_TARG	0x4A22_4000	0x4A22_4FFF	4KiB	Module target port. See Section 2.2.1
	TA_DEBUGSS_CT_TBR _FW_CFG_TARG	0x4A22_5000	0x4A22_5FFF	4KiB	L4 target agent
L3_INSTR	TP_L3_INSTR_FW _CFG_TARG	0x4A22_6000	0x4A22_6FFF	4KiB	Module target port
	TA_L3_INSTR_FW _CFG_TARG	0x4A22_7000	0x4A22_7FFF	4KiB	L4 target agent
Reserved	Reserved	0x4A22_8000	0x4ADF_FFFF	12MiB	Reserved

2.3.2 L4_WKUP Memory Map

The L4_WKUP interconnect is a 256-KiB space composed of the L4_WKUP interconnect configuration registers and the module registers.

Table 2-4 describes the mapping of the registers for the L4_WKUP interconnect.

NOTE: All memory spaces described as modules provide direct access to module registers outside the L4_WKUP interconnect. All other accesses are internal to the L4_WKUP interconnect.

Table 2-4. L4_WKUP Memory Map

Module name	Region name	Start_address (hex)	End_address (hex)	Size	Description
L4_WKUP	L4_WKUP_AP	0x4AE0_0000	0x4AE0_07FF	2KiB	Address protection
	L4_WKUP_LA	0x4AE0_0800	0x4AE0_0FFF	2KiB	Link agent
	L4_WKUP_IA_IP0	0x4AE0_1000	0x4AE0_1FFF	4KiB	Initiator port
Reserved	Reserved	0x4AE0_2000	0x4AE0_3FFF	8KiB	Reserved
COUNTER_32K	TP_COUNTER_32K _TARG	0x4AE0_4000	0x4AE0_4FFF	4KiB	Module target port
	TA_COUNTER_32K _TARG	0x4AE0_5000	0x4AE0_5FFF	4KiB	L4 target agent
PRM	TP_PRM_TARG	0x4AE0_6000	0x4AE0_7FFF	8KiB	Module target port
	TA_PRM_TARG	0x4AE0_8000	0x4AE0_8FFF	4KiB	L4 target agent
Reserved	Reserved	0x4AE0_9000	0x4AE0_BFFF	12KiB	Reserved

Table 2-4. L4_WKUP Memory Map (continued)

Module name	Region name	Start_address (hex)	End_address (hex)	Size	Description
CTRL_MODULE_WKUP	TP_CTRL_MODULE_WKUP_TARG	0x4AE0_C000	0x4AE0_CFFF	4KiB	Module target port
	TA_CTRL_MODULE_WKUP_TARG	0x4AE0_D000	0x4AE0_DFFF	4KiB	L4 target agent
Reserved	Reserved	0x4AE0_E000	0x4AE0_FFFF	8KiB	Reserved
GPIO1	TP_GPIO1_TARG	0x4AE1_0000	0x4AE1_0FFF	4KiB	Module target port
	TA_GPIO1_TARG	0x4AE1_1000	0x4AE1_1FFF	4KiB	L4 target agent
Reserved	Reserved	0x4AE1_2000	0x4AE1_7FFF	24KiB	Reserved
TIMER1	TP_TIMER1_TARG	0x4AE1_8000	0x4AE1_8FFF	4KiB	Module target port
	TA_TIMER1_TARG	0x4AE1_9000	0x4AE1_9FFF	4KiB	L4 target agent
Reserved	Reserved	0x4AE1_A000	0x4AE3_0FFF	94KiB	Reserved
RTI1	TP_RTI1_TARG	0x4AE3_1000	0x4AE3_1FFF	4KiB	Module target port
	TA_RTI1_TARG	0x4AE3_2000	0x4AE3_2FFF	4KiB	L4 target agent
RTI2	TP_RTI2_TARG	0x4AE3_3000	0x4AE3_3FFF	4KiB	Module target port
	TA_RTI2_TARG	0x4AE3_4000	0x4AE3_4FFF	4KiB	L4 target agent
RTI3	TP_RTI3_TARG	0x4AE3_5000	0x4AE3_5FFF	4KiB	Module target port
	TA_RTI3_TARG	0x4AE3_6000	0x4AE3_6FFF	4KiB	L4 target agent
RTI4	TP_RTI4_TARG	0x4AE3_7000	0x4AE3_7FFF	4KiB	Module target port
	TA_RTI4_TARG	0x4AE3_8000	0x4AE3_8FFF	4KiB	L4 target agent
RTI5	TP_RTI5_TARG	0x4AE3_9000	0x4AE3_9FFF	4KiB	Module target port
	TA_RTI5_TARG	0x4AE3_A000	0x4AE3_AFFF	4KiB	L4 target agent
Reserved	Reserved	0x4AE3_B000	0x4AE3_BFFF	4KiB	Reserved
DCAN	TP_DCAN1_TARG	0x4AE3_C000	0x4AE3_DFFF	8KiB	Module target port
	TA_DCAN1_TARG	0x4AE3_E000	0x4AE3_EFFF	4KiB	L4 target agent
Reserved	Reserved	0x4AE3_F000	0x4AFF_FFFF	1796KiB	Reserved

NOTE: 8- and 16-bit peripherals are aligned on 32-bit address boundaries.

2.4 L4_PER Memory Map

The L4_PER interconnect has three memory spaces:

- L4_PER1 memory space (Table 2-5)
- L4_PER2 memory space (Table 2-6)
- L4_PER3 memory space (Table 2-7)

The L4_PER interconnects are composed of the L4_PER interconnect confguration registers and the module registers.

NOTE: All memory spaces described as modules provide direct access to the module registers outside the L4_PER interconnects. All other accesses are internal to the L4_PER interconnects.

2.4.1 L4_PER1 Memory Space Mapping

Table 2-5 describes the mapping of the registers for the L4_PER1 interconnect.

Table 2-5. L4_PER1 Memory Map

Module name	Region name	Start_address (hex)	End_address (hex)	Size	Description
L4_PER1	L4_PER1_AP	0x4800_0000	0x4800_07FF	2KiB	Address protection
	L4_PER1_LA	0x4800_0800	0x4800_0FFF	2KiB	Link agent
	L4_PER1_IA_IP0	0x4800_1000	0x4800_13FF	1KiB	Initiator port 0
	L4_PER1_IA_IP1	0x4800_1400	0x4800_17FF	1KiB	Initiator port 1
	L4_PER1_IA_IP2	0x4800_1800	0x4800_1BFF	1KiB	Initiator port 2
	L4_PER1_IA_IP3	0x4800_1C00	0x4800_1FFF	1KiB	Initiator port 3
Reserved	Reserved	0x4800_1C00	0x4801_FFFF	121KiB	Reserved
UART3	TP_UART3_TARG	0x4802_0000	0x4802_0FFF	4KiB	Module target port
	TA_UART3_TARG	0x4802_1000	0x4802_1FFF	4KiB	L4 target agent
Reserved	Reserved	0x4802_2000	0x4803_1FFF	64KiB	Reserved
TIMER2	TP_TIMER2_TARG	0x4803_2000	0x4803_2FFF	4KiB	Module target port
	TA_TIMER2_TARG	0x4803_3000	0x4803_3FFF	4KiB	L4 target agent
TIMER3	TP_TIMER3_TARG	0x4803_4000	0x4803_4FFF	4KiB	Module target port
	TA_TIMER3_TARG	0x4803_5000	0x4803_5FFF	4KiB	L4 target agent
TIMER4	TP_TIMER4_TARG	0x4803_6000	0x4803_6FFF	4KiB	Module target port
	TA_TIMER4_TARG	0x4803_7000	0x4803_7FFF	4KiB	L4 target agent
Reserved	Reserved	0x4803_8000	0x4805_4FFF	116KiB	Reserved
GPIO2	TP_GPIO2_TARG	0x4805_5000	0x4805_5FFF	4KiB	Module target port
	TA_GPIO2_TARG	0x4805_6000	0x4805_6FFF	4KiB	L4 target agent
GPIO3	TP_GPIO3_TARG	0x4805_7000	0x4805_7FFF	4KiB	Module target port
	TA_GPIO3_TARG	0x4805_8000	0x4805_8FFF	4KiB	L4 target agent
GPIO4	TP_GPIO4_TARG	0x4805_9000	0x4805_9FFF	4KiB	Module target port
	TA_GPIO4_TARG	0x4805_A000	0x4805_AFFF	4KiB	L4 target agent
Reserved	Reserved	0x4805_B000	0x4806_9FFF	52KiB	Reserved
UART1	TP_UART1_TARG	0x4806_A000	0x4806_AFFF	4KiB	Module target port
	TA_UART1_TARG	0x4806_B000	0x4806_BFFF	4KiB	L4 target agent
UART2	TP_UART2_TARG	0x4806_C000	0x4806_CFFF	4KiB	Module target port
	TA_UART2_TARG	0x4806_D000	0x4806_DFFF	4KiB	L4 target agent
Reserved	Reserved	0x4806_E000	0x4806_FFFF	8KiB	Reserved
I2C1	TP_I2C1_TARG	0x4807_0000	0x4807_0FFF	4KiB	Module target port
	TA_I2C1_TARG	0x4807_1000	0x4807_1FFF	4KiB	L4 target agent
I2C2	TP_I2C2_TARG	0x4807_2000	0x4807_2FFF	4KiB	Module target port
	TA_I2C2_TARG	0x4807_3000	0x4807_3FFF	4KiB	L4 target agent
Reserved	Reserved	0x4807_4000	0x4807_7FFF	16KiB	Reserved
ELM	TP_ELM_TARG	0x4807_8000	0x4807_8FFF	4KiB	Module target port
	TA_ELM_TARG	0x4807_9000	0x4807_9FFF	4KiB	L4 target agent
Reserved	Reserved	0x4807_A000	0x4809_7FFF	122KiB	Reserved
McSPI1	TP_MCSPI1_TARG	0x4809_8000	0x4809_8FFF	4KiB	Module target port
	TA_MCSPI1_TARG	0x4809_9000	0x4809_9FFF	4KiB	L4 target agent
McSPI2	TP_MCSPI2_TARG	0x4809_A000	0x4809_AFFF	4KiB	Module target port
	TA_MCSPI2_TARG	0x4809_B000	0x4809_BFFF	4KiB	L4 target agent
Reserved	Reserved	0x4809_C000	0x480B_7FFF	112KiB	Reserved
McSPI3	TP_MCSPI3_TARG	0x480B_8000	0x480B_8FFF	4KiB	Module target port
	TA_MCSPI3_TARG	0x480B_9000	0x480B_9FFF	4KiB	L4 target agent
McSPI4	TP_MCSPI4_TARG	0x480B_A000	0x480B_AFFF	4KiB	Module target port
	TA_MCSPI4_TARG	0x480B_B000	0x480B_BFFF	4KiB	L4 target agent
Reserved	Reserved	0x480B_C000	0x480D_0FFF	84KiB	Reserved

Table 2-5. L4_PER1 Memory Map (continued)

Module name	Region name	Start_address (hex)	End_address (hex)	Size	Description
MMC	TP_MMC_TARG	0x480D_1000	0x480D_1FFF	4KiB	Module target port
	TA_MMC_TARG	0x480D_2000	0x480D_2FFF	4KiB	L4 target agent
Reserved	Reserved	0x480D_3000	0x483F_FFFF	3330KiB	Reserved

2.4.2 L4_PER2 Memory Map

Table 2-6 describes the mapping of the register for the L4_PER2 interconnect.

Table 2-6. L4_PER2 Memory Map

Module name	Region name	Start_address (hex)	End_address (hex)	Size	Description
L4_PER2	L4_PER2_AP	0x4840_0000	0x4840_07FF	2KiB	Address protection
	L4_PER2_LA	0x4840_0800	0x4840_0FFF	2KiB	Link Agent
	L4_PER2_IA_IP0	0x4840_1000	0x4840_13FF	1KiB	Initiator Port 0
	L4_PER2_IA_IP1	0x4840_1400	0x4840_17FF	1KiB	Initiator Port 1
	L4_PER2_IA_IP2	0x4840_1800	0x4840_1BFF	1KiB	Initiator Port 2
Reserved	Reserved	0x4840_1C00	0x4843_5FFF	209KiB	Reserved
McASP2	TP_MCASP2_DAT_TARG	0x4843_6000	0x4843_6FFF	4KiB	Module target port
	TP_MCASP2_DAT_TARG	0x4843_7000	0x4843_7FFF	4KiB	L4 interconnect target agent
Reserved	Reserved	0x4843_8000	0x4843_9FFF	8KiB	Reserved
McASP3	TP_MCASP3_DAT_TARG	0x4843_A000	0x4843_AFFF	4KiB	Module target port
	TP_MCASP3_DAT_TARG	0x4843_B000	0x4843_BFFF	4KiB	L4 interconnect target agent
ATL	TP_ATL_TARG	0x4843_C000	0x4843_CFFF	4KiB	Module target port
	TP_ATL_TARG	0x4843_D000	0x4843_DFFF	4KiB	L4 interconnect target agent
PWMSS	TP_PWMSS_TARG	0x4843_E000	0x4843_EFFF	4KiB	Module target port
	TA_PWMSS_TARG	0x4843_F000	0x4843_FFFF	4KiB	L4 interconnect target agent
Reserved	Reserved	0x4844_0000	0x4845_FFFF	131KiB	Reserved
	TP_MCASP_CFG_TARG	0x4846_0000	0x4846_1FFF	8KiB	Module target port
McASP1	TA_MCASP_CFG_TARG	0x4846_2000	0x4846_2FFF	4KiB	L4 interconnect target agent
Reserved	Reserved	0x4846_3000	0x4846_BFFF	136KiB	Reserved
McASP2	TP_MCASP2_CFG_TARG	0x4846_C000	0x4846_DFFF	8KiB	Module target port
	TP_MCASP2_CFG_TARG	0x4846_E000	0x4846_EFFF	4KiB	L4 interconnect target agent
Reserved	Reserved	0x4846_F000	0x4846_FFFF	4KiB	Reserved
McASP3	TP_MCASP3_CFG_TARG	0x4847_0000	0x4847_1FFF	8KiB	Module target port
	TP_MCASP3_CFG_TARG	0x4847_2000	0x4847_2FFF	4KiB	L4 interconnect target agent
Reserved	Reserved	0x4847_3000	0x4847_FFFF	52KiB	Reserved
	TP_MCAN_TARG	0x4848_0000	0x4848_1FFF	8KiB	Module target port
MCAN	TP_MCAN_TARG	0x4848_2000	0x4848_2FFF	4KiB	L4 interconnect target agent
Reserved	Reserved	0x4848_3000	0x4848_3FFF	4KiB	Reserved
GMAC_SW	TP_GMAC_SW_TARG	0x4848_4000	0x4848_7FFF	16KiB	Module target port
	TA_GMAC_SW_TARG	0x4848_8000	0x4848_8FFF	4KiB	L4 interconnect target agent
Reserved	Reserved	0x4848_9000	0x487F_FFFF	3548KiB	Reserved

2.4.3 L4_PER3 Memory Map

Table 2-7 describes the mapping of the register for the L4_PER3 interconnect.

Table 2-7. L4_PER3 Memory Map

Module name	Region name tag	Start_address (hex)	End_address (hex)	Size	Description
L4_PER3	L4_PER3_AP	0x4880_0000	0x4880_07FF	2KiB	Address protection
	L4_PER3_LA	0x4880_0800	0x4880_0FFF	2KiB	Link agent
	L4_PER3_IA_IP0	0x4880_1000	0x4880_13FF	1KiB	Initiator Port 0
	L4_PER3_IA_IP1	0x4880_1400	0x4880_17FF	1KiB	Initiator Port 1
	L4_PER3_IA_IP2	0x4880_1800	0x4880_1BFF	1KiB	Initiator Port 2
Reserved	Reserved	0x4880_1C00	0x4880_3FFF	9KiB	Reserved
OCMC_RAM	TP_OCMC_RAM _CFG_TARG	0x4880_4000	0x4880_4FFF	4KiB	Module target port
	TA_OCMC_RAM _CFG_TARG	0x4880_5000	0x4880_5FFF	4KiB	L4 interconnect target agent
Reserved	Reserved	0x4880_6000	0x4881_BFFF	90KiB	Reserved
MMU	TP_MMU_TARG	0x4881_C000	0x4881_CFFF	4KiB	Module target port
	TA_MMU_TARG	0x4881_D000	0x4881_DFFF	4KiB	L4 interconnect target agent
Reserved	Reserved	0x4881_E000	0x4881_FFFF	8KiB	Reserved
TIMER5	TP_TIMER5_TARG	0x4882_0000	0x4882_0FFF	4KiB	Module target port
	TA_TIMER5_TARG	0x4882_1000	0x4882_1FFF	4KiB	L4 interconnect target agent
TIMER6	TP_TIMER6_TARG	0x4882_2000	0x4882_2FFF	4KiB	Module target port
	TA_TIMER6_TARG	0x4882_3000	0x4882_3FFF	4KiB	L4 interconnect target agent
TIMER7	TP_TIMER7_TARG	0x4882_4000	0x4882_4FFF	4KiB	Module target port
	TA_TIMER7_TARG	0x4882_5000	0x4882_5FFF	4KiB	L4 interconnect target agent
TIMER8	TP_TIMER8_TARG	0x4882_6000	0x4882_6FFF	4KiB	Module target port
	TA_TIMER8_TARG	0x4882_7000	0x4882_7FFF	4KiB	L4 interconnect target agent
Reserved	Reserved	0x4882_8000	0x4882_FFFF	32KiB	Reserved
Reserved	Reserved	0x4883_0000	0x4883_9FFF	40KiB	Reserved
MAILBOX2	TP_MAILBOX2_TARG	0x4883_A000	0x4883_AFFF	4KiB	Module target port
	TA_MAILBOX2_TARG	0x4883_B000	0x4883_BFFF	4KiB	L4 interconnect target agent
Reserved	Reserved	0x4883_C000	0x4884_7FFF	48KiB	Reserved
Reserved	Reserved	0x4884_8000	0x4884_DFFF	24KiB	Reserved
Reserved	Reserved	0x4885_0000	0x4885_1FFF	8KiB	Reserved
Reserved	Reserved	0x4885_6000	0x4885_DFFF	32KiB	Reserved
Reserved	Reserved	0x4885_E000	0x4896_FFFF	1122KiB	Reserved
VIP	TP_VIP_TARG	0x4897_0000	0x4897_FFFF	64KiB	Module target port
	TA_VIP_TARG	0x4898_0000	0x4898_0FFF	4KiB	L4 interconnect target agent
Reserved	Reserved	0x4898_1000	0x48FF_FFFF	6811KiB	Reserved

2.5 IPU Memory Map

The device implements one IPU subsystem. For more information about IPU, see Chapter 5, *Dual Cortex-M4 IPU Subsystem*.

Table 2-8 describes the IPU memory mapping.

NOTE: Some of the system (L3) resources, such as EVE and EDMA, are not directly accessible by IPU, as they are overlapping with IPU's own resources in its memory space. In such cases, software must properly configure IPU AMMU / L2 MMU so that IPU can access these system resources.

Table 2-8. IPU Memory Map

Region Name	Start_Address (hex)	End_Address (hex)	Size	Description
IPU_BOOT_SPACE(1)	0x0000_0000	0x0000_3FFF	16KiB	IPU boot space
L3_MAIN map	0x0000_0000	0x1FFF_FFFF	512MiB	See Table 2-1
IPU_BITBAND_REGION1	0x2000_0000	0x200F_FFFF	1MiB	IPU bit-band region 1
Reserved	0x2010_0000	0x21FF_FFFF	31MiB	Reserved
IPU_BITBAND_ALIAS1	0x2200_0000	0x23FF_FFFF	32MiB	IPU bit-band alias 1
L3_MAIN map	0x2400_0000	0x3FFF_FFFF	448MiB	See Table 2-1
IPU_BITBAND_REGION2	0x4000_0000	0x400F_FFFF	1MiB	IPU bit-band region 2
Reserved	0x4010_0000	0x402F_FFFF	2MiB	Reserved
L3_MAIN map	0x4030_0000	0x41FF_FFFF	30MiB	See Table 2-1
IPU_BITBAND_ALIAS2	0x4200_0000	0x43FF_FFFF	32MiB	IPU bit-band alias 2
L3_MAIN map	0x4400_0000	0x54FF_FFFF	285MiB	See Table 2-1
IPU_ROM(2)	0x5500_0000	0x5500_7FFF	32KiB	IPU_ROM
Reserved	0x5500_8000	0x5501_FFFF	96KiB	Reserved
IPU_RAM(2)	0x5502_0000	0x5502_FFFF	64KiB	IPU_RAM
Reserved	0x5503_0000	0x5507_FFFF	320KiB	Reserved
IPU_UNICACHE_CFG	0x5508_0000	0x5508_00FF	256B	IPU_UNICACHE config registers
Reserved	0x5508_0100	0x5508_03FF	768B	Reserved
IPU_UNICACHE_SCTM	0x5508_0400	0x5508_07FF	1KiB	IPU_UNICACHE_SCTM config registers
IPU_UNICACHE_MMU(2)	0x5508_0800	0x5508_0FFF	2KiB	IPU_UNICACHE_MMU config registers
IPU_WUGEN	0x5508_1000	0x5508_1FFF	4KiB	IPU_WUGEN config registers
IPU_MMU(2)	0x5508_2000	0x5508_2FFF	4KiB	IPU_MMU config registers
Reserved	0x5508_3000	0x55FF_FFFF	16MiB	Reserved
L3_MAIN map	0x5600_0000	0xDFFF_FFFF	2,3GiB	See Table 2-1
Reserved	0xE000_0000	0xE000_0FFF	4KiB	Reserved
IPU_C0_DWT	0xE000_1000	0xE000_1FFF	4KiB	IPU_C0_DWT config registers
IPU_C0_FPB	0xE000_2000	0xE000_2FFF	4KiB	IPU_C0_FPB config registers
Reserved	0xE000_3000	0xE000_DFFF	44KiB	Reserved
IPU_C0_INTC	0xE000_E000	0xE000_EFFF	4KiB	IPU_C0_INTC config registers
Reserved	0xE000_F000	0xE004_1FFF	204KiB	Reserved
IPU_C0_ICECRUSHER	0xE004_2000	0xE004_2FFF	4KiB	IPU_C0_ICECRUSHER config registers
Reserved	0xE004_3000	0xE00F_DFFF	748KB	Reserved
IPU_C0_RW_TABLE	0xE00F_E000	0xE00F_EFFF	4KiB	IPU_C0 RW table
IPU_C0_ROM_TABLE	0xE00F_F000	0xE00F_FFFF	4KiB	IPU_C0 ROM table
IPU_C1_DWT	0xE000_1000	0xE000_1FFF	4KiB	IPU_C1_DWT config registers

(1) See Section 5.3.7, *IPU Boot Configuration*.
(2) Can also be accessed from L3_MAIN (by other initiators).

Table 2-8. IPU Memory Map (continued)

Region Name	Start_Address (hex)	End_Address (hex)	Size	Description
IPU_C1_FPB	0xE000_2000	0xE000_2FFF	4KiB	IPU_C1_FPB config registers
Reserved	0xE000_3000	0xE000_3FFF	4KiB	Reserved
IPU_C1_INTC	0xE000_E000	0xE000_EFFF	4KiB	IPU_C1_INTC config registers
Reserved	0xE000_F000	0xE004_1FFF	204KiB	Reserved
IPU_C1_ICECRUSHER	0xE004_2000	0xE004_2FFF	4KiB	IPU_C1_ICECRUSHER config registers
Reserved	0xE004_3000	0xE00F_DFFF	748KiB	Reserved
IPU_C1_RW_TABLE	0xE00F_E000	0xE00F_EFFF	4KiB	IPU_C1 RW table
IPU_C1_ROM_TABLE	0xE00F_F000	0xE00F_FFFF	4KiB	IPU_C1 ROM table
L3_MAIN map	0xE010_0000	0xFFFF_FFFF	511MiB	See Table 2-1

Legend:		= IPU private memory space
		= Reserved memory space

2.6 DSP Memory Map

The device implements two DSP subsystems (DSP1, DSP2). For more information about DSP, see Chapter 4, *DSP Subsystems*.

Table 2-9 describes the DSP memory mapping.

Table 2-9. DSP Memory Map

Region Name	Start_Address (hex)	End_Address (hex)	Size	Description
Reserved	0x0000_0000	0x007F_FFFF	8MiB	Reserved
DSP_L2[(1)]	0x0080_0000	0x0084_7FFF	288KiB	DSP L2 SRAM and cache. The L2 SRAM starts at 0x0080_0000 address.
Reserved	0x0084_8000	0x00DF_FFFF	5856KiB	Reserved
DSP_L1P[(1)]	0x00E0_0000	0x00E0_7FFF	32KiB	DSP L1P SRAM
Reserved	0x00E0_8000	0x00EF_FFFF	992KiB	Reserved
DSP_L1D[(1)]	0x00F0_0000	0x00F0_7FFF	32KiB	DSP L1D SRAM
Reserved	0x00F0_8000	0x00FF_FFFF	992KiB	Reserved
DSP_ICFG[(1)]	0x0100_0000	0x01BF_FFFF	12MiB	DSP internal CFG
Reserved	0x01C0_0000	0x01CF_FFFF	1MiB	Reserved
DSP_SYSTEM[(1)]	0x01D0_0000	0x01D0_0FFF	4KiB	DSP system registers block
DSP_MMU0CFG[(1)]	0x01D0_1000	0x01D0_1FFF	4KiB	DSP MMU0 configuration
DSP_MMU1CFG[(1)]	0x01D0_2000	0x01D0_2FFF	4KiB	DSP MMU1 configuration
DSP_FW0CFG[(1)]	0x01D0_3000	0x01D0_3FFF	4KiB	DSP firewall 0 config
DSP_FW1CFG[(1)]	0x01D0_4000	0x01D0_4FFF	4KiB	DSP firewall 1 config
DSP_EDMA_TC0[(1)]	0x01D0_5000	0x01D0_5FFF	4KiB	DSP EDMA transfer controller 0
DSP_EDMA_TC1[(1)]	0x01D0_6000	0x01D0_6FFF	4KiB	DSP EDMA transfer controller 1
DSP_NoC[(1)]	0x01D0_7000	0x01D0_7FFF	4KiB	DSP interconnect registers
Reserved	0x01D0_8000	0x01D0_FFFF	32KiB	Reserved
DSP_EDMA_CC[(1)]	0x01D1_0000	0x01D1_7FFF	32KiB	DSP EDMA channel controller
Reserved	0x01D1_8000	0x01FF_FFFF	2976KiB	Reserved
EVE	0x0200_0000	0x020F_FFFF	1MiB	EVE configuration space
Reserved	0x0210_0000	0x032F_FFFF	18MiB	Reserved
EDMA_TPCC	0x0330_0000	0x033F_FFFF	1MiB	EDMA_TPCC configuration space
EDMA_TC0	0x0340_0000	0x034F_FFFF	1MiB	EDMA_TC0 configuration space
EDMA_TC1	0x0350_0000	0x035F_FFFF	1MiB	EDMA_TC1 configuration space
Reserved	0x0360_0000	0x07FF_FFFF	74MiB	Reserved
DSP_XMC_CTRL[(1)]	0x0800_0000	0x0800_FFFF	64KiB	DSP XMC control registers
DSP_EDI[(1)]	0x0801_0000	0x0801_FFFF	64KiB	DSP internal EDI translation region
Reserved	0x0802_0000	0x13FF_FFFF	192MiB	Reserved
L3_MAIN map	0x1400_0000	0xFFFF_FFFF	3,8GiB	See Table 2-1

Legend:		
		= DSP private memory space
		= Reserved memory space

(1) DSP subsystem internal resources. DSP accesses in ranges [0x0080_0000 – 0x01D1_7FFF] and [0x0800_0000 – 0x0801_FFFF] are performed locally within the DSP subsystem.

2.7 EVE Memory Map

The device implements one EVE subsystem. For more information about EVE, see Section 6.1, *Embedded Vision Engine (EVE) Subsystem*.

Table 2-10 describes the EVE memory mapping.

Table 2-10. EVE Memory Map

Region name	Start_Address (hex)	End_Address (hex)	Size	Description
L3_MAIN map	0x0000_0000	0x2FFF_FFFF	768MiB	See Table 2-1
Reserved	0x3000_0000	0x4001_FFFF	384KiB	Reserved
EVE_DMEM	0x4002_0000	0x4002_7FFF	32KiB	EVE ARP32 data memory (DMEM)
Reserved	0x4002_8000	0x4003_FFFF	96KiB	Reserved
EVE_WBUF	0x4004_0000	0x4004_7FFF	32KiB	EVE VCOP working buffer (WBUF)
Reserved	0x4004_8000	0x4004_FFFF	32KiB	Reserved
EVE_IBUF_LA	0x4005_0000	0x4005_3FFF	16KiB	EVE image buffer low copy A (IBUFLA)
EVE_IBUF_HA	0x4005_4000	0x4005_7FFF	16KiB	EVE image buffer high copy A (IBUFHA)
Reserved	0x4005_8000	0x4006_FFFF	96KiB	Reserved
EVE_IBUF_LB	0x4007_0000	0x4007_3FFF	16KiB	EVE image buffer low copy B (IBUFLB)
EVE_IBUF_HB	0x4007_4000	0x4007_7FFF	16KiB	EVE image buffer high copy B (IBUFHB)
Reserved	0x4007_8000	0x4007_FFFF	32KiB	Reserved
EVE_SYSTEM	0x4008_0000	0x4008_0FFF	4KiB	EVE int, reset, clk, pwr, buffswc (MEMSWITCH_CTL)
EVE_MMU0_CFG	0x4008_1000	0x4008_1FFF	4KiB	EVE MMU0 configuration
EVE_MMU1_CFG	0x4008_2000	0x4008_2FFF	4KiB	EVE MMU1 configuration
EVE_T16C	0x4008_3000	0x4008_3FFF	4KiB	EVE T16 control
EVE_VCOPC	0x4008_4000	0x4008_4FFF	4KiB	EVE VCOP control
EVE_SCTM	0x4008_5000	0x4008_5FFF	4KiB	EVE Counter and Timer module (SCTM)
EVE_EDMA_TC0	0x4008_6000	0x4008_6FFF	4KiB	EVE EDMA Transfer controller 0 (EDMA TC0)
EVE_EDMA_TC1	0x4008_7000	0x4008_7FFF	4KiB	EVE EDMA Transfer controller 1 (EDMA TC1)
EVE_SMSET_CFG	0x4008_8000	0x4008_8FFF	4KiB	EVE SMSET configuration interface
EVE_SMSET_MSG	0x4008_9000	0x4008_9FFF	4KiB	EVE SMSET messaging interface
EVE_NoC	0x4008_A000	0x4008_AFFF	4KiB	EVE interconnect registers
EVE_MBX0	0x4008_B000	0x4008_BFFF	4KiB	EVE Mailbox 0
EVE_MBX1	0x4008_C000	0x4008_CFFF	4KiB	EVE Mailbox 1
EVE_MBX2	0x4008_D000	0x4008_DFFF	4KiB	EVE Mailbox 2
EVE_MBX3	0x4008_E000	0x4008_EFFF	4KiB	EVE Mailbox 3
EVE_MBX4	0x4008_F000	0x4008_FFFF	4KiB	EVE Mailbox 4
EVE_PCACHE_RAW	0x4009_0000	0x4009_7FFF	32KiB	EVE Program Cache Raw
EVE_PCACHE_Tags	0x4009_8000	0x4009_FFFF	32KiB	EVE Program Cache Tag
EVE_EDMA_CC	0x400A_0000	0x400A_7FFF	32KiB	EVE EDMA Channel Controller (EDMA CC)
Reserved	0x400A_8000	0x402F_FFFF	2,4MiB	Reserved
L3_MAIN map	0x4030_0000	0xFFFF_FFFF	3GiB	See Table 2-1

Legend:		= EVE private memory space
		= Reserved memory space

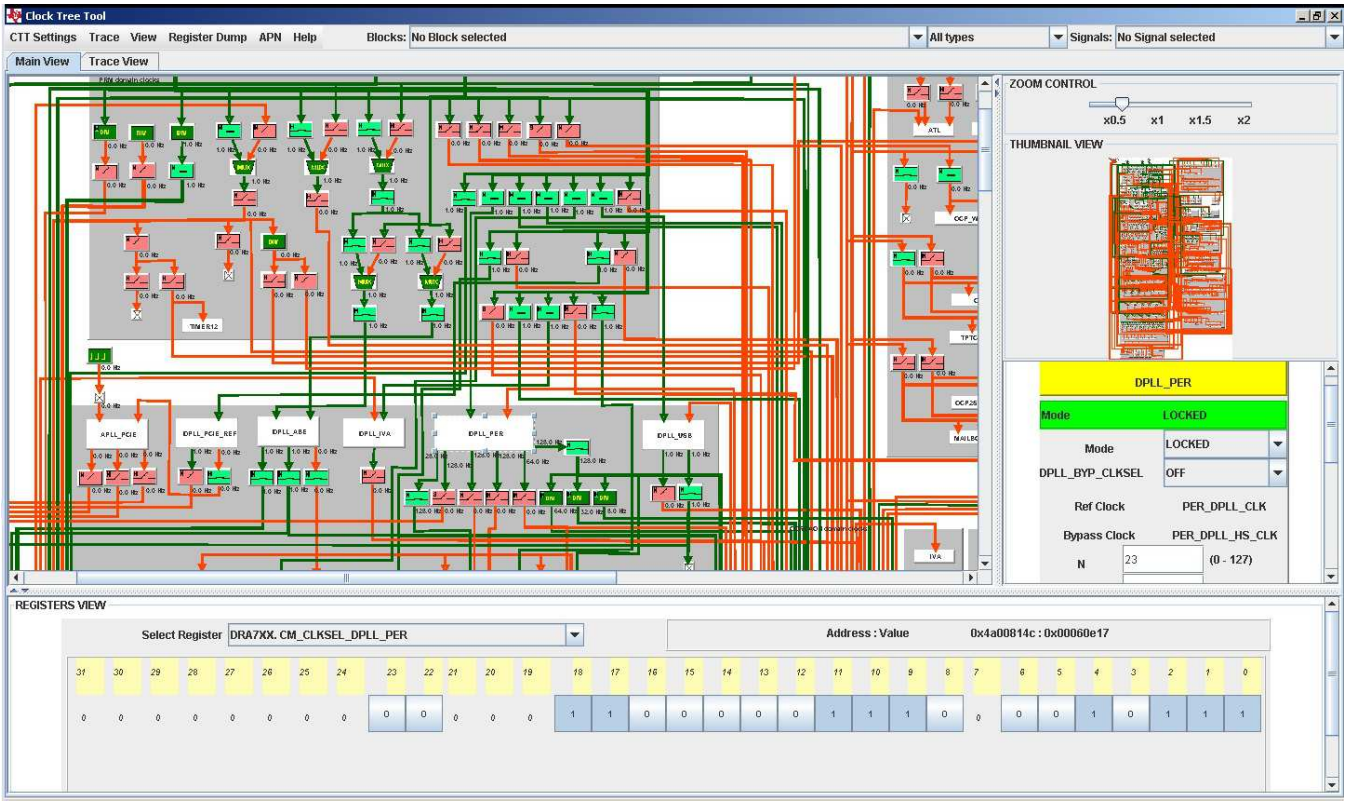
Power, Reset, and Clock Management

This chapter describes the power, reset, and clock management in the device.

Topic **Page**

NOTE: For a detailed visual representation of the distribution and management of the device clocks at the PRCM level, see the clock device interactive software for the device CLOCKTREETOOL-AUTOMOTIVE

Figure 3-1. Clock Tree Tool (CTT)



prcm-095

The Clock Tree Tool (CTT) is a Java-based, stand-alone application. The CTT is interactive clock tree configuration software for the device. The CTT lets the user:

- Visualize the device clock tree
- Interact with clock tree elements and view the effect on PRCM registers
- Interact with the PRCM registers and view the effect on the device clock tree
- View a trace of all the device registers affected by the user interaction with clock tree
- Extract registers and register dumps for Code Composer Studio and Lauterbach

The advantage of the CTT is that the user can visualize the clock tree state of the device on power-on reset and then customize the configuration of the clock tree for the specific use case and identify the device register settings associated to that configuration. Furthermore, the user can dump and read in register settings for the specific scenarios.

Being an interactive visual tool, the CTT gives the user a global view of the device clock tree architecture and allows determining the exact register settings to obtain the specific configuration.

3.1 Device Power Management Introduction

Power management is one of the most important design aspects of any system.

The device power-management architecture ensures maximum performance while offering versatile power-management techniques for maximum design flexibility, depending on application requirements.

This introduction contains the following information:

- Power-management architecture building blocks for the device
- State-of-the-art power-management techniques supported by the power-management architecture of the device

3.1.1 Device Power-Management Architecture Building Blocks

To provide a versatile architecture that supports multiple power-management techniques, the power-management framework is built with three levels of resource management: clock, power, and voltage.

These management levels are enforced by defining the managed entities or building blocks of the power-management architecture, called the clock, power, and voltage domains.

A domain is a group of modules or subsections of the device that share a common entity (for example, common clock source, common voltage source, or common power switch). The group forming the domain is managed by a policy manager. For example, a clock for a clock domain is managed by a dedicated clock manager within the power, reset, and clock management (PRCM) module. The clock manager considers the joint clocking constraints of all the modules belonging to that clock domain (and, hence, receiving that clock).

NOTE: In the following sections, the term <*module*> is used to represent the device IPs (that is, modules or subsystems), other than the PRCM module, that receive clock, reset, or power signals from the PRCM module.

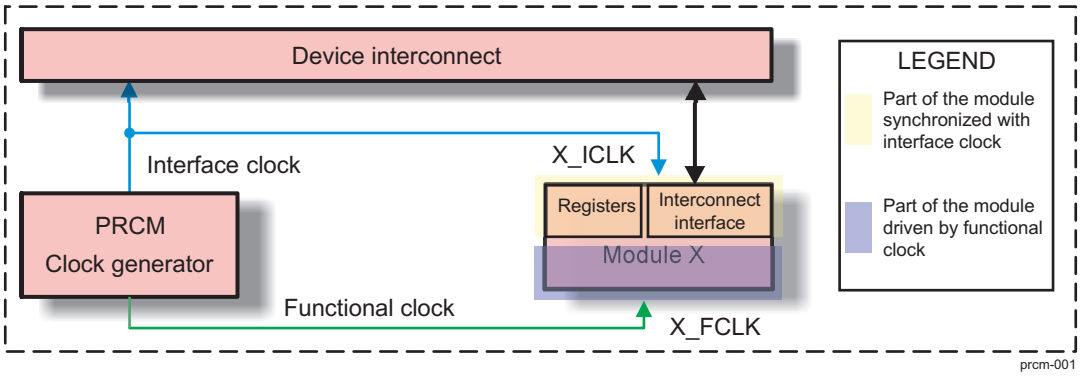
3.1.1.1 Clock Management

The PRCM module manages the gating (that is, switching off) and enabling of the clocks to the device modules. The clocks are managed based on the requirement constraints of the associated modules. The following sections identify the module clock characteristics, management policy, clock domains, and clock domain management. Clock management integration details are described in Section 3.6, *Clock Management Functional Description*.

3.1.1.1.1 Module Interface and Functional Clocks

Each module within the device has specific clock input characteristics requirements. Based on the characteristics of the clocks delivered to the modules, the clocks are divided into two categories: interface clocks and functional clocks (see Figure 3-2).

Figure 3-2. Functional and Interface Clocks



The interface clocks have the following characteristics:

- They ensure proper communication between any module/subsystem and the interconnect.
- In most cases, they supply the system interconnect interface and registers of the module.
- A typical module has one interface clock, but modules with multiple interface clocks may also exist (that is, when connected to multiple interconnect buses).
- Interface clock management is done at the device level.

- From the standpoint of the PRCM module, an interface clock is identified with an _ICLK suffix.

Functional clocks have the following characteristics:

- They supply the functional part of a module or subsystem.
- A module can have one or more functional clocks. Some functional clocks are mandatory, while others are optional. A module needs its mandatory clock(s) to be operational. The optional clocks are used for specific features and can be shut down without stopping the module activity (for example, the clock for the camera).
- From the standpoint of the PRCM module, a functional clock is distributed directly to the related modules through a dedicated clock tree. It is identified with an _FCLK suffix.

Some clocks are qualified as permanent clocks. They are functional clocks, that can stay active while the corresponding entity manages them.

NOTE: At the module level, the interface clocks are always fed by the interface clock outputs of the PRCM module. The functional clocks are fed by a PRCM module functional clock output or a PRCM module interface clock output. In the latter case, the functional and interface module clocks inherit the clock-management features (autoidle features) of the PRCM module interface clock.

3.1.1.1.2 Module-Level Clock Management

Each module in the device may also have specific clock requirements. Certain module clocks must be active when operating in specific modes, or they may be gated. Globally, the activation and gating of the module clocks are managed by the PRCM module. Hence, the PRCM module must be aware of when to activate and when to gate the module clocks.

The PRCM module differentiates the clock-management behavior for device modules based on whether the module can initiate transactions on the device interconnect (called master module) or it cannot initiate transactions and only responds to the transactions initiated by the master (called slave module). Thus, two hardware-based power-management protocols are used:

- Master standby protocol: Clock-management protocol between the PRCM and master modules
- Slave idle protocol: Clock-management protocol between the PRCM and slave modules

Master standby protocol

This protocol is used to indicate that a master module must initiate a transaction on the device interconnect and requests specific (functional and interface) clocks for that purpose. The PRCM module ensures that the required clocks are active when the master module requests the PRCM module to enable them. This is called a module wake-up transition and the module is said to be functional after this transition completes.

Similarly, when the master module no longer requires the clocks, it informs the PRCM module, which can then gate the clocks to the module. The master module is then said to be in standby mode.

Although the protocol is completely hardware-controlled, software must configure the clock-management behavior for the module. This is done by setting the <*Module*>_SYSCONFIG. MIDLEMODE or <*Module*>_SYSCONFIG. STANDBYMODE bit fields, as described in Table 3-1. The behavior, identified in the STANDBYMODE Bit Value column, must be configured.

Table 3-1. Master Module Standby Mode Settings

STANDBYMODE Bit Value	Selected Mode	Description
0x0	Force-standby	The module unconditionally asserts the standby request to the PRCM module, regardless of its internal operations. The PRCM module may gate the functional and interface clocks to the module. This mode must be used carefully because it does not prevent loss of data at the time the clocks are gated.

Table 3-1. Master Module Standby Mode Settings (continued)

STANDBYMODE Bit Value	Selected Mode	Description
0x1	No-standby	The module never asserts the standby request to the PRCM module. This mode is safe from a module point of view because it ensures that the clocks remain active; however, it is not efficient from a power-saving perspective because it never allows the PRCM module output clocks to be gated.
0x2	Smart-standby	The module asserts the standby request based on its internal activity status. The standby signal is asserted only when all ongoing transactions are complete and the module is idled. The PRCM module can then gate the clocks to the module.
0x3	Smart-standby wake-up-capable mode	The module asserts the standby request based on its internal activity status. The standby signal is asserted only when all ongoing transactions are complete and the module is idle. The PRCM module can then gate the clocks to the module. The module may generate (master-related) wake-up events when in standby state. The mode is relevant only if the appropriate module mwakeup output is implemented.

NOTE:

- Smart-standby mode is the preferred mode of operation, while force-standby and no-standby modes are intended for debugging purposes.
- A master module may support all or some of the standby modes listed in Table 3-1. See the power-management section in the module chapter to identify the supported standby mode.

The standby status of a master module is indicated by the CM_<*Clock domain*>_<*Module*>_CLKCTRL[x]. STBYST bit in the PRCM module. Table 3-2 describes the master module standby status.

Table 3-2. Master Module Standby Status

STBYST Bit Value	Description
0x0	The module is functional.
0x1	The module is in standby mode.

Table 3-3 lists the enabling conditions for the master module clocks managed by the standby protocol.

Table 3-3. Master Module Clock Enabling Conditions

Relation		Condition
AND		Clock domain is ready.
	OR	Master module standby request is deasserted.
		Master module wake-up request is asserted.

Slave idle protocol

This hardware protocol allows the PRCM module to control the state of a slave module. The PRCM module informs the slave module, through assertion of an IDLE request, when its clocks (interface and functional) can be gated. The slave can then acknowledge the request from the PRCM module, and the PRCM module is then allowed to gate the clocks to the module. A slave module is said to be in IDLE state when its clocks are gated by the PRCM module.

Similarly, an idled slave module may need to be woken up because of a service request from a master module or because the slave module receives a wake-up event (for example, an interrupt or a direct memory access [DMA] request). In this situation the PRCM module enables the clocks for the module, and then signals the module to wake up by deasserting the IDLE request.

Although the protocol is completely hardware-controlled, software must configure the clock-management behavior for the slave module. This is done by setting the <*Module*>_SYSCONFIG. SIDLEMODE or <*Module*>_SYSCONFIG. IDLEMODE bit field, as described in Table 3-4. The behavior, identified in the IDLEMODE Bit Value column, must be configured by software.

Table 3-4. Module Idle Mode Settings

IDLEMODE Bit Value	Selected Mode	Description
0x0	Force-idle	The module unconditionally acknowledges the IDLE request from the PRCM module, regardless of its internal operations. This mode must be used carefully because it does not prevent loss of data at the time the clock is switched off.
0x1	No-idle	The module never acknowledges any IDLE request from the PRCM module. This mode is safe from a module point of view because it ensures the clocks remain active; however, it is not efficient from a power-saving perspective because it does not allow the PRCM module output clock to be shut off, and thus the power domain to be set to a lower power state.
0x2	Smart-idle	The module acknowledges the IDLE request, basing its decision on its internal activity. Namely, the acknowledge signal is asserted only when all pending transactions, interrupts, or DMA requests are processed. This is the best approach to efficient system power management.
0x3	Smart-idle wake-up-capable mode	The module acknowledges the IDLE request, basing its decision on its internal activity. Namely, the acknowledge signal is asserted only when all pending transactions, interrupts, or DMA requests are processed. This is the best approach to efficient system power management. The module may generate (IRQ- or DMA-request-related) wake-up events when in IDLE state. The mode is relevant only if the appropriate module swakeup output(s) is implemented.

NOTE:

- Smart-idle mode is the preferred mode of operation, while force-idle and no-idle modes are intended for debugging purposes.
- A slave module may support all or some of the idle modes listed in Table 3-4. See the power-management section in the module chapter to identify the supported idle modes.

The idle status of a slave module is indicated by the CM_<*Clock domain*>_<*Module*>_CLKCTRL[x] IDLEST bit field in the PRCM module. Table 3-5 lists the possible idle statuses for a slave module.

Table 3-5. Slave Module Idle Status

IDLEST Bit Value	Idle Status	Description
0x0	Functional	The module is fully functional. The interface and functional clocks are active.
0x1	In transition	The module is performing a wake-up or a sleep transition.
0x2	Interface idle	The module interface clock is idled. The module may remain functional if using a separate functional clock.
0x3	Full idle	The module is fully idle. The interface and functional clocks are gated.

When configured in smart-idle mode, the slave module may acknowledge the IDLE request of the PRCM module based on the activity of its interface and/or functional clocks. To define which module clocks (that is, interface and/or functional) should be considered when responding to the PRCM module request, software must configure the <*Module*>_SYSCONFIG[x] CLOCKACTIVITY bit field.

The CLOCKACTIVITY setting is used internally by the module to determine the part of the module on which the conditions to acknowledge the PRCM module IDLE request are tested. As an example, if the functional clock must remain active when the module is in idle mode, the module must acknowledge a PRCM module IDLE request by considering only the interface clock gating conditions (that is, there is no pending activity on the interconnect).

NOTE: See the power-management section in the module chapter to identify whether this feature is configurable.

Using the CLOCKACTIVITY setting along with smart-idle mode ensures that the clock remains active for the module features that must remain available during the module idle mode. Table 3-6 describes the possible CLOCKACTIVITY settings for a module.

Table 3-6. Slave Module Clock Activity Settings

CLOCKACTIVITY Bit Value	Module Interface Clock	Module Functional Clock	Description
0x0	Gated	Gated	The interface and functional clocks are considered when generating the acknowledgment. This setting also means both clocks may be gated upon a PRCM module IDLE request.
0x1	Active	Gated	The interface clock is not shut down and is not considered when generating the acknowledgment to the PRCM module IDLE request. Only the functional clock is considered.
0x2	Gated	Active	The functional clock is not shut down and is not considered when generating the acknowledgment to the PRCM module IDLE request. Only the interface clock is considered.
0x3	Active	Active	The interface and functional clocks are not shut down. The module can acknowledge the IDLE request without checking the internal functions linked to its clocks.

NOTE:

- The software configuration of the CLOCKACTIVITY settings may not be available for a given module. For some modules, the CLOCKACTIVITY settings can be hardwired.
- A slave module may support all or some of the CLOCKACTIVITY settings listed in Table 3-6.

See the power-management section in the specific module chapter to identify the supported idle feature and settings.

CAUTION

The PRCM module does not have any hardware means to read the CLOCKACTIVITY settings of the module. Software must ensure consistent programming between the CLOCKACTIVITY settings of the module and the clock-gating control bits in the PRCM module. The PRCM module must be configured (where software control is available) not to gate the module clock, which should remain active according to the CLOCKACTIVITY settings of the module.

For idle protocol management on the PRCM module side, the behavior of the PRCM module is configured in the CM_<Clock domain>_<module>_CLKCTRL[1:0] MODULEMODE bit field. Based on the configured behavior, the PRCM module asserts the IDLE request to the module unconditionally (that is, immediately when software requests) or through hardware control when the module idle conditions are satisfied. Table 3-7 describes the configurable behavior of MODULEMODE.

Table 3-7. Slave Module Mode Settings in PRCM

MODULEMODE Bit Value	Selected Mode	Description
0x0	Disabled	The PRCM module unconditionally asserts the module IDLE request. This request applies to the gating of the functional and interface clocks to the module. If acknowledged by the module, the PRCM module can gate all clocks to the module (that is, the module is completely disabled). It can react only to an asynchronous wake-up event (that is, a wake-up event that does not require the module functional clock to be active).

Table 3-7. Slave Module Mode Settings in PRCM (continued)

MODULEMODE Bit Value	Selected Mode	Description
0x1	Auto	This mode applies to a module when the PRCM module manages only its interface clock and not its functional clock. The PRCM module automatically asserts/deasserts the module IDLE request based on the clock domain transitions. If acknowledged by the module, the PRCM module can gate the interface clock to the module.
0x2	Enabled	This mode applies to a module when the PRCM module manages its interface and functional clocks. The functional clock to the module remains active unconditionally, while the PRCM module automatically asserts/deasserts the module IDLE request based on the clock domain transitions. If acknowledged by the module, the PRCM module can gate only the interface clock to the module.
0x3	Reserved	Not available

NOTE: The PRCM module may support all or some of the MODULEMODE module settings listed in Table 3-7. See the CM_<Clock domain>_<module>_CLKCTRL[1:0] MODULEMODE bit field description for the module to identify the supported settings.

NOTE: Modules, which can be configured with DISABLED or AUTO values of ModuleMode, cannot go from IDLE to DISABLED state if ModuleMode is changed from AUTO to DISABLED while module is in IDLE state. Once the CM_<Clock domain>_CLKSTCTRL[1:0] CLKTRCTRL has been set to SW_WKUP, then SW has to poll for PM_<Power_domain>_PWRSTST[1:0] PowerStateSt = 0x03 and PM_<Power_domain>_PWRSTST[20] InTransition = 0x00 before update MODULEMODE bit field.

Table 3-8 and Table 3-9 list the enabling conditions for the slave module clocks managed by the idle protocol.

Table 3-8. Slave Module Interface Clock Enabling Conditions

Relation		Condition
AND		Clock domain is ready.
	OR	Slave module idle status is 0x0 (fully functional).
		Slave module idle status is 0x1 (in transition).
		Slave module wake-up request is asserted.

Table 3-9. Slave Module Functional Clock Enabling Conditions

Relation		Condition
AND		Clock domain is ready.
	OR	Slave module idle status is 0x0 (fully functional).
		Slave module idle status is 0x1 (in transition).
		Slave module idle status is 0x2 (interface clock is idled).
		Slave module wake-up request is asserted.

The module clock domain must be ready for the optional clocks to the module, and any associated clock-enable control is asserted.

NOTE: A given clock can be used by more that one module. Clock-enabling conditions are then ORed together (that is, the clock is provided as soon as one of the enabling conditions is true). As a consequence, the clock is disabled only when all related enabling conditions are false.

Module wake-up request

In IDLE state, a slave module may have to wake up to generate an interrupt or a DMA request. This can be the result of an external request (for example, to the input/output [I/O] port of a general-purpose input/output [GPIO] module) or an internally generated event (for example, WD_TIMER time up). The slave module, with wake-up capability, sends a wake-up request to the PRCM module. The PRCM module then activates the module clocks and acknowledges the module wake-up request.

In IDLE state, some slave modules may require functional clock(s) to generate a wake-up event. Such requests are called synchronous wake-up events on the PRCM module side, while the events generated when the functional or interface module clocks are gated are called asynchronous wake-up events.

NOTE: See the power-management section in the module chapter to identify whether its wake-up event is synchronous or asynchronous.

The standby and idle clock-management protocols allow the configuration of the module-level clock-management interaction between the PRCM module and individual modules of the device. However, the PRCM module may not necessarily gate the clock to the module immediately after the module switches to standby or idle mode at the end of this interaction. This is because the same clock can be shared by other modules that are active and need this shared clock to complete their activity. As a result, the PRCM module provides a second level of clock management called the clock-domain level, as explained in Section 3.1.1.1.3, *Clock Domain*.

3.1.1.1.3 Clock Domain

A clock domain is a group of modules fed by clock signals controlled by the same clock manager in the PRCM module (see Figure 3-3). By gating the clocks in a clock domain, the clocks to all the modules belonging to that clock domain can be cut to lower their active power consumption (that is, the device is on and the clocks to the modules are dynamically switched to ACTIVE or INACTIVE [gated] state). Thus, a clock domain allows control of the dynamic power consumption of the device.

The device is partitioned into multiple clock domains and each clock domain is controlled by an associated clock manager within the PRCM module. This allows the PRCM module to activate and gate individually each device clock domain.

Figure 3-3. Generic Clock Domain

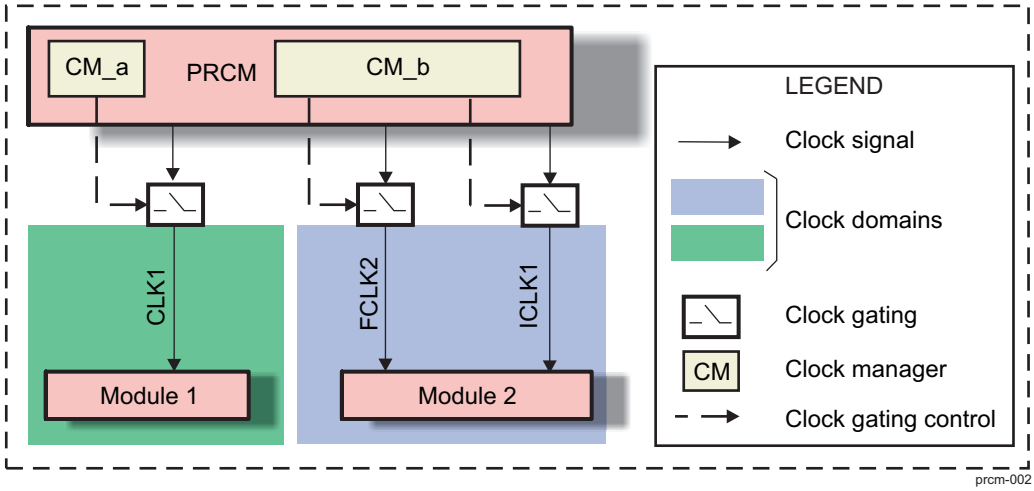


Figure 3-3 is an example of two clock managers: CM_a and CM_b. Each clock manager manages a clock domain. The clock domain of CM_b is composed of two clocks, a functional clock (FCLK2) and an interface clock (ICLK1), while that of CM_a consists of a clock (CLK1) that is used by the module as functional and interface clock. The clocks to Module 2 can be gated independently of the clock to Module 1, thus ensuring power savings when Module 2 is not in use.

The PRCM module lets software check the status of the clock domain functional clocks. The CM_<Clock domain>_CLKSTCTRL[x] CLKACTIVITY_<Clock name>_(G)FCLK bit in the PRCM module identifies the state of the functional clock(s) within the clock domain. Table 3-10 lists the two possible states of the functional clock.

Table 3-10. Clock Domain Functional Clock States

CLKACTIVITY Bit Value	Status	Description
0	Gated	The functional clock of the clock domain is inactive.
1	Active	The functional clock of the clock domain is running.

Similary, the PRCM module lets software check the status of the clock domain interface clocks. The CM_<Clock domain>_CLKSTCTRL[x] CLKACTIVITY_<Clock name>_(G)ICLK bit in the PRCM module identifies the state of the interface clock(s) within the clock domain. Table 3-10 lists the two possible states of the functional clock.

Table 3-11. Clock Domain Interface Clock States

CLKACTIVITY Bit Value	Status	Description
0	Gated	The interface clock of the clock domain is inactive.
1	Active	The interface clock of the clock domain is running.

3.1.1.1.4 Clock Domain-Level Clock Management

The domain clock manager can automatically (that is, based on hardware conditions) and jointly manage the interface clocks within the clock domain. The functional clocks within the clock domain are managed through software settings.

A clock domain can switch between three possible states: ACTIVE, IDLE_TRANSITION (IDLEREQ), and INACTIVE (IDLE). Figure 3-4 shows the sleep and wake-up transitions of the clock domain between ACTIVE and INACTIVE states.

Figure 3-4. Clock Domain State Transitions

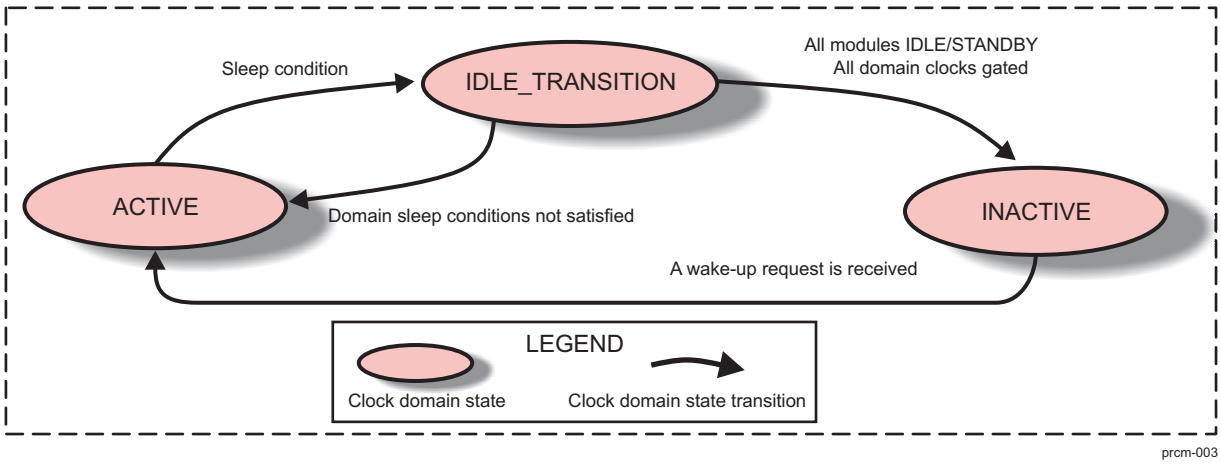


Table 3-12 defines the clock domain states.

Table 3-12. Clock Domain Clock States

State	Description
ACTIVE	• Every nondisabled slave module (that is, those whose MODULEMODE value is not set to disabled) is put out of IDLE state. • All mandatory functional clocks to the active slave modules (that is, not idled) of the clock domain are provided. • All interface clocks to the nondisabled slave modules in the clock domain are provided. • All mandatory functional and interface clocks to the active master modules (that is, not in STANDBY state) in the clock domain are provided. • Every enabled optional clock to the modules in the clock domain is provided.
IDLE_TRANSITION	This is a transitory state. • Every master module in the clock domain is in STANDBY state. • Every IDLE request to all the slave modules in the clock domain is asserted. • The functional clocks to the slave module in enabled state (that is, those whose MODULEMODE values are set to enabled) remain active. • Every enabled optional clock to the modules in the clock domain is provided.
INACTIVE	All clocks within the clock domain are gated. • Every slave module in the clock domain (that is, those whose MODULEMODE value is set to disabled or auto) is in IDLE state and set to disabled or auto mode. • Every master module in the clock domain is in STANDBY state. • Every optional functional clock in the clock domain is gated.

Each clock domain transition behavior is managed by an associated register bit field in the CM_<Clock domain>_CLKSTCTRL[x] CLKTRCTRL PRCM module.

Table 3-13 describes the clock transition mode settings of the clock domain.

Table 3-13. Clock Domain Clock Transition Mode Settings

CLKTRCTRL Bit Value	Selected Mode	Description
0x0	NO_SLEEP	A clock domain sleep transition is never initiated, regardless of the hardware conditions.
0x1	SW_SLEEP	A software-forced sleep transition. The transition is initiated when the associated hardware conditions are satisfied (see Table 3-15).
0x2	SW_WKUP	A software-forced clock domain wake-up transition is initiated, regardless of the hardware conditions identified in Table 3-14.
0x3	HW_AUTO	Hardware-controlled automatic sleep and wake-up transition is initiated by the PRCM module when the associated hardware conditions are satisfied (see Table 3-14 and Table 3-15).

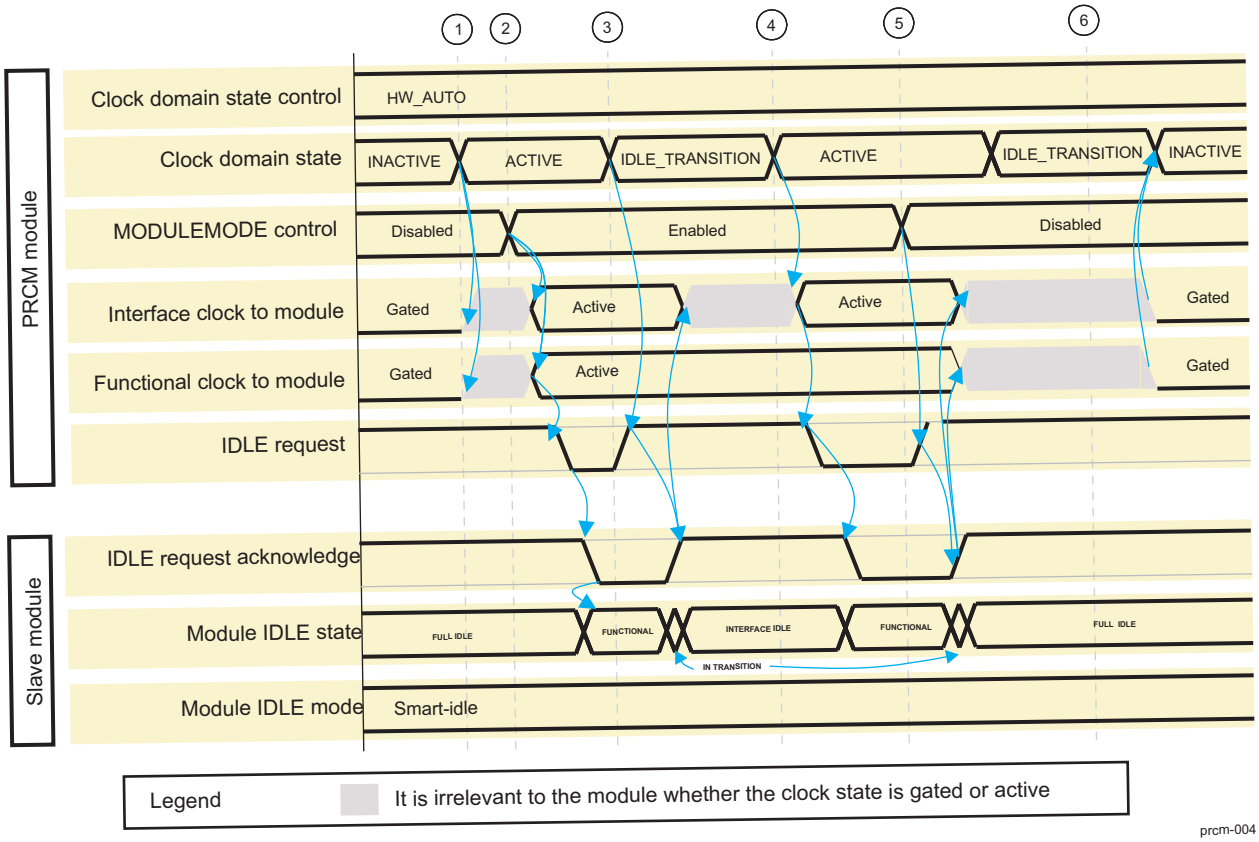
NOTE: Depending on its characteristics, a clock domain may or may not support all the clock transition mode settings described in Table 3-13. See the clock domain clock management section of the specific clock domain to identify the supported clock transition mode settings.

3.1.1.1.5 Clock Domain HW_AUTO Mode Sequences

The sequence diagrams in Figure 3-5 through Figure 3-7 identify the PRCM module hardware-controlled enabling and gating of the functional and interface clocks to the module. They show the changes in the state of the module based on the changes to the clock domain state and module mode settings.

Figure 3-5 shows the behavior of a slave module receiving the interface and functional clocks and having two configurable module modes: disabled and enabled.

Figure 3-5. Clock Domain/Slave Module Clock-Management Interaction Sequence 1

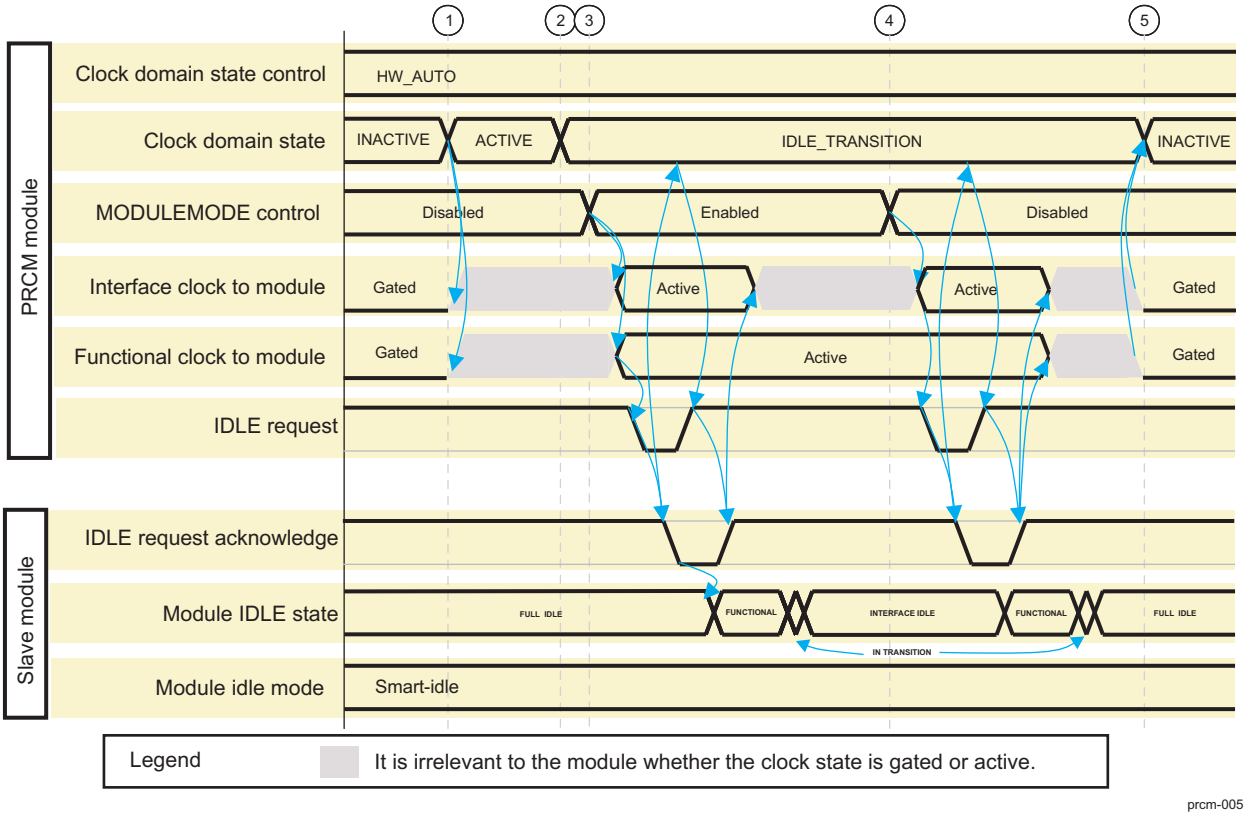


Initially, the clock domain, which includes the slave module, is inactive. The module is in FULL IDLE state and its functional and interface clocks are gated.

1. The clock domain wakes up and changes its state to ACTIVE. Because the MODULEMODE control is still disabled, this event has no effect on the state of the module. The functional and interface clocks can still be restarted automatically, based on the requirements of other modules sharing these clocks.
2. Software changes the module mode to ENABLED. The clocks to the module are automatically restarted. The PRCM module then deasserts a hardware IDLE request signal to the module. The module sends an IDLE request acknowledge to the PRCM module. The module is now effectively awake. The module IDLE state is functional.
3. The clock domain switches to IDLE_TRANSITION state. In turn, the PRCM module requests the module to go into IDLE state by asserting the IDLE request signal. When acknowledged, the clock to the module can be gated, depending on other modules sharing the same clock. The functional clock of the module remains enabled because the module is in enabled mode.
4. The clock domain does not have all conditions to complete the sleep transition and wakes up again. In turn, the interface clock to the module is automatically restarted, and then the module is put out of IDLE state.
5. Software disables the module. The PRCM module requests the module to go to IDLE state by asserting the IDLE request signal. When acknowledged, the clock to the module can be gated, depending on other modules sharing the same clock.
6. The clock domain switches to IDLE_TRANSITION state. When the sleep transition conditions of the clock domain are satisfied, the clocks (functional and interface) are gated. The clock domain then switches to INACTIVE state. This has no effect on the module, which is already in FULL IDLE state.

Figure 3-6 shows the behavior of the same slave module, receiving the interface and functional clocks, when the module mode is changed while the clock domain state is IDLE_TRANSITION.

Figure 3-6. Clock Domain/Slave Module Clock-Management Interaction Sequence 2

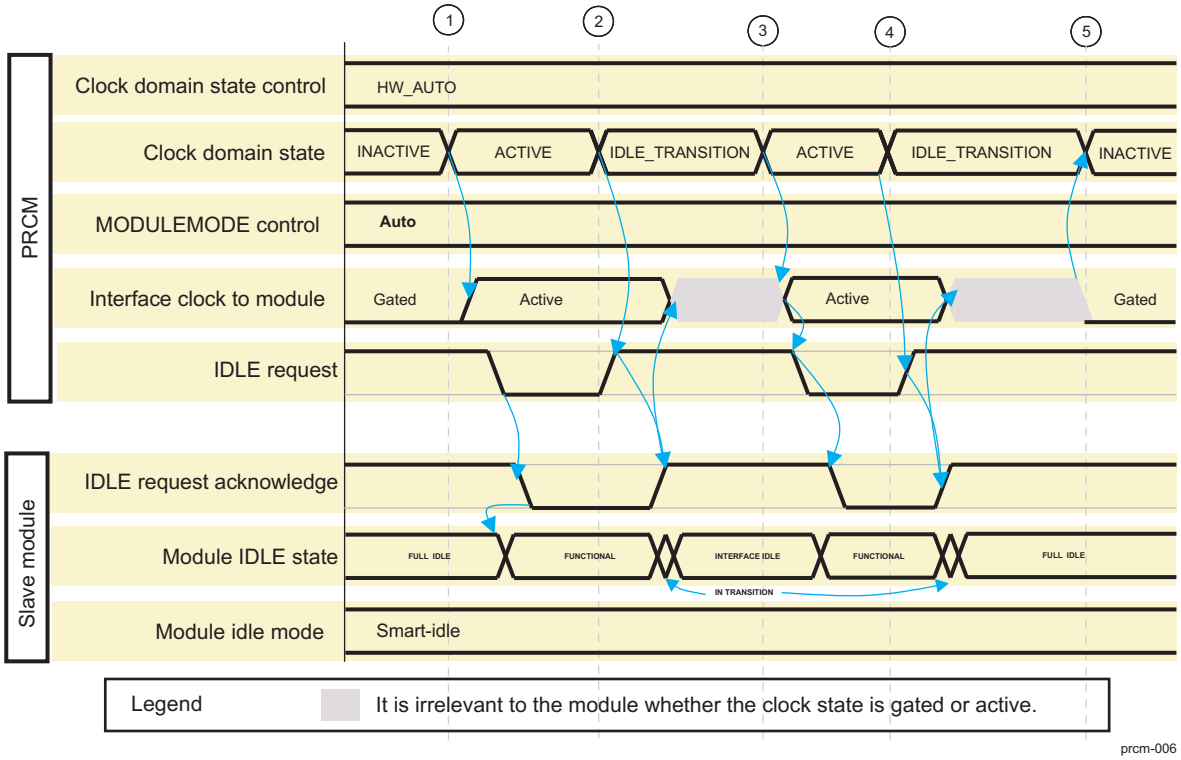


Initially, the clock domain, which includes the slave module, is inactive. The module is in FULL IDLE state and its functional and interface clocks are gated.

1. The clock domain wakes up and changes its state to ACTIVE. Because the MODULEMODE control is disabled, this event has no effect on the state of the module. The functional and interface clocks can still be restarted automatically, based on the requirements of other modules sharing these clocks.
2. The clock domain goes into the IDLE_TRANSITION state. Because the module mode control is disabled, this event has no effect on the module state.
3. Software changes the module mode to ENABLED. The clocks to the module are restarted automatically and then the module is put out of IDLE state. As soon as acknowledged, the module is requested to go back to IDLE state with gating of the interface clock only (that is, the INTERFACE IDLE state). The interface clock to the module can be gated, depending on other modules sharing the same clocks.
4. Software disables the module. The interface clock to the module is restarted automatically. The PRCM module requests the module to go out of IDLE state by asserting the IDLE request signal. As IDLE request acknowledge is de-asseted, PRCM set back the module to IDLE state. When acknowledged, the interface and the functional clocks to the module can be gated, depending on other modules sharing the same clock.
5. When the clock domain sleep transition conditions are satisfied and the functional and interface clocks are gated, the clock domain switches to INACTIVE state. This has no effect on the module, which is already in FULL IDLE state.

Figure 3-7 shows the behavior of a slave module receiving only interface clock and supporting the configurable auto module mode.

Figure 3-7. Clock Domain/Slave Module Clock-Management Interaction Sequence 3



Initially, the clock domain, which includes the slave module, is inactive. The module is in FULL IDLE state and its interface clock is gated.

1. The clock domain wakes up and changes its state to ACTIVE. In turn, the interface clock to the module is automatically restarted. The PRCM module then deasserts a hardware IDLE request signal to the module. The module sends an IDLE request acknowledge to the PRCM module. The module is now effectively awake; that is, the module IDLE state is functional.
2. The clock domain switches to IDLE_TRANSITION state. The PRCM module requests the module to go to IDLE state by asserting the IDLE request signal. When acknowledged, the interface clock to the module can be gated, depending on other modules sharing the same clock.
3. The clock domain does not have all conditions to complete the sleep transition and wakes up again. In turn, the interface clock to the module is automatically restarted, and then the module goes out of IDLE state.
4. This step is the same as Step 2.
5. The clock domain has all conditions to complete the sleep transition. The module is in IDLE state and its clock is gated.

3.1.1.1.6 Clock Domain Sleep/Wake-up

The clock domain manager initiates a domain wake-up transition when the conditions listed in Table 3-14 are satisfied.

Table 3-14. Clock Domain Wake-Up Conditions

Relation	Condition
OR	The SW_WKUP clock transition mode for the clock domain is set (CLKTRCTRL = 0x2).
	At least one wake-up request is asserted by one of the modules of the clock domain.
	At least one dynamic dependency(1) from another clock domain is active.
	At least one static dependency(1) from another clock domain is active.

(1) The dynamic, static, and wake-up dependencies are explained in Section 3.1.1.1.7, *Clock Domain Dependency*.

Table 3-14. Clock Domain Wake-Up Conditions (continued)

Relation	Condition
	At least one wake-up dependency[1] from a module in another clock domain is active.

The clock domain manager initiates a domain sleep transition when the conditions listed in Table 3-15 are satisfied.

Table 3-15. Clock Domain Sleep Conditions

Relation		Condition
AND		All master modules in the clock domain are in STANDBY state.
		No wake-up request is asserted by any module of the clock domain.
		No dynamic domain dependency[1] from any other domain is active.
		No wake-up dependency[1] from any module in another domain is active.
		No static domain dependency[1] from any other domain is active.
	OR	The SW_SLEEP clock transition mode is set for the clock domain (CLKTRCTRL = 0x1).
		The HW_AUTO clock transition mode is set for the clock domain (CLKTRCTRL = 0x3).

(1) The dynamic, static, and wake-up dependencies are explained in Section 3.1.1.1.7, *Clock Domain Dependency*.

3.1.1.1.7 Clock Domain Dependency

A domain dependency is a binary relationship between two clock domains. A clock domain A is said to depend on a clock domain B when a module in clock domain B provides services to a module in clock domain A. As a result, clock domain B must be active when clock domain A is active so that the module in clock domain B is accessible by the module in clock domain A.

Dependency between two clock domains can also exist if one clock domain serves to ensure communication between two modules (for example, the clock domain of the device interconnect).

Thus, a clock domain can support the types of clock domain dependencies described in the following sections.

Table 3-16 and Table 3-17 detail all the domain dependencies:

- **NA/NA**: if no dependency can exist because no corresponding interconnect path exists in the device
- When cell is different than NA/NA, the cell contains two attributes:
  - First attribute for the presence and control method of a static dependency:
    - **SW**: Static dependency is controlled by a software bit. This is the most generic way of control. Depending on the use case and on latency requirement for accessing target domain, software can enable or not the static dependency. When not enabled, access to those domains is performed using dynamic dependencies.
    - **1**: Static dependency is always enabled (hard-wired). This is relevant for a domain that has an "exact standby" system initiator with the target domain being the domain containing the interconnect module to which the initiator is connected.
    - **0**: Static dependency is never enabled (hard-wired). This is relevant for domains that do not have strong access latency requirements, or for which a static dependency is not desired for specific reasons (for example, dependencies with emulation domain). Access to those domains is performed using dynamic dependencies.
    - **NA**: Nonapplicable (means domain has no system initiator able to access the other domain)
  - Second attribute for the presence and control method of a dynamic dependency:
    - **1..n**: Dynamic dependency is always enabled (hard-wired). The number of corresponding interconnect interfaces is also specified. This is relevant for most of the domain-to-domain direct interconnect connection.
    - **0**: Dynamic dependency is never enabled (hard-wired). This is relevant only when a static

dependency is always enabled between same domains, or when the target domain cannot support the wakeup on access feature.

- **NA**: Non-applicable (means that both domains are not directly linked by an OCP interface)

Table 3-16. Device Domain Dependencies (Table 1)

Static/dynamic dependencies from below domains to right-side domains	L4CFG	CRC	L3INSTR	L3MAIN1	COREAON	CUSTEFUSE	DSP1	DSP2	DSS	EVE1
CAM	0/NA	0/NA	0/NA	1/0	0/NA	0/NA	0/NA	0/NA	0/NA	0/NA
L4CFG	NA/NA	NA/NA	0/0	0/35	0/6	0/1	0/NA	0/NA	NA/NA	NA/NA
DSP1	0/NA	SW/NA	0/NA	SW/3SW/3	0/NA	0/NA	NA/NA	SW/NA	SW/NA	SW/NA
DSP2	0/NA	SW/NA	0/NA	SW/3SW/3	0/NA	0/NA	SW/NA	NA/NA	SW/NA	SW/NA
L3INSTR	NA/NA	NA/NA	NA/NA	NA/NA	NA/NA	NA/NA	NA/NA	NA/NA	NA/NA	NA/NA
L3MAIN1	0/1	0/NA	0/0	NA/NA	0/NA	0/NA	0/1	0/1	0/2	0/1
DSS	0/NA	0/NA	0/NA	1/0	0/NA	0/NA	0/NA	0/NA	NA/NA	NA/NA
EMU	0/NA	0/NA	0/NA	0/1	0/NA	0/NA	0/NA	0/NA	0/NA	0/NA
EVE1	0/NA	0/NA	0/NA	1/0	0/NA	0/NA	0/NA	0/NA	0/NA	NA/NA
IPU1	SW/NA	SW/NA	NA/N	SW/1	0/NA	0/NA	SW/NA	SW/NA	SW/NA	SW/NA
GMAC	NA/NA	0/NA	NA/NA	1/0	0/NA	0/NA	0/NA	0/NA	NA/NA	NA/NA
L3INIT	SW/NA	0/NA	0/NA	1/0	0/NA	0/NA	0/NA	0/NA	0/NA	NA/NA
L4PER1	NA/NA	NA/NA	NA/NA	NA/NA	NA/NA	NA/NA	NA/NA	0/NA	0/1	NA/NA
L4PER2	NA/NA	0/1	NA/NA	1/0	NA/NA	NA/NA	SW/NA	SW/NA	NA/NA	NA/NA
L4PER3	NA/NA	NA/NA	NA/NA	0/4	NA/NA	NA/NA	0/NA	0/NA	NA/NA	NA/NA
ISS[(1)]	NA/NA	0/NA	NA/NA	1/0	NA/NA	NA/NA	NA/NA	NA/NA	NA/NA	NA/NA

(1) ISS is not supported on the DRA78x family of devices.

Table 3-17. Device Domain Dependencies (Table 2)

Static/dynamic dependencies from below domains to right-side domains	IPU1	CAM	EMU	IPU	GMAC	L3INIT	L4PER1	L4PER2	L4PER3	EMIF	ISS[(1)]	WKUPAON
CAM	0/NA	NA/NA	0/NA	0/NA	0/NA	0/NA	0/NA	0/NA	0/NA	SW/NA	0/NA	0/NA
L4CFG	NA/NA	0/NA	NA/NA	NA/NA	NA/NA	0/2	NA/NA	NA/NA	NA/NA	0/1	NA/NA	NA/NA
DSP1	0/NA	SW/NA	0/NA	SW/NA	SW/NA	SW/NA	SW/NA	SW/NA	SW/NA	SW/NA	SW/NA	SW/NA
DSP2	0/NA	SW/NA	0/NA	SW/NA	SW/NA	SW/NA	SW/NA	SW/NA	SW/NA	SW/NA	SW/NA	SW/NA
L3INSTR	NA/NA	NA/NA	0/0	NA/NA	NA/NA	NA/NA	NA/NA	NA/NA	NA/NA	NA/NA	NA/NA	NA/NA
L3MAIN1	0/1	0/0	0/NA	0/NA	0/NA	0/NA	0/1	0/4	0/1	0/2	NA/NA	0/1
DSS	0/NA	0/NA	0/NA	NA/NA	NA/NA	0/NA	0/NA	NA/NA	NA/NA	SW/NA	NA/NA	0/NA
EMU	0/NA	0/NA	NA/NA	0/NA	NA/NA	0/NA	0/NA	NA/NA	NA/NA	0/NA	NA/NA	0/NA
EVE1	0/NA	NA/NA	NA/NA	NA/NA	NA/NA	NA/NA	NA/NA	NA/NA	NA/NA	SW/NA	NA/NA	NA/NA
IPU1	NA/NA	0/NA	NA/NA	NA/NA	SW/NA	SW/NA	SW/NA	SW/NA	SW/NA	SW/NA	SW/NA	SW/NA

(1) ISS is not supported on the DRA78x family of devices.

Table 3-17. Device Domain Dependencies (Table 2) (continued)

Static/dynamic dependencies from below domains to right-side domains	IPU1	CAM	EMU	IPU	GMAC	L3INIT	L4PER1	L4PER2	L4PER3	EMIF	ISS[1]	WKUPAON
GMAC	0/NA	NA/NA	NA/NA	0/NA	NA/NA	NA/NA	NA/NA	SW/NA	NA/NA	SW/NA	NA/NA	NA/NA
L3INIT	0/NA	0/NA	0/NA	NA/NA	NA/NA	NA/NA	SW/NA	NA/NA	SW/NA	SW/NA	NA/NA	SW/NA
L4PER1	0/NA	NA/NA	NA/NA	0/2	NA/NA	0/2	NA/NA	NA/NA	NA/NA	NA/NA	NA/NA	NA/NA
L4PER2	SW/NA	NA/NA	NA/NA	0/1	0/1	0/1	NA/NA	NA/NA	NA/NA	NA/NA	NA/NA	NA/NA
L4PER3	0/NA	0/3	NA/NA	0/40/4	NA/NA	0/8	NA/NA	NA/NA	NA/NA	NA/NA	0/1	NA/NA
ISS[1]	0/NA	NA/NA	NA/NA	NA/NA	NA/NA	NA/NA	NA/NA	NA/NA	SW/NA	SW/NA	NA/NA	NA/NA

When a static dependency is hardware-coded between two domains directly linked by one or several interconnect interfaces, then the corresponding dynamic dependency is useless.

Emulation domain (DAP initiator) has no static dependency with any other domain. It has, however, a dynamic dependency with the L3_MAIN2 interconnect. A domain that can access an emulation domain does not have static or dynamic dependency with it.

Domain dependencies are chosen such that any access (even from EMU/DAP) towards a nondisabled target is always completed normally. Using static dependencies allows having minimal access latencies by keeping necessary domains on whenever the initiator is not standby. This may be at the expense of additional power consumption because some domains may stay on while not in use for a long time. By disabling static dependencies, applicative access is still completed normally by waking up, if applicable, the necessary domain on the path from the initiator to the target. Power consumption can be optimized at the expense of additional access latencies.

NOTE: Once the CM_<Clock domain>_CLKSTCTRL[1:0] CLKTRCTRL has been set to SW_WKUP, then SW has to poll for PM_<Power_domain>_PWRSTST[1:0] PowerStateSt = 0x03 and PM_<Power_domain>_PWRSTST[20] InTransition = 0x00 before update MODULEMODE bit field.

3.1.1.1.7.1 Static Dependency

If clock domain A has a master module that can access a slave module in clock domain B, then clock domain A can have a static dependency with clock domain B. Similarly, a static dependency can also exist between domain A and B if domain B conveys the transactions from a domain A module toward a module in any other domain. For example, CD_DSP can have a static dependency with CD_L3_MAIN1 because this domain has a level 3 (L3) interconnect to carry the transactions from the digital signal processor (DSP) module.

This static dependency consists of forcing clock domain B to stay active as long as there is at least one master module of clock domain A that is not in STANDBY state. If clock domains A and B are initially in GATED state, then clock domain B becomes active as soon as clock domain A becomes active when a wake-up request from the master module is received by the PRCM module.

Similarly, as a result of the static dependency, clock domain B can be gated only if all the master modules of clock domain A that can access the slave modules in clock domain B are in STANDBY state.

The static dependency between a source clock domain and a destination clock domain is configured in the PRCM module by setting the CM_<Source Clock domain>_STATICDEP[x] <Destination Clock domain>_STATDEP bit. As a result, the source clock domain forces the destination clock domain to become active and stay active as long as the source clock domain is active.

The destination domain must be put into forced wake-up (CM_<X>_CLKSTCTRL[1:0] CLKTRCTRL = SW_WKUP) before changing a configurable static dependency.

3.1.1.1.7.2 Dynamic Dependency

When clock domains A and B contain modules directly linked to a common device interconnect, these clock domains can have a dynamic dependency.

A dynamic dependency consists of forcing clock domain B to stay active as long as a module from clock domain A is communicating with the module in clock domain B through the interconnect. Clock domain B becomes active as soon as the communication is initiated. This is automatically managed by the PRCM module by monitoring the communication on the interconnect between the modules of the two clock domains.

Similarly, the inverse condition of this dependency can be stated: Clock domain B can be inactive only if there has not been transactions from clock domain A to clock domain B, identified as a sliding window duration on the interconnect activity status.

The size of the sliding window is based on the number of cycles of a prescaled level 4 (L4) clock whose frequency is configured by setting the CM_DYN_DEP_PRESCAL[5:0] PRESCAL bit field. The prescaled clock frequency is given as:

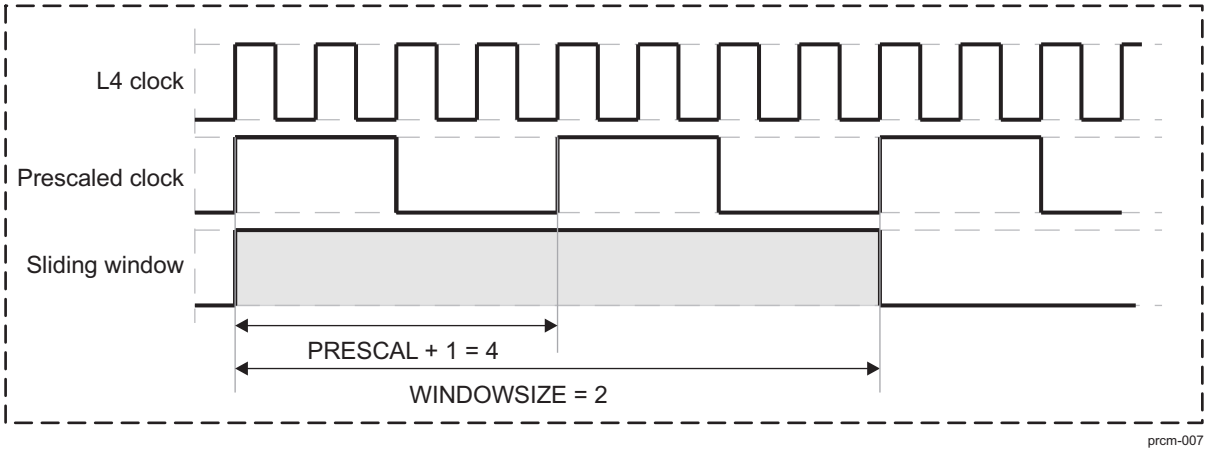
Prescaled clock frequency = L4 interface clock frequency / (PRESCAL + 1)

The size of the sliding window is fixed by setting the CM_<Clock domain>_DYNAMICDEP[27:24] WINDOWSIZE bit field. It is given as:

Sliding window duration = WINDOWSIZE ×Period of Prescaled clock cycle

Figure 3-8 is an example of the sliding window duration equal to eight clock cycles of an L4 clock when PRESCAL is set to 3 and WINDOWSIZE is set to 2.

Figure 3-8. Sliding Window for Dynamic Dependency



This dynamic dependency is also referred to as the autosleep/autowakeup feature.

NOTE:

- The static dependency between two clock domains can be configured by software (PRCM module registers) or hardwired in the PRCM module.
- The dynamic dependency between two clock domains is hardwired in the PRCM module.

A dynamic dependency is said to be active when both of the following conditions are met:

- There has been one or more transaction on the interconnect within the sliding window duration.

Otherwise, a dynamic dependency is said to be inactive.

The dynamic dependency between a source clock domain and a destination clock domain can be read in the PRCM module from the corresponding read-only CM_<Source Clock domain>_DYNAMICDEP[x] <Destination Clock domain>_DYNDEP bit.

NOTE: It is recommended to use dynamic dependencies. They give better power results. Static dependencies should be rarely used (in some cases they can be used as they give shorter latency for a system initiator to access slave).

3.1.1.1.7.3 Wake-Up Dependency

A wake-up dependency is a dependency between the clock domain of a module that owns one or several wake-up signals toward the clock domain of another module needed to service the associated wake-up event. As a result of this dependency, the wake-up event to a module activates not only its clock domain but also the clock domain of the servicing module.

NOTE: To ensure that the clock domain of the servicing module remains active, the wake-up signal that triggers a wake-up dependency stays active as long as the source of the event is not serviced.

Wake-up dependencies allow acceleration of the wake-up transition of multiple domains needed to service the wake-up event by initiating their transition in parallel. The static and dynamic dependencies can allow the wake-up of related domains, but the complete wake-up transition of all the associated domains is slower because of the sequential cascading of their wake-up transitions.

In the device, the source event of the wake-up signal to a slave module can be either of following types:

- Interrupt request to the DSP, or image processor unit (IPU) interrupt controller (INTC)
- DMA request to a DMA controller

Upon wake-up by these types of wake-up events, and for as long as they remain asserted, the PRCM module takes the following actions:

- The power domain of the servicing module (for example DSP, IPU, or DMA) is forced to POWER ON state and the clock domain becomes active.
- The power domain of the device interconnect between the servicing module and the module originator of the wake-up event is forced to POWER ON state and the clock domain becomes active.
- The power domain of the slave module originator of the wake-up event is forced to POWER ON state and the clock domain becomes active.
- The slave module originator of the wake-up event is switched from IDLE to ACTIVE state.

On assertion of a wake-up event of a stand-alone master module, and as long as it remains asserted, the PRCM module takes the following action:

- The power domain of the master module originator of the wake-up event is forced to POWER ON state and the clock domain becomes active.

NOTE: For slave modules, the static and dynamic dependencies of a clock domain are not affected by its wake-up dependency settings. For master modules, the static dependencies are not affected.

Hence, in addition to the activation of the clock domains previously described, all clock domains associated by static dependencies are also activated.

However, the clock domains associated with the wake-up clock domain through dynamic dependencies are activated only if a transaction is initiated to these clock domains.

For each wake-up signal coming from a slave module, the type of the corresponding event can be configured in the PM_<Power domain>_<Originator Module> _WKDEP[x] WKUPDEP_<Originator Module>_[IRQ/DMA]_<Servicing Module> bit of the PRCM module, where <Power domain> is the name of the power domain of the originator module of the wake-up event identified as <Originator Module>. <Servicing Module> refers to the module servicing the wake-up event.

NOTE: When only one event type is associated with the wake-up signal of a slave module, the wake-up dependency (WKUPDEP) for the module clock domain is not configurable and may be hardwired in the PRCM module.

For the master modules, there is no configurable wake-up dependency. Their power domain is switched on and their clock domain is activated by the PRCM module when they assert their wake-up signal.

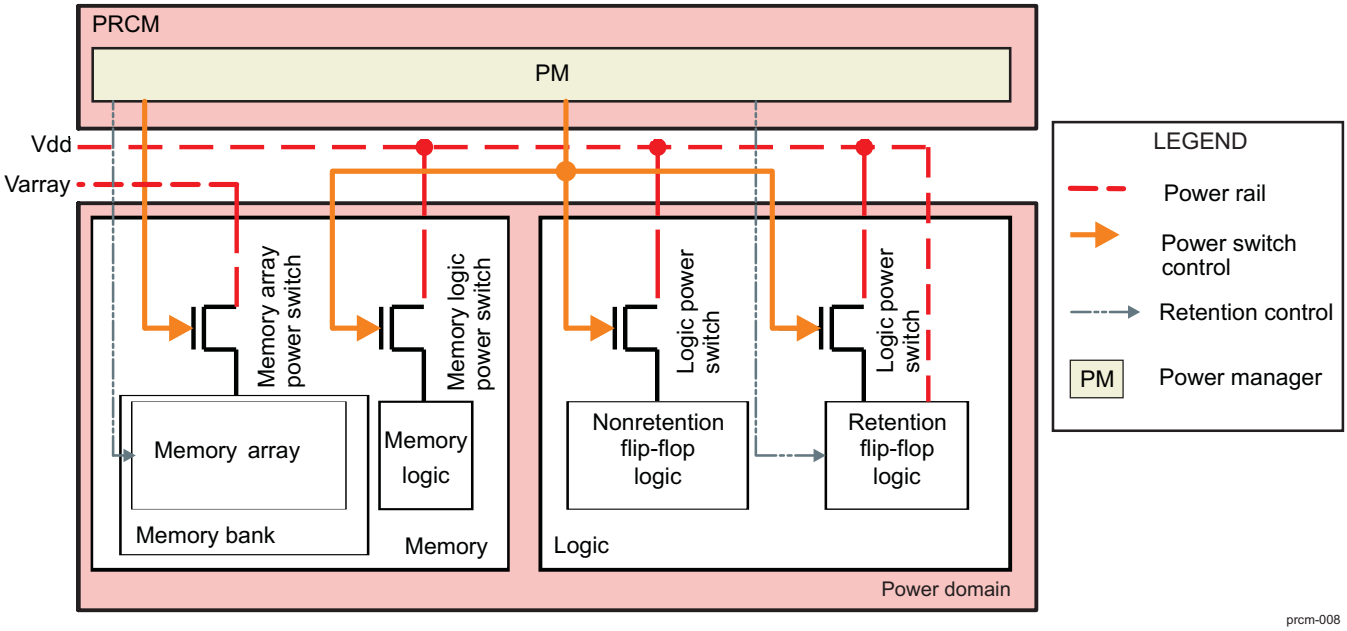
3.1.1.2 Power Management

The PRCM module manages the switching on and off of the power supply to the device modules. To minimize device power consumption, the power to the modules can be switched off when they are not in use. Independent power control of sections of the device lets the PRCM module turn on and off specific sections of the device without affecting other sections. Power management integration details are described in Section 3.7, *Power Management Functional Description*.

3.1.1.2.1 Power Domain

A power domain is a section (that is, a group of modules) of the device with an independent and dedicated power manager (see Figure 3-9). A power domain can be turned on and off without affecting other parts of the device.

Figure 3-9. Generic Power Domain



To minimize device power consumption, the modules are grouped into power domains. A power domain can be split into logic and memory areas.

The memory area contains two entities:

- Memory bank: Composed of memory arrays. It is powered by a dedicated voltage rail and an associated power switch (for example, Varray and memory array power switches).
- Memory logic: Powered by the same voltage source as the logic area of the power domain, but has its dedicated power switch (for example, Vdd and memory logic power switches)

The logic area in the power domain can also be split between retention flip-flops (RFFs) and nonretention flip-flops (DFFs).

Table 3-18 lists the possible states and substates of the logic area in a power domain.

Table 3-18. States of a Logic Area in a Power Domain

State	Substate	Description
ON	ON-ACTIVE	Logic is fully powered and at least one enclosed clock domain is active.
	ON-INACTIVE	Logic is fully powered and all enclosed clock domains are idled.
RETENTION	CSWR (close switch retention)	Logic is fully powered and all enclosed clock domains are idled.
OFF		Logic power switches are off. All the logic (DFF and RFF) is lost except for the context, which has been saved in the scratchpad memory of an always-on power domain.

RETENTION state is useful for quickly switching to low-power idle mode (in which the domain clocks are gated and the domain voltage is less than the on-voltage level) without losing the context, and then quickly switching back to ON-ACTIVE state when necessary. In RETENTION state, power consumption is less than in ON power state.

The behavior of the memory array power switch and memory logic power switch can be selected through software settings in the PRCM module or can be hardwired. Once the behavior is selected, the PRCM module hardware automatically handles these elements to ensure correct power transition sequencing between the power domain states.

Software can also initiate power state changes of the memory array when the associated power domain is in ON power state. This allows the memory array to be turned off and on as needed.

The memory area can be configured to any of the power states listed in Table 3-19.

Table 3-19. States of a Memory Area in a Power Domain

State	Description
ON	The memory array is powered and fully functional.
RETENTION	The memory array is fully powered, but memory is not accessible. The array can be put into retention through an applicable direct retention control signal. Data in memory are always retained.
OFF	The memory array is powered down. Data in memory are lost.

3.1.1.2.2 Module Logic and Memory Context

In case of a power state transition in the logic or memory areas, the context of the module may no longer be valid. This can also be the case when the domain resets are asserted by the device. A specific RM_<Clock Domain Name>_<Module Name>_CONTEXT register provides the status of the device logic and memory context.

- The module logic context consists of simple flip-flops (DFFs) if the module has no logic RFFs.
- If the module has logic retention (full or partial), it is assumed that the context consists of: only RFFs, only DFFs or both RFFs and DFFs.

NOTE: The display subsystem is an exception where the status of DFF and RFF context is given, because only HMDI keys are retained, while most of the display subsystem is not retained.

These context status bits must be cleared by software.

3.1.1.2.3 Power Domain Management

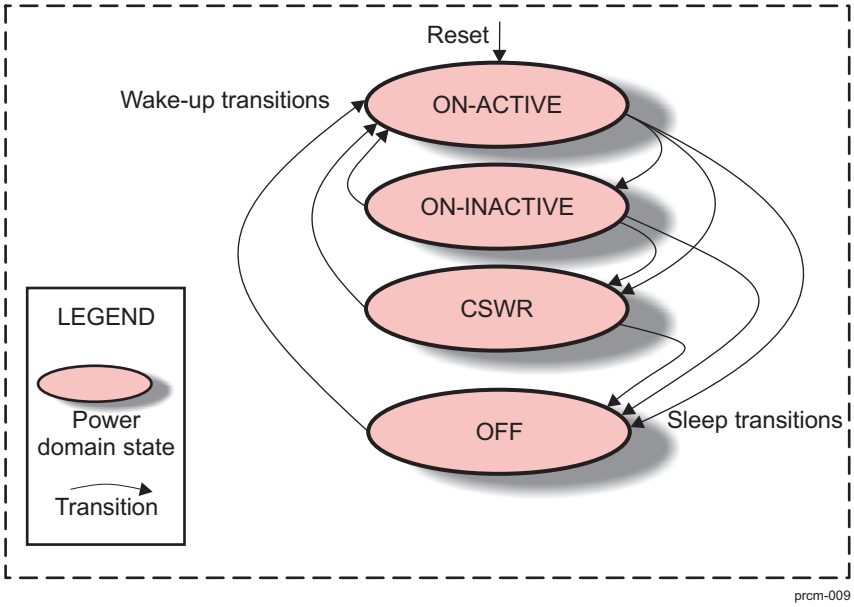
The power manager associated with each power domain is assigned the task of managing the domain power transitions. It ensures that all hardware conditions are satisfied before it can initiate a power domain transition from a source to a target power state (for example, from ON-ACTIVE state to CSWR RETENTION state).

The hardware condition for power domain transition from ON-ACTIVE to any other transition state is:

- All clock domain managers in the power domain are in IDLE state.

Figure 3-10 shows all possible power domain state transitions.

Figure 3-10. Power Domain Transitions



Successive power-down transitions can be performed by lowering the power state from ON-ACTIVE to ON-INACTIVE to RETENTION, and then to OFF and LOWPOWERSTATECHANGE, as long as the hardware condition is satisfied.

However, the power domain wake-up transition from any low-power state (ON-INACTIVE, CSWR or OFF) to ON-ACTIVE state is always direct.

The power domain manager initiates a power domain wake-up transition when the conditions listed in Table 3-20 are satisfied.

Table 3-20. Power Domain Wake-Up Conditions

Relation		Condition
AND		Voltage domain is on.
	OR	There is at least a wake-up condition for one enclosed functional clock domain.
		There is a request for clock generation or distribution enclosed in the power domain.
		There is a PRCM module service request (applicable only to power domains, including PRCM module logic).

The power domain manager initiates a domain sleep transition when the conditions listed in Table 3-21 are satisfied.

Table 3-21. Power Domain Sleep Conditions

Relation	Condition
AND	All functional clock domains enclosed in the power domain are idled.
	All clock generation or distribution enclosed in the power domain is quiet, and corresponding input clocks are gated. For example, DPLL, if present, must be in stop mode.
	There is no PRCM module service request (applicable only to power domains, including PRCM module logic).

Table 3-22 lists the control and status features of the PRCM module power domain.

Table 3-22. Power Domain Control and Status Registers

Register/Bit Field	Type	Description
PM_<Power domain>_PWRSTCTRL[1:0] POWERSTATE	Control	Selects the target power state of the power domain among OFF, ON-ACTIVE, ON-INACTIVE, and RETENTION
PM_<Power domain>_PWRSTCTRL[x] LOWPOWERSTATECHANGE	Control	Power state change request when domain has already performed a sleep transition. Allows going into deeper low-power state without waking up the power domain.
PM_<Power domain>_PWRSTCTRL[2] LOGICRETSTATE	Control	Selects whether the power domain logic is in CSWR or RETENTION state when the domain transitions to RETENTION state
PM_<Power domain>_PWRSTCTRL[x] <memory bank>_RETSTATE	Control	Selects whether the memory bank in the power domain is in ON, or RETENTION state when the power domain is in RETENTION state. The memory bank cannot be in ON state when the power domain is in RETENTION state.
PM_<Power domain>_PWRSTCTRL[x] <memory bank>_ONSTATE	Control	Selects whether the memory bank is in ON, RETENTION, or OFF state when the power domain is in ON state
PM_<Power domain>_PWRSTST[1:0] POWERSTATEST	Status	Identifies the current state of the power domain. It can be OFF, RETENTION, ON-INACTIVE, or ON-ACTIVE.
PM_<Power domain>_PWRSTST[2] LOGICSTATEST	Status	Identifies the current state of the logic area in the power domain. It can be OFF or ON.
PM_<Power domain>_PWRSTST[20] INTRANSITION	Status	Identifies whether a power state transition in the power domain is in progress or there is no ongoing transition
PM_<Power domain>_PWRSTST[x] <memory bank>_STATEST	Status	Identifies the current power state of the memory bank in the power domain. It can be OFF, RETENTION, or ON.
PM_<Power domain>_PWRSTST[25:24] LASTPOWERSTATEENTERED	Status	Identifies the last (previous) power state of the power domain. It can be OFF, RETENTION, ON-INACTIVE, or ON-ACTIVE.

3.1.1.3 Voltage Management

The PRCM module do not provide controls over the Voltage management. All OPP changes will be handled by Application software directly without any PRCM intervention.

Voltage management integration details are described in Section 3.8, *Voltage Management Functional Description*.

3.1.1.3.1 Voltage Domain

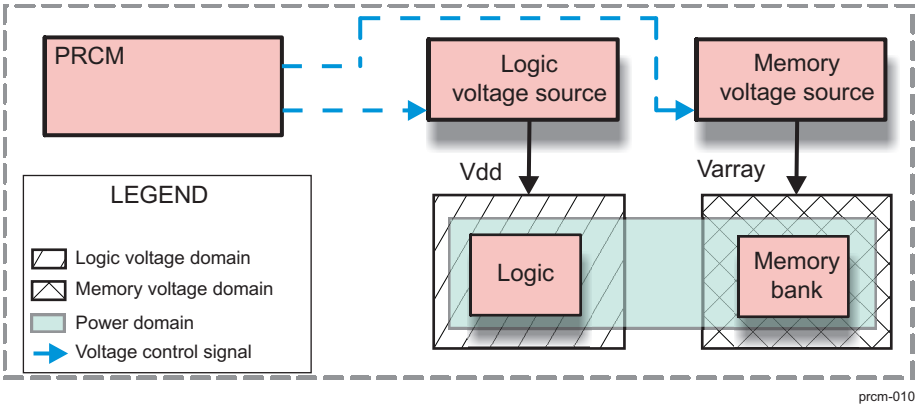
A voltage domain is a section of the device supplied by a dedicated voltage source (that is, an internal LDOs or external switch mode power supply [SMPS]). A voltage domain may or may not be controlled by the PRCM module.

The voltage managers in the PRCM module is one type:

- Dynamically configurable by software to scale the domain voltage level to specific values within the operational voltage range of the device. This is called adaptive voltage scaling (AVS).

Figure 3-11 shows a voltage domain.

Figure 3-11. Generic Voltage Domain

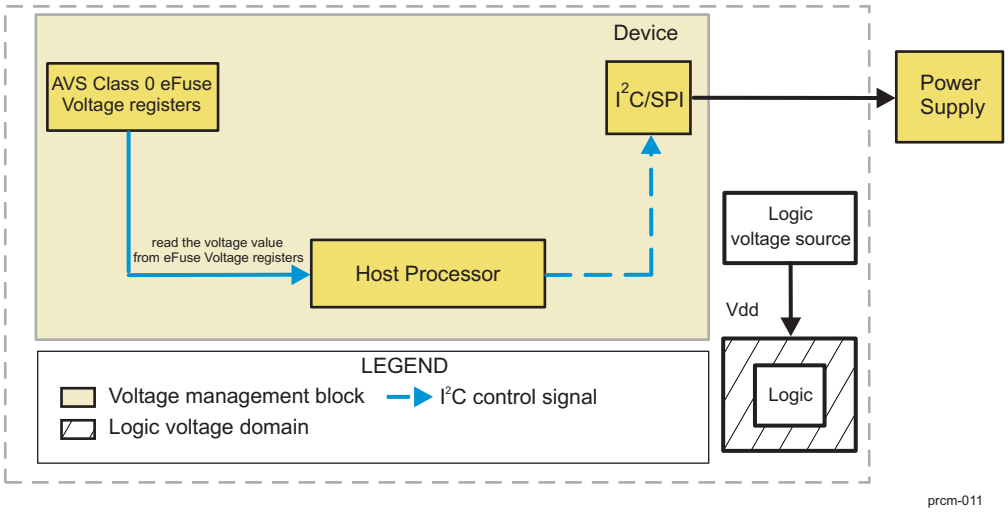


By partitioning the device into independent voltage domains, different operating voltages can be assigned to the different sections of the device (that is, a group of modules or memory banks). The independent voltage control allows voltage scaling of device subsections to ensure that each module or memory bank operates at the optimized operating voltage level based on the application performance requirements.

3.1.1.3.2 Voltage Domain Management

Figure 3-12 shows the different voltage control paths available within a generic logic voltage management block to control the voltage supply to the logic voltage domains of the device.

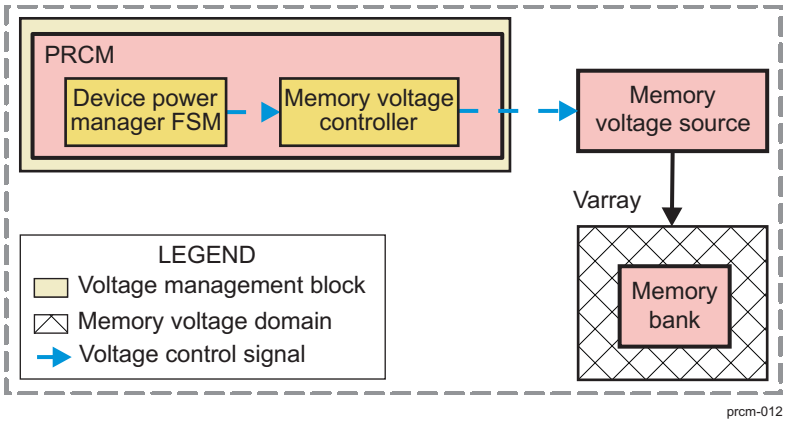
Figure 3-12. Generic Logic Voltage Management



NOTE: For more information about AVS Class 0 eFuse Voltage registers, see chapter AVS Class 0 Associated Registers in the Control Module.

Figure 3-13 shows the voltage control path available within a generic memory voltage management block to control the voltage supply to the memory voltage domains of the device.

Figure 3-13. Generic Memory Voltage Management



The PRCM hardware supports automatic scaling down of the memory array supply whenever the memory domains transition to RETENTION power state. The device power manager FSM manages the voltage scaling of memory voltage domains through the memory voltage controller (or LDO).

3.1.1.3.3 AVS Overview

Adaptive Voltage Scaling (AVS) Class 0 (also referred to as SmartReflex™) is a power-management technique used to control the operating voltage of a device to reduce its active power consumption.

With SmartReflex, the power supply voltage is adapted to the silicon performance in one way:

- Statically adapted to the manufacturing process of a given device

SmartReflex achieves optimal performance/power trade-off for all devices across the technology process spectrum.

The static correction of the device voltage level (see Figure 3-14) is based on the desired performance level and silicon performance characteristics of the device. As a result of process dispersion, each die has its specific silicon performance. The range of the process distribution defines the weak devices (low-performance silicon) and the strong devices (high-performance silicon).

A weak device is a device with the lowest performance tolerated for a process distribution; that is, at the typical voltage, the inherent maximum frequency is the lowest frequency of the chip distribution. Considered as the worst case, weak devices are used to constrain the target frequency of all the chips (OPP definition).

A strong device is a device with the highest performance tolerated for a process distribution. The inherent maximum frequency at the typical voltage is greater than the targeted frequency.

Figure 3-14 shows that with the SmartReflex voltage-control architecture, it is possible to compensate for the device silicon characteristics and obtain optimal performance characteristics. Based on the device characteristics, the device voltage level can be adjusted for specific performance level.

Figure 3-14. SmartReflex Static Voltage Adjustment

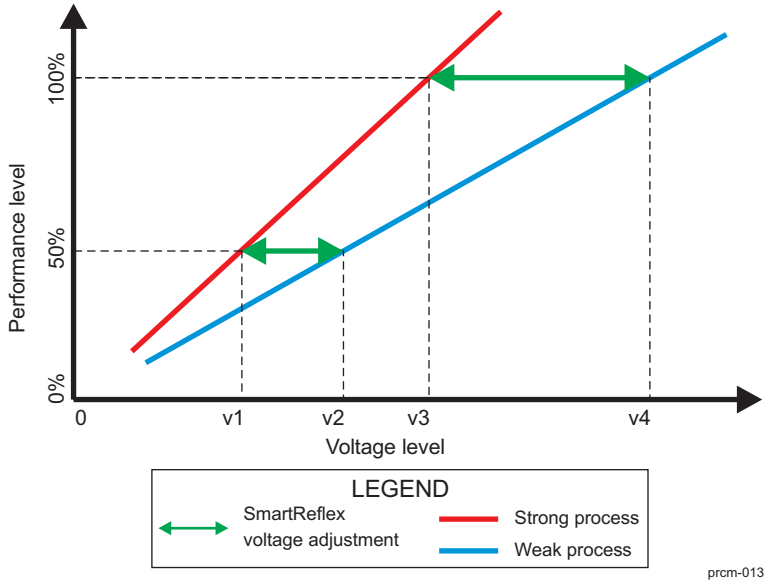
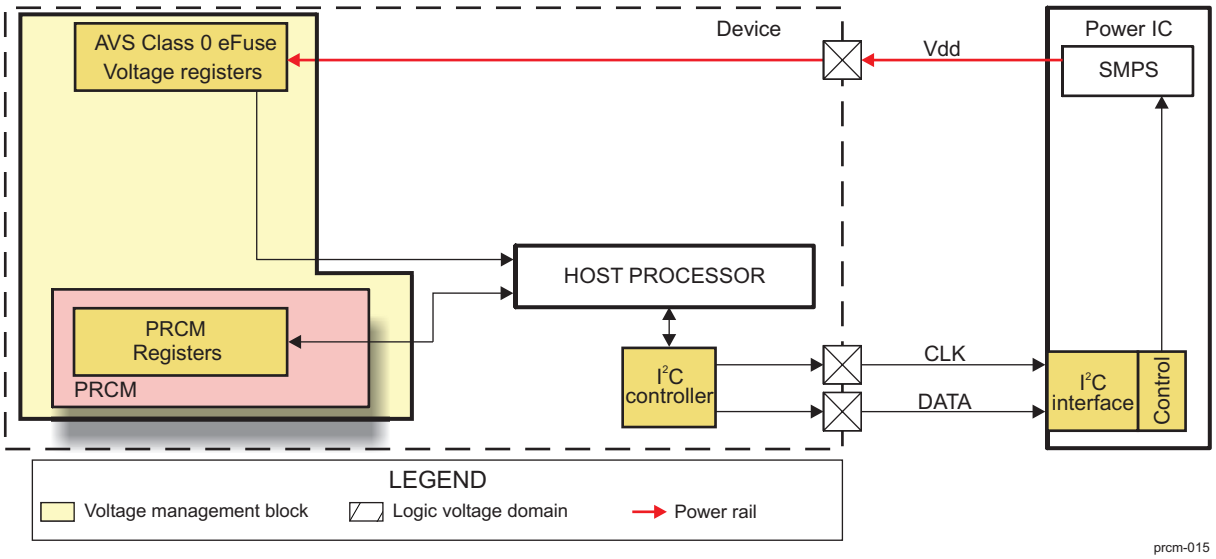


Figure 3-15 is a functional overview of the SmartReflex voltage-control architecture of the device connected to an external power IC.

Figure 3-15. SmartReflex Voltage Control Functional Overview



SmartReflex voltage control consists of the following modules:

- Host Processor
- I^2C interface
- SMPS

The device supports one operational mode for SmartReflex voltage control:

- Class 0, Manufacturing Test Calibration

3.1.1.3.3.1 AVS Class 0 (SmartReflex™) Voltage Control

Adaptive Voltage Scaling (AVS) Class 0 (also referred to as SmartReflex) is a procedure for lowering the voltage on certain device power rails. AVS Class 0 attempts to normalize the power consumption across all devices. The optimal voltage for each AVS supported rail of each device is determined after analysis in the factory. This value is written in the device eFuse where it can be read through dedicated registers. These registers reside in the control module.

3.1.2 Power-Management Techniques

The following sections describe the power-management techniques supported by the device.

NOTE: The values in Figure 3-16 through Figure 3-18, which show the power-management techniques, are hypothetical. They do not represent valid test results on the device.

3.1.2.1 Standby Leakage Management

Standby leakage management (SLM) is a power-management technique that reduces standby power consumption by reducing power leakage.

With SLM, the device switches into low-power system modes automatically or in response to user requests during system standby (that is, in situations when no application is started and system activity is negligible or limited).

When applying SLM, the system remains in the lowest static power mode compatible with the system response time requirement.

This technique trades static power consumption for wake-up latency.

3.1.2.2 Dynamic Voltage and Frequency Scaling

Dynamic voltage and frequency scaling (DVFS) consists of minimizing the idle time of the system. The DVFS technique uses dynamic selection of the optimal operating frequency and voltage to allow a task to be performed in the required length of time. This reduces the active power consumption (power consumed while executing a task) of the device while still meeting task requirements.

NOTE: The values in Figure 3-16 are hypothetical. They are meant only to clarify the concept and do not represent valid test results on the device.

Figure 3-16. Comparison of Energy Consumed With/Without DVFS

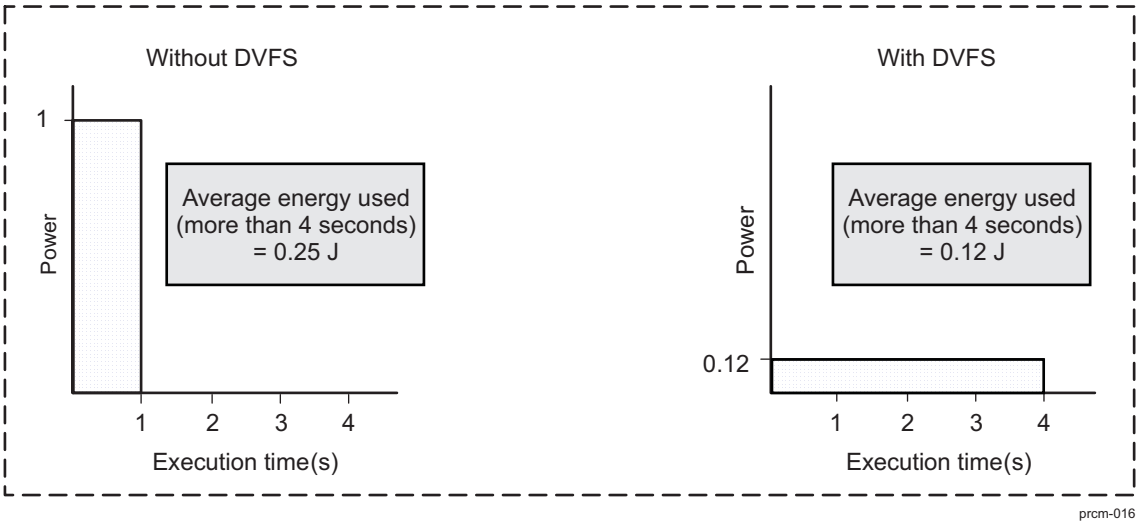


Figure 3-16 shows the DVFS technique by comparing a process executed at maximum frequency and operating voltage without applying DVFS to the same process executed at optimal frequency and voltage using DVFS, based on the task requirements. If a task that must terminate in 4 seconds is performed at maximum operating frequency (see the left side of the figure), it terminates in 1 second, and the remaining 3 seconds are spent in idle mode.

With DVFS (see the right side of the figure), the operating frequency is reduced to optimal level; the task takes the full 4 seconds to complete, but power consumption is reduced. In addition, the voltage can be reduced further to save power so the dynamic and leakage power consumption are reduced.

DVFS requires control over the clock frequency and the operating voltage of the device elements. By intelligently switching the individual elements of the device to their OPPs, the power consumption of the device for a given task can be minimized.

For practical reasons related to the development of the device (flow, tools), DVFS can be used only for a few discrete steps, not over a continuum of voltage and frequency values. Each step, or OPP, is composed of a voltage (V) and frequency (F) pair. For an OPP, the frequency corresponds to the maximum frequency allowed at a voltage, or reciprocally; the voltage corresponds to the minimum voltage allowed for a frequency.

When applying DVFS, a processor or system always runs at the lowest OPP that meets the performance requirement at a given time. The user determines the optimal OPP for a given task and then switches to that OPP to save power.

3.1.2.3 Dynamic Power Switching

Like DVFS, dynamic power switching (DPS) is a power-management technique intended to reduce the active power consumption of a device. However, whereas DVFS reduces dynamic and leakage power consumption, DPS reduces only leakage power consumption, at the expense of a slight overhead in dynamic power consumption.

With DPS, the system switches dynamically between high- and low-consumption system power modes during system active time. When DPS is applied, a processor or system runs at the highest OPP (maximum frequency and voltage) to complete its tasks quickly, followed by an automatic switch to a low-power mode for minimum power consumption. DPS is useful when a real-time application is waiting for an event. The system can switch into a low-power system mode if the wake-up latency conditions allow it.

This technique consists of maximizing the idle period of the system to reduce its power consumption.

NOTE: The values in Figure 3-17 are hypothetical. They are meant only to clarify the concept and do not represent valid test results on the device.

Figure 3-17. Comparison of Energy Consumed With/Without DPS

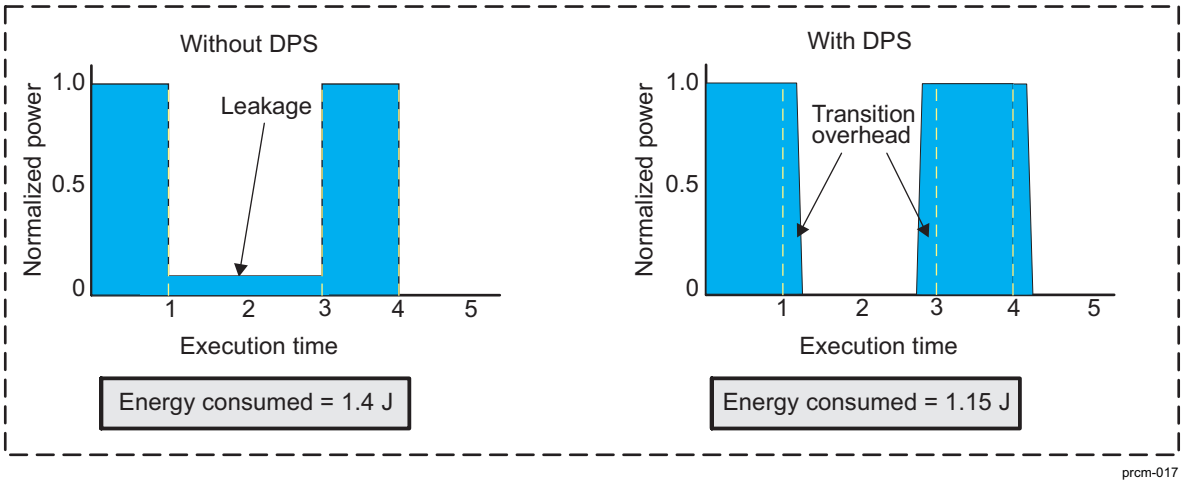


Figure 3-17 compares the behavior of power consumption for the same operation of the device without DPS (see the left side of the figure) and with DPS (see the right side of the figure). When operating without DPS, the device has a constant leakage current in idle mode. By using DPS, the system reduces the leakage current to 0. However, the transitions between system power modes may require storing the information before entering a low-power inactive state and restoring the information after a wake-up event (see Figure 3-17). This results in additional dynamic power consumption, referred to as transition overhead (see Figure 3-17). Transition overhead must be considered for a DPS operation.

For efficient deployment of DPS techniques, it is necessary to predict dynamically the performance requirement of the applications running on the processor. The DPS controller must account for the overhead of wake-up latencies related to domain switching and ensure that they do not significantly affect the performance of the device. Even with transition overhead, the user can identify an optimal idle-time limit, after which the DPS is useful for dynamic power saving.

3.1.2.4 Adaptive Voltage Scaling

With SmartReflex, power-supply voltage is adapted to silicon performance, statically (based on performance points predefined in the manufacturing process of a given device).

AVS achieves the optimal performance/power trade-off for all devices across the technology process spectrum. This ensures optimal power consumption for a given OPP.

3.1.2.5 Combining Power-Management Techniques

The power-management techniques previously described have specific features and are most effective when used under the specific operating conditions of the device. Hence, the best active power savings are obtained by combining the DVFS, DPS, SLM, and AVS techniques. For a given operating state, one or more of the power-saving techniques can be applied to ensure optimal operation with maximum power saving.

AVS must be used at boot time to adapt the voltage to the process characteristics (strong/weak) of the device. AVS can also ensure the maximum available application performance of the device at a given OPP.

When medium application performance is required, or when application performance requirements vary, DVFS can be applied. The voltage and frequency can be scaled to match the closest OPP that meets the performance requirement.

When application performance requirements fall between two OPPs, or when a low application performance is required that is below the lowest performance OPP, DPS can be applied to switch to low-power mode.

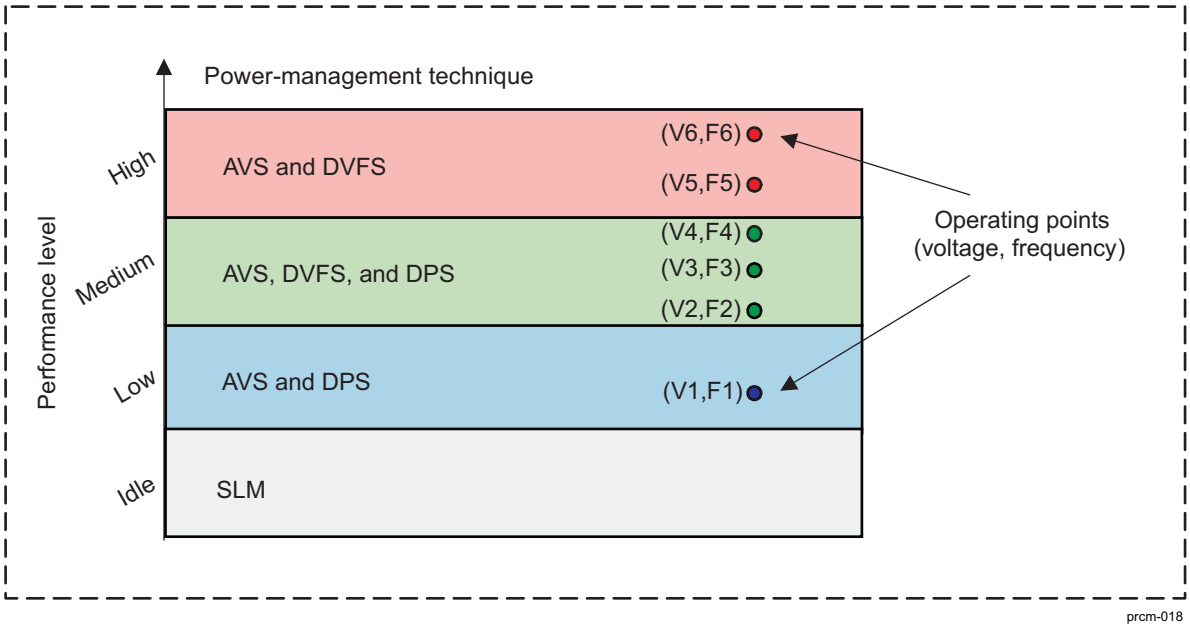
When combining DVFS and DPS, the operating frequency must not be scaled to match the performance requirement without scaling the voltage. Lower operating frequency increases task completion time and reduces idle time. This prevents DPS or reduces its efficiency (DPS becomes more effective as idle time increases). Unless DPS cannot be applied for other reasons, for a given operating point of DVFS the operating frequency must always be set to the maximum allowed at a given voltage. This ensures optimal process completion time and application of DPS.

If DPS cannot be applied in a given context, scaling the frequency while keeping the voltage constant does not save energy; it does, however, reduce peak power consumption. This can have a positive effect on temperature dissipation and battery life.

SLM must be used when no applications are running and performance requirement drops to 0.

NOTE: The OPPs shown in Figure 3-18 are only for indication and clarity of text. They do not correspond to validated OPPs of the device.

Figure 3-18. Performance Level and Applied Power-Management Techniques



3.1.2.5.1 DPS Versus SLM

DPS and SLM are similar concepts: they consist of switching the system between high- and low-power consumption modes. However, their operating timescales differ, principally in the latency allowed for mode transitions.

DPS is generally used in an applicative context (tasks are started). Therefore, mode transitions are related to system performance requirements or the processor load. DPS transition latencies must be small (typically between 10 and 100µs) compared to the time constraints or deadlines of the application so that they do not degrade application performance. DPS requires performance prediction to ensure that transition latencies do not deteriorate device performance to the point that real-time application deadlines are missed or the user experience degrades too much for an interactive application.

SLM is not used in an applicative context (no task started). Mode transitions are related more to system responsiveness, and the transition latencies must be small compared to user sensitivity so that they do not degrade the user experience. For SLM, transition latencies are typically 1-10 ms or more.

DPS and SLM also differ in the type of wake-up event used to exit low-power idle mode. For DPS, wake-up events are application-related (timer, DMA request, peripheral interrupt, etc.); for SLM, wake-up events are user-related (touch screen, key pressed, peripheral connections, etc.).

3.2 PRCM Subsystem Overview

3.2.1 Introduction

The power-management framework of the device significantly reduces dynamic power consumption and static leakage current. This framework incorporates support for state-of-the-art power-management techniques. It ensures optimal device operation with significantly reduced power consumption. The power-management framework (PMFW) of the device is handled by the PRCM which is a logical module composed of the following three physical submodules:

- PRM: Handles device-level power and reset management. It also handles some clocks in the device. This module always remains on, unless no power is supplied to the device pads.
- CM_CORE_AON: Handles device-level clock management of the DSP1, DSP2, COREAON, CORE, IPU, L4PER, DSS, L3INIT, ISS, CAM, CUSTEFUSE, EVE1, EMU and WKUPAON power domains. This module always remains on, unless no power is supplied to the device pads.
- CM_CORE: Handles device-level clock management of the MMR block.

The device supports the power-management techniques with the following features:

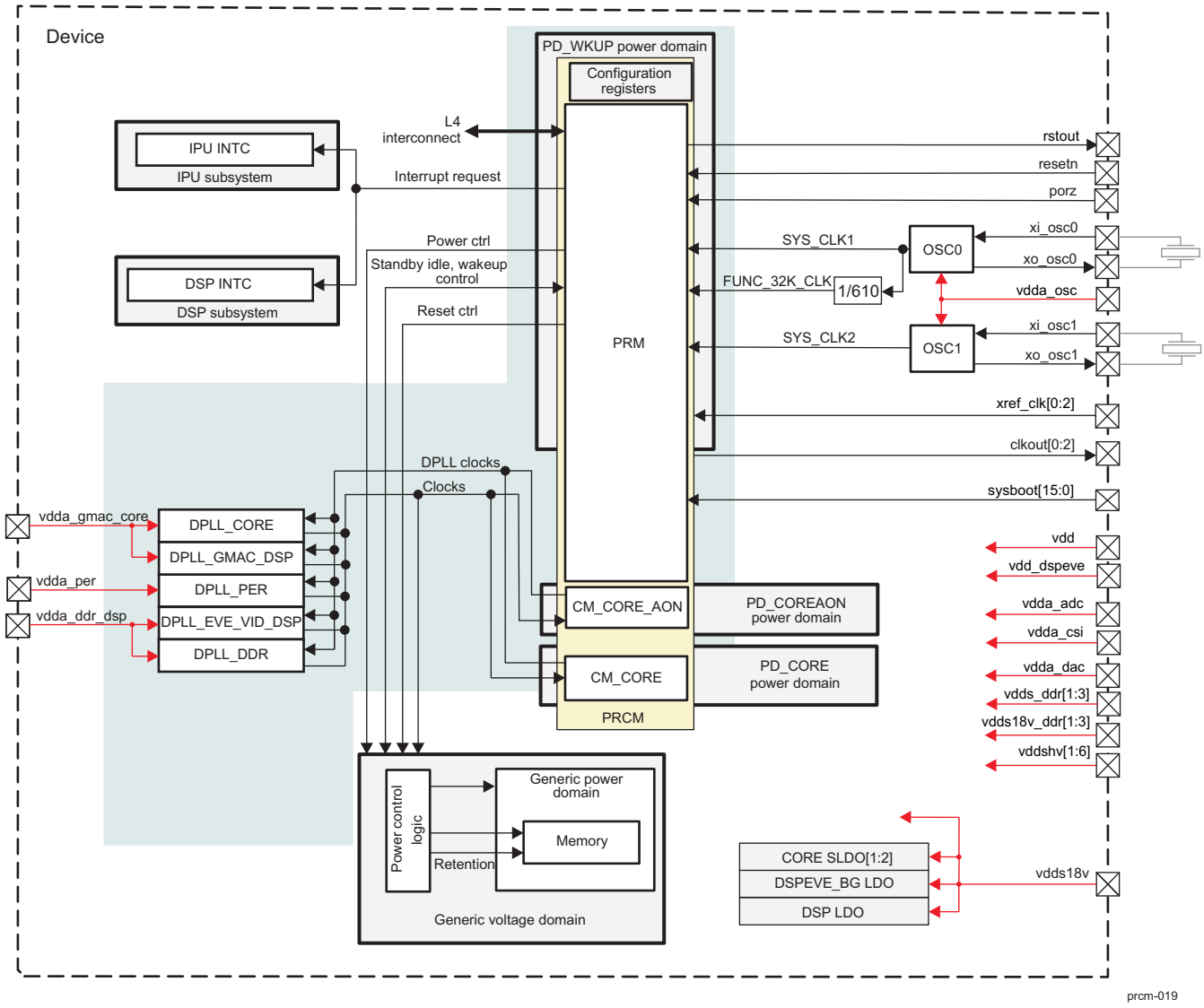
- Partitioning of the device into voltage, power, clock, and reset domains
- Domain isolation that allows any combination of domain ON/OFF states
- Clock tree with selective clock-gating conditions
- Hardware-controlled reset sequencing management
- Power, reset, and clock control hardware mechanism to manage sleep and wake-up dependencies of the power domains
- Support for hardware-controlled autogating of module clocks
- Memory retention capability for preserving memory contents in low-power sleep mode
- DVFS support for the processor and peripherals

The PMFW interfaces with all the components of the device for power, clock, and reset management through associated control signals. It integrates enhanced features to let the device adapt energy consumption dynamically according to changing application and performance requirements. The innovative hardware architecture allows a substantial reduction in leakage current.

The power-management modules are fully configurable through their L4 interface ports.

Figure 3-19 is an overview of the power-management modules and their internal connections with a generic power domain.

Figure 3-19. PMFW Overview



NOTE: The device I/O logic maps the module signals to the different pads of the device. For more information, see the *Pad Configuration Registers* and sections in Chapter 13, *Control Module* and the device Data Manual.

3.2.2 Power-Management Framework Features

The power-management modules:

- Manage independent power domains
- Control scalable logic voltage domains and selectable voltage modes for memory voltage domains
- Handle standby, idle, and wake-up procedures for the modules of the device
- Allow software and partial hardware control
- Monitor and handle wake-up events
- Control system clock and reset input sources
- Manage and distribute clocks and resets with high control granularity
- Handle power-up sequences

- Have debug and emulation features
- Control RFFs of device modules to support DPS

3.3 PRCM Subsystem Environment

The modules of the PMFW receive the external reset, clock, and power signals. See Figure 3-19.

The following sections describe the interfaces for external clock, reset, and power sources.

3.3.1 External Clock Signals

Table 3-23 lists the external clock pins, signal names, their direction, associated module, and description of the signals. If the signal is input to the device, the module is the destination module for the signal; if the signal is an output from the device, the module is the source module of the signal.

Table 3-23. External Clock Signals

Pin	Signal	I/O[1]	PMFW Module	Description
xi_osc0	XI_OSC0	I	OSC0	System Oscillator OSC0 Crystal Input. This is the main system clock of the device.
xo_osc0	XO_OSC0	O	OSC0	System Oscillator OSC0 Crystal drive.
xi_osc1	XI_OSC1	I	OSC1	Auxiliary Oscillator OSC1 Crystal input. This is optional clock input to the device.
xo_osc1	XO_OSC1	O	OSC1	Auxiliary Oscillator OSC1 Crystal drive.
clkout0	CLKOUTMUX0_CLK	O	CM_COREAON	Device Clock output 0. Can be used as a system clock for external devices.
clkout1	CLKOUTMUX1_CLK	O	CM_COREAON	Device Clock output 1. Can be used as a system clock for external devices.
clkout2	CLKOUTMUX2_CLK	O	CM_COREAON	Device Clock output 2. Can be used as a system clock for external devices.
xref_clk0	XREF_CLK0	I	CM_COREAON	External Reference Clock 0. For audio and other peripherals.
xref_clk1	XREF_CLK1	I	CM_COREAON	External Reference Clock 1. For audio and other peripherals.
xref_clk2	XREF_CLK2	I	CM_COREAON	External Reference Clock 2. For McASP modules.

(1) I = Input; O = Output; I/O = Bidirectional

3.3.2 External Boot Signals

The PRM receives SYSBOOT[15:0] information from external pins sys_boot[15:0]. They are used to configure the boot mode of the device. For more information, see Chapter 25, *Initialization*, Section 25.2.4, *Sysboot Configuration* and the device *Data Manual*.

Table 3-24 lists the external boot configuration pins, signal names, their direction, the associated modules, description, and reset values of the signals. If the signal is input to the device, the module is the destination module for the signal; if the signal is an output from the device, the module is the source module of the signal.

Table 3-24. External Boot Signals

Pin	Signal	I/O [1]	PMWF Module	Description	Pin Reset Value
sys_boot[15:0]	SYSBOOT[15:0]	I	PRM	See Section 25.2.4, *Sysboot Configuration*	Z

(1) I = Input; O = Output; I/O = bidirectional

3.3.3 External Reset Signals

The PRM drives the SYS_WARM_OUT_RST reset output signal which is directly mapped as rstoutn device pin.The PRM asserts SYS_WARM_OUT_RST output upon the assertion of any source of global reset (warm or cold). This reset signal keeps external peripherals under reset while the device is in global reset.

Table 3-25 lists the external reset pins, signal names, their direction, associated module, and description of the signals. If the signal is input to the device, the module is the destination module for the signal; if the signal is an output from the device, the module is the source module of the signal.

Table 3-25. External Reset Signals

Pin	Signal	I/O[1]	PMFW Module	Description
resetn	SYS_WARM_IN_RST	I	PRM	Warm Reset Input
rstoutn	SYS_WARM_OUT_RST	O	PRM	Warm Reset to external devices
porz	SYS_PWRON_RST	I	PRM	Power-on Reset input typically from PMIC

(1) I = Input; O = Output; I/O = bidirectional

3.3.4 External Voltage Inputs

Table 3-26 lists the external voltage sources related to the PRCM module.

NOTE: Table 3-26 lists only the voltage sources that are directly managed by the PRCM module or received by parts of the PRCM module (for example, DPLLs, LDOs, etc.). It does not give the device voltage sources not directly associated with the PRCM module. Please refer to the device *Data Manual* for all power pins.

Table 3-26. Voltage Sources

Pin	Net	Managed by the PRCM?	Description
vdd	VDD_CORE_L	Yes (AVS and DVFS)	Subsystems and modules supplied by CORE voltage domain
vdd_dspeve	VDD_DSPEVE_L	Yes (AVS)	Supplies DSP and EVE voltage domain
vdda_osc	VDDS5	No (fixed)	I/O supply for oscillator section
vdda_gmac_core	VDDA_DPLL_GMAC	No (fixed)	DPLL_GMAC_DSP and HSDIVIDER analog power supply
	VDDA_DPLL_CORE	No (fixed)	DPLL_CORE and HSDIVIDER analog power supply
vdda_per	VDDA_DPLL_PER	No (fixed)	DPLL_PER and PER HSDIVIDER analog power supply
vdda_ddr_dsp	VDDA_DPLL_DSP	No (fixed)	DPLL_EVE_VID_DSP analog power supply
	VDDA_DPLL_DDR	No (fixed)	DPLL_DDR and DDR HSDIVIDER analog power supply
vdds18v	VDDS_LDO_BG_BBLDO_DSPEVE	No	Supplies LDO_DSPEVE for DSPEVE SRAM
	VDDA_LDO_DSPEVE	No	DSP LDO's 1.8V power supply
	VDDA_LDO_CORE[1:2]	No	VDDS supply for core SLDO

3.4 PRCM Subsystem Integration

The internal configuration registers of the power-management modules can be accessed for configuration and control through their respective L4_WKUP interconnect. In addition to the L4 interconnect, the internal module interface contains the following interfaces and signals:

- A set of signals for idle/wake-up control for each module
- Clocks and reset signals for the modules
- Power control signals (switches and memories) to the power domains
- Interrupts to the IPU and DSP INTCs
- Phase-locked loop (PLL) control commands for recalibration and bypass of the digital phase-locked loops (DPLLs)

Figure 3-19 shows details of the control interface to a generic power domain.

3.4.1 Device Power-Management Layout

The PMFW sees the device split into voltage domains, power domains, and clock domains. Table 3-27 provides the device-level view with module association to the clock, power, and voltage domains.

Table 3-27. PMFW Device-Level Layout

Voltage Domain	Physical Power Domain	Clock Domain	Module
VD_CORE	PD_WKUPAON	CD_WKUPAON	CTRL_MODULE_WKUP
			L4_WKUP interconnect
			GPIO1
			TIMER1
			COUNTER_32K
			DCAN1
			RTI1
			RTI2
			RTI3
			RTI4
			RTI5
		N/A (the PRCM module)	PRM
	PD_COREAON	CD_COREAON	CM_CORE_AON
			DPLL_CORE
			DPLL_PER
			DPLL_GMAC_DSP
			DPLL_DDR
			WUGEN_IPU
		CD_EMU	DEBUGSS
		CD_L4_PER1	DCC5 (1)
			DCC6 (1)
			DCC7 (1)
			TIMER2
			TIMER3
			TIMER4
			ESM (2)
			ELM
			GPIO2
			GPIO3

(1) DCC modules are not supported on the DRA78x family of devices.
(2) ESM is not supported on on the DRA78x family of devices.

Table 3-27. PMFW Device-Level Layout (continued)

Voltage Domain	Physical Power Domain	Clock Domain	Module
			GPIO4
			I2C1
			I2C2
			L4_PER1 interconnect
			McSPI1
			McSPI2
			McSPI3
			McSPI4
			MMC
			UART1
			UART2
			UART3
		CD_L4PER2	L4_PER2 interconnect
			MCAN
			QSPI
			PWMSS
			ADC
			ATL
			McASP2
			McASP3
		CD_L4PER3	L4_PER3 interconnect
			DCC1[(1)]
			DCC2[(1)]
			DCC3[(1)]
			DCC4[(1)]
		CD_L4_CFG	CTRL_MODULE_CORE
			SPINLOCK
			L4_CFG interconnect
			MAILBOX1
			MAILBOX2
		CD_EMIF	EMIF
			DLL
			EMIF_PHY
		CD_L3_MAIN1	L3_MAIN_1 interconnect
			GPMC
			OCMC_RAM
			MMU1
			MMU2
			EDMA
			TESOC[(3)]
		CD_CRC	CRC
		CD_L3_INSTR	OCP_WP_NOC
			L3_INSTR interconnect
			L3_MAIN_2 interconnect
			CTRL_MODULE_BANDGAP
			DLL_AGING

(3) TESOC is not supported on on the DRA78x family of devices.

Table 3-27. PMFW Device-Level Layout (continued)

Voltage Domain	Physical Power Domain	Clock Domain	Module
		CD_L3INIT	IEEE1500_2_OCP
		CD_IPU	McASP1
			TIMER5
			TIMER6
			TIMER7
			TIMER8
		CD_MMAON	DPLL_EVE_VIP_DSP
		CD_GMAC	GMAC
	PD_CAM	CD_CAM	VIP
	PD_IPU	CD_IPU1	IPU1
	PD_DSS	CD_DSS	DSS, SD_DAC (controlled by DSS)
	PD_CUSTEFUSE	CD_CUSTEFUSE	EFUSE_CTRL_CUST
	PD_ISS	CD_ISS	ISS, CAMERARX[4]
VD_DSPEVE	PD_DSP1	CD_DSP1	DSP1
	PD_DSP2	CD_DSP2	DSP2
	PD_EVE	CD_EVE	EVE

(4) ISS and CAMERARX are not supported on the DRA78x family of devices.

3.4.2 Power-Management Scheme, Reset, and Interrupt Requests

3.4.2.1 Power Domain

Table 3-28 lists the PMFW modules and their associated power domains.

Table 3-28. PMFW Module Power Domains

PMFW Module	Power Domain
PRM	PD_WKUPAON
CM_CORE_AON	PD_COREAON
CM_CORE	PD_CORE

The PRM part of the PRCM module is in the PD_WKUPAON power domain, which is continuously active. It is composed of the logic that must be permanently supplied to manage domain power-state transitions and detect wake-up events.

The CM_CORE_AON part of the PRCM module is in the PD_COREAON power domain, which is an always-on power domain, while the CM_CORE part of the PRCM module is in the PD_CORE power domain, which can be activated and deactivated according to the requirements of the executing applications.

3.4.2.2 Resets

The PMFW modules are reset by independent reset signals (see Table 3-29).

Table 3-29. PMFW Module Reset Signals

PMFW Module	Reset Signal
PRM	SYS_PWRON_RST_IN
CM_CORE_AON	CM_CORE_AON_PWRON_RST
	CM_CORE_AON_RST
CM_CORE	CM_CORE_PWRON_RET_RST

Table 3-29. PMFW Module Reset Signals (continued)

PMFW Module	Reset Signal
	CM_CORE_RET_RST

NOTE: For more information about the reset trigger sources and assertion conditions, see Section 3.5.3, *Reset Sources*.

3.4.2.3 PRCM Interrupt Requests

The PMFW modules can generate the interrupts listed in Table 3-30.

Table 3-30. PMFW Hardware Requests

Interrupt Requests				
Module Instance	**Source Signal Name**	**IRQ_CROSSBAR Input**	**Default Mapping**	**Description**
PRM	PRM_IRQ_MPU	IRQ_CROSSBAR_6	DSP1_IRQ_37 DSP2_IRQ_37 EVE1_IRQ_5	PRM MPU interrupt request
	PRM_IRQ_IPU1	IRQ_CROSSBAR_133	IPU1_IRQ_47	PRM IPU1 interrupt request
	PRM_IRQ_DSP1	IRQ_CROSSBAR_137	-	PRM DSP1 interrupt request
	PRM_IRQ_DSP2	IRQ_CROSSBAR_387	-	PRM DSP2 interrupt request
	PRM_IRQ_EVE1	IRQ_CROSSBAR_388	-	PRM EVE1 interrupt request

NOTE: The **Default Mapping** column in Table 3-30 *PMFW Hardware Requests* shows the default mapping of module IRQ source signals. These module IRQ source signals can also be mapped to other lines of each device Interrupt controller through the IRQ_CROSSBAR module, respectively.
For more information about the IRQ_CROSSBAR module, see Section 13.4.6.4, *IRQ_CROSSBAR Module Functional Description*, in Chapter 13, *Control Module*.
For more information about the device interrupt controllers, see Chapter 12, *Interrupt Controllers*.

3.5 Reset Management Functional Description

3.5.1 Overview

In the device, the reset scheme is managed by the PRM, device-level reset manager.

3.5.1.1 Reset Management Functional Description

The PRM handles the device power-on and warm reset pads: porz, resetn, and rstoutn. The PRM:

- Extends the reset duration beyond the porz pad release
- Provides the reset duration period after the resetn pad assertion
- Routes the device pad resets

The PRM is functionally sensitive to the device POR.

3.5.1.1.1 Power-On Reset

The PRM receives the device POR on the porz reset pad of the device. It extends the duration of the POR module until at least stable 32-kHz clock and SYS_CLK1 are provided to it. The 32-kHz clock and SYS_CLK1 of the device are propagated once the PRCM releases the internal POR.

3.5.1.1.2 Warm Reset

The device warm reset can be received from an external source or the device PRM module as a result of an internal event.

The PRCM can generate the global reset used by the domain Reset Managers from these warm reset input sources:

- SYS_WARM_IN_RST
- GLOBAL_SW_WARM_RST

3.5.1.2 PRM Reset Management Functional Description

The PRM module manages the resets to all power domains inside the device.

NOTE: The PRM module has no knowledge or control over resets generated locally within a module (for example, through the <Module name>_SYSCONFIG[x] SOFTRESET configuration register bit). A software reset has the same effect on the module logic as a hardware reset.

All PRM reset outputs are asynchronously asserted, while the deassertion is synchronous to the SYS_CLK1 clock. The reset managers in PRM use this clock to stall, or delay, deassertion of reset upon source deactivation.

In each power domain one or more reset domains are defined by a unique reset signal that originates from the reset manager and is connected to one or more modules of the device. All the connected modules of the reset domain are reset simultaneously when the reset signal is asserted. Independent control of these reset domains allows sequencing of the release of resets and ensures a safe reset of the entire power domain.

3.5.2 General Characteristics of Reset Signals

Reset signals can be categorized based on four criteria:

- Scope: Global or local reset
- Occurrence: Cold or warm reset
- Source type: Software-controlled or hardware-triggered reset
- Retention type: Retention reset or nonretention reset

3.5.2.1 Scope

A reset signal can be categorized according to its scope (the area of the device affected by the reset):

- Global reset: Affects the entire logic of the device; all modules are reset. Generally, occurs when the device powers up (POR) or an abnormal operation is detected (watchdog timeout, thermal shutdown, etc.)
- Local reset: Affects one power domain, reset domain, or module.

3.5.2.2 Occurrence

A reset signal can be categorized depending on when the reset occurs:

- Cold reset: Occurs on device power up (POR), or in certain emulation modes. Also, it can be software-initiated. Upon cold reset, it is assumed the device is being powered up, and therefore, everything in the device is being reset. That is, cold resets must be considered as global resets.
- Warm reset: Warm reset types are not necessarily applied globally within device. Also, a module can use a warm reset to reset a subset of its logic. This is often done to speed up reset recovery time; that is, the time to transition to a safe operating state, compared to the time required upon receipt of a cold reset. Warm reset events include software-triggered reset per power domain, watchdog time-out, externally triggered and emulation initiated.

Modules that behave differently in cold reset and warm reset have two reset signals: RST and PWRON_RST. These reset signals reconstruct warm reset and cold reset in modules that require them.

The following modules are reset upon global cold reset events and not upon global warm reset events:

- All DPLLs associated with the PRCM module
- EMIF
- Control module: CTRL_MODULE_CORE, CTRL_MODULE_WKUP

NOTE: For information about the PRCM module registers affected by the global warm reset, see the register description in Section 3.12, *PRCM Register Manual*.

NOTE: The DPLLs state will change after warm reset even though their registers are not warm reset-sensitive, refer to Section 3.5.6.8,*Global Warm Reset Sequence* for more details.

3.5.2.3 Source Type

A reset can be categorized depending on whether it is software-controlled or hardware-triggered:

- Software reset: Triggered by setting a bit in a configuration register of the PRCM module
- Hardware reset: Triggered by a signal from a hardware module inside or outside the PRCM module

3.5.2.4 Retention Type

The power domain manager in PRM controls the assertion of two types of local cold reset sources, retention and nonretention reset. They are identified by the naming convention <PowerDomain>_DOM_RET_RST and <PowerDomain>_DOM_RST, respectively.

Upon transitioning a power domain to the ON-ACTIVE power state from the OFF power state:

- Retention type reset source is always asserted.
- Nonretention type reset source is always asserted.

Upon transitioning a power domain to the ON-ACTIVE power state from the RETENTION power state:

- Retention type reset source is never asserted.
- Nonretention type reset source is optionally asserted. The software-selectable option enables the PRM to support two retention mechanisms: CSWRET. Nonretention type reset source has the following behavior:
  - It is not asserted if the power domain state transitions from the CSWR-RETENTION state.

3.5.3 Reset Sources

The reset sources triggering the reset managers in the PRM are described in this section.

3.5.3.1 Global Reset Sources

Table 3-31 lists the global reset sources of the device. The global reset source signals received by the reset manager trigger the reset of all the device modules. For all hardware reset signals, the source of the reset is identified; for the software reset signals, the bit triggering the reset is identified.

Table 3-31. Global Reset Sources

Type[1]	Name	Source/Control	Description
H/C	SYS_PWRON_RST	porz input pin	The entire device is held in reset during porz pin assertion. porz pin is typically asserted by external PMIC at POR. PMIC keeps porz asserted until all voltage rails are ramped-up and stable. PRM extends device internal reset after porz release. Refer to Power Supply Sequences in Device Data Manual.
H/W	SYS_WARMIN_RST	resetn input pin	Warm reset from external device. Internal warm reset is triggered by the active (falling) edge of this signal and is extended with RstTime1.
S/C	GLOBAL_COLD_SW_RST	PRM_RSTCTRL[1] RST_GLOBAL_COLD_SW	Global software cold reset
S/W	GLOBAL_WARM_SW_RST	PRM_RSTCTRL[0] RST_GLOBAL_WARM_SW	Global software warm reset
H/W	TSHUT_CORE_RST	Device thermal sensor, placed close to the ISS[2] and DSP	Asserted when measured temperature is greater than shutdown temperature threshold
H/W	ICEPICK_RST	ICEPick™ module	ICEPick warm reset. It is used only in emulation mode.
H/C	ICEPICK_POR_RST	ICEPick module	ICEPick cold reset. It is used only in emulation mode.
H/W	RTI_DWD_RST	RTI digital watchdog reset	Can be selected among RTI[1:5] in Control Module register CTRL_CORE_CONTROL_IO_2

(1) H = Hardware reset, S = Software reset, C = Cold reset, W = Warm reset
(2) ISS is not supported on the DRA78x family of devices.

3.5.3.2 Local Reset Sources

In addition to the global reset sources the device can have a number of local reset sources for each power domain. The local reset sources can be cold or warm reset sources. They can be software-controlled or hardware-triggered. Table 3-32 identifies the possible types of hardware-triggered local cold reset sources. Some power domains can support one or both of these local cold reset sources. The table does not list the software-triggered local warm reset sources that are listed in the reset management section of the respective power domains. A local reset source signal received by the reset manager resets only a specific part of the device (for example, some modules/subsystems within the power domain).

Table 3-32. Local Reset Sources

Type[1]	Name	Source/Control	Description
H/C	<Power domain>_RET_RST	PRCM	Asserted only for a power domain state transition from OFF to ON-ACTIVE state
H/C	<Power domain>_RST	PRCM	Asserted for any power domain transition to ON-ACTIVE state

(1) H = Hardware reset, C = Cold reset

3.5.4 Reset Logging

A reset of the device is logged in the reset status registers RM_<power domain>_RSTST and in global register PRM_RSTST.

RM_<power domain>_RSTST are reset asynchronously on assertion of a global cold reset. However, a reset status bit is always logged when the reset is released to the domain.

For this reason, after the assertion of a global cold reset, the reset status register is cleared to 0. When the domain reset is released, the register bit to log the global cold reset (the PRM_RSTST[0] GLOBAL_COLD_RST bit) is updated to 1. For the same reason, the reset status register of domains released from reset by software is updated only when software releases the domain reset.

The assertion of a global cold reset prevents logging any other source of reset until after the release of the domain reset. This is valid in the following situations:

- A source of reset other than global cold reset is asserted before, during, or after the active period of a global cold source of reset and before the release of the domain reset signal.
- A source of reset other than global cold reset is asserted and then released, but a global cold reset source is asserted before the release of the domain reset signal.

3.5.5 Reset Domains

A power domain can receive power-on reset (PWRON_RST) and/or normal reset (RST) signals. These signals reset nonretention logic and behave as follows:

- On any global or local cold reset, RST and PWRON_RST are asserted.
- On any global or local warm reset, only RST is asserted.
- On wakeup from OFF, PWRON_RST are asserted.

A power domain can receive two additional retention logic reset signals: power-on retention reset (PWRON_RET_RST) and/or retention reset (RET_RST). These signals behave as follows:

- On any global cold reset, RET_RST and PWRON_RET_RST are asserted.
- On any global cold reset or wakeup from OFF state to ON-ACTIVE state, RET_RST and PWRON_RET_RST are asserted.
- On any global warm reset, only RET_RST is asserted.

This section discusses the trigger sources and attributes for all reset domains of the device. For an explanation of each reset trigger source of the device, see Section 3.5.3, *Reset Sources.*

Table 3-33 identifies the associated power and rest domains for each module.

NOTE: The DPLLs state will change after warm reset even though they are not warm reset-sensitive, refer to Section 3.5.6.8,*Global Warm Reset Sequence* for more details.

Table 3-33. Modules, Power Domains, and Reset Domains Association

Module	Power Domain	Reset Domains
CM_CORE_AON	PD_COREAON	CM_CORE_AON_PWRON_RST, CM_CORE_AON_RST
DPLL_CORE	PD_COREAON	COREAON_PWRON_RST
DPLL_PER	PD_COREAON	COREAON_PWRON_RST
DPLL_PCIE_REF	PD_COREAON	COREAON_PWRON_RST
DPLL_GMAC_DSP	PD_COREAON	COREAON_PWRON_RST
DPLL_DDR	PD_COREAON	COREAON_PWRON_RST
WUGEN_IPU	PD_COREAON	None
SPINNER	PD_COREAON	None
DPLL_EVE_VID_DSP	PD_COREAON	DPLL_EVE_VID_DSP_PWRON_RST
McASP1	PD_COREAON	IPU_RST
TIMER5	PD_COREAON	IPU_RST

Table 3-33. Modules, Power Domains, and Reset Domains Association (continued)

Module	Power Domain	Reset Domains
TIMER6	PD_COREAON	IPU_RST
TIMER7	PD_COREAON	IPU_RST
TIMER8	PD_COREAON	IPU_RST
IPU	PD_IPU	IPU_PWRON_RST, IPU_RET_RST, IPU_CPU0_RST, IPU_CPU1_RST, IPU_RST
VIP	PD_CAM	CAM_RST
EFUSE_CTRL_CUST	PD_CUSTEFUSE	CUSTEFUSE_RST
CM_CORE	PD_CORE	CM_CORE_PWRON_RET_RST, CM_CORE_RET_RST
CTRL_MODULE_CORE	PD_COREAON	CORE_PWRON_RET_RST
CTRL_MODULE_BANDGAP	PD_COREAON	CORE_PWRON_RET_RST
EMIFPHY	PD_CORE	EMIF_DDR_PHY_PWRON_RST
DLL	PD_CORE	DLL_RST
DLL_AGING	PD_CORE	CORE_RST
EMIF	PD_COREAON	CORE_PWRON_RET_RST, CORE_PWRON_RST
GPMC	PD_COREAON	CORE_RET_RST
SPINLOCK	PD_COREAON	CORE_RET_RST
L3_MAIN_2 interconnect	PD_COREAON	CORE_PWRON_RET_RST, CORE_RST
L3_MAIN_1 interconnect	PD_COREAON	CORE_PWRON_RET_RST, CORE_RST
L3_INSTR interconnect	PD_COREAON	CORE_RST
OCP_WP_NOC	PD_COREAON	CORE_PWRON_RET_RST, CORE_RST
L4_CFG interconnect	PD_COREAON	CORE_PWRON_RET_RST, CORE_RST
MAILBOX1	PD_COREAON	CORE_RET_RST
MAILBOX2	PD_COREAON	CORE_RET_RST
MMU	PD_COREAON	CORE_RET_RST
EDMA_TPCC	PD_COREAON	CORE_RET_RST
EDMA_TC0	PD_COREAON	CORE_RET_RST
EDMA_TC1	PD_COREAON	CORE_RET_RST
OCMC_RAM	PD_COREAON	CORE_RST
TESOC[1]	PD_COREAON	CORE_RST
CRC	PD_COREAON	CORE_RST
DSS	PD_DSS	DSS_RET_RST, DSS_RST
SD_DAC	PD_DSS	DSS_RST
ISS[2]	PD_ISS	ISS_RST
CM_EMU	PD_COREAON	EMU_PWRON_RST
DEBUGSS	PD_COREAON	EMU_EARLY_PWRON_RST, EMU_PWRON_RST, EMU_RST
IEEE1500_2_OCP	PD_COREAON	L3INIT_RST
GMAC	PD_COREAON	L3INIT_RST
DCC1[3]	PD_COREAON	L4PER_RST
DCC2[3]	PD_COREAON	L4PER_RST
DCC3[3]	PD_COREAON	L4PER_RST
DCC4[3]	PD_COREAON	L4PER_RST
TIMER2	PD_COREAON	L4PER_RST
TIMER3	PD_COREAON	L4PER_RST

(1) TESOC is not supported on the DRA78x family of devices.
(2) ISS is not supported on the DRA78x family of devices.
(3) DCC modules are not supported on the DRA78x family of devices.

Table 3-33. Modules, Power Domains, and Reset Domains Association (continued)

Module	Power Domain	Reset Domains
TIMER4	PD_COREAON	L4PER_RST
ELM	PD_COREAON	L4PER_RST
GPIO2	PD_COREAON	L4PER_RET_RST
GPIO3	PD_COREAON	L4PER_RET_RST
GPIO4	PD_COREAON	L4PER_RET_RST
ESM(4)	PD_COREAON	L4PER_RST
I2C1	PD_COREAON	L4PER_RET_RST
I2C2	PD_COREAON	L4PER_RST
L4_PER1 interconnect	PD_COREAON	L4PER_PWRON_RET_RST, L4_PER_RST
L4_PER2 interconnect	PD_COREAON	L4PER_PWRON_RET_RST, L4_PER_RST
L4_PER3 interconnect	PD_COREAON	L4PER_PWRON_RET_RST, L4_PER_RST
ADC	PD_COREAON	L4PER_RST
ATL	PD_COREAON	L4PER_RST
McASP2	PD_COREAON	L4PER_RST
McASP3	PD_COREAON	L4PER_RST
McSPI1	PD_COREAON	L4PER_RST
McSPI2	PD_COREAON	L4PER_RST
McSPI3	PD_COREAON	L4PER_RST
McSPI4	PD_COREAON	L4PER_RST
MMC	PD_COREAON	L4PER_RST
MCAN	PD_COREAON	L4PER_RST
UART1	PD_COREAON	L4PER_RET_RST
UART2	PD_COREAON	L4PER_RET_RST
UART3	PD_COREAON	L4PER_RET_RST
QSPI	PD_COREAON	L4PER_RST
PWMSS	PD_COREAON	L4PER_RST
DCC5(3)	PD_COREAON	L4PER_RST
DCC6(3)	PD_COREAON	L4PER_RST
DCC7(3)	PD_COREAON	L4PER_RST
DSP1	PD_DSP1	DSP1_RST, DSP1_PWRON_RST, DSP1_RET_RST, DSP1_SYS_RST
DSP2	PD_DSP2	DSP2_RST, DSP2_PWRON_RST, DSP2_RET_RST, DSP2_SYS_RST
CTRL_MODULE_WKUP	PD_WKUPAON	WKUPAON_PWRON_RST
PRM	PD_WKUPAON	PRM_PWRON_RST, PRM_RST
GPIO1	PD_WKUPAON	WKUPAON_RST
COUNTER_32K	PD_WKUPAON	WKUPAON_RST, WKUPAON_SYS_PWRON_RST
TIMER1	PD_WKUPAON	WKUPAON_RST
L4_WKUP interconnect	PD_WKUPAON	WKUPAON_RST
DCAN1	PD_WKUPAON	WKUPAON_RST
RTI1	PD_WKUPAON	WKUPAON_RST
RTI2	PD_WKUPAON	WKUPAON_RST
RTI3	PD_WKUPAON	WKUPAON_RST
RTI4	PD_WKUPAON	WKUPAON_RST

(4) ESM is not supported on the DRA78x family of devices.

Table 3-33. Modules, Power Domains, and Reset Domains Association (continued)

Module	Power Domain	Reset Domains
RTI5	PD_WKUPAON	WKUPAON_RST
EVE	PD_EVE	EVE_CPU_RST, EVE_PWRON_RST, EVE_RST

Table 3-34 lists the reset sources that trigger the reset domains of the device.

Table 3-34. Reset Sources for the Reset Domains

Reset Domain	Reset Source	Reset Source Type
CM_CORE_AON_PWRON_RST	GLOBAL_COLD_SW_RST	Global cold
	ICEPICK_POR_RST	Global cold
	SYS_PWRON_RST	Global cold
CM_CORE_AON_RST	GLOBAL_COLD_SW_RST	Global cold
	GLOBAL_WARM_SW_RST	Global warm
	ICEPICK_RST	Global warm
	ICEPICK_POR_RST	Global cold
	SYS_PWRON_RST	Global cold
	SYS_WARMIN_RST	Global warm
	TSHUT_CORE_RST	Global warm
COREAON_PWRON_RST	GLOBAL_COLD_SW_RST	Global cold
	ICEPICK_POR_RST	Global cold
	SYS_PWRON_RST	Global cold
COREAON_RST	GLOBAL_COLD_SW_RST	Global cold
	GLOBAL_WARM_SW_RST	Global warm
	ICEPICK_POR_RST	Global cold
	SYS_PWRON_RST	Global cold
	SYS_WARMIN_RST	Global warm
	TSHUT_CORE_RST	Global warm
	ICEPICK_RST	Global warm
DPLL_EVE_VID_DSP_PWRON_RST	GLOBAL_COLD_SW_RST	Global cold
	ICEPICK_POR_RST	Global cold
	SYS_PWRON_RST	Global cold
MMAON_RST	GLOBAL_COLD_SW_RST	Global cold
	GLOBAL_WARM_SW_RST	Global warm
	ICEPICK_POR_RST	Global cold
	SYS_PWRON_RST	Global cold
	SYS_WARMIN_RST	Global warm
	TSHUT_CORE_RST	Global warm
	ICEPICK_RST	Global warm
CAM_RST	GLOBAL_COLD_SW_RST	Global cold
	GLOBAL_WARM_SW_RST	Global warm
	ICEPICK_POR_RST	Global cold
	SYS_PWRON_RST	Global cold
	SYS_WARMIN_RST	Global warm
	TSHUT_CORE_RST	Global warm
	ICEPICK_RST	Global warm
CM_CORE_PWRON_RET_RST	GLOBAL_COLD_SW_RST	Global cold
	ICEPICK_POR_RST	Global cold
	SYS_PWRON_RST	Global cold

Table 3-34. Reset Sources for the Reset Domains (continued)

Reset Domain	Reset Source	Reset Source Type
CM_CORE_RET_RST	GLOBAL_COLD_SW_RST	Global cold
	GLOBAL_WARM_SW_RST	Global warm
	ICEPICK_POR_RST	Global cold
	SYS_PWRON_RST	Global cold
	SYS_WARMIN_RST	Global warm
	TSHUT_CORE_RST	Global warm
	ICEPICK_RST	Global warm
CORE_PWRON_RET_RST	GLOBAL_COLD_SW_RST	Global cold
	SYS_PWRON_RST	Global cold
	ICEPICK_POR_RST	Global cold
CORE_PWRON_RST	GLOBAL_COLD_SW_RST	Global cold
	SYS_PWRON_RST	Global cold
	ICEPICK_POR_RST	Global cold
CORE_RET_RST	GLOBAL_COLD_SW_RST	Global cold
	GLOBAL_WARM_SW_RST	Global warm
	ICEPICK_POR_RST	Global cold
	SYS_PWRON_RST	Global cold
	SYS_WARMIN_RST	Global warm
	TSHUT_CORE_RST	Global warm
	ICEPICK_RST	Global warm
CORE_RST	GLOBAL_COLD_SW_RST	Global cold
	GLOBAL_WARM_SW_RST	Global warm
	ICEPICK_POR_RST	Global cold
	SYS_PWRON_RST	Global cold
	SYS_WARMIN_RST	Global warm
	TSHUT_CORE_RST	Global warm
	ICEPICK_RST	Global warm
CUSTEFUSE_RST	GLOBAL_COLD_SW_RST	Global cold
	GLOBAL_WARM_SW_RST	Global warm
	ICEPICK_POR_RST	Global cold
	SYS_PWRON_RST	Global cold
	SYS_WARMIN_RST	Global warm
	TSHUT_CORE_RST	Global warm
	ICEPICK_RST	Global warm
DLL_RST	DLL_FREQCHANGE_RST	Local warm
	GLOBAL_COLD_SW_RST	Global cold
	GLOBAL_WARM_SW_RST	Global warm
	ICEPICK_POR_RST	Global cold
	SYS_PWRON_RST	Global cold
	SYS_WARMIN_RST	Global warm
	TSHUT_CORE_RST	Global warm
	ICEPICK_RST	Global warm
DPLL_CORE_PWRON_RET_RST	GLOBAL_COLD_SW_RST	Global cold
	ICEPICK_POR_RST	Global cold
	SYS_PWRON_RST	Global cold
DSS_RET_RST	GLOBAL_COLD_SW_RST	Global cold
	GLOBAL_WARM_SW_RST	Global warm

Table 3-34. Reset Sources for the Reset Domains (continued)

Reset Domain	Reset Source	Reset Source Type
	ICEPICK_POR_RST	Global cold
	SYS_PWRON_RST	Global cold
	SYS_WARMIN_RST	Global warm
	TSHUT_CORE_RST	Global warm
	ICEPICK_RST	Global warm
DSS_RST	GLOBAL_COLD_SW_RST	Global cold
	GLOBAL_WARM_SW_RST	Global warm
	ICEPICK_POR_RST	Global cold
	SYS_PWRON_RST	Global cold
	SYS_WARMIN_RST	Global warm
	TSHUT_CORE_RST	Global warm
	ICEPICK_RST	Global warm
IPU_PWRON_RST	GLOBAL_COLD_SW_RST	Global cold
	ICEPICK_POR_RST	Global cold
	SYS_PWRON_RST	Global cold
IPU_RET_RST	GLOBAL_COLD_SW_RST	Global cold
	GLOBAL_WARM_SW_RST	Global warm
	ICEPICK_POR_RST	Global cold
	SYS_PWRON_RST	Global cold
	SYS_WARMIN_RST	Global warm
	TSHUT_CORE_RST	Global warm
	ICEPICK_RST	Global warm
IPU_CPU0_RST	IPU_ICECRUSHER0_RST	Local warm
	GLOBAL_COLD_SW_RST	Global cold
	GLOBAL_WARM_SW_RST	Global warm
	ICEPICK_POR_RST	Global cold
	SYS_PWRON_RST	Global cold
	SYS_WARMIN_RST	Global warm
	TSHUT_CORE_RST	Global warm
	ICEPICK_RST	Global warm
IPU_CPU1_RST	IPU_ICECRUSHER1_RST	Local Warm
	GLOBAL_COLD_SW_RST	Global cold
	GLOBAL_WARM_SW_RST	Global warm
	ICEPICK_POR_RST	Global cold
	SYS_PWRON_RST	Global cold
	SYS_WARMIN_RST	Global warm
	TSHUT_CORE_RST	Global warm
	ICEPICK_RST	Global warm
IPU_RST	GLOBAL_COLD_SW_RST	Global cold
	GLOBAL_WARM_SW_RST	Global warm
	ICEPICK_POR_RST	Global cold
	SYS_PWRON_RST	Global cold
	SYS_WARMIN_RST	Global warm
	TSHUT_CORE_RST	Global warm
	ICEPICK_RST	Global warm
EMU_EARLY_PWRON_RST	GLOBAL_COLD_SW_RST	Global cold
	SYS_PWRON_RST	Global cold

Table 3-34. Reset Sources for the Reset Domains (continued)

Reset Domain	Reset Source	Reset Source Type
EMU_PWRON_RST	GLOBAL_COLD_SW_RST	Global cold
	ICEPICK_POR_RST	Global cold
	SYS_PWRON_RST	Global cold
EMU_RST	GLOBAL_COLD_SW_RST	Global cold
	GLOBAL_WARM_SW_RST	Global warm
	ICEPICK_POR_RST	Global cold
	SYS_PWRON_RST	Global cold
	SYS_WARMIN_RST	Global warm
	TSHUT_CORE_RST	Global warm
	ICEPICK_RST	Global warm
L3INIT_PWRON_RST	GLOBAL_COLD_SW_RST	Global cold
	ICEPICK_POR_RST	Global cold
	SYS_PWRON_RST	Global cold
L3INIT_RET_RST	GLOBAL_COLD_SW_RST	Global cold
	GLOBAL_WARM_SW_RST	Global warm
	ICEPICK_POR_RST	Global cold
	SYS_PWRON_RST	Global cold
	SYS_WARMIN_RST	Global warm
	TSHUT_CORE_RST	Global warm
	ICEPICK_RST	Global warm
L3INIT_RST	GLOBAL_COLD_SW_RST	Global cold
	GLOBAL_WARM_SW_RST	Global warm
	ICEPICK_POR_RST	Global cold
	SYS_PWRON_RST	Global cold
	SYS_WARMIN_RST	Global warm
	TSHUT_CORE_RST	Global warm
	ICEPICK_RST	Global warm
DSP1_RST	GLOBAL_COLD_SW_RST	Global cold
	GLOBAL_WARM_SW_RST	Global warm
	ICEPICK_POR_RST	Global cold
	RM_DSP1_RSTCTRL[0] RST_DSP1_LRST	Local warm
	SYS_PWRON_RST	Global cold
	SYS_WARMIN_RST	Global warm
	DSP1_EMU_RESET_REQ_TR	Local warm
	TSHUT_CORE_RST	Global warm
	ICEPICK_RST	Global warm
DSP1_PWRON_RST	GLOBAL_COLD_SW_RST	Global cold
	ICEPICK_POR_RST	Global cold
	SYS_PWRON_RST	Global cold
DSP1_RET_RST	GLOBAL_COLD_SW_RST	Global cold
	GLOBAL_WARM_SW_RST	Global warm
	ICEPICK_POR_RST	Global cold
	RM_DSP1_RSTCTRL[1] RST_DSP1	Local warm
	SYS_PWRON_RST	Global cold
	SYS_WARMIN_RST	Global warm
	TSHUT_CORE_RST	Global warm
	ICEPICK_RST	Global warm

Table 3-34. Reset Sources for the Reset Domains (continued)

Reset Domain	Reset Source	Reset Source Type
DSP1_SYS_RST	GLOBAL_COLD_SW_RST	Global cold
	GLOBAL_WARM_SW_RST	Global warm
	ICEPICK_POR_RST	Global cold
	RM_DSP1_RSTCTRL[1] RST_DSP1	Local warm
	SYS_PWRON_RST	Global cold
	SYS_WARMIN_RST	Global warm
	TSHUT_CORE_RST	Global warm
	ICEPICK_RST	Global warm
DSP2_RST	GLOBAL_COLD_SW_RST	Global cold
	GLOBAL_WARM_SW_RST	Global warm
	ICEPICK_POR_RST	Global cold
	RM_DSP2_RSTCTRL[0] RST_DSP2_LRST	Local warm
	SYS_PWRON_RST	Global cold
	SYS_WARMIN_RST	Global warm
	DSP2_EMU_RESET_REQ_TR	Local warm
	TSHUT_CORE_RST	Global warm
	ICEPICK_RST	Global warm
DSP2_PWRON_RST	GLOBAL_COLD_SW_RST	Global cold
	ICEPICK_POR_RST	Global cold
	SYS_PWRON_RST	Global cold
DSP2_RET_RST	GLOBAL_COLD_SW_RST	Global cold
	GLOBAL_WARM_SW_RST	Global warm
	ICEPICK_POR_RST	Global cold
	RM_DSP2_RSTCTRL[1] RST_DSP2	Local warm
	SYS_PWRON_RST	Global cold
	SYS_WARMIN_RST	Global warm
	TSHUT_CORE_RST	Global warm
	ICEPICK_RST	Global warm
DSP2_SYS_RST	GLOBAL_COLD_SW_RST	Global cold
	GLOBAL_WARM_SW_RST	Global warm
	ICEPICK_POR_RST	Global cold
	RM_DSP2_RSTCTRL[1] RST_DSP2	Local warm
	SYS_PWRON_RST	Global cold
	SYS_WARMIN_RST	Global warm
	TSHUT_CORE_RST	Global warm
	ICEPICK_RST	Global warm
WKUPAON_PWRON_RST	GLOBAL_COLD_SW_RST	Global cold
	ICEPICK_POR_RST	Global cold
	SYS_PWRON_RST	Global cold
WKUPAON_RST	GLOBAL_COLD_SW_RST	Global cold
	GLOBAL_WARM_SW_RST	Global warm
	ICEPICK_POR_RST	Global cold
	SYS_PWRON_RST	Global cold
	SYS_WARMIN_RST	Global warm
	TSHUT_CORE_RST	Global warm
	ICEPICK_RST	Global warm
WKUPAON_SYS_PWRON_RST	SYS_PWRON_RST	Global cold

Table 3-34. Reset Sources for the Reset Domains (continued)

Reset Domain	Reset Source	Reset Source Type
PRM_RST	GLOBAL_COLD_SW_RST	Global cold
	GLOBAL_WARM_SW_RST	Global warm
	ICEPICK_POR_RST	Global cold
	SYS_PWRON_RST	Global cold
	SYS_WARMIN_RST	Global warm
	TSHUT_CORE_RST	Global warm
	ICEPICK_RST	Global warm
PRM_PWRON_RST	GLOBAL_COLD_SW_RST	Global cold
	ICEPICK_POR_RST	Global cold
	SYS_PWRON_RST	Global cold
LPRM_RST	GLOBAL_COLD_SW_RST	Global cold
	GLOBAL_WARM_SW_RST	Global warm
	ICEPICK_POR_RST	Global cold
	SYS_PWRON_RST	Global cold
	SYS_WARMIN_RST	Global warm
	TSHUT_CORE_RST	Global warm
	ICEPICK_RST	Global warm
LPRM_PWRON_RST	GLOBAL_COLD_SW_RST	Global cold
	ICEPICK_POR_RST	Global cold
	SYS_PWRON_RST	Global cold
ISS_RST	GLOBAL_COLD_SW_RST	Global cold
	GLOBAL_WARM_SW_RST	Global warm
	ICEPICK_POR_RST	Global cold
	SYS_PWRON_RST	Global cold
	SYS_WARMIN_RST	Global warm
	TSHUT_CORE_RST	Global warm
	ICEPICK_RST	Global warm
L4PER_RST	GLOBAL_COLD_SW_RST	Global cold
	GLOBAL_WARM_SW_RST	Global warm
	ICEPICK_POR_RST	Global cold
	SYS_PWRON_RST	Global cold
	SYS_WARMIN_RST	Global warm
	TSHUT_CORE_RST	Global warm
	ICEPICK_RST	Global warm
L4PER_RET_RST	GLOBAL_COLD_SW_RST	Global cold
	GLOBAL_WARM_SW_RST	Global warm
	ICEPICK_POR_RST	Global cold
	SYS_PWRON_RST	Global cold
	SYS_WARMIN_RST	Global warm
	TSHUT_CORE_RST	Global warm
	ICEPICK_RST	Global warm
L4PER_PWRON_RET_RST	GLOBAL_COLD_SW_RST	Global cold
	ICEPICK_POR_RST	Global cold
	SYS_PWRON_RST	Global cold
EVE_CPU_RST	GLOBAL_COLD_SW_RST	Global cold
	GLOBAL_WARM_SW_RST	Global warm
	ICEPICK_POR_RST	Global cold

Table 3-34. Reset Sources for the Reset Domains (continued)

Reset Domain	Reset Source	Reset Source Type
	EVE_EMU_RESET_REQ_TR	Local warm
	SYS_PWRON_RST	Global cold
	SYS_WARMIN_RST	Global warm
	RM_EVE1_RSTCTRL[0] RST_EVE_LRST	Local warm
	TSHUT_CORE_RST	Global warm
	ICEPICK_RST	Global warm
EVE_RST	GLOBAL_COLD_SW_RST	Global cold
	GLOBAL_WARM_SW_RST	Global warm
	ICEPICK_POR_RST	Global cold
	SYS_PWRON_RST	Global cold
	SYS_WARMIN_RST	Global warm
	RM_EVE1_RSTCTRL[1] RST_EVE	Local warm
	TSHUT_CORE_RST	Global warm
	ICEPICK_RST	Global warm
EVE_PWRON_RST	GLOBAL_COLD_SW_RST	Global cold
	ICEPICK_POR_RST	Global cold
	SYS_PWRON_RST	Global cold
DSS_RST	GLOBAL_COLD_SW_RST	Global cold
	GLOBAL_WARM_SW_RST	Global warm
	ICEPICK_POR_RST	Global cold
	SYS_PWRON_RST	Global cold
	SYS_WARMIN_RST	Global warm
	TSHUT_CORE_RST	Global warm
	ICEPICK_RST	Global warm
DSS_RET_RST	GLOBAL_COLD_SW_RST	Global cold
	GLOBAL_WARM_SW_RST	Global warm
	ICEPICK_POR_RST	Global cold
	SYS_PWRON_RST	Global cold
	SYS_WARMIN_RST	Global warm
	TSHUT_CORE_RST	Global warm
	ICEPICK_RST	Global warm

Table 3-35 lists the attributes of the reset manager associated with the reset domains. The clock to the reset manager, the delay count before release of reset, and the reset release stall conditions for the reset domains are listed.

Table 3-35. Reset Domains Attributes

Reset Domain	RM Clock	RM Clock Count	Release Stall Conditions
CM_CORE_AON_PWRON_RST	WKUPAON_GCLK	0x2	None
CM_CORE_AON_RST	WKUPAON_GCLK	0x2	L4_ROOT_CLK clock is not active.
COREAON_PWRON_RST	WKUPAON_GCLK	0x0	None
COREAON_RST	WKUPAON_GCLK	0x0	None
MMAON_RST	WKUPAON_GCLK	ResetTime2[(1)]	DSP_GFCLK is not active.
DPLL_EVE_VID_DSP_PWRON_RST	WKUPAON_GCLK	ResetTime2[(1)]	None
DPLL_EVE_PWRON_RST	WKUPAON_GCLK	ResetTime2[(1)]	None

(1) ResetTime2 is set in the PRM_RSTTIME[14:10] RSTTIME2 bit field.

Table 3-35. Reset Domains Attributes (continued)

Reset Domain	RM Clock	RM Clock Count	Release Stall Conditions
CUSTEFUSE_RST	WKUPAON_GCLK	ResetTime2[1]	CUSTEFUSE_SYS_GFCLK is not active.
CAM_RST	WKUPAON_GCLK	0x0	None
CM_CORE_PWRON_RET_RST	WKUPAON_GCLK	0x2	None
CM_CORE_RET_RST	WKUPAON_GCLK	0x2	L4_ICLK clock is not active.
CORE_PWRON_RET_RST	WKUPAON_GCLK	0x0	None
CORE_PWRON_RST	WKUPAON_GCLK	0x0	None
CORE_RET_RST	WKUPAON_GCLK	0x0	None
CORE_RST	WKUPAON_GCLK	0x0	None
DLL_RST	WKUPAON_GCLK	0x3	None
DMA_RET_RST	WKUPAON_GCLK	0x0	None
DPLL_L3INIT_PWRON_RET_RST	WKUPAON_GCLK	0x0	None
DSS_RET_RST	WKUPAON_GCLK	0x0	None
DSS_RST	WKUPAON_GCLK	0x0	None
IPU_PWRON_RST	WKUPAON_GCLK	ResetTime2[1]	IPU_GCLK clock is not active and automatic restore is complete.
IPU_RET_RST	WKUPAON_GCLK	ResetTime2[1]	IPU_GFCLK clock is not active and the subsystem is reset.
IPU_CPU0_RST	WKUPAON_GCLK	ResetTime2[1]	IPU_GFCLK clock is not active and the subsystem is reset.
IPU_CPU1_RST	WKUPAON_GCLK	ResetTime2[1]	IPU_GFCLK clock is not active.
IPU_RST	WKUPAON_GCLK	ResetTime2[1]	IPU_GFCLK clock is not active and the subsystem is reset.
EMU_EARLY_PWRON_RST	WKUPAON_ICLK	0x20	None
EMU_PWRON_RST	WKUPAON_ICLK	ResetTime2[1]	EMU_SYS_CLK clock is not active.
EMU_RST	WKUPAON_ICLK	ResetTime2[1]	EMU_SYS_CLK clock is not active.
L3INIT_PWRON_RST	WKUPAON_GCLK	0x0	None
L3INIT_RET_RST	WKUPAON_GCLK	0x0	None
L3INIT_RST	WKUPAON_GCLK	0x0	None
DSP1_RST	WKUPAON_GCLK	ResetTime2[1]	DSP1_GFCLK clock is not active and the subsystem is reset.
DSP1_PWRON_RST	WKUPAON_GCLK	ResetTime2[1]	DSP1_GFCLK clock is not active, RM_DSP1_RSTCTRL[1] RST_DSP1 bit is cleared, and automatic restore is complete.

Table 3-35. Reset Domains Attributes (continued)

Reset Domain	RM Clock	RM Clock Count	Release Stall Conditions
DSP1_RET_RST	WKUPAON_GCLK	ResetTime2[1]	DSP1_GFCLK clock is not active and the subsystem is reset.
DSP1_SYS_RST	WKUPAON_GCLK	ResetTime2[1]	DSP1_GFCLK clock is not active and the subsystem is reset.
DSP2_RST	WKUPAON_GCLK	ResetTime2[1]	DSP2_GFCLK clock is not active and the subsystem is reset.
DSP2_PWRON_RST	WKUPAON_GCLK	ResetTime2[1]	DSP2_GFCLK clock is not active, RM_DSP2_RSTCTRL[1] RST_DSP2 bit is cleared, and automatic restore is complete.
DSP2_RET_RST	WKUPAON_GCLK	ResetTime2[1]	DSP2_GFCLK clock is not active and the subsystem is reset.
DSP2_SYS_RST	WKUPAON_GCLK	ResetTime2[1]	DSP2_GFCLK clock is not active and the subsystem is reset.
WKUPAON_PWRON_RST	WKUPAON_GCLK	0x0	None
WKUPAON_RST	WKUPAON_GCLK	0x0	None
WKUPAON_SYS_PWRON_RST	WKUPAON_GCLK	0x0	None
PRM_PWRON_RST	WKUPAON_GCLK	0x0	WKUPAON_ICLK clock is not active.
PRM_RST	WKUPAON_GCLK	0x0	WKUPAON_ICLK clock is not active.
LPRM_PWRON_RST	WKUPAON_GCLK	0x0	WKUPAON_ICLK clock is not active.
LPRM_RST	WKUPAON_GCLK	0x0	WKUPAON_ICLK clock is not active.
ISS_RST	WKUPAON_GCLK	ResetTime2[1]	None
L4PER_PWRON_RET_RST	WKUPAON_GCLK	0x0	None
L4PER_RET_RST	WKUPAON_GCLK	0x0	None
L4PER_RST	WKUPAON_GCLK	0x0	None
EVE_CPU_RST	WKUPAON_GCLK	ResetTime2[1]	None
EVE_PWRON_RST	WKUPAON_GCLK	ResetTime2[1]	EVE_GFCLK is not active.
EVE_RST	WKUPAON_GCLK	ResetTime2[1]	None

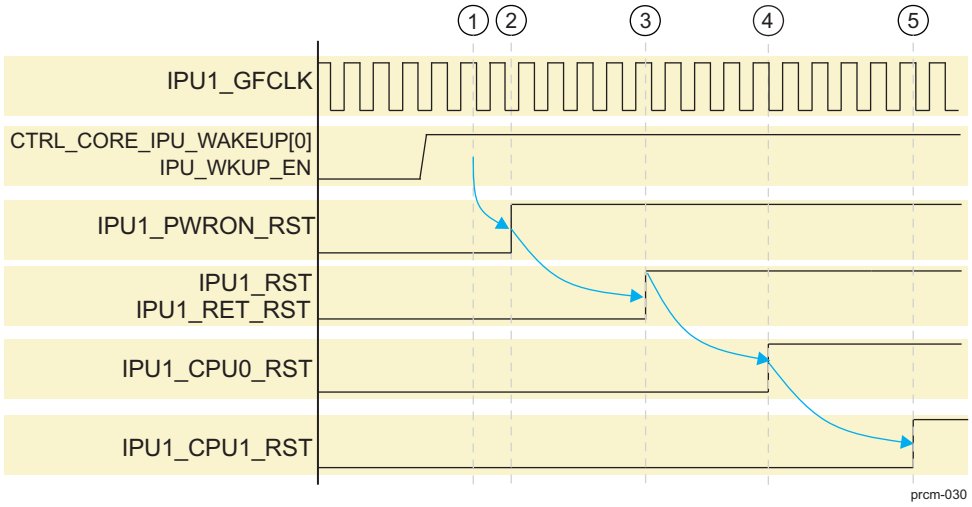
NOTE: WKUPAON_SYS_PWRON_RST is connected directly to the SYS_PWRON_RST source reset.

3.5.6 Reset Sequences

3.5.6.1 IPU1 Subsystem Power-On Reset Sequence

Figure 3-20 shows the power-on reset sequence of the IPU1 subsystem.

Figure 3-20. IPU1 Power-On Reset Sequence



The assumptions on power-on reset assertion are:

- The IPU subsystem is held in reset by the PRCM module and the following are asserted:
  - IPU1_PWRON_RST
  - IPU1_RET_RST
  - IPU1_CPU0_RST
  - IPU1_CPU1_RST
  - IPU1_RST

The power-on reset sequence is:

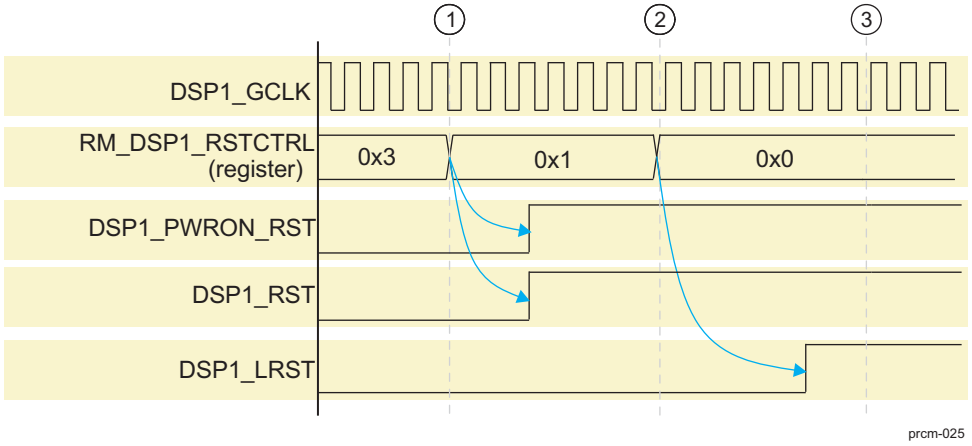
1. Device power-on reset is released. CTRL_CORE_IPU_WAKEUP[0] IPU_WKUP_EN reset value is set to 1 to allow starting up the IPU. This includes initiating the IPU's reset release sequence, as well as setting the MODULEMODE to HW AUTO.
2. The PRCM module releases IPU1_PWRON_RST once the reset manager counter (PRM_RSTTIME[14:10] RSTTIME2) reaches its limit and the IPU1_GFCLK is running. Upon deassertion of the POR signal, the IPU subsystem starts the CPU and CACHE_MMU_IPU initialization sequence.
3. When the reset sequence of Step 2 completes and the reset manager counter (PRM_RSTTIME[14:10] RSTTIME2) expires, the PRCM module releases the IPU1_RST and IPU1_RET_RST signals.
4. The PRCM module releases IPU_CPU0_RST, which causes IPU_C0 to start executing code from IPU_BOOTROM.
5. The PRCM module releases IPU_CPU1_RST to IPU_C1. IPU_C1 is in wait-for-event state until boot ROM code wakes it up to execute code.

NOTE: IPU or other device core can disable IPU subsystem by clearing CTRL_CORE_IPU_WAKEUP[0] IPU_WKUP_EN to 0.

3.5.6.2 DSP1 Subsystem Power-On Reset Sequence

Figure 3-21 shows the power-on reset sequence of the DSP1 subsystem.

Figure 3-21. DSP1 Subsystem Power-On Reset Sequence



The power-on reset to DSP1 is applied when PD_DSP1 is powered. The assumptions ofter power-on reset assertion are:

- PD_DSP1 is on
- The PRCM module provides DSP1_GCLK functional clock to the DSP subsystem, and it has been enabled by IPU software control.

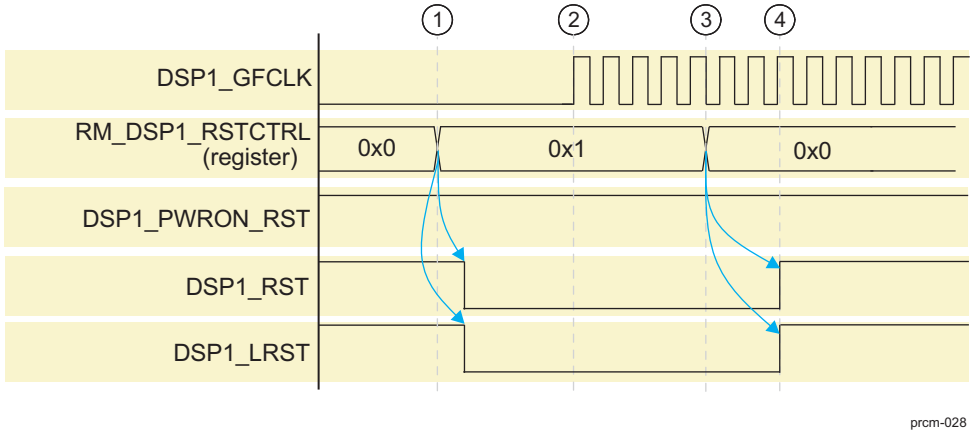
The Power-On Reset sequence is:

1. Software clears the RM_DSP1_RSTCTRL[1] RST_DSP1 bit. This causes the PRCM module to release the DSP1_PWRON_RST used inside DSP1 mainly to reset the emulation logic and the DSP1_RST used to reset all logic inside DSP1. Then software can download data into TCM memory while keeping the CPU under reset.
2. When the memory is initialized, software clears the RM_DSP1_RSTCTRL[0] RST_DSP1_LRST bit. This release DSP1_LRST to the local CPU inside DSP subsystem.

3.5.6.3 DSP1 Subsystem Software Warm Reset Sequence

Figure 3-22 shows the software warm reset sequence of the DSP1 subsystem.

Figure 3-22. DSP1 Subsystem Software Warm Reset Sequence



Before asserting the software reset to the DSP subsystem, the IPU software must ensure that:

- The DSP CPUs are in IDLE state (CM_DSP1_DSP1_CLKCTRL[17:16] IDLEST).
- The DSP subsystem is in STANDBY state (CM_DSP1_DSP1_CLKCTRL[18] STBYST).
- The functional clock to the DSP subsystem has been gated by the PRCM module (CM_DSP1_CLKSTCTRL[8] CLKACTIVITY_DSP1_GFCLK)

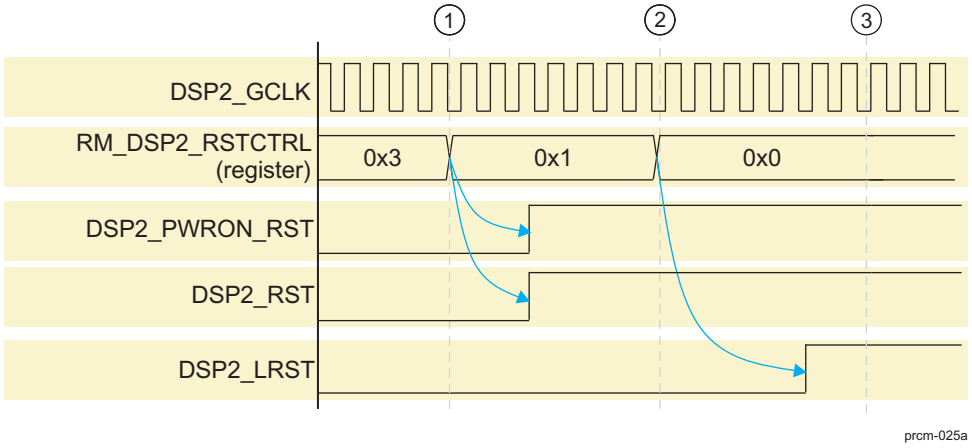
The software reset sequence is:

1. IPU software sets the RM_DSP1_RSTCTRL[1] RST_DSP1 = 0 and RM_DSP1_RSTCTRL[0] RST_DSP1_LRST = 1. This causes the PRCM module to assert DSP1_RST, DSP1_LRST to the DSP subsystem. The DSP1_PWRON_RST remains deasserted.
2. The IPU software enables the functional clock to the DSP subsystem.
3. The IPU software cleares the RM_DSP1_RSTCTRL[1] RST_DSP1 and RM_DSP1_RSTCTRL[0] RST_DSP1_LRST bits. This causes the PRCM module to release DSP1_RST and DSP1_LRST to the DSP subsystem.

3.5.6.4 DSP2 Subsystem Power-On Reset Sequence

Figure 3-23 shows the power-on reset sequence of the DSP2 subsystem.

Figure 3-23. DSP2 Subsystem Power-On Reset Sequence



The power-on reset to DSP2 is applied when PD_DSP2 is powered. The assumptions ofter power-on reset assertion are:

- PD_DSP2 is on.
- The PRCM module provides DSP2_GCLK functional clock to the DSP subsystem, and it has been enabled by IPU software control.

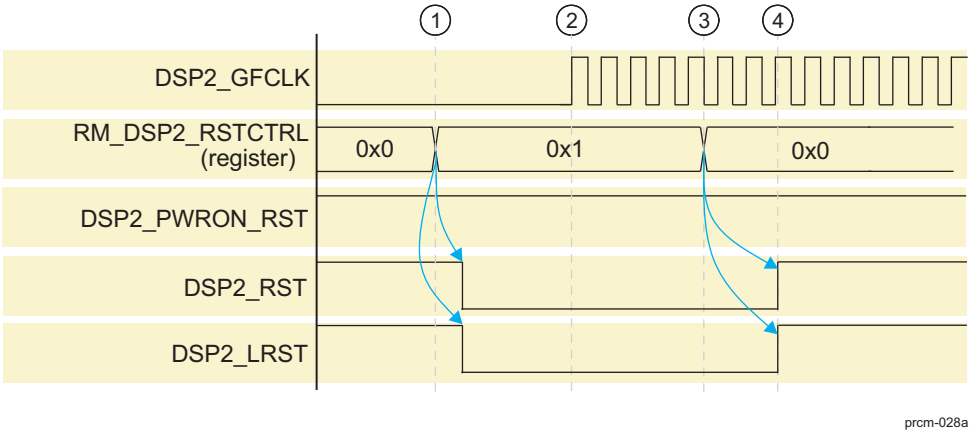
The Power-On Reset sequence is:

1. Software clears the RM_DSP2_RSTCTRL[1] RST_DSP2 bit. This causes the PRCM module to release the DSP2_PWRON_RST used inside DSP2 mainly to reset the emulation logic and the DSP2_RST used to reset all logic inside DSP2. Then software can download data into TCM memory while keeping the CPU under reset.
2. When the memory is initialized, software clears the RM_DSP2_RSTCTRL[0] RST_DSP2_LRST bit. This releases DSP2_LRST to the local CPU inside DSP subsystem.

3.5.6.5 DSP2 Subsystem Software Warm Reset Sequence

Figure 3-22 shows the software warm reset sequence of the DSP2 subsystem.

Figure 3-24. DSP2 Subsystem Software Warm Reset Sequence



Before asserting the software reset to the DSP subsystem, the IPU software must ensure that:

- The DSP CPUs are in IDLE state (CM_DSP2_DSP2_CLKCTRL[17:16] IDLEST).
- The DSP subsystem is in STANDBY state (CM_DSP2_DSP2_CLKCTRL[18] STBYST).
- The functional clock to the DSP subsystem has been gated by the PRCM module (CM_DSP2_CLKSTCTRL[8] CLKACTIVITY_DSP2_GFCLK)

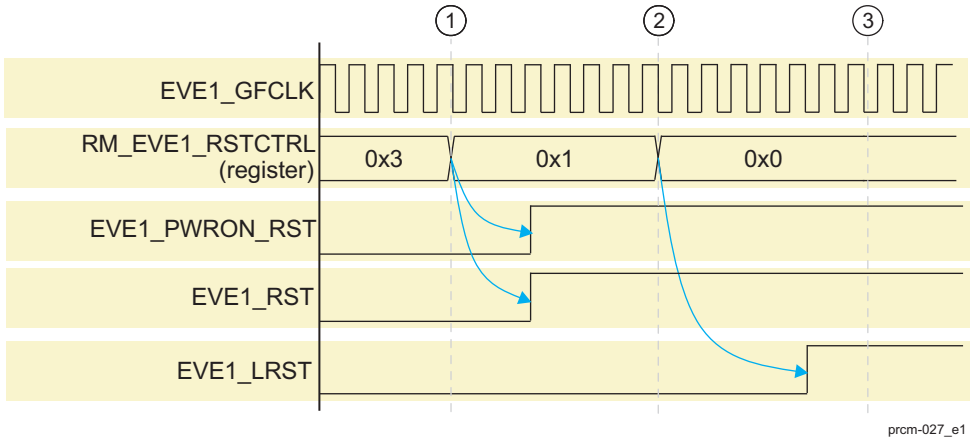
The software reset sequence is:

1. IPU software sets the RM_DSP2_RSTCTRL[1] RST_DSP2 = 0 and RM_DSP2_RSTCTRL[0] RST_DSP2_LRST = 1. This causes the PRCM module to assert DSP2_RST, DSP2_LRST to the DSP subsystem. The DSP2_PWRON_RST remains deasserted.
2. The IPU software enables the functional clock to the DSP subsystem.
3. The IPU software cleares the RM_DSP2_RSTCTRL[1] RST_DSP2 and RM_DSP2_RSTCTRL[0] RST_DSP2_LRST bits. This causes the PRCM module to release DSP2_RST and DSP2_LRST to the DSP subsystem.

3.5.6.6 EVE Subsystem Power-On Reset Sequence

Figure 3-25 shows the power-on reset sequence of the EVE subsystem.

Figure 3-25. EVE Subsystem Power-On Reset Sequence



The power-on reset to EVE is applied when PD_EVE is powered. The assumptions on power-on reset assertion are:

- The PRCM module provides the EVE_GFCLK functional clock to the EVE subsystem, and it has been enabled by IPU software control.

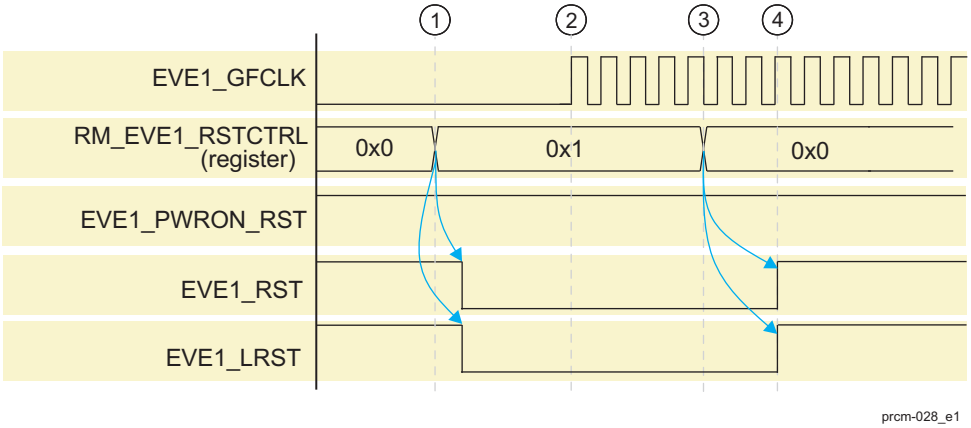
The power-on reset sequence is:

1. Software clears the RM_EVE1_RSTCTRL[1] RST_EVE bit. This causes the PRCM module to release the EVE_PWRON_RST and the EVE_RST signals. Then software can download data into TCM memory while keeping the sequencer CPUs under reset.
2. When the TCM memory is initialized, software clears the RM_EVE1_RSTCTRL[0] RST_EVE_LRST bit .This release EVE_LRST to the local CPU inside EVE subsystem.

3.5.6.7 EVE Subsystem Software Warm Reset Sequence

Figure 3-26 shows the software warm reset sequence of the EVE subsystem.

Figure 3-26. EVE Subsystem Software Warm Reset Sequence



For doing the software reset of EVE the IPU software must ensure that EVE CPU is in IDLE state and EVE is in STANDBY state and the functional clock to EVE has been gated.

The software warm reset sequence is:

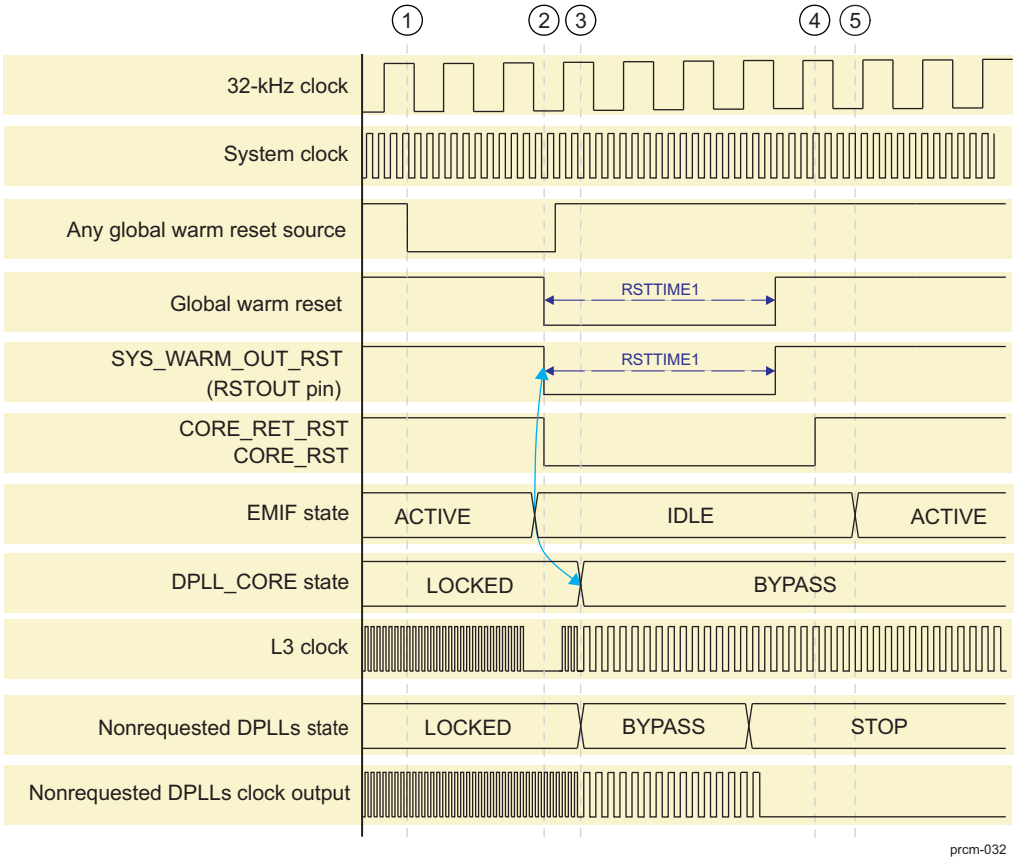
1. IPU software sets the RM_EVE1_RSTCTRL[1] RST_EVE bit. This causes the PRCM module to release the EVE_RST, EVE_LRST to the EVE subsystem. The EVE_PWRON_RST remains deasserted.
2. The IPU software enables the functional clock to the EVE subsystem.
3. The IPU software cleares the RM_EVE1_RSTCTRL[1] RST_EVE = 0 and RM_EVE1_RSTCTRL[0] RST_EVE_LRST = 1. This causes the PRCM module to release EVE_RST and EVE_LRST to the EVE subsystem.

3.5.6.8 Global Warm Reset Sequence

This section describes the global warm reset sequence.

Figure 3-27 shows the global warm reset sequence.

Figure 3-27. Global Warm Reset Sequence



The assumptions are:

- All the logic and memory voltage sources are at their nominal voltage levels.
- The device is active:
  - All resets are released.
  - CORE DPLL is locked.

The steps of a global warm reset sequence are:

1. On assertion of any global warm reset source the PRM signals the EMIF that a global warm reset event has occurred. The EMIF initiates the transition to IDLE state. The PRCM module delays global warm reset to the device for a minimum of 16 L3 clock cycles so that the EMIF switches to IDLE state and switches the external SDRAM to self-refresh mode.
2. The reset managers in the PRM assert the following resets:
   - The external warm reset SYS_WARM_OUT_RST (rstoutn pin).
   - All power domain warm resets are asserted.
   - The PRM and CM registers, sensitive to warm reset, are asynchronously reset.
   - DPLL hardware resets are not asserted.
     - DPLL_CORE transitions to bypass mode once the EMIF switches to IDLE state.
     - DPLL_PER, DPLL_EVE_VID_DSP, and DPLL_DDR transition to idle bypass low-power mode. Then as clock signals are no more requested, they are gated and these DPLLs goes to stop mode.
     - DPLL_GMAC_DSP configuration is not changed.
   - CM gates the clocks that are not needed, as per their default reset setting in the associated registers.

3. The device warm reset (internal and rstoutn pin) duration is set up by the PRM_RSTTIME[9:0] RSTTIME1 bit field. It defines the global warm reset duration in number of FUNC_32K_CLK clock cycles. Default value loaded after POR is 6 clock cycles.
4. The CORE power domain is released from reset (that is, warm reset-sensitive modules in the CORE power domain).
5. The PRCM module switches the EMIF from IDLE state back to ACTIVE state.

NOTE:

- The PD_DSP and PD_EVE power domains are held under reset after global warm reset by assertion of the software source of the reset.
- The following are held under reset after global warm reset until the PRCM module enables their interface clock:
  - PD_L4PER
  - PD_L3INIT
  - PD_DSS
  - PD_CAM

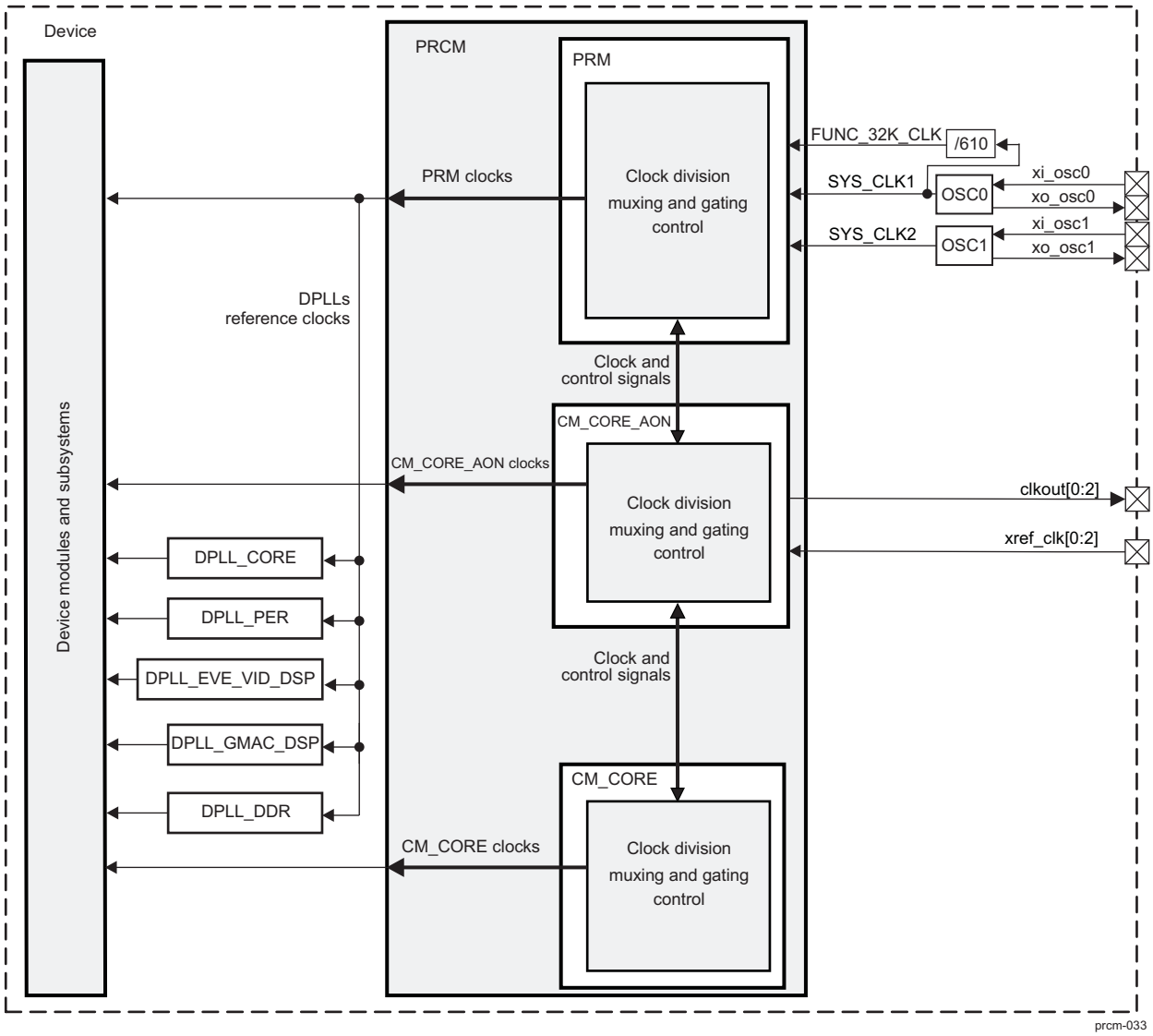
3.6 Clock Management Functional Description

3.6.1 Overview

The PRCM module provides the control for clock generation, division, distribution, synchronization, and gating. It distributes the clock sources to all modules in the device. For information about the clock-management functional architecture of the device, see Section 3.1.1.1, *Clock Management*.

The PRCM module provides clocks to the internal DPLLs for internal high-frequency clock generation. Clock division and gating are handled by the PRM, CM_CORE_AON, and CM_CORE sections of the PRCM module. Figure 3-28 shows the high-level clock-management scheme in the device.

Figure 3-28. PRCM Module Clock Manager Overview



3.6.2 External Clock Inputs

3.6.2.1 High-Frequency System Clock Input

The system clocks SYS_CLK1 and SYS_CLK2, are the main source clocks of the device. SYS_CLK1 and SYS_CLK2 are received from internal oscillators (OSC0 and OSC1) of the PRCM module. They are supplied as the reference clock to the DPLLs and as functional clock to several modules.

3.6.2.2 External Reference Clock Input

The external reference clocks xref_clk[0:2] are received directly from external reference clock source for the device. They are supplied as the functional clock to the TIMERs, and as reference clock to the McASP and other peripherals.

3.6.3 Internal Clock Sources/Generators

The PRCM module clock sources/generators are split into the following parts:

- PRM clock source that receives system clocks SYS_CLK1 and SYS_CLK2 inputs.
- CM_CORE_AON and CM_CORE clock sources that distribute high-frequency clocks
- DPLL clock generators that synthesize high-frequency clocks for the device

Table 3-36. Internal Clock Sources

Clock Name	Source	Frequency	Note
SYS_CLK1	OSC0	19.2 MHz, 20 MHz, or 27 MHz	See Section 3.6.3.1, *PRM Clock Source*
SYS_CLK2	OSC1	19.2 – 32 MHz	See Section 3.6.3.1, *PRM Clock Source*
FUNC_32K_CLK	SYS_CLK1/610	32.786 kHz(1)	See Section 3.6.3.1, *PRM Clock Source*
RCOSC_32K_CLK	On-die RC Oscillator	32 kHz(2)	Section 3.6.3.2.2, *CM_CORE_AON_CLKOUTMUX Overview*
FUNC_96M_AON_CLK	DPLL_PER	96 MHz	See Section 3.6.3.5, *DPLL_PER Description*
FUNC_192M_CLK	DPLL_PER	192 MHz	See Section 3.6.3.5, *DPLL_PER Description*
FUNC_128M_CLK	DPLL_PER	128 MHz	See Section 3.6.3.5, *DPLL_PER Description*
CORE_CLK	DPLL_CORE	532 MHz	See Section 3.6.3.6, *DPLL_CORE Description*

(1) Bacause FUNC_32K_CLK is derived from the SYS_CLK1, and SYS_CLK1 can be one of three different clock rates, FUNC_32K_CLK will also have three different rates. FUNC_32K_CLK will be close to its nominal frequency only at SYS_CLK1 rate of 20 MHz.

(2) Resistor-Capacitor (RC) oscillator frequency will depend on PVT variation. See Data Manual for more details.

3.6.3.1 PRM Clock Source

The PRM clock source receives the SYS_CLK1 and SYS_CLK2 clocks from external input pins. The PRM manages the low-frequency clocks associated with these input clocks. The PRM sources various versions (through gating controls) of these externally sourced clocks to supply:

- PRCM-managed DPLLs with a reference clock, which is permanently supplied with always-on buffers
- The DSS with a reference clock, which is permanently supplied with always-on buffers
- The various timers and watchdog timers with clocks supplied by always-on buffers
- The clocks for the CM clock generator and the CORE power domain
- Timer functional clocks
- The bandgap and control module for thermal sensor feature

Figure 3-29 is a logical representation of the PRM clock source.

Figure 3-29. PRM Clock Manager Overview

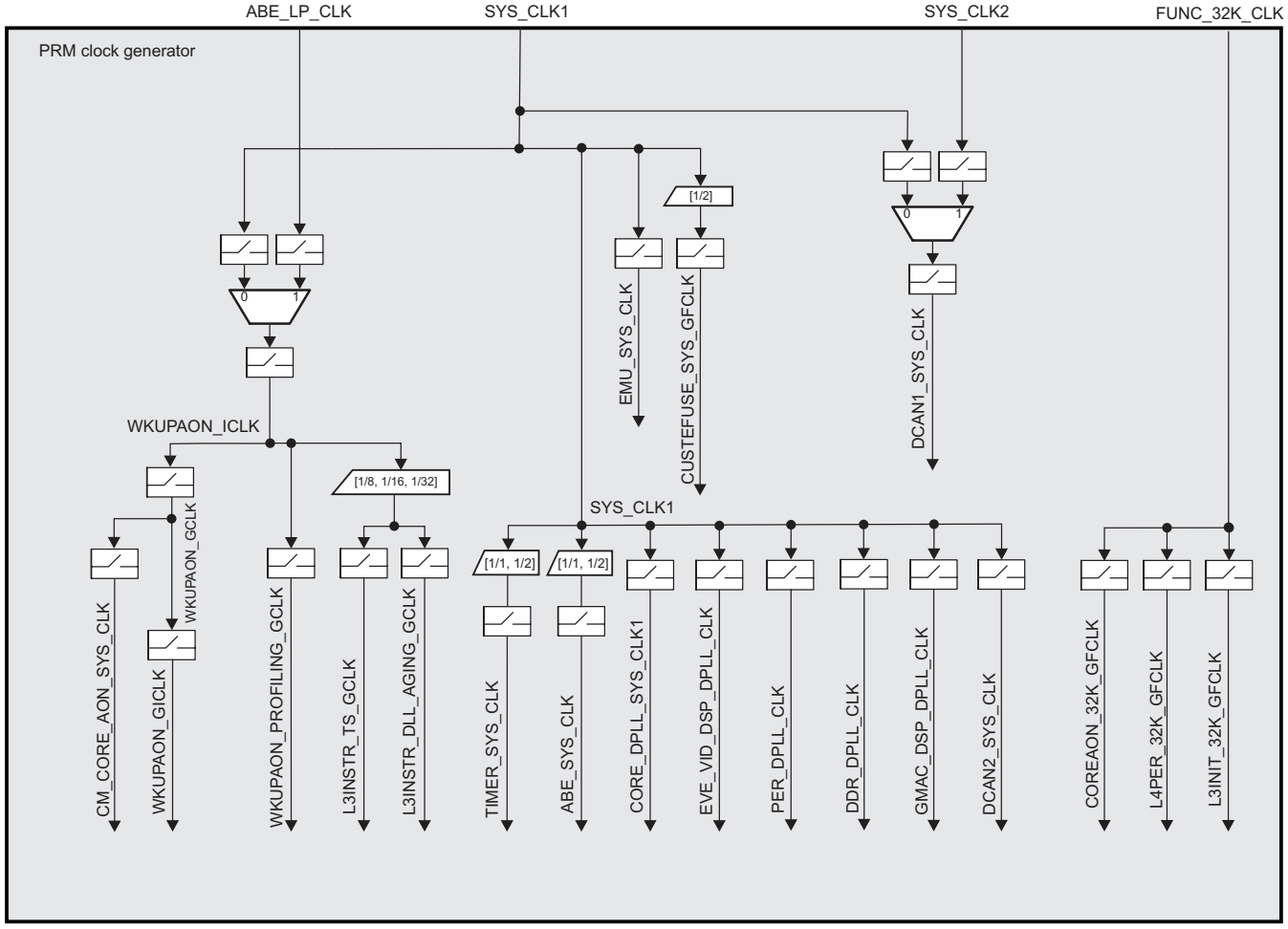


Table 3-37 identifies controls for clock dividers or muxes in the PRM.

Table 3-37. PRM Clock Division and Muxing Control

Divider/Mux	Control Bit Field
Mux WKUPAON_ICLK	CM_CLKSEL_WKUPAON[0] CLKSEL
Mux DCAN1_SYS_CLK	CM_WKUPAON_DCAN1_CLKCTRL[24] CLKSEL
Divider ABE_SYS_CLK	CM_CLKSEL_ABE_SYS[0] CLKSEL
Divider TIMER_SYS_CLK	CM_CLKSEL_TIMER_SYS[0] CLKSEL
Divider L3INSTR_TS_GCLK and L3INSTR_DLL_AGING_GCLK	CM_L3INSTR_CTRL_MODULE_BANDGAP_CLKCTRL[25:24] CLKSEL

NOTE: For clock signals control (gating/ungating management), see Section 3.1.1.1, *Clock Management*.

The PRM provides a 32-kHz gated and ungated clock for use by portions of the PD_WKUPAON power domain and some peripherals outside the PD_WKUPAON power domain.

It also provides the system clock to the DSS, a gated and buffered version to the WKUPAON_GICLK interconnect clock, and the DPLLs controlled by the PRCM module.

3.6.3.2 CM Clock Source

The clock manager (CM) is primarily responsible for generating interface and functional clocks from the internal clocks provided by DPLL_CORE and DPLL_PER. The CM is physically divided into two independent entities: CM_CORE_AON, which is placed in the PD_COREAON always-on power domain, and CM_CORE, which is placed in the PD_CORE switchable power domain. The split is done to provide control over various entities, such as modules, DPLLs, and clocks, during low-power use case scenarios when the PD_CORE power domain can be switched to RETENTION state.

3.6.3.2.1 CM_CORE_AON Clock Generator

CM_CORE_AON receives system clocks (SYS_CLK1 and SYS_CLK2) from the PRM, which serve as its functional clocks. CM_CORE_AON provides a gated clock to

- PD_CAM
- PD_CORE
- PD_CUSTEFUSE
- PD_DSP1
- PD_DSP2
- PD_DSS
- PD_EMU
- PD_EVE
- PD_IPU
- PD_L3INIT
- PD_ISS
- PD_L4PER
- PD_WKUPAON
- PD_COREAON
- PD_MMAON

and DPLLs:

- DPLL_DDR
- DPLL_GMAC_DSP
- DPLL_EVE_VID_DSP
- DPLL_PER
- DPLL_CORE

and their associated HSDividers.

Figure 3-30 shows the various functional and interface clocks generated by CM_CORE_AON.

Figure 3-30. CM_CORE_AON Overview (a)

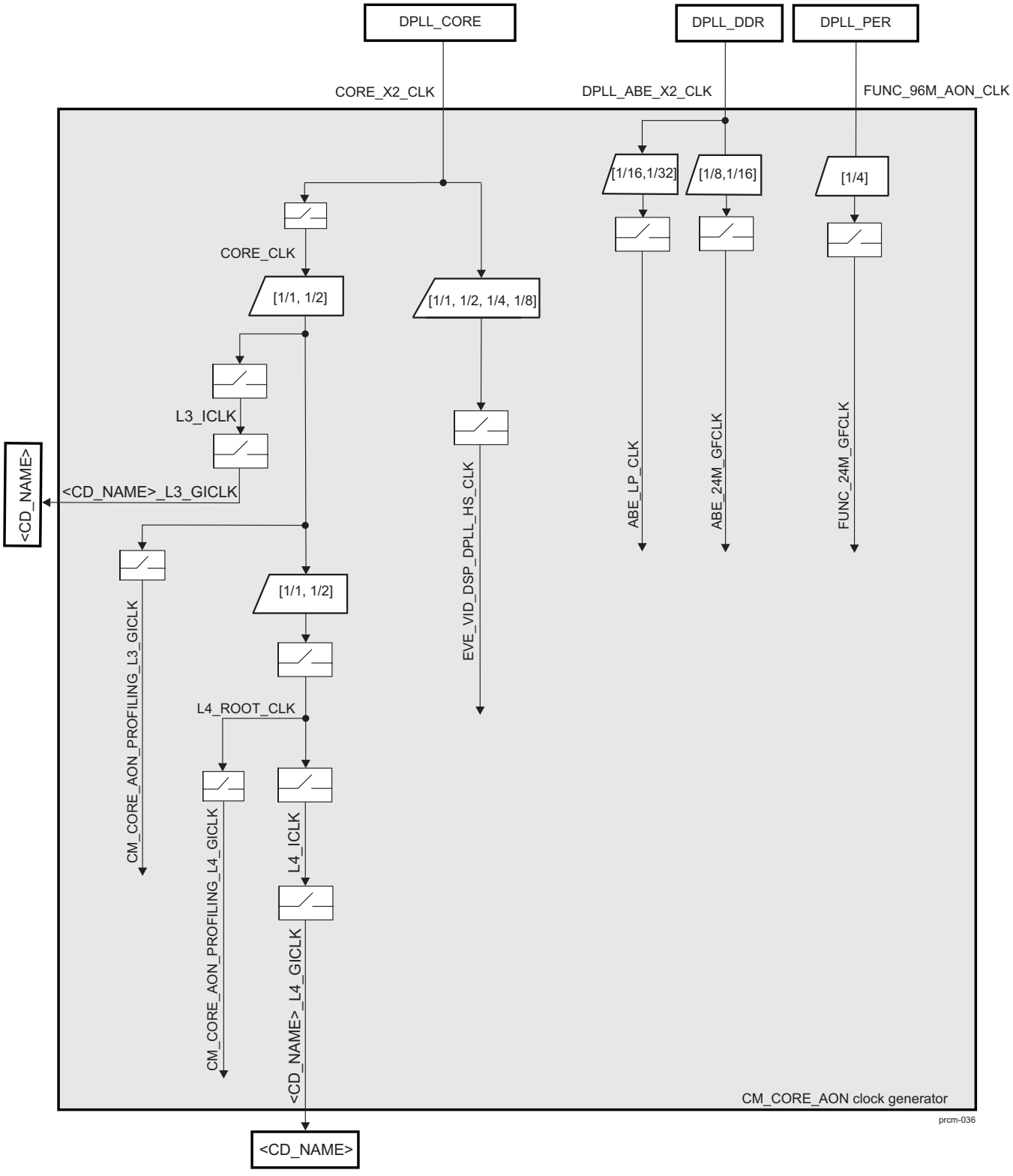


Table 3-38 identifies controls for clock dividers or muxes in the CM_CORE_AON (a) clock source.

Table 3-38. CM_CORE_AON (a) Clock Division and Muxing Control

Divider/Mux	Control Bit Field
Divider L3_ICLK	CM_CLKSEL_CORE[4] CLKSEL_L3
Divider L4_ROOT_CLK	CM_CLKSEL_CORE[8] CLKSEL_L4
Divider EVE_VID_DSP_DPLL_HS_CLK	CM_BYPCLK_DPLL_DSP[1:0] CLKSEL
Divider ABE_LP_CLK	CM_CLKSEL_ABE_LP_CLK[1:0] CLKSEL
Divider ABE_24M_FCLK	CM_CLKSEL_ABE_24M[1:0] CLKSEL

Figure 3-31. CM_CORE_AON Overview (b)

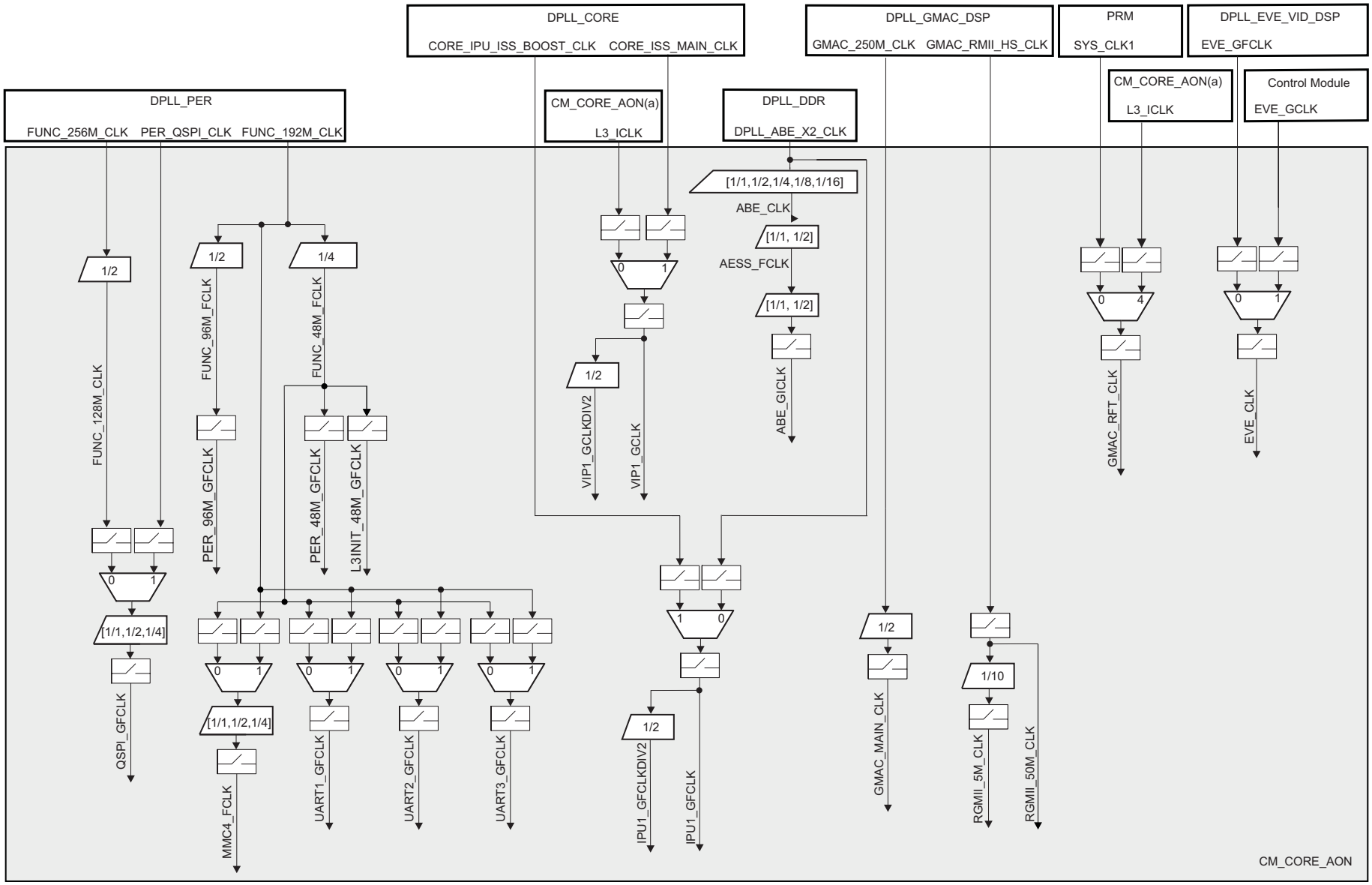


Table 3-39 identifies controls for clock dividers or muxes in the CM_CORE_AON (b) clock source.

Table 3-39. CM_CORE_AON (b) Clock Division and Muxing Control

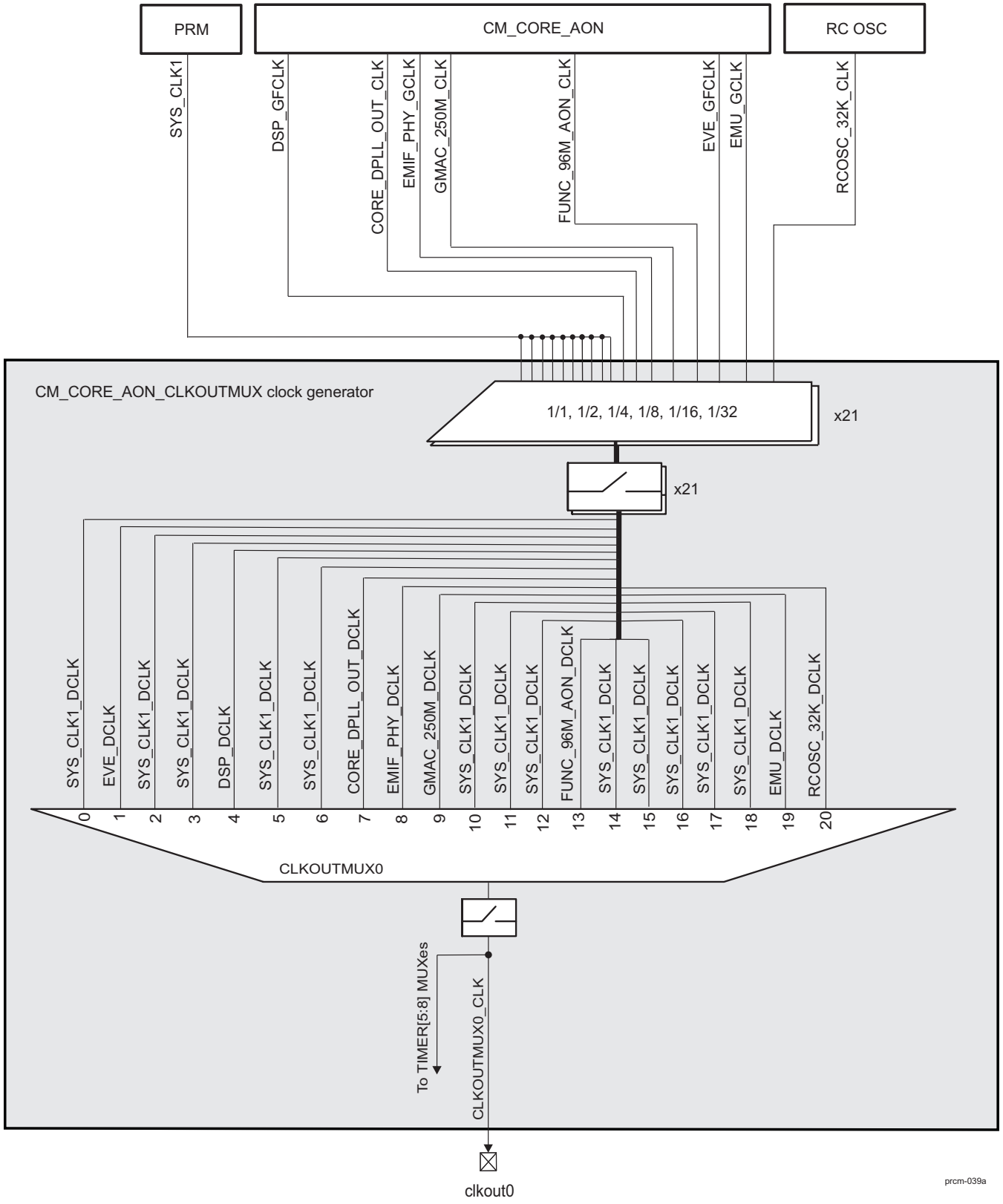
Divider/Mux/Switch	Control Bit Field
Mux MMC4_FCLK	CM_L4PER_MMC4_CLKCTRL[24] CLKSEL_MUX
Divider MMC4_FCLK	CM_L4PER_MMC4_CLKCTRL[26:25] CLKSEL_DIV
Mux UART1_GFCLK	CM_L4PER_UART1_CLKCTRL[24] CLKSEL
Mux UART2_GFCLK	CM_L4PER_UART2_CLKCTRL[24] CLKSEL
Mux UART3_GFCLK	CM_L4PER_UART3_CLKCTRL[24] CLKSEL
Mux QSPI_GFCLK	CM_L4PER2_QSPI_CLKCTRL[24] CLKSEL_SOURCE
Divider QSPI_GFCLK	CM_L4PER2_QSPI_CLKCTRL[26:25] CLKSEL_DIV
Mux VIP1_GCLK	CM_CAM_VIP1_CLKCTRL[24] CLKSEL
Divider ABE_CLK	CM_CLKSEL_ABE_CLK_DIV[2:0] CLKSEL
Divider AESS_FCLK	CM_CLKSEL_AESS_FCLK_DIV[0] CLKSEL
Divider ABE_GICLK	CM_CLKSEL_ABE_GICLK_DIV[0] CLKSEL
Mux IPU1_GFCLK	CM_IPU1_IPU1_CLKCTRL[24] CLKSEL
Mux GMAC_RFT_CLK	CM_GMAC_GMAC_CLKCTRL[27:25] CLKSEL_RFT
Mux EVE_CLK	CM_CLKSEL_EVE_CLK[0] CLKSEL

NOTE: For clock signals control (gating/ungating management), see Section 3.1.1.1, *Clock Management*.

3.6.3.2.2 CM_CORE_AON_CLKOUTMUX Overview

Figure 3-32 is an overview of part of CM_CORE_AON_CLKOUTMUX containing the CLKOUTMUX0 mux.

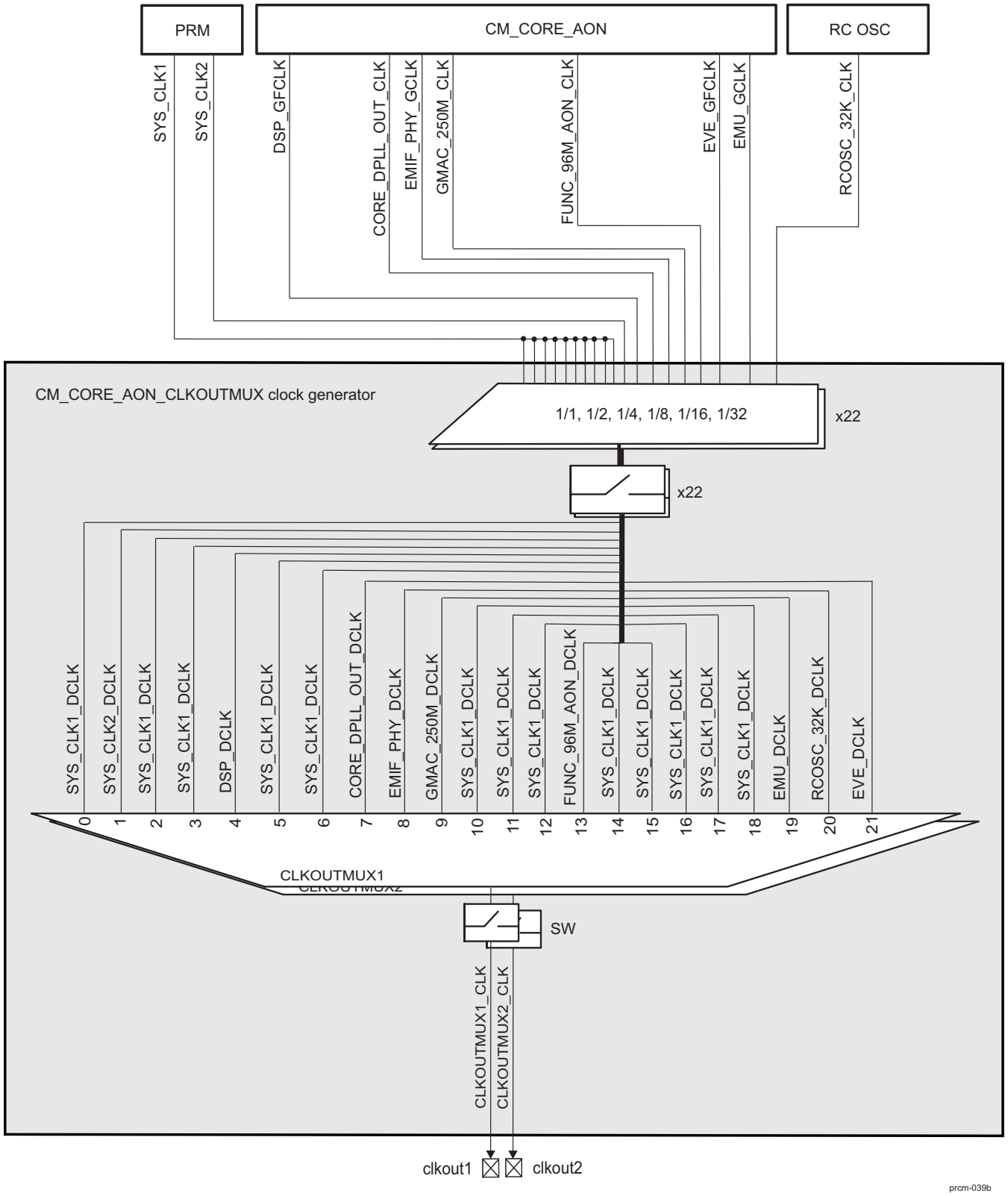
Figure 3-32. CM_CORE_AON_CLKOUTMUX Clock Manager Overview (CLKOUTMUX0)



NOTE: The RCOSC_32K_CLK clock, provided by the On-die 32K RC OSC is not accurate 32KHz clock. The frequency may vary with temperature and silicon characteristics.

Figure 3-33 is an overview of part of CM_CORE_AON_CLKOUTMUX containing the CLKOUTMUX1 and CLKOUTMUX2 muxes.

Figure 3-33. CM_CORE_AON_CLKOUTMUX Clock Manager Overview (CLKOUTMUX1 and CLKOUTMUX2)



NOTE: The RCOSC_32K_CLK clock, provided by the On-die 32K RC OSC is not accurate 32KHz clock. The frequency may vary with temperature and silicon characteristics.

Table 3-40 identifies controls for clock dividers or muxes in the CM_CORE_AON_CLKOUTMUX.

Table 3-40. CM_CORE_AON_CLKOUTMUX Clock Division and Muxing Control

Divider/Mux	Control Bit Field
Mux CLKOUTMUX0_CLK	CM_CLKSEL_CLKOUTMUX0[4:0] CLKSEL
Mux CLKOUTMUX1_CLK	CM_CLKSEL_CLKOUTMUX1[4:0] CLKSEL
Mux CLKOUTMUX2_CLK	CM_CLKSEL_CLKOUTMUX2[4:0] CLKSEL
Divider SYS_CLK1_DCLK (0)	CM_CLKSEL_SYS_CLK1_CLKOUTMUX[2:0] CLKSEL
Divider EVE_DCLK	CM_CLKSEL_EVE_GFCLK_CLKOUTMUX[2:0] CLKSEL
Divider SYS_CLK1_DCLK (2)	CM_CLKSEL_PER_ABE_X1_CLK_CLKOUTMUX[2:0] CLKSEL
Divider SYS_CLK1_DCLK (3)	CM_CLKSEL_MPU_GCLK_CLKOUTMUX[2:0] CLKSEL
Divider DSP_DCLK	CM_CLKSEL_DSP_GFCLK_CLKOUTMUX[2:0] CLKSEL
Divider SYS_CLK1_DCLK (5)	CM_CLKSEL_IVA_GCLK_CLKOUTMUX[2:0] CLKSEL
Divider SYS_CLK1_DCLK (6)	CM_CLKSEL_GPU_GCLK_CLKOUTMUX[2:0] CLKSEL
Divider CORE_DPLL_OUT_DCLK	CM_CLKSEL_CORE_DPLL_OUT_CLK_CLKOUTMUX[2:0] CLKSEL
Divider EMIF_PHY_DCLK	CM_CLKSEL_EMIF_PHY_GCLK_CLKOUTMUX[2:0] CLKSEL
Divider GMAC_250M_DCLK	CM_CLKSEL_GMAC_250M_CLK_CLKOUTMUX[2:0] CLKSEL
Divider SYS_CLK1_DCLK (10)	CM_CLKSEL_VIDEO2_CLK_CLKOUTMUX[2:0] CLKSEL
Divider SYS_CLK1_DCLK (11)	CM_CLKSEL_VIDEO1_CLK_CLKOUTMUX[2:0] CLKSEL
Divider SYS_CLK1_DCLK (12)	CM_CLKSEL_HDMI_CLK_CLKOUTMUX[2:0] CLKSEL
Divider FUNC_96M_AON_DCLK	CM_CLKSEL_FUNC_96M_AON_CLK_CLKOUTMUX[2:0] CLKSEL
Divider SYS_CLK1_DCLK (14)	CM_CLKSEL_L3INIT_480M_GFCLK_CLKOUTMUX[2:0] CLKSEL
Divider SYS_CLK1_DCLK (15)	CM_CLKSEL_USB_OTG_CLK_CLKOUTMUX[2:0] CLKSEL
Divider SYS_CLK1_DCLK (16)	CM_CLKSEL_SATA_CLK_CLKOUTMUX[2:0] CLKSEL
Divider SYS_CLK1_DCLK (17)	CM_CLKSEL_PCIE2_CLK_CLKOUTMUX[2:0] CLKSEL
Divider SYS_CLK1_DCLK (18)	CM_CLKSEL_PCIE1_CLK_CLKOUTMUX[2:0] CLKSEL
Divider EMU_DCLK	CM_CLKSEL_EMU_CLK_CLKOUTMUX[2:0] CLKSEL
Divider RCOSC_32K_DCLK	CM_CLKSEL_SECURE_32K_CLK_CLKOUTMUX[2:0] CLKSEL
Enable CLKOUTMUX0_CLK	At least one of CM_IPU_TIMER5_CLKCTRL, CM_IPU_TIMER6_CLKCTRL, CM_IPU_TIMER7_CLKCTRL, or CM_IPU_TIMER8_CLKCTRL[27:24] CLKSEL set to 0xB
Enable CLKOUTMUX1_CLK	CM_COREAON_DUMMY_MODULE1_CLKCTRL[8] OPTFCLKEN_CLKOUTMUX1_CLK
Enable CLKOUTMUX2_CLK	CM_COREAON_DUMMY_MODULE2_CLKCTRL[8] OPTFCLKEN_CLKOUTMUX2_CLK

3.6.3.2.3 CM_CORE_AON_TIMER Overview

Figure 3-34 and Figure 3-35 are an overview of the CM_CORE_AON_TIMER related to the device timers.

Figure 3-34. CM_CORE_AON_TIMER1 Clock Manager Overview

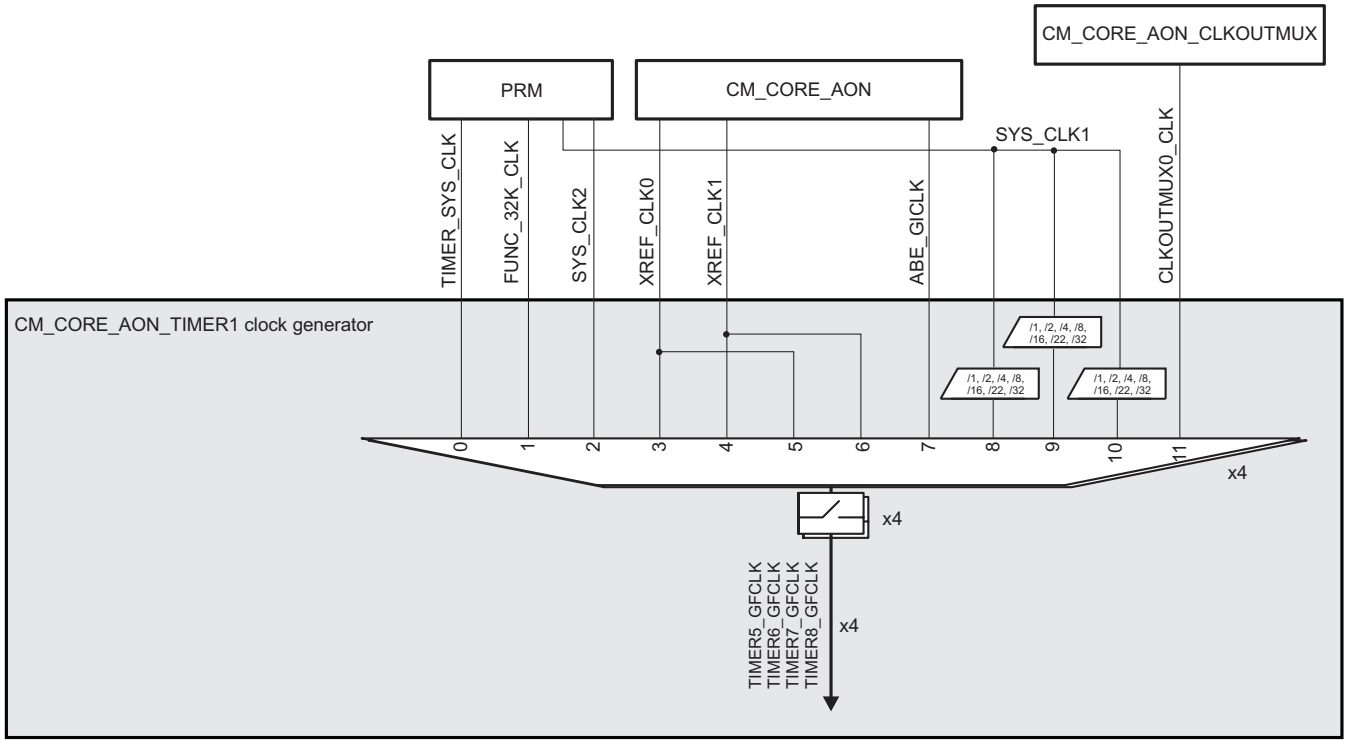


Figure 3-35. CM_CORE_AON_TIMER2 Clock Manager Overview

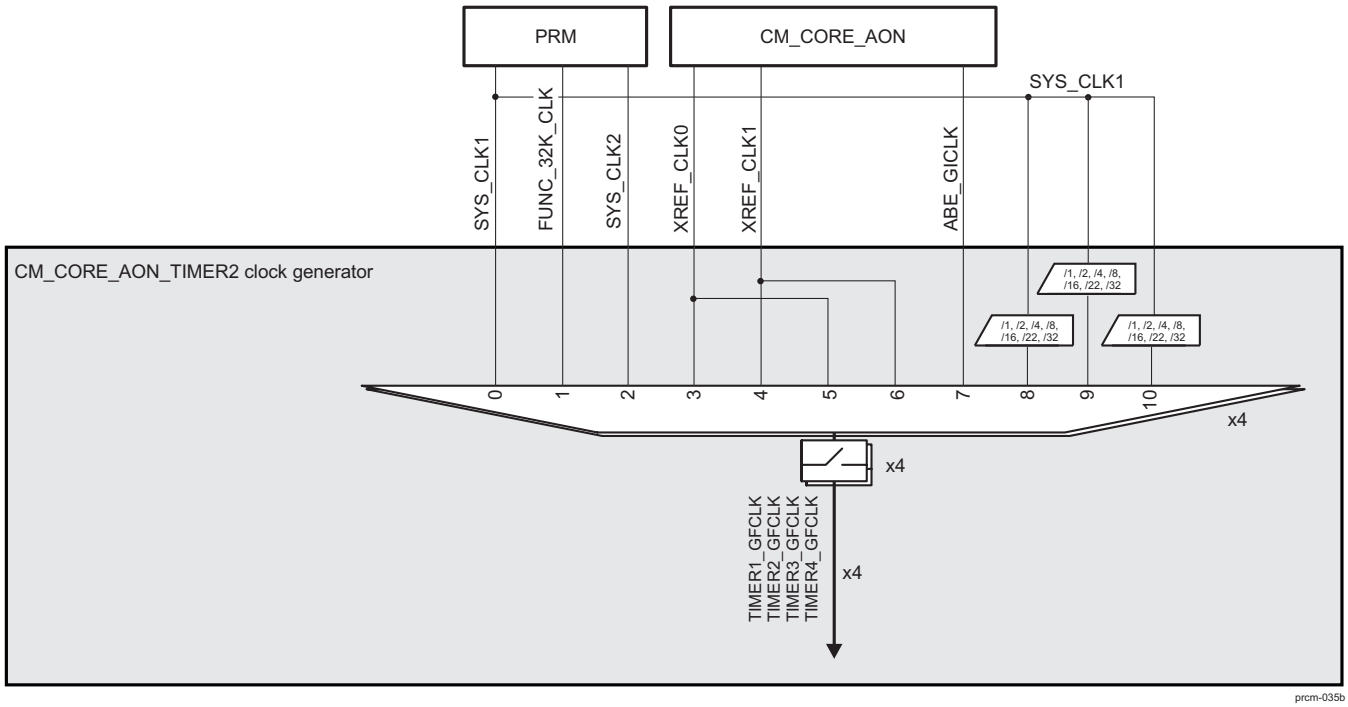


Table 3-41 identifies controls for clock dividers or muxes in the CM_CORE_AON_TIMER.

Table 3-41. CM_CORE_AON_TIMER Clock Division and Muxing Control

Divider/Mux	Control Bit Field
Mux TIMER1_GFCLK	CM_WKUPAON_TIMER1_CLKCTRL[27:24] CLKSEL
Mux TIMER2_GFCLK	CM_L4PER_TIMER2_CLKCTRL[27:24] CLKSEL
Mux TIMER3_GFCLK	CM_L4PER_TIMER3_CLKCTRL[27:24] CLKSEL
Mux TIMER4_GFCLK	CM_L4PER_TIMER4_CLKCTRL[27:24] CLKSEL
Mux TIMER5_GFCLK	CM_IPU_TIMER5_CLKCTRL[27:24] CLKSEL
Mux TIMER6_GFCLK	CM_IPU_TIMER6_CLKCTRL[27:24] CLKSEL
Mux TIMER7_GFCLK	CM_IPU_TIMER7_CLKCTRL[27:24] CLKSEL
Mux TIMER8_GFCLK	CM_IPU_TIMER8_CLKCTRL[27:24] CLKSEL
Divider SYS_CLK1_DCLK (8)	CM_CLKSEL_VIDEO1_TIMER[2:0] CLKSEL
Divider SYS_CLK1_DCLK (9)	CM_CLKSEL_VIDEO2_TIMER[2:0] CLKSEL
Divider SYS_CLK1_DCLK (10)	CM_CLKSEL_HDMI_TIMER[2:0] CLKSEL

NOTE: For clock signals control (gating/ungating management), see Section 3.1.1.1, *Clock Management*.

3.6.3.2.4 CM_CORE_AON_MCASP1 Overview

Figure 3-36 and Figure 3-37 are an overview of the CM_CORE_AON_MCASP related to McASP.

Figure 3-36. CM_CORE_AON_MCASP Clock Manager Overview

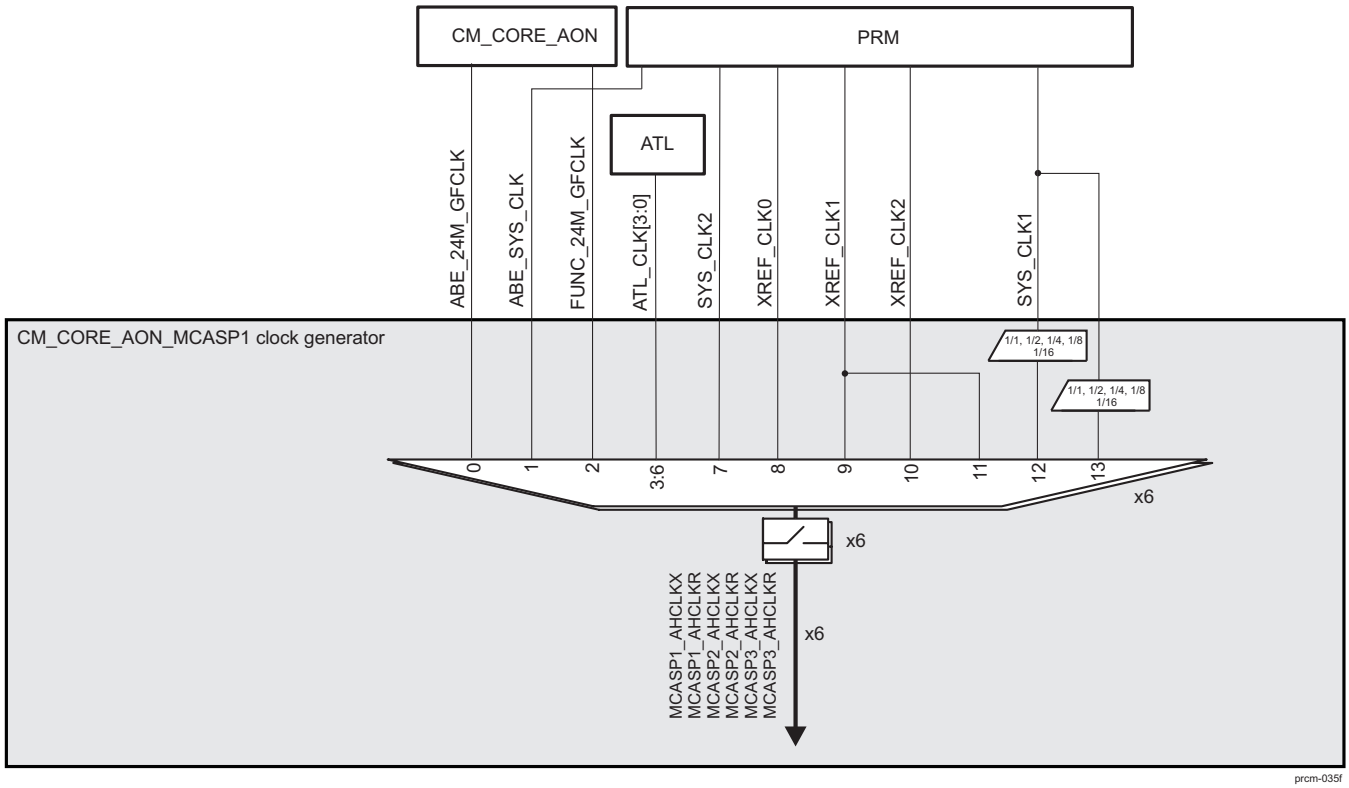


Figure 3-37. CM_CORE_AON_MCASP Clock Manager Overview (AUXCLK)

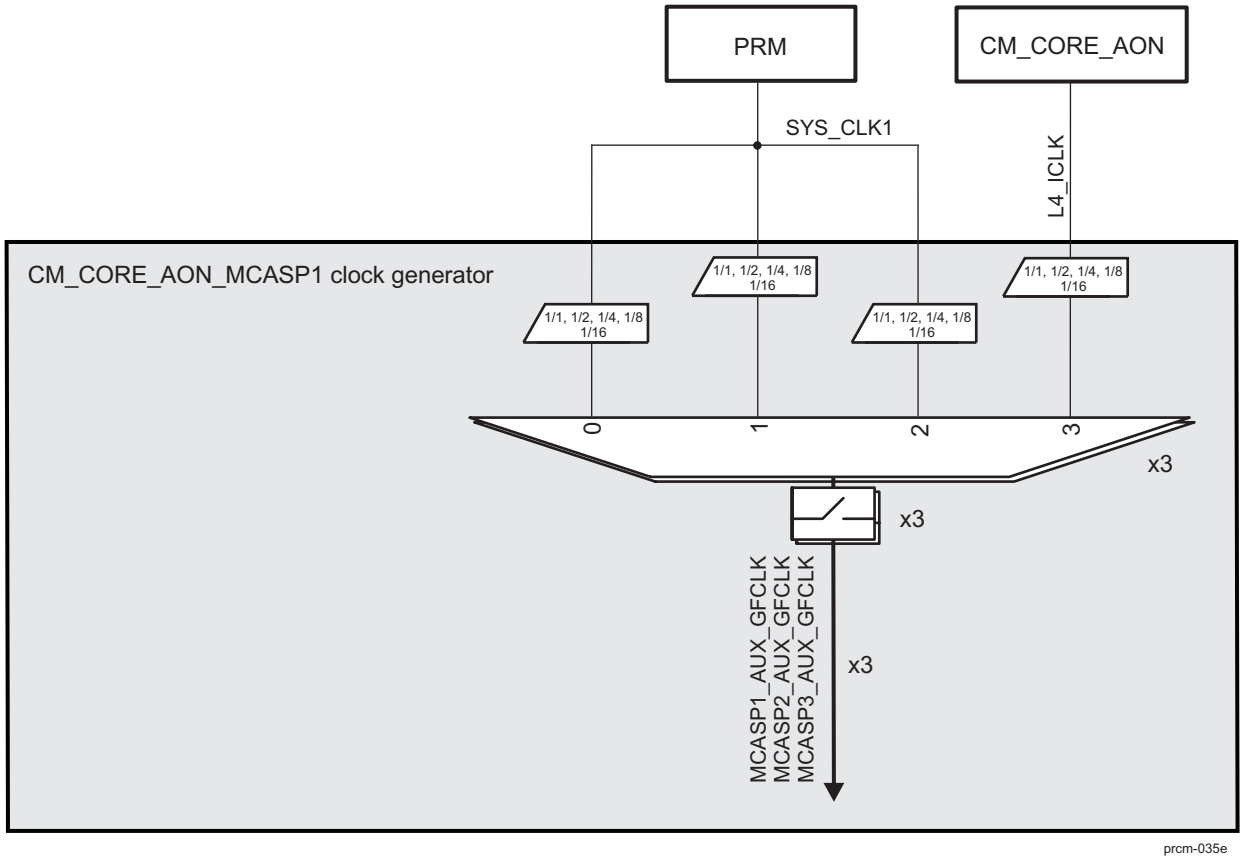


Table 3-42 identifies controls for clock dividers or muxes in the CM_CORE_AON_MCASP.

Table 3-42. CM_CORE_AON_MCASP Clock Division and Muxing Control

Divider/Mux	Control Bit Field
Mux MCASP1_AHCLKX	CM_IPU_MCASP1_CLKCTRL[27:24] CLKSEL_AHCLKX
Mux MCASP1_AHCLKR	CM_IPU_MCASP1_CLKCTRL[31:28] CLKSEL_AHCLKR
Mux MCASP2_AHCLKX	CM_L4PER2_MCASP4_CLKCTRL[27:24] CLKSEL_AHCLKX
Mux MCASP2_AHCLKR	CM_L4PER2_MCASP6_CLKCTRL[27:24] CLKSEL_AHCLKR
Mux MCASP3_AHCLKX	CM_L4PER2_MCASP5_CLKCTRL[27:24] CLKSEL_AHCLKX
Mux MCASP3_AHCLKR	CM_L4PER2_MCASP7_CLKCTRL[27:24] CLKSEL_AHCLKR
Divider SYS_CLK1_DCLK (12)	CM_CLKSEL_MLB_MCASP[2:0] CLKSEL
Divider SYS_CLK1_DCLK (13)	CM_CLKSEL_MLBP_MCASP[2:0] CLKSEL
Mux MCASP1_AUX_GFCLK	CM_IPU_MCASP1_CLKCTRL[23:22] CLKSEL_AUX_CLK
Mux MCASP2_AUX_GFCLK	CM_L4PER2_MCASP4_CLKCTRL[23:22] CLKSEL_AUX_CLK
Mux MCASP3_AUX_GFCLK	CM_L4PER2_MCASP5_CLKCTRL[23:22] CLKSEL_AUX_CLK
Divider L4_ICLK_DCLK	CM_CLKSEL_PER_ABE_X1_GFCLK_MCASP_AUX[2:0] CLKSEL
Divider SYS_CLK1_DCLK (1)	CM_CLKSEL_VIDEO1_MCASP_AUX[2:0] CLKSEL
Divider SYS_CLK1_DCLK (2)	CM_CLKSEL_VIDEO2_MCASP_AUX[2:0] CLKSEL
Divider SYS_CLK1_DCLK (3)	CM_CLKSEL_HDMI_MCASP_AUX[2:0] CLKSEL

NOTE: For clock signals control (gating/ungating management), see Section 3.1.1.1, *Clock Management*.

3.6.3.3 Clock Control in Control Module

Part of clock mux and division is exported to device Control Module as shown in Figure 3-38

Figure 3-38. Clock Control in Control Module Overview

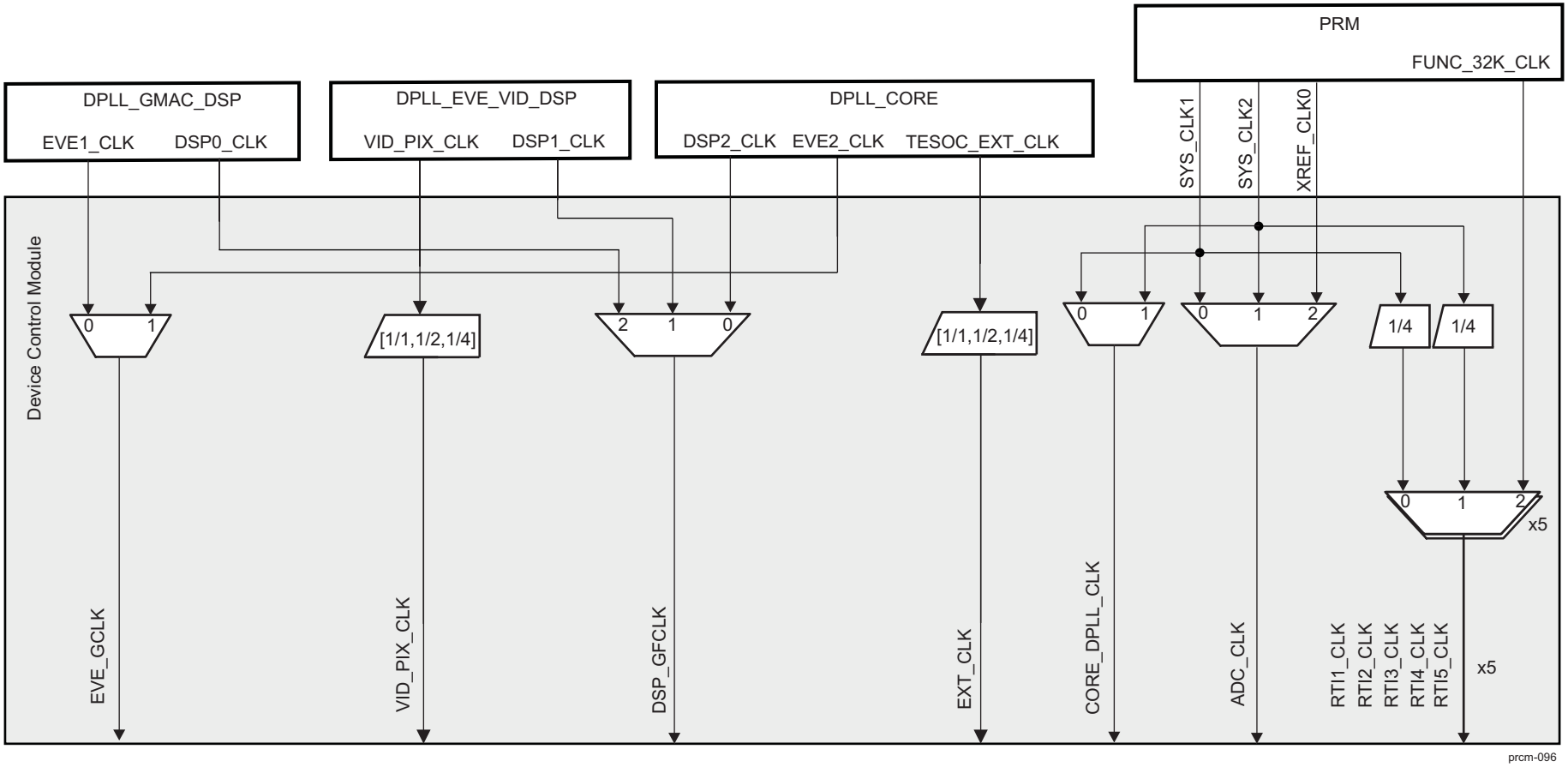


Table 3-43 identifies controls for clock dividers or muxes in the Control Module.

Table 3-43. Control Module Clock Division and Muxing Control

Divider/Mux	Control Bit Field
Mux EVE_GCLK	CTRL_CORE_PRCM_CLKSEL_CONTROL[28] EVE_CLKSEL
Mux DSP_GFCLK	CTRL_CORE_PRCM_CLKSEL_CONTROL[25:24] DSP_CLKSEL
Mux ADC_CLK	CTRL_CORE_PRCM_CLKSEL_CONTROL[21:20] ADC_CLKSEL
Mux RTI1_CLK	CTRL_CORE_PRCM_CLKSEL_CONTROL[17:16] RTI1_CLKSEL
Mux RTI2_CLK	CTRL_CORE_PRCM_CLKSEL_CONTROL[13:12] RTI2_CLKSEL
Mux RTI3_CLK	CTRL_CORE_PRCM_CLKSEL_CONTROL[9:8] RTI3_CLKSEL
Mux RTI4_CLK	CTRL_CORE_PRCM_CLKSEL_CONTROL[5:4] RTI4_CLKSEL
Mux RTI5_CLK	CTRL_CORE_PRCM_CLKSEL_CONTROL[1:0] RTI5_CLKSEL
Divider VID_PIX_CLK	CTRL_CORE_PRCM_CLKDIV_CONTROL2[26:25] VID_PIX_CLK_DIV
Divider TESOC_EXT_CLK	CTRL_CORE_PRCM_CLKDIV_CONTROL2[10:9] TESOC_EXT_CLK_DIV
Divider enable VID_PIX_CLK	CTRL_CORE_PRCM_CLKDIV_CONTROL2[23] VID_PIX_CLK_HSDIV_EN
Divider enable TESOC_EXT_CLK	CTRL_CORE_PRCM_CLKDIV_CONTROL2[7] TESOC_HSDIV_EN
Divider enable MCAN_CLK	CTRL_CORE_SMA_SW_17[7] MCAN_CLK_HSDIV_EN
Mux CORE_DPLL_CLK	CTRL_CORE_SMA_SW_16[0] CORE_DPLL_INPUT_CLK_SELECTION

3.6.3.3.1 Programming Guide For Control Module

DPLL management and clock control is partly exported to the device Control Module. Below are listed the software guidelines for functions managed by Control Module.

1. All control module registers relating to clock selection and clock division must be programmed prior to locking the corresponding DPLL and enabling the corresponding module in PRCM
2. IPU clock selection in PRCM defaults to selecting CORE_IPU_ISS_BOOST_CLK (from DPLL_CORE H22) as the source. If desired to select DPLL_ABE_X2_CLK (from H12 of DPLL_DDR), the IPU clock mux select in CM_IPU1_IPU1_CLKCTRL[24] CLKSEL must be programmed to 0
3. TENABLEDIV is required to be pulsed for latching HSDIVIDER values
    1. for HSDIVIDER values coming from PRCM, TENABLEDIV is driven by respective DPLL sources.
    2. for HSDIVIDER values coming from control module (DBG_TRC_EXPT_CLK_DIV, DBG_ATB_CLK_DIV, TESOC_HSDIV, DBG_STM_EXPT_CLK_DIV, VID_PIX_CLK_HSDIV, and MCAN_CLK_HSDIV), TENABLEDIV has an override control from the control module registers CTRL_CORE_PRCM_CLKDIV_CONTROL1, CTRL_CORE_PRCM_CLKDIV_CONTROL2, and CTRL_CORE_SMA_SW_17.
4. For Debug Subsystem clocks (ATB_CLK, STM_EXPT_CLK, and TRCEXPT_CLK), that get the HSDIVIDER values from control module, the programming sequence is as follows:
    1. ROM code locks DPLL_CORE and/or DPLL_PER.
    2. To program HSDIVIDERs for DebugSS clocks, the HSDIVIDER values must be programmed first in the control module.
    3. Corresponding TENABLEDIV_OVR must be programmed to 1 (TENABLEDIV controlled from the control module)
    4. Corresponding TENABLEDIV_CTRL must be programmed to 1 and then 0 again.
5. For TESOC_EXT_CLK and VID_PIX_CLK clocks, clock enable bits must be set. That is, TESOC_HSDIV_EN and VID_PIX_CLK_HSDIV_EN in CTRL_CORE_PRCM_CLKDIV_CONTROL2 register.
6. CTRLCLK of CAMERARX is driven by FUNC_96M_FCLK coming from DPLL_PER. This implies either of the following modules has to be enabled: I2C1 or I2C2.
7. The following sequence must be followed to select SYS_CLK2 as input clock to DPLL_CORE during run time:
    1. Ensure DPLL_CORE is in BYPASS mode
    2. Set CM_CLKSEL_DPLL_CORE[23] DPLL_BYP_CLKSEL to 1 to select the BYP_CLK_M2 input of DPLL

3. Switch the SYS_CLK mux to select SYS_CLK2. That is, set bit CTRL_CORE_SMA_SW_16[0] CORE_DPLL_INPUT_CLK_SELECTION to 1
4. Program new DPLL parameters to match SYS_CLK2 clock speed
5. Switch DPLL_CORE input back to REF_CLK. Clear bit CM_CLKSEL_DPLL_CORE[23] DPLL_BYP_CLKSEL to 0
6. Initiate the DPLL lock

3.6.3.4 Generic DPLL Overview

To generate high-frequency clocks, the device supports multiple on-chip DPLLs controlled by the PRCM and Control module.

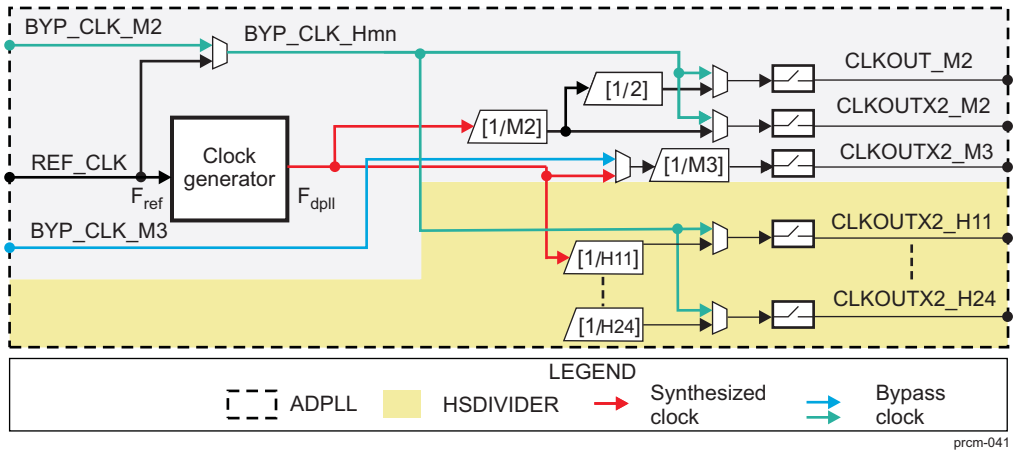
- DPLL_CORE
- DPLL_PER
- DPLL_EVE_VID_DSP
- DPLL_GMAC_DSP
- DPLL_DDR

All DPLLs support the features described in the following sections, unless otherwise identified.

3.6.3.4.1 DPLLs Output Clocks Parameters

Figure 3-39 shows the functional architecture of a generic DPLL.

Figure 3-39. Generic DPLL Functional Diagram



The DPLL has three input clocks:

- REF_CLK: Used to generate the synthesized clock but can also be used as the bypass clock for some outputs of the DPLL whenever the DPLL enters bypass mode. It is mandatory for the DPLL clock synthesis.
- BYP_CLK_M2: Selectable bypass clock for the output of an M2 post-divider (optional)
- BYP_CLK_M3: Selectable bypass clock for the output of an M3 post-divider (optional)

The DPLL provides the bypass clock used by HSDIVIDER: BYP_CLK_Hmn, which is output by the multiplexer between the BYP_CLK_M2 and REF_CLK clock signals.

The DPLL can be programmed to be locked at any frequency given by one of the following equation:

- $F_{dpll} = F_{ref} \times 2 \times M / (N + 1)$

Where:

- F_{dpll} is the DPLL lock frequency.
- F_{ref} is the REF_CLK frequency. F_{ref} is also known as CLKINP.
- M is the software-configured multiplication ratio binary value.

- N is the software-configured division ratio binary value.

NOTE: It is preferred to minimize the value for N parameter (it minimizes lock time and jitter). Then M should be chosen to provide correct frequency (with lowest delta as possible).

It internally generates three main clocks: CLKOUT_M2, CLKOUTX2_M2, and CLKOUTX2_M3 as shown in Table 3-44.

Table 3-44. CLKOUT_M2, CLKOUTX2_M2, and CLKOUTX2_M3 Frequencies With DPLL State

Output	Equation	DPLL Mode
CLKOUT_M2	F_{dpll} / (2 x M2)	Locked (typical case)
	F_{ref} or BYP_CLK_M2	Before lock or during relock
	F_{dpll} / M2	DC corrector logic is used (DCC_EN bit is set) with M2 set to 0x1.
CLKOUTX2_M2	F_{dpll} / M2	Locked (typical case)
	F_{ref} or BYP_CLK_M2	Before lock or during relock
	F_{dpll} / M2	DC corrector logic is used (DCC_EN bit is set).
CLKOUTX2_M3	F_{dpll} / M3 or BYP_CLK_M3/M3	Locked (typical case)
	0	Before lock or during relock
	F_{dpll} / M3 or BYP_CLK_M3/M3	DC corrector logic is used (DCC_EN bit is set).

Where:

- M2 is the software-configured division ratio binary value.
- M3 is the software-configured division ratio binary value.
- CLKOUT_M2 and CLKOUTX2_M2 bypass clock input can be switched when the DPLL is not in locked state by using the M2 bypass clock select control bit.
- CLKOUTX2_M3 output clock can be switched when the DPLL is in locked state by using the M3 clock select control bit.

NOTE:

- A value of 0 for M2 and M3 division ratios is not allowed. They are set to 1 after reset.
- CLKOUT_M2 is generated based on a fixed divide-by-2 ratio, except in bypass mode.

The DPLL can contain one or two HSDIVIDER modules to produce more clocks with divided ratio based on the DPLL synthesized clock frequency. HSDIVIDER1 provides four extra post-dividers from H11 to H14 (the output clocks are CLKOUTX2_H11 through CLKOUTX2_H14). HSDIVIDER2 provides four extra post-dividers from H21 through H24 (the output clocks are CLKOUTX2_H21 through CLKOUTX2_H24). The HSDIVIDER output clock frequency is given by the equations in Table 3-45.

Table 3-45. CLKOUTX2_Hmn Frequencies With DPLL State

Equation	DPLL Mode
CLKOUTX2_Hmn = F_{dpll} /Hmn	Locked
CLKOUTX2_Hmn = BYP_CLK_Hmn	Before lock or during relock

Where:

- F_{dpll} is the DPLL lock frequency.
- Hmn is the software-configured division ratio binary value.
- n is in the range from 1 to 4.

- m is equal to 1 or 2.

NOTE: Hmn division ratio is set to 1 after reset.

All clock outputs of the DPLL can be gated. The PRCM module provides the DPLL with a clock-gating control signal to enable or disable the clock. DPLL provides the PRCM module with a clock activity status signal to let the PRCM module hardware know when the clock is effectively running or effectively gated. The PRCM module provides a CM_IDLEST_DPLL_dpll_name[0] ST_DPLL_CLK status bit, which indicates the lock state or nonlock state of the DPLL.

3.6.3.4.2 Enable Control, Status, and Low-Power Operation Mode

The DPLL has a manual mode control bit field, which allows the setting of the different operating modes of the DPLL. When the DPLL is switched to lock mode, the current values of the multiplication ratio (M) and the division ratio (N) are latched in the DPLL. The DPLL then starts the lock or relock sequence to synthesize the corresponding output frequency clock.

The status of the synthesized clock output of the DPLL is represented by the CLKOUT status bit. It can be gated or active.

The DPLLs can be switched to low-power operation mode (also called LPMODE) to optimize DPLL power consumption when the input and output clock frequencies are low. This mode can be software-enabled using the low-power mode control bit of the DPLL.

It must be enabled only if both of the following operating conditions are satisfied:

- F_{ref} / (N + 1) is less than or equal to 1 MHz.
- F_{ref} ×M / (N + 1) is less than or equal to 100 MHz.

Where:

- F_{ref} is the REF_CLK frequency.
- M is the software-configured multiplication ratio binary value.
- N is the software-configured division ratio binary value.

3.6.3.4.3 DPLL Power Modes

DPLL supports several power modes. Each mode results in a tradeoff between power savings and relock time. The PRCM module allows only a few modes for each DPLL, depending on the use of the DPLL.

Table 3-46 lists the DPLL power modes.

Table 3-46. DPLL Power Modes

Power Mode	CLKOUT State	Logic Current (mA)	Analog Current (mA)	Freq Lock Time	Phase Lock Time
Low-power stop	Clock stopped	0.065 (leakage)	0.009 (leakage)	2.5 µs + 70 × (N+1)/$_{ref}$)	2.5 µs + 120 × (N+1) / F_{ref})
Fast-relock stop	Clock stopped	0.065 (leakage)	0.009 (leakage)	2.5 µs + 70 × (N+1) / F_{ref})	2.5 µs + 120 × (N+1) / F_{ref})
Low-power bypass	Bypass clock/clock stopped	0.065 (leakage)	0.009 (leakage)	2.5 µs + 70 × (N+1) / F_{ref})	2.5 µs + 120 × (N+1) / F_{ref})
Fast-relock bypass	Bypass clock/clock stopped	0.065 (leakage)	0.5 (leakage)	0.05 µs + 70 × (N+1) / F_{ref})	0.05 µs + (120 × (N+1) / F_{ref})
Lock	Synthesized clock	0.95 (active)	3 (active)	N/A	N/A

Where:

- F_{ref} is the REF_CLK frequency.

A DPLL power mode can be achieved on a software request (manual) and/or automatically (automatic), depending on the specific hardware conditions.

A DPLL can switch from one mode to another as a result of the following:

- Software-programmed transition (manual): Software configures the dedicated DPLL manual mode control feature for the next desired DPLL mode. It must ensure that the transition can be performed based on the clock activity on the device.
- Combined software-programmed and hardware-conditions-based transition (auto): This mode allows the DPLL to automatically transition to a low-power state (that is, any state other than the LOCK state) when the output clocks are gated or the destination clock domain is inactive, and to switch back to the LOCK state when the output clock is needed (that is, the clock is ungated or the clock domain becomes active). The desired low-power state for the automatic transition is configured in the dedicated Auto Mode Control parameter of the DPLL.

NOTE: With T_{ref} = 1 / F_{ref} = (N + 1) / CLKINP; (Fref = CLKINP / (N + 1)):

$T_{ref}$is the REF_CLK period.

This formula indicates that a smaller N divider value provides a smaller time for switching the clock after an M2 post-divider change.

A compromise is necessary between the clock switching latency and power consumption. Having a smaller N value:

- Requires a higher M2 post-divider value to obtain the same target frequency.
- Results in a higher DPLL lock frequency, and then higher power consumption.

NOTE:

- A manual transition can be performed from any power mode to any other power mode.
- An automatic transition can be performed from lock mode to any low-power mode.

NOTE: When the DPLL is in Low-Power bypass mode, Auto-idle mode is disabled and no clock is requested, the DPLL makes a transition to Low-power stop mode.

3.6.3.4.4 DPLL Recalibration

Each time the DPLL is reset or performs a lock sequence (following a change in the value of multiplier M or divider N), it performs a recalibration of the output frequency, based on voltage and temperature conditions. In lock mode, the DPLL maintains a steady lock frequency output by compensating for voltage and temperature changes within a certain range. However, if the voltage or temperature drifts outside the range or shows a significant or fast change, the DPLL may not be able to track and compensate it. It would need a recalibration, which is signaled by assertion of a recalibration flag.

NOTE:

- The recalibration mechanism is active only while the DPLL is in lock mode. When the DPLL is in off or bypass mode (low-power or fast-relock), it does not assert the recalibration flag.
- If the DPLL drifts out of the operating range limits while not locked, and then when it tries to relock, it fails to lock within the normal delay and recalibrates automatically before eventually locking. The only difference between this case and a standard relock is the recalibration delay.

During recalibration, the DPLL loses lock and output clock switches to the bypass clock.

The DPLL can automatically start recalibration when the recalibration flag is asserted, or recalibration can be triggered by software control. The trigger setting of the recalibration can be configured by the corresponding registers of the DPLL in the PRCM module. The software-controlled recalibration mode is selected by default.

Software-controlled recalibration: The DPLL continues its tracking mechanism as long as the recalibration is not triggered by software (that is, by enabling the recalibration-enable control parameter). If the DPLL reaches upper or lower bounds of the DCO control code and software has still not triggered recalibration, the DPLL stops its tracking mechanism. The output clock remains active, but frequency and jitter are not ensured to meet the requirement.

Automatic recalibration: The DPLL immediately starts the recalibration as soon as the recalibration flag is asserted.

NOTE: Automatic recalibration of the DPLL can start at any time. While relocking, the DPLL switches to bypass mode, which introduces a frequency change. For modules that are sensitive to frequency change while operating, this can introduce operational instability. For example, the external memory EMIF controller is sensitive to a frequency change on the DPLL because its embedded DLL relocks on a frequency change. Any EMIF access during this DLL relock period can be corrupted. It is, therefore, important to stall EMIF access during DPLL recalibration.

To allow software to recalibrate the DPLL at the correct time depending on the device activity, the PRCM module can generate a wake-up event on the processor power domain, followed by an interrupt on the processor subsystem when the DPLL recalibration flag is asserted.

Table 3-47 lists the DPLL recalibration and control parameters.

Table 3-47. DPLL Recalibration Control Parameters

Parameter	Register	Description
Recalibration-enable control	CM_CLKMODE_DPLL_<module>[8] DPLL_DRIFTGUARD_EN	Enable/disable the DPLL automatic recalibration feature.
Recalibration-interrupt mask control	PRM_IRQENABLE_MPU PRM_IRQENABLE_IPU1 PRM_IRQENABLE_DSP1 PRM_IRQENABLE_DSP2	Mask/unmask the DPLL recalibration interrupt to processor.
Recalibration-interrupt status	PRM_IRQSTATUS_MPU_2PRM_IRQSTATUS_MPU PRM_IRQSTATUS_IPU1 PRM_IRQSTATUS_DSP1 PRM_IRQSTATUS_DSP2	Status of the DPLL recalibration interrupt to processor

3.6.3.4.5 DPLL Output Power Down

The DCO clock LDO (DCOCLKLDO) of the DPLL can be powered down if all output dividers of the DPLL are powered down. The PRCM module automatically reenables the power to the LDO when an output divider is powered up or the DPLL switches to bypass mode.

Table 3-48. DPLL Power-Down Control Parameters

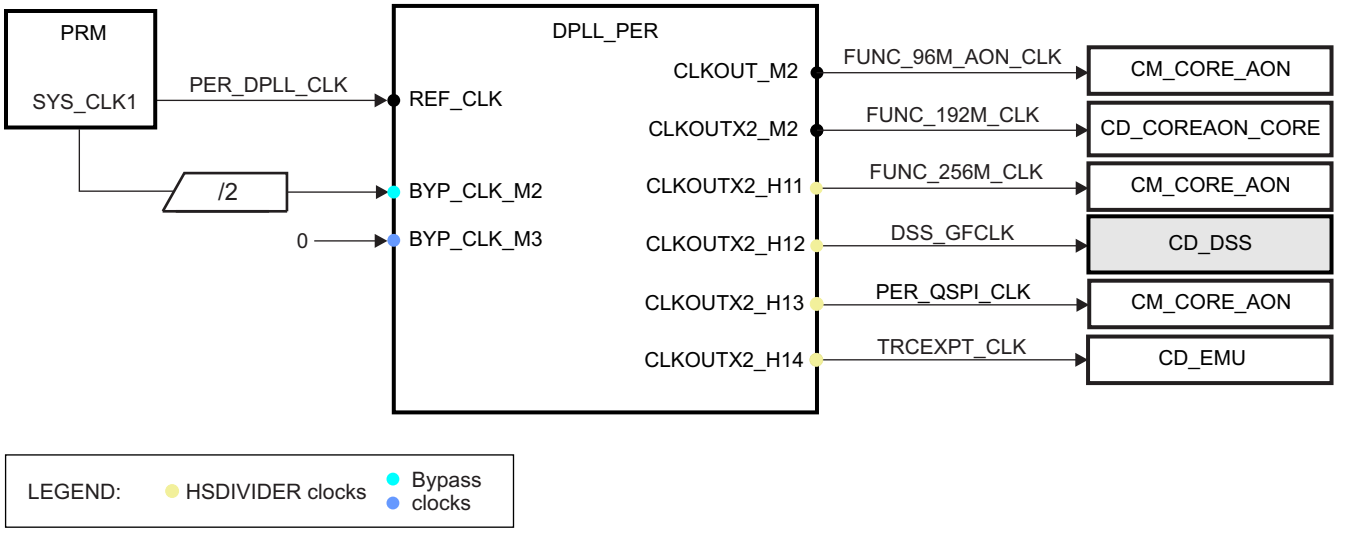
Parameter	Description
DCO clock LDO power down control	Enable/disable automatic power-down feature if all output dividers are powered down.

3.6.3.5 DPLL_PER Description

3.6.3.5.1 DPLL_PER Overview

Figure 3-40 is an overview of the DPLL. For a functional overview of a generic DPLL module, see Section 3.6.3.4, *Generic DPLL Overview*.

Figure 3-40. DPLL_PER Overview



3.6.3.5.2 DPLL_PER Synthesized Clock Parameters

This section describes the clock synthesis and clock out divider parameters of the DPLL. For an explanation of the clock synthesis and output divider parameters of the DPLL module, see Section 3.6.3.4, *Generic DPLL Overview*

Table 3-49 lists the clock synthesis parameters of the DPLL.

Table 3-49. DPLL_PER Clock Synthesis Parameters

Parameter Name	Control Bit Field
M	CM_CLKSEL_DPLL_PER[18:8] DPLL_MULT
N	CM_CLKSEL_DPLL_PER[6:0] DPLL_DIV

Table 3-50 lists the clock output divider parameters of the DPLL.

Table 3-50. DPLL_PER Clock Output Parameters

Clock Output/Divider	Parameter Name	Control/Status Bit Field
CLKOUT_M2	Status	CM_DIV_M2_DPLL_PER[9] CLKST
CLKOUT_M2	Divider control	CM_DIV_M2_DPLL_PER[4:0] DIVHS
CLKOUTX2_M2	Status	CM_DIV_M2_DPLL_PER[11] CLKX2ST
CLKOUTX2_H11	Status	CM_DIV_H11_DPLL_PER[9] CLKST
CLKOUTX2_H11	Divider control	CM_DIV_H11_DPLL_PER[5:0] DIVHS
CLKOUTX2_H12	Status	CM_DIV_H12_DPLL_PER[9] CLKST
CLKOUTX2_H12	Divider control	CM_DIV_H12_DPLL_PER[5:0] DIVHS
CLKOUTX2_H13	Status	CM_DIV_H13_DPLL_PER[9] CLKST
CLKOUTX2_H13	Divider control	CM_DIV_H13_DPLL_PER[5:0] DIVHS
CLKOUTX2_H14	Status	CTRL_CORE_PRCM_CLKDIV_CONTROL1[26] DBG_TRC_EXPT_CLK_HSDIV_ EN_ACK

Table 3-50. DPLL_PER Clock Output Parameters (continued)

Clock Output/Divider	Parameter Name	Control/Status Bit Field
CLKOUTX2_H14	Divider control	CTRL_CORE_PRCM_CLKDIV_CONTROL1[13:8] DBG_TRC_EXPT_CLK_DIV

3.6.3.5.3 DPLL_PER Power Modes

This section identifies the operating modes supported by the DPLL and the control bit fields to set its operating modes. For an explanation of the DPLL operating modes and the associated control and status features, see Section 3.6.3.4.2, *Enable Control, Status, and Low-Power Operation Mode*, and Section 3.6.3.4.3, *DPLL Power Modes*.

Table 3-51 lists the operating modes supported by the DPLL.

Table 3-51. DPLL_PER Modes

Low-Power Stop	Fast-Relock Stop	Low-Power Bypass	Fast-Relock Bypass	Lock
Not available	Not available	Available	Available	Available

Table 3-52 lists the control bit fields for the operating mode control of the DPLL.

Table 3-52. DPLL_PER Mode Control Parameters

Parameter Name	Control Bit Field
Low-Power Mode Control	CM_CLKMODE_DPLL_PER[10] DPLL_LPMODE_EN
Manual Mode Control	CM_CLKMODE_DPLL_PER[2:0] DPLL_EN
Auto Mode Control	CM_AUTOIDLE_DPLL_PER[2:0] AUTO_DPLL_MODE

3.6.3.5.4 DPLL_PER Recalibration

Table 3-53 lists the control bit fields for the recalibration feature enable and interrupts of the DPLL. For an explanation of the DPLL recalibration feature, see Section 3.6.3.4.4, *DPLL Recalibration*.

Table 3-53. DPLL_PER Recalibration Feature Parameters

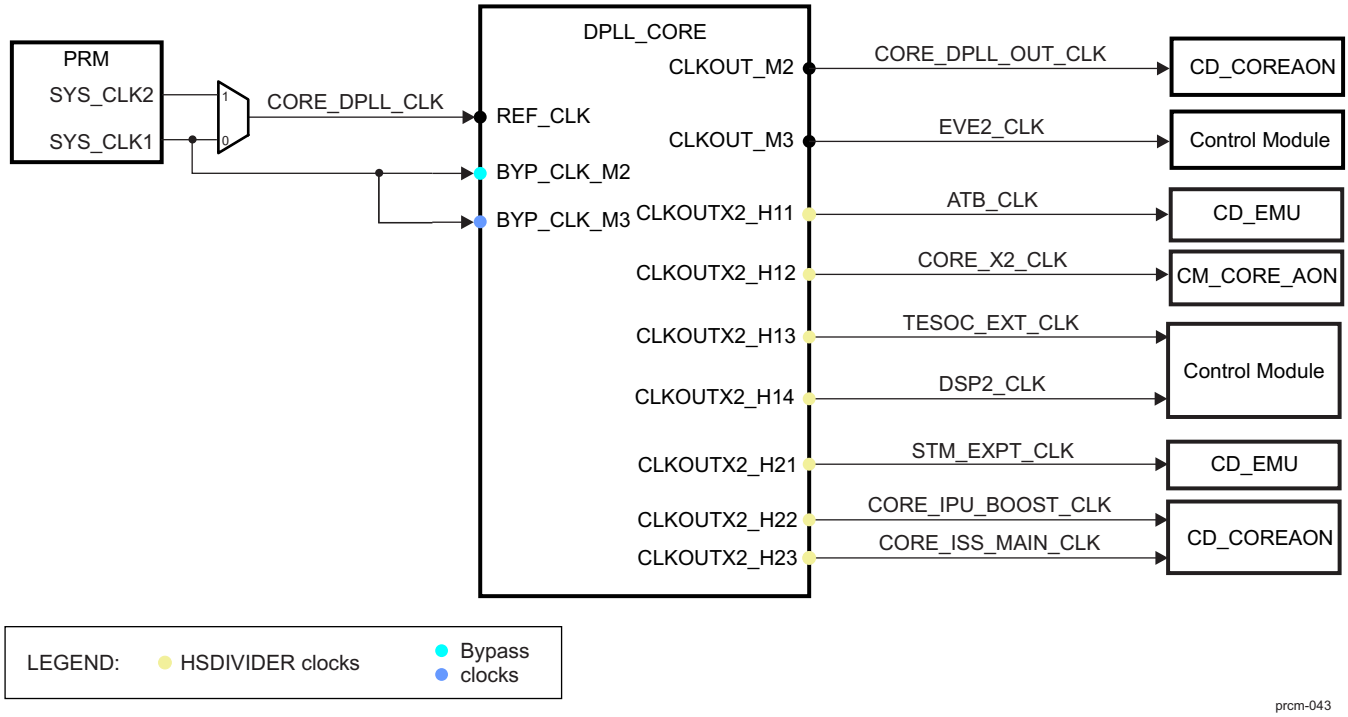
Parameter Name	Control/Status Bit Field
Recalibration Enable Control	CM_CLKMODE_DPLL_PER[8] DPLL_DRIFTGUARD_EN
Recalibration Interrupt Status	PRM_IRQSTATUS_IPU1[3] DPLL_PER_RECAL_ST
Recalibration Interrupt Mask Control	PRM_IRQENABLE_IPU1[3] DPLL_PER_RECAL_EN
Recalibration Interrupt Status	PRM_IRQSTATUS_DSP1[3] DPLL_PER_RECAL_ST
Recalibration Interrupt Mask Control	PRM_IRQENABLE_DSP1[3] DPLL_PER_RECAL_EN
Recalibration Interrupt Status	PRM_IRQSTATUS_DSP2[3] DPLL_PER_RECAL_ST
Recalibration Interrupt Mask Control	PRM_IRQENABLE_DSP2[3] DPLL_PER_RECAL_EN
Recalibration Interrupt Status	PRM_IRQSTATUS_EVE1[3] DPLL_PER_RECAL_ST
Recalibration Interrupt Mask Control	PRM_IRQENABLE_EVE1[3] DPLL_PER_RECAL_EN

3.6.3.6 DPLL_CORE Description

3.6.3.6.1 DPLL_CORE Overview

Figure 3-41 is an overview of the DPLL. For a functional overview of a generic DPLL module, see Section 3.6.3.4, *Generic DPLL Overview*.

Figure 3-41. DPLL_CORE Overview



(1) TESOC is not supported on the DRA78x family of devices.

3.6.3.6.2 DPLL_CORE Synthesized Clock Parameters

This section describes the clock synthesis and clock-out divider parameters of the DPLL. For an explanation of the clock synthesis and output divider parameters of the DPLL module, see Section 3.6.3.4, *Generic DPLL Overview*.

Table 3-54 lists the clock synthesis parameters of the DPLL.

Table 3-54. DPLL_CORE Clock Synthesis Parameters

Parameter Name	Control Bit Field
M (integer)	CM_CLKSEL_DPLL_CORE[18:8] DPLL_MULT
M (fractional)	CTRL_CORE_SMA_SW_22[17:0] CORE_DPLL_REGMF_CONTROL
N	CM_CLKSEL_DPLL_CORE[6:0] DPLL_DIV
M (restore)	CM_CLKSEL_DPLL_CORE_RESTORE[18:8] DPLL_MULT
N (restore)	CM_CLKSEL_DPLL_CORE_RESTORE[6:0] DPLL_DIV

Table 3-55 lists the clock output divider parameters of the DPLL.

Table 3-55. DPLL_CORE Clock Output Parameters

Clock Output/Divider	Parameter Name	Control/Status Bit Field
CLKOUT_M2	Status	CM_DIV_M2_DPLL_CORE[9] CLKST
CLKOUT_M2	Divider control	CM_DIV_M2_DPLL_CORE[4:0] DIVHS
CLKOUTX2_M3	Status	CM_DIV_M3_DPLL_DSP[9] CLKST
CLKOUTX2_M3	Divider control	CM_DIV_M3_DPLL_DSP[4:0] DIVHS
CLKOUTX2_H11	Status	CTRL_CORE_PRCM_CLKDIV_CONTROL1[24] DBG_ATB_CLK_HSDIV_EN_ACK

Table 3-55. DPLL_CORE Clock Output Parameters (continued)

Clock Output/Divider	Parameter Name	Control/Status Bit Field
CLKOUTX2_H11	Divider control	CTRL_CORE_PRCM_CLKDIV_CONTROL1[5:0] DBG_ATB_CLK_DIV
CLKOUTX2_H12	Status	CM_DIV_H12_DPLL_CORE[9] CLKST
CLKOUTX2_H12	Divider control	CM_DIV_H12_DPLL_CORE[5:0] DIVHS
CLKOUTX2_H13	Status	CTRL_CORE_PRCM_CLKDIV_CONTROL2[8] TESOC_HSDIV_EN_ACK
CLKOUTX2_H13	Divider control	CTRL_CORE_PRCM_CLKDIV_CONTROL2[5:0] TESOC_HSDIV(1)
CLKOUTX2_H14	Status	CM_DIV_H14_DPLL_CORE[9] CLKST
CLKOUTX2_H14	Divider control	CM_DIV_H14_DPLL_CORE[5:0] DIVHS
CLKOUTX2_H21	Status	CTRL_CORE_PRCM_CLKDIV_CONTROL1[28] DBG_STM_EXPT_CLK_HSDIV_EN_ACK
CLKOUTX2_H21	Divider control	CTRL_CORE_PRCM_CLKDIV_CONTROL1[21:16] DBG_STM_EXPT_CLK_DIV
CLKOUTX2_H22	Status	CM_DIV_H22_DPLL_CORE[9] CLKST
CLKOUTX2_H22	Divider control	CM_DIV_H22_DPLL_CORE[5:0] DIVHS
CLKOUTX2_H23	Status	CM_DIV_H23_DPLL_CORE[9] CLKST
CLKOUTX2_H23	Divider control	CM_DIV_H23_DPLL_CORE[5:0] DIVHS

(1) TESOC is not supported on the DRA78x family of devices.

3.6.3.6.3 DPLL_CORE Power Modes

This section identifies the operating modes supported by the DPLL and the control bit fields to set its operating modes. For an explanation of the DPLL operating modes and associated control and status features, see Section 3.6.3.4.2, *Enable Control, Status, and Low-Power Operation Mode,* and Section 3.6.3.4.3, *DPLL Power Modes*.

Table 3-56 lists the operating modes supported by the DPLL.

Table 3-56. DPLL_CORE Modes

Low-Power Stop	Fast-Relock Stop	Low-Power Bypass	Fast-Relock Bypass	Lock
Not available	Not available	Available	Available	Available

Table 3-57 lists the control bit fields for the operating mode control of the DPLL.

Table 3-57. DPLL_CORE Mode Control Parameters

Parameter Name	Control Bit Field
Low-Power Mode Control	CM_CLKMODE_DPLL_CORE[10] DPLL_LPMODE_EN
Manual Mode Control	CM_CLKMODE_DPLL_CORE[2:0] DPLL_EN
Auto Mode Control	CM_AUTOIDLE_DPLL_CORE[2:0] AUTO_DPLL_MODE
Low-Power Mode Control (Restore)	CM_CLKMODE_DPLL_CORE_RESTORE[10] DPLL_LPMODE_EN
Manual Mode Control (Restore)	CM_CLKMODE_DPLL_CORE_RESTORE[2:0] DPLL_EN
Auto Mode Control (Restore)	CM_AUTOIDLE_DPLL_CORE_RESTORE[2:0] AUTO_DPLL_MODE

3.6.3.6.4 DPLL_CORE Recalibration

Table 3-58 lists the control bit fields for the recalibration feature enable and interrupts of the DPLL. For an explanation of the DPLL recalibration feature, see Section 3.6.3.4.4, *DPLL Recalibration*.

Table 3-58. DPLL_CORE Recalibration Feature Parameters

Parameter Name	Control/Status Bit Field
Recalibration Enable Control	CM_CLKMODE_DPLL_CORE[8] DPLL_DRIFTGUARD_EN
Recalibration Interrupt Status	PRM_IRQSTATUS_IPU1[0] DPLL_CORE_RECAL_ST
Recalibration Interrupt Mask Control	PRM_IRQENABLE_IPU1[0] DPLL_CORE_RECAL_EN
Recalibration Interrupt Status	PRM_IRQSTATUS_DSP1[0] DPLL_CORE_RECAL_ST
Recalibration Interrupt Mask Control	PRM_IRQENABLE_DSP1[0] DPLL_CORE_RECAL_EN
Recalibration Interrupt Status	PRM_IRQSTATUS_DSP2[0] DPLL_CORE_RECAL_ST
Recalibration Interrupt Mask Control	PRM_IRQENABLE_DSP2[0] DPLL_CORE_RECAL_EN
Recalibration Interrupt Status	PRM_IRQSTATUS_EVE1[0] DPLL_CORE_RECAL_ST
Recalibration Interrupt Mask Control	PRM_IRQENABLE_EVE1[0] DPLL_CORE_RECAL_EN

3.6.3.6.5 Fractional M-factor

DPLL_CORE supports fractional synthesis, that is, the frequency multiplication factor M can be programmed as fractional. The fractional part of M-factor is programmed in the 18-bit CTRL_CORE_SMA_SW_22[17:0] CORE_DPLL_REGMF_CONTROL register. Similarly to REGM, REGMF value is loaded into DPLL at the rising edge of TENABLE signal. Fractional synthesis is typically associated with additional jitter overhead. Refer to the device-specific Data Manual for DPLL jitter characteristics.

The allowed range of M for fractional synthesis to operate is M = 20 and M = 2042.

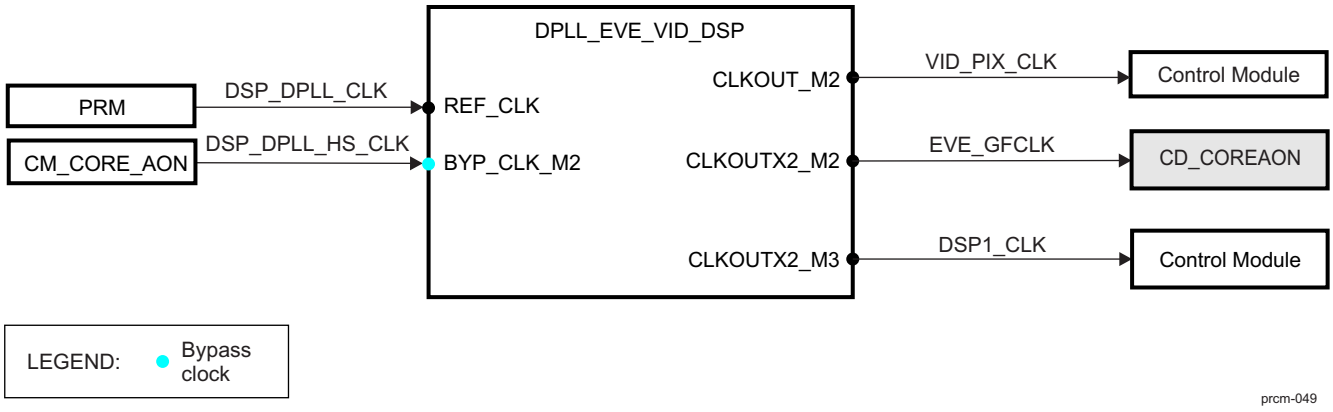
The integer-only division takes place when CORE_DPLL_REGMF_CONTROL is programmed to 0x0.

3.6.3.7 DPLL_EVE_VID_DSP Description

3.6.3.7.1 DPLL_EVE_VID_DSP Overview

Figure 3-42 is an overview of the DPLL. For a functional overview of a generic DPLL module, see Section 3.6.3.4, *Generic DPLL Overview*.

Figure 3-42. DPLL_EVE_VID_DSP Overview



3.6.3.7.2 DPLL_EVE_VID_DSP Synthesized Clock Parameters

This section lists the clock synthesis and clock out divider parameters of the DPLL. For an explanation of the clock synthesis and output divider parameters of the DPLL module, see Section 3.6.3.4, *Generic DPLL Overview*.

Table 3-59 lists the clock synthesis parameters of the DPLL.

Table 3-59. DPLL_EVE_VID_DSP Clock Synthesis Parameters

Parameter Name	Control Bit Field
M	CM_CLKSEL_DPLL_DSP[18:8] DPLL_MULT
N	CM_CLKSEL_DPLL_DSP[6:0] DPLL_DIV

Table 3-60 lists the clock output divider parameters of the DPLL.

Table 3-60. DPLL_EVE_VID_DSP Clock Output Parameters

Clock Output/Divider	Parameter Name	Control/Status Bit Field
CLKOUT_M2	Status	CTRL_CORE_PRCM_CLKDIV_CONTROL2[24] VID_PIX_CLK_HSDIV_EN_ACK
CLKOUT_M2	Divider control	CTRL_CORE_PRCM_CLKDIV_CONTROL2[21:16] VID_PIX_CLK_HSDIV
CLKOUTX2_M3	Status	CM_DIV_M2_DPLL_DSP[9] CLKST
CLKOUTX2_M3	Divider control	CM_DIV_M2_DPLL_DSP[4:0] DIVHS

3.6.3.7.3 DPLL_EVE_VID_DSP Power Modes

This section identifies the operating modes supported by the DPLL and the control bit fields to set its operating modes. For an explanation of the DPLL operating modes and associated control and status features, see Section 3.6.3.4.2, *Enable Control, Status, and Low-Power Operation Mode,* and Section 3.6.3.4.3, *DPLL Power Modes*.

Table 3-61 lists the operating modes supported by the DPLL.

Table 3-61. DPLL_EVE_VID_DSP Modes

Low-Power Stop	Fast-Relock Stop	Low-Power Bypass	Fast-Relock Bypass	Lock
Available	Available	Available	Available	Not available

Table 3-62 lists the control bit fields for the operating mode control of the DPLL.

Table 3-62. DPLL_EVE_VID_DSP Mode Control Parameters

Parameter Name	Control Bit Field
Manual Mode Control	CM_CLKMODE_DPLL_DSP[2:0] DPLL_EN
Auto Mode Control	CM_AUTOIDLE_DPLL_DSP[2:0] AUTO_DPLL_MODE

3.6.3.7.4 DPLL_EVE_VID_DSP Recalibration

Table 3-63 lists the control bit fields for the recalibration feature enable and interrupts of the DPLL. For an explanation of the DPLL recalibration feature, see Section 3.6.3.4.4, *DPLL Recalibration*.

Table 3-63. DPLL_EVE_VID_DSP Recalibration Feature Parameters

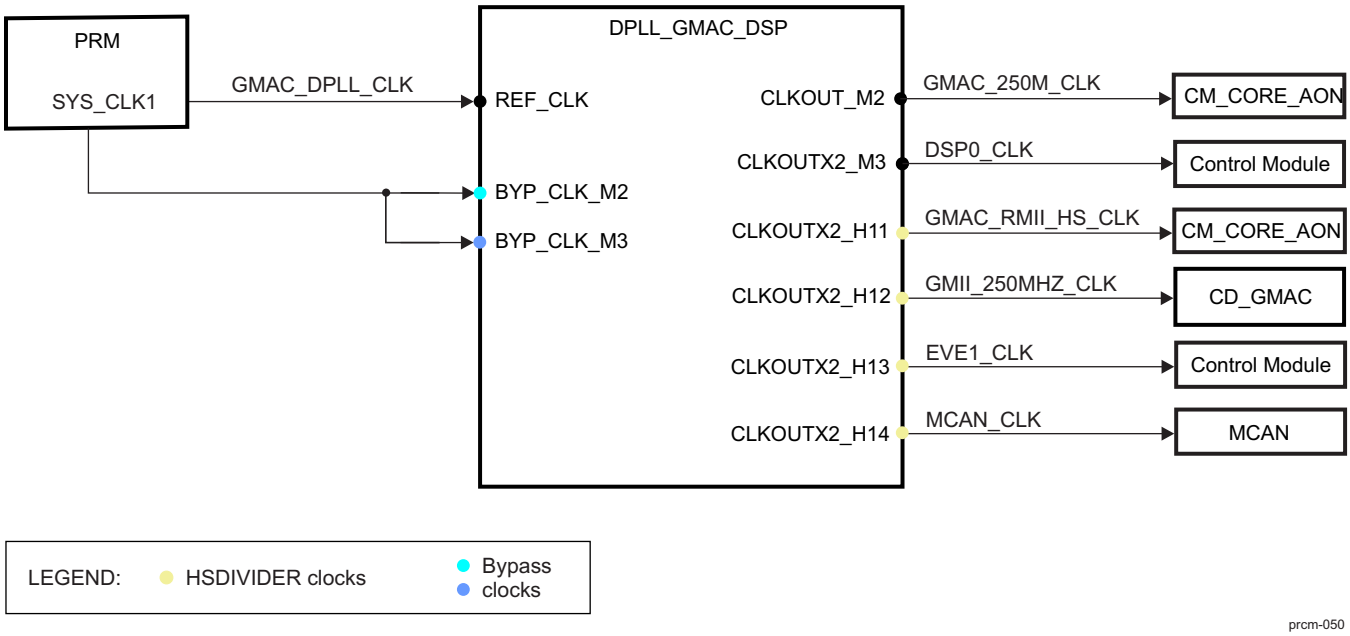
Parameter Name	Control/Status Bit Field
Recalibration Enable Control	CM_CLKMODE_DPLL_DSP[8] DPLL_DRIFTGUARD_EN
Recalibration Interrupt Status	PRM_IRQSTATUS_IPU1[11] DPLL_DSP_RECAL_ST
Recalibration Interrupt Mask Control	PRM_IRQENABLE_IPU1[11] DPLL_DSP_RECAL_EN
Recalibration Interrupt Status	PRM_IRQSTATUS_DSP1[11] DPLL_DSP_RECAL_ST
Recalibration Interrupt Mask Control	PRM_IRQENABLE_DSP1[11] DPLL_DSP_RECAL_EN
Recalibration Interrupt Status	PRM_IRQSTATUS_DSP2[11] DPLL_DSP_RECAL_ST
Recalibration Interrupt Mask Control	PRM_IRQENABLE_DSP2[11] DPLL_DSP_RECAL_EN
Recalibration Interrupt Status	PRM_IRQSTATUS_EVE1[11] DPLL_DSP_RECAL_ST
Recalibration Interrupt Mask Control	PRM_IRQENABLE_EVE1[11] DPLL_DSP_RECAL_EN

3.6.3.8 DPLL_GMAC_DSP Description

3.6.3.8.1 DPLL_GMAC_DSP Overview

Figure 3-43 is an overview of the DPLL. For a functional overview of a generic DPLL module, see Section 3.6.3.4, *Generic DPLL Overview*.

Figure 3-43. DPLL_GMAC_DSP Overview



3.6.3.8.2 DPLL_GMAC_DSP Synthesized Clock Parameters

This section lists the clock synthesis and clock out divider parameters of the DPLL. For an explanation of the clock synthesis and output divider parameters of the DPLL module, see Section 3.6.3.4, *Generic DPLL Overview*.

Table 3-64 lists the clock synthesis parameters of the DPLL.

Table 3-64. DPLL_GMAC_DSP Clock Synthesis Parameters

Parameter Name	Control Bit Field
M	CM_CLKSEL_DPLL_GMAC[18:8] DPLL_MULT
N	CM_CLKSEL_DPLL_GMAC[6:0] DPLL_DIV

Table 3-65 lists the clock output divider parameters of the DPLL.

Table 3-65. DPLL_GMAC_DSP Clock Output Parameters

Clock Output/Divider	Parameter Name	Control/Status Bit Field
CLKOUT_M2	Status	CM_DIV_M2_DPLL_GMAC[9] CLKST
CLKOUT_M2	Divider control	CM_DIV_M2_DPLL_GMAC[4:0] DIVHS
CLKOUTX2_M3	Status	CM_DIV_M3_DPLL_GMAC[9] CLKST
CLKOUTX2_M3	Divider control	CM_DIV_M3_DPLL_GMAC[4:0] DIVHS
CLKOUTX2_H11	Status	CM_DIV_H11_DPLL_GMAC[9] CLKST
CLKOUTX2_H11	Divider control	CM_DIV_H11_DPLL_GMAC[5:0] DIVHS
CLKOUTX2_H12	Status	CM_DIV_H12_DPLL_GMAC[9] CLKST
CLKOUTX2_H12	Divider control	CM_DIV_H12_DPLL_GMAC[5:0] DIVHS

Table 3-65. DPLL_GMAC_DSP Clock Output Parameters (continued)

Clock Output/Divider	Parameter Name	Control/Status Bit Field
CLKOUTX2_H13	Status	CM_DIV_H13_DPLL_GMAC[9] CLKST
CLKOUTX2_H13	Divider control	CM_DIV_H13_DPLL_GMAC[5:0] DIVHS
CLKOUTX2_H14	Status	CTRL_CORE_SMA_SW_17[10] MCAN_CLK_HSDIV_EN_ACK
CLKOUTX2_H14	Divider control	CTRL_CORE_SMA_SW_17[5:0] MCAN_CLK_HSDIV

3.6.3.8.3 DPLL_GMAC_DSP Power Modes

This section identifies the operating modes supported by the DPLL and the control bit fields to set its operating modes. For an explanation of the DPLL operating modes and associated control and status features, see Section 3.6.3.4.2, *Enable Control, Status, and Low-Power Operation Mode,* and Section 3.6.3.4.3, *DPLL Power Modes*.

Table 3-66 lists the operating modes supported by the DPLL.

Table 3-66. DPLL_GMAC_DSP Modes

Low-Power Stop	Fast-Relock Stop	Low-Power Bypass	Fast-Relock Bypass	Lock
Not available	Not available	Available	Available	Available

Table 3-67 lists the control bit fields for the operating mode control of the DPLL.

Table 3-67. DPLL_GMAC_DSP Mode Control Parameters

Parameter Name	Control Bit Field
Manual Mode Control	CM_CLKMODE_DPLL_GMAC[2:0] DPLL_EN
Auto Mode Control	CM_AUTOIDLE_DPLL_GMAC[2:0] AUTO_DPLL_MODE

3.6.3.8.4 DPLL_GMAC_DSP Recalibration

Table 3-68 lists the control bit fields for the recalibration feature enable and interrupts of the DPLL. For an explanation of the DPLL recalibration feature, see Section 3.6.3.4.4, *DPLL Recalibration*.

Table 3-68. DPLL_GMAC_DSP Recalibration Feature Parameters

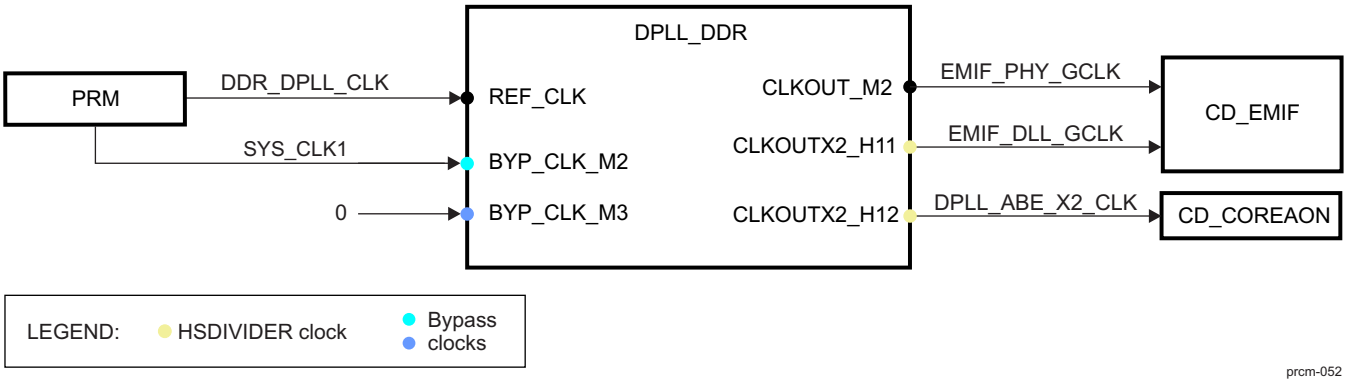
Parameter Name	Control/Status Bit Field
Recalibration Enable Control	CM_CLKMODE_DPLL_GMAC[8] DPLL_DRIFTGUARD_EN
Recalibration Interrupt Status	PRM_IRQSTATUS_IPU1[5] DPLL_GMAC_RECAL_ST
Recalibration Interrupt Mask Control	PRM_IRQENABLE_IPU1[5] DPLL_GMAC_RECAL_EN
Recalibration Interrupt Status	PRM_IRQSTATUS_DSP1[5] DPLL_GMAC_RECAL_ST
Recalibration Interrupt Mask Control	PRM_IRQENABLE_DSP1[5] DPLL_GMAC_RECAL_EN
Recalibration Interrupt Status	PRM_IRQSTATUS_DSP2[5] DPLL_GMAC_RECAL_ST
Recalibration Interrupt Mask Control	PRM_IRQENABLE_DSP2[5] DPLL_GMAC_RECAL_EN
Recalibration Interrupt Status	PRM_IRQSTATUS_EVE1[5] DPLL_GMAC_RECAL_ST
Recalibration Interrupt Mask Control	PRM_IRQENABLE_EVE1[5] DPLL_GMAC_RECAL_EN

3.6.3.9 DPLL_DDR Description

3.6.3.9.1 DPLL_DDR Overview

Figure 3-44 is an overview of the DPLL. For a functional overview of a generic DPLL module, see Section 3.6.3.4, *Generic DPLL Overview*.

Figure 3-44. DPLL_DDR Overview



3.6.3.9.2 DPLL_DDR Synthesized Clock Parameters

This section lists the clock synthesis and clock out divider parameters of the DPLL. For an explanation of the clock synthesis and output divider parameters of the DPLL module, see Section 3.6.3.4, *Generic DPLL Overview*.

Table 3-69 lists the clock synthesis parameters of the DPLL.

Table 3-69. DPLL_DDR Clock Synthesis Parameters

Parameter Name	Control Bit Field
M	CM_CLKSEL_DPLL_DDR[18:8] DPLL_MULT
N	CM_CLKSEL_DPLL_DDR[6:0] DPLL_DIV

Table 3-70 lists the clock output divider parameters of the DPLL.

Table 3-70. DPLL_DDR Clock Output Parameters

Clock Output/Divider	Parameter Name	Control/Status Bit Field
CLKOUT_M2	Status	CM_DIV_M2_DPLL_DDR[9] CLKST
CLKOUT_M2	Divider control	CM_DIV_M2_DPLL_DDR[4:0] DIVHS
CLKOUTX2_H11	Status	CM_DIV_H11_DPLL_DDR[9] CLKST
CLKOUTX2_H11	Divider control	CM_DIV_H11_DPLL_DDR[5:0] DIVHS
CLKOUTX2_H12	Status	CM_DIV_M2_DPLL_ABE[9] CLKST
CLKOUTX2_H12	Divider control	CM_DIV_M2_DPLL_ABE[5:0] DIVHS

3.6.3.9.3 DPLL_DDR Power Modes

This section identifies the operating modes supported by the DPLL and the control bit fields to set its operating modes. For an explanation of the DPLL operating modes and associated control and status features, see Section 3.6.3.4.2, *Enable Control, Status, and Low-Power Operation Mode,* and Section 3.6.3.4.3, *DPLL Power Modes*.

Table 3-71 lists the operating modes supported by the DPLL.

Table 3-71. DPLL_DDR Modes

Low-Power Stop	Fast-Relock Stop	Low-Power Bypass	Fast-Relock Bypass	Lock
Not available	Not available	Available	Available	Available

Table 3-72 lists the control bit fields for the operating mode control of the DPLL.

Table 3-72. DPLL_DDR Mode Control Parameters

Parameter Name	Control Bit Field
Manual Mode Control	CM_CLKMODE_DPLL_DDR[2:0] DPLL_EN
Auto Mode Control	CM_AUTOIDLE_DPLL_DDR[2:0] AUTO_DPLL_MODE

3.6.3.9.4 DPLL_DDR Recalibration

Table 3-73 lists the control bit fields for the recalibration feature enable and interrupts of the DPLL. For an explanation of the DPLL recalibration feature, see Section 3.6.3.4.4, *DPLL Recalibration*.

Table 3-73. DPLL_DDR Recalibration Feature Parameters

Parameter Name	Control/Status Bit Field
Recalibration Enable Control	CM_CLKMODE_DPLL_DDR[8] DPLL_DRIFTGUARD_EN
Recalibration Interrupt Status	PRM_IRQSTATUS_IPU1[7] DPLL_DDR_RECAL_ST
Recalibration Interrupt Mask Control	PRM_IRQENABLE_IPU1[7] DPLL_DDR_RECAL_EN
Recalibration Interrupt Status	PRM_IRQSTATUS_DSP1[7] DPLL_DDR_RECAL_ST
Recalibration Interrupt Mask Control	PRM_IRQENABLE_DSP1[7] DPLL_DDR_RECAL_EN
Recalibration Interrupt Status	PRM_IRQSTATUS_DSP2[7] DPLL_DDR_RECAL_ST
Recalibration Interrupt Mask Control	PRM_IRQENABLE_DSP2[7] DPLL_DDR_RECAL_EN
Recalibration Interrupt Status	PRM_IRQSTATUS_EVE1[7] DPLL_DDR_RECAL_ST
Recalibration Interrupt Mask Control	PRM_IRQENABLE_EVE1[7] DPLL_DDR_RECAL_EN

3.6.4 Clock Domains

This section describes the clock domains available in the device.

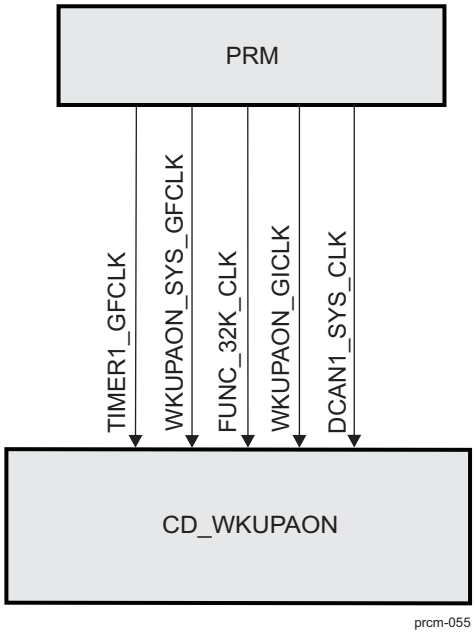
3.6.4.1 CD_WKUPAON Clock Domain

This section identifies the modes supported by the clock domain and the associated control and status bits. It also identifies its dependencies with other clock domains of the device.

3.6.4.1.1 Overview

Figure 3-45 is an overview of the clock domain.

Figure 3-45. CD_WKUPAON Overview



3.6.4.1.2 Clock Domain Modes

Table 3-74 lists the clock domain modes supported by the clock domain.

Table 3-74. CD_WKUPAON Clock Domain Modes

NO_SLEEP	SW_SLEEP	SW_WKUP	HW_AUTO
Available	Not available	Available	Available

Table 3-75 lists the clock domain state transition control and status bits for the clock in this clock domain

Table 3-75. CD_WKUPAON Control and Status Parameters

Parameter Name	Control/Status Bit Field
ABE_LP_CLK Clock Status	CM_WKUPAON_CLKSTCTRL[9] CLKACTIVITY_ABE_LP_CLK
WKUPAON_GICLK Clock Status	CM_WKUPAON_CLKSTCTRL[12] CLKACTIVITY_WKUPAON_GICLK
SYS_CLK Clock Status; includes profiling EMU_SYS_CLK and all functional SYS_CLK	CM_WKUPAON_CLKSTCTRL[8] CLKACTIVITY_SYS_CLK
SYS_CLK Clock Status of functional branches, exclude activity of the EMU_SYS_GCLK clock	CM_WKUPAON_CLKSTCTRL[14] CLKACTIVITY_SYS_CLK_FUNC
WKUPAON_SYS_GFCLK Clock Status	CM_WKUPAON_CLKSTCTRL[11] CLKACTIVITY_WKUPAON_SYS_GFCLK
WKUPAON_32K_GFCLK Clock Control for GPIO1	CM_WKUPAON_GPIO1_CLKCTRL[8] OPTFCLKEN_DBCLK
DCAN1_SYS_CLK Clock Status	CM_WKUPAON_CLKSTCTRL[16] CLKACTIVITY_DCAN1_SYS_CLK
TIMER1_GFCLK Clock Status	CM_WKUPAON_CLKSTCTRL[17] CLKACTIVITY_TIMER1_GFCLK
Clock Domain State Transition Control	CM_WKUPAON_CLKSTCTRL[1:0] CLKTRCTRL

3.6.4.1.3 Clock Domain Dependency

CD_WKUPAON has no static dependency or dynamic dependency with any other clock domain of the device.

3.6.4.1.3.1 Wake-Up Dependency

Table 3-76 lists the wake-up dependency settings for the modules of this clock domain.

Table 3-76. CD_WKUPAON Wake-Up Dependency Association Parameters

Originator Module	Originator Clock Domain	Servicing Clock Domain	Default Setting	Control Bit Field	Access Type
GPIO1	CD_WKUPAON	CD_DSP1	Disabled	PM_WKUPAON_GPIO1_WKDEP[2] WKUPDEP_GPIO1_IRQ1_DSP1	Read/write
GPIO1	CD_WKUPAON	CD_DSP2	Disabled	PM_WKUPAON_GPIO1_WKDEP[5] WKUPDEP_GPIO1_IRQ1_DSP2	Read/write
GPIO1	CD_WKUPAON	CD_DSP1	Disabled	PM_WKUPAON_GPIO1_WKDEP[12] WKUPDEP_GPIO1_IRQ2_DSP1	Read/write
GPIO1	CD_WKUPAON	CD_DSP2	Disabled	PM_WKUPAON_GPIO1_WKDEP[15] WKUPDEP_GPIO1_IRQ2_DSP2	Read/write
GPIO1	CD_WKUPAON	CD_IPU1	Disabled	PM_WKUPAON_GPIO1_WKDEP[4] WKUPDEP_GPIO1_IRQ1_IPU1	Read/write
GPIO1	CD_WKUPAON	CD_IPU1	Disabled	PM_WKUPAON_GPIO1_WKDEP[14] WKUPDEP_GPIO1_IRQ2_IPU1	Read/write
GPIO1	CD_WKUPAON	CD_EVE1	Disabled	PM_WKUPAON_GPIO1_WKDEP[6] WKUPDEP_GPIO1_IRQ1_EVE1	Read/write
GPIO1	CD_WKUPAON	CD_EVE1	Disabled	PM_WKUPAON_GPIO1_WKDEP[16] WKUPDEP_GPIO1_IRQ2_EVE1	Read/write
TIMER1	CD_WKUPAON	CD_DSP1	Disabled	PM_WKUPAON_TIMER1_WKDEP[2] WKUPDEP_TIMER1_DSP1	Read/write
TIMER1	CD_WKUPAON	CD_DSP2	Disabled	PM_WKUPAON_TIMER1_WKDEP[5] WKUPDEP_TIMER1_DSP2	Read/write
TIMER1	CD_WKUPAON	CD_IPU1	Disabled	PM_WKUPAON_TIMER1_WKDEP[4] WKUPDEP_TIMER1_IPU1	Read/write
TIMER1	CD_WKUPAON	CD_EVE1	Disabled	PM_WKUPAON_TIMER1_WKDEP[6] WKUPDEP_TIMER1_EVE1	Read/write

3.6.4.1.4 Clock Domain Module Attributes

Table 3-77 lists for each module of the clock domain the clocks it receives and their role (that is, functional or interface clock).

Table 3-77. CD_WKUPAON Modules Clocks Association

Module	Clock	Clock Type
GPIO1	WKUPAON_GICLK	Interface
	WKUPAON_SYS_GFCLK	Functional
COUNTER_32K	FUNC_32K_CLK	Functional
	WKUPAON_GICLK	Interface
TIMER1	TIMER1_GFCLK	Functional
	WKUPAON_GICLK	Interface
CTRL_MODULE_WKUP	WKUPAON_GICLK	Interface
L4_WKUP interconnect	WKUPAON_GICLK	Interface
DCAN1	DCAN1_SYS_CLK	Functional
	WKUPAON_GICLK	Interface
RTI1	RTI1_CLK	Functional
	WKUPAON_GICLK	Interface

Table 3-77. CD_WKUPAON Modules Clocks Association (continued)

Module	Clock	Clock Type
RTI2	RTI2_CLK	Functional
	WKUPAON_GICLK	Interface
RTI3	RTI3_CLK	Functional
	WKUPAON_GICLK	Interface
RTI4	RTI4_CLK	Functional
	WKUPAON_GICLK	Interface
RTI5	RTI5_CLK	Functional
	WKUPAON_GICLK	Interface

Table 3-78 lists the supported wake-up request generation capability for each module of the clock domain.

Table 3-78. CD_WKUPAON Modules Wake-Up Request

Module	Wake-Up Feature
CTRL_MODULE_WKUP	None
DCAN1	Slave wake-up request (IPU1-IRQ, DSP1-IRQ, DSP2-IRQ, EVE1-IRQ)
GPIO1	Slave wake-up request (IPU1-IRQ, DSP1-IRQ, DSP2-IRQ, EVE1-IRQ,)
COUNTER_32K	None
TIMER1	Slave wake-up request (IPU1-IRQ, DSP1-IRQ, DSP2-IRQ, EVE1-IRQ)
WD_TIMER2	Slave wake-up request (IPU1-IRQ, DSP1-IRQ, DSP2-IRQ, EVE1-IRQ)
L4_WKUP interconnect	None
RTI1	None
RTI2	None
RTI3	None
RTI4	None
RTI5	None

Table 3-79 lists the supported clock-management modes and associated software control bit fields for each module of the power domain.

Table 3-79. CD_WKUPAON Modules Clock-Management Modes and Control

Module	Clock-Management Protocol	Status Bit Field	Role
GPIO1	Slave	CM_WKUPAON_GPIO1_CLKCTRL[17:16] IDLEST	Idle status
COUNTER_32K	Slave	CM_WKUPAON_COUNTER_32K_CLKCTRL[17:16] IDLEST	Idle status
TIMER1	Slave	CM_WKUPAON_TIMER1_CLKCTRL[17:16] IDLEST	Idle status
L4_WKUP interconnect	Slave	CM_WKUPAON_L4_WKUP_CLKCTRL[17:16] IDLEST	Idle status
DCAN1	Slave	CM_WKUPAON_DCAN1_CLKCTRL[17:16] IDLEST	Idle status
RTI1	Slave	CM_WKUPAON_RTI1_CLKCTRL[17:16] IDLEST	Idle status
RTI2	Slave	CM_WKUPAON_RTI2_CLKCTRL[17:16] IDLEST	Idle status
RTI3	Slave	CM_WKUPAON_RTI3_CLKCTRL[17:16] IDLEST	Idle status
RTI4	Slave	CM_WKUPAON_RTI4_CLKCTRL[17:16] IDLEST	Idle status
RTI5	Slave	CM_WKUPAON_RTI5_CLKCTRL[17:16] IDLEST	Idle status

Table 3-80 lists the supported clock management modes and associated software control bit fields for each module of the power domain.

Table 3-80. CD_WKUPAON Modules Slave Clock-Management Modes and Control

Module	Disabled	Auto	Enabled	Control Bit Field	Access Type
GPIO1	Available	Available	N/A	CM_WKUPAON_GPIO1_CLKCTRL[1:0] MODULEMODE	Read/write
COUNTER_32K	N/A	Available	N/A	CM_WKUPAON_COUNTER_32K_CLKCTRL[1:0] MODULEMODE	Read only
TIMER1	Available	N/A	Available	CM_WKUPAON_TIMER1_CLKCTRL[1:0] MODULEMODE	Read/write
L4_WKUP interconnect	N/A	Available	N/A	CM_WKUPAON_L4_WKUP_CLKCTRL[1:0] MODULEMODE	Read only
DCAN1	Available	N/A	Available	CM_WKUPAON_DCAN1_CLKCTRL[1:0] MODULEMODE	Read/write
RTI1	N/A	Available	N/A	CM_WKUPAON_RTI1_CLKCTRL[1:0] MODULEMODE	Read only
RTI2	N/A	Available	N/A	CM_WKUPAON_RTI2_CLKCTRL[1:0] MODULEMODE	Read only
RTI3	N/A	Available	N/A	CM_WKUPAON_RTI3_CLKCTRL[1:0] MODULEMODE	Read only
RTI4	N/A	Available	N/A	CM_WKUPAON_RTI4_CLKCTRL[1:0] MODULEMODE	Read only
RTI5	N/A	Available	N/A	CM_WKUPAON_RTI5_CLKCTRL[1:0] MODULEMODE	Read only

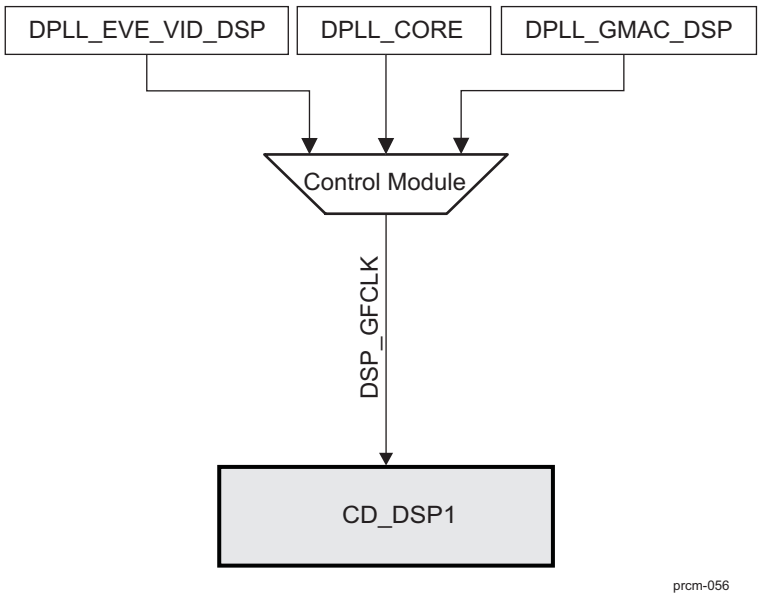
3.6.4.2 CD_DSP1 Clock Domain

This section identifies the modes supported by the clock domain and the associated control and status bits. It also identifies its dependencies with other clock domains of the device.

3.6.4.2.1 Overview

Figure 3-46 is an overview of the clock domain.

Figure 3-46. CD_DSP1 Overview



3.6.4.2.2 Clock Domain Modes

Table 3-81 lists the modes supported by the clock domain.

Table 3-81. CD_DSP1 Clock Domain Modes

NO_SLEEP	SW_SLEEP	SW_WKUP	HW_AUTO
Available	Available	Available	Available

Table 3-82 lists the clock domain state transition control and status bits for the clock in this clock domain.

Table 3-82. CD_DSP1 Control and Status Parameters

Parameter Name	Control/Status Bit Field
DSP_GFCLK Clock Status	CM_DSP1_CLKSTCTRL[8] CLKACTIVITY_DSP1_GFCLK
Clock Domain State Transition Control	CM_DSP1_CLKSTCTRL[1:0] CLKTRCTRL

3.6.4.2.3 Clock Domain Dependency

CD_DSP1 has no module wake-up dependency with any other clock domain of the device.

3.6.4.2.3.1 Static Dependency

Table 3-83 lists the static dependency of the clock domain with respect to other clock domains of the device.

Table 3-83. CD_DSP1 Static Dependency Association Parameters

Clock Domain Name	Default Setting	Control Bit Field	Access Type
CD_CAM	Disabled	CM_DSP1_STATICDEP[9] CAM_STATDEP	Read write
CD_L3MAIN1	Always enabled	CM_DSP1_STATICDEP[5] L3MAIN1_STATDEP	Read only
CD_L3INIT	Disabled	CM_DSP1_STATICDEP[7] L3INIT_STATDEP	Read/write
CD_L4_CFG	Disabled	CM_DSP1_STATICDEP[12] L4CFG_STATDEP	Read only
CD_L4PER1	Disabled	CM_DSP1_STATICDEP[13] L4PER_STATDEP	Read/write
CD_L4PER2	Disabled	CM_DSP1_STATICDEP[26] L4PER2_STATDEP	Read/write
CD_L4PER3	Disabled	CM_DSP1_STATICDEP[27] L4PER3_STATDEP	Read/write
CD_EMIF	Disabled	CM_DSP1_STATICDEP[4] EMIF_STATDEP	Read/write
CD_WKUPAON	Disabled	CM_DSP1_STATICDEP[15] WKUPAON_STATDEP	Read/write
CD_COREAON	Always disabled	CM_DSP1_STATICDEP[16] COREAON_STATDEP	Read only
CD_CUSTEFUSE	Always disabled	CM_DSP1_STATICDEP[17] CUSTEFUSE_STATDEP	Read only
CD_DSP2	Disabled	CM_DSP1_STATICDEP[18] DSP2_STATDEP	Read/write
CD_IPU	Disabled	CM_DSP1_STATICDEP[24] IPU_STATDEP	Read/write
CD_IPU1	Disabled	CM_DSP1_STATICDEP[23] IPU1_STATDEP	Read/write
CD_DSS	Disabled	CM_DSP1_STATICDEP[8] DSS_STATDEP	Read/write
CD_GMAC	Disabled	CM_DSP1_STATICDEP[25] GMAC_STATDEP	Read/write
CD_ISS	Disabled	CM_DSP1_STATICDEP[28] ISS_STATDEP	Read/write
CD_CRC	Disabled	CM_DSP1_STATICDEP[30] CRC_STATDEP	Read/write
CD_EVE1	Disabled	CM_DSP1_STATICDEP[19] EVE1_STATDEP	Read/write

3.6.4.2.3.2 Dynamic Dependency

Table 3-84 lists the dynamic dependency of the clock domain with respect to other clock domains of the device.

Table 3-84. CD_DSP1 Dynamic Dependency Association Parameters

Clock Domain Name	Default Setting	Control Bit Field	Access Type
CD_L3MAIN1	Always enabled	CM_DSP1_DYNAMICDEP[5] L3MAIN1_DYNDEP	Read only

3.6.4.2.4 Clock Domain Module Attributes

Table 3-85 lists for each module of the clock domain the clocks the module receives and their role (that is, functional or interface clock).

Table 3-85. CD_DSP1 Modules Clocks Association

Module	Clock	Clock Type
DSP1	DSP1_GFCLK	Interface and functional

Table 3-86 lists the supported wake-up request generation capability for each module of the clock domain.

Table 3-86. CD_DSP1 Modules Wake-Up Request

Module	Wake-Up Feature
DSP1	Master wake-up request

Table 3-87 lists the supported clock-management modes and associated software control bit fields for each module of the power domain.

Table 3-87. CD_DSP1 Modules Clock-Management Modes and Control

Module	Clock-Management Protocol	Status Bit Field	Role
DSP1	Master/slave	CM_DSP1_DSP1_CLKCTRL[18] STBYST	Standby status
		CM_DSP1_DSP1_CLKCTRL[17:16] IDLEST	Idle status

Table 3-88 lists the supported clock-management modes and associated software control bit fields for each module of the power domain.

Table 3-88. CD_DSP1 Modules Slave Clock-Management Modes and Control

Module	Disabled	Auto	Enabled	Control Bit Field	Access Type
DSP	Available	Available	N/A	CM_DSP1_DSP1_CLKCTRL[1:0] MODULEMODE	Read/write

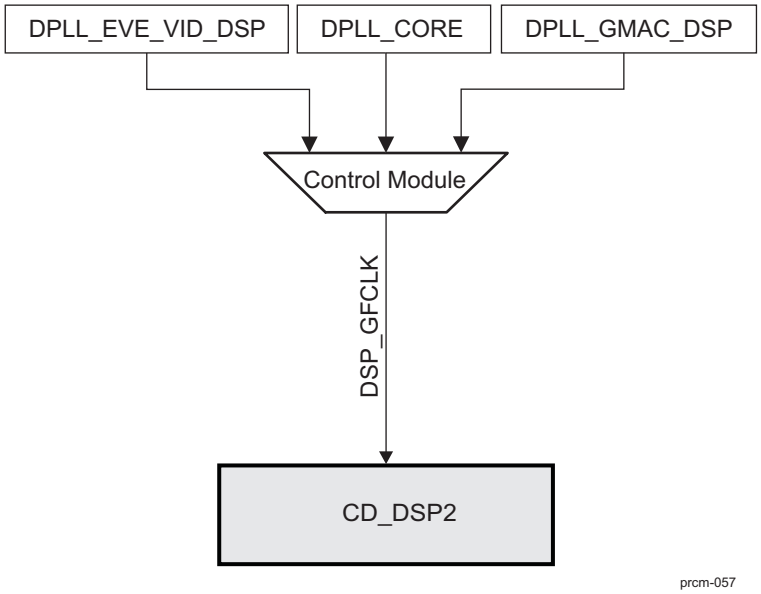
3.6.4.3 CD_DSP2 Clock Domain

This section identifies the modes supported by the clock domain and the associated control and status bits. It also identifies its dependencies with other clock domains of the device.

3.6.4.3.1 Overview

Figure 3-47 is an overview of the clock domain.

Figure 3-47. CD_DSP2 Overview



3.6.4.3.2 Clock Domain Modes

Table 3-89 lists the modes supported by the clock domain.

Table 3-89. CD_DSP2 Clock Domain Modes

NO_SLEEP	SW_SLEEP	SW_WKUP	HW_AUTO
Available	Available	Available	Available

Table 3-90 lists the clock domain state transition control and status bits for the clock in this clock domain.

Table 3-90. CD_DSP2 Control and Status Parameters

Parameter Name	Control/Status Bit Field
DSP2_GFCLK Clock Status	CM_DSP2_CLKSTCTRL[8] CLKACTIVITY_DSP2_GFCLK
Clock Domain State Transition Control	CM_DSP2_CLKSTCTRL[1:0] CLKTRCTRL

3.6.4.3.3 Clock Domain Dependency

CD_DSP2 has no module wake-up dependency with any other clock domain of the device.

3.6.4.3.3.1 Static Dependency

Table 3-91 lists the static dependency of the clock domain with respect to other clock domains of the device.

Table 3-91. CD_DSP2 Static Dependency Association Parameters

Clock Domain Name	Default Setting	Control Bit Field	Access Type
CD_CAM	Disabled	CM_DSP2_STATICDEP[9] CAM_STATDEP	Read write
CD_L3MAIN1	Always enabled	CM_DSP2_STATICDEP[5] L3MAIN1_STATDEP	Read only
CD_L3INIT	Disabled	CM_DSP2_STATICDEP[7] L3INIT_STATDEP	Read/write
CD_L4_CFG	Disabled	CM_DSP2_STATICDEP[12] L4CFG_STATDEP	Read only
CD_L4PER1	Disabled	CM_DSP2_STATICDEP[13] L4PER_STATDEP	Read/write
CD_L4PER2	Disabled	CM_DSP2_STATICDEP[26] L4PER2_STATDEP	Read/write

Table 3-91. CD_DSP2 Static Dependency Association Parameters (continued)

Clock Domain Name	Default Setting	Control Bit Field	Access Type
CD_L4PER3	Disabled	CM_DSP2_STATICDEP[27] L4PER3_STATDEP	Read/write
CD_EMIF	Disabled	CM_DSP2_STATICDEP[4] EMIF_STATDEP	Read/write
CD_WKUPAON	Disabled	CM_DSP2_STATICDEP[15] WKUPAON_STATDEP	Read/write
CD_COREAON	Always disabled	CM_DSP2_STATICDEP[16] COREAON_STATDEP	Read only
CD_CUSTEFUSE	Always disabled	CM_DSP2_STATICDEP[17] CUSTEFUSE_STATDEP	Read only
CD_DSP1	Disabled	CM_DSP2_STATICDEP[1] DSP1_STATDEP	Read/write
CD_IPU	Disabled	CM_DSP2_STATICDEP[24] IPU_STATDEP	Read/write
CD_IPU1	Disabled	CM_DSP2_STATICDEP[23] IPU1_STATDEP	Read/write
CD_DSS	Disabled	CM_DSP2_STATICDEP[8] DSS_STATDEP	Read/write
CD_GMAC	Disabled	CM_DSP2_STATICDEP[25] GMAC_STATDEP	Read/write
CD_ISS	Disabled	CM_DSP2_STATICDEP[28] ISS_STATDEP	Read/write
CD_CRC	Disabled	CM_DSP2_STATICDEP[30] CRC_STATDEP	Read/write
CD_EVE1	Disabled	CM_DSP2_STATICDEP[19] EVE1_STATDEP	Read/write

3.6.4.3.3.2 Dynamic Dependency

Table 3-92 lists the dynamic dependency of the clock domain with respect to other clock domains of the device.

Table 3-92. CD_DSP2 Dynamic Dependency Association Parameters

Clock Domain Name	Default Setting	Control Bit Field	Access Type
CD_L3MAIN1	Always enabled	CM_DSP1_DYNAMICDEP[5] L3MAIN1_DYNDEP	Read only

3.6.4.3.4 Clock Domain Module Attributes

Table 3-93 lists for each module of the clock domain the clocks the module receives and their role (that is, functional or interface clock).

Table 3-93. CD_DSP2 Modules Clocks Association

Module	Clock	Clock Type
DSP2	DSP2_GFCLK	Interface and functional

Table 3-94 lists the supported wake-up request generation capability for each module of the clock domain.

Table 3-94. CD_DSP2 Modules Wake-Up Request

Module	Wake-Up Feature
DSP2	Master wake-up request

Table 3-95 lists the supported clock-management modes and associated software control bit fields for each module of the power domain.

Table 3-95. CD_DSP2 Modules Clock-Management Modes and Control

Module	Clock-Management Protocol	Status Bit Field	Role
DSP2	Master/slave	CM_DSP2_DSP2_CLKCTRL[18] STBYST	Standby status
		CM_DSP2_DSP2_CLKCTRL[17:16] IDLEST	Idle status

Table 3-96 lists the supported clock-management modes and associated software control bit fields for each module of the power domain.

Table 3-96. CD_DSP2 Modules Slave Clock-Management Modes and Control

Module	Disabled	Auto	Enabled	Control Bit Field	Access Type
DSP2	Available	Available	N/A	CM_DSP2_DSP2_CLKCTRL[1:0] MODULEMODE	Read/write

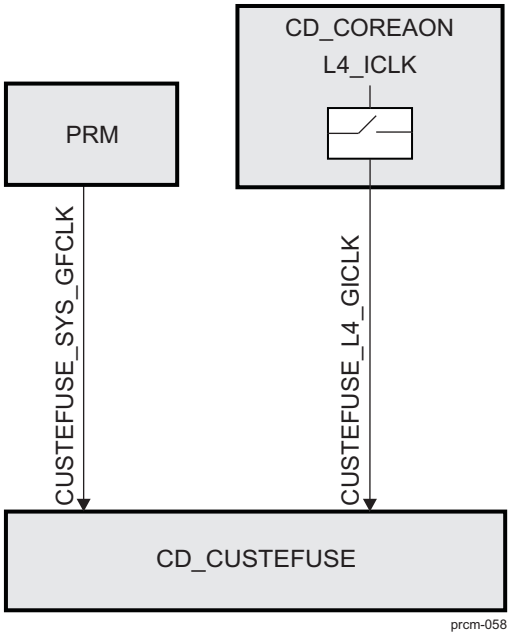
3.6.4.4 CD_CUSTEFUSE Clock Domain

This section identifies the modes supported by the clock domain and the associated control and status bits. It also identifies its dependencies with other clock domains of the device.

3.6.4.4.1 Overview

Figure 3-48 is an overview of the clock domain.

Figure 3-48. CD_CUSTEFUSE Overview



3.6.4.4.2 Clock Domain Modes

Table 3-97 lists the modes supported by the clock domain.

Table 3-97. CD_CUSTEFUSE Clock Domain Modes

NO_SLEEP	SW_SLEEP	SW_WKUP	HW_AUTO
Available	N/A	Available	Available

Table 3-98 lists the clock domain state transition control and status bits for the clock in this clock domain.

Table 3-98. CD_CUSTEFUSE Control and Status Parameters

Parameter Name	Control/Status Bit Field
CUSTEFUSE_SYS_GFCLK Clock Status	CM_CUSTEFUSE_CLKSTCTRL[9] CLKACTIVITY_CUSTEFUSE_SYS_GFCLK

Table 3-98. CD_CUSTEFUSE Control and Status Parameters (continued)

Parameter Name	Control/Status Bit Field
CUSTEFUSE_L4_GICLK Clock Status	CM_CUSTEFUSE_CLKSTCTRL[8] CLKACTIVITY_CUSTEFUSE_L4_GICLK
Clock Domain State Transition Control	CM_CUSTEFUSE_CLKSTCTRL[1:0] CLKTRCTRL

3.6.4.4.3 Clock Domain Dependency

CD_CUSTEFUSE has no static or dynamic dependency with any other clock domain of the device.

3.6.4.4.4 Clock Domain Module Attributes

Table 3-99 lists for each module of the clock domain the clocks the module receives and their role (that is, functional or interface clock).

Table 3-99. CD_CUSTEFUSE Modules Clocks Association

Module	Clock	Clock Type
EFUSE_CTRL_CUST	CUSTEFUSE_SYS_GFCLK	Functional
	CUSTEFUSE_L4_GICLK	Interface

Table 3-100 lists the supported wake-up request generation capability for each module of the clock domain.

Table 3-100. CD_CUSTEFUSE Modules Wake-Up Request

Module	Wake-Up Feature
EFUSE_CTRL_CUST	None

Table 3-101 lists the supported clock-management modes and associated software control bit fields for each module of the power domain.

Table 3-101. CD_CUSTEFUSE Modules Clock-Management Modes and Control

Module	Clock-Management Protocol	Status Bit Field	Role
EFUSE_CTRL_CUST	Idle	CM_CUSTEFUSE_EFUSE_CTRL_CUST_CLKCTRL[17:16] IDLEST	Idle status

Table 3-102 lists the supported clock-management modes and associated software control bit fields for each module of the power domain.

Table 3-102. CD_CUSTEFUSE Modules Slave Clock-Management Modes and Control

Module	Disabled	Auto	Enabled	Control Bit Field	Access Type
EFUSE_CTRL_CUST	Available	N/A	Available	CM_CUSTEFUSE_EFUSE_CTRL_CUST_CLKCTRL[1:0] MODULEMODE	Read/write

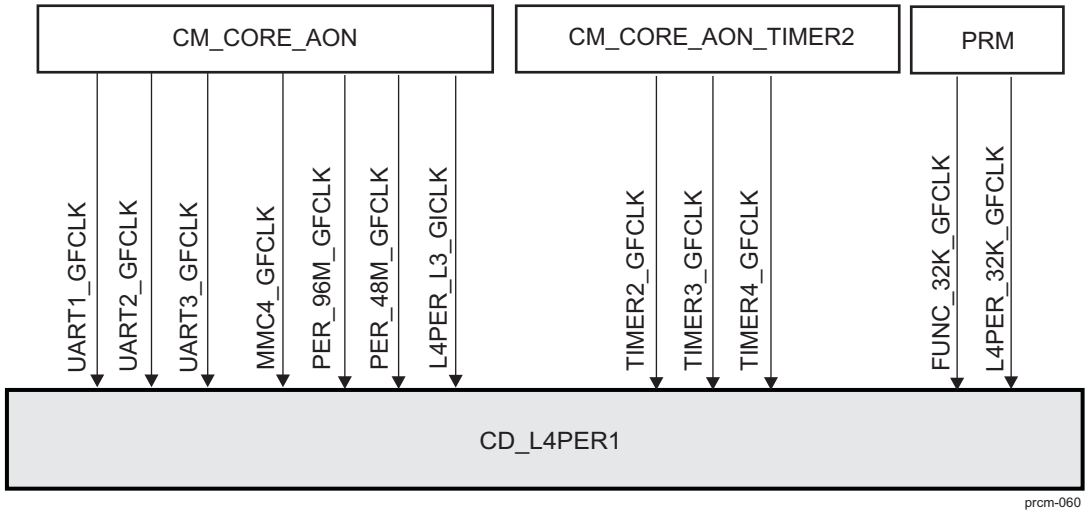
3.6.4.5 CD_L4PER1 Clock Domain

This section identifies the modes supported by the clock domain and the associated control and status bits. It also identifies its dependencies with other clock domains of the device.

3.6.4.5.1 Overview

Figure 3-49 is an overview of the clock domain.

Figure 3-49. CD_L4PER1 Overview



3.6.4.5.2 Clock Domain Modes

Table 3-103 lists the clock domain modes supported by the clock domain.

Table 3-103. CD_L4PER1 Clock Domain Modes

NO_SLEEP	SW_SLEEP	SW_WKUP	HW_AUTO
Available	Available	Available	Available

Table 3-104 lists the clock domain state transition control and status bits for the clock in this clock domain.

Table 3-104. CD_L4PER1 Control and Status Parameters

Parameter Name	Control/Status Bit Field
TIMER2_GFCLK clock status	CM_L4PER_CLKSTCTRL[11] CLKACTIVITY_TIMER2_GFCLK
TIMER3_GFCLK clock status	CM_L4PER_CLKSTCTRL[12] CLKACTIVITY_TIMER3_GFCLK
TIMER4_GFCLK clock status	CM_L4PER_CLKSTCTRL[13] CLKACTIVITY_TIMER4_GFCLK
L4PER_L3_GICLK clock status	CM_L4PER_CLKSTCTRL[8] CLKACTIVITY_L4PER_L3_GICLK
L4PER_32K_GFCLK clock status	CM_L4PER_CLKSTCTRL[27] CLKACTIVITY_PER_32K_GFCLK
PER_48M_GFCLK clock status	CM_L4PER_CLKSTCTRL[20] CLKACTIVITY_PER_48M_GFCLK
PER_96M_GFCLK clock status	CM_L4PER_CLKSTCTRL[21] CLKACTIVITY_PER_96M_GFCLK
GPIO_GFCLK clock status	CM_L4PER_CLKSTCTRL[24] CLKACTIVITY_GPIO_GFCLK
UART1_GFCLK clock status	CM_L4PER_CLKSTCTRL[15] CLKACTIVITY_UART1_GFCLK
UART2_GFCLK clock status	CM_L4PER_CLKSTCTRL[16] CLKACTIVITY_UART2_GFCLK
UART3_GFCLK clock status	CM_L4PER_CLKSTCTRL[17] CLKACTIVITY_UART3_GFCLK
MMC4_GFCLK clock status	CM_L4PER_CLKSTCTRL[23] CLKACTIVITY_MMC4_GFCLK
Clock Domain State Transition Control	CM_L4PER_CLKSTCTRL[1:0] CLKTRCTRL

3.6.4.5.3 Clock Domain Dependency

CD_L4PER1 has no static dependency with any other clock domain of the device.

3.6.4.5.3.1 Dynamic Dependency

Table 3-105 lists the dynamic dependency of the clock domain with respect to other clock domains of the device.

Table 3-105. CD_L4PER1 Dynamic Dependency Association Parameters

Clock Domain Name	Default Setting	Control Bit Field	Access Type
CD_DSS	Always enabled	CM_L4PER_DYNAMICDEP[8] DSS_DYNDEP	Read only
CD_L3INIT	Always enabled	CM_L4PER_DYNAMICDEP[7] L3INIT_DYNDEP	Read only
CD_IPU	Always enabled	CM_L4PER_DYNAMICDEP[3] IPU_DYNDEP	Read only

3.6.4.5.3.2 Wake-Up Dependency

Table 3-106 lists the wake-up dependency settings for the modules of this clock domain.

Table 3-106. CD_L4PER1 Wake-Up Dependency Association Parameters

Originator Module	Originator Clock Domain	Servicing Clock Domain	Default Setting	Control Bit Field	Access Type
TIMER2	CD_L4PER	CD_DSP, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_TIMER2_WKDEP[2] WKUPDEP_TIMER2_DSP1	Read/write
TIMER2	CD_L4PER	CD_DSP2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_TIMER2_WKDEP[5] WKUPDEP_TIMER2_DSP2	Read/write
TIMER2	CD_L4PER	CD_IPU1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_TIMER2_WKDEP[4] WKUPDEP_TIMER2_IPU1	Read/write
TIMER2	CD_L4PER	CD_EVE1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_TIMER2_WKDEP[6] WKUPDEP_TIMER2_EVE1	Read/write
TIMER3	CD_L4PER	CD_IPU1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_TIMER3_WKDEP[4] WKUPDEP_TIMER3_IPU1	Read/write
TIMER3	CD_L4PER	CD_DSP, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_TIMER3_WKDEP[2] WKUPDEP_TIMER3_DSP1	Read/write
TIMER3	CD_L4PER	CD_DSP2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_TIMER3_WKDEP[5] WKUPDEP_TIMER3_DSP2	Read/write
TIMER3	CD_L4PER	CD_EVE1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_TIMER3_WKDEP[6] WKUPDEP_TIMER3_EVE1	Read/write

Table 3-106. CD_L4PER1 Wake-Up Dependency Association Parameters (continued)

Originator Module	Originator Clock Domain	Servicing Clock Domain	Default Setting	Control Bit Field	Access Type
TIMER4	CD_L4PER	CD_IPU1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_TIMER4_WKDEP[4] WKUPDEP_TIMER4_IPU1	Read/write
TIMER4	CD_L4PER	CD_DSP, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_TIMER4_WKDEP[2] WKUPDEP_TIMER4_DSP1	Read/write
TIMER4	CD_L4PER	CD_DSP2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_TIMER4_WKDEP[5] WKUPDEP_TIMER4_DSP2	Read/write
TIMER4	CD_L4PER	CD_EVE1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_TIMER4_WKDEP[6] WKUPDEP_TIMER4_EVE1	Read/write
GPIO2	CD_L4PER	CD_DSP, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Enabled	PM_L4PER_GPIO2_WKDEP[2] WKUPDEP_GPIO2_IRQ1_DSP1	Read/write
GPIO2	CD_L4PER	CD_DSP2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Enabled	PM_L4PER_GPIO2_WKDEP[5] WKUPDEP_GPIO2_IRQ1_DSP2	Read/write
GPIO2	CD_L4PER	CD_DSP, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Enabled	PM_L4PER_GPIO2_WKDEP[12] WKUPDEP_GPIO2_IRQ2_DSP1	Read/write
GPIO2	CD_L4PER	CD_DSP2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Enabled	PM_L4PER_GPIO2_WKDEP[15] WKUPDEP_GPIO2_IRQ2_DSP2	Read/write
GPIO2	CD_L4PER	CD_IPU1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_GPIO2_WKDEP[4] WKUPDEP_GPIO2_IRQ1_IPU1	Read/write
GPIO2	CD_L4PER	CD_IPU1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_GPIO2_WKDEP[14] WKUPDEP_GPIO2_IRQ2_IPU1	Read/write
GPIO2	CD_L4PER	CD_EVE1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_GPIO2_WKDEP[6] WKUPDEP_GPIO2_IRQ1_EVE1	Read/write
GPIO2	CD_L4PER	CD_EVE1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_GPIO2_WKDEP[16] WKUPDEP_GPIO2_IRQ2_EVE1	Read/write

Table 3-106. CD_L4PER1 Wake-Up Dependency Association Parameters (continued)

Originator Module	Originator Clock Domain	Servicing Clock Domain	Default Setting	Control Bit Field	Access Type
GPIO3	CD_L4PER	CD_DSP, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_GPIO3_WKDEP[2] WKUPDEP_GPIO3_IRQ1_DSP1	Read/write
GPIO3	CD_L4PER	CD_DSP2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_GPIO3_WKDEP[5] WKUPDEP_GPIO3_IRQ1_DSP2	Read/write
GPIO3	CD_L4PER	CD_DSP, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_GPIO3_WKDEP[12] WKUPDEP_GPIO3_IRQ2_DSP1	Read/write
GPIO3	CD_L4PER	CD_DSP2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_GPIO3_WKDEP[15] WKUPDEP_GPIO3_IRQ2_DSP2	Read/write
GPIO3	CD_L4PER	CD_IPU1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_GPIO3_WKDEP[4] WKUPDEP_GPIO3_IRQ1_IPU1	Read/write
GPIO3	CD_L4PER	CD_IPU1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_GPIO3_WKDEP[14] WKUPDEP_GPIO3_IRQ2_IPU1	Read/write
GPIO3	CD_L4PER	CD_EVE1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_GPIO3_WKDEP[6] WKUPDEP_GPIO3_IRQ1_EVE1	Read/write
GPIO3	CD_L4PER	CD_EVE1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_GPIO3_WKDEP[16] WKUPDEP_GPIO3_IRQ2_EVE1	Read/write
GPIO4	CD_L4PER	CD_DSP, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_GPIO4_WKDEP[2] WKUPDEP_GPIO4_IRQ1_DSP1	Read/write
GPIO4	CD_L4PER	CD_DSP2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_GPIO4_WKDEP[5] WKUPDEP_GPIO4_IRQ1_DSP2	Read/write
GPIO4	CD_L4PER	CD_DSP, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_GPIO4_WKDEP[12] WKUPDEP_GPIO4_IRQ2_DSP1	Read/write
GPIO4	CD_L4PER	CD_DSP2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_GPIO4_WKDEP[15] WKUPDEP_GPIO4_IRQ2_DSP2	Read/write

Table 3-106. CD_L4PER1 Wake-Up Dependency Association Parameters (continued)

Originator Module	Originator Clock Domain	Servicing Clock Domain	Default Setting	Control Bit Field	Access Type
GPIO4	CD_L4PER	CD_IPU1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_GPIO4_WKDEP[4] WKUPDEP_GPIO4_IRQ1_IPU1	Read/write
GPIO4	CD_L4PER	CD_IPU1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_GPIO4_WKDEP[14] WKUPDEP_GPIO4_IRQ2_IPU1	Read/write
GPIO4	CD_L4PER	CD_EVE1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_GPIO4_WKDEP[6] WKUPDEP_GPIO4_IRQ1_EVE1	Read/write
GPIO4	CD_L4PER	CD_EVE1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_GPIO4_WKDEP[16] WKUPDEP_GPIO4_IRQ2_EVE1	Read/write
I2C1	CD_L4PER	CD_IPU1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_I2C1_WKDEP[4] WKUPDEP_I2C1_IRQ_IPU1	Read/write
I2C1	CD_L4PER	CD_DSP, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_I2C1_WKDEP[2] WKUPDEP_I2C1_IRQ_DSP1	Read/write
I2C1	CD_L4PER	CD_DSP2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_I2C1_WKDEP[5] WKUPDEP_I2C1_IRQ_DSP2	Read/write
I2C1	CD_L4PER	CD_DSP, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Enabled	PM_L4PER_I2C1_WKDEP[12] WKUPDEP_I2C1_DMA_DSP1	Read/write
I2C1	CD_L4PER	CD_DSP2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Enabled	PM_L4PER_I2C1_WKDEP[15] WKUPDEP_I2C1_DMA_DSP2	Read/write
I2C1	CD_L4PER	CD_EVE1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_I2C1_WKDEP[6] WKUPDEP_I2C1_IRQ_EVE1	Read/write
I2C2	CD_L4PER	CD_IPU1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_I2C2_WKDEP[4] WKUPDEP_I2C2_IRQ_IPU1	Read/write
I2C2	CD_L4PER	CD_DSP, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_I2C2_WKDEP[2] WKUPDEP_I2C2_IRQ_DSP1	Read/write

Table 3-106. CD_L4PER1 Wake-Up Dependency Association Parameters (continued)

Originator Module	Originator Clock Domain	Servicing Clock Domain	Default Setting	Control Bit Field	Access Type
I2C2	CD_L4PER	CD_DSP2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_I2C2_WKDEP[5] WKUPDEP_I2C2_IRQ_DSP2	Read/write
I2C2	CD_L4PER	CD_DSP, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Enabled	PM_L4PER_I2C2_WKDEP[12] WKUPDEP_I2C2_DMA_DSP1	Read/write
I2C2	CD_L4PER	CD_DSP2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Enabled	PM_L4PER_I2C2_WKDEP[15] WKUPDEP_I2C2_DMA_DSP2	Read/write
I2C2	CD_L4PER	CD_EVE1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_I2C2_WKDEP[6] WKUPDEP_I2C2_IRQ_EVE1	Read/write
McSPI1	CD_L4PER	CD_DSP1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_MCSPI1_WKDEP[2] WKUPDEP_MCSPI1_DSP1	Read/write
McSPI1	CD_L4PER	CD_DSP2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_MCSPI1_WKDEP[5] WKUPDEP_MCSPI1_DSP2	Read/write
McSPI1	CD_L4PER	CD_IPU1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_MCSPI1_WKDEP[4] WKUPDEP_MCSPI1_IPU1	Read/write
McSPI1	CD_L4PER	CD_EVE1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_MCSPI1_WKDEP[6] WKUPDEP_MCSPI1_EVE1	Read/write
McSPI2	CD_L4PER	CD_DSP1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_MCSPI2_WKDEP[2] WKUPDEP_MCSPI2_DSP1	Read/write
McSPI2	CD_L4PER	CD_DSP2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_MCSPI2_WKDEP[5] WKUPDEP_MCSPI2_DSP2	Read/write
McSPI2	CD_L4PER	CD_IPU1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_MCSPI2_WKDEP[4] WKUPDEP_MCSPI2_IPU1	Read/write
McSPI2	CD_L4PER	CD_EVE1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_MCSPI2_WKDEP[6] WKUPDEP_MCSPI2_EVE1	Read/write

Table 3-106. CD_L4PER1 Wake-Up Dependency Association Parameters (continued)

Originator Module	Originator Clock Domain	Servicing Clock Domain	Default Setting	Control Bit Field	Access Type
McSPI3	CD_L4PER	CD_DSP1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_MCSPI3_WKDEP[2] WKUPDEP_MCSPI3_DSP1	Read/write
McSPI3	CD_L4PER	CD_DSP2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_MCSPI3_WKDEP[5] WKUPDEP_MCSPI3_DSP2	Read/write
McSPI3	CD_L4PER	CD_IPU1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_MCSPI3_WKDEP[4] WKUPDEP_MCSPI3_IPU1	Read/write
McSPI3	CD_L4PER	CD_EVE1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_MCSPI3_WKDEP[6] WKUPDEP_MCSPI3_EVE1	Read/write
McSPI4	CD_L4PER	CD_DSP1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_MCSPI4_WKDEP[2] WKUPDEP_MCSPI4_DSP1	Read/write
McSPI4	CD_L4PER	CD_DSP2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_MCSPI4_WKDEP[5] WKUPDEP_MCSPI4_DSP2	Read/write
McSPI4	CD_L4PER	CD_IPU1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_MCSPI4_WKDEP[4] WKUPDEP_MCSPI4_IPU1	Read/write
McSPI4	CD_L4PER	CD_EVE1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_MCSPI4_WKDEP[6] WKUPDEP_MCSPI4_EVE1	Read/write
MMC	CD_L4PER	CD_DSP, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_MMC4_WKDEP[2] WKUPDEP_MMC4_DSP1	Read/write
MMC	CD_L4PER	CD_DSP2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_MMC4_WKDEP[5] WKUPDEP_MMC4_DSP2	Read/write
MMC	CD_L4PER	CD_IPU1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_MMC4_WKDEP[4] WKUPDEP_MMC4_IPU1	Read/write
MMC	CD_L4PER	CD_EVE1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_MMC4_WKDEP[6] WKUPDEP_MMC4_EVE1	Read/write

Table 3-106. CD_L4PER1 Wake-Up Dependency Association Parameters (continued)

Originator Module	Originator Clock Domain	Servicing Clock Domain	Default Setting	Control Bit Field	Access Type
UART1	CD_L4PER	CD_DSP, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_UART1_WKDEP[2] WKUPDEP_UART1_DSP1	Read/write
UART1	CD_L4PER	CD_DSP2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_UART1_WKDEP[5] WKUPDEP_UART1_DSP2	Read/write
UART1	CD_L4PER	CD_IPU1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_UART1_WKDEP[4] WKUPDEP_UART1_IPU1	Read/write
UART1	CD_L4PER	CD_EVE1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_UART1_WKDEP[6] WKUPDEP_UART1_EVE1	Read/write
UART2	CD_L4PER	CD_DSP, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_UART2_WKDEP[2] WKUPDEP_UART2_DSP1	Read/write
UART2	CD_L4PER	CD_DSP2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_UART2_WKDEP[5] WKUPDEP_UART2_DSP2	Read/write
UART2	CD_L4PER	CD_IPU1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_UART2_WKDEP[4] WKUPDEP_UART2_IPU1	Read/write
UART2	CD_L4PER	CD_EVE1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_UART2_WKDEP[6] WKUPDEP_UART2_EVE1	Read/write
UART3	CD_L4PER	CD_DSP, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_UART3_WKDEP[2] WKUPDEP_UART3_DSP1	Read/write
UART3	CD_L4PER	CD_DSP2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_UART3_WKDEP[5] WKUPDEP_UART3_DSP2	Read/write
UART3	CD_L4PER	CD_IPU1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_UART3_WKDEP[4] WKUPDEP_UART3_IPU1	Read/write
UART3	CD_L4PER	CD_EVE1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER_UART3_WKDEP[6] WKUPDEP_UART3_EVE1	Read/write

3.6.4.5.4 Clock Domain Module Attributes

Table 3-107 lists for each module of the clock domain the clocks the module receives and their role (that is, functional or interface clock).

Table 3-107. CD_L4PER1 Modules Clocks Association

Module	Clock	Clock Type
DCC6[1]	L4PER_L3_GICLK	Interface[2]
DCC7[1]	L4PER_L3_GICLK	Interface[2]
TIMER2	TIMER2_GFCLK	Functional
	L4PER_L3_GICLK	Interface[2]
TIMER3	TIMER3_GFCLK	Functional
	L4PER_L3_GICLK	Interface[2]
TIMER4	TIMER4_GFCLK	Functional
	L4PER_L3_GICLK	Interface[2]
DCC5[1]	L4PER_L3_GICLK	Interface[2]
ELM	L4PER_L3_GICLK	Interface[2]
GPIO2	GPIO_GFCLK	Functional
	L4PER_L3_GICLK	Interface[2]
GPIO3	GPIO_GFCLK	Functional
	L4PER_L3_GICLK	Interface[2]
GPIO4	GPIO_GFCLK	Functional
	L4PER_L3_GICLK	Interface[2][2]
ESM[3]	L4PER_L3_GICLK	Interface and Functional[2]
I2C1	L4PER_L3_GICLK	Interface[2]
	PER_96M_GFCLK	Functional
I2C2	L4PER_L3_GICLK	Interface[2]
	PER_96M_GFCLK	Functional
L4_PER1 interconnect	L4PER_L3_GICLK	Interface[2]
McSPI1	L4PER_L3_GICLK	Interface[2]
	PER_48M_GFCLK	Functional
McSPI2	L4PER_L3_GICLK	Interface[2]
	PER_48M_GFCLK	Functional
McSPI3	L4PER_L3_GICLK	Interface[2]
	PER_48M_GFCLK	Functional
McSPI4	L4PER_L3_GICLK	Interface[2]
	PER_48M_GFCLK	Functional
MMC	L4PER_32K_GFCLK	Functional
	L4PER_L3_GICLK	Interface[2]
	MMC4_GFCLK	Functional
UART1	L4PER_L3_GICLK	Interface[2]
	UART1_GFCLK	Functional
UART2	L4PER_L3_GICLK	Interface[2]
	UART2_GFCLK	Functional
UART3	L4PER_L3_GICLK	Interface[2]
	UART3_GFCLK	Functional

(1) DCC modules are not supported on the DRA78x family of devices.

(2) The L4 clock, required by the L4 interface, is created internally to the module. It is derived from the L4PER_L3_GICLK clock divided by 2 using a gated version of L4_ICLK.

(3) ESM is not supported on the DRA78x family of devices.

Table 3-108 lists the supported wake-up request generation capability for each module of the clock domain.

Table 3-108. CD_L4PER1 Modules Wake-Up Request

Module	Wake-Up Feature
TIMER2	Slave wake-up request (IPU1-IRQ, DSP1-IRQ, DSP2-IRQ, EVE1-IRQ)
TIMER3	Slave wake-up request (IPU1-IRQ, DSP1-IRQ, DSP2-IRQ, EVE1-IRQ)
TIMER4	Slave wake-up request (IPU1-IRQ, DSP1-IRQ, DSP2-IRQ, EVE1-IRQ)
ELM	None
GPIO2	Slave wake-up request (IPU1-IRQ, DSP1-IRQ, DSP2-IRQ, EVE1-IRQ)
GPIO3	Slave wake-up request (IPU1-IRQ, DSP1-IRQ, DSP2-IRQ, EVE1-IRQ)
GPIO4	Slave wake-up request (IPU1-IRQ, DSP1-IRQ, DSP2-IRQ, EVE1-IRQ)
ESM(1)	None
I2C1	Slave wake-up request (IPU1-IRQ, DSP1-IRQ, DSP2-IRQ, EVE1-IRQ)
I2C2	Slave wake-up request (IPU1-IRQ, DSP1-IRQ, DSP2-IRQ, EVE1-IRQ)
L4_PER1 interconnect	None
McSPI1	Slave wake-up request (IPU1-IRQ, DSP1-IRQ, DSP2-IRQ, EVE1-IRQ)
McSPI2	Slave wake-up request (IPU1-IRQ, DSP1-IRQ, DSP2-IRQ, EVE1-IRQ)
McSPI3	Slave wake-up request (IPU1-IRQ, DSP1-IRQ, DSP2-IRQ, EVE1-IRQ)
McSPI4	Slave wake-up request (IPU1-IRQ, DSP1-IRQ, DSP2-IRQ, EVE1-IRQ)
MMC	Slave wake-up request (IPU1-IRQ, DSP1-IRQ, DSP2-IRQ, EVE1-IRQ)
UART1	Slave wake-up request (IPU1-IRQ, DSP1-IRQ, DSP2-IRQ, EVE1-IRQ)
UART2	Slave wake-up request (IPU1-IRQ, DSP1-IRQ, DSP2-IRQ, EVE1-IRQ)
UART3	Slave wake-up request (IPU1-IRQ, DSP1-IRQ, DSP2-IRQ, EVE1-IRQ)

(1) ESM is not supported on the DRA78x family of devices.

Table 3-109 lists the supported clock-management modes and associated software control bit fields for each module of the power domain.

Table 3-109. CD_L4PER1 Modules Clock-Management Modes and Control

Module	Clock-Management Protocol	Status Bit Field	Role
DCC6(1)	Slave	CM_L4PER_DCC6_CLKCTRL[17:16] IDLEST	Idle status
DCC7(1)	Slave	CM_L4PER_DCC7_CLKCTRL[17:16] IDLEST	Idle status
TIMER2	Slave	CM_L4PER_TIMER2_CLKCTRL[17:16] IDLEST	Idle status
TIMER3	Slave	CM_L4PER_TIMER3_CLKCTRL[17:16] IDLEST	Idle status
TIMER4	Slave	CM_L4PER_TIMER4_CLKCTRL[17:16] IDLEST	Idle status
DCC5(1)	Slave	CM_L4PER_DCC5_CLKCTRL[17:16] IDLEST	Idle status
ELM	Slave	CM_L4PER_ELM_CLKCTRL[17:16] IDLEST	Idle status

(1) DCC modules are not supported on the DRA78x family of devices.

Table 3-109. CD_L4PER1 Modules Clock-Management Modes and Control (continued)

Module	Clock-Management Protocol	Status Bit Field	Role
GPIO2	Slave	CM_L4PER_GPIO2_CLKCTRL[17:16] IDLEST	Idle status
GPIO3	Slave	CM_L4PER_GPIO3_CLKCTRL[17:16] IDLEST	Idle status
GPIO4	Slave	CM_L4PER_GPIO4_CLKCTRL[17:16] IDLEST	Idle status
ESM(2)	Slave	CM_L4PER_ESM_CLKCTRL[17:16] IDLEST	Idle status
I2C1	Slave	CM_L4PER_I2C1_CLKCTRL[17:16] IDLEST	Idle status
I2C2	Slave	CM_L4PER_I2C2_CLKCTRL[17:16] IDLEST	Idle status
L4_PER1 interconnect	Slave	CM_L4PER_L4_PER1_CLKCTRL[17:16] IDLEST	Idle status
McSPI1	Slave	CM_L4PER_MCSPI1_CLKCTRL[17:16] IDLEST	Idle status
McSPI2	Slave	CM_L4PER_MCSPI2_CLKCTRL[17:16] IDLEST	Idle status
McSPI3	Slave	CM_L4PER_MCSPI3_CLKCTRL[17:16] IDLEST	Idle status
McSPI4	Slave	CM_L4PER_MCSPI4_CLKCTRL[17:16] IDLEST	Idle status
MMC	Slave	CM_L4PER_MMC4_CLKCTRL[17:16] IDLEST	Idle status
UART1	Slave	CM_L4PER_UART1_CLKCTRL[17:16] IDLEST	Idle status
UART2	Slave	CM_L4PER_UART2_CLKCTRL[17:16] IDLEST	Idle status
UART3	Slave	CM_L4PER_UART3_CLKCTRL[17:16] IDLEST	Idle status

(2) ESM is not supported on the DRA78x family of devices.

Table 3-110 lists the supported clock-management modes and associated software control bit fields for each module of the power domain.

Table 3-110. CD_L4PER1 Modules Slave Clock-Management Modes and Control

Module	Disabled	Auto	Enabled	Control Bit Field	Access Type
DCC6(1)	Available	N/A	Available	CM_L4PER_DCC6_CLKCTRL[1:0] MODULEMODE	Read/write
DCC7(1)	Available	N/A	Available	CM_L4PER_DCC7_CLKCTRL[1:0] MODULEMODE	Read/write
TIMER2	Available	N/A	Available	CM_L4PER_TIMER2_CLKCTRL[1:0] MODULEMODE	Read/write
TIMER3	Available	N/A	Available	CM_L4PER_TIMER3_CLKCTRL[1:0] MODULEMODE	Read/write
TIMER4	Available	N/A	Available	CM_L4PER_TIMER4_CLKCTRL[1:0] MODULEMODE	Read/write
DCC5(1)	Available	N/A	Available	CM_L4PER_DCC5_CLKCTRL[1:0] MODULEMODE	Read/write
ELM	N/A	Available	N/A	CM_L4PER_ELM_CLKCTRL[1:0] MODULEMODE	Read only
GPIO2	Available	Available	N/A	CM_L4PER_GPIO2_CLKCTRL[1:0] MODULEMODE	Read/write
GPIO3	Available	Available	N/A	CM_L4PER_GPIO3_CLKCTRL[1:0] MODULEMODE	Read/write
GPIO4	Available	Available	N/A	CM_L4PER_GPIO4_CLKCTRL[1:0] MODULEMODE	Read/write
ESM(2)	Available	N/A	Available	CM_L4PER_ESM_CLKCTRL[1:0] MODULEMODE	Read/write
I2C1	Available	N/A	Available	CM_L4PER_I2C1_CLKCTRL[1:0] MODULEMODE	Read/write
I2C2	Available	N/A	Available	CM_L4PER_I2C2_CLKCTRL[1:0] MODULEMODE	Read/write
L4_PER1 interconnect	N/A	Available	N/A	CM_L4PER_L4_PER1_CLKCTRL[1:0] MODULEMODE	Read only

(1) DCC modules are not supported on the DRA78x family of devices.
(2) ESM is not supported on the DRA78x family of devices.

Table 3-110. CD_L4PER1 Modules Slave Clock-Management Modes and Control (continued)

Module	Disabled	Auto	Enabled	Control Bit Field	Access Type
McSPI1	Available	N/A	Available	CM_L4PER_MCSPI1_CLKCTRL[1:0] MODULEMODE	Read/write
McSPI2	Available	N/A	Available	CM_L4PER_MCSPI2_CLKCTRL[1:0] MODULEMODE	Read/write
McSPI3	Available	N/A	Available	CM_L4PER_MCSPI3_CLKCTRL[1:0] MODULEMODE	Read/write
McSPI4	Available	N/A	Available	CM_L4PER_MCSPI4_CLKCTRL[1:0] MODULEMODE	Read/write
MMC	Available	N/A	Available	CM_L4PER_MMC4_CLKCTRL[1:0] MODULEMODE	Read/write
UART1	Available	N/A	Available	CM_L4PER_UART1_CLKCTRL[1:0] MODULEMODE	Read/write
UART2	Available	N/A	Available	CM_L4PER_UART2_CLKCTRL[1:0] MODULEMODE	Read/write
UART3	Available	N/A	Available	CM_L4PER_UART3_CLKCTRL[1:0] MODULEMODE	Read/write

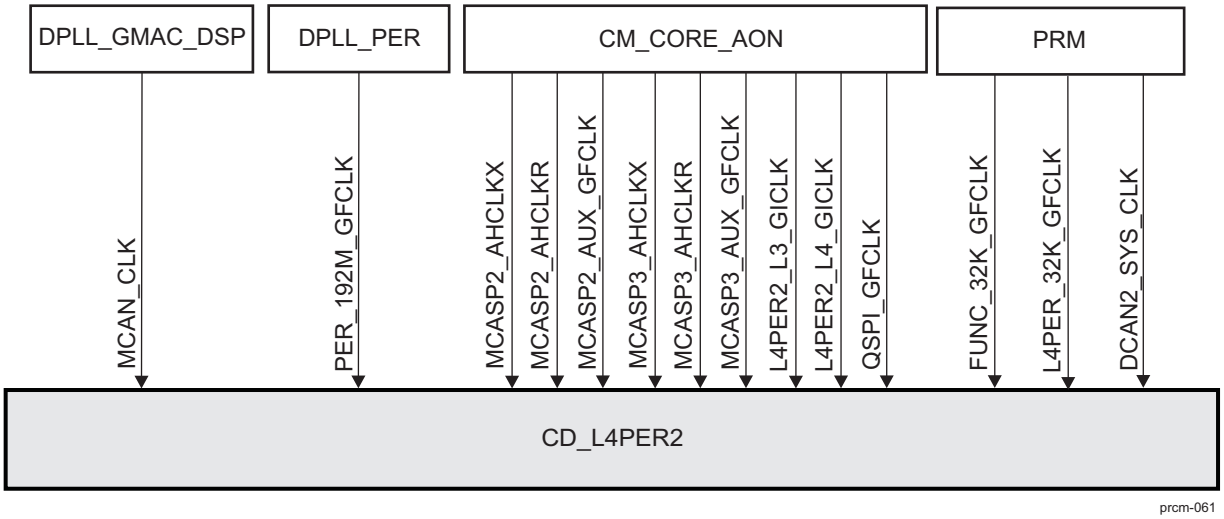
3.6.4.6 CD_L4PER2 Clock Domain

This section identifies the modes supported by the clock domain and the associated control and status bits. It also identifies its dependencies with other clock domains of the device.

3.6.4.6.1 Overview

Figure 3-50 is an overview of the clock domain.

Figure 3-50. CD_L4PER2 Overview



3.6.4.6.2 Clock Domain Modes

Table 3-111 lists the clock domain modes supported by the clock domain.

Table 3-111. CD_L4PER2 Clock Domain Modes

NO_SLEEP	SW_SLEEP	SW_WKUP	HW_AUTO
Available	Available	Available	Available

Table 3-112 lists the clock domain state transition control and status bits for the clock in this clock domain.

Table 3-112. CD_L4PER2 Control and Status Parameters

Parameter Name	Control/Status Bit Field
QSPI_GFCLK clock status	CM_L4PER2_CLKSTCTRL[12] CLKACTIVITY_QSPI_GFCLK
L4PER2_L3_GICLK clock status	CM_L4PER2_CLKSTCTRL[16] CLKACTIVITY_L4PER2_L3_GICLK
MCASP2_AHCLKX clock status	CM_L4PER2_CLKSTCTRL[22] CLKACTIVITY_MCASP4_AHCLKX
MCASP2_AUX_GFCLK clock status	CM_L4PER2_CLKSTCTRL[23] CLKACTIVITY_MCASP4_AUX_GFCLK
MCASP2_AHCLKR clock status	CM_L4PER2_CLKSTCTRL[26] CLKACTIVITY_MCASP6_AHCLKX
MCASP3_AHCLKX clock status	CM_L4PER2_CLKSTCTRL[25] CLKACTIVITY_MCASP5_AHCLKX
MCASP3_AUX_GFCLK clock status	CM_L4PER2_CLKSTCTRL[24] CLKACTIVITY_MCASP5_AUX_GFCLK
MCASP3_AHCLKR clock status	CM_L4PER2_CLKSTCTRL[28] CLKACTIVITY_MCASP7_AHCLK
PER_192M_GFCLK clock status	CM_L4PER2_CLKSTCTRL[13] CLKACTIVITY_PER_192M_GFCLK
Clock Domain State Transition Control	CM_L4PER2_CLKSTCTRL[1:0] CLKTRCTRL

3.6.4.6.3 Clock Domain Dependency

CD_L4PER2 has no static dependency with any other clock domain of the device.

3.6.4.6.3.1 Dynamic Dependency

Table 3-113 lists the dynamic dependency of the clock domain with respect to other clock domains of the device.

Table 3-113. CD_L4PER2 Dynamic Dependency Association Parameters

Clock Domain Name	Default Setting	Control Bit Field	Access Type
CD_L4_CFG	Always enabled	CM_L4PER2_DYNAMICDEP[12] L4CFG_DYNDEP	Read only
CD_L3INIT	Always enabled	CM_L4PER2_DYNAMICDEP[7] L3INIT_DYNDEP	Read only
CD_IPU	Always enabled	CM_L4PER2_DYNAMICDEP[3] IPU_DYNDEP	Read only
CD_CRC	Always enabled	CM_L4PER2_DYNAMICDEP[6] CRC_DYNDEP	Read only
CD_GMAC	Always enabled	CM_L4PER2_DYNAMICDEP[22] GMAC_DYNDEP	Read only

3.6.4.6.3.2 Wake-Up Dependency

Table 3-114 lists the wake-up dependency settings for the modules of this clock domain.

Table 3-114. CD_L4PER2 Wake-Up Dependency Association Parameters

Originator Module	Originator Clock Domain	Servicing Clock Domain	Default Setting	Control Bit Field	Access Type
MCAN	CD_L4PER2	CD_DSP, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER2_DCAN2_WKDEP[2] WKUPDEP_DCAN2_DSP1	Read/write
MCAN	CD_L4PER2	CD_DSP2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER2_DCAN2_WKDEP[5] WKUPDEP_DCAN2_DSP2	Read/write

Table 3-114. CD_L4PER2 Wake-Up Dependency Association Parameters (continued)

Originator Module	Originator Clock Domain	Servicing Clock Domain	Default Setting	Control Bit Field	Access Type
MCAN	CD_L4PER2	CD_IPU1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER2_DCAN2_WKDEP[4] WKUPDEP_DCAN2_IPU1	Read/write
MCAN	CD_L4PER2	CD_EVE1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER2_DCAN2_WKDEP[6] WKUPDEP_DCAN2_EVE1	Read/write
McASP2	CD_L4PER2	CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Enabled	PM_L4PER2_MCASP4_WKDEP[12] WKUPDEP_MCASP4_DMA_DSP1	Read/write
McASP2	CD_L4PER2	CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Enabled	PM_L4PER2_MCASP4_WKDEP[15] WKUPDEP_MCASP4_DMA_DSP2	Read/write
McASP2	CD_L4PER2	CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER2_MCASP4_WKDEP[2] WKUPDEP_MCASP4_IRQ_DSP1	Read/write
McASP2	CD_L4PER2	CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER2_MCASP4_WKDEP[5] WKUPDEP_MCASP4_IRQ_DSP2	Read/write
McASP2	CD_L4PER2	CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER2_MCASP4_WKDEP[4] WKUPDEP_MCASP4_IRQ_IPU1	Read/write
McASP2	CD_L4PER2	CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER2_MCASP4_WKDEP[6] WKUPDEP_MCASP4_IRQ_EVE1	Read/write
McASP3	CD_L4PER2	CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Enabled	PM_L4PER2_MCASP5_WKDEP[12] WKUPDEP_MCASP5_DMA_DSP1	Read/write
McASP3	CD_L4PER2	CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Enabled	PM_L4PER2_MCASP5_WKDEP[15] WKUPDEP_MCASP5_DMA_DSP2	Read/write
McASP3	CD_L4PER2	CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER2_MCASP5_WKDEP[2] WKUPDEP_MCASP5_IRQ_DSP1	Read/write
McASP3	CD_L4PER2	CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER2_MCASP5_WKDEP[5] WKUPDEP_MCASP5_IRQ_DSP2	Read/write
McASP3	CD_L4PER2	CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER2_MCASP5_WKDEP[4] WKUPDEP_MCASP5_IRQ_IPU1	Read/write
McASP3	CD_L4PER2	CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER2_MCASP5_WKDEP[6] WKUPDEP_MCASP5_IRQ_EVE1	Read/write

Table 3-114. CD_L4PER2 Wake-Up Dependency Association Parameters (continued)

Originator Module	Originator Clock Domain	Servicing Clock Domain	Default Setting	Control Bit Field	Access Type
QSPI	CD_L4PER2	CD_DSP, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER2_QSPI_WKDEP[2] WKUPDEP_QSPI_DSP1	Read/write
QSPI	CD_L4PER2	CD_DSP2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER2_QSPI_WKDEP[5] WKUPDEP_QSPI_DSP2	Read/write
QSPI	CD_L4PER2	CD_IPU1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER2_QSPI_WKDEP[4] WKUPDEP_QSPI_IPU1	Read/write
QSPI	CD_L4PER2	CD_EVE1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER2_QSPI_WKDEP[6] WKUPDEP_QSPI_EVE1	Read/write
ADC	CD_L4PER2	CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Enabled	PM_L4PER2_ADC_WKDEP[12] WKUPDEP_ADC_DMA_DSP1	Read/write
ADC	CD_L4PER2	CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Enabled	PM_L4PER2_ADC_WKDEP[15] WKUPDEP_ADC_DMA_DSP2	Read/write
ADC	CD_L4PER2	CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER2_ADC_WKDEP[2] WKUPDEP_ADC_IRQ_DSP1	Read/write
ADC	CD_L4PER2	CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER2_ADC_WKDEP[5] WKUPDEP_ADC_IRQ_DSP2	Read/write
ADC	CD_L4PER2	CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER2_ADC_WKDEP[4] WKUPDEP_ADC_IRQ_IPU1	Read/write
ADC	CD_L4PER2	CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_L4PER2_ADC_WKDEP[6] WKUPDEP_ADC_IRQ_EVE1	Read/write

3.6.4.6.4 Clock Domain Module Attributes

Table 3-115 lists for each module of the clock domain the clocks the module receives and their role (that is, functional or interface clock).

Table 3-115. CD_L4PER2 Modules Clocks Association

Module	Clock	Clock Type
L4_PER2	L4PER2_L3_GICLK	Interface(1)
MCAN	L4PER2_L3_GICLK	Interface(1)
	MCAN_CLK	Functional
ATL	L4PER2_L3_GICLK	Interface(1) and Functional

(1) The L4 clock, required by the L4 interface, is created internally to the module. It is derived from the L4PER2_L3_GICLK clock divided by 2 using a gated version of L4_ICLK.

Table 3-115. CD_L4PER2 Modules Clocks Association (continued)

Module	Clock	Clock Type
McASP2	L4PER2_L3_GICLK	Interface(1)
	MCASP2_AHCLKR	Functional
	MCASP2_AHCLKX	Functional
	MCASP2_AUX_GFCLK	Functional
McASP3	L4PER2_L3_GICLK	Interface(1)
	MCASP3_AHCLKR	Functional
	MCASP3_AHCLKX	Functional
	MCASP3_AUX_GFCLK	Functional
ADC	L4PER2_L3_GICLK	Interface(1)
	ADC_CLK	Functional
QSPI	L4PER2_L3_GICLK	Interface
	QSPI_GFCLK	Functional
PWMSS1	L4PER2_L3_GICLK	Interface(1) and Functional(1)

Table 3-116 lists the supported wake-up request generation capability for each module of the clock domain.

Table 3-116. CD_L4PER2 Modules Wake-Up Request

Module	Wake-Up Feature
L4_PER2	None
MCAN	Slave wake-up request (IPU1-IRQ, DSP1-IRQ, DSP2-IRQ, EVE1-IRQ)
ATL	Slave wake-up request (IPU1-IRQ, DSP1-IRQ, DSP2-IRQ, EVE1-IRQ)
McASP2	Slave wake-up request (IPU1-IRQ, DSP1-IRQ, DSP2-IRQ, EVE1-IRQ)
McASP3	Slave wake-up request (IPU1-IRQ, DSP1-IRQ, DSP2-IRQ, EVE1-IRQ)
ADC	Slave wake-up request (IPU1-IRQ, DSP1-IRQ, DSP2-IRQ, EVE1-IRQ)
QSPI	Slave wake-up request (IPU1-IRQ, DSP1-IRQ, DSP2-IRQ, EVE1-IRQ)
PWMSS1	None

Table 3-117 lists the supported clock-management modes and associated software control bit fields for each module of the power domain.

Table 3-117. CD_L4PER2 Modules Clock-Management Modes and Control

Module	Clock-Management Protocol	Status Bit Field	Role
L4_PER2	Slave	CM_L4PER2_L4_PER2_CLKCTRL[17:16] IDLEST	Idle status
MCAN	Slave	CM_L4PER2_DCAN2_CLKCTRL[17:16] IDLEST	Idle status
ATL	Slave	CM_L4PER2_ATL_CLKCTRL[17:16] IDLEST	Idle status
McASP2	Slave	CM_L4PER2_MCASP4_CLKCTRL[17:16] IDLEST	Idle status
McASP3	Slave	CM_L4PER2_MCASP5_CLKCTRL[17:16] IDLEST	Idle status
ADC	Slave	CM_L4PER2_ADC_CLKCTRL[17:16] IDLEST	Idle status
QSPI	Slave	CM_L4PER2_QSPI_CLKCTRL[17:16] IDLEST	Idle status

Table 3-117. CD_L4PER2 Modules Clock-Management Modes and Control (continued)

Module	Clock-Management Protocol	Status Bit Field	Role
PWMSS1	Slave	CM_L4PER2_PWMSS1_CLKCTRL[17:16] IDLEST	Idle status

Table 3-118 lists the supported clock-management modes and associated software control bit fields for each module of the power domain.

Table 3-118. CD_L4PER2 Modules Slave Clock-Management Modes and Control

Module	Disabled	Auto	Enabled	Control Bit Field	Access Type
L4_PER2	N/A	Available	N/A	CM_L4PER2_L4_PER2_CLKCTRL[1:0] MODULEMODE	Read only
MCAN	Available	N/A	Available	CM_L4PER2_DCAN2_CLKCTRL[1:0] MODULEMODE	Read/write
ATL	Available	N/A	Available	CM_L4PER2_ATL_CLKCTRL[1:0] MODULEMODE	Read/write
McASP2	Available	N/A	Available	CM_L4PER2_MCASP4_CLKCTRL[1:0] MODULEMODE	Read/write
McASP3	Available	N/A	Available	CM_L4PER2_MCASP5_CLKCTRL[1:0] MODULEMODE	Read/write
QSPI	Available	N/A	Available	CM_L4PER2_QSPI_CLKCTRL[1:0] MODULEMODE	Read/write
ADC	Available	N/A	Available	CM_L4PER2_ADC_CLKCTRL[1:0] MODULEMODE	Read/write
PWMSS1	Available	N/A	Available	CM_L4PER2_PWMSS1_CLKCTRL[1:0] MODULEMODE	Read/write

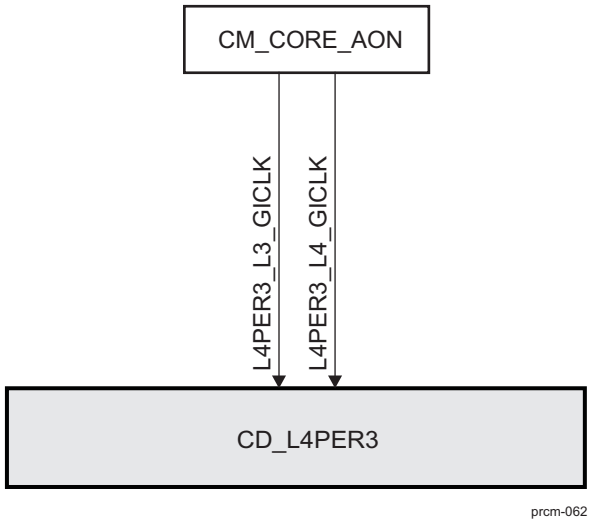
3.6.4.7 CD_L4PER3 Clock Domain

This section identifies the modes supported by the clock domain and the associated control and status bits. It also identifies its dependencies with other clock domains of the device.

3.6.4.7.1 Overview

Figure 3-51 is an overview of the clock domain.

Figure 3-51. CD_L4PER3 Overview



3.6.4.7.2 Clock Domain Modes

Table 3-119 lists the clock domain modes supported by the clock domain.

Table 3-119. CD_L4PER3 Clock Domain Modes

NO_SLEEP	SW_SLEEP	SW_WKUP	HW_AUTO
Available	Available	Available	Available

Table 3-120 lists the clock domain state transition control and status bits for the clock in this clock domain.

Table 3-120. CD_L4PER3 Control and Status Parameters

Parameter Name	Control/Status Bit Field
L4PER3_L3_GICLK clock status	CM_L4PER3_CLKSTCTRL[8] CLKACTIVITY_L4PER3_L3_GICLK
Clock Domain State Transition Control	CM_L4PER3_CLKSTCTRL[1:0] CLKTRCTRL

3.6.4.7.3 Clock Domain Dependency

CD_L4PER3 has no static dependency with any other clock domain of the device.

3.6.4.7.3.1 Dynamic Dependency

Table 3-121 lists the dynamic dependency of the clock domain with respect to other clock domains of the device.

Table 3-121. CD_L4PER3 Dynamic Dependency Association Parameters

Clock Domain Name	Default Setting	Control Bit Field	Access Type
CD_L4_CFG	Always enabled	CM_L4PER3_DYNAMICDEP[12] L4CFG_DYNDEP	Read only
CD_L3INIT	Always enabled	CM_L4PER3_DYNAMICDEP[7] L3INIT_DYNDEP	Read only
CD_IPU	Always enabled	CM_L4PER3_DYNAMICDEP[3] IPU_DYNDEP	Read only
CD_L3MAIN1	Always enabled	CM_L4PER3_DYNAMICDEP[5] L3MAIN1_DYNDEP	Read only
CD_CAM	Always enabled	CM_L4PER3_DYNAMICDEP[9] CAM_DYNDEP	Read only
CD_ISS	Always enabled	CM_L4PER3_DYNAMICDEP[31] ISS_DYNDEP	Read only

3.6.4.7.3.2 Wake-Up Dependency

CD_L4PER3 has no module wake-up dependency with any other clock domain of the device.

3.6.4.7.4 Clock Domain Module Attributes

Table 3-122 lists for each module of the clock domain the clocks the module receives and their role (that is, functional or interface clock).

Table 3-122. CD_L4PER3 Modules Clocks Association

Module	Clock	Clock Type
L4_PER3	L4PER3_L3_GICLK	Interface(1)
DCC1(2)	L4PER3_L3_GICLK	Interface(1)
DCC2(2)	L4PER3_L3_GICLK	Interface(1)
DCC3(2)	L4PER3_L3_GICLK	Interface(1)

(1) The L4 clock, required by the L4 interface, is created internally to the module. It is derived from the L4PER3_L3_GICLK clock divided by 2 using a gated version of L4_ICLK.
(2) DCC modules are not supported on the DRA78x family of devices.

Table 3-122. CD_L4PER3 Modules Clocks Association (continued)

Module	Clock	Clock Type
DCC4[2]	L4PER3_L3_GICLK	Interface[1]

Table 3-123 lists the supported wake-up request generation capability for each module of the clock domain.

Table 3-123. CD_L4PER3 Modules Wake-Up Request

Module	Wake-Up Feature
L4_PER3	None

Table 3-124 lists the supported clock-management modes and associated software control bit fields for each module of the power domain.

Table 3-124. CD_L4PER3 Modules Clock-Management Modes and Control

Module	Clock-Management Protocol	Status Bit Field	Role
L4_PER3	Slave	CM_L4PER3_L4_PER3_CLKCTRL[17:16] IDLEST	Idle status
DCC1[1]	Slave	CM_L4PER3_DCC1_CLKCTRL[17:16] IDLEST	Idle status
DCC2[1]	Slave	CM_L4PER3_DCC2_CLKCTRL[17:16] IDLEST	Idle status
DCC3[1]	Slave	CM_L4PER3_DCC3_CLKCTRL[17:16] IDLEST	Idle status
DCC4[1]	Slave	CM_L4PER3_DCC4_CLKCTRL[17:16] IDLEST	Idle status

(1) DCC modules are not supported on the DRA78x family of devices.

Table 3-125 lists the supported clock-management modes and associated software control bit fields for each module of the power domain.

Table 3-125. CD_L4PER3 Modules Slave Clock-Management Modes and Control

Module	Disabled	Auto	Enabled	Control Bit Field	Access Type
L4_PER3	N/A	Available	N/A	CM_L4PER3_L4_PER3_CLKCTRL[1:0] MODULEMODE	Read only
DCC1[1]	Available	N/A	Available	CM_L4PER3_DCC1_CLKCTRL[1:0] MODULEMODE	Read/write
DCC2[1]	Available	N/A	Available	CM_L4PER3_DCC2_CLKCTRL[1:0] MODULEMODE	Read/write
DCC3[1]	Available	N/A	Available	CM_L4PER3_DCC3_CLKCTRL[1:0] MODULEMODE	Read/write
DCC4[1]	Available	N/A	Available	CM_L4PER3_DCC4_CLKCTRL[1:0] MODULEMODE	Read/write

(1) DCC modules are not supported on the DRA78x family of devices.

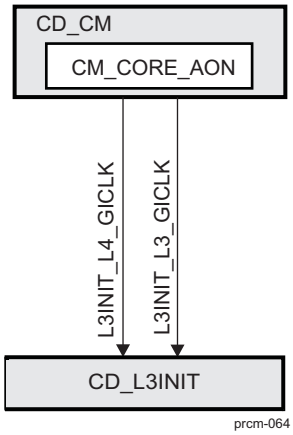
3.6.4.8 CD_L3INIT Clock Domain

This section identifies the modes supported by the clock domain and the associated control and status bits. It also identifies its dependencies with other clock domains of the device.

3.6.4.8.1 Overview

Figure 3-52 is an overview of the clock domain.

Figure 3-52. CD_L3INIT Overview



3.6.4.8.2 Clock Domain Modes

Table 3-126 lists the clock domain modes supported by the clock domain.

Table 3-126. CD_L3INIT Clock Domain Modes

NO_SLEEP	SW_SLEEP	SW_WKUP	HW_AUTO
Available	Available	Available	Available

Table 3-127 lists the clock domain state transition control and status bits for the clock in this clock domain.

Table 3-127. CD_L3INIT Control and Status Parameters

Parameter Name	Control/Status Bit Field
L3INIT_L3_GICLK Clock Status	CM_L3INIT_CLKSTCTRL[8] CLKACTIVITY_L3INIT_L3_GICLK
L3INIT_L4_GICLK Clock Status	CM_L3INIT_CLKSTCTRL[9] CLKACTIVITY_L3INIT_L4_GICLK
Clock Domain State Transition Control	CM_L3INIT_CLKSTCTRL[1:0] CLKTRCTRL

3.6.4.8.3 Clock Domain Dependency

3.6.4.8.3.1 Static Dependency

Table 3-128 lists the static dependency of the clock domain with respect to other clock domains of the device.

Table 3-128. CD_L3INIT Static Dependency Association Parameters

Clock Domain Name	Default Setting	Control Bit Field	Access Type
CD_L3_MAIN1	Always enabled	CM_L3INIT_STATICDEP[5] L3MAIN1_STATDEP	Read only
CD_L4_CFG	Disabled	CM_L3INIT_STATICDEP[12] L4CFG_STATDEP	Read/write
CD_L4PER	Disabled	CM_L3INIT_STATICDEP[13] L4PER_STATDEP	Read/write
CD_EMIF	Disabled	CM_L3INIT_STATICDEP[4] EMIF_STATDEP	Read/write
CD_WKUPAON	Disabled	CM_L3INIT_STATICDEP[15] WKUPAON_STATDEP	Read/write
CD_L4PER3	Disabled	CM_L3INIT_STATICDEP[27] L4PER3_STATDEP	Read/write

3.6.4.8.3.2 Dynamic Dependency

Table 3-129 lists the dynamic dependency of the clock domain with respect to other clock domains of the device.

Table 3-129. CD_L3INIT Dynamic Dependency Association Parameters

Clock Domain Name	Default Setting	Control Bit Field	Access Type
CD_L3_MAIN1	Always disabled	CM_L3INIT_DYNAMICDEP[5] L3MAIN1_DYNDEP	Read only

3.6.4.8.3.3 Wake-Up Dependency

CD_L3INIT has no module wake-up dependency with any other clock domain of the device.

3.6.4.8.4 Clock Domain Module Attributes

Table 3-130 lists for each module of the clock domain the clocks the module receives and their role (that is, functional or interface clock).

Table 3-130. CD_L3INIT Modules Clocks Association

Module	Clock	Clock Type
IEEE1500_2_OCP	L3INIT_L3_GICLK	Interface and functional(1)

(1) The L4 clock, required by the L4 interface, is created internally to the module. It is derived from the L3INIT_L3_GICLK clock divided by 2 using a gated version of L4_ICLK.

Table 3-131 lists the supported wake-up request generation capability for each module of the clock domain.

Table 3-131. CD_L3INIT Modules Wake-Up Request

Module	Wake-Up Feature
IEEE1500_2_OCP	Master wake-up request

Table 3-132 lists the supported clock-management modes and associated software control bit fields for each module of the power domain.

Table 3-132. CD_L3INIT Modules Clock-Management Modes and Control

Module	Clock-Management Protocol	Status Bit Field	Role
IEEE1500_2_OCP	Slave/master	CM_L3INIT_IEEE1500_2_OCP_CLKCTRL[18] STBYST	Standby status
		CM_L3INIT_IEEE1500_2_OCP_CLKCTRL[17:16] IDLEST	Idle status

Table 3-133 lists the supported clock-management modes and associated software control bit fields for each module of the power domain.

Table 3-133. CD_L3INIT Modules Slave Clock-Management Modes and Control

Module	Disabled	Auto	Enabled	Control Bit Field	Access Type
IEEE1500_2_OCP	N/A	Available	N/A	CM_L3INIT_IEEE1500_2_OCP_CLKCTRL[1:0] MODULEMODE	Read only

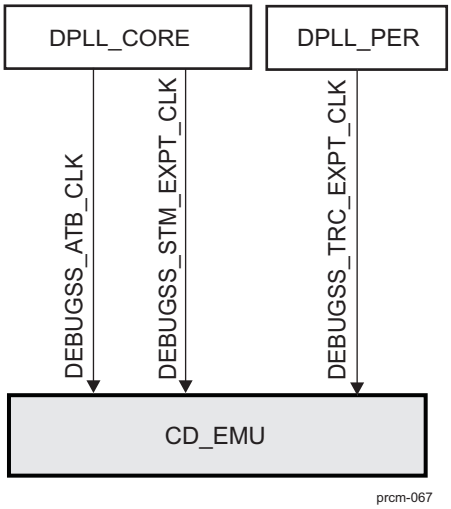
3.6.4.9 CD_EMU Clock Domain

This section identifies the modes supported by the clock domain and the associated control and status bits. It also identifies its dependencies with other clock domains of the device.

3.6.4.9.1 Overview

Figure 3-53 is an overview of the clock domain.

Figure 3-53. CD_EMU Overview



3.6.4.9.2 Clock Domain Modes

Table 3-134 lists the clock domain modes supported by the clock domain.

Table 3-134. CD_EMU Clock Domain Modes

NO_SLEEP	SW_SLEEP	SW_WKUP	HW_AUTO
Not available	Not available	Available	Available

Table 3-135 lists the clock domain state transition control and status bits for the clock in this clock domain.

Table 3-135. CD_EMU Control and Status Parameters

Parameter Name	Control/Status Bit Field
EMU_SYS_GCLK Clock Status	CM_EMU_CLKSTCTRL[8] CLKACTIVITY_EMU_SYS_CLK
Clock Domain State Transition Control	CM_EMU_CLKSTCTRL[1:0] CLKTRCTRL

3.6.4.9.3 Clock Domain Dependency

CD_EMU has no static or module wake-up dependency with any other clock domain of the device.

3.6.4.9.3.1 Dynamic Dependency

Table 3-136 lists the dynamic dependency of the clock domain with respect to other clock domains of the device.

Table 3-136. CD_EMU Dynamic Dependency Association Parameters

Clock Domain Name	Default Setting	Control Bit Field	Access Type
CD_L3_MAIN1	Always enabled	CM_EMU_DYNAMICDEP[5] L3MAIN1_DYNDEP	Read only

3.6.4.9.4 Clock Domain Module Attributes

Table 3-137 lists for each module of the clock domain the clocks the module receives and their role (that is, functional or interface clock).

Table 3-137. CD_EMU Modules Clocks Association

Module	Clock	Clock Type
DEBUGSS	DEBUGSS_ATB_CLK	Functional
	DEBUGSS_TRC_EXPT_CLK	Functional
	DEBUGSS_STM_EXPT_CLK	Functional

Table 3-138 lists the supported wake-up request generation capability for each module of the clock domain.

Table 3-138. CD_EMU Modules Wake-Up Request

Module	Wake-Up Feature
DEBUGSS	Master wake-up request

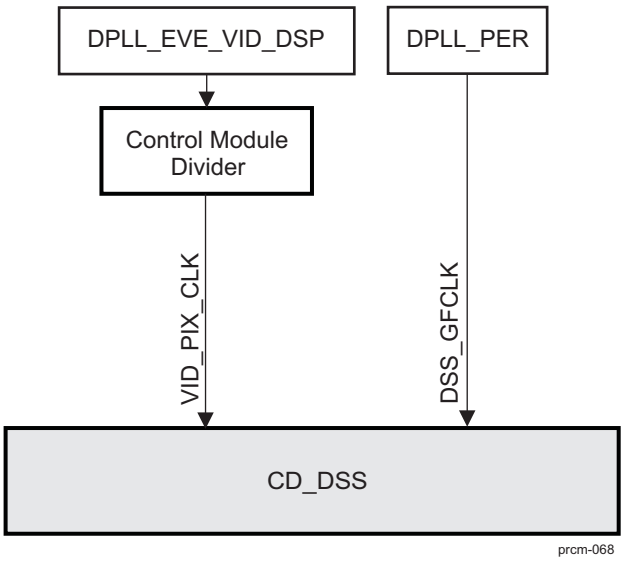
3.6.4.10 CD_DSS Clock Domain

This section identifies the modes supported by the clock domain and the associated control and status bits. It also identifies its dependencies with other clock domains of the device.

3.6.4.10.1 Overview

Figure 3-54 is an overview of the clock domain.

Figure 3-54. CD_DSS Overview



3.6.4.10.2 Clock Domain Modes

Table 3-139 lists the clock domain modes supported by the clock domain.

Table 3-139. CD_DSS Clock Domain Modes

NO_SLEEP	SW_SLEEP	SW_WKUP	HW_AUTO
Available	Available	Available	Available

Table 3-140 lists the clock domain state transition control and status bits for the clock in this clock domain.

Table 3-140. CD_DSS Control and Status Parameters

Parameter Name	Control/Status Bit Field
DSS_GFCLK Clock Status	CM_DSS_CLKSTCTRL[9] CLKACTIVITY_DSS_GFCLK
Clock Domain State Transition Control	CM_DSS_CLKSTCTRL[1:0] CLKTRCTRL

3.6.4.10.3 Clock Domain Dependency

3.6.4.10.3.1 Static Dependency

Table 3-141 lists the static dependency of the clock domain with respect to other clock domains of the device.

Table 3-141. CD_DSS Static Dependency Association Parameters

Clock Domain Name	Default Setting	Control Bit Field	Access Type
CD_L3_MAIN1	Always enabled	CM_DSS_STATICDEP[5] L3MAIN1_STATDEP	Read only
CD_EMIF	Disabled	CM_DSS_STATICDEP[4] EMIF_STATDEP	Read/write

3.6.4.10.3.2 Dynamic Dependency

Table 3-142 lists the dynamic dependency of the clock domain with respect to other clock domains of the device.

Table 3-142. CD_DSS Dynamic Dependency Association Parameters

Clock Domain Name	Default Setting	Control Bit Field	Access Type
CD_L3_MAIN1	Alway disabled	CM_DSS_DYNAMICDEP[5] L3MAIN1_DYNDEP	Read only

3.6.4.10.3.3 Wake-Up Dependency

Table 3-143 lists the wake-up dependency settings for the modules of this clock in the clock domain

Table 3-143. CD_DSS Wake-Up Dependency Association Parameters

Originator Module	Originator Clock Domain	Servicing Clock Domain	Default Setting	Control Bit Field	Access Type
DSS	CD_DSS	CD_DSP, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_DSS_DSS_WKDEP[2] WKUPDEP_DISPC_DSP1	Read/write
DSS	CD_DSS	CD_DSP2, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_DSS_DSS_WKDEP[5] WKUPDEP_DISPC_DSP2	Read/write
DSS	CD_DSS	CD_IPU1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_DSS_DSS_WKDEP[4] WKUPDEP_DISPC_IPU1	Read/write
DSS	CD_DSS	CD_EVE1, CD_L3_MAIN1, CD_L4PER1, CD_L4PER2, CD_L4PER3	Disabled	PM_DSS_DSS_WKDEP[6] WKUPDEP_DISPC_EVE1	Read/write

3.6.4.10.4 Clock Domain Module Attributes

Table 3-144 lists for each module of the clock domain the clocks the module receives and their role (that is, functional or interface clock).

Table 3-144. CD_DSS Modules Clocks Association

Module	Clock	Clock Type
DSS	DSS_GFCLK	Functional
	VID_PIX_CLK	Functional and Interface

Table 3-145 lists the supported wake-up request generation capability for each module of the clock domain.

Table 3-145. CD_DSS Modules Wake-Up Request

Module	Wake-Up Feature
DSS	Slave wake-up request (IPU1-IRQ, DSP1-IRQ, DSP2-IRQ, EVE1-IRQ)

Table 3-146 lists the supported clock-management modes and associated software control bit fields for each module of the power domain.

Table 3-146. CD_DSS Modules Clock-Management Modes and Control

Module	Clock-Management Protocol	Status Bit Field	Role
DSS	Master/slave	CM_DSS_DSS_CLKCTRL[18] STBYST	Standby status
		CM_DSS_DSS_CLKCTRL[17:16] IDLEST	Idle status

Table 3-147 lists the supported clock-management modes and associated software control bit fields for each module of the power domain.

Table 3-147. CD_DSS Modules Slave Clock-Management Modes and Control

Module	Disabled	Auto	Enabled	Control Bit Field	Access Type
DSS	Available	N/A	Available	CM_DSS_DSS_CLKCTRL[1:0] MODULEMODE	Read/write

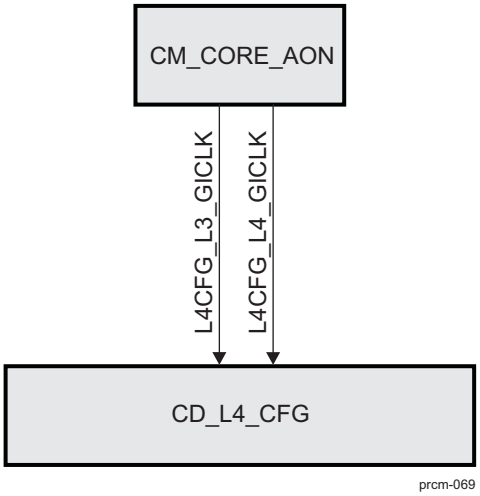
3.6.4.11 CD_L4_CFG Clock Domain

This section identifies the modes supported by the clock domain and the associated control and status bits. It also identifies its dependencies with other clock domains of the device.

3.6.4.11.1 Overview

Figure 3-55 is an overview of the clock domain.

Figure 3-55. CD_L4_CFG Overview



3.6.4.11.2 Clock Domain Modes

Table 3-148 lists the clock domain modes supported by the clock domain.

Table 3-148. CD_L4_CFG Clock Domain Modes

NO_SLEEP	SW_SLEEP	SW_WKUP	HW_AUTO
Available	Not available	Not available	Available

Table 3-149 lists the clock domain state transition control and status bits for the clock in this clock domain.

Table 3-149. CD_L4_CFG Control and Status Parameters

Parameter Name	Control/Status Bit Field
L4CFG_L4_GICLK Clock Status	CM_L4CFG_CLKSTCTRL[8] CLKACTIVITY_L4CFG_L4_GICLK
L4CFG_L3_GICLK Clock Status	CM_L4CFG_CLKSTCTRL[9] CLKACTIVITY_L4CFG_L3_GICLK
Clock Domain State Transition Control	CM_L4CFG_CLKSTCTRL[1:0] CLKTRCTRL

3.6.4.11.3 Clock Domain Dependency

CD_L4_CFG has no static or module wake-up dependency with any other clock domain of the device.

3.6.4.11.3.1 Dynamic Dependency

Table 3-150 lists the dynamic dependency of the clock domain with respect to other clock domains of the device.

Table 3-150. CD_L4_CFG Dynamic Dependency Association Parameters

Clock Domain Name	Default Setting	Control Bit Field	Access Type
CD_CUSTEFUSE	Always enabled	CM_L4CFG_DYNAMICDEP[17] CUSTEFUSE_DYNDEP	Read only
CD_L3_MAIN1	Always enabled	CM_L4CFG_DYNAMICDEP[5] L3MAIN1_DYNDEP	Read only
CD_L3INIT	Always enabled	CM_L4CFG_DYNAMICDEP[7] L3INIT_DYNDEP	Read only
CD_EMIF	Always enabled	CM_L4CFG_DYNAMICDEP[4] EMIF_DYNDEP	Read only
CD_COREAON	Always enabled	CM_L4CFG_DYNAMICDEP[16] COREAON_DYNDEP	Read only

3.6.4.11.4 Clock Domain Module Attributes

Table 3-151 lists for each module of the clock domain the clocks the module receives and their role (that is, functional or interface clock).

Table 3-151. CD_L4_CFG Modules Clocks Association

Module	Clock	Clock Type
SPINLOCK	L4CFG_L3_GICLK	Functional and Interface(1)
L4_CFG interconnect	L4CFG_L3_GICLK	Interface(1)
MAILBOX1	L4CFG_L3_GICLK	Interface(1)
MAILBOX2	L4CFG_L3_GICLK	Interface(1)

(1) The L4 clock, required by the L4 interface, is created internally to the module. It is derived from the L4CFG_L3_GICLK clock divided by 2 using a gated version of L4_ICLK.

Table 3-152 lists the supported wake-up request generation capability for each module of the clock domain.

Table 3-152. CD_L4_CFG Modules Wake-Up Request

Module	Wake-Up Feature
SPINLOCK	None
L4_CFG interconnect	None
MAILBOX1	None
MAILBOX2	None

Table 3-153 lists the supported clock-management modes and associated software control bit fields for each module of the power domain.

Table 3-153. CD_L4_CFG Modules Clock-Management Modes and Control

Module	Clock-Management Protocol	Status Bit Field	Role
SPINLOCK	Slave	CM_L4CFG_SPINLOCK_CLKCTRL[17:16] IDLEST	Idle status
L4_CFG interconnect	Slave	CM_L4CFG_L4_CFG_CLKCTRL[17:16] IDLEST	Idle status
MAILBOX1	Slave	CM_L4CFG_MAILBOX1_CLKCTRL[17:16] IDLEST	Idle status
MAILBOX2	Slave	CM_L4CFG_MAILBOX2_CLKCTRL[17:16] IDLEST	Idle status

Table 3-154 lists the supported clock-management modes and associated software control bit fields for each module of the power domain.

Table 3-154. CD_L4_CFG Modules Slave Clock-Management Modes and Control

Module	Disabled	Auto	Enabled	Control Bit Field	Access Type
SPINLOCK	N/A	Available	N/A	CM_L4CFG_SPINLOCK_CLKCTRL[1:0] MODULEMODE	Read only
L4_CFG interconnect	N/A	Available	N/A	CM_L4CFG_L4_CFG_CLKCTRL[1:0] MODULEMODE	Read only
MAILBOX1	N/A	Available	N/A	CM_L4CFG_MAILBOX1_CLKCTRL[1:0] MODULEMODE	Read only
MAILBOX2	N/A	Available	N/A	CM_L4CFG_MAILBOX2_CLKCTRL[1:0] MODULEMODE	Read only

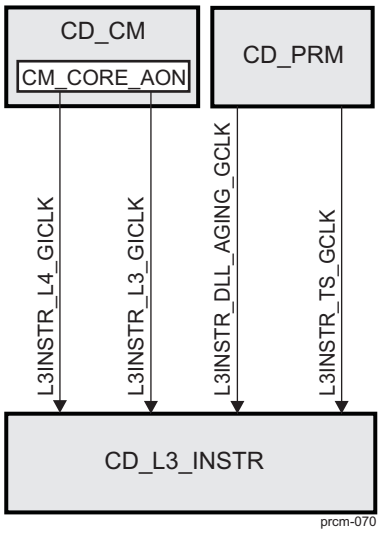
3.6.4.12 CD_L3_INSTR Clock Domain

This section identifies the modes supported by the clock domain and the associated control and status bits. It also identifies its dependencies with other clock domains of the device.

3.6.4.12.1 Overview

Figure 3-56 is an overview of the clock domain.

Figure 3-56. CD_L3_INSTR Overview



3.6.4.12.2 Clock Domain Modes

Table 3-155 lists the clock domain modes supported by the clock domain.

Table 3-155. CD_L3_INSTR Clock Domain Modes

NO_SLEEP	SW_SLEEP	SW_WKUP	HW_AUTO
Not available	Not available	Not available	Available

Table 3-156 lists the clock domain state transition control and status bits for the clock in this clock domain.

Table 3-156. CD_L3_INSTR Control and Status Parameters

Parameter Name	Control/Status Bit Field
L3INSTR_L3_GICLK Clock Status	CM_L3INSTR_CLKSTCTRL[8] CLKACTIVITY_L3INSTR_L3_GICLK
L3INSTR_DLL_AGING_GCLK Clock Status	CM_L3INSTR_CLKSTCTRL[9] CLKACTIVITY_L3INSTR_DLL_AGING_GCLK
L3INSTR_TS_GCLK Clock Status	CM_L3INSTR_CLKSTCTRL[10] CLKACTIVITY_L3INSTR_TS_GCLK
Clock Domain State Transition Control	CM_L3INSTR_CLKSTCTRL[1:0] CLKTRCTRL

3.6.4.12.3 Clock Domain Dependency

CD_L3_INSTR has no static, dynamic, or module wake-up dependency with any other clock domain of the device.

For aging, DLL requires a low-frequency clock that must run as long as the CORE power domain is on. To match this requirement, the DLL_AGING module has been created and instantiated in this clock domain, which is the last to transition in low-power state.

3.6.4.12.4 Clock Domain Module Attributes

Table 3-157 lists for each module of the clock domain the clocks the module receives and their role (that is, functional or interface clock).

Table 3-157. CD_L3_INSTR Modules Clocks Association

Module	Clock	Clock Type
L3_MAIN_2 interconnect	L3INSTR_L3_GICLK	Interface(1)
L3_INSTR interconnect	L3INSTR_L3_GICLK	Interface
OCP_WP_NOC	L3INSTR_L3_GICLK	Interface(1)
DLL_AGING	L3INSTR_DLL_AGING_GCLK	Functional
CTRL_MODULE_BANDGAP	L3INSTR_TS_GCLK	Functional

(1) The L4 clock, required by the L4 interface, is created internally to the module. It is derived from the L3INSTR_L3_GICLK clock divided by 2 using a gated version of L4_ICLK.

Table 3-158 lists the supported wake-up request generation capability for each module of the clock domain.

Table 3-158. CD_L3_INSTR Modules Wake-Up Request

Module	Wake-Up Feature
L3_MAIN_2 interconnect	None
L3_INSTR interconnect	None
OCP_WP_NOC	None
DLL_AGING	None
CTRL_MODULE_BANDGAP	None

Table 3-159 lists the supported clock-management modes and associated software control bit fields for each module of the power domain.

Table 3-159. CD_L3_INSTR Modules Clock-Management Modes and Control

Module	Clock-Management Protocol	Status Bit Field	Role
L3_MAIN_3 interconnect	Slave	CM_L3INSTR_L3_MAIN_2_CLKCTRL[17:16] IDLEST	Idle status
		CM_L3INSTR_L3_MAIN_2_CLKCTRL_RESTORE[17:16] IDLEST	Idle status
L3_INSTR interconnect	Slave	CM_L3INSTR_L3_INSTR_CLKCTRL[17:16] IDLEST	Idle status
		CM_L3INSTR_L3_INSTR_CLKCTRL_RESTORE[17:16] IDLEST	Idle status
OCP_WP_NOC	Slave	CM_L3INSTR_OCP_WP_NOC_CLKCTRL[17:16] IDLEST	Idle status
		CM_L3INSTR_OCP_WP_NOC_CLKCTRL_RESTORE[17:16] IDLEST	Idle status
DLL_AGING	Slave	CM_L3INSTR_DLL_AGING_CLKCTRL[17:16] IDLEST	Idle status
CTRL_MODULE_BANDGAP	Slave	CM_L3INSTR_CTRL_MODULE_BANDGAP_CLKCTRL[17:16] IDLEST	Idle status

Table 3-160 lists the supported clock-management modes and associated software control bit fields for each module of the power domain.

Table 3-160. CD_L3_INSTR Modules Slave Clock-Management Modes and Control

Module	Disabled	Auto	Enabled	Control Bit Field	Access Type
L3_MAIN_2 interconnect	Available	Available	N/A	CM_L3INSTR_L3_MAIN_2_CLKCTRL[1:0] MODULEMODE	Read/write
L3_MAIN_2 interconnect	Available	Available	N/A	CM_L3INSTR_L3_MAIN_2_CLKCTRL_RESTORE[1:0] MODULEMODE	Read/write
L3_INSTR interconnect	Available	Available	N/A	CM_L3INSTR_L3_INSTR_CLKCTRL[1:0] MODULEMODE	Read/write
L3_INSTR interconnect	Available	Available	N/A	CM_L3INSTR_L3_INSTR_CLKCTRL_RESTORE[1:0] MODULEMODE	Read/write
OCP_WP_NOC	Available	Available	N/A	CM_L3INSTR_OCP_WP_NOC_CLKCTRL[1:0] MODULEMODE	Read/write
OCP_WP_NOC	Available	Available	N/A	CM_L3INSTR_OCP_WP_NOC_CLKCTRL_RESTORE[1:0] MODULEMODE	Read/write
DLL_AGING	N/A	Available	N/A	CM_L3INSTR_DLL_AGING_CLKCTRL[1:0] MODULEMODE	Read only
CTRL_MODULE_BANDGAP	N/A	Available	N/A	CM_L3INSTR_CTRL_MODULE_BANDGAP_CLKCTRL[1:0] MODULEMODE	Read only

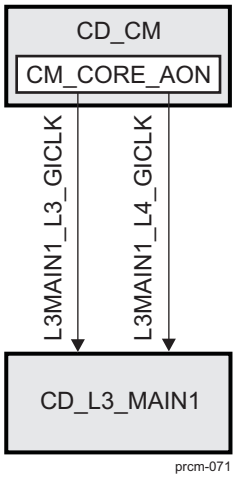
3.6.4.13 CD_L3_MAIN1 Clock Domain

This section identifies the modes supported by the clock domain and the associated control and status bits. It also identifies its dependencies with other clock domains of the device.

3.6.4.13.1 Overview

Figure 3-57 is an overview of the clock domain.

Figure 3-57. CD_L3_MAIN1 Overview



3.6.4.13.2 Clock Domain Modes

Table 3-161 lists the clock domain modes supported by the clock domain.

Table 3-161. CD_L3_MAIN1 Clock Domain Modes

NO_SLEEP	SW_SLEEP	SW_WKUP	HW_AUTO
Available	Not available	Not available	Available

Table 3-162 lists the clock domain state transition control and status bits for the clock in this clock domain.

Table 3-162. CD_L3_MAIN1 Control and Status Parameters

Parameter Name	Control/Status Bit Field
L3MAIN1_L3_GICLK Clock Status	CM_L3MAIN1_CLKSTCTRL[8] CLKACTIVITY_L3MAIN1_L3_GICLK
L3MAIN1_L4_GICLK Clock Status	CM_L3MAIN1_CLKSTCTRL[9] CLKACTIVITY_L3MAIN1_L4_GICLK
Clock Domain State Transition Control	CM_L3MAIN1_CLKSTCTRL[1:0] CLKTRCTRL

3.6.4.13.3 Clock Domain Dependency

CD_L3_MAIN1 has no static or module wake-up dependency with any other clock domain of the device.

3.6.4.13.3.1 Dynamic Dependency

Table 3-163 lists the dynamic dependency of the clock domain with respect to other clock domains of the device.

Table 3-163. CD_L3_MAIN1 Dynamic Dependency Association Parameters

Clock Domain Name	Default Setting	Control Bit Field	Access Type
CD_IPU	Always enabled	CM_L3MAIN1_DYNAMICDEP[3] IPU_DYNDEP	Read only
CD_IPU1	Always enabled	CM_L3MAIN1_DYNAMICDEP[18] IPU1_DYNDEP	Read only
CD_DSP1	Always enabled	CM_L3MAIN1_DYNAMICDEP[1] DSP1_DYNDEP	Read only
CD_DSP2	Always enabled	CM_L3MAIN1_DYNAMICDEP[20] DSP2_DYNDEP	Read only
CD_EMIF	Always enabled	CM_L3MAIN1_DYNAMICDEP[4] EMIF_DYNDEP	Read only
CD_DSS	Always enabled	CM_L3MAIN1_DYNAMICDEP[8] DSS_DYNDEP	Read only
CD_L4PER	Always enabled	CM_L3MAIN1_DYNAMICDEP[13] L4PER_DYNDEP	Read only
CD_L4PER2	Always enabled	CM_L3MAIN1_DYNAMICDEP[22] L4PER2_DYNDEP	Read only
CD_L4PER3	Always enabled	CM_L3MAIN1_DYNAMICDEP[23] L4PER3_DYNDEP	Read only
CD_L4_CFG	Always enabled	CM_L3MAIN1_DYNAMICDEP[12] L4CFG_DYNDEP	Read only
CD_WKUPAON	Always enabled	CM_L3MAIN1_DYNAMICDEP[15] WKUPAON_DYNDEP	Read only
CD_EVE1	Always enabled	CM_L3MAIN1_DYNAMICDEP[28] EVE1_DYNDEP	Read only

3.6.4.13.4 Clock Domain Module Attributes

Table 3-164 lists for each module of the clock domain the clocks the module receives and their role (that is, functional or interface clock).

Table 3-164. CD_L3_MAIN1 Modules Clocks Association

Module	Clock	Clock Type
L3_MAIN_1 interconnect	L3MAIN1_L3_GICLK	Interface(1)
GPMC	L3MAIN1_L3_GICLK	Interface and Functional
OCMC_RAM1	L3MAIN1_L3_GICLK	Interface(1)
TESOC(2)	L3MAIN1_L3_GICLK	Interface
	TESOC_EXT_CLK	Functional
MMU_EDMA	L3MAIN1_L3_GICLK	Interface(1)
EDMA_TPCC	L3MAIN1_L3_GICLK	Interface

(1) The L4 clock, required by the L4 interface, is created internally to the module. It is derived from the L3MAIN1_L3_GICLK clock divided by 2 using a gated version of L4_ICLK.

(2) TESOC is not supported on the DRA78x family of devices.

Table 3-164. CD_L3_MAIN1 Modules Clocks Association (continued)

Module	Clock	Clock Type
EDMA_TC0	L3MAIN1_L3_GICLK	Interface
EDMA_TC1	L3MAIN1_L3_GICLK	Interface

Table 3-165 lists the supported wake-up request generation capability for each module of the clock domain.

Table 3-165. CD_L3_MAIN1 Modules Wake-Up Request

Module	Wake-Up Feature
L3_MAIN1 interconnect	None
GPMC	None
OCMC_RAM1	Slave wake-up request (IPU1-IRQ, DSP1-IRQ, DSP2-IRQ, EVE1-IRQ)
TESOC(1)	Slave wake-up request (IPU1-IRQ, DSP1-IRQ, DSP2-IRQ, EVE1-IRQ)
MMU_EDMA	None
EDMA_TPCC	Slave wake-up request (IPU1-IRQ, DSP1-IRQ, DSP2-IRQ, EVE1-IRQ)
EDMA_TC0	Slave wake-up request (IPU1-IRQ, DSP1-IRQ, DSP2-IRQ, EVE1-IRQ)/ Master wake-up request
EDMA_TC1	Slave wake-up request (IPU1-IRQ, DSP1-IRQ, DSP2-IRQ, EVE1-IRQ)/ Master wake-up request

(1) TESOC is not supported on the DRA78x family of devices.

Table 3-166 lists the supported clock-management modes and associated software control bit fields for each module of the power domain.

Table 3-166. CD_L3_MAIN1 Modules Clock-Management Modes and Control

Module	Clock-Management Protocol	Status Bit Field	Role
L3_MAIN1 interconnect	Slave	CM_L3MAIN1_L3_MAIN_1_CLKCTRL[17:16] IDLEST	Idle status
GPMC	Slave	CM_L3MAIN1_GPMC_CLKCTRL[17:16] IDLEST	Idle status
OCMC_RAM1	Slave	CM_L3MAIN1_OCMC_RAM1_CLKCTRL[17:16] IDLEST	Idle status
TESOC(1)	Slave	CM_L3MAIN1_TESOC_CLKCTRL[17:16] IDLEST	Idle status
MMU_EDMA	Slave	CM_L3MAIN1_MMU_EDMA_CLKCTRL[17:16] IDLEST	Idle status
EDMA_TPCC	Slave	CM_L3MAIN1_TPCC_CLKCTRL[17:16] IDLEST	Idle status
EDMA_TC0	Master/slave	CM_L3MAIN1_TPTC1_CLKCTRL[18] STBYST	Standby status
		CM_L3MAIN1_TPTC1_CLKCTRL[17:16] IDLEST	Idle status
EDMA_TC1	Master/slave	CM_L3MAIN1_TPTC2_CLKCTRL[18] STBYST	Standby status
		CM_L3MAIN1_TPTC2_CLKCTRL[17:16] IDLEST	Idle status

(1) TESOC is not supported on the DRA78x family of devices.

Table 3-167 lists the supported clock-management modes and associated software control bit fields for each module of the power domain.

Table 3-167. CD_L3_MAIN1 Modules Slave Clock-Management Modes and Control

Module	Disabled	Auto	Enabled	Control Bit Field	Access Type
L3_MAIN1 interconnect	N/A	Available	N/A	CM_L3MAIN1_L3_MAIN_1_CLKCTRL[1:0] MODULEMODE	Read only

Table 3-167. CD_L3_MAIN1 Modules Slave Clock-Management Modes and Control (continued)

Module	Disabled	Auto	Enabled	Control Bit Field	Access Type
GPMC	Available	Available	N/A	CM_L3MAIN1_GPMC_CLKCTRL [1:0] MODULEMODE	Read/write
OCMC_RAM1	N/A	Available	N/A	CM_L3MAIN1_OCMC_RAM1_CLKCTRL [1:0] MODULEMODE	Read only
TESOC[(1)]	N/A	Available	N/A	CM_L3MAIN1_TESOC_CLKCTRL [1:0] MODULEMODE	Read only
MMU_EDMA	N/A	Available	N/A	CM_L3MAIN1_MMU_EDMA_CLKCTRL[1:0] MODULEMODE	Read only
EDMA_TPCC	N/A	Available	N/A	CM_L3MAIN1_TPCC_CLKCTRL[1:0] MODULEMODE	Read only
EDMA_TC0	Available	Available	N/A	CM_L3MAIN1_TPTC1_CLKCTRL[1:0] MODULEMODE	Read/write
EDMA_TC1	Available	Available	N/A	CM_L3MAIN1_TPTC2_CLKCTRL[1:0] MODULEMODE	Read/write

(1) TESOC is not supported on the DRA78x family of devices.

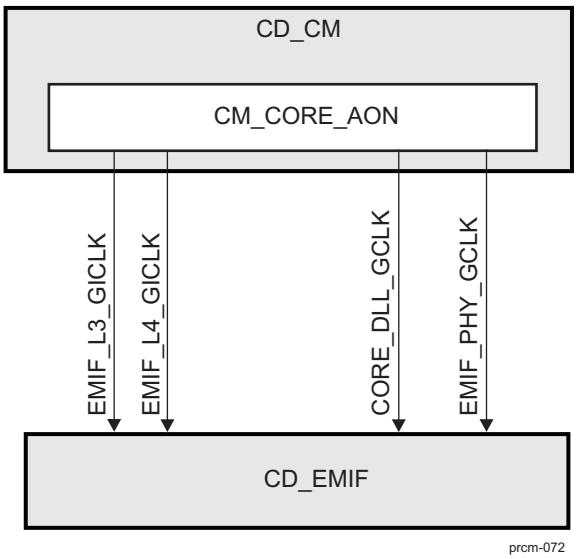
3.6.4.14 CD_EMIF Clock Domain

This section identifies the modes supported by the clock domain and the associated control and status bits. It also identifies its dependencies with other clock domains of the device.

3.6.4.14.1 Overview

Figure 3-58 is an overview of the clock domain.

Figure 3-58. CD_EMIF Overview



3.6.4.14.2 Clock Domain Modes

Table 3-168 lists the clock domain modes supported by the clock domain.

Table 3-168. CD_EMIF Clock Domain Modes

NO_SLEEP	SW_SLEEP	SW_WKUP	HW_AUTO
Available	Not available	Available	Available

Table 3-169 lists the clock domain state transition control and status bits for the clock in this clock domain.

Table 3-169. CD_EMIF Control and Status Parameters

Parameter Name	Control/Status Bit Field
CORE_DLL_GCLK Clock Status	CM_EMIF_CLKSTCTRL[9] CLKACTIVITY_EMIF_DLL_GCLK
CORE_DLL_GCLK Clock Control	CM_EMIF_EMIF_DLL_CLKCTRL[8] OPTFCLKEN_DLL_CLK
EMIF_L3_GICLK Clock Status	CM_EMIF_CLKSTCTRL[8] CLKACTIVITY_EMIF_L3_GICLK
EMIF_PHY_GCLK Clock Status	CM_EMIF_CLKSTCTRL[10] CLKACTIVITY_EMIF_PHY_GCLK
Clock Domain State Transition Control	CM_EMIF_CLKSTCTRL[1:0] CLKTRCTRL

3.6.4.14.3 Clock Domain Dependency

CD_EMIF has no static, dynamic, or module wake-up dependency with any other clock domain of the device.

3.6.4.14.4 Clock Domain Module Attributes

Table 3-170 lists for each module of the clock domain the clocks the module receives and their role (that is, functional or interface clock).

Table 3-170. CD_EMIF Modules Clocks Association

Module	Clock	Clock Type
DLL	EMIF_DLL_GCLK	Functional
EMIF1	EMIF_PHY_GCLK	Interface
	EMIF_L3_GICLK	Interface
	L3_EOCP_GICLK	Interface

Table 3-171 lists the supported wake-up request generation capability for each module of the clock domain.

Table 3-171. CD_EMIF Modules Wake-Up Request

Module	Wake-Up Feature
DLL	None
EMIF1	None

Table 3-172 lists the supported clock-management modes and associated software control bit fields for each module of the power domain.

Table 3-172. CD_EMIF Modules Clock-Management Modes and Control

Module	Clock-Management Protocol	Status Bit Field	Role
EMIF1	Slave	CM_EMIF_EMIF1_CLKCTRL[17:16] IDLEST	Idle status

Table 3-173 lists the supported clock-management modes and associated software control bit fields for each module of the power domain.

Table 3-173. CD_EMIF Modules Slave Clock-Management Modes and Control

Module	Disabled	Auto	Enabled	Control Bit Field	Access Type
EMIF1	Available	Available	N/A	CM_EMIF_EMIF1_CLKCTRL[1:0] MODULEMODE	Read/write

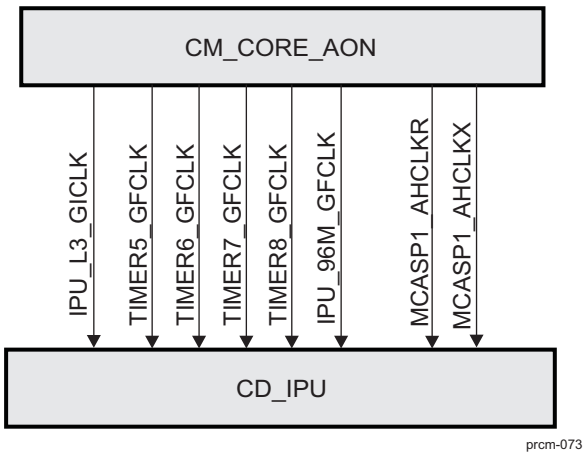
3.6.4.15 CD_IPU Clock Domain

This section identifies the modes supported by the clock domain and the associated control and status bits. It also identifies its dependencies with other clock domains of the device.

3.6.4.15.1 Overview

Figure 3-59 is an overview of the clock domain.

Figure 3-59. CD_IPU Overview



3.6.4.15.2 Clock Domain Modes

Table 3-174 lists the clock domain modes supported by the clock domain.

Table 3-174. CD_IPU Clock Domain Modes

NO_SLEEP	SW_SLEEP	SW_WKUP	HW_AUTO
Available	Available	Available	Available

Table 3-175 lists the clock domain state transition control and status bits for the clock in this clock domain.

Table 3-175. CD_IPU Control and Status Parameters

Parameter Name	Control/Status Bit Field
IPU_L3_GICLK Clock Status	CM_IPU_CLKSTCTRL[8] CLKACTIVITY_IPU_L3_GICLK
IPU_96M_GFCLK Clock Status	CM_IPU_CLKSTCTRL[13] CLKACTIVITY_IPU_96M_GFCLK
TIMER5_GFCLK Clock Status	CM_IPU_CLKSTCTRL[9] CLKACTIVITY_TIMER5_GFCLK
TIMER6_GFCLK Clock Status	CM_IPU_CLKSTCTRL[10] CLKACTIVITY_TIMER6_GFCLK
TIMER7_GFCLK Clock Status	CM_IPU_CLKSTCTRL[11] CLKACTIVITY_TIMER7_GFCLK
TIMER8_GFCLK Clock Status	CM_IPU_CLKSTCTRL[12] CLKACTIVITY_TIMER8_GFCLK
MCASP1_AHCLKR Clock Status	CM_IPU_CLKSTCTRL[18] CLKACTIVITY_MCASP1_AHCLKR
MCASP1_AHCLKX Clock Status	CM_IPU_CLKSTCTRL[17] CLKACTIVITY_MCASP1_AHCLKX
MCASP1_AUX_GFCLK Clock Status	CM_IPU_CLKSTCTRL[16] CLKACTIVITY_MCASP1_AUX_GFCLK
Clock Domain State Transition Control	CM_IPU_CLKSTCTRL[1:0] CLKTRCTRL

3.6.4.15.3 Clock Domain Dependency

CD_IPU has no module wake-up dependency with any other clock domain of the device.

3.6.4.15.3.1 Static Dependency

Table 3-176 lists the static dependency of the clock domain with respect to other clock domains of the device.

Table 3-176. CD_IPU Static Dependency Association Parameters

Clock Domain Name	Default Setting	Control Bit Field	Access Type
CD_DSS	Disabled	CM_IPU1_STATICDEP[8] DSS_STATDEP	Read/write
CD_CAM	Alway disabled	CM_IPU1_STATICDEP[9] CAM_STATDEP	Read only
CD_L3_MAIN1	Enabled	CM_IPU1_STATICDEP[5] L3MAIN1_STATDEP	Read/write
CD_L3INIT	Disabled	CM_IPU1_STATICDEP[7] L3INIT_STATDEP	Read/write
CD_L4_CFG	Enabled	CM_IPU1_STATICDEP[12] L4CFG_STATDEP	Read/write
CD_L4PER1	Disabled	CM_IPU1_STATICDEP[13] L4PER_STATDEP	Read/write
CD_L4PER2	Disabled	CM_IPU1_STATICDEP[26] L4PER2_STATDEP	Read/write
CD_L4PER3	Disabled	CM_IPU1_STATICDEP[27] L4PER3_STATDEP	Read/write
CD_EMIF	Enabled	CM_IPU1_STATICDEP[4] EMIF_STATDEP	Read/write
CD_IPU	Disabled	CM_IPU1_STATICDEP[24] IPU_STATDEP	Read/write
CD_DSP1	Disabled	CM_IPU1_STATICDEP[1] DSP1_STATDEP	Read/write
CD_DSP2	Disabled	CM_IPU1_STATICDEP[18] DSP2_STATDEP	Read/write
CD_WKUPAON	Enabled	CM_IPU1_STATICDEP[15] WKUPAON_STATDEP	Read/write
CD_COREAON	Always disabled	CM_IPU1_STATICDEP[16] COREAON_STATDEP	Read only
CD_CUSTEFUSE	Always disabled	CM_IPU1_STATICDEP[17] CUSTEFUSE_STATDEP	Read only
CD_GMAC	Disabled	CM_IPU1_STATICDEP[25] GMAC_STATDEP	Read/write
CD_ISS	Enabled	CM_IPU1_STATICDEP[28] ISS_STATDEP	Read/write
CD_CRC	Enabled	CM_IPU1_STATICDEP[30] CRC_STATDEP	Read/write
CD_EVE1	Disabled	CM_IPU1_STATICDEP[19] EVE1_STATDEP	Read/write

3.6.4.15.3.2 Dynamic Dependency

Table 3-177 lists the dynamic dependency of the clock domain with respect to other clock domains of the device.

Table 3-177. CD_IPU Dynamic Dependency Association Parameters

Clock Domain Name	Default Setting	Control Bit Field	Access Type
CD_L3_MAIN1	Always enabled	CM_IPU1_DYNAMICDEP[5] L3MAIN1_DYNDEP	Read only

3.6.4.15.4 Clock Domain Module Attributes

Table 3-178 lists for each module of the clock domain the clocks the module receives and their role (that is, functional or interface clock).

Table 3-178. CD_IPU Modules Clocks Association

Module	Clock	Clock Type
TIMER5	IPU_L3_GICLK	Interface(1)
	TIMER5_GFCLK	Functional
TIMER6	IPU_L3_GICLK	Interface(1)

(1) The L4 clock, required by the L4 interface, is created internally to the module. It is derived from the L4PER_L3_GICLK clock divided by 2 using a gated version of L4_ICLK.

Table 3-178. CD_IPU Modules Clocks Association (continued)

Module	Clock	Clock Type
	TIMER6_GFCLK	Functional
TIMER7	IPU_L3_GICLK	Interface(1)
	TIMER7_GFCLK	Functional
TIMER8	IPU_L3_GICLK	Interface(1)
	TIMER8_GFCLK	Functional
McASP1	IPU_L3_GICLK	Interface(1)
	MCASP1_AHCLKR	Functional
	MCASP1_AHCLKX	Functional
	MCASP1_AUX_GFCLK	Functional

Table 3-179 lists the supported wake-up request generation capability for each module of the clock domain.

Table 3-179. CD_IPU Modules Wake-Up Request

Module	Wake-Up Feature
TIMER5	Slave wake-up request (IPU1-IRQ, DSP1-IRQ, DSP2-IRQ, EVE1-IRQ)
TIMER6	Slave wake-up request (IPU1-IRQ, DSP1-IRQ, DSP2-IRQ, EVE1-IRQ)
TIMER7	Slave wake-up request (IPU1-IRQ, DSP1-IRQ, DSP2-IRQ, EVE1-IRQ)
TIMER8	Slave wake-up request (IPU1-IRQ, DSP1-IRQ, DSP2-IRQ, EVE1-IRQ)
McASP1	Slave wake-up request (IPU1-IRQ, DSP1-IRQ, DSP2-IRQ, EVE1-IRQ)

Table 3-180 lists the supported clock-management modes and associated software control bit fields for each module of the power domain.

Table 3-180. CD_IPU Modules Clock-Management Modes and Control

Module	Clock-Management Protocol	Status Bit Field	Role
TIMER5	Slave	CM_IPU_TIMER5_CLKCTRL[17:16] IDLEST	Idle status
TIMER6	Slave	CM_IPU_TIMER6_CLKCTRL[17:16] IDLEST	Idle status
TIMER7	Slave	CM_IPU_TIMER7_CLKCTRL[17:16] IDLEST	Idle status
TIMER8	Slave	CM_IPU_TIMER8_CLKCTRL[17:16] IDLEST	Idle status
McASP1	Slave	CM_IPU_MCASP1_CLKCTRL[17:16] IDLEST	Idle status

Table 3-181 lists the supported clock-management modes and associated software control bit fields for each module of the power domain.

Table 3-181. CD_IPU Modules Slave Clock-Management Modes and Control

Module	Disabled	Auto	Enabled	Control Bit Field	Access Type
McASP1	Available	N/A	Available	CM_IPU_MCASP1_CLKCTRL[1:0] MODULEMODE	Read/write
TIMER5	Available	N/A	Available	CM_IPU_TIMER5_CLKCTRL[1:0] MODULEMODE	Read/write
TIMER6	Available	N/A	Available	CM_IPU_TIMER6_CLKCTRL[1:0] MODULEMODE	Read/write
TIMER7	Available	N/A	Available	CM_IPU_TIMER7_CLKCTRL[1:0] MODULEMODE	Read/write

Table 3-181. CD_IPU Modules Slave Clock-Management Modes and Control (continued)

Module	Disabled	Auto	Enabled	Control Bit Field	Access Type
TIMER8	Available	N/A	Available	CM_IPU_TIMER8_CLKCTRL[1:0] MODULEMODE	Read/write

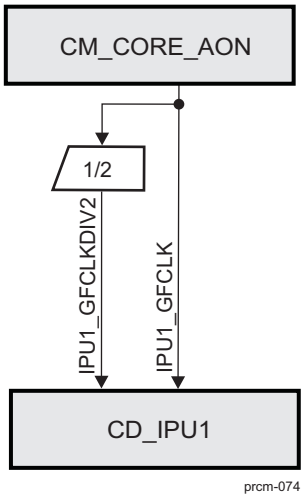
3.6.4.16 CD_IPU1 Clock Domain

This section identifies the modes supported by the clock domain and the associated control and status bits. It also identifies its dependencies with other clock domains of the device.

3.6.4.16.1 Overview

Figure 3-60 is an overview of the clock domain.

Figure 3-60. CD_IPU1 Overview



3.6.4.16.2 Clock Domain Modes

Table 3-182 lists the clock domain modes supported by the clock domain.

Table 3-182. CD_IPU1 Clock Domain Modes

NO_SLEEP	SW_SLEEP	SW_WKUP	HW_AUTO
Available	Available	Available	Available

Table 3-183 lists the clock domain state transition control and status bits for the clock in this clock domain.

Table 3-183. CD_IPU1 Control and Status Parameters

Parameter Name	Control/Status Bit Field
IPU1_GFCLK Clock Status	CM_IPU1_CLKSTCTRL[8] CLKACTIVITY_IPU1_GFCLK
Clock Domain State Transition Control	CM_IPU1_CLKSTCTRL[1:0] CLKTRCTRL

3.6.4.16.3 Clock Domain Dependency

CD_IPU1 has no module wake-up dependency with any other clock domain of the device.

3.6.4.16.3.1 Static Dependency

Table 3-184 lists the static dependency of the clock domain with respect to other clock domains of the device.

Table 3-184. CD_IPU1 Static Dependency Association Parameters

Clock Domain Name	Default Setting	Control Bit Field	Access Type
CD_DSS	Disabled	CM_IPU1_STATICDEP[8] DSS_STATDEP	Read/write
CD_CAM	Always disabled	CM_IPU1_STATICDEP[9] CAM_STATDEP	Read only
CD_L3_MAIN1	Enabled	CM_IPU1_STATICDEP[5] L3MAIN1_STATDEP	Read/write
CD_L3INIT	Disabled	CM_IPU1_STATICDEP[7] L3INIT_STATDEP	Read/write
CD_L4_CFG	Enabled	CM_IPU1_STATICDEP[12] L4CFG_STATDEP	Read/write
CD_L4PER1	Disabled	CM_IPU1_STATICDEP[13] L4PER_STATDEP	Read/write
CD_L4PER2	Disabled	CM_IPU1_STATICDEP[26] L4PER2_STATDEP	Read/write
CD_L4PER3	Disabled	CM_IPU1_STATICDEP[27] L4PER3_STATDEP	Read/write
CD_EMIF	Enabled	CM_IPU1_STATICDEP[4] EMIF_STATDEP	Read/write
CD_IPU	Disabled	CM_IPU1_STATICDEP[24] IPU_STATDEP	Read/write
CD_DSP1	Disabled	CM_IPU1_STATICDEP[1] DSP1_STATDEP	Read/write
CD_DSP2	Disabled	CM_IPU1_STATICDEP[18] DSP2_STATDEP	Read/write
CD_WKUPAON	Enabled	CM_IPU1_STATICDEP[15] WKUPAON_STATDEP	Read/write
CD_COREAON	Alway disabled	CM_IPU1_STATICDEP[16] COREAON_STATDEP	Read only
CD_CUSTEFUSE	Alway disabled	CM_IPU1_STATICDEP[17] CUSTEFUSE_STATDEP	Read only
CD_GMAC	Disabled	CM_IPU1_STATICDEP[25] GMAC_STATDEP	Read/write
CD_ISS	Enabled	CM_IPU1_STATICDEP[28] ISS_STATDEP	Read/write
CD_CRC	Enabled	CM_IPU1_STATICDEP[30] CRC_STATDEP	Read/write
CD_EVE1	Disabled	CM_IPU1_STATICDEP[19] EVE1_STATDEP	Read/write

3.6.4.16.3.2 Dynamic Dependency

Table 3-185 lists the dynamic dependency of the clock domain with respect to other clock domains of the device.

Table 3-185. CD_IPU1 Dynamic Dependency Association Parameters

Clock Domain Name	Default Setting	Control Bit Field	Access Type
CD_L3_MAIN1	Always enabled	CM_IPU1_DYNAMICDEP[5] L3MAIN1_DYNDEP	Read only

3.6.4.16.4 Clock Domain Module Attributes

Table 3-186 lists for each module of the clock domain the clocks the module receives and their role (that is, functional or interface clock).

Table 3-186. CD_IPU1 Modules Clocks Association

Module	Clock	Clock Type
IPU1	IPU1_GFCLK	Functional
	IPU1_GFCLKDIV2[(1)]	Interface

(1) IPU1_GFCLKDIV2 is divided by 2 version of the IPU1_GFCLK. For more information about the IPU1_GFCLK additional division by 2, see Section 5.2.1*IPU Subsystem Clock Distribution.*

Table 3-187 lists the supported wake-up request generation capability for each module of the clock domain.

Table 3-187. CD_IPU1 Modules Wake-Up Request

Module	Wake-Up Feature
IPU1	Master wake-up request

Table 3-188 lists the supported clock-management modes and associated software control bit fields for each module of the power domain.

Table 3-188. CD_IPU1 Modules Clock-Management Modes and Control

Module	Clock-Management Protocol	Status Bit Field	Role
IPU1	Master/slave	CM_IPU1_IPU1_CLKCTRL[18] STBYST	Standby status
		CM_IPU1_IPU1_CLKCTRL[17:16] IDLEST	Idle status

Table 3-189 lists the supported clock-management modes and associated software control bit fields for each module of the power domain.

Table 3-189. CD_IPU1 Modules Slave Clock-Management Modes and Control

Module	Disabled	Auto	Enabled	Control Bit Field	Access Type
IPU1	Available	Available	N/A	CM_IPU1_IPU1_CLKCTRL[1:0] MODULEMODE	Read/write

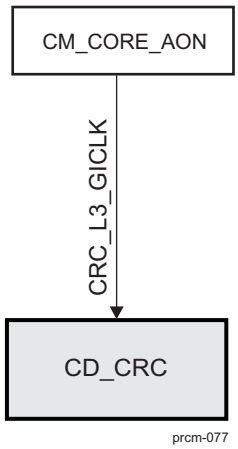
3.6.4.17 CD_CRC Clock Domain

This section identifies the modes supported by the clock domain and the associated control and status bits. It also identifies its dependencies with other clock domains of the device.

3.6.4.17.1 Overview

Figure 3-61 is an overview of the clock domain.

Figure 3-61. CD_CRC Overview



3.6.4.17.2 Clock Domain Modes

Table 3-190 lists the clock domain modes supported by the clock domain.

Table 3-190. CD_CRC Clock Domain Modes

NO_SLEEP	SW_SLEEP	SW_WKUP	HW_AUTO
Available	Not available	Available	Available

Table 3-191 lists the clock domain state transition control and status bits for the clock in this clock domain.

Table 3-191. CD_CRC Control and Status Parameters

Parameter Name	Control/Status Bit Field
CRC_L3_GICLK Clock Status	CM_CRC_CLKSTCTRL[8] CLKACTIVITY_CRC_L3_GICLK
Clock Domain State Transition Control	CM_CRC_CLKSTCTRL[1:0] CLKTRCTRL

3.6.4.17.3 Clock Domain Module Attributes

Table 3-192 lists for each module of the clock domain the clocks the module receives and their role (that is, functional or interface clock).

Table 3-192. CD_CRC Modules Clocks Association

Module	Clock	Clock Type
CRC	CRC_L3_GICLK	Interface

Table 3-193 lists the supported wake-up request generation capability for each module of the clock domain.

Table 3-193. CD_CRC Modules Wake-Up Request

Module	Wake-Up Feature
CRC	None

Table 3-194 lists the supported clock-management modes and associated software control bit fields for each module of the power domain.

Table 3-194. CD_CRC Modules Clock-Management Modes and Control

Module	Clock-Management Protocol	Status Bit Field	Role
CRC	Slave	CM_CRC_CRC_CLKCTRL[17:16] IDLEST	Idle status

Table 3-195 lists the supported clock-management modes and associated software control bit fields for each module of the power domain.

Table 3-195. CD_CRC Modules Slave Clock-Management Modes and Control

Module	Disabled	Auto	Enabled	Control Bit Field	Access Type
CRC	Available	N/A	Available	CM_CRC_CRC_CLKCTRL[1:0] MODULEMODE	Read/write

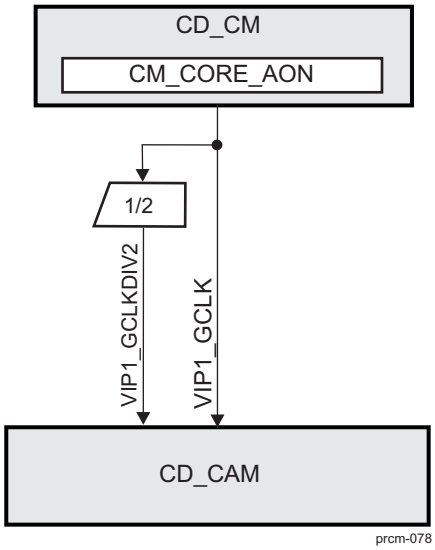
3.6.4.18 CD_CAM Clock Domain

This section identifies the modes supported by the clock domain and the associated control and status bits. It also identifies its dependencies with other clock domains of the device.

3.6.4.18.1 Overview

Figure 3-62 is an overview of the clock domain.

Figure 3-62. CD_CAM Overview



3.6.4.18.2 Clock Domain Modes

Table 3-196 lists the clock domain modes supported by the clock domain.

Table 3-196. CD_CAM Clock Domain Modes

NO_SLEEP	SW_SLEEP	SW_WKUP	HW_AUTO
Available	Available	Available	Available

Table 3-197 lists the clock domain state transition control and status bits for the clock in this clock domain.

Table 3-197. CD_CAM Control and Status Parameters

Parameter Name	Control/Status Bit Field
VIP1_GCLK Clock Status	CM_CAM_CLKSTCTRL[8] CLKACTIVITY_VIP1_GCLK
Clock Domain State Transition Control	CM_CAM_CLKSTCTRL[1:0] CLKTRCTRL

3.6.4.18.3 Clock Domain Dependency

CD_CAM has no module wake-up dependency with any other clock domain of the device.

3.6.4.18.3.1 Static Dependency

Table 3-198 lists the static dependency of the clock domain with respect to other clock domains of the device.

Table 3-198. CD_CAM Static Dependency Association Parameters

Clock Domain Name	Default Setting	Control Bit Field	Access Type
CD_L3_MAIN1	Always enabled	CM_CAM_STATICDEP[5] L3MAIN1_STATDEP	Read only
CD_L4_CFG	Disabled	CM_CAM_STATICDEP[12] L4CFG_STATDEP	Read only
CD_EMIF	Always enabled	CM_CAM_STATICDEP[4] EMIF_STATDEP	Read/write
CD_EVE1	Disabled	CM_CAM_STATICDEP[19] EVE1_STATDEP	Read/write
CD_GMAC	Disabled	CM_CAM_STATICDEP[25] GMAC_STATDEP	Read/write
CD_L4PER3	Disabled	CM_CAM_STATICDEP[27] L4PER3_STATDEP	Read/write
CD_ISS	Disabled	CM_CAM_STATICDEP[28] ISS_STATDEP	Read/write

3.6.4.18.3.2 Dynamic Dependency

CD_CAM has no dynamic dependency with any other clock domain of the device.

3.6.4.18.4 Clock Domain Module Attributes

Table 3-199 lists for each module of the clock domain the clocks the module receives and their role (that is, functional or interface clock).

Table 3-199. CD_CAM Modules Clocks Association

Module	Clock	Clock Type
VIP	VIP1_GCLK	Functional
	VIP1_GCLKDIV2[1]	Interface

(1) VIP1_GCLKDIV2 is divided by 2 version of the VIP1_GCLK. For more information about the VIP1_GCLK additional division by 2, please see Section 7.3*VIP Integration*

Table 3-200 lists the supported wake-up request generation capability for each module of the clock domain.

Table 3-200. CD_CAM Modules Wake-Up Request

Module	Wake-Up Feature
VIP	Slave wake-up request (IPU1-IRQ, DSP1-IRQ, DSP2-IRQ, EVE1-IRQ)/ Master wake-up request

Table 3-201 lists the supported clock-management modes and associated software control bit fields for each module of the power domain.

Table 3-201. CD_CAM Modules Clock-Management Modes and Control

Module	Clock-Management Protocol	Status Bit Field	Role
VIP	Master/slave	CM_CAM_VIP1_CLKCTRL[18] STBYST	Standby status
		CM_CAM_VIP1_CLKCTRL[17:16] IDLEST	Idle status

Table 3-202 lists the supported clock-management modes and associated software control bit fields for each module of the power domain.

Table 3-202. CD_CAM Modules Slave Clock-Management Modes and Control

Module	Disabled	Auto	Enabled	Control Bit Field	Access Type
VIP	Available	Available	N/A	CM_CAM_VIP1_CLKCTRL[1:0] MODULEMODE	Read/write

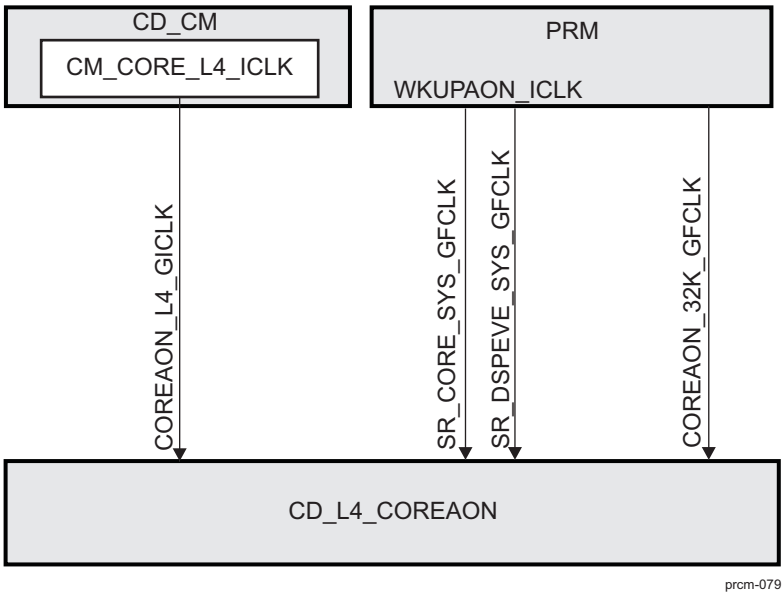
3.6.4.19 CD_COREAON_L4 Clock Domain

This section identifies the modes supported by the clock domain and the associated control and status bits. It also identifies its dependencies with other clock domains of the device.

3.6.4.19.1 CD_COREAON_L4 Overview

Figure 3-63 is an overview of the clock domain.

Figure 3-63. CD_COREAON_L4 Overview



3.6.4.19.2 Clock Domain Modes

Table 3-203 lists the clock domain modes supported by the clock domain.

Table 3-203. CD_COREAON_L4 Clock Domain Modes

NO_SLEEP	SW_SLEEP	SW_WKUP	HW_AUTO
Available	N/A	Available	Available

Table 3-204 lists the clock domain state transition control and status bits for the clock in this clock domain.

Table 3-204. CD_COREAON_L4 Control and Status Parameters

Parameter Name	Control/Status Bit Field
COREAON_32K_GFCLK Clock Status	CM_COREAON_CLKSTCTRL[12] CLKACTIVITY_COREAON_32K_GFCLK
SR_CORE_SYS_GFCLK Clock Status	CM_COREAON_CLKSTCTRL[11] CLKACTIVITY_SR_CORE_SYS_GFCLK
COREAON_L4_GICLK Clock Status	CM_COREAON_CLKSTCTRL[8] CLKACTIVITY_COREAON_L4_GICLK
SR_DSPEVE_SYS_GFCLK Clock Status	CM_COREAON_CLKSTCTRL[13] CLKACTIVITY_SR_DSPEVE_SYS_GFCLK
ABE_GICLK Clock Status	CM_COREAON_CLKSTCTRL[16] CLKACTIVITY_ABE_GICLK
Clock Domain State Transition Control	CM_COREAON_CLKSTCTRL[1:0] CLKTRCTRL

3.6.4.19.3 Clock Domain Dependency

CD_COREAON_L4 has no static or dynamic dependency with any other clock domain of the device.

3.6.4.19.3.1 Wake-Up Dependency

There is no wake-up dependency settings for the modules of this CD_COREAON_L4 clock domain.

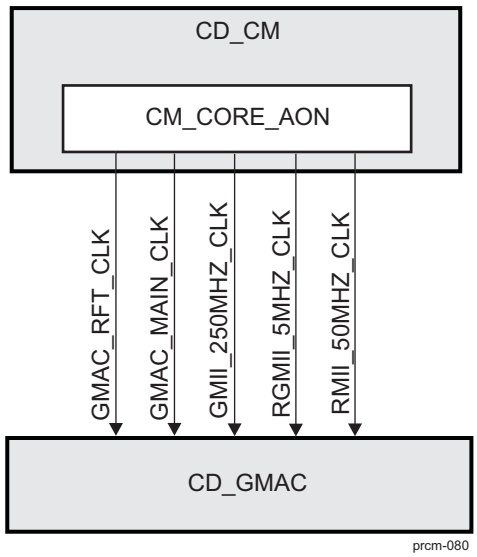
3.6.4.20 CD_GMAC Clock Domain

This section identifies the modes supported by the clock domain and the associated control and status bits. It also identifies its dependencies with other clock domains of the device.

3.6.4.20.1 Overview

Figure 3-64 is an overview of the clock domain.

Figure 3-64. CD_GMAC Overview



3.6.4.20.2 Clock Domain Modes

Table 3-205 lists the clock domain modes supported by the clock domain.

Table 3-205. CD_GMAC Clock Domain Modes

NO_SLEEP	SW_SLEEP	SW_WKUP	HW_AUTO
Available	Available	Available	Available

Table 3-206 lists the clock domain state transition control and status bits for the clock in this clock domain.

Table 3-206. CD_CAM Control and Status Parameters

Parameter Name	Control/Status Bit Field
GMII_250MHZ_CLK Clock Status	CM_GMAC_CLKSTCTRL[8] CLKACTIVITY_GMII_250MHZ_CLK
RGMII_5MHZ_CLK Clock Status	CM_GMAC_CLKSTCTRL[9] CLKACTIVITY_RGMII_5MHZ_CLK
RMII_50MHZ_CLK Clock Status	CM_GMAC_CLKSTCTRL[10] CLKACTIVITY_RMII_50MHZ_CLK
GMAC_MAIN_CLK Clock Status	CM_GMAC_CLKSTCTRL[12] CLKACTIVITY_GMAC_MAIN_CLK
GMAC_RFT_CLK Clock Status	CM_GMAC_CLKSTCTRL[11] CLKACTIVITY_GMAC_RFT_CLK
Clock Domain State Transition Control	CM_GMAC_CLKSTCTRL[1:0] CLKTRCTRL

3.6.4.20.3 Clock Domain Dependency

3.6.4.20.3.1 Static Dependency

Table 3-207 lists the static dependency of the clock domain with respect to other clock domains of the device.

Table 3-207. CD_GMAC Static Dependency Association Parameters

Clock Domain Name	Default Setting	Control Bit Field	Access Type
CD_L3_MAIN1	Always enabled	CM_GMAC_STATICDEP[5] L3MAIN1_STATDEP	Read only
CD_EMIF	Disabled	CM_GMAC_STATICDEP[4] EMIF_STATDEP	Read/write
CD_L4PER2	Disabled	CM_GMAC_STATICDEP[26] L4PER2_STATDEP	Read/write

3.6.4.20.3.2 Dynamic Dependency

Table 3-208 lists the dynamic dependency of the clock domain with respect to other clock domains of the device.

Table 3-208. CD_GMAC Dynamic Dependency Association Parameters

Clock Domain Name	Default Setting	Control Bit Field	Access Type
CD_L3_MAIN1	Always disabled	CM_GMAC_DYNAMICDEP[5] L3MAIN1_DYNDEP	Read only

3.6.4.20.4 Clock Domain Module Attributes

Table 3-209 lists for each module of the clock domain the clocks the module receives and their role (that is, functional or interface clock).

Table 3-209. CD_GMAC Modules Clocks Association

Module	Clock	Clock Type
GMAC_SW	GMAC_MAIN_CLK	Interface
	GMAC_RFT_CLK	Functional
	GMII_250MHZ_CLK	Functional
	RGMII_5MHZ_CLK	Functional
	RMII_50MHZ_CLK	Functional

Table 3-210 lists the supported wake-up request generation capability for each module of the clock domain.

Table 3-210. CD_GMAC Modules Wake-Up Request

Module	Wake-Up Feature
GMAC_SW	None

Table 3-211 lists the supported clock-management modes and associated software control bit fields for each module of the power domain.

Table 3-211. CD_GMAC Modules Clock-Management Modes and Control

Module	Clock-Management Protocol	Status Bit Field	Role
GMAC_SW	Master/slave	CM_GMAC_GMAC_CLKCTRL[18] STBYST	Standby status
		CM_GMAC_GMAC_CLKCTRL[17:16] IDLEST	Idle status

Table 3-211. CD_GMAC Modules Clock-Management Modes and Control (continued)

Module	Clock-Management Protocol	Status Bit Field	Role
		CM_GMAC_GMAC_CLKCTRL[24] CLKSEL_REF	Select the source of the functional clock
		CM_GMAC_GMAC_CLKCTRL[27:25] CLKSEL_RFT	Select the source of the CPTS_RFT_CLK

Table 3-212 lists the supported clock-management modes and associated software control bit fields for each module of the power domain.

Table 3-212. CD_GMAC Modules Slave Clock-Management Modes and Control

Module	Disabled	Auto	Enabled	Control Bit Field	Access Type
GMAC_SW	Available	N/A	Available	CM_GMAC_GMAC_CLKCTRL[1:0] MODULEMODE	Read/write

3.6.4.21 CD_ISS Clock Domain

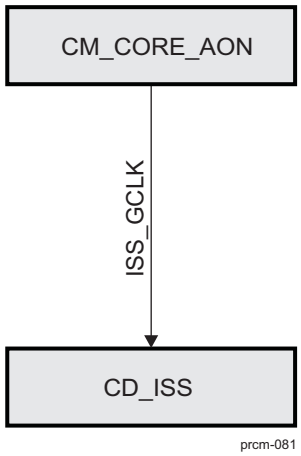
This section identifies the modes supported by the clock domain and the associated control and status bits. It also identifies its dependencies with other clock domains of the device.

NOTE: ISS and CAMERARX are not supported on the DRA78x family of devices.

3.6.4.21.1 CD_ISS Overview

Figure 3-65 is an overview of the clock domain.

Figure 3-65. CD_ISS Overview



3.6.4.21.2 Clock Domain Modes

Table 3-213 lists the clock domain modes supported by the clock domain.

Table 3-213. CD_ISS Clock Domain Modes

NO_SLEEP	SW_SLEEP	SW_WKUP	HW_AUTO
Available	Available	Available	Available

Table 3-214 lists the clock domain state transition control and status bits for the clock in this clock domain.

Table 3-214. CD_ISS Control and Status Parameters

Parameter Name	Control/Status Bit Field
CLKACTIVITY_ISS_GCLK Clock Status	CM_ISS_CLKSTCTRL[8] CLKACTIVITY_ISS_GCLK
Clock Domain State Transition Control	CM_ISS_CLKSTCTRL[1:0] CLKTRCTRL

3.6.4.21.3 Clock Domain Dependency

Table 3-215 lists the static dependency of the clock domain with respect to other clock domains of the device.

Table 3-215. CD_ISS Static Dependency Association Parameters

Clock Domain Name	Default Setting	Control Bit Field	Access Type
CD_L3_MAIN1	Always enabled	CM_ISS_STATICDEP[5] L3MAIN1_STATDEP	Read only
CD_EMIF	Disabled	CM_ISS_STATICDEP[4] EMIF_STATDEP	Read/write
CD_L4PER3	Disabled	CM_ISS_STATICDEP[27] L4PER3_STATDEP	Read/write

3.6.4.21.3.1 Wake-Up Dependency

Table 3-216 lists the wake-up dependency settings for the modules of this clock domain.

Table 3-216. CD_ISS Wake-Up Dependency Association Parameters

Originator Module	Originator Clock Domain	Servicing Clock Domain	Default Setting	Control Bit Field	Access Type
ISS	CD_ISS	CD_IPU1, CD_L3_MAIN1, L4PER1, L4PER2, L4PER3	Disabled	PM_ISS_ISS_WKDEP[4] WKUPDEP_ISS_IPU1	Read/write
	CD_ISS	CD_DSP1, CD_L3_MAIN1, L4PER1, L4PER2, L4PER3	Disabled	PM_ISS_ISS_WKDEP[2] WKUPDEP_ISS_DSP1	Read/write
	CD_ISS	CD_DSP2, CD_L3_MAIN1, L4PER1, L4PER2, L4PER3	Disabled	PM_ISS_ISS_WKDEP[5] WKUPDEP_ISS_DSP2	Read/write
	CD_ISS	CD_EVE1,CD_L3_MAIN1, L4PER1, L4PER2, L4PER3	Disabled	PM_ISS_ISS_WKDEP[6] WKUPDEP_ISS_EVE1	Read/write

3.6.4.21.4 Clock Domain Module Attributes

Table 3-217 lists for each module of the clock domain the clocks the module receives and their role (that is, functional or interface clock).

Table 3-217. CD_ISS Modules Clocks Association

Module	Clock	Clock Type
ISS	ISS_GCLK	Interface and functional
CAMERARX	FUNC_96M_FCLK	Interface

Table 3-218 lists the supported wake-up request generation capability for each module of the clock domain.

Table 3-218. CD_ISS Modules Wake-Up Request

Module	Wake-Up Feature
ISS	Master wake-up request
	Slave wake-up request (IPU1-IRQ, DSP1-IRQ, DSP2-IRQ, EVE1-IRQ)

Table 3-219 lists the supported clock-management modes and associated software control bit fields for each module of the power domain.

Table 3-219. CD_ISS Modules Clock-Management Modes and Control

Module	Clock-Management Protocol	Status Bit Field	Role
ISS	Master/Slave	CM_ISS_ISS_CLKCTRL[18] STBYST	Standby status
		CM_ISS_ISS_CLKCTRL[17:16] IDLEST	Idle status

Table 3-220 lists the supported clock-management modes and associated software control bit fields for each module of the power domain.

Table 3-220. CD_ISS Modules Slave Clock-Management Modes and Control

Module	Disabled	Auto	Enabled	Control Bit Field	Access Type
ISS	Available	Available	N/A	CM_ISS_ISS_CLKCTRL[1:0] MODULEMODE	Read/write
CAMERARX	Available	N/A	Available	CM_L4PER_I2C1_CLKCTRL[1:0] MODULEMODE or CM_L4PER_I2C2_CLKCTRL[1:0] MODULEMODE	Read/write

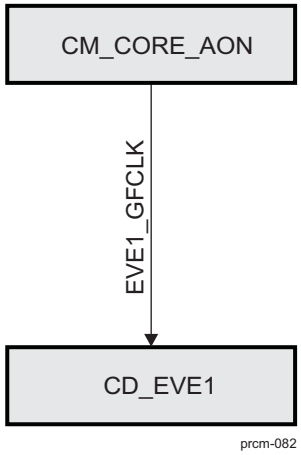
3.6.4.22 CD_EVE1 Clock Domain

This section identifies the modes supported by the clock domain and the associated control and status bits. It also identifies its dependencies with other clock domains of the device.

3.6.4.22.1 CD_EVE1 Overview

Figure 3-66 is an overview of the clock domain.

Figure 3-66. CD_EVE1 Overview



3.6.4.22.2 Clock Domain Modes

Table 3-221 lists the clock domain modes supported by the clock domain.

Table 3-221. CD_EVE1 Clock Domain Modes

NO_SLEEP	SW_SLEEP	SW_WKUP	HW_AUTO
Available	Available	Available	Available

Table 3-222 lists the clock domain state transition control and status bits for the clock in this clock domain.

Table 3-222. CD_EVE1 Control and Status Parameters

Parameter Name	Control/Status Bit Field
EVE1_GFCLK Clock Status	CM_EVE1_CLKSTCTRL[8] CLKACTIVITY_EVE1_GFCLK
Clock Domain State Transition Control	CM_EVE1_CLKSTCTRL[1:0] CLKTRCTRL

3.6.4.22.3 Clock Domain Dependency

Table 3-223 lists the static dependency of the clock domain with respect to other clock domains of the device.

Table 3-223. CD_EVE1 Static Dependency Association Parameters

Clock Domain Name	Default Setting	Control Bit Field	Access Type
CD_L3_MAIN1	Always enabled	CM_EVE1_STATICDEP[5] L3MAIN1_STATDEP	Read only
CD_EMIF	Always enabled	CM_EVE1_STATICDEP[4] EMIF_STATDEP	Read/write

3.6.4.22.3.1 Wake-Up Dependency

Table 3-224 lists the wake-up dependency settings for the modules of this clock domain.

Table 3-224. CD_EVE1 Wake-Up Dependency Association Parameters

Originator Module	Originator Clock Domain	Servicing Clock Domain	Default Setting	Control Bit Field	Access Type
EVE1	CD_EVE1	CD_IPU1, CD_L3_MAIN1, L4PER1, L4PER2, L4PER3	Disabled	PM_EVE1_EVE1_WKDEP[4] WKUPDEP_EVE1_IPU1	Read/write
	CD_EVE1	CD_DSP1, CD_L3_MAIN1, L4PER1, L4PER2, L4PER3	Disabled	PM_EVE1_EVE1_WKDEP[2] WKUPDEP_EVE1_DSP1	Read/write
	CD_EVE1	CD_DSP2, CD_L3_MAIN1, L4PER1, L4PER2, L4PER3	Disabled	PM_EVE1_EVE1_WKDEP[5] WKUPDEP_EVE1_DSP2	Read/write

3.6.4.22.4 Clock Domain Module Attributes

Table 3-225 lists for each module of the clock domain the clocks the module receives and their role (that is, functional or interface clock).

Table 3-225. CD_EVE1 Modules Clocks Association

Module	Clock	Clock Type
EVE1	EVE1_GFCLK	Interface

Table 3-226 lists the supported wake-up request generation capability for each module of the clock domain.

Table 3-226. CD_EVE1 Modules Wake-Up Request

Module	Wake-Up Feature
EVE1	Slave wake-up request (IPU1-IRQ, DSP1-IRQ, DSP2-IRQ, EVE1-IRQ, EVE2-IRQ)

Table 3-227 lists the supported clock-management modes and associated software control bit fields for each module of the power domain.

Table 3-227. CD_EVE1 Modules Clock-Management Modes and Control

Module	Clock-Management Protocol	Status Bit Field	Role
EVE1	Master/Slave	CM_EVE1_EVE1_CLKCTRL[18] STBYST	Standby status
		CM_EVE1_EVE1_CLKCTRL[17:16] IDLEST	Idle status

Table 3-228 lists the supported clock-management modes and associated software control bit fields for each module of the power domain.

Table 3-228. CD_EVE1 Modules Slave Clock-Management Modes and Control

Module	Disabled	Auto	Enabled	Control Bit Field	Access Type
EVE1	Available	Available	N/A	CM_EVE1_EVE1_CLKCTRL[1:0] MODULEMODE	Read/write

3.7 Power Management Functional Description

This section describes the functional concepts of power management at the power domain level in the device.

The following power domains support dynamic power switching (DPS) with switching times of less than 5 µs.

- PD_CORE

3.7.1 PD_WKUPAON Description

PD_WKUPAON contains the following reset domains:

- WKUPAON_PWRON_RST
- WKUPAON_RST
- WKUPAON_SYS_PWRON_RST
- PRM_PWRON_RST
- PRM_RST
- LPRM_PWRON_RST
- LPRM_RST

PD_WKUPAON contains the CD_WKUPAON clock domain.

Table 3-229 lists the logic retention capability for each module of the power domain.

Table 3-229. PD_WKUPAON Modules Power Attributes

Module	Logic Retention	DFF Context Status	RFF Context Status
CTRL_MODULE_WKUP	No	None	None
GPIO1	No	RM_WKUPAON_GPIO1_CONTEXT[0] LOSTCONTEXT_DFF	None
COUNTER_32K	No	RM_WKUPAON_COUNTER_32K_CONTEXT[0] LOSTCONTEXT_DFF	None
TIMER1	No	RM_WKUPAON_TIMER1_CONTEXT[0] LOSTCONTEXT_DFF	None
L4_WKUP interconnect	No	RM_WKUPAON_L4_WKUP_CONTEXT[0] LOSTCONTEXT_DFF	None
PRM	No	None	None
DCAN1	No	RM_WKUPAON_DCAN1_CONTEXT[0] LOSTCONTEXT_DFF	None

3.7.1.1 Power Domain Modes

The PD_WKUPAON power domain is an always-on power domain and does not switch to RETENTION state. There is no logic power state control or status bit field for this power domain.

The PD_WKUPAON power domain has memory bank (DCAN_MEM) which is related to DCAN1.

3.7.1.1.1 Logic and Memory Area Power Modes

Table 3-230 lists the power modes supported by the memory area of the power domain. A memory area power mode is identified with respect to a power state of the power domain. The Logic On, Logic Retention, and Logic Off columns in the table identify the power states of the power domain. The values in these columns identify the supported power state of the memory area (identified in the Memory Bank column) or the module memory inside the memory area (identified in the Module Memory column).

Table 3-230. PD_WKUPAON Memory Area Power Modes

Memory Bank	Module Memory	Logic On	Logic Retention	Logic Off
WKUP_BANK		ON		
	DCAN1 - DCAN_MEM	always_on		

3.7.2 PD_DSP1 Description

PD_DSP1 contains the following reset domains:

- DSP1_RST
- DSP1_PWRON_RST
- DSP1_RET_RST
- DSP1_SYS_RST

PD_DSP1 contains the CD_DSP1 clock domain.

Table 3-231 lists the logic retention capability for each module of the power domain.

Table 3-231. PD_DSP1 Modules Power Attributes

Module	Logic Retention	DFF Context Status	RFF Context Status
DSP1	No	RM_DSP1_DSP1_CONTEXT[0] LOSTCONTEXT_DFF	None

3.7.2.1 Power Domain Modes

This section describes the various power modes supported by the logic and memory areas of the power domain. It also identifies the associated software control and status bit fields. For a functional description of the power-management architecture of a generic power domain, see Section 3.1.1.2.1, *Power Domain*.

3.7.2.1.1 Logic and Memory Area Power Modes

Table 3-232 lists the power modes supported by the logic area of the power domain.

Table 3-232. PD_DSP1 Logic Area Power Modes

Off	Retention-CSWR	On-Inactive	On-Active
Available	N/A	Available	Available

Table 3-233 lists the power modes supported by the memory area of the power domain. A memory area power mode is identified with respect to a power state of the power domain. The Logic On, Logic Retention, and Logic Off columns in the table identify the power states of the power domain. The values in these columns identify the supported power state of the memory area (identified in the Memory Bank column) or the module memory inside the memory area (identified in the Module Memory column).

Table 3-233. PD_DSP1 Memory Area Power Modes

Memory Bank	Module Memory	Logic On	Logic Retention	Logic Off
DSP1_EDMA		ON		OFF
	DSP DSP_DMA	always_on		
DSP1_L1		ON		OFF
	DSP DSP_L1	always_on		
DSP1_L2		ON		OFF
	DSP DSP_L2	always_on		

3.7.2.1.2 Logic and Memory Area Power Modes Control and Status

Table 3-234 lists the power mode controls for the power domain.

Table 3-234. PD_DSP1 Power Modes Control Parameters

Parameter Name	Memory Bank	Control Bit Field	Access Type
Power Domain Low-Power State Change Control		PM_DSP1_PWRSTCTRL[4] LOWPOWERSTATECHANGE	Read/write
Memory Area State Control (logic in ON state)	DSP1_EDMA	PM_DSP1_PWRSTCTRL[21:20] DSP1_EDMA_ONSTATE	Read only
Memory Area State Control (logic in ON state)	DSP1_L2	PM_DSP1_PWRSTCTRL[19:18] DSP1_L2_ONSTATE	Read only
Memory Area State Control (logic in ON state)	DSP1_L1	PM_DSP1_PWRSTCTRL[17:16] DSP1_L1_ONSTATE	Read only
Power Domain State Transition Control		PM_DSP1_PWRSTCTRL[1:0] POWERSTATE	Read/write

Table 3-235 lists the power mode status for the power domain.

Table 3-235. PD_DSP1 Power Modes Status Parameters

Parameter Name	Memory Bank	Status Bit Field
Power Domain Last Power State Entered Status		PM_DSP1_PWRSTST[25:24] LASTPOWERSTATEENTERED
Memory Area State Status	DSP1_EDMA	PM_DSP1_PWRSTST[9:8] DSP1_EDMA_STATEST
Memory Area State Status	DSP1_L2	PM_DSP1_PWRSTST[7:6] DSP1_L2_STATEST
Memory Area State Status	DSP1_L1	PM_DSP1_PWRSTST[5:4] DSP1_L1_STATEST
Power Domain State Transition Status		PM_DSP1_PWRSTST[20] INTRANSITION
Logic Area State Status		PM_DSP1_PWRSTST[2] LOGICSTATEST
Power Domain State Status		PM_DSP1_PWRSTST[1:0] POWERSTATEST

3.7.3 PD_DSP2 Description

PD_DSP2 contains the following reset domains:

- DSP2_RST
- DSP2_PWRON_RST
- DSP2_RET_RST
- DSP2_SYS_RST

PD_DSP2 contains the CD_DSP2 clock domain.

Table 3-236 lists the logic retention capability for each module of the power domain.

Table 3-236. PD_DSP2 Modules Power Attributes

Module	Logic Retention	DFF Context Status	RFF Context Status
DSP2	No	RM_DSP2_DSP2_CONTEXT[1] LOSTCONTEXT_DFF	None

3.7.3.1 Power Domain Modes

This section describes the various power modes supported by the logic and memory areas of the power domain. It also identifies the associated software control and status bit fields. For a functional description of the power-management architecture of a generic power domain, see Section 3.1.1.2.1, *Power Domain*.

3.7.3.1.1 Logic and Memory Area Power Modes

Table 3-237 lists the power modes supported by the logic area of the power domain.

Table 3-237. PD_DSP2 Logic Area Power Modes

Off	Retention-CSWR	On-Inactive	On-Active
Available	N/A	Available	Available

Table 3-238 lists the power modes supported by the memory area of the power domain. A memory area power mode is identified with respect to a power state of the power domain. The Logic On, Logic Retention, and Logic Off columns in the table identify the power states of the power domain. The values in these columns identify the supported power state of the memory area (identified in the Memory Bank column) or the module memory inside the memory area (identified in the Module Memory column).

Table 3-238. PD_DSP2 Memory Area Power Modes

Memory Bank	Module Memory	Logic On	Logic Retention	Logic Off
DSP2_EDMA		ON	RETENTION	OFF
	DSP DSP_DMA	always_on	always_retention	
DSP2_L1		ON		OFF
	DSP DSP_L1	always_on		
DSP2_L2		ON		OFF
	DSP DSP_L2	always_on		

3.7.3.1.2 Logic and Memory Area Power Modes Control and Status

Table 3-239 lists the power mode controls for the power domain.

Table 3-239. PD_DSP2 Power Modes Control Parameters

Parameter Name	Memory Bank	Control Bit Field	Access Type
Power Domain Low-Power State Change Control		PM_DSP2_PWRSTCTRL[4] LOWPOWERSTATECHANGE	Read/write
Memory Area State Control (logic in ON state)	DSP2_EDMA	PM_DSP2_PWRSTCTRL[21:20] DSP2_EDMA_ONSTATE	Read only
Memory Area State Control (logic in ON state)	DSP2_L2	PM_DSP2_PWRSTCTRL[19:18] DSP2_L2_ONSTATE	Read only
Memory Area State Control (logic in ON state)	DSP2_L1	PM_DSP2_PWRSTCTRL[17:16] DSP2_L1_ONSTATE	Read only
Power Domain State Transition Control		PM_DSP2_PWRSTCTRL[1:0] POWERSTATE	Read/write

Table 3-240 lists the power mode status for the power domain.

Table 3-240. PD_DSP2 Power Modes Status Parameters

Parameter Name	Memory Bank	Status Bit Field
Power domain transition status		PM_DSP2_PWRSTST[20] INTRANSITION
Memory Area State Status	DSP2_L1	PM_DSP2_PWRSTST[5:4] DSP2_L1_STATEST
Memory Area State Status	DSP2_L2	PM_DSP2_PWRSTST[7:6] DSP2_L2_STATEST
Memory Area State Status	DSP2_EDMA	PM_DSP2_PWRSTST[9:8] DSP2_EDMA_STATEST
Logic Area Retention State Control		PM_DSP2_PWRSTST[2] LOGICSTATEST
Power Domain State Transition Control		PM_DSP2_PWRSTST[1:0] POWERSTATE
Last low power state entered		PM_DSP2_PWRSTST[25:24] LASTPOWERSTATEENTERED

3.7.4 PD_CUSTEFUSE Description

PD_CUSTEFUSE contains the following reset domains:

- CUSTEFUSE_RST

PD_CUSTEFUSE contains the CD_CUSTEFUSE clock domain.

Table 3-241 lists the logic retention capability for each module of the power domain.

Table 3-241. PD_CUSTEFUSE Modules Power Attributes

Module	Logic Retention	DFF Context Status	RFF Context Status
EFUSE_CTRL_CUST	No	RM_CUSTEFUSE_EFUSE_CTRL_CUST_CONTEXT[0] LOSTCONTEXT_DFF	None

3.7.4.1 Power Domain Modes

This section describes the various power modes supported by the logic and memory areas of the power domain. It also identifies the associated software control and status bit fields. For a functional description of the power-management architecture of a generic power domain, see Section 3.1.1.2.1, *Power Domain*.

3.7.4.1.1 Logic and Memory Area Power Modes

Table 3-242 lists the power modes supported by the logic area of the power domain.

Table 3-242. PD_CUSTEFUSE Logic Area Power Modes

Off	Retention-CSWR	On-Inactive	On-Active
Available	N/A	Available	Available

There is no memory bank implemented for the PD_CUSTEFUSE.

3.7.4.1.2 Logic and Memory Area Power Modes Control and Status

Table 3-234 lists the power mode controls for the power domain.

Table 3-243. PD_CUSTEFUSE Power Modes Control Parameters

Parameter Name	Control Bit Field	Access Type
Power Domain Low-Power State Change Control	PM_CUSTEFUSE_PWRSTCTRL[4] LOWPOWERSTATECHANGE	Read/write
Power Domain State Transition Control	PM_CUSTEFUSE_PWRSTCTRL[1:0] POWERSTATE	Read/write

Table 3-235 lists the power mode status for the power domain.

Table 3-244. PD_CUSTEFUSE Power Modes Status Parameters

Parameter Name	Status Bit Field
Power Domain Last Power State Entered Status	PM_CUSTEFUSE_PWRSTST[25:24] LASTPOWERSTATEENTERED
Power Domain State Transition Status	PM_CUSTEFUSE_PWRSTST[20] INTRANSITION
Logic Area State Status	PM_CUSTEFUSE_PWRSTST[2] LOGICSTATEST
Power Domain State Status	PM_CUSTEFUSE_PWRSTST[1:0] POWERSTATEST

3.7.5 PD_IPU Description

PD_IPU contains the following reset domains:

- IPU1_PWRON_RST
- IPU1_RST
- IPU1_RET_RST
- IPU1_CPU0_RST
- IPU1_CPU1_RST

PD_IPU contains the CD_IPU1 clock domain.

Table 3-245 lists the logic retention capability for each module of the power domain.

Table 3-245. PD_IPU Module Power Attributes

Module	Logic Retention	DFF Context Status	RFF Context Status
IPU1	Partial	RM_IPU1_IPU1_CONTEXT[0] LOSTCONTEXT_DFF	RM_IPU1_IPU1_CONTEXT[1] LOSTCONTEXT_RFF

3.7.5.1 Power Domain Modes

This section describes the various power modes supported by the logic and memory areas of the power domain. It also identifies the associated software control and status bit fields. For a functional description of the power-management architecture of a generic power domain, see Section 3.1.1.2.1, *Power Domain*.

3.7.5.1.1 Logic and Memory Area Power Modes

Table 3-246 lists the power modes supported by the logic area of the power domain.

Table 3-246. PD_IPU Logic Area Power Modes

Off	Retention-CSWR	On-Inactive	On-Active
Available	N/A	Available	Available

Table 3-247 lists the power modes supported by the memory area of the power domain. A memory area power mode is identified with respect to a power state of the power domain. The Logic On, Logic Retention, and Logic Off columns in the table identify the power states of the power domain. The values in these columns identify the supported power state of the memory area (identified in the Memory Bank column) or the module memory inside the memory area (identified in the Module Memory column).

Table 3-247. PD_IPU Memory Area Power Modes

Memory Bank	Module Memory	Logic On	Logic Retention	Logic Off
AESSMEM		ON	RETENTION	OFF
	IPU1 IPU_L2RAM_MEM	always_on	software_control	
PERIPHMEM		ON		OFF
	IPU1 - IPU_UNICACHE_MEM	always_on	always_retention	

3.7.5.1.2 Logic and Memory Area Power Modes Control and Status

Table 3-248 lists the power mode controls for the power domain.

Table 3-248. PD_IPU Power Modes Control Parameters

Parameter Name	Memory Bank	Control Bit Field	Access Type
Memory Area State Control (logic in RETENTION state)	AESSMEM	PM_IPU_PWRSTCTRL[8] AESSMEM_RETSTATE	Read/write
Memory Area State Control (logic in RETENTION state)	PERIPHMEM	PM_IPU_PWRSTCTRL[10] PERIPHMEM_RETSTATE	Read/write
Power Domain Low-Power State Change Control		PM_IPU_PWRSTCTRL[4] LOWPOWERSTATECHANGE	Read/write
Logic Area Retention State Control		PM_IPU_PWRSTCTRL[2] LOGICRETSTATE	Read only
Memory Area State Control (logic in ON state)	AESSMEM	PM_IPU_PWRSTCTRL[17:16] AESSMEM_ONSTATE	Read only
Memory Area State Control (logic in ON state)	PERIPHMEM	PM_IPU_PWRSTCTRL[21:20] PERIPHMEM_ONSTATE	Read only
Power Domain State Transition Control		PM_IPU_PWRSTCTRL[1:0] POWERSTATE	Read/write

Table 3-249 lists the status of the power modes for the power domain.

Table 3-249. PD_IPU Power Mode Status Parameters

Parameter Name	Memory Bank	Status Bit Field
Power Domain Last Power State Entered Status		PM_IPU_PWRSTST[25:24] LASTPOWERSTATEENTERED
Memory Area State Status	AESSMEM	PM_IPU_PWRSTST[5:4] AESSMEM_STATEST
Memory Area State Status	PERIPHMEM	PM_IPU_PWRSTST[9:8] PERIPHMEM_STATEST
Power Domain State Transition Status		PM_IPU_PWRSTST[20] INTRANSITION
Logic Area State Status		PM_IPU_PWRSTST[2] LOGICSTATEST
Power Domain State Status		PM_IPU_PWRSTST[1:0] POWERSTATEST

3.7.6 PD_DSS Description

PD_DSS contains the following reset domains:

- DSS_RET_RST
- DSS_RST

PD_DSS contains the CD_DSS clock domain.

Table 3-250 lists the logic retention capability for each module of the power domain.

Table 3-250. PD_DSS Modules Power Attributes

Module	Logic Retention	DFF Context Status	RFF Context Status
DSS	Partial	RM_DSS_DSS_CONTEXT[0] LOSTCONTEXT_DFF	RM_DSS_DSS_CONTEXT[1] LOSTCONTEXT_RFF

3.7.6.1 Power Domain Modes

This section describes the various power modes supported by the logic and memory areas of the power domain. It also identifies the associated software control and status bit fields. For a functional description of the power-management architecture of a generic power domain, see Section 3.1.1.2.1, *Power Domain*.

3.7.6.1.1 Logic and Memory Area Power Modes

Table 3-251 lists the power modes supported by the logic area of the power domain.

Table 3-251. PD_DSS Logic Area Power Modes

Off	Retention-CSWR	On-Inactive	On-Active
Available	Available	Not available	Available

Table 3-252 lists the power modes supported by the memory area of the power domain. A memory area power mode is identified with respect to a power state of the power domain. The Logic On, Logic Retention, and Logic Off columns identify the power states of the power domain. The values in these columns identify the supported power state of the memory area (identified in the Memory Bank column) or the module memory inside the memory area (identified in the Module Memory column).

Table 3-252. PD_DSS Memory Area Power Modes

Memory Bank	Module Memory	Logic On	Logic Retention	Logic Off
DSS_MEM		ON		OFF
	DSS DSSMEM	always_on	always_off	

3.7.6.1.2 Logic and Memory Area Power Mode Control and Status

Table 3-253 lists the power modes controls for the power domain.

Table 3-253. PD_DSS Power Modes Control Parameters

Parameter Name	Memory Bank	Control Bit Field	Access Type
Memory Area State Control (logic in RETENTION state)	DSS_MEM	PM_DSS_PWRSTCTRL[8] DSS_MEM_RETSTATE	Read only
Power Domain Low-Power State Change Control		PM_DSS_PWRSTCTRL[4] LOWPOWERSTATECHANGE	Read/write
Logic Area RETENTION State Control		PM_DSS_PWRSTCTRL[2] LOGICRETSTATE	Read only
Memory Area State Control (logic in ON state)	DSS_MEM	PM_DSS_PWRSTCTRL[17:16] DSS_MEM_ONSTATE	Read only
Power Domain State Transition Control		PM_DSS_PWRSTCTRL[1:0] POWERSTATE	Read/write

Table 3-254 lists the status of the power modes for the power domain.

Table 3-254. PD_DSS Power Modes Status Parameters

Parameter Name	Memory Bank	Status Bit Field
Memory Area State Status	DSS_MEM	PM_DSS_PWRSTST[5:4] DSS_MEM_STATEST
Power Domain Last Low Power State Entered Status		PM_DSS_PWRSTST[25:24] LASTPOWERSTATEENTERED
Power Domain State Transition Status		PM_DSS_PWRSTST[20] INTRANSITION
Logic Area State Status		PM_DSS_PWRSTST[2] LOGICSTATEST
Power Domain Current Power State Status		PM_DSS_PWRSTST[1:0] POWERSTATEST

3.7.7 PD_CAM Description

PD_CAM contains the CAM_RST reset domain.

PD_CAM contains the CD_CAM clock domain.

Table 3-255 lists the logic retention capability for each module of the power domain.

Table 3-255. PD_CAM Modules Power Attributes

Module	Logic Retention	DFF Context Status	RFF Context Status
VIP1	No	RM_CAM_VIP1_CONTEXT[0] LOSTCONTEXT_DFF	None

3.7.7.1 Power Domain Modes

This section describes the various power modes supported by the logic and memory areas of the power domain. It also identifies the associated software control and status bit fields. For a functional description of the power-management architecture of a generic power domain, see Section 3.1.1.2.1, *Power Domain*.

3.7.7.1.1 Logic and Memory Area Power Modes

Table 3-256 lists the power modes supported by the logic area of the power domain.

Table 3-256. PD_CAM Logic Area Power Modes

Off	Retention-CSWR	On-Inactive	On-Active
Available	Not available	Available	Available

Table 3-257 lists the power modes supported by the memory area of the power domain. A memory area power mode is identified with respect to a power state of the power domain. The Logic On, Logic Retention, and Logic Off columns identify the power states of the power domain. The values in these columns identify the supported power state of the memory area (identified in the Memory Bank column) or the module memory inside the memory area (identified in the Module Memory column).

Table 3-257. PD_CAM Memory Area Power Modes

Memory Bank	Module Memory	Logic On	Logic Retention	Logic Off
VIP_BANK		ON		OFF
	VIP1 VIP_MEM	always_on	always_retention	

3.7.7.1.2 Logic and Memory Area Power Mode Control and Status

Table 3-258 lists the power mode controls for the power domain.

Table 3-258. PD_CAM Power Mode Control Parameters

Parameter Name	Memory Bank	Control Bit Field	Access Type
Power Domain Low-Power State Change Control		PM_CAM_PWRSTCTRL[4] LOWPOWERSTATECHANGE	Read/write
Memory Area State Control (Logic in ON state)	VIP_BANK	PM_CAM_PWRSTCTRL[17:16] VIP_BANK_ONSTATE	Read only
Power Domain State Transition Control		PM_CAM_PWRSTCTRL[1:0] POWERSTATE	Read/write

Table 3-259 lists the status of the power modes for the power domain.

Table 3-259. PD_CAM Power Modes Status Parameters

Parameter Name	Memory Bank	Status Bit Field
Memory Area State Status	VIP_BANK	PM_CAM_PWRSTST[5:4] VIP_BANK_STATEST
Power Domain Last Power State Entered Status		PM_CAM_PWRSTST[25:24] LASTPOWERSTATEENTERED

Table 3-259. PD_CAM Power Modes Status Parameters (continued)

Parameter Name	Memory Bank	Status Bit Field
Power Domain State Transition Status		PM_CAM_PWRSTST[20] INTRANSITION
Logic Area State Status		PM_CAM_PWRSTST[2] LOGICSTATEST
Power Domain State Status		PM_CAM_PWRSTST[1:0] POWERSTATEST

3.7.8 PD_MMAON Description

PD_MMAON contains the following reset domains:

- MMAON_RST

PD_MMAON has no associated clock domains.

Table 3-260 lists the logic retention capability for each module of the power domain.

Table 3-260. PD_MMAON Module Power Attributes

Module	Logic Retention	DFF Context Status	RFF Context Status
DSP SYS Wakeup Logic	No	None	None

3.7.8.1 Power Domain Modes

The PD_MMAON power domain is an always-on power domain and does not switch to RETENTION state. There is no logic power-state control or status bit field for this power domain.

The PD_MMAON power domain has no memory banks.

3.7.9 PD_COREAON Description

PD_COREAON contains the following reset domains:

- CM_CORE_AON_PWRON_RST
- CM_CORE_AON_RST
- COREAON_PWRON_RST
- COREAON_RST
- IPU_RST
- IPU_RET_RST
- L4PER_PWRON_RET_RST
- L4PER_RET_RST
- L4PER_RST
- EMU_EARLY_PWRON_RST
- EMU_PWRON_RST
- EMU_RST
- DLL_RST
- DMA_RET_RST
- DPLL_DSP_PWRON_RST
- DPLL_EVE_PWRON_RST

PD_COREAON contains the the following clock domains: CD_COREAON_L4, CD_MMAON, CD_IPU, CD_EMU, CD_L4_PER1, CD_L4_PER2, CD_L4_PER3, CD_L4_CFG, CD_EMIF, CD_L3_MAIN1, CD_CRC, CD_DMA, CD_L3_INSTR.

Table 3-261 lists the logic retention capability for each module of the power domain.

Table 3-261. PD_COREAON Module Power Attributes

Module	Logic Retention	DFF Context Status	RFF Context Status
CM_CORE_AON	No	None	None
DPLL_CORE	No	None	None
DPLL_PER	No	None	None
DPLL_DDR	No	None	None
DPLL_GMAC_DSP	No	None	None
DPLL_EVE_VID_DSP	No	None	None
McASP1	No	None	None
TIMER5	No	None	None
TIMER6	No	None	None
TIMER7	No	None	None
TIMER8	No	None	None
IEEE1500_2_OCP	No	None	None
OCP2SCP1	No	None	None
GMAC_SW	No	None	None
MCAN	No	None	None
ATL	No	None	None
McASP2	No	None	None
McASP3	No	None	None
ELM	No	None	None
GPIO2	No	None	None
GPIO3	No	None	None
GPIO4	No	None	None
ESM(1)	No	None	None
I2C1	No	None	None
I2C2	No	None	None
L4_PER1	No	None	None
L4_PER2	No	None	None
L4_PER3	No	None	None
ADC	No	None	None
McSPI1	No	None	None
McSPI2	No	None	None
McSPI3	No	None	None
McSPI4	No	None	None
MMC	No	None	None
PWMSS1	No	None	None
QSPI	No	None	None
TIMER2	No	None	None
TIMER3	No	None	None
TIMER4	No	None	None
DCC5(2)	No	None	None
DCC6(2)	No	None	None
DCC7(2)	No	None	None
DCC1(2)	No	None	None
DCC2(2)	No	None	None
DCC3(2)	No	None	None
DCC4(2)	No	None	None

(1) ESM is not supported on the DRA78x family of devices.
(2) DCC modules are not supported on the DRA78x family of devices.

Table 3-261. PD_COREAON Module Power Attributes (continued)

Module	Logic Retention	DFF Context Status	RFF Context Status
UART1	No	None	None
UART2	No	None	None
UART3	No	None	None
DEBUGSS	No	None	None
CONTROL_MODULE_CORE	No	None	None
CONTROL_MODULE_BANDGAP	No	None	None
CM_CORE	No	None	None
CRC	No	None	None
DLL	No	None	None
DLL_AGING	No	None	None
EMIF1	No	None	None
GPMC	No	None	None
SPINLOCK	No	None	None
L3_MAIN_1 interconnect	No	None	None
L3_MAIN_2 interconnect	No	None	None
L3_INSTR interconnect	No	None	None
L4_CFG interconnect	No	None	None
MAILBOX1	No	None	None
MAILBOX2	No	None	None
OCMC_RAM1	No	None	None
TESOC[3]	No	None	None
OCP_WP_NOC	No	None	None
MMU_EDMA	No	None	None
TPCC	No	None	None
TPTC1	No	None	None
TPTC2	No	None	None
VCP1	No	None	None
WUGEN_IPU	No	None	None
SPINNER	No	None	None

(3) TESOC is not supported on the DRA78x family of devices.

3.7.9.1 Power Domain Modes

The PD_COREAON power domain is an always-on power domain and does not switch to RETENTION state. There is no logic power-state control or status bit field for this power domain.

The PD_COREAON power domain has no memory banks.

3.7.10 PD_ISS Description

NOTE: ISS is not supported on the DRA78x family of devices.

PD_ISS contains the following reset domains:

- ISS_RST

PD_ISS contains the CD_ISS clock domain.

Table 3-262 lists the logic retention capability for each module of the power domain.

Table 3-262. PD_ISS Modules Power Attributes

Module	Logic Retention	DFF Context Status	RFF Context Status
ISS	No	RM_ISS_ISS_CONTEXT[0] LOSTCONTEXT_DFF	None

3.7.10.1 Power Domain Modes

This section describes the various power modes supported by the logic and memory areas of the power domain. It also identifies the associated software control and status bit fields. For a functional description of the power-management architecture of a generic power domain, see Section 3.1.1.2.1, *Power Domain*.

3.7.10.1.1 Logic and Memory Area Power Modes

Table 3-263 lists the power modes supported by the logic area of the power domain.

Table 3-263. PD_ISS Logic Area Power Modes

Off	Retention-CSWR	On-Inactive	On-Active
Available	Available	Available	Available

Table 3-264 lists the power modes supported by the memory area of the power domain. A memory area power mode is identified with respect to a power state of the power domain. The Logic On, Logic Retention, and Logic Off columns identify the power states of the power domain. The values in these columns identify the supported power state of the memory area (identified in the Memory Bank column) or the module memory inside the memory area (identified in the Module Memory column).

Table 3-264. PD_ISS Memory Area Power Modes

Memory Bank	Module Memory	Logic On	Logic Retention	Logic Off
ISS_BANK		ON	RETENTION	OFF
	ISS ISS_MEM	always_on	always_retention	

3.7.10.1.2 Logic and Memory Area Power Modes Control and Status

Table 3-265 lists the power mode controls for the power domain.

Table 3-265. PD_ISS Power Modes Control Parameters

Parameter Name	Memory Bank	Control Bit Field	Access Type
Memory Area State Control (Logic in ON state)	ISS_BANK	PM_ISS_PWRSTCTRL[17:16] ISS_BANK_ONSTATE	Read only
Memory Area State Control (logic in RETENTION state)	ISS_BANK	PM_ISS_PWRSTCTRL[8] ISS_BANK_RETSTATE	Read/write
Power Domain Low-Power State Change Control		PM_ISS_PWRSTCTRL[4] LOWPOWERSTATECHANGE	Read/write
Logic Area Retention State Control		PM_ISS_PWRSTCTRL[2] LOGICRETSTATE	Read/write
Power Domain State Transition Control		PM_ISS_PWRSTCTRL[1:0] POWERSTATE	Read/write

Table 3-266 lists the status of the power modes for the power domain.

Table 3-266. PD_ISS Power Modes Status Parameters

Parameter Name	Memory Bank	Status Bit Field
Power Domain Last Power State Entered Status		PM_ISS_PWRSTST[25:24] LASTPOWERSTATEENTERED

Table 3-266. PD_ISS Power Modes Status Parameters (continued)

Parameter Name	Memory Bank	Status Bit Field
Memory Area State Status	ISS_BANK	PM_ISS_PWRSTST[5:4] ISS_BANK_STATEST
Power Domain State Transition Status		PM_ISS_PWRSTST[20] INTRANSITION
Logic Area State Status		PM_ISS_PWRSTST[2] LOGICSTATEST
Power Domain State Status		PM_ISS_PWRSTST[1:0] POWERSTATEST

3.7.11 PD_EVE1 Description

PD_EVE1 contains the following reset domains:

- EVE1_CPU_RST
- EVE1_PWRON_RST
- EVE1_RST

PD_EVE1 contains the CD_EVE1 clock domain.

Table 3-267 lists the logic retention capability for each module of the power domain.

Table 3-267. PD_EVE1 Modules Power Attributes

Module	Logic Retention	DFF Context Status	RFF Context Status
EVE1	No	RM_EVE1_EVE1_CONTEXT[0] LOSTCONTEXT_DFF	None

3.7.11.1 Power Domain Modes

This section describes the various power modes supported by the logic and memory areas of the power domain. It also identifies the associated software control and status bit fields. For a functional description of the power-management architecture of a generic power domain, see Section 3.1.1.2.1, *Power Domain*.

3.7.11.1.1 Logic and Memory Area Power Modes

Table 3-268 lists the power modes supported by the logic area of the power domain.

Table 3-268. PD_EVE1 Logic Area Power Modes

Off	Retention-CSWR	On-Inactive	On-Active
Available	N/A	Available	Available

Table 3-269 lists the power modes supported by the memory area of the power domain. A memory area power mode is identified with respect to a power state of the power domain. The Logic On, Logic Retention, and Logic Off columns identify the power states of the power domain. The values in these columns identify the supported power state of the memory area (identified in the Memory Bank column) or the module memory inside the memory area (identified in the Module Memory column).

Table 3-269. PD_EVE1 Memory Area Power Modes

Memory Bank	Module Memory	Logic On	Logic Retention	Logic Off
EVE1_BANK		ON	RETENTION	OFF
	EVE1 EVE_BANK	always_on	always_off	

3.7.11.1.2 Logic and Memory Area Power Modes Control and Status

Table 3-270 lists the power mode controls for the power domain.

Table 3-270. PD_EVE1 Power Modes Control Parameters

Parameter Name	Memory Bank	Control Bit Field	Access Type
Memory Area State Control (Logic in ON state)	EVE1_BANK	PM_EVE1_PWRSTCTRL[17:16] EVE1_BANK_ONSTATE	Read only
Power Domain Low-Power State Change Control		PM_EVE1_PWRSTCTRL[4] LOWPOWERSTATECHANGE	Read/write
Power Domain State Transition Control		PM_EVE1_PWRSTCTRL[1:0] POWERSTATE	Read/write

Table 3-271 lists the status of the power modes for the power domain.

Table 3-271. PD_EVE1 Power Modes Status Parameters

Parameter Name	Memory Bank	Status Bit Field
Power Domain Last Power State Entered Status		PM_EVE1_PWRSTST[25:24] LASTPOWERSTATEENTERED
Memory Area State Status	EVE1_BANK	PM_EVE1_PWRSTST[5:4] EVE1_BANK_STATEST
Power Domain State Transition Status		PM_EVE1_PWRSTST[20] INTRANSITION
Logic Area State Status		PM_EVE1_PWRSTST[2] LOGICSTATEST
Power Domain State Status		PM_EVE1_PWRSTST[1:0] POWERSTATEST

3.8 Voltage-Management Functional Description

This section describes the voltage domains and voltage control architecture. It also explains the interactions between the device and the external power IC.

NOTE: For more information about the device power pads connection, see the device data manual.

3.8.1 Overview

The voltage-management architecture of the device is based on voltage sources managed by the PRCM module. They define the voltage domains within the device (see Section 3.1.1.3, *Voltage Management*). This partition of the voltage domains ensures independent voltage control of each voltage domain through dedicated LDO. The following voltage domains are managed by the PRCM module:

- VDD_CORE_L
- VDD_DSPEVE_L

NOTE: For the association of the device power supply pin to the power domain, see Table 3-26.

The PRCM module supports the AVS technique for all power domains VDD_CORE_L and VDD_DSPEVE_L voltage domains.

At boot time, the device is set with VDD_CORE_L at OPP_NOM.

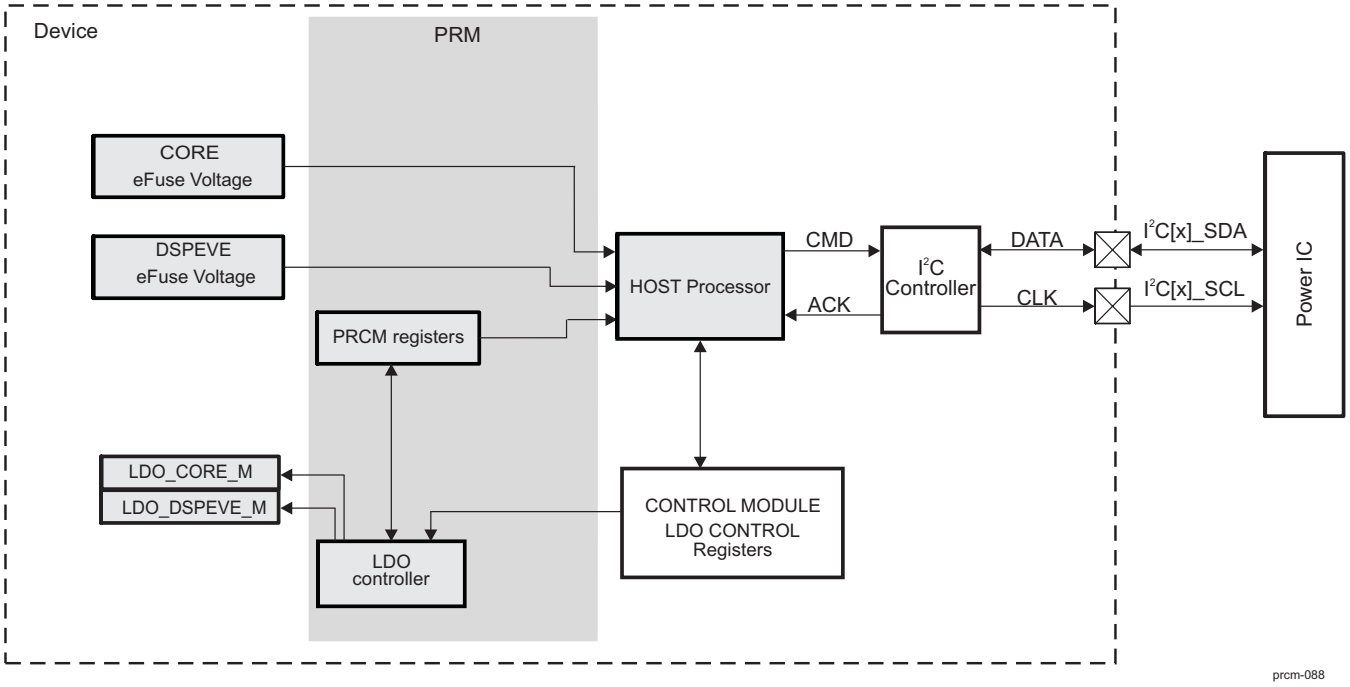
3.8.2 Voltage-Control Architecture

The PRM is split over several blocks that manage the different voltage sources.

- A device PRCM for managing the I/O wake-up control and system clock control sequencing during device sleep and wake-up transitions.
- LDO regulator controllers for memory arrays voltage management, and WAKEUP logic.
- BANDGAPs reference voltage sleep control

Figure 3-67 shows the architecture for PRM voltage control.

Figure 3-67. PRM Voltage Control Architecture



3.8.3 Internal LDOs Control

3.8.3.1 VDD_CORE_L and VDD_DSPEVE_L Control

3.8.3.1.1 Adaptive Voltage Scaling

As explained in Section 3.1.2.4, *Adaptive Voltage Scaling*, the SmartReflex technology. With SmartReflex, the power supply voltage can be adapted to the silicon performance statically (for example, adapted to the manufacturing process of a given device) or dynamically.

3.8.3.2 Memory LDOs

Embedded SRAM LDOs are used to supply power to the split-rail memory arrays. The PRM generates the controls used to select LDO operating mode: on-active, on-retention, or off.

Split-rail type SRAMs are used in the device to implement the larger memories. These SRAMs feature memory array and periphery logic, which are on separate supplies to allow independent power management. Proper memory operation, however, requires the SRAM array voltage never to be operated at a level lower than the SRAM periphery logic.

Memory LDO can switch to on and retention mode:

- On (active) mode: 1.15 V is the normal voltage reference used through all functional OPPs whenever memories must be functional. When logic voltage level VDD_x_L becomes higher than the associated memory voltage level VDD_x_M, the LDO operates in tracking mode and follows its respective VDD_x_M or VDD_x_ABB voltage level.
- Retention mode: 0.6 V is set when software allows and when all memory banks belonging to the LDO memory voltage are in RETENTION or OFF state. In this mode, the output voltage is generated from the corresponding VDD_WKUP_L logic voltage source.

Section 3.8.3.4, *Memory LDOs Transitions*, describes the state transition conditions and sequences for the memory LDOs.

NOTE: The voltage levels associated with the different modes may depend on the device characteristics.

3.8.3.3 BANDGAP Control

BANDGAPs provides voltage reference for internal LDOs. The PRCM module automatically controls the switching between ON and OFF states of the BANDGAPs, based on the power state of the device. It is completely transparent to user software.

BANDGAPs startup time is 100µs. The PRM_BANDGAP_SETUP[7:0] STARTUP_COUNT bit field must be set accordingly.

3.8.3.4 Memory LDO Transitions

The transition trigger condition for on-to-retention mode is:

- At least all memory banks associated with the LDO are in RETENTION state.

The transition sequence is:

1. Transition trigger conditions are true (that is, satisfied).
2. Switch LDO reference to:
   - Retention mode value if on-to-retention mode transition
3. Start counter for LDO stabilization.
4. When counter expires, voltage transition is complete.

3.8.3.5 VDD_WKUP_L Transitions

The ON-to-SLEEP transition trigger conditions are:

- Power domain states: Device is in low-power state (STANDBY).
- Clock states: Only the 32K clock is active in CD_WKUPAON.

The SLEEP-to-ON transition trigger conditions are:

- Any device wake-up event other than emulation wake-up event

The EMULATION-to-ON transition trigger conditions are:

- Power domain states: device is in active mode.
- Clock states: any clock other than 32K clock is active in PD_WKUPAON.
- PD_EMU in OFF state.

The EMULATION-to-SLEEP transition trigger conditions are:

- Power domain states: Device is in low-power state (STANDBY).
- Clock states: Only the 32K clock is active in CD_WKUPAON.
- PD_EMU in OFF state.

The SLEEP or ON-to-EMULATION transition trigger condition is:

- Any wake-up event on PD_EMU

The transition sequence is:

1. Transition trigger conditions are met.
2. VDD_WKUP_L initiates the transition.
3. Starts a counter for VDD_WKUP_L transition.
4. VDD_WKUP_L transition completes.

3.9 Device Low-Power States

The device low-power states are the result of any valid combination of power domain states in which all the power domains are no longer in ACTIVE state. In such a situation the PRCM module hardware can trigger events to further lower the consumption of the device and the system.

These device low-power states are characterized by the system power consumption, wake-up latency, and required functionality.

The low-power state is:

- "**STANDBY**" : Any combination of logic voltage domain states (SLEEP or RETENTION) other than ACTIVE state.

Once the PRCM module hardware detects any valid combination of power domain states, and if a proper programming model of the PRCM module is set, the PRCM module automatically triggers the transition into the device low-power mode.

3.9.1 Device Wake-Up Source Summary

The wake-up events can be asynchronous or synchronous. Synchronous wake-up events require the 32-kHz clock or the system clock to be active, while asynchronous wake-up events do not require an active clock.

The *Modules Attributes* subsection of each clock domain in Section 3.6, *Clock Management Functional Description*, describes the wake-up capability support for each module of the corresponding power domain.

While the device is in STANDBY mode, additional asynchronous wakeup events from other domains are able to wake up the device.

Table 3-272 identifies which modules in which power domains can be configured to generate a wake-up while the device is in a low-power mode.

Table 3-272. Wake-Up Sources During Device Low Power Mode

Device Power Mode name	VD_CORE VOLTAGE STATE	VD_DSPEVE VOLTAGE STATE	Domain containing wake-up source	Wakeup sources
Really OFF	OFF	OFF	N/A	Application of Power Supply and Power on Reset Sequence
STANDBY	Active Voltage Level		PD_IPU (1)	IPU1, McASP1, TIMER5 ,TIMER6,TIMER7, TIMER8
			PD_CORE	OCMC_RAM1, TESOC(2), EDMA_TPCC, EDMA_TPTC1, EDMA_TPTC2
			PD_L3INIT (1)	IEEE1500_2_OCP
			PD_L4PER	MCAN, GPIO2, GPIO3, GPIO4, I2C1, I2C2, McSPI1, McSPI2, McSPI3, McSPI4, QSPI, TIMER2, TIMER3, TIMER4, UART1, UART2, UART3, ADC
			PD_DSP1	DSP1
			PD_DSP2	DSP2
			PD_EVE1	EVE1
			PD_WKUPAON	DCAN1, GPIO1,TIMER1
			PD_EMU	(Debug_logic) Any ForceActive directive Dynamic depedency towards L3_MAIN

(1) Only modules able to generate asynchronous wake-up have to be taken into account, and only when domain is in INACT or CSWRET state.

(2) TESOC is not supported on the DRA78x family of devices.

3.9.2 Wakeup Upon Global Warm Reset

When global warm reset is the source of the device wakeup, the sequence is modified as follows:

- The PRCM module releases its reset line. In parallel, the device reset manager counts for global reset extension (set up by the PRM_RSTTIME[9:0] RSTTIME1 bit field). The PRCM module holds the other asserted resets until the global reset counter overflows.

The hardware blocks and modifies the CM_MPU_CLKSTCTRL[1:0] CLKTRCTRL bit fields.CM_SHADOW_FREQ_CONFIG1[0] FREQ_UPDATE and

3.9.3 Global Warm Reset During a Device Wake-Up Sequence

If a global warm reset occurs before the PRCM module completes the voltage stabilization count, the global warm reset is applied immediately. As a consequence, the sequence described in Section 3.9.2, *Wakeup Upon Global Warm Reset*, is performed.

If the global warm reset occurs after the PRCM module completes voltage stabilization (during phase 1 of automatic restore):

1. Global warm reset is delayed and applied after phase 1 restore completes.
2. Phase 2 of restore is discarded.
3. IPU boots after the warm reset sequence completes.

If the global warm reset occurs during phase 2 of automatic restore:

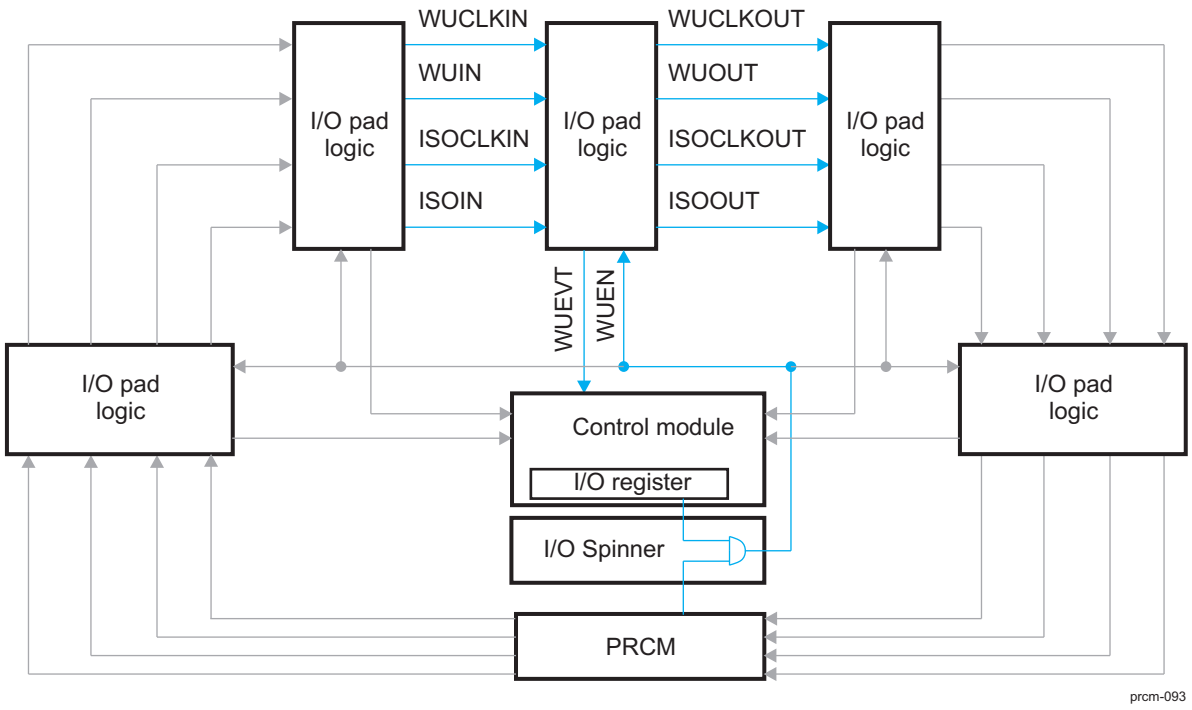
1. Global warm reset is applied immediately.
2. Phase 2 of restore is discarded.
3. IPU boots after the warm reset sequence completes.

3.9.4 I/O Management

Figure 3-68 is an overview of the power-management modules and their internal connections with a generic power domain.

ISOIN, ISOCLKIN, WUCLKIN, WUIN, ISOOUT, ISOCLKOUT, WUCLKOUT and WUOUT connections are built with daisy chaining approach and are all VDD_WKUP_L level signals driven by PRM. WUEN and WUEVNT are VDD level signals driven by the control module and/or IO control module. ISOOVR and ISOBYPASS (for DDR IO only) are VDD level signals driven by the PRM.

Figure 3-68. I/O Pads Daisy-Chain Configuration



NOTE: ISOBYPASS control from PRM is routed to all the EMIF i/f IOs and ISOOVR control from PRM is routed to all the remaining IOs. These control signals are routed using ALWON VDD buffers. The ISOOVR pin on EMIF IOs is tied to 0 using ALWON tie-off cells. PRM also provides an output status "IO_ISO_ACTIVE" which is set when EMIF IOs are isolated or transitioning between isolation and functional mode, and which is cleared when EMIF IOs are functional.

it is required that the ISOCLKIN is forced to 1 during global power-on reset (PRM_PWRON_RST_n = 0).

3.9.4.1 Isolation / Wakeup Sequence

Enabling Wake-Up Feature:

- Program Control module MMR to assert "WKEN" for each IO (To enable Wakeup feature of IO)
- Write the bit PRM_IO_PMCTRL[8] WUCLK_CTRL to 1 to assert high the signal WUCLKIN.
- Write the bit PRM_IO_PMCTRL[8] WUCLK_CTRL to 0 to assert low the signal WUCLKIN.
- This will latch WKEN, Latch the current pad input value.
- The PRCM register PRM_IO_PMCTRL[9] WUCLK_STATUS logs the signal WUCLK of the last pad of the IO ring.(Should be 0).

Device goes into sleep mode (There is no need to put IO in ISOLATION and hence no need to toggle ISOCLKIN and ISOIN (As core supply is still ON)) :

- WKUP event is generated by one of the IOs

- WUOUT of the last IO is asserted HIGH.
- Because of #9, PRM interrupt is generated towards MPU/Host processor
- MPU/Host processor disables the WUKP feature of each IO and power up the required domains.

DISABLING WKUP feature:

- Write the bit PRM_IO_PMCTRL[8] WUCLK_CTRL to 1 to assert high the signal WUCLKIN.
- Write the bit PRM_IO_PMCTRL[8] WUCLK_CTRL to 0 to assert low the signal WUCLKIN.
- The PRCM register PRM_IO_PMCTRL[9] WUCLK_STATUS logs the signal WUOUT of the last pad of the IO ring.

3.9.4.1.1 Software-Controlled I/O Isolation

The PRM_IO_PMCTRL[4] ISOOVR_EXTEND bit allows extending the non-EMIF I/O isolation. This feature can be used by software to restore modules driving output, such as GPIO, while non-EMIF I/Os are still isolated. Once software completes the relevant module restore, it clears the bit and hardware performs full-isolation-to-EMIF on the hardware-controlled I/O transition.

The PRM_IO_PMCTRL[5] IO_ON_STATUS bit is available for software to check completion of the EMIF on transition.

3.10 PRCM Module Programming Guide

3.10.1 DPLLs Low-Level Programming Models

This section describes the low-level hardware programming sequences for the configuration and use of the module.

3.10.1.1 Global Initialization

3.10.1.1.1 Surrounding Module Global Initialization

This section identifies the requirements for initializing the surrounding modules when the module is to be used for the first time after a device reset. This initialization of surrounding modules is based on the integration and environment of the DPLLs.

Table 3-273 describes the global initialization of the surrounding modules.

Table 3-273. Global Initialization of Surrounding Modules

Surrounding Modules	Comments
PRCM	Ensure that the DPLL reference clock (gated version of system clock) is active.

3.10.1.1.2 DPLL Global Initialization

3.10.1.1.2.1 Main Sequence DPLL Global Initialization

This procedure initializes the DPLL after a POR or software reset and then locks it to the desired synthesized clock frequency.

Table 3-274. DPLL Global Initialization

Step	Register/Bit Field/Programming Model	Value
Configure recalibration parameters.	See Section 3.10.1.1.2.2.	
Set DPLL automatic idle mode.	CM_AUTOIDLE_<DPLL name>[2:0] AUTO_DPLL_MODE	xx(1)
Configure synthesized clock parameters.	See Section 3.10.1.1.2.3.	
Configure output clocks parameters.	See Section 3.10.1.1.2.4.	
Lock DPLL.	CM_CLKMODE_<DPLL name>[2:0] DPLL_EN	0x7

(1) It depends on the desired auto idle mode. See Section 3.6.3.4.3, *DPLL Power Modes*.

3.10.1.1.2.2 Subsequence Recalibration Parameter Configuration

This procedure enables the recalibration feature and the associated processor interrupt flag.

Table 3-275. DPLL Recalibration Parameter Configuration

Step	Register/Bit Field/Programming Model	Value
Clear recalibration interrupt status.	PRM_IRQSTATUS _<Processor name>[x] <DPLL name>_RECAL_ST	0x0
Unmask recalibration interrupt flag.	PRM_IRQENABLE _<Processor name>[x] <DPLL name>_RECAL_EN	0x1
Enable recalibration feature.	CM_CLKMODE_<DPLL name>[8] DPLL_DRIFTGUARD_EN	0x1

3.10.1.1.2.3 Subsequence Synthesized Clock Parameter Configuration

This procedure configures the settings for the synthesized clock of the DPLL.

Table 3-276. DPLL Synthesized Clock Parameter Configuration

Step	Register/Bit Field/Programming Model	Value
Set DPLL clock synthesis multiplier.	CM_CLKSEL_<DPLL name>[18:8] DPLL_MULT	xx(1)
Set DPLL clock synthesis divider.	CM_CLKSEL_<DPLL name>[6:0] DPLL_DIV	xx(1)
IF : Low-power mode operation conditions satisfied?	Software test condition. See Section 3.6.3.4.2.	
Enable DPLL low-power operation mode.	CM_CLKMODE_<DPLL name>[10] DPLL_LPMODE_EN	0x1
ENDIF		

(1) It depends on the desired synthesized clock frequency. See Section 3.6.3.4, *Generic DPLL Overview*.

3.10.1.1.2.4 Subsequence Output Clock Parameter Configuration

This procedure configures the settings for the output clocks of the DPLL.

Table 3-277. DPLL Output Clock Parameter Configuration

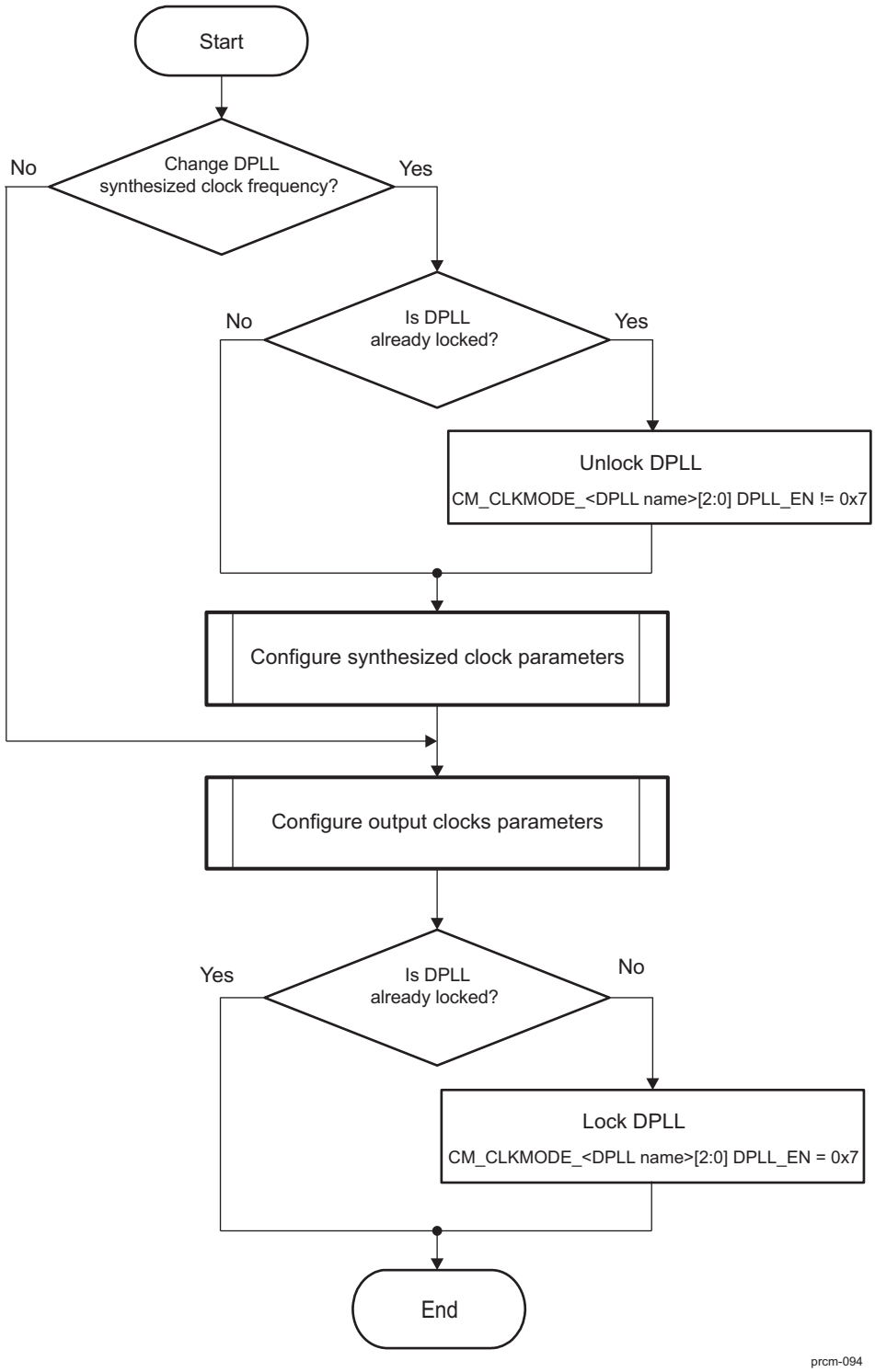
Step	Register/Bit Field/Programming Model	Value
Set output clock dividers (that is, M2, M3, and Hmn), where m is 1 or 2, and n is from 1 to 4. It depends on the available clock output of the DPLL.	CM_DIV_M2_<DPLL name>[4:0] DIVHS CM_DIV_M3_<DPLL name>[4:0] DIVHS CM_DIV_Hmn_<DPLL name>[5:0] DIVHS	xx(1)

(1) It depends on the desired output clock frequency. See Section 3.6.3.4, *Generic DPLL Overview*.

3.10.1.2 DPLL Output Frequency Change

Figure 3-69 shows the DPLL output-frequency change.

Figure 3-69. DPLL Output-Frequency Change



To unlock a DPLL, a mode different from the Lock Mode (0x7) should be programmed in the CM_CLKMODE_<DPLL NAME>[2:0] DPLL_EN bit field. The modes that can be programmed in the DPLL_EN bit field and can unlock the DPLL are:

- For all DPLLs: Idle Low Power bypass mode (0x5) and Idle Fast Relock bypass mode (0x6)

Table 3-278 and Table 3-279 summarize register and subprocess call sequences for DPLL output frequency changes.

Table 3-278. Register Call Summary for Sequence DPLL Output Frequency Change

Register Name
CM_CLKMODE_DPLL name

Table 3-279. Subprocess Call Summary for Sequence DPLL Output Frequency Change

Subprocess Name	Cross-Reference
Configure synthesized clock parameters.	See Section 3.10.1.1.2.3.
Configure output clocks parameters.	See Section 3.10.1.1.2.4.

3.10.2 Clock Management Low-Level Programming Models

This section describes the low-level hardware programming sequences for the configuration and use of the clocks in the device.

3.10.2.1 Global Initialization

3.10.2.1.1 Surrounding Module Global Initialization

This section identifies the requirements for initializing the surrounding modules when the module is to be used for the first time after a device reset. This initialization of surrounding modules is based on the integration and environment of the DPLLs.

Table 3-280 describes the global initialization of the surrounding modules.

Table 3-280. Global Initialization of Surrounding Modules

Surrounding Modules	Comments
PRCM DPLLs	Ensure that the DPLLS managed by the PRCM module are initialized.

3.10.2.1.2 Clock Management Global Initialization

3.10.2.1.2.1 Main Sequence Clock Domain Global Initialization

This procedure initializes the clock domain of the device after a POR or software reset.

Table 3-281. Clock Domain Global Initialization

Step	Register/Bit Field/Programming Model	Value
Configure module clock-management feature of master modules in the clock domain.	<Module name>_SYSCONFIG[x] MIDLEMODE <Module name>_SYSCONFIG[x] STANDBYMODE	xx(1)
Configure module clock-management feature of slave modules in the clock domain.	See Section 3.10.3.3.	
Enable/disable static sleep dependency with other clock domains (that is, destination clock domains). Not all dependencies are configurable.	CM_<Clock Domain name>_STATICDEP[x] <Destination Clock Domain name>_STATDEP	0x0: Disable 0x1: Enable
Set dynamic dependency window size.	CM_<Clock Domain name>_DYNAMICDEP[27:24] WINDOWSIZE	xx(2)

(1) See the module register for valid modes.
(2) It depends on the desired size of the window. See Section 3.1.1.1.7.2, *Clock Domain Dependency*.

Table 3-281. Clock Domain Global Initialization (continued)

Step	Register/Bit Field/Programming Model	Value
Enable/disable module wake-up dependency for the modules of the clock domain. It is available when the module can generate an interrupt or a DMA request to a service provider module (for example, a processor or DMA).	PM_<Clock Domain name>_<Module name>_WKDEP[x] WKUPDEP_<Module name>_<DMA/IRQ reques>t_<DMA/Processor name>	0x0: Disable 0x1: Enable
Set clock domain state transition feature.	CM_<Clock Domain name>_CLKSTCTRL[1:0] CLKTRCTRL	xx(3)

(3) It depends on the desired state of the clock domain. See Table 3-12.

3.10.2.1.2.2 Subsequence Slave Module Clock-Management Parameters Configuration

This procedure configures the SSC parameters for the DPLL and enables the SSC feature.

Table 3-282. Slave Module Clock-Management Parameter Configuration

Step	Register/Bit Field/Programming Model	Value
Configure module idle mode feature.	<Module name>_SYSCONFIG[x] SIDLEMODE <Module name>_SYSCONFIG[x] IDLEMODE	xx(1)
IF : Smart-idle mode is selected	<Module name>_SYSCONFIG[x] SIDLEMODE <Module name>_SYSCONFIG[x] IDLEMODE	0x10
Configure module clock requirement feature.	<Module name>_SYSCONFIG[x] CLOCKACTIVITY	x(1)
ENDIF		
Configure module management behavior on the PRCM module side.	CM_<Clock Domain name>_<Module name>_CLKCTRL[1:0] MODULEMODE	xx(2)

(1) See the module register for valid settings.
(2) The selected value depends on the desired clock-management behavior.

3.10.2.2 Clock Domain Sleep Transition and Troubleshooting

This procedure initiates a sleep transition on a clock domain and allows debugging if the transition does not occur.

Table 3-283. Clock Domain Sleep Transition and Troubleshooting

Step	Register/Bit Field/Programming Model	Value
Set clock domain sleep transition state.	CM_<Clock Domain name>_CLKSTCTRL[1:0] CLKTRCTRL	0x1: SW_SLEEP 0x3: HW_AUTO
IF : Clock domain sleep transition not initiated?		
Check that all clock domain master modules are in standby mode.	CM_<Clock Domain name>_<Module name>_CLKCTRL[18] STBYST	0x1: Module in standby
Check that all clock domain slave modules are in idle mode.	CM_<Clock Domain name>_<Module name>_CLKCTRL[17:16] IDLEST	0x1: In transition 0x2: Interface clock idled 0x3: Module idled
ENDIF		

3.10.2.3 Enable/Disable Software-Programmable Static Dependency

To change the setting of a software-programmable static dependency, use the procedure described in Table 3-284.

Table 3-284. Enable/Disable Software-Programmable Static Dependency

Step	Register/Bit Field/Programming Model	Value
Force destination domain to be awake (SW_WKUP).	CM_<Dest_CDname>_CLKSTCTRL[1:0] CLKTRCTRL	0x2

Table 3-284. Enable/Disable Software-Programmable Static Dependency (continued)

Step	Register/Bit Field/Programming Model	Value
Wait until power domain that encloses the destination domain is ON.	PM_<Dest_PDname>_PWRSTST	=0x3
Change the static dependency.	CM_<Src_CDname>_STATICDEP[x] Dest_CDname_STATDEP	0x1
Put destination domain back to automatic transition (HW_AUTO).	CM_<Dest_CDname> _CLKSTCTRL[1:0] CLKTRCTRL	0x3

3.10.3 Power Management Low-Level Programming Models

This section describes the low-level hardware programming sequences for the configuration and control of the power domain in the device.

3.10.3.1 Global Initialization

3.10.3.1.1 Surrounding Module Global Initialization

Initialization of any surrounding modules within the device is not required. The external power IC and the device clocks should be active.

3.10.3.1.2 Power Management Global Initialization

3.10.3.1.2.1 Main Sequence Power Domain Global Initialization and Setting

This procedure initializes the power domain of the device after a POR or software reset.

Table 3-285. Power Domain Global Initialization

Step	Register/Bit Field/Programming Model	Value
Configure memory area power state when the power domain is on. Not all memory area states are programmable.	PM_<Power Domain name>_PWRSTCTRL[x] <Memory Bank name>_ONSTATE	0x0: OFF 0x1: RETAINED 0x3: ON
Configure memory area power state when the power domain transitions to RETENTION state. Not all memory area states are programmable.	PM_<Power Domain name>_PWRSTCTRL[x] <Memory Bank name>_RETSTATE	0x0: OFF 0x1: RETAINED
Configure logic area RETENTION power state when the power domain transitions to RETENTION state.	PM_<Power Domain name>_PWRSTCTRL[2] LOGICRETSTATE	0x1: CSWR
Select target power state of the power domain. Not all states are programmable.	PM_<Power Domain name>_PWRSTCTRL[1:0] POWERSTATE	0x0: OFF 0x1: RETENTION 0x2: ON-INACTIVE 0x3: ON-ACTIVE
Wait until power state change is complete.	PM_<Power Domain name>_PWRSTST[1:0] POWERSTATEST	0x0: OFF 0x1: RETENTION 0x2: ON-INACTIVE 0x3: ON-ACTIVE

3.10.3.2 Forced Memory Area State Change With Power Domain ON

This procedure initiates a forced memory area state change while the power domain is ON.

Table 3-286. Forced Memory Area State Change With Power Domain ON

Step	Register/Bit Field/Programming Model	Value
Configure memory area target power state. Not all memory area states are programmable.	PM_<Power Domain name>_PWRSTCTRL[x] <Memory Bank name>_ONSTATE	0x0: OFF 0x1: RETAINED 0x3: ON

Table 3-286. Forced Memory Area State Change With Power Domain ON (continued)

Step	Register/Bit Field/Programming Model	Value
Get memory area current state.	PM_<Power Domain name>_PWRSTST[x] <Memory Bank name>_STATEST	0x0: OFF 0x1: RETAINED 0x3: ON

3.10.3.3 Forced Power Domain Low-Power State Transition

Table 3-287. Forced Power Domain Low-Power State Transition

Step	Register/Bit Field/Programming Model	Value
Select target low-power state of the power domain. Not all states are programmable.	PM_<Power Domain name>_PWRSTCTRL[1:0] POWERSTATE	0x0: OFF 0x1: RETENTION
Force power domain low-power state transition.	PM_<Power Domain name>_PWRSTCTRL[4] LOWPOWERSTATECHANGE	0x1: Force change
Wait until state change is complete.	PM_<Power Domain name>_PWRSTCTRL[4] LOWPOWERSTATECHANGE	0x0: Change complete
Get current power state.	PM_<Power Domain name>_PWRSTST[1:0] POWERSTATEST	0x0: OFF 0x1: RETENTION

3.11 PRCM Software Configuration for OPP_PLUS

NOTE: Contact your TI software support representative for details on OPP_PLUS configuration.

3.12 PRCM Register Manual

NOTE: The following registers are related to clock control but are located in the device Control Module:

- CTRL_CORE_PRCM_CLKSEL_CONTROL
- CTRL_CORE_PRCM_CLKDIV_CONTROL1
- CTRL_CORE_PRCM_CLKDIV_CONTROL2
- CTRL_CORE_SMA_SW_16
- CTRL_CORE_SMA_SW_17
- CTRL_CORE_SMA_SW_22

3.12.1 Not Supported Functionality (Registers and Bitfields)

NOTE: Table 3-288 shows the PRCM registers and bits which functionality is not supported in this device. Within all PRCM "*Register Description*" sections the non-functional bits are shaded.

Table 3-288. Not Supported Functionality (Registers and Bits)

Register/Field/Value	Register/Field/Value	Register/Field/Value
CM_CLKSEL_ABE	CM_L4PER_GPIO5_CLKCTRL	PRM_SLDO_MPU_SETUP
CM_SSC_DELTAMSTEP_DPLL_CORE	CM_L4PER_GPIO6_CLKCTRL	PRM_SLDO_MPU_CTRL
CM_SSC_MODFREQDIV_DPLL_CORE	CM_L4PER2_PWMSS2_CLKCTRL	PRM_SLDO_GPU_SETUP
CM_CLKMODE_DPLL_MPU	CM_L4PER2_PWMSS3_CLKCTRL	PRM_SLDO_GPU_CTRL
CM_IDLEST_DPLL_MPU	CM_L4PER_I2C3_CLKCTRL	PRM_ABBLDO_MPU_SETUP
CM_AUTOIDLE_DPLL_MPU	CM_L4PER_I2C4_CLKCTRL	PRM_ABBLDO_MPU_CTRL
CM_CLKSEL_DPLL_MPU	CM_L4PER_GPIO7_CLKCTRL	PRM_ABBLDO_GPU_SETUP
CM_DIV_M2_DPLL_MPU	CM_L4PER_GPIO8_CLKCTRL	PRM_ABBLDO_GPU_CTRL
CM_SSC_DELTAMSTEP_DPLL_MPU	CM_L4PER_MMC3_CLKCTRL	PRM_MODEM_IF_CTRL
CM_SSC_MODFREQDIV_DPLL_MPU	CM_L4PER_UART5_CLKCTRL	PRM_VOLTST_MPU
CM_BYPCLK_DPLL_MPU	CM_L4SEC_CLKSTCTRL	PRM_VOLTST_MM
CM_CLKMODE_DPLL_IVA	CM_L4SEC_STATICDEP	PRM_SLDO_IVA_SETUP
CM_IDLEST_DPLL_IVA	CM_L4SEC_DYNAMICDEP	PRM_ABBLDO_DSPEVE_CTRL
CM_AUTOIDLE_DPLL_IVA	CM_L4PER2_MCASP8_CLKCTRL	PRM_ABBLDO_IVA_CTRL
CM_CLKSEL_DPLL_IVA	CM_L4SEC_AES1_CLKCTRL	PRM_SLDO_IVA_CTRL
CM_DIV_M2_DPLL_IVA	CM_L4SEC_AES2_CLKCTRL	PRM_ABBLDO_DSPEVE_SETUP
CM_DIV_M3_DPLL_IVA	CM_L4SEC_DES3DES_CLKCTRL	PRM_ABBLDO_IVA_SETUP
CM_SSC_DELTAMSTEP_DPLL_IVA	CM_L4SEC_FPKA_CLKCTRL	RM_DSS_BB2D_CONTEXT
CM_SSC_MODFREQDIV_DPLL_IVA	CM_L4SEC_RNG_CLKCTRL	RM_DSS_SDVENC_CONTEXT
CM_BYPCLK_DPLL_IVA	CM_L4SEC_SHA2MD51_CLKCTRL	PM_EVE2_PWRSTCTRL
CM_CLKMODE_DPLL_ABE	CM_L4PER2_UART7_CLKCTRL	PM_EVE2_PWRSTST
CM_IDLEST_DPLL_ABE	CM_L4SEC_DMA_CRYPTO_CLKCTRL	RM_EVE2_RSTCTRL
CM_AUTOIDLE_DPLL_ABE	CM_L4PER2_UART8_CLKCTRL	RM_EVE2_RSTST
CM_CLKSEL_DPLL_ABE	CM_L4PER2_UART9_CLKCTRL	PM_EVE2_EVE2_WKDEP
CM_DIV_M3_DPLL_ABE	CM_L4SEC_SHA2MD52_CLKCTRL	RM_EVE2_EVE2_CONTEXT
CM_SSC_DELTAMSTEP_DPLL_ABE	CM_COREAON_IO_SRCOMP_CLKCTRL_RESTORE	PM_EVE3_PWRSTCTRL
CM_SSC_MODFREQDIV_DPLL_ABE	SRCONFIG	PM_EVE3_PWRSTST
CM_SSC_DELTAMSTEP_DPLL_DDR	SRSTATUS	RM_EVE3_RSTCTRL
CM_SSC_MODFREQDIV_DPLL_DDR	SENVAL	RM_EVE3_RSTST
CM_SSC_DELTAMSTEP_DPLL_DSP	SENMIN	PM_EVE3_EVE3_WKDEP

Table 3-288. Not Supported Functionality (Registers and Bits) (continued)

Register/Field/Value	Register/Field/Value	Register/Field/Value
CM_SSC_MODFREQDIV_DPLL_DSP	SENMAX	RM_EVE3_EVE3_CONTEXT
CM_RESTORE_ST	SENAVG	PM_EVE4_PWRSTCTRL
CM_CLKMODE_DPLL_EVE	AVGWEIGHT	PM_EVE4_PWRSTST
CM_IDLEST_DPLL_EVE	NVALUERECIPROCAL	RM_EVE4_RSTCTRL
CM_AUTOIDLE_DPLL_EVE	IRQ_EOI	RM_EVE4_RSTST
CM_CLKSEL_DPLL_EVE	IRQSTATUS_RAW	PM_EVE4_EVE4_WKDEP
CM_DIV_M2_DPLL_EVE	IRQSTATUS	RM_EVE4_EVE4_CONTEXT
CM_DIV_M3_DPLL_EVE	IRQENABLE_SET	PM_GPU_PWRSTCTRL
CM_SSC_DELTAMSTEP_DPLL_EVE	IRQENABLE_CLR	PM_GPU_PWRSTST
CM_SSC_MODFREQDIV_DPLL_EVE	SENERROR	RM_GPU_GPU_CONTEXT
CM_BYPCLK_DPLL_EVE	ERRCONFIG	PM_IPU_I2C5_WKDEP
CM_SSC_DELTAMSTEP_DPLL_GMAC	CM_CLKSEL_ABE_PLL_REF	RM_IPU_I2C5_CONTEXT
CM_SSC_MODFREQDIV_DPLL_GMAC	CM_CLKSEL_ABE_PLL_BYPAS	PM_IPU_UART6_WKDEP
CM_CLKMODE_DPLL_GPU	CM_CLKSEL_ABE_PLL_SYS	RM_IPU_UART6_CONTEXT
CM_IDLEST_DPLL_GPU	CM_CLKSEL_HDMI_PLL_SYS	PM_IVA_PWRSTCTRL
CM_AUTOIDLE_DPLL_GPU	CM_CLKSEL_VIDEO1_PLL_SYS	PM_IVA_PWRSTST
CM_CLKSEL_DPLL_GPU	CM_CLKSEL_VIDEO2_PLL_SYS	RM_IVA_RSTCTRL
CM_DIV_M2_DPLL_GPU	CM_CLKSEL_ADC_GFCLK	RM_IVA_RSTST
CM_DIV_M3_DPLL_GPU	PM_COREAON_SMARTREFLEX_MPU_WKDEP	RM_IVA_IVA_CONTEXT
CM_SSC_DELTAMSTEP_DPLL_GPU	RM_COREAON_SMARTREFLEX_MPU_CONTEXT	RM_IVA_SL2_CONTEXT
CM_SSC_MODFREQDIV_DPLL_GPU	PM_COREAON_SMARTREFLEX_CORE_WKDEP	PM_L3INIT_MMC1_WKDEP
CM_EVE2_CLKSTCTRL	RM_COREAON_SMARTREFLEX_CORE_CONTEXT	RM_L3INIT_MMC1_CONTEXT
CM_EVE2_STATICDEP	PM_COREAON_SMARTREFLEX_GPU_WKDEP	PM_L3INIT_MMC2_WKDEP
CM_EVE2_EVE2_CLKCTRL	RM_COREAON_SMARTREFLEX_GPU_CONTEXT	RM_L3INIT_MMC2_CONTEXT
CM_EVE3_CLKSTCTRL	PM_COREAON_SMARTREFLEX_DSPEVE_WKDEP	PM_L3INIT_USB_OTG_SS2_WKDEP
CM_EVE3_STATICDEP	RM_COREAON_SMARTREFLEX_DSPEVE_CONTEXT	RM_L3INIT_USB_OTG_SS2_CONTEXT
CM_EVE3_EVE3_CLKCTRL	PM_COREAON_SMARTREFLEX_IVAHD_WKDEP	PM_L3INIT_USB_OTG_SS3_WKDEP
CM_EVE4_CLKSTCTRL	RM_COREAON_SMARTREFLEX_IVAHD_CONTEXT	RM_L3INIT_USB_OTG_SS3_CONTEXT
CM_EVE4_STATICDEP	RM_COREAON_DUMMY_MODULE1_CONTEXT	PM_L3INIT_USB_OTG_SS4_WKDEP
CM_EVE4_EVE4_CLKCTRL	RM_COREAON_DUMMY_MODULE2_CONTEXT	RM_L3INIT_USB_OTG_SS4_CONTEXT
CM_IPU_I2C5_CLKCTRL	RM_COREAON_DUMMY_MODULE3_CONTEXT	RM_L3INIT_MLB_SS_CONTEXT
CM_IPU_UART6_CLKCTRL	RM_COREAON_DUMMY_MODULE4_CONTEXT	PM_L3INIT_SATA_WKDEP
CM_MPU_CLKSTCTRL	RM_L3MAIN1_MMU_PCIESS_CONTEXT	RM_L3INIT_SATA_CONTEXT
CM_MPU_STATICDEP	PM_L3MAIN1_OCMC_RAM3_WKDEP	PM_PCIE_PCIESS1_WKDEP
CM_MPU_DYNAMICDEP	RM_L3MAIN1_OCMC_RAM3_CONTEXT	RM_PCIE_PCIESS1_CONTEXT
CM_MPU_MPU_CLKCTRL	RM_L3MAIN1_OCMC_ROM_CONTEXT	PM_PCIE_PCIESS2_WKDEP
CM_MPU_MPU_MPU_DBG_CLKCTRL	RM_L3MAIN1_SPARE_CME_CONTEXT	RM_PCIE_PCIESS2_CONTEXT

Table 3-288. Not Supported Functionality (Registers and Bits) (continued)

Register/Field/Value	Register/Field/Value	Register/Field/Value
CM_SSC_DELTAMSTEP_DPLL_CORE_RESTORE	RM_L3MAIN1_SPARE_HDMI_CONTEXT	RM_L3INIT_OCP2SCP1_CONTEXT
CM_SSC_MODFREQDIV_DPLL_CORE_RESTORE	RM_L3MAIN1_SPARE_ICM_CONTEXT	RM_L3INIT_OCP2SCP3_CONTEXT
CM_MPU_CLKSTCTRL_RESTORE	RM_L3MAIN1_SPARE_IVA2_CONTEXT	PM_L3INIT_USB_OTG_SS1_WKDEP
CM_RTC_CLKSTCTRL	RM_L3MAIN1_SPARE_SATA2_CONTEXT	RM_L3INIT_USB_OTG_SS1_CONTEXT
CM_RTC_RTCSS_CLKCTRL	RM_L3MAIN1_SPARE_UNKNOWN4_CONTEXT	RM_L4PER2_PRUSS1_CONTEXT
CM_CAM_VIP2_CLKCTRL	RM_L3MAIN1_SPARE_UNKNOWN5_CONTEXT	RM_L4PER2_PRUSS2_CONTEXT
CM_CAM_VIP3_CLKCTRL	RM_L3MAIN1_SPARE_UNKNOWN6_CONTEXT	PM_L4PER_DCC6_WKDEPPM_L4PER_DCC6_WKDEP
CM_CLKSEL_USB_60MHZ	RM_L3MAIN1_SPARE_VIDEOPLL1_CONTEXT	PM_L4PER_DCC7_WKDEPPM_L4PER_DCC7_WKDEP
CM_CLKMODE_DPLL_USB	RM_L3MAIN1_SPARE_VIDEOPLL2_CONTEXT	
CM_IDLEST_DPLL_USB	RM_L3MAIN1_SPARE_VIDEOPLL3_CONTEXT	PM_L4PER_GPIO5_WKDEP
CM_AUTOIDLE_DPLL_USB	RM_IPU2_RSTCTRL	RM_L4PER_GPIO5_CONTEXT
CM_CLKSEL_DPLL_USB	RM_IPU2_RSTST	PM_L4PER_GPIO6_WKDEP
CM_DIV_M2_DPLL_USB	RM_IPU2_IPU2_CONTEXT	RM_L4PER_GPIO6_CONTEXT
CM_SSC_DELTAMSTEP_DPLL_USB	RM_DMA_DMA_SYSTEM_CONTEXT	RM_L4PER2_PWMSS2_CONTEXT
CM_SSC_MODFREQDIV_DPLL_USB	RM_EMIF_DMM_CONTEXT	RM_L4PER2_PWMSS3_CONTEXT
CM_CLKDCOLDO_DPLL_USB	RM_L4CFG_SAR_ROM_CONTEXT	PM_L4PER_I2C3_WKDEP
CM_CLKMODE_DPLL_PCIE_REF	RM_L4CFG_OCP2SCP2_CONTEXT	RM_L4PER_I2C3_CONTEXT
CM_IDLEST_DPLL_PCIE_REF	RM_L4CFG_MAILBOX3_CONTEXT	PM_L4PER_I2C4_WKDEP
CM_AUTOIDLE_DPLL_PCIE_REF	RM_L4CFG_MAILBOX4_CONTEXT	RM_L4PER_I2C4_CONTEXT
CM_CLKSEL_DPLL_PCIE_REF	RM_L4CFG_MAILBOX5_CONTEXT	PM_L4PER_GPIO7_WKDEP
CM_DIV_M2_DPLL_PCIE_REF	RM_L4CFG_MAILBOX6_CONTEXT	RM_L4PER_GPIO7_CONTEXT
CM_SSC_DELTAMSTEP_DPLL_PCIE_REF	RM_L4CFG_MAILBOX7_CONTEXT	PM_L4PER_GPIO8_WKDEP
CM_SSC_MODFREQDIV_DPLL_PCIE_REF	RM_L4CFG_MAILBOX8_CONTEXT	RM_L4PER_GPIO8_CONTEXT
CM_CLKMODE_APLL_PCIE	RM_L4CFG_MAILBOX9_CONTEXT	PM_L4PER_MMC3_WKDEP
CM_IDLEST_APLL_PCIE	RM_L4CFG_MAILBOX10_CONTEXT	RM_L4PER_MMC3_CONTEXT
CM_DIV_M2_APLL_PCIE	RM_L4CFG_MAILBOX11_CONTEXT	PM_L4PER2_MCASP3_WKDEP
CM_CLKVCOLDO_APLL_PCIE	RM_L4CFG_MAILBOX12_CONTEXT	RM_L4PER2_MCASP3_CONTEXT
CM_COREAON_SMARTREFLEX_MPU_CLKCTRL	RM_L4CFG_MAILBOX13_CONTEXT	PM_L4PER_UART5_WKDEP
CM_COREAON_SMARTREFLEX_CORE_CLKCTRL	RM_L4CFG_SPARE_SMARTREFLEX_RTC_CONTEXT	RM_L4PER_UART5_CONTEXT
CM_COREAON_USB_PHY1_CORE_CLKCTRL	RM_L4CFG_SPARE_SMARTREFLEX_SDRAM_CONTEXT	PM_L4PER2_MCASP5_WKDEP
CM_COREAON_IO_SRCOMP_CLKCTRL	RM_L4CFG_SPARE_SMARTREFLEX_WKUP_CONTEXT	RM_L4PER2_MCASP5_CONTEXT
CM_COREAON_SMARTREFLEX_GPU_CLKCTRL	RM_L4CFG_IO_DELAY_BLOCK_CONTEXT	PM_L4PER2_MCASP6_WKDEP
CM_COREAON_SMARTREFLEX_DSPEVE_CLKCTRL	PRM_CLKREQCTRL	RM_L4PER2_MCASP6_CONTEXT
CM_COREAON_SMARTREFLEX_IVAHD_CLKCTRL	PRM_VOLTCTRL	PM_L4PER2_MCASP7_WKDEP

Table 3-288. Not Supported Functionality (Registers and Bits) (continued)

Register/Field/Value	Register/Field/Value	Register/Field/Value
CM_COREAON_USB_PHY2_CORE_CLKCTRL	PRM_PWRREQCTRL	RM_L4PER2_MCASP7_CONTEXT
CM_COREAON_USB_PHY3_CORE_CLKCTRL	PRM_VOLTSETUP_WARMRESET	PM_L4PER2_MCASP8_WKDEP
CM_COREAON_DUMMY_MODULE3_CLKCTRL	PRM_VOLTSETUP_CORE_OFF	RM_L4PER2_MCASP8_CONTEXT
CM_L3MAIN1_MMU_PCIESS_CLKCTRL	PRM_VOLTSETUP_MPU_OFF	PM_L4PER2_MCASP4_WKDEP
CM_L3MAIN1_OCMC_RAM3_CLKCTRL	PRM_VOLTSETUP_MM_OFF	RM_L4PER2_MCASP4_CONTEXT
CM_L3MAIN1_OCMC_ROM_CLKCTRL	PRM_VOLTSETUP_CORE_RET_SLEEP	RM_L4SEC_AES1_CONTEXT
CM_L3MAIN1_VCP1_CLKCTRL	PRM_VOLTSETUP_MPU_RET_SLEEP	RM_L4SEC_AES2_CONTEXT
CM_L3MAIN1_VCP2_CLKCTRL	PRM_VOLTSETUP_MM_RET_SLEEP	RM_L4SEC_DES3DES_CONTEXT
CM_L3MAIN1_SPARE_CME_CLKCTRL	PRM_VP_CORE_CONFIG	RM_L4SEC_FPKA_CONTEXT
CM_L3MAIN1_SPARE_HDMI_CLKCTRL	PRM_VP_CORE_STATUS	RM_L4SEC_RNG_CONTEXT
CM_L3MAIN1_SPARE_ICM_CLKCTRL	PRM_VP_CORE_VLIMITTO	RM_L4SEC_SHA2MD51_CONTEXT
CM_L3MAIN1_SPARE_IVA2_CLKCTRL	PRM_VP_CORE_VOLTAGE	PM_L4PER2_UART7_WKDEP
CM_L3MAIN1_SPARE_SATA2_CLKCTRL	PRM_VP_CORE_VSTEPMAX	RM_L4PER2_UART7_CONTEXT
CM_L3MAIN1_SPARE_UNKNOWN4_CLKCTRL	PRM_VP_CORE_VSTEPMIN	RM_L4SEC_DMA_CRYPTO_CONTEXT
CM_L3MAIN1_SPARE_UNKNOWN5_CLKCTRL	PRM_VP_MPU_CONFIG	PM_L4PER2_UART8_WKDEP
CM_L3MAIN1_SPARE_UNKNOWN6_CLKCTRL	PRM_VP_MPU_STATUS	RM_L4PER2_UART8_CONTEXT
CM_L3MAIN1_SPARE_VIDEOPLL1_CLKCTRL	PRM_VP_MPU_VLIMITTO	PM_L4PER2_UART9_WKDEP
CM_L3MAIN1_SPARE_VIDEOPLL2_CLKCTRL	PRM_VP_MPU_VOLTAGE	RM_L4PER2_UART9_CONTEXT
CM_L3MAIN1_SPARE_VIDEOPLL3_CLKCTRL	PRM_VP_MPU_VSTEPMAX	RM_L4SEC_SHA2MD52_CONTEXT
CM_DMA_CLKSTCTRL	PRM_VP_MPU_VSTEPMIN	PM_MPU_PWRSTCTRL
CM_DMA_STATICDEP	PRM_VP_MM_CONFIG	PM_MPU_PWRSTST
CM_DMA_DYNAMICDEP	PRM_VP_MM_STATUS	RM_MPU_MPU_CONTEXT
CM_DMA_DMA_SYSTEM_CLKCTRL	PRM_VP_MM_VLIMITTO	PRM_IRQSTATUS_MPU_2
CM_EMIF_DMM_CLKCTRL	PRM_VP_MM_VOLTAGE	PRM_IRQENABLE_MPU_2
CM_L4CFG_SAR_ROM_CLKCTRL	PRM_VP_MM_VSTEPMAX	PRM_IRQSTATUS_IPU2
CM_L4CFG_OCP2SCP2_CLKCTRL	PRM_VP_MM_VSTEPMIN	PRM_IRQENABLE_IPU2
CM_L4CFG_MAILBOX3_CLKCTRL	PRM_VC_SMPS_CORE_CONFIG	PRM_IRQENABLE_EVE2
CM_L4CFG_MAILBOX4_CLKCTRL	PRM_VC_SMPS_MM_CONFIG	PRM_IRQENABLE_EVE3
CM_L4CFG_MAILBOX5_CLKCTRL	PRM_VC_SMPS_MPU_CONFIG	PRM_IRQENABLE_EVE4
CM_L4CFG_MAILBOX6_CLKCTRL	PRM_VC_VAL_CMD_VDD_CORE_L	PRM_IRQSTATUS_EVE2
CM_L4CFG_MAILBOX7_CLKCTRL	PRM_VC_VAL_CMD_VDD_MM_L	PRM_IRQSTATUS_EVE3
CM_L4CFG_MAILBOX8_CLKCTRL	PRM_VC_VAL_CMD_VDD_MPU_L	PRM_IRQSTATUS_EVE4
CM_L4CFG_MAILBOX9_CLKCTRL	PRM_VC_VAL_BYPASS	PM_RTC_RTCSS_WKDEP
CM_L4CFG_MAILBOX10_CLKCTRL	PRM_VC_CORE_ERRST	RM_RTC_RTCSS_CONTEXT
CM_L4CFG_MAILBOX11_CLKCTRL	PRM_VC_MM_ERRST	CM_WKUPAON_WD_TIMER1_CLKCTRL
CM_L4CFG_MAILBOX12_CLKCTRL	PRM_VC_MPU_ERRST	CM_WKUPAON_WD_TIMER2_CLKCTRL
CM_L4CFG_MAILBOX13_CLKCTRL	PRM_VC_BYPASS_ERRST	CM_WKUPAON_SAR_RAM_CLKCTRL
CM_L4CFG_SPARE_SMARTREFLEX_RTC_CLKCTRL	PRM_VC_CFG_I2C_MODE	CM_WKUPAON_KBD_CLKCTRL
CM_L4CFG_SPARE_SMARTREFLEX_SDRAM_CLKCTRL	PRM_VC_CFG_I2C_CLK	CM_WKUPAON_UART10_CLKCTRL

Table 3-288. Not Supported Functionality (Registers and Bits) (continued)

Register/Field/Value	Register/Field/Value	Register/Field/Value
CM_L4CFG_SPARE_SMARTREFLEX_WKUP_CLKCTRL	PRM_SRAM_WKUP_SETUP	CM_WKUPAON_SCRM_CLKCTRL
CM_L4CFG_IO_DELAY_BLOCK_CLKCTRL	CM_GMAC_GMAC_CLKCTRL[27:25] CLKSEL_RFT=0x0	CM_WKUPAON_IO_SRCOMP_CLKCTRL
CM_DSS_BB2D_CLKCTRL	CM_GMAC_GMAC_CLKCTRL[27:25] CLKSEL_RFT=0x1	CM_WKUPAON_ADC_CLKCTRL
CM_GPU_CLKSTCTRL	CM_GMAC_GMAC_CLKCTRL[27:25] CLKSEL_RFT=0x2	CM_WKUPAON_SPARE_SAFETY1_CLKCTRL
CM_GPU_STATICDEP	CM_GMAC_GMAC_CLKCTRL[27:25] CLKSEL_RFT=0x3	PM_WKUPAON_WD_TIMER1_WKDEP
CM_GPU_DYNAMICDEP	CM_GMAC_GMAC_CLKCTRL[24] CLKSEL_REF=0x1	RM_WKUPAON_WD_TIMER1_CONTEXT
CM_GPU_GPU_CLKCTRL	CM_L4PER_CLKSTCTRL[26] CLKACTIVITY_UART5_GFCLK	PM_WKUPAON_WD_TIMER2_WKDEP
CM_IVA_CLKSTCTRL	CM_L4PER_CLKSTCTRL[22] CLKACTIVITY_MMC3_GFCLK	RM_WKUPAON_WD_TIMER2_CONTEXT
CM_IVA_STATICDEP	CM_L4PER_CLKSTCTRL[18] CLKACTIVITY_UART4_GFCLK	RM_WKUPAON_SAR_RAM_CONTEXT
CM_IVA_DYNAMICDEP	CM_L4PER_CLKSTCTRL[14] CLKACTIVITY_DCC5_GFCLK	PM_WKUPAON_KBD_WKDEP
CM_IVA_IVA_CLKCTRL	CM_L4PER_CLKSTCTRL[10] CLKACTIVITY_DCC7_GFCLK	RM_WKUPAON_KBD_CONTEXT
CM_IVA_SL2_CLKCTRL	CM_L4PER_CLKSTCTRL[9] CLKACTIVITY_DCC6_GFCLK	PM_WKUPAON_UART10_WKDEP
CM_L3INIT_MMC1_CLKCTRL	CM_L4PER_DYNAMICDEP[14] L4SEC_DYNDEP	RM_WKUPAON_UART10_CONTEXT
CM_L3INIT_MMC2_CLKCTRL	CM_L4PER_DCC6_CLKCTRL[27:24] CLKSEL	PM_WKUPAON_ADC_WKDEP
CM_L3INIT_USB_OTG_SS2_CLKCTRL	CM_L4PER_DCC7_CLKCTRL[27:24] CLKSEL	RM_WKUPAON_ADC_CONTEXT
CM_L3INIT_USB_OTG_SS3_CLKCTRL	CM_L4PER_TIMER2_CLKCTRL[27:24] CLKSEL=0x5	RM_WKUPAON_SPARE_SAFETY1_CONTEXT
CM_L3INIT_USB_OTG_SS4_CLKCTRL	CM_L4PER_TIMER2_CLKCTRL[27:24] CLKSEL=0x6	CM_L4PER2_CLKSTCTRL[31] CLKACTIVITY_MCASP8_AUX_GFCLK
CM_L3INIT_MLB_SS_CLKCTRL	CM_L4PER_TIMER2_CLKCTRL[27:24] CLKSEL=0x8	CM_L4PER2_CLKSTCTRL[30] CLKACTIVITY_MCASP8_AHCLKX
CM_L3INIT_SATA_CLKCTRL	CM_L4PER_TIMER2_CLKCTRL[27:24] CLKSEL=0x9	CM_L4PER2_CLKSTCTRL[29] CLKACTIVITY_MCASP7_AUX_GFCLK
CM_PCIE_CLKSTCTRL	CM_L4PER_TIMER2_CLKCTRL[27:24] CLKSEL=0xA	CM_L4PER2_CLKSTCTRL[27] CLKACTIVITY_MCASP6_AUX_GFCLK
CM_PCIE_STATICDEP	CM_L4PER_TIMER3_CLKCTRL[27:24] CLKSEL=0x5	CM_L4PER2_CLKSTCTRL[21] CLKACTIVITY_MCASP3_AUX_GFCLK
CM_PCIE_PCIESS1_CLKCTRL	CM_L4PER_TIMER3_CLKCTRL[27:24] CLKSEL=0x6	CM_L4PER2_CLKSTCTRL[20] CLKACTIVITY_MCASP3_AHCLKX
CM_PCIE_PCIESS2_CLKCTRL	CM_L4PER_TIMER3_CLKCTRL[27:24] CLKSEL=0x8	CM_L4PER2_CLKSTCTRL[19] CLKACTIVITY_MCASP2_AUX_GFCLK
CM_L3INIT_OCP2SCP1_CLKCTRL	CM_L4PER_TIMER3_CLKCTRL[27:24] CLKSEL=0x9	CM_L4PER2_CLKSTCTRL[18] CLKACTIVITY_MCASP2_AHCLKR
CM_L3INIT_OCP2SCP3_CLKCTRL	CM_L4PER_TIMER3_CLKCTRL[27:24] CLKSEL=0xA	CM_L4PER2_CLKSTCTRL[17] CLKACTIVITY_MCASP2_AHCLKX
CM_L3INIT_USB_OTG_SS1_CLKCTRL	CM_L4PER_TIMER4_CLKCTRL[27:24] CLKSEL=0x5	CM_L4PER2_CLKSTCTRL[14] CLKACTIVITY_ICSS_IEP_CLK
CM_L4PER2_PRUSS1_CLKCTRL	CM_L4PER_TIMER4_CLKCTRL[27:24] CLKSEL=0x6	CM_L4PER2_CLKSTCTRL[11] CLKACTIVITY_UART9_GFCLK
CM_L4PER2_PRUSS2_CLKCTRL	CM_L4PER_TIMER4_CLKCTRL[27:24] CLKSEL=0x8	CM_L4PER2_CLKSTCTRL[10] CLKACTIVITY_UART8_GFCLK

Table 3-288. Not Supported Functionality (Registers and Bits) (continued)

Register/Field/Value	Register/Field/Value	Register/Field/Value
CM_CAM_CLKSTCTRL[9] CLKACTIVITY_VIP2_GCLK	CM_L4PER_TIMER4_CLKCTRL[27:24] CLKSEL=0x9	CM_L4PER2_CLKSTCTRL[9] CLKACTIVITY_UART7_GFCLK
CM_CAM_CLKSTCTRL[10] CLKACTIVITY_VIP3_GCLK	CM_L4PER_TIMER4_CLKCTRL[27:24] CLKSEL=0xA	CM_L4PER2_CLKSTCTRL[8] CLKACTIVITY_ICSS_CLK
CM_CAM_STATICDEP[2] IVA_STATDEP	CM_L4PER_DCC5_CLKCTRL[27:24] CLKSEL	CM_L4PER3_CLKSTCTRL[12] CLKACTIVITY_DCC4_GFCLK
CM_CAM_STATICDEP[20] EVE2_STATDEP	CM_L4PER3_DCC1_CLKCTRL[27:24] CLKSEL	CM_L4PER3_CLKSTCTRL[11] CLKACTIVITY_DCC3_GFCLK
CM_CAM_STATICDEP[21] EVE3_STATDEP	CM_L4PER3_DCC2_CLKCTRL[27:24] CLKSEL	CM_L4PER3_CLKSTCTRL[10] CLKACTIVITY_DCC2_GFCLK
CM_CAM_STATICDEP[22] EVE4_STATDEP	CM_L4PER3_DCC3_CLKCTRL[27:24] CLKSEL	CM_L4PER3_CLKSTCTRL[9] CLKACTIVITY_DCC1_GFCLK
CM_CLKMODE_DPLL_PER[15]DPLL_SSC_TYPE	CM_L4PER3_DCC4_CLKCTRL[27:24] CLKSEL	PM_CAM_VIP1_WKDEP[9] WKUPDEP_VIP1_EVE4
CM_CLKMODE_DPLL_PER[14]DPLL_SSC_DOWNSPREAD	CM_L4PER2_ADC_CLKCTRL[31:28] CLKSEL_AHCLKR	PM_CAM_VIP1_WKDEP[8] WKUPDEP_VIP1_EVE3
CM_CLKMODE_DPLL_PER[13]DPLL_SSC_ACK	CM_L4PER2_ADC_CLKCTRL[27:24] CLKSEL_AHCLKX	PM_CAM_VIP1_WKDEP[7] WKUPDEP_VIP1_EVE2
CM_CLKMODE_DPLL_PER[12]DPLL_SSC_EN	CM_L4PER2_ADC_CLKCTRL[23:22] CLKSEL_AUX_CLK	PM_CAM_VIP1_WKDEP[1] WKUPDEP_VIP1_IPU2
CM_COREAON_CLKSTCTRL[14]CLKACTIVITY_COREAON_IO_SRCOMP_GFCLK	PM_IPU_MCASP1_WKDEP[13] WKUPDEP_MCASP1_DMA_SDMA	PM_CAM_VIP1_WKDEP[0] WKUPDEP_VIP1_MPU
CM_L3MAIN1_DYNAMICDEP[31]EVE4_DYNDEP	PM_IPU_MCASP1_WKDEP[9] WKUPDEP_MCASP1_IRQ_EVE4	CM_CLKSEL_SYS[2:0] SYS_CLKSEL=0x1
CM_L3MAIN1_DYNAMICDEP[30]EVE3_DYNDEP	PM_IPU_MCASP1_WKDEP[8]WKUPDEP_MCASP1_IRQ_EVE3	CM_CLKSEL_SYS[2:0] SYS_CLKSEL=0x3
CM_L3MAIN1_DYNAMICDEP[29]EVE2_DYNDEP	PM_IPU_MCASP1_WKDEP[7] WKUPDEP_MCASP1_IRQ_EVE2	CM_CLKSEL_SYS[2:0] SYS_CLKSEL=0x5
CM_L3MAIN1_DYNAMICDEP[21] PCIE_DYNDEP	PM_IPU_MCASP1_WKDEP[1] WKUPDEP_MCASP1_IRQ_IPU2	CM_CLKSEL_SYS[2:0] SYS_CLKSEL=0x7
CM_L3MAIN1_DYNAMICDEP[14] L4SEC_DYNDEP	PM_IPU_MCASP1_WKDEP[0] WKUPDEP_MCASP1_IRQ_MPU	PM_L3MAIN1_OCMC_RAM1_WKDEP[9] WKUPDEP_OCMC_RAM1_EVE4
CM_L3MAIN1_DYNAMICDEP[10] GPU_DYNDEP	PM_IPU_TIMER5_WKDEP[9] WKUPDEP_TIMER5_EVE4	PM_L3MAIN1_OCMC_RAM1_WKDEP[8] WKUPDEP_OCMC_RAM1_EVE3
CM_L3MAIN1_DYNAMICDEP[2] IVA_DYNDEP	PM_IPU_TIMER5_WKDEP[8] WKUPDEP_TIMER5_EVE3	PM_L3MAIN1_OCMC_RAM1_WKDEP[7] WKUPDEP_OCMC_RAM1_EVE2
CM_L3MAIN1_DYNAMICDEP[0] IPU2_DYNDEP	PM_IPU_TIMER5_WKDEP[7] WKUPDEP_TIMER5_EVE2	PM_L3MAIN1_OCMC_RAM1_WKDEP[1] WKUPDEP_OCMC_RAM1_IPU2
CM_CRC_CRC_CLKCTRL[27:26] CLKSEL_SOURCE2	PM_IPU_TIMER5_WKDEP[1] WKUPDEP_TIMER5_IPU2	PM_L3MAIN1_OCMC_RAM1_WKDEP[0] WKUPDEP_OCMC_RAM1_MPU
CM_CRC_CRC_CLKCTRL[25:24] CLKSEL_SOURCE1	PM_IPU_TIMER5_WKDEP[0] WKUPDEP_TIMER5_MPU	PM_L3MAIN1_TESOC_WKDEP[9] WKUPDEP_TESOC_EVE4
CM_CRC_CLKSTCTRL[9] CLKACTIVITY_CRC_GFCLK	PM_IPU_TIMER6_WKDEP[9] WKUPDEP_TIMER6_EVE4	PM_L3MAIN1_TESOC_WKDEP[8] WKUPDEP_TESOC_EVE3
CM_L4CFG_DYNAMICDEP[19] MPU_DYNDEP	PM_IPU_TIMER6_WKDEP[8] WKUPDEP_TIMER6_EVE3	PM_L3MAIN1_TESOC_WKDEP[7] WKUPDEP_TESOC_EVE2
CM_L4CFG_DYNAMICDEP[11] SDMA_DYNDEP	PM_IPU_TIMER6_WKDEP[7] WKUPDEP_TIMER6_EVE2	PM_L3MAIN1_TESOC_WKDEP[1] WKUPDEP_TESOC_IPU2
CM_DSS_CLKSTCTRL[18] CLKACTIVITY_HDMI_PHY_GFCLK	PM_IPU_TIMER6_WKDEP[1] WKUPDEP_TIMER6_IPU2	PM_L3MAIN1_TESOC_WKDEP[0] WKUPDEP_TESOC_MPU
CM_DSS_CLKSTCTRL[17] CLKACTIVITY_HDMI_CEC_GFCLK	PM_IPU_TIMER6_WKDEP[0] WKUPDEP_TIMER6_MPU	PM_L3MAIN1_TPCC_WKDEP[9] WKUPDEP_OCMC_RAM1_EVE4
CM_DSS_CLKSTCTRL[14] CLKACTIVITY_SDVENC_GFCLK	PM_IPU_TIMER7_WKDEP[9] WKUPDEP_TIMER7_EVE4	PM_L3MAIN1_TPCC_WKDEP[8] WKUPDEP_OCMC_RAM1_EVE3

Table 3-288. Not Supported Functionality (Registers and Bits) (continued)

Register/Field/Value	Register/Field/Value	Register/Field/Value
CM_DSS_CLKSTCTRL[13] CLKACTIVITY_BB2D_GFCLK	PM_IPU_TIMER7_WKDEP[8] WKUPDEP_TIMER7_EVE3	PM_L3MAIN1_TPCC_WKDEP[7] WKUPDEP_OCMC_RAM1_EVE2
CM_DSS_CLKSTCTRL[12] CLKACTIVITY_VIDEO2_DPLL_CLK	PM_IPU_TIMER7_WKDEP[7] WKUPDEP_TIMER7_EVE2	PM_L3MAIN1_TPCC_WKDEP[1] WKUPDEP_OCMC_RAM1_IPU2
CM_DSS_CLKSTCTRL[11] CLKACTIVITY_HDMI_DPLL_CLK	PM_IPU_TIMER7_WKDEP[1] WKUPDEP_TIMER7_IPU2	PM_L3MAIN1_TPCC_WKDEP[0] WKUPDEP_OCMC_RAM1_MPU
CM_DSS_CLKSTCTRL[10] CLKACTIVITY_VIDEO1_DPLL_CLK	PM_IPU_TIMER7_WKDEP[0] WKUPDEP_TIMER7_MPU	PM_L3MAIN1_TPTC1_WKDEP[9] WKUPDEP_OCMC_RAM1_EVE4
CM_DSS_DSS_CLKCTRL[13] OPTFCLKEN_VIDEO2_CLK	PM_IPU_TIMER8_WKDEP[9] WKUPDEP_TIMER8_EVE4	PM_L3MAIN1_TPTC1_WKDEP[8] WKUPDEP_OCMC_RAM1_EVE3
CM_DSS_DSS_CLKCTRL[12] OPTFCLKEN_VIDEO1_CLK	PM_IPU_TIMER8_WKDEP[8] WKUPDEP_TIMER8_EVE3	PM_L3MAIN1_TPTC1_WKDEP[7] WKUPDEP_OCMC_RAM1_EVE2
CM_DSS_DSS_CLKCTRL[10] OPTFCLKEN_HDMI_CLK	PM_IPU_TIMER8_WKDEP[7] WKUPDEP_TIMER8_EVE2	PM_L3MAIN1_TPTC1_WKDEP[1] WKUPDEP_OCMC_RAM1_IPU2
CM_L3INIT_CLKSTCTRL[24] CLKACTIVITY_SATA_REF_GFCLK	PM_IPU_TIMER8_WKDEP[1] WKUPDEP_TIMER8_IPU2	PM_L3MAIN1_TPTC1_WKDEP[0] WKUPDEP_OCMC_RAM1_MPU
CM_L3INIT_CLKSTCTRL[22] CLKACTIVITY_L3INIT_960M_GFCLK	PM_IPU_TIMER8_WKDEP[0] WKUPDEP_TIMER8_MPU	PM_L3MAIN1_TPTC2_WKDEP[9] WKUPDEP_OCMC_RAM1_EVE4
CM_L3INIT_CLKSTCTRL[21] CLKACTIVITY_L3INIT_480M_GFCLK	RM_PCIESS_RSTCTRL	PM_L3MAIN1_TPTC2_WKDEP[8] WKUPDEP_OCMC_RAM1_EVE3
CM_L3INIT_CLKSTCTRL[20] CLKACTIVITY_USB_OTG_SS_REF_CLK	RM_PCIESS_RSTST	PM_L3MAIN1_TPTC2_WKDEP[7] WKUPDEP_OCMC_RAM1_EVE2
CM_L3INIT_CLKSTCTRL[19] CLKACTIVITY_MLB_SYS_L3_GFCLK	PM_L4PER_DCC6_WKDEP	PM_L3MAIN1_TPTC2_WKDEP[1] WKUPDEP_OCMC_RAM1_IPU2
CM_L3INIT_CLKSTCTRL[18] CLKACTIVITY_MLB_SPB_L4_GICLK	PM_L4PER_DCC7_WKDEP	PM_L3MAIN1_TPTC2_WKDEP[0] WKUPDEP_OCMC_RAM1_MPU
CM_L3INIT_CLKSTCTRL[17] CLKACTIVITY_MLB_SHB_L3_GICLK	PM_L4PER_DCC1_WKDEP	PM_L4PER_GPIO2_WKDEP[19] WKUPDEP_GPIO2_IRQ2_EVE4
CM_L3INIT_CLKSTCTRL[16] CLKACTIVITY_MMC2_GFCLK	PM_L4PER_DCC2_WKDEP	PM_L4PER_GPIO2_WKDEP[18] WKUPDEP_GPIO2_IRQ2_EVE3
CM_L3INIT_CLKSTCTRL[15] CLKACTIVITY_MMC1_GFCLK	PM_L4PER_DCC3_WKDEP	PM_L4PER_GPIO2_WKDEP[17] WKUPDEP_GPIO2_IRQ2_EVE2
CM_L3INIT_CLKSTCTRL[14] CLKACTIVITY_HSI_GFCLK	PM_L4PER_DCC4_WKDEP	PM_L4PER_GPIO2_WKDEP[11] WKUPDEP_GPIO2_IRQ2_IPU2
CM_L3INIT_CLKSTCTRL[13] CLKACTIVITY_USB_DPLL_HS_CLK	PM_L4PER_DCC5_WKDEP	PM_L4PER_GPIO2_WKDEP[10] WKUPDEP_GPIO2_IRQ2_MPU
CM_L3INIT_CLKSTCTRL[12] CLKACTIVITY_USB_DPLL_CLK	PM_L4PER_TIMER2_WKDEP[9] WKUPDEP_TIMER2_EVE4	PM_L4PER_GPIO2_WKDEP[9] WKUPDEP_GPIO2_IRQ1_EVE4
CM_L3INIT_CLKSTCTRL[10] CLKACTIVITY_L3INIT_USB_LFPS_TX_GFCLK	PM_L4PER_TIMER2_WKDEP[8] WKUPDEP_TIMER2_EVE3	PM_L4PER_GPIO2_WKDEP[8] WKUPDEP_GPIO2_IRQ1_EVE3
CM_L3INIT_STATICDEP[14] L4SEC_STATDEP	PM_L4PER_TIMER2_WKDEP[7] WKUPDEP_TIMER2_EVE2	PM_L4PER_GPIO2_WKDEP[7] WKUPDEP_GPIO2_IRQ1_EVE2
CM_L3INIT_STATICDEP[2] IVA_STATDEP	PM_L4PER_TIMER2_WKDEP[1] WKUPDEP_TIMER2_IPU2	PM_L4PER_GPIO2_WKDEP[1] WKUPDEP_GPIO2_IRQ1_IPU2
PRM_RSTST[16] TSHUT_IVA_RST	PM_L4PER_TIMER2_WKDEP[0] WKUPDEP_TIMER2_MPU	PM_L4PER_GPIO2_WKDEP[0] WKUPDEP_GPIO2_IRQ1_MPU
PRM_RSTST[14] LLI_RST	PM_L4PER_TIMER3_WKDEP[9] WKUPDEP_TIMER3_EVE4	PM_L4PER_GPIO3_WKDEP[19] WKUPDEP_GPIO3_IRQ2_EVE4
PRM_RSTST[12] TSHUT_MM_RST	PM_L4PER_TIMER3_WKDEP[8] WKUPDEP_TIMER3_EVE3	PM_L4PER_GPIO3_WKDEP[18] WKUPDEP_GPIO3_IRQ2_EVE3
PRM_RSTST[11] TSHUT_MPU_RST	PM_L4PER_TIMER3_WKDEP[7] WKUPDEP_TIMER3_EVE2	PM_L4PER_GPIO3_WKDEP[17] WKUPDEP_GPIO3_IRQ2_EVE2
PRM_RSTST[10] C2C_RST	PM_L4PER_TIMER3_WKDEP[1] WKUPDEP_TIMER3_IPU2	PM_L4PER_GPIO3_WKDEP[11] WKUPDEP_GPIO3_IRQ2_IPU2

Table 3-288. Not Supported Functionality (Registers and Bits) (continued)

Register/Field/Value	Register/Field/Value	Register/Field/Value
PRM_RSTST[8] VDD_CORE_VOLT_MGR_RST	PM_L4PER_TIMER3_WKDEP[0] WKUPDEP_TIMER3_MPU	PM_L4PER_GPIO3_WKDEP[10] WKUPDEP_GPIO3_IRQ2_MPU
PRM_RSTST[7] VDD_MM_VOLT_MGR_RST	PM_L4PER_TIMER4_WKDEP[9] WKUPDEP_TIMER4_EVE4	PM_L4PER_GPIO3_WKDEP[9] WKUPDEP_GPIO3_IRQ1_EVE4
PRM_RSTST[6] VDD_MPU_VOLT_MGR_RST	PM_L4PER_TIMER4_WKDEP[8] WKUPDEP_TIMER4_EVE3	PM_L4PER_GPIO3_WKDEP[8] WKUPDEP_GPIO3_IRQ1_EVE3
PM_DSS_DSS_WKDEP[29] WKUPDEP_DSI1_B_EVE4	PM_L4PER_TIMER4_WKDEP[7] WKUPDEP_TIMER4_EVE2	PM_L4PER_GPIO3_WKDEP[7] WKUPDEP_GPIO3_IRQ1_EVE2
PM_DSS_DSS_WKDEP[28] WKUPDEP_DSI1_B_EVE3	PM_L4PER_TIMER4_WKDEP[1] WKUPDEP_TIMER4_IPU2	PM_L4PER_GPIO3_WKDEP[1] WKUPDEP_GPIO3_IRQ1_IPU2
PM_DSS_DSS_WKDEP[27] WKUPDEP_DSI1_B_EVE2	PM_L4PER_TIMER4_WKDEP[0] WKUPDEP_TIMER4_MPU	PM_L4PER_GPIO3_WKDEP[0] WKUPDEP_GPIO3_IRQ1_MPU
PM_DSS_DSS_WKDEP[26] WKUPDEP_DSI1_B_EVE1	PM_L4PER_MCSPI1_WKDEP[9] WKUPDEP_MCSPI1_EVE4	PM_L4PER_GPIO4_WKDEP[19] WKUPDEP_GPIO4_IRQ2_EVE4
PM_DSS_DSS_WKDEP[25] WKUPDEP_DSI1_B_DSP2	PM_L4PER_MCSPI1_WKDEP[8] WKUPDEP_MCSPI1_EVE3	PM_L4PER_GPIO4_WKDEP[18] WKUPDEP_GPIO4_IRQ2_EVE3
PM_DSS_DSS_WKDEP[24] WKUPDEP_DSI1_B_IPU1	PM_L4PER_MCSPI1_WKDEP[7] WKUPDEP_MCSPI1_EVE2	PM_L4PER_GPIO4_WKDEP[17] WKUPDEP_GPIO4_IRQ2_EVE2
PM_DSS_DSS_WKDEP[23] WKUPDEP_DSI1_B_SDMA	PM_L4PER_MCSPI1_WKDEP[3] WKUPDEP_MCSPI1_SDMA	PM_L4PER_GPIO4_WKDEP[11] WKUPDEP_GPIO4_IRQ2_IPU2
PM_DSS_DSS_WKDEP[22] WKUPDEP_DSI1_B_DSP1	PM_L4PER_MCSPI1_WKDEP[1] WKUPDEP_MCSPI1_IPU2	PM_L4PER_GPIO4_WKDEP[10] WKUPDEP_GPIO4_IRQ2_MPU
PM_DSS_DSS_WKDEP[21] WKUPDEP_DSI1_B_IPU2	PM_L4PER_MCSPI1_WKDEP[0] WKUPDEP_MCSPI1_MPU	PM_L4PER_GPIO4_WKDEP[9] WKUPDEP_GPIO4_IRQ1_EVE4
PM_DSS_DSS_WKDEP[20] WKUPDEP_DSI1_B_MPU	PM_L4PER_MCSPI2_WKDEP[9] WKUPDEP_MCSPI2_EVE4	PM_L4PER_GPIO4_WKDEP[8] WKUPDEP_GPIO4_IRQ1_EVE3
PM_DSS_DSS_WKDEP[19] WKUPDEP_DSI1_A_EVE4	PM_L4PER_MCSPI2_WKDEP[8] WKUPDEP_MCSPI2_EVE3	PM_L4PER_GPIO4_WKDEP[7] WKUPDEP_GPIO4_IRQ1_EVE2
PM_DSS_DSS_WKDEP[18] WKUPDEP_DSI1_A_EVE3	PM_L4PER_MCSPI2_WKDEP[7] WKUPDEP_MCSPI2_EVE2	PM_L4PER_GPIO4_WKDEP[1] WKUPDEP_GPIO4_IRQ1_IPU2
PM_DSS_DSS_WKDEP[17] WKUPDEP_DSI1_A_EVE2	PM_L4PER_MCSPI2_WKDEP[3] WKUPDEP_MCSPI2_SDMA	PM_L4PER_GPIO4_WKDEP[0] WKUPDEP_GPIO4_IRQ1_MPU
PM_DSS_DSS_WKDEP[16] WKUPDEP_DSI1_A_EVE1	PM_L4PER_MCSPI2_WKDEP[1] WKUPDEP_MCSPI2_IPU2	PM_L4PER2_ADC_WKDEP[13] WKUPDEP_ADC_DMA_SDMA
PM_DSS_DSS_WKDEP[15] WKUPDEP_DSI1_A_DSP2	PM_L4PER_MCSPI2_WKDEP[0] WKUPDEP_MCSPI2_MPU	PM_L4PER2_ADC_WKDEP[9] WKUPDEP_ADC_IRQ_EVE4
PM_DSS_DSS_WKDEP[14] WKUPDEP_DSI1_A_IPU1	PM_L4PER_MCSPI3_WKDEP[9] WKUPDEP_MCSPI3_EVE4	PM_L4PER2_ADC_WKDEP[8] WKUPDEP_ADC_IRQ_EVE3
PM_DSS_DSS_WKDEP[13] WKUPDEP_DSI1_A_SDMA	PM_L4PER_MCSPI3_WKDEP[8] WKUPDEP_MCSPI3_EVE3	PM_L4PER2_ADC_WKDEP[7] WKUPDEP_ADC_IRQ_EVE2
PM_DSS_DSS_WKDEP[12] WKUPDEP_DSI1_A_DSP1	PM_L4PER_MCSPI3_WKDEP[7] WKUPDEP_MCSPI3_EVE2	PM_L4PER2_ADC_WKDEP[1] WKUPDEP_ADC_IRQ_IPU2
PM_DSS_DSS_WKDEP[11] WKUPDEP_DSI1_A_IPU2	PM_L4PER_MCSPI3_WKDEP[3] WKUPDEP_MCSPI3_SDMA	PM_L4PER2_ADC_WKDEP[0] WKUPDEP_ADC_IRQ_MPU
PM_DSS_DSS_WKDEP[10] WKUPDEP_DSI1_A_MPU	PM_L4PER_MCSPI3_WKDEP[1] WKUPDEP_MCSPI3_IPU2	PRM_IRQSTATUS_DSP1[31] ABB_MPU_DONE_ST
PM_DSS_DSS_WKDEP[9] WKUPDEP_DISPC_EVE4	PM_L4PER_MCSPI3_WKDEP[0] WKUPDEP_MCSPI3_MPU	PRM_IRQSTATUS_DSP1[30] ABB_IVA_DONE_ST
PM_DSS_DSS_WKDEP[8] WKUPDEP_DISPC_EVE3	PM_L4PER_MCSPI4_WKDEP[9] WKUPDEP_MCSPI4_EVE4	PRM_IRQSTATUS_DSP1[29] ABB_DSPEVE_DONE_ST
PM_DSS_DSS_WKDEP[7] WKUPDEP_DISPC_EVE2	PM_L4PER_MCSPI4_WKDEP[8] WKUPDEP_MCSPI4_EVE3	PRM_IRQSTATUS_DSP1[28] ABB_GPU_DONE_ST
PM_DSS_DSS_WKDEP[3] WKUPDEP_DISPC_SDMA	PM_L4PER_MCSPI4_WKDEP[7] WKUPDEP_MCSPI4_EVE2	PRM_IRQSTATUS_DSP1[6] DPLL_GPU_RECAL_ST
PM_DSS_DSS_WKDEP[1] WKUPDEP_DISPC_IPU2	PM_L4PER_MCSPI4_WKDEP[3] WKUPDEP_MCSPI4_SDMA	PRM_IRQSTATUS_DSP1[4] DPLL_ABE_RECAL_ST

Table 3-288. Not Supported Functionality (Registers and Bits) (continued)

Register/Field/Value	Register/Field/Value	Register/Field/Value
PM_DSS_DSS_WKDEP[0] WKUPDEP_DISPC_MPU	PM_L4PER_MCSPI4_WKDEP[1] WKUPDEP_MCSPI4_IPU2	PRM_IRQSTATUS_DSP1[2] DPLL_IVA_RECAL_ST
PM_DSS_DSS2_WKDEP	PM_L4PER_MCSPI4_WKDEP[0] WKUPDEP_MCSPI4_MPU	PRM_IRQSTATUS_DSP1[1] DPLL_MPU_RECAL_ST
PM_EVE1_EVE1_WKDEP[9] WKUPDEP_EVE1_EVE4	PM_L4PER_MMC4_WKDEP[9] WKUPDEP_MMC4_EVE4	PRM_IRQENABLE_DSP1[31] ABB_MPU_DONE_EN
PM_EVE1_EVE1_WKDEP[8] WKUPDEP_EVE1_EVE3	PM_L4PER_MMC4_WKDEP[8] WKUPDEP_MMC4_EVE3	PRM_IRQENABLE_DSP1[30] ABB_IVA_DONE_EN
PM_EVE1_EVE1_WKDEP[7] WKUPDEP_EVE1_EVE2	PM_L4PER_MMC4_WKDEP[7] WKUPDEP_MMC4_EVE2	PRM_IRQENABLE_DSP1[29] ABB_DSPEVE_DONE_EN
PM_EVE1_EVE1_WKDEP[3] WKUPDEP_EVE1_SDMA	PM_L4PER_MMC4_WKDEP[3] WKUPDEP_MMC4_SDMA	PRM_IRQENABLE_DSP1[28] ABB_GPU_DONE_EN
PM_EVE1_EVE1_WKDEP[1] WKUPDEP_EVE1_IPU2	PM_L4PER_MMC4_WKDEP[1] WKUPDEP_MMC4_IPU2	PRM_IRQENABLE_DSP1[6] DPLL_GPU_RECAL_EN
PM_EVE1_EVE1_WKDEP[0] WKUPDEP_EVE1_MPU	PM_L4PER_MMC4_WKDEP[0] WKUPDEP_MMC4_MPU	PRM_IRQENABLE_DSP1[4] DPLL_ABE_RECAL_EN
RM_IPU1_RSTCTRL	PRM_IRQSTATUS_MPU[30] ABB_IVA_DONE_ST	PRM_IRQENABLE_DSP1[2] DPLL_IVA_RECAL_EN
PM_L4PER_I2C1_WKDEP[13] WKUPDEP_I2C1_DMA_SDMA	PRM_IRQSTATUS_MPU[29] ABB_DSPEVE_DONE_ST	PRM_IRQENABLE_DSP1[1] DPLL_MPU_RECAL_EN
PM_L4PER_I2C1_WKDEP[9] WKUPDEP_I2C1_IRQ_EVE4	PRM_IRQSTATUS_MPU[28] ABB_GPU_DONE_ST	PRM_IRQSTATUS_DSP2[31] ABB_MPU_DONE_ST
PM_L4PER_I2C1_WKDEP[8] WKUPDEP_I2C1_IRQ_EVE3	PRM_IRQSTATUS_MPU[6] DPLL_GPU_RECAL_ST	PRM_IRQSTATUS_DSP2[30] ABB_IVA_DONE_ST
PM_L4PER_I2C1_WKDEP[7] WKUPDEP_I2C1_IRQ_EVE2	PRM_IRQSTATUS_MPU[4] DPLL_ABE_RECAL_ST	PRM_IRQSTATUS_DSP2[29] ABB_DSPEVE_DONE_ST
PM_L4PER_I2C1_WKDEP[1] WKUPDEP_I2C1_IRQ_IPU2	PRM_IRQSTATUS_MPU[2] DPLL_IVA_RECAL_ST	PRM_IRQSTATUS_DSP2[28] ABB_GPU_DONE_ST
PM_L4PER_I2C1_WKDEP[0] WKUPDEP_I2C1_IRQ_MPU	PRM_IRQSTATUS_MPU[1] DPLL_MPU_RECAL_ST	PRM_IRQSTATUS_DSP2[6] DPLL_GPU_RECAL_ST
PM_L4PER_I2C2_WKDEP[9] WKUPDEP_I2C2_IRQ_EVE4	PRM_IRQENABLE_MPU[30] ABB_IVA_DONE_EN	PRM_IRQSTATUS_DSP2[4] DPLL_ABE_RECAL_ST
PM_L4PER_I2C2_WKDEP[8] WKUPDEP_I2C2_IRQ_EVE3	PRM_IRQENABLE_MPU[29] ABB_DSPEVE_DONE_EN	PRM_IRQSTATUS_DSP2[2] DPLL_IVA_RECAL_ST
PM_L4PER_I2C2_WKDEP[7] WKUPDEP_I2C2_IRQ_EVE2	PRM_IRQENABLE_MPU[28] ABB_GPU_DONE_EN	PRM_IRQSTATUS_DSP2[1] DPLL_MPU_RECAL_ST
PM_L4PER_I2C2_WKDEP[1] WKUPDEP_I2C2_IRQ_IPU2	PRM_IRQENABLE_MPU[6] DPLL_GPU_RECAL_EN	PRM_IRQENABLE_DSP2[31] ABB_MPU_DONE_EN
PM_L4PER_I2C2_WKDEP[0] WKUPDEP_I2C2_IRQ_MPU	PRM_IRQENABLE_MPU[4] DPLL_ABE_RECAL_EN	PRM_IRQENABLE_DSP2[30] ABB_IVA_DONE_EN
PM_L4PER2_QSPI_WKDEP[9] WKUPDEP_QSPI_EVE4	PRM_IRQENABLE_MPU[2] DPLL_IVA_RECAL_EN	PRM_IRQENABLE_DSP2[29] ABB_DSPEVE_DONE_EN
PM_L4PER2_QSPI_WKDEP[8] WKUPDEP_QSPI_EVE3	PRM_IRQENABLE_MPU[1] DPLL_MPU_RECAL_EN	PRM_IRQENABLE_DSP2[28] ABB_GPU_DONE_EN
PM_L4PER2_QSPI_WKDEP[7] WKUPDEP_QSPI_EVE2	PRM_IRQSTATUS_IPU1[31] ABB_MPU_DONE_ST	PRM_IRQENABLE_DSP2[6] DPLL_GPU_RECAL_EN
PM_L4PER2_QSPI_WKDEP[1] WKUPDEP_QSPI_IPU2	PRM_IRQSTATUS_IPU1[30] ABB_IVA_DONE_ST	PRM_IRQENABLE_DSP2[4] DPLL_ABE_RECAL_EN
PM_L4PER2_QSPI_WKDEP[0] WKUPDEP_QSPI_MPU	PRM_IRQSTATUS_IPU1[29] ABB_DSPEVE_DONE_ST	PRM_IRQENABLE_DSP2[2] DPLL_IVA_RECAL_EN
PM_L4PER_UART1_WKDEP[9] WKUPDEP_UART1_EVE4	PRM_IRQSTATUS_IPU1[28] ABB_GPU_DONE_ST	PRM_IRQENABLE_DSP2[1] DPLL_MPU_RECAL_EN
PM_L4PER_UART1_WKDEP[8] WKUPDEP_UART1_EVE3	PRM_IRQSTATUS_IPU1[6] DPLL_GPU_RECAL_ST	PM_WKUPAON_GPIO1_WKDEP[0] WKUPDEP_GPIO1_IRQ1_MPU
PM_L4PER_UART1_WKDEP[7] WKUPDEP_UART1_EVE2	PRM_IRQSTATUS_IPU1[4] DPLL_ABE_RECAL_ST	CM_DIV_M3_DPLL_CORE

Table 3-288. Not Supported Functionality (Registers and Bits) (continued)

Register/Field/Value	Register/Field/Value	Register/Field/Value
PM_L4PER_UART1_WKDEP[3] WKUPDEP_UART1_SDMA	PRM_IRQSTATUS_IPU1[2] DPLL_IVA_RECAL_ST	CM_DIV_H11_DPLL_CORE
PM_L4PER_UART1_WKDEP[1] WKUPDEP_UART1_IPU2	PRM_IRQSTATUS_IPU1[1] DPLL_MPU_RECAL_ST	CM_EMIF_EMIF2_CLKCTRL
PM_L4PER_UART1_WKDEP[0] WKUPDEP_UART1_MPU	CM_WKUPAON_CLKSTCTRL[18] CLKACTIVITY_UART10_GFCLK	RM_WKUPAON_TIMER12_CONTEXT
PM_L4PER_UART2_WKDEP[9] WKUPDEP_UART2_EVE4	CM_WKUPAON_CLKSTCTRL[10] CLKACTIVITY_ADC_GFCLK	CM_DIV_H13_DPLL_CORE
PM_L4PER_UART2_WKDEP[8] WKUPDEP_UART2_EVE3	CM_WKUPAON_TIMER1_CLKCTRL[27:24] CLKSEL=0x5	CM_WKUPAON_TIMER12_CLKCTRL
PM_L4PER_UART2_WKDEP[7] WKUPDEP_UART2_EVE2	CM_WKUPAON_TIMER1_CLKCTRL[27:24] CLKSEL=0x6	CM_DIV_H21_DPLL_CORE
PM_L4PER_UART2_WKDEP[3] WKUPDEP_UART2_SDMA	CM_WKUPAON_TIMER1_CLKCTRL[27:24] CLKSEL=0x8	PM_WKUPAON_TIMER12_WKDEP
PM_L4PER_UART2_WKDEP[1] WKUPDEP_UART2_IPU2	CM_WKUPAON_TIMER1_CLKCTRL[27:24] CLKSEL=0x9	PM_CUSTEFUSE_PWRSTCTRL[1:0] POWERSTATE=0x2
PM_L4PER_UART2_WKDEP[0] WKUPDEP_UART2_MPU	CM_WKUPAON_TIMER1_CLKCTRL[27:24] CLKSEL=0xA	PM_CUSTEFUSE_PWRSTST[1:0] POWERSTATEST=0x2
PM_L4PER_UART3_WKDEP[9] WKUPDEP_UART3_EVE4	PM_WKUPAON_TIMER1_WKDEP[9] WKUPDEP_TIMER1_EVE4	PM_CUSTEFUSE_PWRSTST[1:0] POWERSTATEST=0x1
PM_L4PER_UART3_WKDEP[8] WKUPDEP_UART3_EVE3	PM_WKUPAON_TIMER1_WKDEP[8] WKUPDEP_TIMER1_EVE3	PM_CUSTEFUSE_PWRSTST[25:24] LASTPOWERSTATEENTERED=0x1
PM_L4PER_UART3_WKDEP[7] WKUPDEP_UART3_EVE2	PM_WKUPAON_TIMER1_WKDEP[7] WKUPDEP_TIMER1_EVE2	PM_CUSTEFUSE_PWRSTST[25:24] LASTPOWERSTATEENTERED=0x2
PM_L4PER_UART3_WKDEP[3] WKUPDEP_UART3_SDMA	PM_WKUPAON_TIMER1_WKDEP[1] WKUPDEP_TIMER1_IPU2	PM_CUSTEFUSE_PWRSTST[2] LOGICSTATEST=0x0
PM_L4PER_UART3_WKDEP[1] WKUPDEP_UART3_IPU2	PM_WKUPAON_TIMER1_WKDEP[0] WKUPDEP_TIMER1_MPU	PM_DSP1_PWRSTCTRL[1:0] POWERSTATE=0x2
PM_L4PER_UART3_WKDEP[0] WKUPDEP_UART3_MPU	PM_WKUPAON_DCAN1_WKDEP[9] WKUPDEP_DCAN1_EVE4	PM_DSP1_PWRSTST[1:0] POWERSTATEST=0x1
PM_L4PER2_DCAN2_WKDEP[9] WKUPDEP_DCAN2_EVE4	PM_WKUPAON_DCAN1_WKDEP[8] WKUPDEP_DCAN1_EVE3	PM_DSP1_PWRSTST[1:0] POWERSTATEST=0x2
PM_L4PER2_DCAN2_WKDEP[8] WKUPDEP_DCAN2_EVE3	PM_WKUPAON_DCAN1_WKDEP[7] WKUPDEP_DCAN1_EVE2	PM_DSP1_PWRSTST[25:24] LASTPOWERSTATEENTERED=0x1
PM_L4PER2_DCAN2_WKDEP[7] WKUPDEP_DCAN2_EVE2	PM_WKUPAON_DCAN1_WKDEP[3] WKUPDEP_DCAN1_SDMA	PM_DSP1_PWRSTST[25:24] LASTPOWERSTATEENTERED=0x2
PM_L4PER2_DCAN2_WKDEP[3] WKUPDEP_DCAN2_SDMA	PM_WKUPAON_DCAN1_WKDEP[1] WKUPDEP_DCAN1_IPU2	PM_DSP1_PWRSTST[2] LOGICSTATEST=0x0
PM_L4PER2_DCAN2_WKDEP[1] WKUPDEP_DCAN2_IPU2	PM_WKUPAON_DCAN1_WKDEP[0] WKUPDEP_DCAN1_MPU	PM_DSP2_PWRSTCTRL[1:0] POWERSTATE=0x2
PM_L4PER2_DCAN2_WKDEP[0] WKUPDEP_DCAN2_MPU	PM_WKUPAON_GPIO1_WKDEP[19] WKUPDEP_GPIO1_IRQ2_EVE4	PM_DSP2_PWRSTST[1:0] POWERSTATEST=0x1
PRM_IRQSTATUS_MPU[30] ABB_IVA_DONE_ST	PM_WKUPAON_GPIO1_WKDEP[18] WKUPDEP_GPIO1_IRQ2_EVE3	PM_DSP2_PWRSTST[1:0] POWERSTATEST=0x2
PRM_IRQSTATUS_MPU[29] ABB_DSPEVE_DONE_ST	PM_WKUPAON_GPIO1_WKDEP[17] WKUPDEP_GPIO1_IRQ2_EVE2	PM_DSP2_PWRSTST[2] LOGICSTATEST=0x0
PRM_IRQSTATUS_MPU[28] ABB_GPU_DONE_ST	PM_WKUPAON_GPIO1_WKDEP[11] WKUPDEP_GPIO1_IRQ2_IPU2	PM_DSP2_PWRSTST[9:8] DSP2_EDMA_STATEST=0x1
PRM_IRQSTATUS_MPU[6] DPLL_GPU_RECAL_ST	PM_WKUPAON_GPIO1_WKDEP[10] WKUPDEP_GPIO1_IRQ2_MPU	PM_DSP2_PWRSTST[25:24] LASTPOWERSTATEENTERED=0x1
PRM_IRQSTATUS_MPU[4] DPLL_ABE_RECAL_ST	PM_WKUPAON_GPIO1_WKDEP[9] WKUPDEP_GPIO1_IRQ1_EVE4	PM_DSP2_PWRSTST[25:24] LASTPOWERSTATEENTERED=0x2
PRM_IRQSTATUS_MPU[2] DPLL_IVA_RECAL_ST	PM_WKUPAON_GPIO1_WKDEP[8] WKUPDEP_GPIO1_IRQ1_EVE3	PM_DSS_PWRSTCTRL[1:0] POWERSTATE=0x1
PRM_IRQSTATUS_MPU[1] DPLL_MPU_RECAL_ST	PM_WKUPAON_GPIO1_WKDEP[7] WKUPDEP_GPIO1_IRQ1_EVE2	PM_DSS_PWRSTCTRL[1:0] POWERSTATE=0x2

Table 3-288. Not Supported Functionality (Registers and Bits) (continued)

Register/Field/Value	Register/Field/Value	Register/Field/Value
PRM_IRQENABLE_MPU[30] ABB_IVA_DONE_EN	PM_WKUPAON_GPIO1_WKDEP[1] WKUPDEP_GPIO1_IRQ1_IPU2	PM_DSS_PWRSTCTRL[2] LOGICRETSTATE
PRM_IRQENABLE_MPU[29] ABB_DSPEVE_DONE_EN	PRM_PHASE1_CNDP	PM_DSS_PWRSTCTRL[8] DSS_MEM_RETSTATE
PRM_IRQENABLE_MPU[28] ABB_GPU_DONE_EN	PRM_PHASE2A_CNDP	PM_DSS_PWRSTST[1:0] POWERSTATEST=0x1
PRM_IRQENABLE_MPU[6] DPLL_GPU_RECAL_EN	PRM_PHASE2B_CNDP	PM_DSS_PWRSTST[1:0] POWERSTATEST=0x2
PRM_IRQENABLE_MPU[4] DPLL_ABE_RECAL_EN	PM_CAM_PWRSTST[25:24] LASTPOWERSTATEENTERED=0x1	PM_DSS_PWRSTST[25:24] LASTPOWERSTATEENTERED=0x1
PRM_IRQENABLE_MPU[2] DPLL_IVA_RECAL_EN	PM_CAM_PWRSTST[25:24] LASTPOWERSTATEENTERED=0x2	PM_DSS_PWRSTST[25:24] LASTPOWERSTATEENTERED=0x2
PRM_IRQENABLE_MPU[1] DPLL_MPU_RECAL_EN	PM_CAM_PWRSTST[1:0] POWERSTATEST=0x1	PM_DSS_PWRSTST[2] LOGICSTATEST=0x0
PM_ISS_ISS_WKDEP[9] WKUPDEP_ISS_EVE4	PM_CAM_PWRSTST[1:0] POWERSTATEST=0x2	PM_EVE1_PWRSTCTRL[1:0] POWERSTATE=0x2
PM_ISS_ISS_WKDEP[8] WKUPDEP_ISS_EVE3	PM_CAM_PWRSTST[2] LOGICSTATEST=0x0	PM_EVE1_PWRSTST[1:0] POWERSTATEST=0x2
PM_ISS_ISS_WKDEP[7] WKUPDEP_ISS_EVE2	PM_CORE_PWRSTCTRL[1:0] POWERSTATE=0x1	PM_EVE1_PWRSTST[1:0] POWERSTATEST=0x1
PM_ISS_ISS_WKDEP[1] WKUPDEP_ISS_IPU2	PM_CORE_PWRSTCTRL[1:0] POWERSTATE=0x2	PM_EVE1_PWRSTST[25:24] LASTPOWERSTATEENTERED=0x1
PM_ISS_ISS_WKDEP[0] WKUPDEP_ISS_MPU	PM_CORE_PWRSTCTRL[2] LOGICRETSTATE	PM_EVE1_PWRSTST[25:24] LASTPOWERSTATEENTERED=0x2
PM_L3INIT_PWRSTCTRL[1:0] POWERSTATE=0x1	PM_CORE_PWRSTCTRL[10] IPU_L2RAM_RETSTATE	PM_EVE1_PWRSTST[2] LOGICSTATEST=0x0
PM_L3INIT_PWRSTCTRL[1:0] POWERSTATE=0x2	PM_CORE_PWRSTCTRL[11] IPU_UNICACHE_RETSTATE	PM_IPU_PWRSTCTRL[1:0] POWERSTATE=0x1
PM_L3INIT_PWRSTCTRL[2] LOGICRETSTATE	PM_CORE_PWRSTST[2] LOGICSTATEST=0x0	PM_IPU_PWRSTCTRL[1:0] POWERSTATE=0x2
PM_L3INIT_PWRSTCTRL[8] L3INIT_BANK1_RETSTATE	PM_CORE_PWRSTST[1:0] POWERSTATEST=0x1	PM_IPU_PWRSTCTRL[2] LOGICRETSTATE
PM_L3INIT_PWRSTCTRL[9] L3INIT_BANK2_RETSTATE	PM_CORE_PWRSTST[1:0] POWERSTATEST=0x2	PM_IPU_PWRSTCTRL[8] AESSMEM_RETSTATE
PM_L3INIT_PWRSTST[1:0] POWERSTATEST=0x1	PM_CORE_PWRSTST[9:8] IPU_L2RAM_STATEST=0x1	PM_IPU_PWRSTCTRL[10] PERIPHMEM_RETSTATE
PM_L3INIT_PWRSTST[1:0] POWERSTATEST=0x2	PM_CORE_PWRSTST[11:10] IPU_UNICACHE_STATEST=0x1	PM_IPU_PWRSTST[1:0] POWERSTATEST=0x1
PM_L3INIT_PWRSTST[7:6] L3INIT_BANK2_STATEST=0x1	PM_CORE_PWRSTST[25:24] LASTPOWERSTATEENTERED=0x1	PM_IPU_PWRSTST[1:0] POWERSTATEST=0x2
PM_L3INIT_PWRSTST[25:24] LASTPOWERSTATEENTERED=0x1	PM_CORE_PWRSTST[25:24] LASTPOWERSTATEENTERED=0x2	PM_IPU_PWRSTST[5:4] AESSMEM_STATEST=0x1
PM_L3INIT_PWRSTST[25:24] LASTPOWERSTATEENTERED=0x2	PM_ISS_PWRSTST[25:24] LASTPOWERSTATEENTERED=0x2	PM_IPU_PWRSTST[25:24] LASTPOWERSTATEENTERED=0x1
PM_L3INIT_PWRSTST[2] LOGICSTATEST=0x0	PM_ISS_PWRSTST[1:0] POWERSTATEST=0x2	PM_IPU_PWRSTST[25:24] LASTPOWERSTATEENTERED=0x2
PM_ISS_PWRSTCTRL[1:0] POWERSTATE=0x1	PM_ISS_PWRSTST[2] LOGICSTATEST=0x0	PM_IPU_PWRSTST[2] LOGICSTATEST=0x0
PM_ISS_PWRSTCTRL[1:0] POWERSTATE=0x2	PM_ISS_PWRSTST[7:6] L3INIT_BANK2_STATEST=0x1	PM_L4PER_PWRSTCTRL[2] LOGICRETSTATE
PM_ISS_PWRSTCTRL[2] LOGICRETSTATE	PM_ISS_PWRSTST[25:24] LASTPOWERSTATEENTERED=0x1	PM_L4PER_PWRSTCTRL[8] RETAINED_BANK_RETSTATE
PM_ISS_PWRSTCTRL[8] ISS_BANK_RETSTATE	PM_L4PER_PWRSTCTRL[1:0] POWERSTATE=0x1	PM_L4PER_PWRSTCTRL[9] NONRETAINED_BANK_RETSTATE
PM_ISS_PWRSTST[1:0] POWERSTATEST=0x1	PM_L4PER_PWRSTCTRL[1:0] POWERSTATE=0x2	PM_L4PER_PWRSTST[1:0] POWERSTATEST=0x1

Table 3-288. Not Supported Functionality (Registers and Bits) (continued)

Register/Field/Value	Register/Field/Value	Register/Field/Value
PM_L4PER_PWRSTST[1:0] POWERSTATEST=0x2	PM_L4PER_PWRSTST[5:4] RETAINED_BANK_STATEST=0x1	PM_L4PER_PWRSTST[25:24] LASTPOWERSTATEENTERED=0x2
PM_L4PER_PWRSTST[2] LOGICSTATEST=0x0	PM_L4PER_PWRSTST[25:24] LASTPOWERSTATEENTERED=0x1	CM_DIV_H24_DPLL_CORE
CM_DIV_M3_DPLL_DDR	CM_DIV_H14_DPLL_GMAC	CM_DIV_M3_DPLL_PER
CM_DIV_H14_DPLL_PER	PRM_IRQENABLE_DSP1[13]DPLL_USB _RECAL_EN	

3.12.2 PRCM Instance Summary

Table 3-289. PRCM Instance Summary

Module Name	Module Base Address	Size
OCP_SOCKET_CM_CORE_AON	0x4A00 5000	256 Bytes
CKGEN_CM_CORE_AON	0x4A00 5100	504 Bytes
MPU_CM_CORE_AON	0x4A00 5300	44 Bytes
DSP1_CM_CORE_AON	0x4A00 5400	36 Bytes
IPU_CM_CORE_AON	0x4A00 5500	132 Bytes
DSP2_CM_CORE_AON	0x4A00 5600	36 Bytes
EVE1_CM_CORE_AON	0x4A00 5640	36 Bytes
EVE2_CM_CORE_AON	0x4A00 5680	36 Bytes
EVE3_CM_CORE_AON	0x4A00 56C0	36 Bytes
EVE4_CM_CORE_AON	0x4A00 5700	36 Bytes
RTC_CM_CORE_AON	0x4A00 5740	8 Bytes
ISS_CM_CORE_AON	0x4A00 5760	12 Bytes
RESTORE_CM_CORE_AON	0x4A00 5E00	84 Bytes
INSTR_CM_CORE_AON	0x4A00 5F00	52 Bytes
OCP_SOCKET_CM_CORE	0x4A00 8000	244 Bytes
CKGEN_CM_CORE	0x4A00 8104	296 Bytes
COREAON_CM_CORE	0x4A00 8600	212 Bytes
CORE_CM_CORE	0x4A00 8700	1876 Bytes
IVA_CM_CORE	0x4A00 8F00	44 Bytes
CAM_CM_CORE	0x4A00 9000	76 Bytes
DSS_CM_CORE	0x4A00 9100	64 Bytes
GPU_CM_CORE	0x4A00 9200	36 Bytes
L3INIT_CM_CORE	0x4A00 9300	244 Bytes
CUSTEFUSE_CM_CORE	0x4A00 9600	36 Bytes
L4PER_CM_CORE	0x4A00 9700	536 Bytes
RESTORE_CM_CORE	0x4A00 9E18	88 Bytes
SMARTREFLEX_CORE	0x4A0D D000	80 Bytes
SMARTREFLEX_DSPEVE	0x4A18 3000	80 Bytes
OCP_SOCKET_PRM	0x4AE0 6000	248 Bytes
CKGEN_PRM	0x4AE0 6100	228 Bytes
MPU_PRM	0x4AE0 6300	40 Bytes
DSP1_PRM	0x4AE0 6400	40 Bytes
IPU_PRM	0x4AE0 6500	136 Bytes
COREAON_PRM	0x4AE0 6628	184 Bytes
CORE_PRM	0x4AE0 6700	1864 Bytes

Table 3-289. PRCM Instance Summary (continued)

Module Name	Module Base Address	Size
IVA_PRM	0x4AE0 6F00	48 Bytes
CAM_PRM	0x4AE0 7000	80 Bytes
DSS_PRM	0x4AE0 7100	64 Bytes
GPU_PRM	0x4AE0 7200	40 Bytes
L3INIT_PRM	0x4AE0 7300	248 Bytes
L4PER_PRM	0x4AE0 7400	512 Bytes
CUSTEFUSE_PRM	0x4AE0 7600	40 Bytes
WKUPAON_PRM	0x4AE0 7724	188 Bytes
WKUPAON_CM	0x4AE0 7800	220 Bytes
EMU_PRM	0x4AE0 7900	40 Bytes
EMU_CM	0x4AE0 7A00	16 Bytes
DSP2_PRM	0x4AE0 7B00	40 Bytes
EVE1_PRM	0x4AE0 7B40	40 Bytes
EVE2_PRM	0x4AE0 7B80	40 Bytes
EVE3_PRM	0x4AE0 7BC0	40 Bytes
EVE4_PRM	0x4AE0 7C00	40 Bytes
RTC_PRM	0x4AE0 7C60	8 Bytes
ISS_PRM	0x4AE0 7C80	40 Bytes
DEVICE_PRM	0x4AE0 7D00	312 Bytes
INSTR_PRM	0x4AE0 7F00	52 Bytes

3.12.3 CKGEN_CM_CORE_AON registers

3.12.3.1 CKGEN_CM_CORE_AON Register Summary

Table 3-290. CKGEN_CM_CORE_AON Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	CKGEN_CM_CORE_AON Physical Address
CM_CLKSEL_CORE	RW	32	0x0000 0000	0x4A00 5100
CM_CLKSEL_ABE	RW	32	0x0000 0008	0x4A00 5108
CM_DLL_CTRL	RW	32	0x0000 0010	0x4A00 5110
CM_CLKMODE_DPLL_CORE	RW	32	0x0000 0020	0x4A00 5120
CM_IDLEST_DPLL_CORE	R	32	0x0000 0024	0x4A00 5124
CM_AUTOIDLE_DPLL_CORE	RW	32	0x0000 0028	0x4A00 5128
CM_CLKSEL_DPLL_CORE	RW	32	0x0000 002C	0x4A00 512C
CM_DIV_M2_DPLL_CORE	RW	32	0x0000 0030	0x4A00 5130
CM_DIV_M3_DPLL_CORE	RW	32	0x0000 0034	0x4A00 5134
CM_DIV_H11_DPLL_CORE	RW	32	0x0000 0038	0x4A00 5138
CM_DIV_H12_DPLL_CORE	RW	32	0x0000 003C	0x4A00 513C
CM_DIV_H13_DPLL_CORE	RW	32	0x0000 0040	0x4A00 5140
CM_DIV_H14_DPLL_CORE	RW	32	0x0000 0044	0x4A00 5144
CM_SSC_DELTAMSTEP_DPLL_CORE	RW	32	0x0000 0048	0x4A00 5148
CM_SSC_MODFREQDIV_DPLL_CORE	RW	32	0x0000 004C	0x4A00 514C
CM_DIV_H21_DPLL_CORE	RW	32	0x0000 0050	0x4A00 5150
CM_DIV_H22_DPLL_CORE	RW	32	0x0000 0054	0x4A00 5154
CM_DIV_H23_DPLL_CORE	RW	32	0x0000 0058	0x4A00 5158
CM_DIV_H24_DPLL_CORE	RW	32	0x0000 005C	0x4A00 515C

Table 3-290. CKGEN_CM_CORE_AON Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	CKGEN_CM_CORE_AON Physical Address
CM_CLKMODE_DPLL_MPU	RW	32	0x0000 0060	0x4A00 5160
CM_IDLEST_DPLL_MPU	R	32	0x0000 0064	0x4A00 5164
CM_AUTOIDLE_DPLL_MPU	RW	32	0x0000 0068	0x4A00 5168
CM_CLKSEL_DPLL_MPU	RW	32	0x0000 006C	0x4A00 516C
CM_DIV_M2_DPLL_MPU	RW	32	0x0000 0070	0x4A00 5170
CM_SSC_DELTAMSTEP_DPLL_MPU	RW	32	0x0000 0088	0x4A00 5188
CM_SSC_MODFREQDIV_DPLL_MPU	RW	32	0x0000 008C	0x4A00 518C
CM_BYPCLK_DPLL_MPU	RW	32	0x0000 009C	0x4A00 519C
CM_CLKMODE_DPLL_IVA	RW	32	0x0000 00A0	0x4A00 51A0
CM_IDLEST_DPLL_IVA	R	32	0x0000 00A4	0x4A00 51A4
CM_AUTOIDLE_DPLL_IVA	RW	32	0x0000 00A8	0x4A00 51A8
CM_CLKSEL_DPLL_IVA	RW	32	0x0000 00AC	0x4A00 51AC
CM_DIV_M2_DPLL_IVA	RW	32	0x0000 00B0	0x4A00 51B0
CM_DIV_M3_DPLL_IVA	RW	32	0x0000 00B4	0x4A00 51B4
CM_SSC_DELTAMSTEP_DPLL_IVA	RW	32	0x0000 00C8	0x4A00 51C8
CM_SSC_MODFREQDIV_DPLL_IVA	RW	32	0x0000 00CC	0x4A00 51CC
CM_BYPCLK_DPLL_IVA	RW	32	0x0000 00DC	0x4A00 51DC
CM_CLKMODE_DPLL_ABE	RW	32	0x0000 00E0	0x4A00 51E0
CM_IDLEST_DPLL_ABE	R	32	0x0000 00E4	0x4A00 51E4
CM_AUTOIDLE_DPLL_ABE	RW	32	0x0000 00E8	0x4A00 51E8
CM_CLKSEL_DPLL_ABE	RW	32	0x0000 00EC	0x4A00 51EC
CM_DIV_M2_DPLL_ABE	RW	32	0x0000 00F0	0x4A00 51F0
CM_DIV_M3_DPLL_ABE	RW	32	0x0000 00F4	0x4A00 51F4
CM_SSC_DELTAMSTEP_DPLL_ABE	RW	32	0x0000 0108	0x4A00 5208
CM_SSC_MODFREQDIV_DPLL_ABE	RW	32	0x0000 010C	0x4A00 520C
CM_CLKMODE_DPLL_DDR	RW	32	0x0000 0110	0x4A00 5210
CM_IDLEST_DPLL_DDR	R	32	0x0000 0114	0x4A00 5214
CM_AUTOIDLE_DPLL_DDR	RW	32	0x0000 0118	0x4A00 5218
CM_CLKSEL_DPLL_DDR	RW	32	0x0000 011C	0x4A00 521C
CM_DIV_M2_DPLL_DDR	RW	32	0x0000 0120	0x4A00 5220
CM_DIV_M3_DPLL_DDR	RW	32	0x0000 0124	0x4A00 5224
CM_DIV_H11_DPLL_DDR	RW	32	0x0000 0128	0x4A00 5228
CM_SSC_DELTAMSTEP_DPLL_DDR	RW	32	0x0000 012C	0x4A00 522C
CM_SSC_MODFREQDIV_DPLL_DDR	RW	32	0x0000 0130	0x4A00 5230
CM_CLKMODE_DPLL_DSP	RW	32	0x0000 0134	0x4A00 5234
CM_IDLEST_DPLL_DSP	R	32	0x0000 0138	0x4A00 5238
CM_AUTOIDLE_DPLL_DSP	RW	32	0x0000 013C	0x4A00 523C
CM_CLKSEL_DPLL_DSP	RW	32	0x0000 0140	0x4A00 5240
CM_DIV_M2_DPLL_DSP	RW	32	0x0000 0144	0x4A00 5244
CM_DIV_M3_DPLL_DSP	RW	32	0x0000 0148	0x4A00 5248
CM_SSC_DELTAMSTEP_DPLL_DSP	RW	32	0x0000 014C	0x4A00 524C
CM_SSC_MODFREQDIV_DPLL_DSP	RW	32	0x0000 0150	0x4A00 5250
CM_BYPCLK_DPLL_DSP	RW	32	0x0000 0154	0x4A00 5254
CM_SHADOW_FREQ_CONFIG1	RW	32	0x0000 0160	0x4A00 5260
CM_SHADOW_FREQ_CONFIG2	RW	32	0x0000 0164	0x4A00 5264
CM_DYN_DEP_PRESCAL	RW	32	0x0000 0170	0x4A00 5270
CM_RESTORE_ST	RW	32	0x0000 0180	0x4A00 5280

Table 3-290. CKGEN_CM_CORE_AON Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	CKGEN_CM_CORE_AON Physical Address
CM_CLKMODE_DPLL_EVE	RW	32	0x0000 0184	0x4A00 5284
CM_IDLEST_DPLL_EVE	R	32	0x0000 0188	0x4A00 5288
CM_AUTOIDLE_DPLL_EVE	RW	32	0x0000 018C	0x4A00 528C
CM_CLKSEL_DPLL_EVE	RW	32	0x0000 0190	0x4A00 5290
CM_DIV_M2_DPLL_EVE	RW	32	0x0000 0194	0x4A00 5294
CM_DIV_M3_DPLL_EVE	RW	32	0x0000 0198	0x4A00 5298
CM_SSC_DELTAMSTEP_DPLL_EVE	RW	32	0x0000 019C	0x4A00 529C
CM_SSC_MODFREQDIV_DPLL_EVE	RW	32	0x0000 01A0	0x4A00 52A0
CM_BYPCLK_DPLL_EVE	RW	32	0x0000 01A4	0x4A00 52A4
CM_CLKMODE_DPLL_GMAC	RW	32	0x0000 01A8	0x4A00 52A8
CM_IDLEST_DPLL_GMAC	R	32	0x0000 01AC	0x4A00 52AC
CM_AUTOIDLE_DPLL_GMAC	RW	32	0x0000 01B0	0x4A00 52B0
CM_CLKSEL_DPLL_GMAC	RW	32	0x0000 01B4	0x4A00 52B4
CM_DIV_M2_DPLL_GMAC	RW	32	0x0000 01B8	0x4A00 52B8
CM_DIV_M3_DPLL_GMAC	RW	32	0x0000 01BC	0x4A00 52BC
CM_DIV_H11_DPLL_GMAC	RW	32	0x0000 01C0	0x4A00 52C0
CM_DIV_H12_DPLL_GMAC	RW	32	0x0000 01C4	0x4A00 52C4
CM_DIV_H13_DPLL_GMAC	RW	32	0x0000 01C8	0x4A00 52C8
CM_DIV_H14_DPLL_GMAC	RW	32	0x0000 01CC	0x4A00 52CC
CM_SSC_DELTAMSTEP_DPLL_GMAC	RW	32	0x0000 01D0	0x4A00 52D0
CM_SSC_MODFREQDIV_DPLL_GMAC	RW	32	0x0000 01D4	0x4A00 52D4
CM_CLKMODE_DPLL_GPU	RW	32	0x0000 01D8	0x4A00 52D8
CM_IDLEST_DPLL_GPU	R	32	0x0000 01DC	0x4A00 52DC
CM_AUTOIDLE_DPLL_GPU	RW	32	0x0000 01E0	0x4A00 52E0
CM_CLKSEL_DPLL_GPU	RW	32	0x0000 01E4	0x4A00 52E4
CM_DIV_M2_DPLL_GPU	RW	32	0x0000 01E8	0x4A00 52E8
CM_DIV_M3_DPLL_GPU	RW	32	0x0000 01EC	0x4A00 52EC
CM_SSC_DELTAMSTEP_DPLL_GPU	RW	32	0x0000 01F0	0x4A00 52F0
CM_SSC_MODFREQDIV_DPLL_GPU	RW	32	0x0000 01F4	0x4A00 52F4

3.12.3.2 CKGEN_CM_CORE_AON Register Description

Table 3-291. CM_CLKSEL_CORE

Address Offset	0x0000 0000		
Physical Address	0x4A00 5100	**Instance**	CKGEN_CM_CORE_AON
Description	CORE module clock selection.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							CLKSEL_L4	RESERVED			CLKSEL_L3	RESERVED			

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	CLKSEL_L4	Selects L4 interconnect clock (L4_clk) 0x0: RESERVED 0x1: L4_CLK is L3_CLK divided by 2	R	0x1
7:5	RESERVED		R	0x0
4	CLKSEL_L3	Selects L3 interconnect clock (L3_clk) 0x0: L3_CLK is CORE_CLK divided by 1 0x1: L3_CLK is CORE_CLK divided by 2	RW	0x0
3:0	RESERVED		R	0x0

Table 3-292. Register Call Summary for Register CM_CLKSEL_CORE

Clock Management Functional Description
- CM_CORE_AON Clock Generator: [0][1]

PRCM Register Manual
- CKGEN_CM_CORE_AON Register Summary: [2]
- CKGEN_CM_CORE_AON Register Description: [3]
- RESTORE_CM_CORE_AON Register Description: [4][5]

Table 3-293. CM_CLKSEL_ABE

Address Offset	0x0000 0008		
Physical Address	0x4A00 5108	**Instance**	CKGEN_CM_CORE_AON
Description	ABE module clock selection.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																					SLIMBUS1_CLK_GATE	RESERVED	PAD_CLKS_GATE	RESERVED							CLKSEL_OPP

Bits	Field Name	Description	Type	Reset
31:11	RESERVED		R	0x0
10	SLIMBUS1_CLK_GATE	Gating control for SLIMBUS_CLK clock tree in ABE. SLIMBUS module always gets the ungated version. 0x0: The clock is gated 0x1: The clock is enabled	RW	0x0
9	RESERVED		R	0x0
8	PAD_CLKS_GATE	Gating control for PAD_CLKS clock tree in ABE 0x0: The clock is gated 0x1: The clock is enabled	RW	0x0
7:2	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
1:0	CLKSEL_OPP	Selects the OPP divider ABE domain 0x0: ABE_CLK is divide by 1 of DPLL_ABE_X2_CLK 0x1: ABE_CLK is divide by 2 of DPLL_ABE_X2_CLK 0x2: ABE_CLK is divide by 4 of DPLL_ABE_X2_CLK 0x3: Reserved	RW	0x0

Table 3-294. CM_DLL_CTRL

Address Offset	0x0000 0010		
Physical Address	0x4A00 5110	**Instance**	CKGEN_CM_CORE_AON
Description	Special register for DLL control		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															DLL_OVERRIDE

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	DLL_OVERRIDE	Control if DLL lock and code outputs are overriden or not 0x0: Lock and code outputs are not overriden 0x1: Lock output is overriden to '1' and code output is overriden with a value coming from control module.	RW	0x1

Table 3-295. Register Call Summary for Register CM_DLL_CTRL

PRCM Register Manual
- CKGEN_CM_CORE_AON Register Summary: [0]
- CKGEN_CM_CORE_AON Register Description: [1]

Table 3-296. CM_CLKMODE_DPLL_CORE

Address Offset	0x0000 0020		
Physical Address	0x4A00 5120	**Instance**	CKGEN_CM_CORE_AON
Description	This register allows controlling the DPLL modes.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																DPLL_SSC_TYPE	DPLL_SSC_DOWNSPREAD	DPLL_SSC_ACK	DPLL_SSC_EN	DPLL_REGM4XEN	DPLL_LPMODE_EN	DPLL_RELOCK_RAMP_EN	DPLL_DRIFTGUARD_EN	DPLL_RAMP_RATE			DPLL_RAMP_LEVEL				DPLL_EN

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0
15	DPLL_SSC_TYPE	Select between Triangular and SquareWave Spread Spectrum Clocking 0x0: Triangular Spread Spectrum Clocking is selected 0x1: Square Wave Spread Spectrum Clocking is selected (only available under proper licensing agreement)	RW	0x0
14	DPLL_SSC_DOWNSPREAD	Control if only low frequency spread is required 0x0: When SSC is enabled, clock frequency is spread on both sides of the programmed frequency 0x1: When SSC is enabled, clock frequency is spread only on the lower side of the programmed frequency	RW	0x0
13	DPLL_SSC_ACK	Acknowledgement from the DPLL regarding start and stop of Spread Spectrum Clocking feature 0x0: SSC has been turned off on PLL o/ps 0x1: SSC has been turned on on PLL o/ps	R	0x0
12	DPLL_SSC_EN	Enable or disable Spread Spectrum Clocking 0x0: SSC disabled 0x1: SSC enabled	RW	0x0
11	DPLL_REGM4XEN	Enable the REGM4XEN mode of the DPLL. Please check the DPLL documentation to check when this mode can be enabled. 0x0: REGM4XEN mode of the DPLL is disabled	R	0x0
10	DPLL_LPMODE_EN	Set the DPLL in Low Power mode. Check the DPLL documentation to see when this can be enabled. 0x0: Low power mode of the DPLL is disabled 0x1: Low power mode of the DPLL is enabled	RW	0x0
9	DPLL_RELOCK_RAMP_EN	If enabled, the clock ramping feature is used applied during the lock process, as well as the relock process. If disabled, the clock ramping feature is used only during the first lock.	RW	0x0
8	DPLL_DRIFTGUARD_EN	This bit allows to enable or disable the automatic recalibration feature of the DPLL. The DPLL will automatically start a recalibration process upon assertion of the DPLL's RECAL flag if this bit is set. 0x0: DRIFTGUARD feature is disabled 0x1: DRIFTGUARD feature is enabled	RW	0x0
7:5	DPLL_RAMP_RATE	Selects the time in terms of DPLL REFCLKs spent at each stage of the clock ramping process 0x0: 2 REFCLKs 0x1: 4 REFCLKs 0x2: 8 REFCLKs 0x3: 16 REFCLKs 0x4: 32 REFCLKs 0x5: 64 REFCLKs 0x6: 128 REFCLKs 0x7: 512 REFCLKs	RW	0x0

Bits	Field Name	Description	Type	Reset
4:3	DPLL_RAMP_LEVEL	The DPLL provides an output clock frequency ramping feature when switching from bypass clock to normal clock during lock and re-lock. The frequency ramping will happen in a maximum of 4 steps in frequency before the DPLL's frequency lock indicator is asserted. This register is used to enable/disable the DPLL ramping feature. If enabled, it is also used to select the algorithm used for clock ramping 0x0: CLKOUT = No ramping CLKOUTX2 = No ramping 0x1: CLKOUT = Bypass clk - Fout/8 - Fout/4 - Fout/2 - Fout CLKOUTX2 = Bypass clk - Foutx2/8 - Foutx2/4 - Foutx2/2 - Foutx2 0x2: CLKOUT = Bypass clk - Fout/4 - Fout/2 - Fout/1.5 - Fout CLKOUTX2 = Bypass clk - Foutx2/4 - Foutx2/2 - Foutx2/1.5 - Foutx2 0x3: Reserved	RW	0x0
2:0	DPLL_EN	DPLL control. 0x0: Reserved 0x1: Reserved 0x2: Reserved 0x3: Reserved 0x4: Reserved 0x5: Put the DPLL in Idle Bypass Low Power mode. 0x6: Put the DPLL in Idle Bypass Fast Relock mode. 0x7: Enables the DPLL in Lock mode	RW	0x5

Table 3-297. Register Call Summary for Register CM_CLKMODE_DPLL_CORE

Clock Management Functional Description
- DPLL_CORE Power Modes: [0][1]
- DPLL_CORE Recalibration: [2]

PRCM Register Manual
- CKGEN_CM_CORE_AON Register Summary: [5]
- RESTORE_CM_CORE_AON Register Description: [6][7]

Table 3-298. CM_IDLEST_DPLL_CORE

Address Offset	0x0000 0024		
Physical Address	0x4A00 5124	**Instance**	CKGEN_CM_CORE_AON
Description	This register allows monitoring DPLL activity. This register is read only and automatically updated. [warm reset insensitive]		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																											ST_DPLL_INIT	ST_DPLL_MODE			ST_DPLL_CLK

Bits	Field Name	Description	Type	Reset
31:5	RESERVED		R	0x0
4	ST_DPLL_INIT	DPLL init status (for debug purpose). 0x0: DPLL is not init 0x1: DPLL has been init	R	0x0
3:1	ST_DPLL_MODE	DPLL mode status (for debug purpose). 0x0: Transient state. From reset to any LP idle state or from any power state to any power state (Power states are Low Power Stop mode, Fast Relock Stop mode, Idle Bypass Low Power mode and Idle Bypass Fast Relock mode). 0x1: The DPLL is in Low Power Stop mode. 0x2: The DPLL is in Fast Relock Stop mode. 0x3: Reserved 0x4: Reserved 0x5: The DPLL is in Idle Bypass Low Power mode. 0x6: The DPLL is in Idle Bypass Fast Relock mode. 0x7: Reserved	R	0x0
0	ST_DPLL_CLK	DPLL lock status 0x0: DPLL is either in bypass mode or in stop mode. 0x1: DPLL is LOCKED	R	0x0

Table 3-299. Register Call Summary for Register CM_IDLEST_DPLL_CORE

PRCM Register Manual
- CKGEN_CM_CORE_AON Register Summary: [0]

Table 3-300. CM_AUTOIDLE_DPLL_CORE

Address Offset	0x0000 0028		
Physical Address	0x4A00 5128	**Instance**	CKGEN_CM_CORE_AON
Description	This register provides automatic control over the DPLL activity.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														AUTO_DPLL_MODE	

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0
2:0	AUTO_DPLL_MODE	DPLL automatic control. 0x0: DPLL auto control disabled 0x1: The DPLL is automatically put in Low Power Stop mode when its DPLL generated clocks are not required anymore. It is also restarted automatically. 0x2: The DPLL is automatically put in Fast Relock Stop mode when its DPLL generated clocks are not required anymore. It is also restarted automatically. 0x3: Reserved 0x4: Reserved 0x5: The DPLL is automatically put in Idle Bypass Low Power mode when its DPLL generated clocks are not required anymore. It is also restarted automatically. 0x6: The DPLL is automatically put in Idle Bypass Fast Relock mode when its DPLL generated clocks are not required anymore. It is also restarted automatically. 0x7: Reserved	RW	0x0

Table 3-301. Register Call Summary for Register CM_AUTOIDLE_DPLL_CORE

Clock Management Functional Description
- DPLL_CORE Power Modes: [0]

PRCM Register Manual
- CKGEN_CM_CORE_AON Register Summary: [1]
- RESTORE_CM_CORE_AON Register Description: [2]

Table 3-302. CM_CLKSEL_DPLL_CORE

Address Offset	0x0000 002C		
Physical Address	0x4A00 512C	**Instance**	CKGEN_CM_CORE_AON
Description	This register provides controls over the DPLL.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DPLL_BYP_CLKSEL	DCC_EN	RESERVED	DPLL_CLKOUTHIF_CLKSEL	RESERVED	DPLL_MULT											RESERVED	DPLL_DIV						

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23	DPLL_BYP_CLKSEL	Allows control of the BYPASS clock of the PLL and the associated HSDIVIDER. Same as ULOWCLKEN on DPLL. In DPLL Locked mode, 0 - No impact 1 - No impact In DPLL Bypass mode, 0 - CLKINP is selected as the BYPASS clock for CLKOUT/CLKOUTX2 1 - CLKINPULOW is selected as the BYPASS clock for CLKOUT/CLKOUTX2	RW	0x0

Bits	Field Name	Description	Type	Reset
22	DCC_EN	Duty-cycle corrector for high frequency clock 0x0: Duty-cycle corrector is disabled	R	0x0
21	RESERVED		R	0x0
20	DPLL_CLKOUTHIF_CLKSEL	Selects the source of the DPLL CLKOUTHIF clock. Same as CLKINPHIFSEL pin on the DPLL 0x0: CLKOUTHIF is generated from the DPLL oscillator (DCO) 0x1: CLKOUTHIF is generated from CLKINPHIF	RW	0x0
19	RESERVED		R	0x0
18:8	DPLL_MULT	DPLL multiplier factor (2 to 2047). (equal to input M of DPLL; M=2 to 2047 = DPLL multiplies by M). [warm reset insensitive] 0x0: Reserved 0x1: Reserved	RW	0x0
7	RESERVED		R	0x0
6:0	DPLL_DIV	DPLL divider factor (0 to 127) (equal to input N of DPLL; actual division factor is N+1). [warm reset insensitive]	RW	0x0

Table 3-303. Register Call Summary for Register CM_CLKSEL_DPLL_CORE

Clock Management Functional Description
- Programming Guide For Control Module: [0][1]
- DPLL_CORE Synthesized Clock Parameters: [2][3]

PRCM Register Manual
- CKGEN_CM_CORE_AON Register Summary: [4]
- RESTORE_CM_CORE_AON Register Description: [5][6]

Table 3-304. CM_DIV_M2_DPLL_CORE

Address Offset	0x0000 0030		
Physical Address	0x4A00 5130	**Instance**	CKGEN_CM_CORE_AON
Description	This register provides controls over the M2 divider of DPLL_CORE.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						CLKST	RESERVED					DIVHS			

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	CLKST	DPLL CLKOUT status 0x0: The clock output is gated 0x1: The clock output is enabled	R	0x0
8:5	RESERVED		R	0x0
4:0	DIVHS	This field programs the M2 post-divider factor (1 to 31) of DPLL_CORE. 0x0: Reserved 0x1: M2 = /1 0x2: M2 = /2 ... 0x1F: M2 = /31	RW	0x1

Table 3-305. Register Call Summary for Register CM_DIV_M2_DPLL_CORE

Clock Management Functional Description • DPLL_CORE Synthesized Clock Parameters: [0][1]
PRCM Register Manual • CKGEN_CM_CORE_AON Register Summary: [2] • RESTORE_CM_CORE_AON Register Description: [3][4]

Table 3-306. CM_DIV_M3_DPLL_CORE

Address Offset	0x0000 0034		
Physical Address	0x4A00 5134	**Instance**	CKGEN_CM_CORE_AON
Description	This register provides controls over the M3 divider of the DPLL.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						CLKST	RESERVED					DIVHS			

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	CLKST	DPLL CLKOUTHIF status 0x0: The clock output is gated 0x1: The clock output is enabled	R	0x0
8:5	RESERVED		R	0x0
4:0	DIVHS	This field programs the M3 post-divider factor (1 to 31) of DPLL_CORE. 0x0: Reserved 0x1: M3 = /1 0x2: M3 = /2 ... 0x1F: M3 = /31	RW	0x1

Table 3-307. CM_DIV_H11_DPLL_CORE

Address Offset	0x0000 0038		
Physical Address	0x4A00 5138	**Instance**	CKGEN_CM_CORE_AON
Description	This register provides controls over the CLKOUT1 o/p of the HSDIVIDER1.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						CLKST	RESERVED					DIVHS			

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	CLKST	HSDIVIDER1 CLKOUT1 status 0x0: The clock output is gated 0x1: The clock output is enabled	R	0x0

Bits	Field Name	Description	Type	Reset
8:6	RESERVED		R	0x0
5:0	DIVHS	This field programs the H11 post-divider factor (1 to 63) of DPLL_CORE. 0x0: Reserved 0x1: H11 = /1 0x2: H11 = /2 ... 0x3F: H11 = /63	RW	0x1

Table 3-308. CM_DIV_H12_DPLL_CORE

Address Offset	0x0000 003C		
Physical Address	0x4A00 513C	**Instance**	CKGEN_CM_CORE_AON
Description	This register provides controls over the CLKOUT2 o/p of the HSDIVIDER1 DPLL_CORE.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						CLKST	RESERVED							DIVHS	

Bits	Field Name	Description	Type	Reset
31:10	RESERVED	Reserved	R	0x0
9	CLKST	HSDIVIDER1 CLKOUT2 status 0x0: The clock output is gated 0x1: The clock output is enabled	R	0x0
8:6	RESERVED	Reserved	R	0x0
5:0	DIVHS	This field programs the H12 post-divider factor (1 to 63) of DPLL_CORE. 0x0: Reserved 0x1: H12 = /1 0x2: H12 = /2 ... 0x3F: H12 = /63	RW	0x1

Table 3-309. Register Call Summary for Register CM_DIV_H12_DPLL_CORE

Clock Management Functional Description
- DPLL_CORE Synthesized Clock Parameters: [0][1]

PRCM Register Manual
- CKGEN_CM_CORE_AON Register Summary: [2]
- CKGEN_CM_CORE_AON Register Description: [3]
- RESTORE_CM_CORE_AON Register Description: [4][5][6]

Table 3-310. CM_DIV_H13_DPLL_CORE

Address Offset	0x0000 0040		
Physical Address	0x4A00 5140	**Instance**	CKGEN_CM_CORE_AON
Description	This register provides controls over the CLKOUT3 o/p of the HSDIVIDER1.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						CLKST	RESERVED			DIVHS					

Bits	Field Name	Description	Type	Reset
31:10	RESERVED	Reserved	R	0x0
9	CLKST	HSDIVIDER1 CLKOUT3 status 0x0: The clock output is gated 0x1: The clock output is enabled	R	0x0
8:6	RESERVED	Reserved	R	0x0
5:0	DIVHS	This field programs the H13 post-divider factor (1 to 63) of DPLL_CORE. 0x0: Reserved 0x1: H13 = /1 0x2: H13 = /2 ... 0x3F: H13 = /63	RW	0x1

Table 3-311. CM_DIV_H14_DPLL_CORE

Address Offset	0x0000 0044		
Physical Address	0x4A00 5144	**Instance**	CKGEN_CM_CORE_AON
Description	This register provides controls over the CLKOUT4 o/p of the HSDIVIDER1 DPLL_CORE.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						CLKST	RESERVED			DIVHS					

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	CLKST	HSDIVIDER1 CLKOUT4 status 0x0: The clock output is gated 0x1: The clock output is enabled	R	0x0
8:6	RESERVED		R	0x0
5:0	DIVHS	This field programs the H14 post-divider factor (1 to 63) of DPLL_CORE. When a value of 63 is programmed in this register, HS divider will perform division by 2.5 that is divided by 2 at top level. 0x0: Reserved 0x1: H14 = /1 0x2: H14 = /2 ... 0x3F: H14 = /63	RW	0x1

Table 3-312. Register Call Summary for Register CM_DIV_H14_DPLL_CORE

Clock Management Functional Description • DPLL_CORE Synthesized Clock Parameters: [0][1]
PRCM Register Manual • CKGEN_CM_CORE_AON Register Summary: [2] • RESTORE_CM_CORE_AON Register Description: [3][4]

Table 3-313. CM_SSC_DELTAMSTEP_DPLL_CORE

Address Offset	0x0000 0048		
Physical Address	0x4A00 5148	**Instance**	CKGEN_CM_CORE_AON
Description	Control the DeltaMStep parameter for Spread Spectrum Clocking. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DELTAMSTEP																							

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19:0	DELTAMSTEP	DeltaMStep is split into fractional and integer part. For Triangular Spread Spectrum: [19:18] for integer part, [17:0] for fractional part For Square Wave Spread Spectrum [19:14] for integer part, [13:0] for fractional part	RW	0x0

Table 3-314. CM_SSC_MODFREQDIV_DPLL_CORE

Address Offset	0x0000 004C		
Physical Address	0x4A00 514C	**Instance**	CKGEN_CM_CORE_AON
Description	Control the Modulation Frequency (Fm) for Spread Spectrum Clocking. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																					MODFREQDIV_EXPONENT			RESERVED	MODFREQDIV_MANTISSA						

Bits	Field Name	Description	Type	Reset
31:11	RESERVED		R	0x0
10:8	MODFREQDIV_EXPONENT	Set the Exponent component of MODFREQDIV factor	RW	0x0
7	RESERVED		R	0x0
6:0	MODFREQDIV_MANTISSA	Set the Mantissa component of MODFREQDIV factor	RW	0x0

Table 3-315. CM_DIV_H21_DPLL_CORE

Address Offset	0x0000 0050		
Physical Address	0x4A00 5150	**Instance**	CKGEN_CM_CORE_AON
Description	This register provides controls over the CLKOUT1 o/p of the 2nd HSDIVIDER.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						CLKST	RESERVED					DIVHS			

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	CLKST	HSDIVIDER2 CLKOUT2 status 0x0: The clock output is gated 0x1: The clock output is enabled	R	0x0
8:6	RESERVED		R	0x0
5:0	DIVHS	This field programs the H21 post-divider factor (1 to 63) of DPLL_CORE. 0x0: Reserved 0x1: H21 = /1 0x2: H21 = /2 ... 0x3F: H21 = /63	RW	0x1

Table 3-316. CM_DIV_H22_DPLL_CORE

Address Offset	0x0000 0054		
Physical Address	0x4A00 5154	**Instance**	CKGEN_CM_CORE_AON
Description	This register provides controls over the CLKOUT2 o/p of the 2nd HSDIVIDER DPLL_CORE.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						CLKST	RESERVED					DIVHS			

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	CLKST	HSDIVIDER2 CLKOUT2 status 0x0: The clock output is gated 0x1: The clock output is enabled	R	0x0
8:6	RESERVED		R	0x0

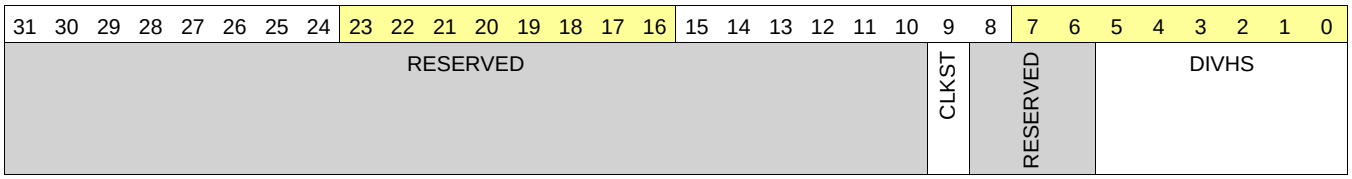
Bits	Field Name	Description	Type	Reset
5:0	DIVHS	This field programs the H22 post-divider factor (1 to 63) of DPLL_CORE. 0x0: Reserved 0x1: H22 = /1 0x2: H22 = /2 ... 0x3F: H22 = /63	RW	0x1

Table 3-317. Register Call Summary for Register CM_DIV_H22_DPLL_CORE

Clock Management Functional Description • DPLL_CORE Synthesized Clock Parameters: [0][1]
PRCM Register Manual • CKGEN_CM_CORE_AON Register Summary: [2] • RESTORE_CM_CORE_AON Register Description: [3][4]

Table 3-318. CM_DIV_H23_DPLL_CORE

Address Offset	0x0000 0058		
Physical Address	0x4A00 5158	**Instance**	CKGEN_CM_CORE_AON
Description	This register provides controls over the CLKOUT3 o/p of the 2nd HSDIVIDER DPLL_CORE.		
Type	RW		



Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	CLKST	HSDIVIDER2 CLKOUT3 status 0x0: The clock output is gated 0x1: The clock output is enabled	R	0x0
8:6	RESERVED		R	0x0
5:0	DIVHS	This field programs the H23 post-divider factor (1 to 63) of DPLL_CORE. 0x0: Reserved 0x1: H23 = /1 0x2: H23 = /2 ... 0x3F: H23 = /63	RW	0x1

Table 3-319. Register Call Summary for Register CM_DIV_H23_DPLL_CORE

Clock Management Functional Description • DPLL_CORE Synthesized Clock Parameters: [0][1]
PRCM Register Manual • CKGEN_CM_CORE_AON Register Summary: [2] • RESTORE_CM_CORE_AON Register Description: [3][4]

Table 3-320. CM_DIV_H24_DPLL_CORE

Address Offset	0x0000 005C		
Physical Address	0x4A00 515C	**Instance**	CKGEN_CM_CORE_AON
Description	This register provides controls over the CLKOUT4 o/p of the 2nd HSDIVIDER DPLL_CORE.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						CLKST	RESERVED					DIVHS			

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	CLKST	HSDIVIDER2 CLKOUT4 status 0x0: The clock output is gated 0x1: The clock output is enabled	R	0x0
8:6	RESERVED		R	0x0
5:0	DIVHS	This field programs the H24 post-divider factor (1 to 63) of DPLL_CORE. 0x0: Reserved 0x1: H24 = /1 0x2: H24 = /2 ... 0x3F: H24 = /63	RW	0x1

Table 3-321. CM_CLKMODE_DPLL_MPU

Address Offset	0x0000 0060		
Physical Address	0x4A00 5160	**Instance**	CKGEN_CM_CORE_AON
Description	This register allows controlling the DPLL modes.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																DPLL_SSC_TYPE	DPLL_SSC_DOWNSPREAD	DPLL_SSC_ACK	DPLL_SSC_EN	DPLL_REGM4XEN	DPLL_LPMODE_EN	DPLL_RELOCK_RAMP_EN	DPLL_DRIFTGUARD_EN	DPLL_RAMP_RATE			DPLL_RAMP_LEVEL				DPLL_EN

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0
15	DPLL_SSC_TYPE	Select between Triangular and SquareWave Spread Spectrum Clocking 0x0: Triangular Spread Spectrum Clocking is selected 0x1: Square Wave Spread Spectrum Clocking is selected (only available under proper licensing agreement)	RW	0x0

Bits	Field Name	Description	Type	Reset
14	DPLL_SSC_DOWNSPREAD	Control if only low frequency spread is required 0x0: When SSC is enabled, clock frequency is spread on both sides of the programmed frequency 0x1: When SSC is enabled, clock frequency is spread only on the lower side of the programmed frequency	RW	0x0
13	DPLL_SSC_ACK	Acknowledgement from the DPLL regarding start and stop of Spread Spectrum Clocking feature 0x0: SSC has been turned off on PLL o/ps 0x1: SSC has been turned on on PLL o/ps	R	0x0
12	DPLL_SSC_EN	Enable or disable Spread Spectrum Clocking 0x0: SSC disabled 0x1: SSC enabled	RW	0x0
11	DPLL_REGM4XEN	Enable the REGM4XEN mode of the DPLL. Please check the DPLL documentation to check when this mode can be enabled. 0x0: REGM4XEN mode of the DPLL is disabled	R	0x0
10	DPLL_LPMODE_EN	Set the DPLL in Low Power mode. Check the DPLL documentation to see when this can be enabled. 0x0: Low power mode of the DPLL is disabled 0x1: Low power mode of the DPLL is enabled	RW	0x0
9	DPLL_RELOCK_RAMP_EN	If enabled, the clock ramping feature is used applied during the lock process, as well as the relock process. If disabled, the clock ramping feature is used only during the first lock.	RW	0x0
8	DPLL_DRIFTGUARD_EN	This bit allows to enable or disable the automatic recalibration feature of the DPLL. The DPLL will automatically start a recalibration process upon assertion of the DPLL's RECAL flag if this bit is set. 0x0: DRIFTGUARD feature is disabled 0x1: DRIFTGUARD feature is enabled	RW	0x0
7:5	DPLL_RAMP_RATE	Selects the time in terms of DPLL REFCLKs spent at each stage of the clock ramping process 0x0: 2 REFCLKs 0x1: 4 REFCLKs 0x2: 8 REFCLKs 0x3: 16 REFCLKs 0x4: 32 REFCLKs 0x5: 64 REFCLKs 0x6: 128 REFCLKs 0x7: 512 REFCLKs	RW	0x0
4:3	DPLL_RAMP_LEVEL	The DPLL provides an output clock frequency ramping feature when switching from bypass clock to normal clock during lock and re-lock. The frequency ramping will happen in a maximum of 4 steps in frequency before the DPLL's frequency lock indicator is asserted. This register is used to enable/disable the DPLL ramping feature. If enabled, it is also used to select the algorithm used for clock ramping 0x0: CLKOUT = No ramping CLKOUTX2 = No ramping 0x1: CLKOUT = Bypass clk - Fout/8 - Fout/4 - Fout/2 - Fout CLKOUTX2 = Bypass clk - Foutx2/8 - Foutx2/4 - Foutx2/2 - Foutx2 0x2: CLKOUT = Bypass clk - Fout/4 - Fout/2 - Fout/1.5 - Fout CLKOUTX2 = Bypass clk - Foutx2/4 - Foutx2/2 - Foutx2/1.5 - Foutx2 0x3: Reserved	RW	0x0

Bits	Field Name	Description	Type	Reset
2:0	DPLL_EN	DPLL control. 0x0: Reserved 0x1: Reserved 0x2: Reserved 0x3: Reserved 0x4: Reserved 0x5: Put the DPLL in Idle Bypass Low Power mode. 0x6: Put the DPLL in Idle Bypass Fast Relock mode. 0x7: Enables the DPLL in Lock mode	RW	0x5

Table 3-322. CM_IDLEST_DPLL_MPU

Address Offset	0x0000 0064		
Physical Address	0x4A00 5164	**Instance**	CKGEN_CM_CORE_AON
Description	This register allows monitoring DPLL activity. This register is read only and automatically updated. [warm reset insensitive]		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																											ST_DPLL_INIT	ST_DPLL_MODE			ST_DPLL_CLK

Bits	Field Name	Description	Type	Reset
31:5	RESERVED		R	0x0
4	ST_DPLL_INIT	DPLL init status (for debug purpose). 0x0: DPLL is not init 0x1: DPLL has been init	R	0x0
3:1	ST_DPLL_MODE	DPLL mode status (for debug purpose). 0x0: Transient state. From reset to any LP idle state or from any power state to any power state (Power states are Low Power Stop mode, Fast Relock Stop mode, Idle Bypass Low Power mode and Idle Bypass Fast Relock mode). 0x1: The DPLL is in Low Power Stop mode. 0x2: The DPLL is in Fast Relock Stop mode. 0x3: Reserved 0x4: Reserved 0x5: The DPLL is in Idle Bypass Low Power mode. 0x6: The DPLL is in Idle Bypass Fast Relock mode. 0x7: Reserved	R	0x0
0	ST_DPLL_CLK	DPLL lock status 0x0: DPLL is either in bypass mode or in stop mode. 0x1: DPLL is LOCKED	R	0x0

Table 3-323. CM_AUTOIDLE_DPLL_MPU

Address Offset	0x0000 0068		
Physical Address	0x4A00 5168	**Instance**	CKGEN_CM_CORE_AON
Description	This register provides automatic control over the DPLL activity.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														AUTO_DPLL_MODE	

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0
2:0	AUTO_DPLL_MODE	DPLL automatic control; 0x0: DPLL auto control disabled 0x1: The DPLL is automatically put in Low Power Stop mode when its DPLL generated clocks are not required anymore. It is also restarted automatically. 0x2: The DPLL is automatically put in Fast Relock Stop mode when its DPLL generated clocks are not required anymore. It is also restarted automatically. 0x3: Reserved 0x4: Reserved 0x5: The DPLL is automatically put in Idle Bypass Low Power mode when its DPLL generated clocks are not required anymore. It is also restarted automatically. 0x6: The DPLL is automatically put in Idle Bypass Fast Relock mode when its DPLL generated clocks are not required anymore. It is also restarted automatically. 0x7: Reserved	RW	0x0

Table 3-324. CM_CLKSEL_DPLL_MPU

Address Offset	0x0000 006C		
Physical Address	0x4A00 516C	**Instance**	CKGEN_CM_CORE_AON
Description	This register provides controls over the DPLL.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DPLL_BYP_CLKSEL	DCC_EN	RESERVED						DPLL_MULT								RESERVED	DPLL_DIV						

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23	DPLL_BYP_CLKSEL	Only CLKINPULOW bypass clock supported for this PLL	R	0x1
22	DCC_EN	Duty-cycle corrector for high frequency clock 0x0: Duty-cycle corrector is disabled 0x1: Duty-cycle corrector is enabled	RW	0x0
21:19	RESERVED		R	0x0
18:8	DPLL_MULT	DPLL multiplier factor (2 to 2047). (equal to input M of DPLL; M=2 to 2047 = DPLL multiplies by M). [warm reset insensitive] 0x0: Reserved 0x1: Reserved	RW	0x0
7	RESERVED		R	0x0
6:0	DPLL_DIV	DPLL divider factor (0 to 127) (equal to input N of DPLL; actual division factor is N+1). [warm reset insensitive]	RW	0x0

Table 3-325. CM_DIV_M2_DPLL_MPU

Address Offset	0x0000 0070		
Physical Address	0x4A00 5170	**Instance**	CKGEN_CM_CORE_AON
Description	This register provides controls over the M2 divider of the DPLL.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						CLKST	RESERVED					DIVHS			

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	CLKST	DPLL CLKOUT status 0x0: The clock output is gated 0x1: The clock output is enabled	R	0x0
8:5	RESERVED		R	0x0
4:0	DIVHS	This field programs the M2 post-divider factor (1 to 31) of DPLL_MPU. 0x0: Reserved 0x1: M2 = /1 0x2: M2 = /2 **...** 0x1F: M2 = /31	RW	0x1

Table 3-326. CM_SSC_DELTAMSTEP_DPLL_MPU

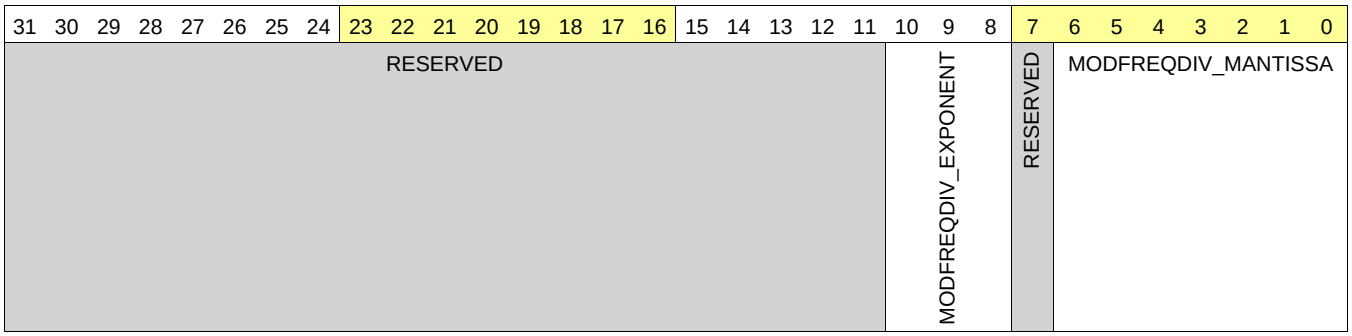
Address Offset	0x0000 0088		
Physical Address	0x4A00 5188	**Instance**	CKGEN_CM_CORE_AON
Description	Control the DeltaMStep parameter for Spread Spectrum Clocking. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																DELTAMSTEP															

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19:0	DELTAMSTEP	DeltaMStep is split into fractional and integer part. For Triangular Spread Spectrum: [19:18] for integer part, [17:0] for fractional part For Square Wave Spread Spectrum [19:14] for integer part, [13:0] for fractional part	RW	0x0

Table 3-327. CM_SSC_MODFREQDIV_DPLL_MPU

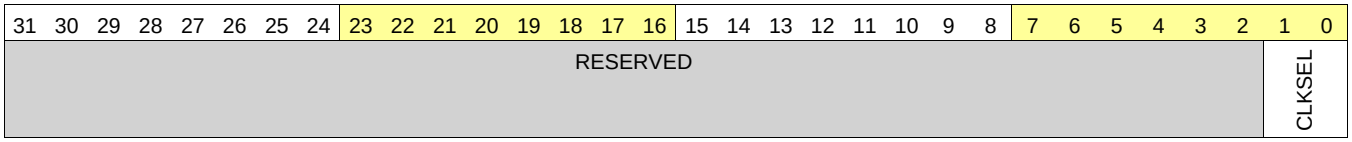
Address Offset	0x0000 008C		
Physical Address	0x4A00 518C	**Instance**	CKGEN_CM_CORE_AON
Description	Control the Modulation Frequency (Fm) for Spread Spectrum Clocking. [warm reset insensitive]		
Type	RW		



Bits	Field Name	Description	Type	Reset
31:11	RESERVED		R	0x0
10:8	MODFREQDIV_EXPONENT	Set the Exponent component of MODFREQDIV factor	RW	0x0
7	RESERVED		R	0x0
6:0	MODFREQDIV_MANTISSA	Set the Mantissa component of MODFREQDIV factor	RW	0x0

Table 3-328. CM_BYPCLK_DPLL_MPU

Address Offset	0x0000 009C		
Physical Address	0x4A00 519C	**Instance**	CKGEN_CM_CORE_AON
Description	Control MPU PLL BYPASS clock. [warm reset insensitive]		
Type	RW		



Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1:0	CLKSEL	Select the DPLL MPU bypass clock 0x0: DPLL_MPU bypass clock is CORE_X2_CLK divided by 1 0x1: DPLL_MPU bypass clock is CORE_X2_CLK divided by 2 0x2: DPLL_MPU bypass clock is CORE_X2_CLK divided by 4 0x3: DPLL_MPU bypass clock is CORE_X2_CLK divided by 8	RW	0x0

Table 3-329. CM_CLKMODE_DPLL_IVA

Address Offset	0x0000 00A0		
Physical Address	0x4A00 51A0	**Instance**	CKGEN_CM_CORE_AON
Description	This register allows controlling the DPLL modes.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																DPLL_SSC_TYPE	DPLL_SSC_DOWNSPREAD	DPLL_SSC_ACK	DPLL_SSC_EN	DPLL_REGM4XEN	DPLL_LPMODE_EN	DPLL_RELOCK_RAMP_EN	DPLL_DRIFTGUARD_EN	DPLL_RAMP_RATE			DPLL_RAMP_LEVEL				DPLL_EN

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0
15	DPLL_SSC_TYPE	Select between Triangular and SquareWave Spread Spectrum Clocking 0x0: Triangular Spread Spectrum Clocking is selected 0x1: Square Wave Spread Spectrum Clocking is selected (only available under proper licensing agreement)	RW	0x0
14	DPLL_SSC_DOWNSPREAD	Control if only low frequency spread is required 0x0: When SSC is enabled, clock frequency is spread on both sides of the programmed frequency 0x1: When SSC is enabled, clock frequency is spread only on the lower side of the programmed frequency	RW	0x0
13	DPLL_SSC_ACK	Acknowledgement from the DPLL regarding start and stop of Spread Spectrum Clocking feature 0x0: SSC has been turned off on PLL o/ps 0x1: SSC has been turned on on PLL o/ps	R	0x0
12	DPLL_SSC_EN	Enable or disable Spread Spectrum Clocking 0x0: SSC disabled 0x1: SSC enabled	RW	0x0
11	DPLL_REGM4XEN	Enable the REGM4XEN mode of the DPLL. Please check the DPLL documentation to check when this mode can be enabled. 0x0: REGM4XEN mode of the DPLL is disabled	R	0x0
10	DPLL_LPMODE_EN	Set the DPLL in Low Power mode. Check the DPLL documentation to see when this can be enabled. 0x0: Low power mode of the DPLL is disabled 0x1: Low power mode of the DPLL is enabled	RW	0x0
9	DPLL_RELOCK_RAMP_EN	If enabled, the clock ramping feature is used applied during the lock process, as well as the relock process. If disabled, the clock ramping feature is used only during the first lock.	RW	0x0
8	DPLL_DRIFTGUARD_EN	This bit allows to enable or disable the automatic recalibration feature of the DPLL. The DPLL will automatically start a recalibration process upon assertion of the DPLL's RECAL flag if this bit is set. 0x0: DRIFTGUARD feature is disabled 0x1: DRIFTGUARD feature is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
7:5	DPLL_RAMP_RATE	Selects the time in terms of DPLL REFCLKs spent at each stage of the clock ramping process 0x0: 2 REFCLKs 0x1: 4 REFCLKs 0x2: 8 REFCLKs 0x3: 16 REFCLKs 0x4: 32 REFCLKs 0x5: 64 REFCLKs 0x6: 128 REFCLKs 0x7: 512 REFCLKs	RW	0x0
4:3	DPLL_RAMP_LEVEL	The DPLL provides an output clock frequency ramping feature when switching from bypass clock to normal clock during lock and re-lock. The frequency ramping will happen in a maximum of 4 steps in frequency before the DPLL's frequency lock indicator is asserted. This register is used to enable/disable the DPLL ramping feature. If enabled, it is also used to select the algorithm used for clock ramping 0x0: CLKOUT = No ramping CLKOUTX2 = No ramping 0x1: CLKOUT = Bypass clk - Fout/8 - Fout/4 - Fout/2 - Fout CLKOUTX2 = Bypass clk - Foutx2/8 - Foutx2/4 - Foutx2/2 - Foutx2 0x2: CLKOUT = Bypass clk - Fout/4 - Fout/2 - Fout/1.5 - Fout CLKOUTX2 = Bypass clk - Foutx2/4 - Foutx2/2 - Foutx2/1.5 - Foutx2 0x3: Reserved	RW	0x0
2:0	DPLL_EN	DPLL control. 0x0: Reserved 0x1: Reserved 0x2: Reserved 0x3: Reserved 0x4: Reserved 0x5: Put the DPLL in Idle Bypass Low Power mode. 0x6: Put the DPLL in Idle Bypass Fast Relock mode. 0x7: Enables the DPLL in Lock mode	RW	0x5

Table 3-330. CM_IDLEST_DPLL_IVA

Address Offset	0x0000 00A4		
Physical Address	0x4A00 51A4	**Instance**	CKGEN_CM_CORE_AON
Description	This register allows monitoring DPLL activity. This register is read only and automatically updated. [warm reset insensitive]		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																											ST_DPLL_INIT	ST_DPLL_MODE			ST_DPLL_CLK

Bits	Field Name	Description	Type	Reset
31:5	RESERVED		R	0x0
4	ST_DPLL_INIT	DPLL init status (for debug purpose) 0x0: DPLL is not init 0x1: DPLL has been init	R	0x0
3:1	ST_DPLL_MODE	DPLL mode status (for debug purpose). 0x0: Transient state. From reset to any LP idle state or from any power state to any power state (Power states are Low Power Stop mode, Fast Relock Stop mode, Idle Bypass Low Power mode and Idle Bypass Fast Relock mode). 0x1: The DPLL is in Low Power Stop mode. 0x2: The DPLL is in Fast Relock Stop mode. 0x3: Reserved 0x4: Reserved 0x5: The DPLL is in Idle Bypass Low Power mode. 0x6: The DPLL is in Idle Bypass Fast Relock mode. 0x7: Reserved	R	0x0
0	ST_DPLL_CLK	DPLL lock status 0x0: DPLL is either in bypass mode or in stop mode. 0x1: DPLL is LOCKED	R	0x0

Table 3-331. CM_AUTOIDLE_DPLL_IVA

Address Offset	0x0000 00A8		
Physical Address	0x4A00 51A8	**Instance**	CKGEN_CM_CORE_AON
Description	This register provides automatic control over the DPLL activity.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														AUTO_DPLL_MODE	

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0
2:0	AUTO_DPLL_MODE	DPLL automatic control; 0x0: DPLL auto control disabled 0x1: The DPLL is automatically put in Low Power Stop mode when its DPLL generated clocks are not required anymore. It is also restarted automatically. 0x2: The DPLL is automatically put in Fast Relock Stop mode when its DPLL generated clocks are not required anymore. It is also restarted automatically. 0x3: Reserved 0x4: Reserved 0x5: The DPLL is automatically put in Idle Bypass Low Power mode when its DPLL generated clocks are not required anymore. It is also restarted automatically. 0x6: The DPLL is automatically put in Idle Bypass Fast Relock mode when its DPLL generated clocks are not required anymore. It is also restarted automatically. 0x7: Reserved	RW	0x0

Table 3-332. CM_CLKSEL_DPLL_IVA

Address Offset	0x0000 00AC		
Physical Address	0x4A00 51AC	**Instance**	CKGEN_CM_CORE_AON
Description	This register provides controls over the DPLL.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DPLL_BYP_CLKSEL	DCC_EN	RESERVED						DPLL_MULT								RESERVED	DPLL_DIV						

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23	DPLL_BYP_CLKSEL	Allows control of the BYPASS clock of the PLL and the associated HSDIVIDER. Same as ULOWCLKEN on DPLL. In DPLL Locked mode, 0 - No impact 1 - No impact In DPLL Bypass mode, 0 - CLKINP is selected as the BYPASS clock for CLKOUT/CLKOUTX2 1 - CLKINPULOW is selected as the BYPASS clock for CLKOUT/CLKOUTX2	RW	0x0
22	DCC_EN	Duty-cycle corrector for high frequency clock 0x0: Duty-cycle corrector is disabled	R	0x0
21:19	RESERVED		R	0x0
18:8	DPLL_MULT	DPLL multiplier factor (2 to 2047). (equal to input M of DPLL; M=2 to 2047 = DPLL multiplies by M). [warm reset insensitive] 0x0: Reserved 0x1: Reserved	RW	0x0
7	RESERVED		R	0x0
6:0	DPLL_DIV	DPLL divider factor (0 to 127) (equal to input N of DPLL; actual division factor is N+1). [warm reset insensitive]	RW	0x0

Table 3-333. CM_DIV_M2_DPLL_IVA

Address Offset	0x0000 00B0		
Physical Address	0x4A00 51B0	**Instance**	CKGEN_CM_CORE_AON
Description	This register provides controls over the M2 divider of the DPLL.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						CLKST	RESERVED						DIVHS		

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	CLKST	DPLL CLKOUT status 0x0: The clock output is gated 0x1: The clock output is enabled	R	0x0
8:5	RESERVED		R	0x0
4:0	DIVHS	This field programs the M2 post-divider factor (1 to 31) of DPLL_IVA. 0x0: Reserved 0x1: M2 = /1 0x2: M2 = /2 **...** 0x1F: M2 = /31	RW	0x1

Table 3-334. CM_DIV_M3_DPLL_IVA

Address Offset	0x0000 00B4		
Physical Address	0x4A00 51B4	**Instance**	CKGEN_CM_CORE_AON
Description	This register provides controls over the M3 divider of the DPLL.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						CLKST	RESERVED						DIVHS		

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	CLKST	DPLL CLKOUTHIF status 0x0: The clock output is gated 0x1: The clock output is enabled	R	0x0
8:5	RESERVED		R	0x0
4:0	DIVHS	This field programs the M3 post-divider factor (1 to 31) of DPLL_IVA. 0x0: Reserved 0x1: M3 = /1 0x2: M3 = /2 **...** 0x1F: M3 = /31	RW	0x1

Table 3-335. CM_SSC_DELTAMSTEP_DPLL_IVA

Address Offset	0x0000 00C8		
Physical Address	0x4A00 51C8	**Instance**	CKGEN_CM_CORE_AON
Description	Control the DeltaMStep parameter for Spread Spectrum Clocking. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																DELTAMSTEP															

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19:0	DELTAMSTEP	DeltaMStep is split into fractional and integer part. For Triangular Spread Spectrum: [19:18] for integer part, [17:0] for fractional part For Square Wave Spread Spectrum [19:14] for integer part, [13:0] for fractional part	RW	0x0

Table 3-336. CM_SSC_MODFREQDIV_DPLL_IVA

Address Offset	0x0000 00CC		
Physical Address	0x4A00 51CC	**Instance**	CKGEN_CM_CORE_AON
Description	Control the Modulation Frequency (Fm) for Spread Spectrum Clocking. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																					MODFREQDIV_EXPONENT			RESERVED	MODFREQDIV_MANTISSA						

Bits	Field Name	Description	Type	Reset
31:11	RESERVED		R	0x0
10:8	MODFREQDIV_EXPONENT	Set the Exponent component of MODFREQDIV factor	RW	0x0
7	RESERVED		R	0x0
6:0	MODFREQDIV_MANTISSA	Set the Mantissa component of MODFREQDIV factor	RW	0x0

Table 3-337. CM_BYPCLK_DPLL_IVA

Address Offset	0x0000 00DC		
Physical Address	0x4A00 51DC	**Instance**	CKGEN_CM_CORE_AON
Description	Control IVA PLL BYPASS clock. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														CLKSEL	

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1:0	CLKSEL	Select the DPLL IVA bypass clock 0x0: DPLL_IVA bypass clock is CORE_X2_CLK divided by 1 0x1: DPLL_IVA bypass clock is CORE_X2_CLK divided by 2 0x2: DPLL_IVA bypass clock is CORE_X2_CLK divided by 4 0x3: DPLL_IVA bypass clock is CORE_X2_CLK divided by 8	RW	0x0

Table 3-338. CM_CLKMODE_DPLL_ABE

Address Offset	0x0000 00E0		
Physical Address	0x4A00 51E0	**Instance**	CKGEN_CM_CORE_AON
Description	This register allows controlling the DPLL modes.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																DPLL_SSC_TYPE	DPLL_SSC_DOWNSPREAD	DPLL_SSC_ACK	DPLL_SSC_EN	DPLL_REGM4XEN	DPLL_LPMODE_EN	DPLL_RELOCK_RAMP_EN	DPLL_DRIFTGUARD_EN	DPLL_RAMP_RATE			DPLL_RAMP_LEVEL				DPLL_EN

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0
15	DPLL_SSC_TYPE	Select between Triangular and SquareWave Spread Spectrum Clocking 0x0: Triangular Spread Spectrum Clocking is selected 0x1: Square Wave Spread Spectrum Clocking is selected (only available under proper licensing agreement)	RW	0x0
14	DPLL_SSC_DOWNSPREAD	Control if only low frequency spread is required 0x0: When SSC is enabled, clock frequency is spread on both sides of the programmed frequency 0x1: When SSC is enabled, clock frequency is spread only on the lower side of the programmed frequency	RW	0x0
13	DPLL_SSC_ACK	Acknowledgement from the DPLL regarding start and stop of Spread Spectrum Clocking feature 0x0: SSC has been turned off on PLL o/ps 0x1: SSC has been turned on on PLL o/ps	R	0x0
12	DPLL_SSC_EN	Enable or disable Spread Spectrum Clocking 0x0: SSC disabled 0x1: SSC enabled	RW	0x0
11	DPLL_REGM4XEN	Enable the REGM4XEN mode of the DPLL. Please check the DPLL documentation to check when this mode can be enabled. 0x0: REGM4XEN mode of the DPLL is disabled 0x1: REGM4XEN mode of the DPLL is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
10	DPLL_LPMODE_EN	Set the DPLL in Low Power mode. Check the DPLL documentation to see when this can be enabled. 0x0: Low power mode of the DPLL is disabled 0x1: Low power mode of the DPLL is enabled	RW	0x0
9	DPLL_RELOCK_RAMP_EN	If enabled, the clock ramping feature is used applied during the lock process, as well as the relock process. If disabled, the clock ramping feature is used only during the first lock.	RW	0x0
8	DPLL_DRIFTGUARD_EN	This bit allows to enable or disable the automatic recalibration feature of the DPLL. The DPLL will automatically start a recalibration process upon assertion of the DPLL's RECAL flag if this bit is set. 0x0: DRIFTGUARD feature is disabled 0x1: DRIFTGUARD feature is enabled	RW	0x0
7:5	DPLL_RAMP_RATE	Selects the time in terms of DPLL REFCLKs spent at each stage of the clock ramping process 0x0: 2 REFCLKs 0x1: 4 REFCLKs 0x2: 8 REFCLKs 0x3: 16 REFCLKs 0x4: 32 REFCLKs 0x5: 64 REFCLKs 0x6: 128 REFCLKs 0x7: 512 REFCLKs	RW	0x0
4:3	DPLL_RAMP_LEVEL	The DPLL provides an output clock frequency ramping feature when switching from bypass clock to normal clock during lock and re-lock. The frequency ramping will happen in a maximum of 4 steps in frequency before the DPLL's frequency lock indicator is asserted. This register is used to enable/disable the DPLL ramping feature. If enabled, it is also used to select the algorithm used for clock ramping 0x0: CLKOUT = No ramping CLKOUTX2 = No ramping 0x1: CLKOUT = Bypass clk - Fout/8 - Fout/4 - Fout/2 - Fout CLKOUTX2 = Bypass clk - Foutx2/8 - Foutx2/4 - Foutx2/2 - Foutx2 0x2: CLKOUT = Bypass clk - Fout/4 - Fout/2 - Fout/1.5 - Fout CLKOUTX2 = Bypass clk - Foutx2/4 - Foutx2/2 - Foutx2/1.5 - Foutx2 0x3: Reserved	RW	0x0
2:0	DPLL_EN	DPLL control. 0x0: Reserved 0x1: Reserved 0x2: Reserved 0x3: Reserved 0x4: Reserved 0x5: Put the DPLL in Idle Bypass Low Power mode. 0x6: Put the DPLL in Idle Bypass Fast Relock mode. 0x7: Enables the DPLL in Lock mode	RW	0x5

Table 3-339. CM_IDLEST_DPLL_ABE

Address Offset	0x0000 00E4		
Physical Address	0x4A00 51E4	**Instance**	CKGEN_CM_CORE_AON
Description	This register allows monitoring DPLL activity. This register is read only and automatically updated. [warm reset insensitive]		

Table 3-339. CM_IDLEST_DPLL_ABE (continued)

Type	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																											ST_DPLL_INIT	ST_DPLL_MODE			ST_DPLL_CLK

Bits	Field Name	Description	Type	Reset
31:5	RESERVED		R	0x0
4	ST_DPLL_INIT	DPLL init status (for debug purpose). 0x0: DPLL is not init 0x1: DPLL has been init	R	0x0
3:1	ST_DPLL_MODE	DPLL mode status (for debug purpose). 0x0: Transient state. From reset to any LP idle state or from any power state to any power state (Power states are Low Power Stop mode, Fast Relock Stop mode, Idle Bypass Low Power mode and Idle Bypass Fast Relock mode). 0x1: The DPLL is in Low Power Stop mode. 0x2: The DPLL is in Fast Relock Stop mode. 0x3: Reserved 0x4: Reserved 0x5: The DPLL is in Idle Bypass Low Power mode. 0x6: The DPLL is in Idle Bypass Fast Relock mode. 0x7: Reserved	R	0x0
0	ST_DPLL_CLK	DPLL lock status 0x0: DPLL is either in bypass mode or in stop mode. 0x1: DPLL is LOCKED	R	0x0

Table 3-340. CM_AUTOIDLE_DPLL_ABE

Address Offset	0x0000 00E8		
Physical Address	0x4A00 51E8	Instance	CKGEN_CM_CORE_AON
Description	This register provides automatic control over the DPLL activity.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														AUTO_DPLL_MODE	

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0
2:0	AUTO_DPLL_MODE	DPLL automatic control; 0x0: DPLL auto control disabled 0x1: The DPLL is automatically put in Low Power Stop mode when its DPLL generated clocks are not required anymore. It is also restarted automatically. 0x2: The DPLL is automatically put in Fast Relock Stop mode when its DPLL generated clocks are not required anymore. It is also restarted automatically. 0x3: Reserved 0x4: Reserved 0x5: The DPLL is automatically put in Idle Bypass Low Power mode when its DPLL generated clocks are not required anymore. It is also restarted automatically. 0x6: The DPLL is automatically put in Idle Bypass Fast Relock mode when its DPLL generated clocks are not required anymore. It is also restarted automatically. 0x7: Reserved	RW	0x0

Table 3-341. CM_CLKSEL_DPLL_ABE

Address Offset	0x0000 00EC		
Physical Address	0x4A00 51EC	**Instance**	CKGEN_CM_CORE_AON
Description	This register provides controls over the DPLL.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DPLL_BYP_CLKSEL	DCC_EN	RESERVED			DPLL_MULT											RESERVED	DPLL_DIV						

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23	DPLL_BYP_CLKSEL	Only CLKINPULOW bypass clock supported for this PLL	R	0x1
22	DCC_EN	Duty-cycle corrector for high frequency clock 0x0: Duty-cycle corrector is disabled	R	0x0
21:19	RESERVED		R	0x0
18:8	DPLL_MULT	DPLL multiplier factor (2 to 2047). (equal to input M of DPLL; M=2 to 2047 = DPLL multiplies by M). [warm reset insensitive] 0x0: Reserved 0x1: Reserved	RW	0x0
7	RESERVED		R	0x0
6:0	DPLL_DIV	DPLL divider factor (0 to 127) (equal to input N of DPLL; actual division factor is N+1). [warm reset insensitive]	RW	0x0

Table 3-342. CM_DIV_M2_DPLL_ABE

Address Offset	0x0000 00F0		
Physical Address	0x4A00 51F0	**Instance**	CKGEN_CM_CORE_AON
Description	This register provides controls over the H12 divider of the DPLL_DDR.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																				CLKX2ST	RESERVED	CLKST	RESERVED					DIVHS			

Bits	Field Name	Description	Type	Reset
31:12	RESERVED		R	0x0
11	CLKX2ST	DPLL CLKOUTX2 status 0x0: The clock output is gated 0x1: The clock output is enabled	R	0x0
10	RESERVED		R	0x0
9	CLKST	DPLL CLKOUT status 0x0: The clock output is gated 0x1: The clock output is enabled	R	0x0
8:5	RESERVED		R	0x0
4:0	DIVHS	This field programs the H12 post-divider factor (1 to 31) of DPLL_DDR. 0x0: Reserved 0x1: H12 = /1 0x2: H12 = /2 ... 0x1F: H12 = /31	RW	0x1

Table 3-343. Register Call Summary for Register CM_DIV_M2_DPLL_ABE

Clock Management Functional Description
- DPLL_DDR Synthesized Clock Parameters: [0][1]

PRCM Register Manual
- CKGEN_CM_CORE_AON Register Summary: [2]

Table 3-344. CM_DIV_M3_DPLL_ABE

Address Offset	0x0000 00F4		
Physical Address	0x4A00 51F4	**Instance**	CKGEN_CM_CORE_AON
Description	This register provides controls over the M3 divider of the DPLL.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						CLKST	RESERVED					DIVHS			

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	CLKST	DPLL CLKOUTHIF status 0x0: The clock output is gated 0x1: The clock output is enabled	R	0x0
8:5	RESERVED		R	0x0
4:0	DIVHS	This field programs the M3 post-divider factor (1 to 31) of DPLL_ABE. 0x0: Reserved 0x1: M3 = /1 0x2: M3 = /2 ... 0x1F: M3 = /31	RW	0x1

Table 3-345. CM_SSC_DELTAMSTEP_DPLL_ABE

Address Offset	0x0000 0108		
Physical Address	0x4A00 5208	**Instance**	CKGEN_CM_CORE_AON
Description	Control the DeltaMStep parameter for Spread Spectrum Clocking. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DELTAMSTEP																							

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19:0	DELTAMSTEP	DeltaMStep is split into fractional and integer part. For Triangular Spread Spectrum: [19:18] for integer part, [17:0] for fractional part For Square Wave Spread Spectrum [19:14] for integer part, [13:0] for fractional part	RW	0x0

Table 3-346. CM_SSC_MODFREQDIV_DPLL_ABE

Address Offset	0x0000 010C		
Physical Address	0x4A00 520C	**Instance**	CKGEN_CM_CORE_AON
Description	Control the Modulation Frequency (Fm) for Spread Spectrum Clocking. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																					MODFREQDIV_EXPONENT			RESERVED	MODFREQDIV_MANTISSA						

Bits	Field Name	Description	Type	Reset
31:11	RESERVED		R	0x0
10:8	MODFREQDIV_EXPONENT	Set the Exponent component of MODFREQDIV factor	RW	0x0
7	RESERVED		R	0x0
6:0	MODFREQDIV_MANTISSA	Set the Mantissa component of MODFREQDIV factor	RW	0x0

Table 3-347. CM_CLKMODE_DPLL_DDR

Address Offset	0x0000 0110		
Physical Address	0x4A00 5210	**Instance**	CKGEN_CM_CORE_AON
Description	This register allows controlling the DPLL modes.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																DPLL_SSC_TYPE	DPLL_SSC_DOWNSPREAD	DPLL_SSC_ACK	DPLL_SSC_EN	DPLL_REGM4XEN	DPLL_LPMODE_EN	DPLL_RELOCK_RAMP_EN	DPLL_DRIFTGUARD_EN	DPLL_RAMP_RATE			DPLL_RAMP_LEVEL				DPLL_EN

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0
15	DPLL_SSC_TYPE	Select between Triangular and SquareWave Spread Spectrum Clocking 0x0: Triangular Spread Spectrum Clocking is selected 0x1: Square Wave Spread Spectrum Clocking is selected (only available under proper licensing agreement)	RW	0x0
14	DPLL_SSC_DOWNSPREAD	Control if only low frequency spread is required 0x0: When SSC is enabled, clock frequency is spread on both sides of the programmed frequency 0x1: When SSC is enabled, clock frequency is spread only on the lower side of the programmed frequency	RW	0x0
13	DPLL_SSC_ACK	Acknowledgement from the DPLL regarding start and stop of Spread Spectrum Clocking feature 0x0: SSC has been turned off on PLL o/ps 0x1: SSC has been turned on on PLL o/ps	R	0x0
12	DPLL_SSC_EN	Enable or disable Spread Spectrum Clocking 0x0: SSC disabled 0x1: SSC enabled	RW	0x0
11	DPLL_REGM4XEN	Enable the REGM4XEN mode of the DPLL. Please check the DPLL documentation to check when this mode can be enabled. 0x0: REGM4XEN mode of the DPLL is disabled	R	0x0
10	DPLL_LPMODE_EN	Set the DPLL in Low Power mode. Check the DPLL documentation to see when this can be enabled. 0x0: Low power mode of the DPLL is disabled 0x1: Low power mode of the DPLL is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
9	DPLL_RELOCK_RAMP_EN	If enabled, the clock ramping feature is used applied during the lock process, as well as the relock process. If disabled, the clock ramping feature is used only during the first lock.	RW	0x0
8	DPLL_DRIFTGUARD_EN	This bit allows to enable or disable the automatic recalibration feature of the DPLL. The DPLL will automatically start a recalibration process upon assertion of the DPLL's RECAL flag if this bit is set. 0x0: DRIFTGUARD feature is disabled 0x1: DRIFTGUARD feature is enabled	RW	0x0
7:5	DPLL_RAMP_RATE	Selects the time in terms of DPLL REFCLKs spent at each stage of the clock ramping process 0x0: 2 REFCLKs 0x1: 4 REFCLKs 0x2: 8 REFCLKs 0x3: 16 REFCLKs 0x4: 32 REFCLKs 0x5: 64 REFCLKs 0x6: 128 REFCLKs 0x7: 512 REFCLKs	RW	0x0
4:3	DPLL_RAMP_LEVEL	The DPLL provides an output clock frequency ramping feature when switching from bypass clock to normal clock during lock and re-lock. The frequency ramping will happen in a maximum of 4 steps in frequency before the DPLL's frequency lock indicator is asserted. This register is used to enable/disable the DPLL ramping feature. If enabled, it is also used to select the algorithm used for clock ramping 0x0: CLKOUT = No ramping CLKOUTX2 = No ramping 0x1: CLKOUT = Bypass clk - Fout/8 - Fout/4 - Fout/2 - Fout CLKOUTX2 = Bypass clk - Foutx2/8 - Foutx2/4 - Foutx2/2 - Foutx2 0x2: CLKOUT = Bypass clk - Fout/4 - Fout/2 - Fout/1.5 - Fout CLKOUTX2 = Bypass clk - Foutx2/4 - Foutx2/2 - Foutx2/1.5 - Foutx2 0x3: Reserved	RW	0x0
2:0	DPLL_EN	DPLL control. 0x0: Reserved 0x1: Reserved 0x2: Reserved 0x3: Reserved 0x4: Reserved 0x5: Put the DPLL in Idle Bypass Low Power mode. 0x6: Put the DPLL in Idle Bypass Fast Relock mode. 0x7: Enables the DPLL in Lock mode	RW	0x5

Table 3-348. Register Call Summary for Register CM_CLKMODE_DPLL_DDR

Clock Management Functional Description
- DPLL_DDR Power Modes: [0]
- DPLL_DDR Recalibration: [1]

PRCM Register Manual
- CKGEN_CM_CORE_AON Register Summary: [4]
- CKGEN_CM_CORE_AON Register Description: [5]

Table 3-349. CM_IDLEST_DPLL_DDR

Address Offset	0x0000 0114		
Physical Address	0x4A00 5214	**Instance**	CKGEN_CM_CORE_AON
Description	This register allows monitoring DPLL activity. This register is read only and automatically updated. [warm reset insensitive]		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																											ST_DPLL_INIT	ST_DPLL_MODE			ST_DPLL_CLK

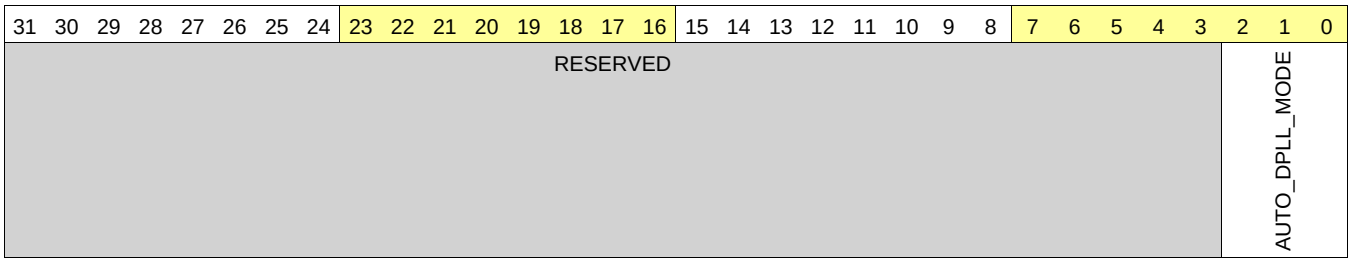
Bits	Field Name	Description	Type	Reset
31:5	RESERVED		R	0x0
4	ST_DPLL_INIT	DPLL init status (for debug purpose). 0x0: DPLL is not init 0x1: DPLL has been init	R	0x0
3:1	ST_DPLL_MODE	DPLL mode status (for debug purpose). 0x0: Transient state. From reset to any LP idle state or from any power state to any power state (Power states are Low Power Stop mode, Fast Relock Stop mode, Idle Bypass Low Power mode and Idle Bypass Fast Relock mode). 0x1: The DPLL is in Low Power Stop mode. 0x2: The DPLL is in Fast Relock Stop mode. 0x3: Reserved 0x4: Reserved 0x5: The DPLL is in Idle Bypass Low Power mode. 0x6: The DPLL is in Idle Bypass Fast Relock mode. 0x7: Reserved	R	0x0
0	ST_DPLL_CLK	DPLL lock status 0x0: DPLL is either in bypass mode or in stop mode. 0x1: DPLL is LOCKED	R	0x0

Table 3-350. Register Call Summary for Register CM_IDLEST_DPLL_DDR

PRCM Register Manual
- CKGEN_CM_CORE_AON Register Summary: [0]

Table 3-351. CM_AUTOIDLE_DPLL_DDR

Address Offset	0x0000 0118		
Physical Address	0x4A00 5218	**Instance**	CKGEN_CM_CORE_AON
Description	This register provides automatic control over the DPLL activity.		
Type	RW		



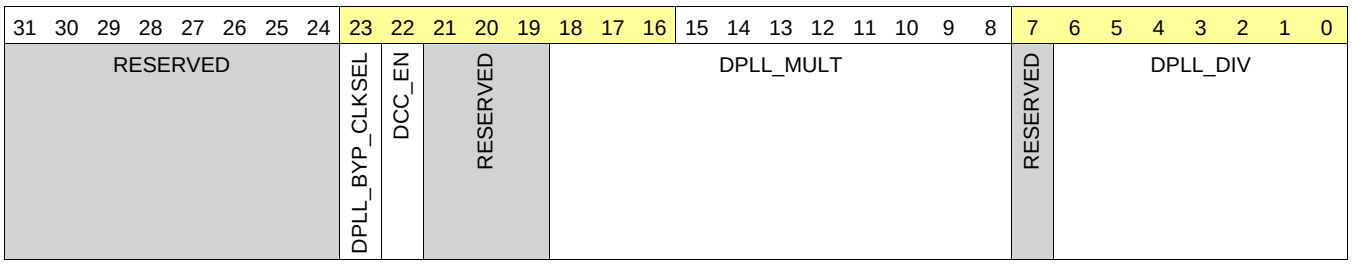
Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0
2:0	AUTO_DPLL_MODE	DPLL automatic control; 0x0: DPLL auto control disabled 0x1: The DPLL is automatically put in Low Power Stop mode when its DPLL generated clocks are not required anymore. It is also restarted automatically. 0x2: The DPLL is automatically put in Fast Relock Stop mode when its DPLL generated clocks are not required anymore. It is also restarted automatically. 0x3: Reserved 0x4: Reserved 0x5: The DPLL is automatically put in Idle Bypass Low Power mode when its DPLL generated clocks are not required anymore. It is also restarted automatically. 0x6: The DPLL is automatically put in Idle Bypass Fast Relock mode when its DPLL generated clocks are not required anymore. It is also restarted automatically. 0x7: Reserved	RW	0x0

Table 3-352. Register Call Summary for Register CM_AUTOIDLE_DPLL_DDR

Clock Management Functional Description • DPLL_DDR Power Modes: [0]
PRCM Register Manual • CKGEN_CM_CORE_AON Register Summary: [1]

Table 3-353. CM_CLKSEL_DPLL_DDR

Address Offset	0x0000 011C		
Physical Address	0x4A00 521C	**Instance**	CKGEN_CM_CORE_AON
Description	This register provides controls over the DPLL.		
Type	RW		



Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23	DPLL_BYP_CLKSEL	Allows control of the BYPASS clock of the PLL and the associated HSDIVIDER. Same as ULOWCLKEN on DPLL. In DPLL Locked mode, 0 - No impact 1 - No impact In DPLL Bypass mode, 0 - CLKINP is selected as the BYPASS clock for CLKOUT/CLKOUTX2 1 - CLKINPULOW is selected as the BYPASS clock for CLKOUT/CLKOUTX2	RW	0x0
22	DCC_EN	Duty-cycle corrector for high frequency clock 0x0: Duty-cycle corrector is disabled 0x1: Duty-cycle corrector is enabled	RW	0x0
21:19	RESERVED		R	0x0
18:8	DPLL_MULT	DPLL multiplier factor (2 to 2047). (equal to input M of DPLL; M=2 to 2047 = DPLL multiplies by M). [warm reset insensitive] 0x0: Reserved 0x1: Reserved	RW	0x0
7	RESERVED		R	0x0
6:0	DPLL_DIV	DPLL divider factor (0 to 127) (equal to input N of DPLL; actual division factor is N+1). [warm reset insensitive]	RW	0x0

Table 3-354. Register Call Summary for Register CM_CLKSEL_DPLL_DDR

Clock Management Functional Description
- DPLL_DDR Synthesized Clock Parameters: [0][1]

PRCM Register Manual
- CKGEN_CM_CORE_AON Register Summary: [2]

Table 3-355. CM_DIV_M2_DPLL_DDR

Address Offset	0x0000 0120		
Physical Address	0x4A00 5220	**Instance**	CKGEN_CM_CORE_AON
Description	This register provides controls over the M2 divider of DPLL_DDR.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						CLKST	RESERVED					DIVHS			

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	CLKST	DPLL CLKOUT status 0x0: The clock output is gated 0x1: The clock output is enabled	R	0x0
8:5	RESERVED		R	0x0
4:0	DIVHS	This field programs the M2 post-divider factor (1 to 31) of DPLL_DDR. 0x0: Reserved 0x1: M2 = /1 0x2: M2 = /2 ... 0x1F: M2 = /31	RW	0x1

Table 3-356. Register Call Summary for Register CM_DIV_M2_DPLL_DDR

Clock Management Functional Description • DPLL_DDR Synthesized Clock Parameters: [0][1]
PRCM Register Manual • CKGEN_CM_CORE_AON Register Summary: [2] • CKGEN_CM_CORE_AON Register Description: [3]

Table 3-357. CM_DIV_M3_DPLL_DDR

Address Offset	0x0000 0124		
Physical Address	0x4A00 5224	**Instance**	CKGEN_CM_CORE_AON
Description	This register provides controls over the M3 divider of DPLL_DDR.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						CLKST	RESERVED					DIVHS			

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	CLKST	DPLL CLKOUTHIF status 0x0: The clock output is gated 0x1: The clock output is enabled	R	0x0
8:5	RESERVED		R	0x0
4:0	DIVHS	This field programs the M3 post-divider factor (1 to 31) of DPLL_DDR. 0x0: Reserved 0x1: M3 = /1 0x2: M3 = /2 ... 0x1F: M3 = /31	RW	0x1

Table 3-358. CM_DIV_H11_DPLL_DDR

Address Offset	0x0000 0128		
Physical Address	0x4A00 5228	**Instance**	CKGEN_CM_CORE_AON
Description	This register provides controls over the CLKOUT1 o/p of the HSDIVIDER1 DPLL_DDR.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						CLKST	RESERVED					DIVHS			

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	CLKST	HSDIVIDER1 CLKOUT1 status 0x0: The clock output is gated 0x1: The clock output is enabled	R	0x0

Bits	Field Name	Description	Type	Reset
8:6	RESERVED		R	0x0
5:0	DIVHS	This field programs the H11 post-divider factor (1 to 63) of DPLL_DDR. 0x0: Reserved 0x1: H11 = /1 0x2: H11 = /2 ... 0x3F: H11 = /63	RW	0x1

Table 3-359. Register Call Summary for Register CM_DIV_H11_DPLL_DDR

Clock Management Functional Description • DPLL_DDR Synthesized Clock Parameters: [0][1]
PRCM Register Manual • CKGEN_CM_CORE_AON Register Summary: [2]

Table 3-360. CM_SSC_DELTAMSTEP_DPLL_DDR

Address Offset	0x0000 012C		
Physical Address	0x4A00 522C	**Instance**	CKGEN_CM_CORE_AON
Description	Control the DeltaMStep parameter for Spread Spectrum Clocking. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DELTAMSTEP																							

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19:0	DELTAMSTEP	DeltaMStep is split into fractional and integer part. For Triangular Spread Spectrum: [19:18] for integer part, [17:0] for fractional part For Square Wave Spread Spectrum [19:14] for integer part, [13:0] for fractional part	RW	0x0

Table 3-361. CM_SSC_MODFREQDIV_DPLL_DDR

Address Offset	0x0000 0130		
Physical Address	0x4A00 5230	**Instance**	CKGEN_CM_CORE_AON
Description	Control the Modulation Frequency (Fm) for Spread Spectrum Clocking. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																					MODFREQDIV_EXPONENT			RESERVED	MODFREQDIV_MANTISSA						

Bits	Field Name	Description	Type	Reset
31:11	RESERVED		R	0x0
10:8	MODFREQDIV_EXPONENT	Set the Exponent component of MODFREQDIV factor	RW	0x0
7	RESERVED		R	0x0
6:0	MODFREQDIV_MANTISSA	Set the Mantissa component of MODFREQDIV factor	RW	0x0

Table 3-362. CM_CLKMODE_DPLL_DSP

Address Offset	0x0000 0134		
Physical Address	0x4A00 5234	**Instance**	CKGEN_CM_CORE_AON
Description	This register allows controlling the DPLL modes.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																DPLL_SSC_TYPE	DPLL_SSC_DOWNSPREAD	DPLL_SSC_ACK	DPLL_SSC_EN	DPLL_REGM4XEN	DPLL_LPMODE_EN	DPLL_RELOCK_RAMP_EN	DPLL_DRIFTGUARD_EN	DPLL_RAMP_RATE			DPLL_RAMP_LEVEL				DPLL_EN

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0
15	DPLL_SSC_TYPE	Select between Triangular and SquareWave Spread Spectrum Clocking 0x0: Triangular Spread Spectrum Clocking is selected 0x1: Square Wave Spread Spectrum Clocking is selected (only available under proper licensing agreement)	RW	0x0
14	DPLL_SSC_DOWNSPREAD	Control if only low frequency spread is required 0x0: When SSC is enabled, clock frequency is spread on both sides of the programmed frequency 0x1: When SSC is enabled, clock frequency is spread only on the lower side of the programmed frequency	RW	0x0
13	DPLL_SSC_ACK	Acknowledgement from the DPLL regarding start and stop of Spread Spectrum Clocking feature 0x0: SSC has been turned off on PLL o/ps 0x1: SSC has been turned on on PLL o/ps	R	0x0
12	DPLL_SSC_EN	Enable or disable Spread Spectrum Clocking 0x0: SSC disabled 0x1: SSC enabled	RW	0x0
11	DPLL_REGM4XEN	Enable the REGM4XEN mode of the DPLL. Please check the DPLL documentation to check when this mode can be enabled. 0x0: REGM4XEN mode of the DPLL is disabled	R	0x0
10	DPLL_LPMODE_EN	Set the DPLL in Low Power mode. Check the DPLL documentation to see when this can be enabled. 0x0: Low power mode of the DPLL is disabled 0x1: Low power mode of the DPLL is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
9	DPLL_RELOCK_RAMP_EN	If enabled, the clock ramping feature is used applied during the lock process, as well as the relock process. If disabled, the clock ramping feature is used only during the first lock.	RW	0x0
8	DPLL_DRIFTGUARD_EN	This bit allows to enable or disable the automatic recalibration feature of the DPLL. The DPLL will automatically start a recalibration process upon assertion of the DPLL's RECAL flag if this bit is set. 0x0: DRIFTGUARD feature is disabled 0x1: DRIFTGUARD feature is enabled	RW	0x0
7:5	DPLL_RAMP_RATE	Selects the time in terms of DPLL REFCLKs spent at each stage of the clock ramping process 0x0: 2 REFCLKs 0x1: 4 REFCLKs 0x2: 8 REFCLKs 0x3: 16 REFCLKs 0x4: 32 REFCLKs 0x5: 64 REFCLKs 0x6: 128 REFCLKs 0x7: 512 REFCLKs	RW	0x0
4:3	DPLL_RAMP_LEVEL	The DPLL provides an output clock frequency ramping feature when switching from bypass clock to normal clock during lock and re-lock. The frequency ramping will happen in a maximum of 4 steps in frequency before the DPLL's frequency lock indicator is asserted. This register is used to enable/disable the DPLL ramping feature. If enabled, it is also used to select the algorithm used for clock ramping 0x0: CLKOUT = No ramping CLKOUTX2 = No ramping 0x1: CLKOUT = Bypass clk - Fout/8 - Fout/4 - Fout/2 - Fout CLKOUTX2 = Bypass clk - Foutx2/8 - Foutx2/4 - Foutx2/2 - Foutx2 0x2: CLKOUT = Bypass clk - Fout/4 - Fout/2 - Fout/1.5 - Fout CLKOUTX2 = Bypass clk - Foutx2/4 - Foutx2/2 - Foutx2/1.5 - Foutx2 0x3: Reserved	RW	0x0
2:0	DPLL_EN	DPLL control. 0x0: Reserved 0x1: Reserved 0x2: Reserved 0x3: Reserved 0x4: Reserved 0x5: Put the DPLL in Idle Bypass Low Power mode. 0x6: Put the DPLL in Idle Bypass Fast Relock mode. 0x7: Enables the DPLL in Lock mode	RW	0x5

Table 3-363. Register Call Summary for Register CM_CLKMODE_DPLL_DSP

Clock Management Functional Description
- DPLL_EVE_VID_DSP Power Modes: [0]
- DPLL_EVE_VID_DSP Recalibration: [1]

PRCM Register Manual
- CKGEN_CM_CORE_AON Register Summary: [4]

Table 3-364. CM_IDLEST_DPLL_DSP

Address Offset	0x0000 0138		
Physical Address	0x4A00 5238	**Instance**	CKGEN_CM_CORE_AON
Description	This register allows monitoring DPLL activity. This register is read only and automatically updated. [warm reset insensitive]		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																											ST_DPLL_INIT	ST_DPLL_MODE			ST_DPLL_CLK

Bits	Field Name	Description	Type	Reset
31:5	RESERVED		R	0x0
4	ST_DPLL_INIT	DPLL init status (for debug purpose) 0x0: DPLL is not init 0x1: DPLL has been init	R	0x0
3:1	ST_DPLL_MODE	DPLL mode status (for debug purpose). 0x0: Transient state. From reset to any LP idle state or from any power state to any power state (Power states are Low Power Stop mode, Fast Relock Stop mode, Idle Bypass Low Power mode and Idle Bypass Fast Relock mode). 0x1: The DPLL is in Low Power Stop mode. 0x2: The DPLL is in Fast Relock Stop mode. 0x3: Reserved 0x4: Reserved 0x5: The DPLL is in Idle Bypass Low Power mode. 0x6: The DPLL is in Idle Bypass Fast Relock mode. 0x7: Reserved	R	0x0
0	ST_DPLL_CLK	DPLL lock status 0x0: DPLL is either in bypass mode or in stop mode. 0x1: DPLL is LOCKED	R	0x0

Table 3-365. Register Call Summary for Register CM_IDLEST_DPLL_DSP

PRCM Register Manual
- CKGEN_CM_CORE_AON Register Summary: [0]

Table 3-366. CM_AUTOIDLE_DPLL_DSP

Address Offset	0x0000 013C		
Physical Address	0x4A00 523C	**Instance**	CKGEN_CM_CORE_AON
Description	This register provides automatic control over the DPLL activity.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														AUTO_DPLL_MODE	

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0
2:0	AUTO_DPLL_MODE	DPLL automatic control; 0x0: DPLL auto control disabled 0x1: The DPLL is automatically put in Low Power Stop mode when its DPLL generated clocks are not required anymore. It is also restarted automatically. 0x2: The DPLL is automatically put in Fast Relock Stop mode when its DPLL generated clocks are not required anymore. It is also restarted automatically. 0x3: Reserved 0x4: Reserved 0x5: The DPLL is automatically put in Idle Bypass Low Power mode when its DPLL generated clocks are not required anymore. It is also restarted automatically. 0x6: The DPLL is automatically put in Idle Bypass Fast Relock mode when its DPLL generated clocks are not required anymore. It is also restarted automatically. 0x7: Reserved	RW	0x0

Table 3-367. Register Call Summary for Register CM_AUTOIDLE_DPLL_DSP

Clock Management Functional Description
- DPLL_EVE_VID_DSP Power Modes: [0]

PRCM Register Manual
- CKGEN_CM_CORE_AON Register Summary: [1]

Table 3-368. CM_CLKSEL_DPLL_DSP

Address Offset	0x0000 0140		
Physical Address	0x4A00 5240	**Instance**	CKGEN_CM_CORE_AON
Description	This register provides controls over the DPLL.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DPLL_BYP_CLKSEL	DCC_EN	RESERVED						DPLL_MULT								RESERVED	DPLL_DIV						

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23	DPLL_BYP_CLKSEL	Allows control of the BYPASS clock of the PLL and the associated HSDIVIDER. Same as ULOWCLKEN on DPLL. In DPLL Locked mode, 0 - No impact 1 - No impact In DPLL Bypass mode, 0 - CLKINP is selected as the BYPASS clock for CLKOUT/CLKOUTX2 1 - CLKINPULOW is selected as the BYPASS clock for CLKOUT/CLKOUTX2	RW	0x0
22	DCC_EN	Duty-cycle corrector for high frequency clock 0x0: Duty-cycle corrector is disabled	R	0x0
21:19	RESERVED		R	0x0
18:8	DPLL_MULT	DPLL multiplier factor (2 to 2047). (equal to input M of DPLL; M=2 to 2047 = DPLL multiplies by M). [warm reset insensitive] 0x0: Reserved 0x1: Reserved	RW	0x0
7	RESERVED		R	0x0
6:0	DPLL_DIV	DPLL divider factor (0 to 127) (equal to input N of DPLL; actual division factor is N+1). [warm reset insensitive]	RW	0x0

Table 3-369. Register Call Summary for Register CM_CLKSEL_DPLL_DSP

Clock Management Functional Description
- DPLL_EVE_VID_DSP Synthesized Clock Parameters: [0][1]

PRCM Register Manual
- CKGEN_CM_CORE_AON Register Summary: [2]

Table 3-370. CM_DIV_M2_DPLL_DSP

Address Offset	0x0000 0144		
Physical Address	0x4A00 5244	**Instance**	CKGEN_CM_CORE_AON
Description	This register provides controls over the M3 divider of DPLL_EVE_VID_DSP.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						CLKST	RESERVED							DIVHS	

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	CLKST	DPLL CLKOUT status 0x0: The clock output is gated 0x1: The clock output is enabled	R	0x0
8:5	RESERVED		R	0x0
4:0	DIVHS	This field programs the M3 post-divider factor (1 to 31) of DPLL_EVE_VID_DSP. 0x0: Reserved 0x1: M3 = /1 0x2: M3 = /2 ... 0x1F: M3 = /31	RW	0x1

Table 3-371. Register Call Summary for Register CM_DIV_M2_DPLL_DSP

Clock Management Functional Description
• DPLL_EVE_VID_DSP Synthesized Clock Parameters: [0][1]
PRCM Register Manual
• CKGEN_CM_CORE_AON Register Summary: [2]

Table 3-372. CM_DIV_M3_DPLL_DSP

Address Offset	0x0000 0148		
Physical Address	0x4A00 5248	**Instance**	CKGEN_CM_CORE_AON
Description	This register provides controls over the M3 divider of DPLL_CORE.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						CLKST	RESERVED					DIVHS			

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	CLKST	DPLL CLKOUTHIF status 0x0: The clock output is gated 0x1: The clock output is enabled	R	0x0
8:5	RESERVED		R	0x0
4:0	DIVHS	This field programs the M3 post-divider factor (1 to 31) of DPLL_CORE. 0x0: Reserved 0x1: M3 = /1 0x2: M3 = /2 ... 0x1F: M3 = /31	RW	0x1

Table 3-373. Register Call Summary for Register CM_DIV_M3_DPLL_DSP

Clock Management Functional Description
• DPLL_CORE Synthesized Clock Parameters: [0][1]
PRCM Register Manual
• CKGEN_CM_CORE_AON Register Summary: [2]

Table 3-374. CM_SSC_DELTAMSTEP_DPLL_DSP

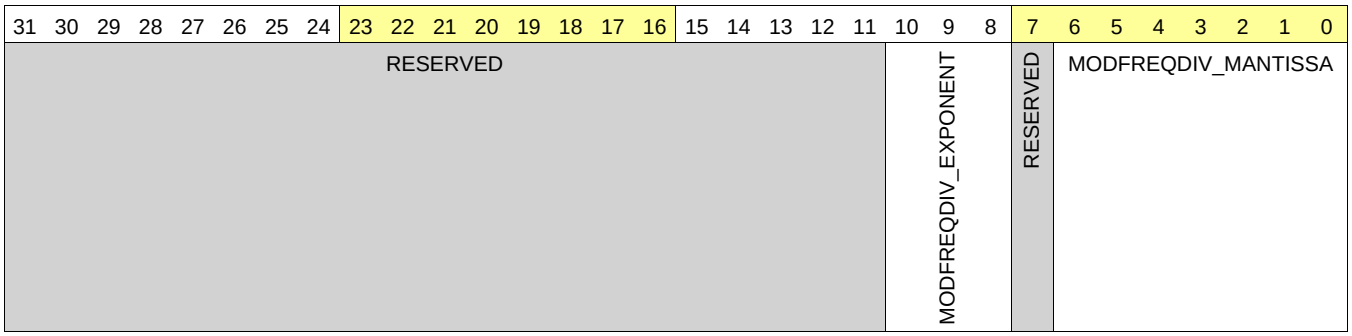
Address Offset	0x0000 014C		
Physical Address	0x4A00 524C	**Instance**	CKGEN_CM_CORE_AON
Description	Control the DeltaMStep parameter for Spread Spectrum Clocking. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																DELTAMSTEP															

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19:0	DELTAMSTEP	DeltaMStep is split into fractional and integer part. For Triangular Spread Spectrum: [19:18] for integer part, [17:0] for fractional part For Square Wave Spread Spectrum [19:14] for integer part, [13:0] for fractional part	RW	0x0

Table 3-375. CM_SSC_MODFREQDIV_DPLL_DSP

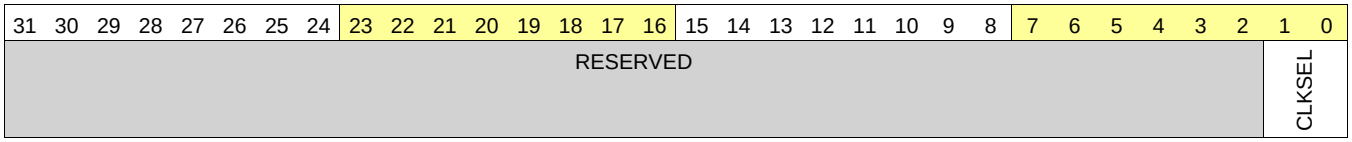
Address Offset	0x0000 0150		
Physical Address	0x4A00 5250	**Instance**	CKGEN_CM_CORE_AON
Description	Control the Modulation Frequency (Fm) for Spread Spectrum Clocking. [warm reset insensitive]		
Type	RW		



Bits	Field Name	Description	Type	Reset
31:11	RESERVED		R	0x0
10:8	MODFREQDIV_EXPONENT	Set the Exponent component of MODFREQDIV factor	RW	0x0
7	RESERVED		R	0x0
6:0	MODFREQDIV_MANTISSA	Set the Mantissa component of MODFREQDIV factor	RW	0x0

Table 3-376. CM_BYPCLK_DPLL_DSP

Address Offset	0x0000 0154		
Physical Address	0x4A00 5254	**Instance**	CKGEN_CM_CORE_AON
Description	Control IVA PLL BYPASS clock. [warm reset insensitive]		
Type	RW		



Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1:0	CLKSEL	Select the DPLL IVA bypass clock 0x0: DPLL_IVA bypass clock is CORE_X2_CLK divided by 1 0x1: DPLL_IVA bypass clock is CORE_X2_CLK divided by 2 0x2: DPLL_IVA bypass clock is CORE_X2_CLK divided by 4 0x3: DPLL_IVA bypass clock is CORE_X2_CLK divided by 8	RW	0x0

Table 3-377. Register Call Summary for Register CM_BYPCLK_DPLL_DSP

Clock Management Functional Description • CM_CORE_AON Clock Generator: [0]
PRCM Register Manual • CKGEN_CM_CORE_AON Register Summary: [1]

Table 3-378. CM_SHADOW_FREQ_CONFIG1

Address Offset	0x0000 0160		
Physical Address	0x4A00 5260	**Instance**	CKGEN_CM_CORE_AON
Description	Shadow register to program new DPLL configuration affecting EMIF and GPMC (L3 clock) functional frequency during DVFS. The PRCM h/w automatically applies the new configuration after EMIF/GPMC have been put in idle state.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED													DPLL_DDR_DPLL_EN			DPLL_DDR_M2_DIV								RESERVED				DLL_RESET	DLL_OVERRIDE	RESERVED	FREQ_UPDATE

Bits	Field Name	Description	Type	Reset
31:19	RESERVED		R	0x0
18:16	DPLL_DDR_DPLL_EN	Shadow register for CM_CLKMODE_DPLL_DDR.DPLL_EN. The main register is automatically loaded with the shadow register value after EMIF IDLE if the FREQ_UPDATE field is set to '1'. 0x0: Reserved 0x1: Reserved 0x2: Reserved 0x3: Reserved 0x4: Reserved 0x5: Put the DPLL in Idle Bypass Low Power mode. 0x6: Put the DPLL in Idle Bypass Fast Relock mode. 0x7: Enables the DPLL in Lock mode	RW	0x5
15:11	DPLL_DDR_M2_DIV	Shadow register for CM_DIV_M2_DPLL_DDR.DIVHS. The main register is automatically loaded with the shadow register value after EMIF IDLE if the FREQ_UPDATE field is set to '1'. Divide value from 1 to 31. 0x0: Reserved	RW	0x1
10:4	RESERVED		R	0x0
3	DLL_RESET	Specify if DLL should be reset or not during the frequency change hardware sequence. 0x0: DLL is not reset during the frequency change hardware sequence 0x1: DLL is reset automatically during the frequency change hardware sequence	RW	0x1

Bits	Field Name	Description	Type	Reset
2	DLL_OVERRIDE	Shadow register for CM_DLL_CTRL.DLL_OVERRIDE.The main register is automatically loaded with the shadow register value after EMIF IDLE if the FREQ_UPDATE field is set to '1'. 0x0: Lock and code outputs are not overriden 0x1: Lock output is overriden to '1' and code output is overriden with a value coming from control module.	RW	0x1
1	RESERVED		R	0x0
0	FREQ_UPDATE	Writing '1' indicates that a new configuration is available. It is automatically cleared by h/w after the configuration has been applied.	RW	0x0

Table 3-379. Register Call Summary for Register CM_SHADOW_FREQ_CONFIG1

Device Low-Power States
- Wakeup Upon Global Warm Reset: [0]

PRCM Register Manual
- CKGEN_CM_CORE_AON Register Summary: [1]
- CKGEN_CM_CORE_AON Register Description: [2][3]
- RESTORE_CM_CORE_AON Register Description: [4][5]

Table 3-380. CM_SHADOW_FREQ_CONFIG2

Address Offset	0x0000 0164		
Physical Address	0x4A00 5264	**Instance**	CKGEN_CM_CORE_AON
Description	Shadow register to program new DPLL configuration affecting GPMC (L3 clock) functional frequency during DVFS. The PRCM h/w automatically applies the new configuration after EMIF/GPMC have been put in idle state.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								DPLL_CORE_H12_DIV						CLKSEL_L3	GPMC_FREQ_UPDATE

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x0
7:2	DPLL_CORE_H12_DIV	Shadow register for CM_DIV_H12_DPLL_CORE.DIVHS. The main register is automatically loaded with the shadow register value after GPMC IDLE if the CM_SHADOW_FREQ_CONFIG1.FREQ_UPDATE field is set to '1' and GPMC_FREQ_UPDATE is set to '1'. Divide value from 1 to 31. 0x0: Reserved	RW	0x1
1	CLKSEL_L3	Shadow register for CM_CLKSEL_CORE.CLKSEL_L3. The main register is automatically loaded with the shadow register value after GPMC IDLE if the CM_SHADOW_FREQ_CONFIG1.FREQ_UPDATE field is set to '1' and GPMC_FREQ_UPDATE is set to '1'. 0x0: L3_CLK is CORE_CLK divided by 1 0x1: L3_CLK is CORE_CLK divided by 2	RW	0x0

Bits	Field Name	Description	Type	Reset
0	GPMC_FREQ_UPDATE	Controls whether or not GPMC has to be put automatically into idle during the frequency change operation. 0x0: GPMC is not put automatically into idle during frequency change operation. 0x1: GPMC is put automatically into idle during frequency change operation.	RW	0x0

Table 3-381. Register Call Summary for Register CM_SHADOW_FREQ_CONFIG2

Voltage-Management Functional Description
PRCM Register Manual • CKGEN_CM_CORE_AON Register Summary: [1] • RESTORE_CM_CORE_AON Register Description: [2][3]

Table 3-382. CM_DYN_DEP_PRESCAL

Address Offset	0x0000 0170		
Physical Address	0x4A00 5270	**Instance**	CKGEN_CM_CORE_AON
Description	Control the time unit of the sliding window for dynamic dependencies (auto-sleep feature).		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								PRESCAL							

Bits	Field Name	Description	Type	Reset
31:6	RESERVED		R	0x0
5:0	PRESCAL	Time unit is equal to (PRESCAL + 1) L4 clock cycles.	RW	0x20

Table 3-383. Register Call Summary for Register CM_DYN_DEP_PRESCAL

Device Power Management Introduction • Clock Domain Dependency: [0]
PRCM Register Manual • CKGEN_CM_CORE_AON Register Summary: [1] • DSP1_CM_CORE_AON Register Description: [2] • DSP2_CM_CORE_AON Register Description: [3] • IPU_CM_CORE_AON Register Description: [4] • MPU_CM_CORE_AON Register Description: [5] • RESTORE_CM_CORE_AON Register Description: [6] • CORE_CM_CORE Register Description: [7][8][9] • L4PER_CM_CORE Register Description: [10][11][12] • EMU_CM Register Description: [13]

Table 3-384. CM_RESTORE_ST

Address Offset	0x0000 0180		
Physical Address	0x4A00 5280	**Instance**	CKGEN_CM_CORE_AON
Description	Automatic restore status. This register is used by the system DMA to write a predefined value at the end of each automatic restore phase. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																													PHASE2B_COMPLETED	PHASE2A_COMPLETED	PHASE1_COMPLETED

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0
2	PHASE2B_COMPLETED	Indicates if restore phase 2b is completed.	RW	0x0
1	PHASE2A_COMPLETED	Indicates if restore phase 2a is completed.	RW	0x0
0	PHASE1_COMPLETED	Indicates if restore phase 1 is completed.	RW	0x0

Table 3-385. CM_CLKMODE_DPLL_EVE

Address Offset	0x0000 0184		
Physical Address	0x4A00 5284	**Instance**	CKGEN_CM_CORE_AON
Description	This register allows controlling the DPLL modes.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																DPLL_SSC_TYPE	DPLL_SSC_DOWNSPREAD	DPLL_SSC_ACK	DPLL_SSC_EN	DPLL_REGM4XEN	DPLL_LPMODE_EN	DPLL_RELOCK_RAMP_EN	DPLL_DRIFTGUARD_EN	DPLL_RAMP_RATE			DPLL_RAMP_LEVEL		DPLL_EN		

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0
15	DPLL_SSC_TYPE	Select between Triangular and SquareWave Spread Spectrum Clocking 0x0: Triangular Spread Spectrum Clocking is selected 0x1: Square Wave Spread Spectrum Clocking is selected (only available under proper licensing agreement)	RW	0x0
14	DPLL_SSC_DOWNSPREAD	Control if only low frequency spread is required 0x0: When SSC is enabled, clock frequency is spread on both sides of the programmed frequency 0x1: When SSC is enabled, clock frequency is spread only on the lower side of the programmed frequency	RW	0x0
13	DPLL_SSC_ACK	Acknowledgement from the DPLL regarding start and stop of Spread Spectrum Clocking feature 0x0: SSC has been turned off on PLL o/ps 0x1: SSC has been turned on on PLL o/ps	R	0x0
12	DPLL_SSC_EN	Enable or disable Spread Spectrum Clocking 0x0: SSC disabled 0x1: SSC enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
11	DPLL_REGM4XEN	Enable the REGM4XEN mode of the DPLL. Please check the DPLL documentation to check when this mode can be enabled. 0x0: REGM4XEN mode of the DPLL is disabled	R	0x0
10	DPLL_LPMODE_EN	Set the DPLL in Low Power mode. Check the DPLL documentation to see when this can be enabled. 0x0: Low power mode of the DPLL is disabled 0x1: Low power mode of the DPLL is enabled	RW	0x0
9	DPLL_RELOCK_RAMP_EN	If enabled, the clock ramping feature is used applied during the lock process, as well as the relock process. If disabled, the clock ramping feature is used only during the first lock.	RW	0x0
8	DPLL_DRIFTGUARD_EN	This bit allows to enable or disable the automatic recalibration feature of the DPLL. The DPLL will automatically start a recalibration process upon assertion of the DPLL's RECAL flag if this bit is set. 0x0: DRIFTGUARD feature is disabled 0x1: DRIFTGUARD feature is enabled	RW	0x0
7:5	DPLL_RAMP_RATE	Selects the time in terms of DPLL REFCLKs spent at each stage of the clock ramping process 0x0: 2 REFCLKs 0x1: 4 REFCLKs 0x2: 8 REFCLKs 0x3: 16 REFCLKs 0x4: 32 REFCLKs 0x5: 64 REFCLKs 0x6: 128 REFCLKs 0x7: 512 REFCLKs	RW	0x0
4:3	DPLL_RAMP_LEVEL	The DPLL provides an output clock frequency ramping feature when switching from bypass clock to normal clock during lock and re-lock. The frequency ramping will happen in a maximum of 4 steps in frequency before the DPLL's frequency lock indicator is asserted. This register is used to enable/disable the DPLL ramping feature. If enabled, it is also used to select the algorithm used for clock ramping 0x0: CLKOUT = No ramping CLKOUTX2 = No ramping 0x1: CLKOUT = Bypass clk - Fout/8 - Fout/4 - Fout/2 - Fout CLKOUTX2 = Bypass clk - Foutx2/8 - Foutx2/4 - Foutx2/2 - Foutx2 0x2: CLKOUT = Bypass clk - Fout/4 - Fout/2 - Fout/1.5 - Fout CLKOUTX2 = Bypass clk - Foutx2/4 - Foutx2/2 - Foutx2/1.5 - Foutx2 0x3: Reserved	RW	0x0
2:0	DPLL_EN	DPLL control. 0x0: Reserved 0x1: Reserved 0x2: Reserved 0x3: Reserved 0x4: Reserved 0x5: Put the DPLL in Idle Bypass Low Power mode. 0x6: Put the DPLL in Idle Bypass Fast Relock mode. 0x7: Enables the DPLL in Lock mode	RW	0x5

Table 3-386. CM_IDLEST_DPLL_EVE

Address Offset	0x0000 0188		
Physical Address	0x4A00 5288	**Instance**	CKGEN_CM_CORE_AON
Description	This register allows monitoring DPLL activity. This register is read only and automatically updated. [warm reset insensitive]		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																											ST_DPLL_INIT	ST_DPLL_MODE			ST_DPLL_CLK

Bits	Field Name	Description	Type	Reset
31:5	RESERVED		R	0x0
4	ST_DPLL_INIT	DPLL init status (for debug purpose) 0x0: DPLL is not init 0x1: DPLL has been init	R	0x0
3:1	ST_DPLL_MODE	DPLL mode status (for debug purpose). 0x0: Transient state. From reset to any LP idle state or from any power state to any power state (Power states are Low Power Stop mode, Fast Relock Stop mode, Idle Bypass Low Power mode and Idle Bypass Fast Relock mode). 0x1: The DPLL is in Low Power Stop mode. 0x2: The DPLL is in Fast Relock Stop mode. 0x3: Reserved 0x4: Reserved 0x5: The DPLL is in Idle Bypass Low Power mode. 0x6: The DPLL is in Idle Bypass Fast Relock mode. 0x7: Reserved	R	0x0
0	ST_DPLL_CLK	DPLL lock status 0x0: DPLL is either in bypass mode or in stop mode. 0x1: DPLL is LOCKED	R	0x0

Table 3-387. CM_AUTOIDLE_DPLL_EVE

Address Offset	0x0000 018C		
Physical Address	0x4A00 528C	**Instance**	CKGEN_CM_CORE_AON
Description	This register provides automatic control over the DPLL activity.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														AUTO_DPLL_MODE	

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0
2:0	AUTO_DPLL_MODE	DPLL automatic control; 0x0: DPLL auto control disabled 0x1: The DPLL is automatically put in Low Power Stop mode when its DPLL generated clocks are not required anymore. It is also restarted automatically. 0x2: The DPLL is automatically put in Fast Relock Stop mode when its DPLL generated clocks are not required anymore. It is also restarted automatically. 0x3: Reserved 0x4: Reserved 0x5: The DPLL is automatically put in Idle Bypass Low Power mode when its DPLL generated clocks are not required anymore. It is also restarted automatically. 0x6: The DPLL is automatically put in Idle Bypass Fast Relock mode when its DPLL generated clocks are not required anymore. It is also restarted automatically. 0x7: Reserved	RW	0x0

Table 3-388. CM_CLKSEL_DPLL_EVE

Address Offset	0x0000 0190		
Physical Address	0x4A00 5290	**Instance**	CKGEN_CM_CORE_AON
Description	This register provides controls over the DPLL.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DPLL_BYP_CLKSEL	DCC_EN	RESERVED						DPLL_MULT								RESERVED	DPLL_DIV						

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23	DPLL_BYP_CLKSEL	Allows control of the BYPASS clock of the PLL and the associated HSDIVIDER. Same as ULOWCLKEN on DPLL. In DPLL Locked mode, 0 - No impact 1 - No impact In DPLL Bypass mode, 0 - CLKINP is selected as the BYPASS clock for CLKOUT/CLKOUTX2 1 - CLKINPULOW is selected as the BYPASS clock for CLKOUT/CLKOUTX2	RW	0x0
22	DCC_EN	Duty-cycle corrector for high frequency clock 0x0: Duty-cycle corrector is disabled	R	0x0
21:19	RESERVED		R	0x0
18:8	DPLL_MULT	DPLL multiplier factor (2 to 2047). (equal to input M of DPLL; M=2 to 2047 = DPLL multiplies by M). [warm reset insensitive] 0x0: Reserved 0x1: Reserved	RW	0x0
7	RESERVED		R	0x0
6:0	DPLL_DIV	DPLL divider factor (0 to 127) (equal to input N of DPLL; actual division factor is N+1). [warm reset insensitive]	RW	0x0

Table 3-389. CM_DIV_M2_DPLL_EVE

Address Offset	0x0000 0194		
Physical Address	0x4A00 5294	**Instance**	CKGEN_CM_CORE_AON
Description	This register provides controls over the M2 divider of the DPLL.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						CLKST	RESERVED					DIVHS			

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	CLKST	DPLL CLKOUT status 0x0: The clock output is gated 0x1: The clock output is enabled	R	0x0
8:5	RESERVED		R	0x0
4:0	DIVHS	This field programs the M2 post-divider factor (1 to 31) of DPLL_EVE_VID_DSP. 0x0: Reserved 0x1: M2 = /1 0x2: M2 = /2 **...** 0x1F: M2 = /31	RW	0x1

Table 3-390. CM_DIV_M3_DPLL_EVE

Address Offset	0x0000 0198		
Physical Address	0x4A00 5298	**Instance**	CKGEN_CM_CORE_AON
Description	This register provides controls over the M3 divider of the DPLL.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						CLKST	RESERVED					DIVHS			

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	CLKST	DPLL CLKOUTHIF status 0x0: The clock output is gated 0x1: The clock output is enabled	R	0x0
8:5	RESERVED		R	0x0
4:0	DIVHS	This field programs the M3 post-divider factor (1 to 31) of DPLL_EVE_VID_DSP. 0x0: Reserved 0x1: M3 = /1 0x2: M3 = /2 **...** 0x1F: M3 = /31	RW	0x1

Table 3-391. CM_SSC_DELTAMSTEP_DPLL_EVE

Address Offset	0x0000 019C		
Physical Address	0x4A00 529C	**Instance**	CKGEN_CM_CORE_AON
Description	Control the DeltaMStep parameter for Spread Spectrum Clocking. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																DELTAMSTEP															

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19:0	DELTAMSTEP	DeltaMStep is split into fractional and integer part. For Triangular Spread Spectrum: [19:18] for integer part, [17:0] for fractional part For Square Wave Spread Spectrum [19:14] for integer part, [13:0] for fractional part	RW	0x0

Table 3-392. CM_SSC_MODFREQDIV_DPLL_EVE

Address Offset	0x0000 01A0		
Physical Address	0x4A00 52A0	**Instance**	CKGEN_CM_CORE_AON
Description	Control the Modulation Frequency (Fm) for Spread Spectrum Clocking. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																					MODFREQDIV_EXPONENT			RESERVED	MODFREQDIV_MANTISSA						

Bits	Field Name	Description	Type	Reset
31:11	RESERVED		R	0x0
10:8	MODFREQDIV_EXPONENT	Set the Exponent component of MODFREQDIV factor	RW	0x0
7	RESERVED		R	0x0
6:0	MODFREQDIV_MANTISSA	Set the Mantissa component of MODFREQDIV factor	RW	0x0

Table 3-393. CM_BYPCLK_DPLL_EVE

Address Offset	0x0000 01A4		
Physical Address	0x4A00 52A4	**Instance**	CKGEN_CM_CORE_AON
Description	Control IVA PLL BYPASS clock. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														CLKSEL	

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1:0	CLKSEL	Select the DPLL IVA bypass clock 0x0: DPLL_IVA bypass clock is CORE_X2_CLK divided by 1 0x1: DPLL_IVA bypass clock is CORE_X2_CLK divided by 2 0x2: DPLL_IVA bypass clock is CORE_X2_CLK divided by 4 0x3: DPLL_IVA bypass clock is CORE_X2_CLK divided by 8	RW	0x0

Table 3-394. CM_CLKMODE_DPLL_GMAC

Address Offset	0x0000 01A8		
Physical Address	0x4A00 52A8	**Instance**	CKGEN_CM_CORE_AON
Description	This register allows controlling the DPLL modes.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																DPLL_SSC_TYPE	DPLL_SSC_DOWNSPREAD	DPLL_SSC_ACK	DPLL_SSC_EN	DPLL_REGM4XEN	DPLL_LPMODE_EN	DPLL_RELOCK_RAMP_EN	DPLL_DRIFTGUARD_EN	DPLL_RAMP_RATE			DPLL_RAMP_LEVEL				DPLL_EN

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0
15	DPLL_SSC_TYPE	Select between Triangular and SquareWave Spread Spectrum Clocking 0x0: Triangular Spread Spectrum Clocking is selected 0x1: Square Wave Spread Spectrum Clocking is selected (only available under proper licensing agreement)	RW	0x0
14	DPLL_SSC_DOWNSPREAD	Control if only low frequency spread is required 0x0: When SSC is enabled, clock frequency is spread on both sides of the programmed frequency 0x1: When SSC is enabled, clock frequency is spread only on the lower side of the programmed frequency	RW	0x0
13	DPLL_SSC_ACK	Acknowledgement from the DPLL regarding start and stop of Spread Spectrum Clocking feature 0x0: SSC has been turned off on PLL o/ps 0x1: SSC has been turned on on PLL o/ps	R	0x0
12	DPLL_SSC_EN	Enable or disable Spread Spectrum Clocking 0x0: SSC disabled 0x1: SSC enabled	RW	0x0
11	DPLL_REGM4XEN	Enable the REGM4XEN mode of the DPLL. Please check the DPLL documentation to check when this mode can be enabled. 0x0: REGM4XEN mode of the DPLL is disabled	R	0x0

Bits	Field Name	Description	Type	Reset
10	DPLL_LPMODE_EN	Set the DPLL in Low Power mode. Check the DPLL documentation to see when this can be enabled. 0x0: Low power mode of the DPLL is disabled 0x1: Low power mode of the DPLL is enabled	RW	0x0
9	DPLL_RELOCK_RAMP_EN	If enabled, the clock ramping feature is used applied during the lock process, as well as the relock process. If disabled, the clock ramping feature is used only during the first lock.	RW	0x0
8	DPLL_DRIFTGUARD_EN	This bit allows to enable or disable the automatic recalibration feature of the DPLL. The DPLL will automatically start a recalibration process upon assertion of the DPLL's RECAL flag if this bit is set. 0x0: DRIFTGUARD feature is disabled 0x1: DRIFTGUARD feature is enabled	RW	0x0
7:5	DPLL_RAMP_RATE	Selects the time in terms of DPLL REFCLKs spent at each stage of the clock ramping process 0x0: 2 REFCLKs 0x1: 4 REFCLKs 0x2: 8 REFCLKs 0x3: 16 REFCLKs 0x4: 32 REFCLKs 0x5: 64 REFCLKs 0x6: 128 REFCLKs 0x7: 512 REFCLKs	RW	0x0
4:3	DPLL_RAMP_LEVEL	The DPLL provides an output clock frequency ramping feature when switching from bypass clock to normal clock during lock and re-lock. The frequency ramping will happen in a maximum of 4 steps in frequency before the DPLL's frequency lock indicator is asserted. This register is used to enable/disable the DPLL ramping feature. If enabled, it is also used to select the algorithm used for clock ramping 0x0: CLKOUT = No ramping CLKOUTX2 = No ramping 0x1: CLKOUT = Bypass clk - Fout/8 - Fout/4 - Fout/2 - Fout CLKOUTX2 = Bypass clk - Foutx2/8 - Foutx2/4 - Foutx2/2 - Foutx2 0x2: CLKOUT = Bypass clk - Fout/4 - Fout/2 - Fout/1.5 - Fout CLKOUTX2 = Bypass clk - Foutx2/4 - Foutx2/2 - Foutx2/1.5 - Foutx2 0x3: Reserved	RW	0x0
2:0	DPLL_EN	DPLL control. 0x0: Reserved 0x1: Reserved 0x2: Reserved 0x3: Reserved 0x4: Reserved 0x5: Put the DPLL in Idle Bypass Low Power mode. 0x6: Put the DPLL in Idle Bypass Fast Relock mode. 0x7: Enables the DPLL in Lock mode	RW	0x5

Table 3-395. Register Call Summary for Register CM_CLKMODE_DPLL_GMAC

Clock Management Functional Description

- DPLL_GMAC_DSP Power Modes: [0]
- DPLL_GMAC_DSP Recalibration: [1]

Table 3-395. Register Call Summary for Register CM_CLKMODE_DPLL_GMAC (continued)

PRCM Register Manual • CKGEN_CM_CORE_AON Register Summary: [4]

Table 3-396. CM_IDLEST_DPLL_GMAC

Address Offset	0x0000 01AC		
Physical Address	0x4A00 52AC	**Instance**	CKGEN_CM_CORE_AON
Description	This register allows monitoring DPLL activity. This register is read only and automatically updated. [warm reset insensitive]		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																											ST_DPLL_INIT	ST_DPLL_MODE			ST_DPLL_CLK

Bits	Field Name	Description	Type	Reset
31:5	RESERVED		R	0x0
4	ST_DPLL_INIT	DPLL init status (for debug purpose). 0x0: DPLL is not init 0x1: DPLL has been init	R	0x0
3:1	ST_DPLL_MODE	DPLL mode status (for debug purpose). 0x0: Transient state. From reset to any LP idle state or from any power state to any power state (Power states are Low Power Stop mode, Fast Relock Stop mode, Idle Bypass Low Power mode and Idle Bypass Fast Relock mode). 0x1: The DPLL is in Low Power Stop mode. 0x2: The DPLL is in Fast Relock Stop mode. 0x3: Reserved 0x4: Reserved 0x5: The DPLL is in Idle Bypass Low Power mode. 0x6: The DPLL is in Idle Bypass Fast Relock mode. 0x7: Reserved	R	0x0
0	ST_DPLL_CLK	DPLL lock status 0x0: DPLL is either in bypass mode or in stop mode. 0x1: DPLL is LOCKED	R	0x0

Table 3-397. Register Call Summary for Register CM_IDLEST_DPLL_GMAC

PRCM Register Manual • CKGEN_CM_CORE_AON Register Summary: [0]

Table 3-398. CM_AUTOIDLE_DPLL_GMAC

Address Offset	0x0000 01B0		
Physical Address	0x4A00 52B0	**Instance**	CKGEN_CM_CORE_AON
Description	This register provides automatic control over the DPLL activity.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														AUTO_DPLL_MODE	

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0
2:0	AUTO_DPLL_MODE	DPLL automatic control; 0x0: DPLL auto control disabled 0x1: The DPLL is automatically put in Low Power Stop mode when its DPLL generated clocks are not required anymore. It is also restarted automatically. 0x2: The DPLL is automatically put in Fast Relock Stop mode when its DPLL generated clocks are not required anymore. It is also restarted automatically. 0x3: Reserved 0x4: Reserved 0x5: The DPLL is automatically put in Idle Bypass Low Power mode when its DPLL generated clocks are not required anymore. It is also restarted automatically. 0x6: The DPLL is automatically put in Idle Bypass Fast Relock mode when its DPLL generated clocks are not required anymore. It is also restarted automatically. 0x7: Reserved	RW	0x0

Table 3-399. Register Call Summary for Register CM_AUTOIDLE_DPLL_GMAC

Clock Management Functional Description
- DPLL_GMAC_DSP Power Modes: [0]

PRCM Register Manual
- CKGEN_CM_CORE_AON Register Summary: [1]

Table 3-400. CM_CLKSEL_DPLL_GMAC

Address Offset	0x0000 01B4		
Physical Address	0x4A00 52B4	**Instance**	CKGEN_CM_CORE_AON
Description	This register provides controls over the DPLL.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DPLL_BYP_CLKSEL	DCC_EN	RESERVED	DPLL_CLKOUTHIF_CLKSEL	RESERVED	DPLL_MULT											RESERVED	DPLL_DIV						

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23	DPLL_BYP_CLKSEL	Allows control of the BYPASS clock of the PLL and the associated HSDIVIDER. Same as ULOWCLKEN on DPLL. In DPLL Locked mode, 0 - No impact 1 - No impact In DPLL Bypass mode, 0 - CLKINP is selected as the BYPASS clock for CLKOUT/CLKOUTX2 1 - CLKINPULOW is selected as the BYPASS clock for CLKOUT/CLKOUTX2	RW	0x0
22	DCC_EN	Duty-cycle corrector for high frequency clock 0x0: Duty-cycle corrector is disabled 0x1: Duty-cycle corrector is enabled	RW	0x0
21	RESERVED		R	0x0
20	DPLL_CLKOUTHIF_CLKSEL	Selects the source of the DPLL CLKOUTHIF clock. Same as CLKINPHIFSEL pin on the DPLL 0x0: CLKOUTHIF is generated from the DPLL oscillator (DCO) 0x1: CLKOUTHIF is generated from CLKINPHIF	RW	0x0
19	RESERVED		R	0x0
18:8	DPLL_MULT	DPLL multiplier factor (2 to 2047). (equal to input M of DPLL; M=2 to 2047 = DPLL multiplies by M). [warm reset insensitive] 0x0: Reserved 0x1: Reserved	RW	0x0
7	RESERVED		R	0x0
6:0	DPLL_DIV	DPLL divider factor (0 to 127) (equal to input N of DPLL; actual division factor is N+1). [warm reset insensitive]	RW	0x0

Table 3-401. Register Call Summary for Register CM_CLKSEL_DPLL_GMAC

Clock Management Functional Description
- DPLL_GMAC_DSP Synthesized Clock Parameters: [0][1]

PRCM Register Manual
- CKGEN_CM_CORE_AON Register Summary: [2]

Table 3-402. CM_DIV_M2_DPLL_GMAC

Address Offset	0x0000 01B8		
Physical Address	0x4A00 52B8	**Instance**	CKGEN_CM_CORE_AON
Description	This register provides controls over the M2 divider of DPLL_GMAC_DSP.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						CLKST	RESERVED						DIVHS		

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	CLKST	DPLL CLKOUT status 0x0: The clock output is gated 0x1: The clock output is enabled	R	0x0
8:5	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
4:0	DIVHS	This field programs the M2 post-divider factor (1 to 31) of DPLL_GMAC_DSP. 0x0: Reserved 0x1: M2 = /1 0x2: M2 = /2 ... 0x1F: M2 = /31	RW	0x1

Table 3-403. Register Call Summary for Register CM_DIV_M2_DPLL_GMAC

Clock Management Functional Description • DPLL_GMAC_DSP Synthesized Clock Parameters: [0][1]
PRCM Register Manual • CKGEN_CM_CORE_AON Register Summary: [2]

Table 3-404. CM_DIV_M3_DPLL_GMAC

Address Offset	0x0000 01BC		
Physical Address	0x4A00 52BC	**Instance**	CKGEN_CM_CORE_AON
Description	This register provides controls over the M3 divider of DPLL_GMAC_DSP.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						CLKST	RESERVED							DIVHS	

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	CLKST	DPLL CLKOUTHIF status 0x0: The clock output is gated 0x1: The clock output is enabled	R	0x0
8:5	RESERVED		R	0x0
4:0	DIVHS	This field programs the M3 post-divider factor (1 to 31) of DPLL_GMAC_DSP. 0x0: Reserved 0x1: M3 = /1 0x2: M3 = /2 ... 0x1F: M3 = /31	RW	0x1

Table 3-405. Register Call Summary for Register CM_DIV_M3_DPLL_GMAC

Clock Management Functional Description • DPLL_GMAC_DSP Synthesized Clock Parameters: [0][1]
PRCM Register Manual • CKGEN_CM_CORE_AON Register Summary: [2]

Table 3-406. CM_DIV_H11_DPLL_GMAC

Address Offset	0x0000 01C0		
Physical Address	0x4A00 52C0	**Instance**	CKGEN_CM_CORE_AON
Description	This register provides controls over the CLKOUT1 o/p of the HSDIVIDER1 DPLL_GMAC_DSP.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						CLKST	RESERVED					DIVHS			

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	CLKST	HSDIVIDER1 CLKOUT1 status 0x0: The clock output is gated 0x1: The clock output is enabled	R	0x0
8:6	RESERVED		R	0x0
5:0	DIVHS	This field programs the H11 post-divider factor (1 to 63) of DPLL_GMAC_DSP. 0x0: Reserved 0x1: H11 = /1 0x2: H11 = /2 ... 0x3F: H11 = /63	RW	0x1

Table 3-407. Register Call Summary for Register CM_DIV_H11_DPLL_GMAC

Clock Management Functional Description
- DPLL_GMAC_DSP Synthesized Clock Parameters: [0][1]

PRCM Register Manual
- CKGEN_CM_CORE_AON Register Summary: [2]

Table 3-408. CM_DIV_H12_DPLL_GMAC

Address Offset	0x0000 01C4		
Physical Address	0x4A00 52C4	**Instance**	CKGEN_CM_CORE_AON
Description	This register provides controls over the CLKOUT2 o/p of the HSDIVIDER1 DPLL_GMAC_DSP.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						CLKST	RESERVED					DIVHS			

Bits	Field Name	Description	Type	Reset
31:10	RESERVED	Reserved	R	0x0
9	CLKST	HSDIVIDER1 CLKOUT2 status 0x0: The clock output is gated 0x1: The clock output is enabled	R	0x0

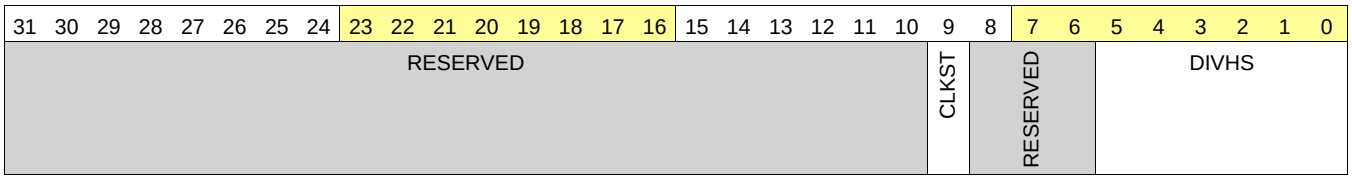
Bits	Field Name	Description	Type	Reset
8:6	RESERVED	Reserved	R	0x0
5:0	DIVHS	This field programs the H12 post-divider factor (1 to 63) of DPLL_GMAC_DSP. 0x0: Reserved 0x1: H12 = /1 0x2: H12 = /2 ... 0x3F: H12 = /63	RW	0x1

Table 3-409. Register Call Summary for Register CM_DIV_H12_DPLL_GMAC

Clock Management Functional Description • DPLL_GMAC_DSP Synthesized Clock Parameters: [0][1]
PRCM Register Manual • CKGEN_CM_CORE_AON Register Summary: [2]

Table 3-410. CM_DIV_H13_DPLL_GMAC

Address Offset	0x0000 01C8		
Physical Address	0x4A00 52C8	**Instance**	CKGEN_CM_CORE_AON
Description	This register provides controls over the CLKOUT3 o/p of the HSDIVIDER1 DPLL_GMAC_DSP.		
Type	RW		



Bits	Field Name	Description	Type	Reset
31:10	RESERVED	Reserved	R	0x0
9	CLKST	HSDIVIDER1 CLKOUT3 status 0x0: The clock output is gated 0x1: The clock output is enabled	R	0x0
8:6	RESERVED	Reserved	R	0x0
5:0	DIVHS	This field programs the H13 post-divider factor (1 to 63) of DPLL_GMAC_DSP. 0x0: Reserved 0x1: H13 = /1 0x2: H13 = /2 ... 0x3F: H13 = /63	RW	0x1

Table 3-411. Register Call Summary for Register CM_DIV_H13_DPLL_GMAC

Clock Management Functional Description • DPLL_GMAC_DSP Synthesized Clock Parameters: [0][1]
PRCM Register Manual • CKGEN_CM_CORE_AON Register Summary: [2]

Table 3-412. CM_DIV_H14_DPLL_GMAC

Address Offset	0x0000 01CC		
Physical Address	0x4A00 52CC	**Instance**	CKGEN_CM_CORE_AON
Description	This register provides controls over the CLKOUT4 o/p of the HSDIVIDER1 DPLL_GMAC_DSP.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						CLKST	RESERVED					DIVHS			

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	CLKST	HSDIVIDER1 CLKOUT4 status 0x0: The clock output is gated 0x1: The clock output is enabled	R	0x0
8:6	RESERVED		R	0x0
5:0	DIVHS	This field programs the H14 post-divider factor (1 to 63) of DPLL_GMAC_DSP. 0x0: Reserved 0x1: H14 = /1 0x2: H14 = /2 ... 0x3F: H14 = /63	RW	0x1

Table 3-413. CM_SSC_DELTAMSTEP_DPLL_GMAC

Address Offset	0x0000 01D0		
Physical Address	0x4A00 52D0	**Instance**	CKGEN_CM_CORE_AON
Description	Control the DeltaMStep parameter for Spread Spectrum Clocking. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												DELTAMSTEP																			

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19:0	DELTAMSTEP	DeltaMStep is split into fractional and integer part. For Triangular Spread Spectrum: [19:18] for integer part, [17:0] for fractional part For Square Wave Spread Spectrum [19:14] for integer part, [13:0] for fractional part	RW	0x0

Table 3-414. CM_SSC_MODFREQDIV_DPLL_GMAC

Address Offset	0x0000 01D4		
Physical Address	0x4A00 52D4	**Instance**	CKGEN_CM_CORE_AON
Description	Control the Modulation Frequency (Fm) for Spread Spectrum Clocking. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																					MODFREQDIV_EXPONENT			RESERVED	MODFREQDIV_MANTISSA						

Bits	Field Name	Description	Type	Reset
31:11	RESERVED		R	0x0
10:8	MODFREQDIV_EXPONENT	Set the Exponent component of MODFREQDIV factor	RW	0x0
7	RESERVED		R	0x0
6:0	MODFREQDIV_MANTISSA	Set the Mantissa component of MODFREQDIV factor	RW	0x0

Table 3-415. CM_CLKMODE_DPLL_GPU

Address Offset	0x0000 01D8		
Physical Address	0x4A00 52D8	**Instance**	CKGEN_CM_CORE_AON
Description	This register allows controlling the DPLL modes.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																DPLL_SSC_TYPE	DPLL_SSC_DOWNSPREAD	DPLL_SSC_ACK	DPLL_SSC_EN	DPLL_REGM4XEN	DPLL_LPMODE_EN	DPLL_RELOCK_RAMP_EN	DPLL_DRIFTGUARD_EN	DPLL_RAMP_RATE			DPLL_RAMP_LEVEL				DPLL_EN

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0
15	DPLL_SSC_TYPE	Select between Triangular and SquareWave Spread Spectrum Clocking 0x0: Triangular Spread Spectrum Clocking is selected 0x1: Square Wave Spread Spectrum Clocking is selected (only available under proper licensing agreement)	RW	0x0
14	DPLL_SSC_DOWNSPREAD	Control if only low frequency spread is required 0x0: When SSC is enabled, clock frequency is spread on both sides of the programmed frequency 0x1: When SSC is enabled, clock frequency is spread only on the lower side of the programmed frequency	RW	0x0
13	DPLL_SSC_ACK	Acknowledgement from the DPLL regarding start and stop of Spread Spectrum Clocking feature 0x0: SSC has been turned off on PLL o/ps 0x1: SSC has been turned on on PLL o/ps	R	0x0

Bits	Field Name	Description	Type	Reset
12	DPLL_SSC_EN	Enable or disable Spread Spectrum Clocking 0x0: SSC disabled 0x1: SSC enabled	RW	0x0
11	DPLL_REGM4XEN	Enable the REGM4XEN mode of the DPLL. Please check the DPLL documentation to check when this mode can be enabled. 0x0: REGM4XEN mode of the DPLL is disabled	R	0x0
10	DPLL_LPMODE_EN	Set the DPLL in Low Power mode. Check the DPLL documentation to see when this can be enabled. 0x0: Low power mode of the DPLL is disabled 0x1: Low power mode of the DPLL is enabled	RW	0x0
9	DPLL_RELOCK_RAMP_EN	If enabled, the clock ramping feature is used applied during the lock process, as well as the relock process. If disabled, the clock ramping feature is used only during the first lock.	RW	0x0
8	DPLL_DRIFTGUARD_EN	This bit allows to enable or disable the automatic recalibration feature of the DPLL. The DPLL will automatically start a recalibration process upon assertion of the DPLL's RECAL flag if this bit is set. 0x0: DRIFTGUARD feature is disabled 0x1: DRIFTGUARD feature is enabled	RW	0x0
7:5	DPLL_RAMP_RATE	Selects the time in terms of DPLL REFCLKs spent at each stage of the clock ramping process 0x0: 2 REFCLKs 0x1: 4 REFCLKs 0x2: 8 REFCLKs 0x3: 16 REFCLKs 0x4: 32 REFCLKs 0x5: 64 REFCLKs 0x6: 128 REFCLKs 0x7: 512 REFCLKs	RW	0x0
4:3	DPLL_RAMP_LEVEL	The DPLL provides an output clock frequency ramping feature when switching from bypass clock to normal clock during lock and re-lock. The frequency ramping will happen in a maximum of 4 steps in frequency before the DPLL's frequency lock indicator is asserted. This register is used to enable/disable the DPLL ramping feature. If enabled, it is also used to select the algorithm used for clock ramping 0x0: CLKOUT = No ramping CLKOUTX2 = No ramping 0x1: CLKOUT = Bypass clk - Fout/8 - Fout/4 - Fout/2 - Fout CLKOUTX2 = Bypass clk - Foutx2/8 - Foutx2/4 - Foutx2/2 - Foutx2 0x2: CLKOUT = Bypass clk - Fout/4 - Fout/2 - Fout/1.5 - Fout CLKOUTX2 = Bypass clk - Foutx2/4 - Foutx2/2 - Foutx2/1.5 - Foutx2 0x3: Reserved	RW	0x0
2:0	DPLL_EN	DPLL control. 0x0: Reserved 0x1: Reserved 0x2: Reserved 0x3: Reserved 0x4: Reserved 0x5: Put the DPLL in Idle Bypass Low Power mode. 0x6: Put the DPLL in Idle Bypass Fast Relock mode. 0x7: Enables the DPLL in Lock mode	RW	0x5

Table 3-416. CM_IDLEST_DPLL_GPU

Address Offset	0x0000 01DC		
Physical Address	0x4A00 52DC	**Instance**	CKGEN_CM_CORE_AON
Description	This register allows monitoring DPLL activity. This register is read only and automatically updated. [warm reset insensitive]		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																											ST_DPLL_INIT	ST_DPLL_MODE			ST_DPLL_CLK

Bits	Field Name	Description	Type	Reset
31:5	RESERVED		R	0x0
4	ST_DPLL_INIT	DPLL init status (for debug purpose). 0x0: DPLL is not init 0x1: DPLL has been init	R	0x0
3:1	ST_DPLL_MODE	DPLL mode status (for debug purpose). 0x0: Transient state. From reset to any LP idle state or from any power state to any power state (Power states are Low Power Stop mode, Fast Relock Stop mode, Idle Bypass Low Power mode and Idle Bypass Fast Relock mode). 0x1: The DPLL is in Low Power Stop mode. 0x2: The DPLL is in Fast Relock Stop mode. 0x3: Reserved 0x4: Reserved 0x5: The DPLL is in Idle Bypass Low Power mode. 0x6: The DPLL is in Idle Bypass Fast Relock mode. 0x7: Reserved	R	0x0
0	ST_DPLL_CLK	DPLL lock status 0x0: DPLL is either in bypass mode or in stop mode. 0x1: DPLL is LOCKED	R	0x0

Table 3-417. CM_AUTOIDLE_DPLL_GPU

Address Offset	0x0000 01E0		
Physical Address	0x4A00 52E0	**Instance**	CKGEN_CM_CORE_AON
Description	This register provides automatic control over the DPLL activity.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														AUTO_DPLL_MODE	

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0
2:0	AUTO_DPLL_MODE	DPLL automatic control. 0x0: DPLL auto control disabled 0x1: The DPLL is automatically put in Low Power Stop mode when its DPLL generated clocks are not required anymore. It is also restarted automatically. 0x2: The DPLL is automatically put in Fast Relock Stop mode when its DPLL generated clocks are not required anymore. It is also restarted automatically. 0x3: Reserved 0x4: Reserved 0x5: The DPLL is automatically put in Idle Bypass Low Power mode when its DPLL generated clocks are not required anymore. It is also restarted automatically. 0x6: The DPLL is automatically put in Idle Bypass Fast Relock mode when its DPLL generated clocks are not required anymore. It is also restarted automatically. 0x7: Reserved	RW	0x0

Table 3-418. CM_CLKSEL_DPLL_GPU

Address Offset	0x0000 01E4		
Physical Address	0x4A00 52E4	**Instance**	CKGEN_CM_CORE_AON
Description	This register provides controls over the DPLL.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DPLL_BYP_CLKSEL	DCC_EN	RESERVED	DPLL_CLKOUTHIF_CLKSEL	RESERVED	DPLL_MULT											RESERVED	DPLL_DIV						

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23	DPLL_BYP_CLKSEL	Allows control of the BYPASS clock of the PLL and the associated HSDIVIDER. Same as ULOWCLKEN on DPLL. In DPLL Locked mode, 0 - No impact 1 - No impact In DPLL Bypass mode, 0 - CLKINP is selected as the BYPASS clock for CLKOUT/CLKOUTX2 1 - CLKINPULOW is selected as the BYPASS clock for CLKOUT/CLKOUTX2	RW	0x0
22	DCC_EN	Duty-cycle corrector for high frequency clock 0x0: Duty-cycle corrector is disabled	R	0x0
21	RESERVED		R	0x0
20	DPLL_CLKOUTHIF_CLKSEL	Selects the source of the DPLL CLKOUTHIF clock. Same as CLKINPHIFSEL pin on the DPLL 0x0: CLKOUTHIF is generated from the DPLL oscillator (DCO) 0x1: CLKOUTHIF is generated from CLKINPHIF	RW	0x0
19	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
18:8	DPLL_MULT	DPLL multiplier factor (2 to 2047). (equal to input M of DPLL; M=2 to 2047 = DPLL multiplies by M). [warm reset insensitive] 0x0: Reserved 0x1: Reserved	RW	0x0
7	RESERVED		R	0x0
6:0	DPLL_DIV	DPLL divider factor (0 to 127) (equal to input N of DPLL; actual division factor is N+1). [warm reset insensitive]	RW	0x0

Table 3-419. CM_DIV_M2_DPLL_GPU

Address Offset	0x0000 01E8		
Physical Address	0x4A00 52E8	**Instance**	CKGEN_CM_CORE_AON
Description	This register provides controls over the M2 divider of the DPLL.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						CLKST	RESERVED					DIVHS			

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	CLKST	DPLL CLKOUT status 0x0: The clock output is gated 0x1: The clock output is enabled	R	0x0
8:5	RESERVED		R	0x0
4:0	DIVHS	This field programs the M2 post-divider factor (1 to 31) of DPLL_GPU. 0x0: Reserved 0x1: M2 = /1 0x2: M2 = /2 ... 0x1F: M2 = /31	RW	0x1

Table 3-420. CM_DIV_M3_DPLL_GPU

Address Offset	0x0000 01EC		
Physical Address	0x4A00 52EC	**Instance**	CKGEN_CM_CORE_AON
Description	This register provides controls over the M3 divider of the DPLL.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						CLKST	RESERVED					DIVHS			

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	CLKST	DPLL CLKOUTHIF status 0x0: The clock output is gated 0x1: The clock output is enabled	R	0x0

Bits	Field Name	Description	Type	Reset
8:5	RESERVED		R	0x0
4:0	DIVHS	This field programs the M3 post-divider factor (1 to 31) of DPLL_GPU. 0x0: Reserved 0x1: M3 = /1 0x2: M3 = /2 ... 0x1F: M3 = /31	RW	0x1

Table 3-421. CM_SSC_DELTAMSTEP_DPLL_GPU

Address Offset	0x0000 01F0		
Physical Address	0x4A00 52F0	**Instance**	CKGEN_CM_CORE_AON
Description	Control the DeltaMStep parameter for Spread Spectrum Clocking. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DELTAMSTEP																							

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19:0	DELTAMSTEP	DeltaMStep is split into fractional and integer part. For Triangular Spread Spectrum: [19:18] for integer part, [17:0] for fractional part For Square Wave Spread Spectrum [19:14] for integer part, [13:0] for fractional part	RW	0x0

Table 3-422. CM_SSC_MODFREQDIV_DPLL_GPU

Address Offset	0x0000 01F4		
Physical Address	0x4A00 52F4	**Instance**	CKGEN_CM_CORE_AON
Description	Control the Modulation Frequency (Fm) for Spread Spectrum Clocking. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																					MODFREQDIV_EXPONENT			RESERVED	MODFREQDIV_MANTISSA						

Bits	Field Name	Description	Type	Reset
31:11	RESERVED		R	0x0
10:8	MODFREQDIV_EXPONENT	Set the Exponent component of MODFREQDIV factor	RW	0x0
7	RESERVED		R	0x0
6:0	MODFREQDIV_MANTISSA	Set the Mantissa component of MODFREQDIV factor	RW	0x0

3.12.4 DSP1_CM_CORE_AON registers

3.12.4.1 DSP1_CM_CORE_AON Register Summary

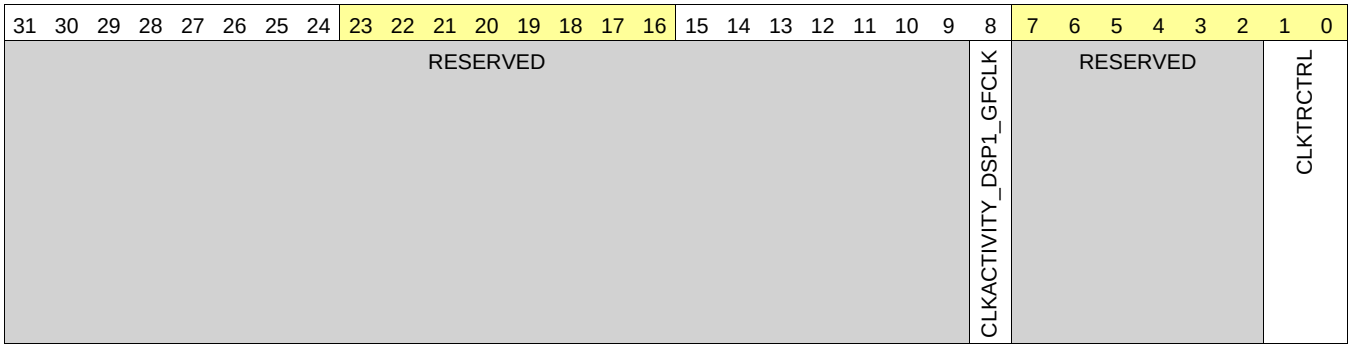
Table 3-423. DSP1_CM_CORE_AON Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	DSP1_CM_CORE_AON Physical Address
CM_DSP1_CLKSTCTRL	RW	32	0x0000 0000	0x4A00 5400
CM_DSP1_STATICDEP	RW	32	0x0000 0004	0x4A00 5404
CM_DSP1_DYNAMICDEP	RW	32	0x0000 0008	0x4A00 5408
CM_DSP1_DSP1_CLKCTRL	RW	32	0x0000 0020	0x4A00 5420

3.12.4.2 DSP1_CM_CORE_AON Register Description

Table 3-424. CM_DSP1_CLKSTCTRL

Address Offset	0x0000 0000		
Physical Address	0x4A00 5400	**Instance**	DSP1_CM_CORE_AON
Description	This register enables the DSP domain power state transition. It controls the HW supervised domain power state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain.		
Type	RW		



Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	CLKACTIVITY_DSP1_GFCLK	This field indicates the state of the DSP_ROOT_CLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
7:2	RESERVED		R	0x0
1:0	CLKTRCTRL	Controls the clock state transition of the DSP clock domain. 0x0: NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may however occur. 0x1: SW_SLEEP: Start a software forced sleep transition on the domain. 0x2: SW_WKUP: Start a software forced wake-up transition on the domain. 0x3: HW_AUTO: Automatic transition is enabled. Sleep and wakeup transition are based upon hardware conditions.	RW	0x3

Table 3-425. Register Call Summary for Register CM_DSP1_CLKSTCTRL

Reset Management Functional Description • DSP1 Subsystem Software Warm Reset Sequence: [0]
Clock Management Functional Description • Clock Domain Modes: [1][2]
PRCM Register Manual • DSP1_CM_CORE_AON Register Summary: [3]

Table 3-426. CM_DSP1_STATICDEP

Address Offset	0x0000 0004		
Physical Address	0x4A00 5404	**Instance**	DSP1_CM_CORE_AON
Description	This register controls the static domain depedencies from DSP domain towards 'target' domains. It is relevant only for domain having system initiator(s).		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	CRC_STATDEP	PCIE_STATDEP	ISS_STATDEP	L4PER3_STATDEP	L4PER2_STATDEP	GMAC_STATDEP	IPU_STATDEP	IPU1_STATDEP	EVE4_STATDEP	EVE3_STATDEP	EVE2_STATDEP	EVE1_STATDEP	DSP2_STATDEP	CUSTEFUSE_STATDEP	COREAON_STATDEP	WKUPAON_STATDEP	L4SEC_STATDEP	L4PER_STATDEP	L4CFG_STATDEP	RESERVED	GPU_STATDEP	CAM_STATDEP	DSS_STATDEP	L3INIT_STATDEP	RESERVED	L3MAIN1_STATDEP	EMIF_STATDEP	RESERVED	IVA_STATDEP	RESERVED	IPU2_STATDEP

Bits	Field Name	Description	Type	Reset
31	RESERVED		R	0x0
30	CRC_STATDEP	Static dependency towards L3INIT clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
29	PCIE_STATDEP	Static dependency towards PCIE cLOCK dOMAIN 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
28	ISS_STATDEP	Static dependency towards ISS Clock Domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
27	L4PER3_STATDEP	Static dependency towards L4PER3 Clock Domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
26	L4PER2_STATDEP	Static dependency towards L4PER2 Clock Domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
25	GMAC_STATDEP	Static dependency towards GMAC Clock Domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
24	IPU_STATDEP	Static dependency towards IPUClock Domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
23	IPU1_STATDEP	Static dependency towards IPU1 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
22	EVE4_STATDEP	Static dependency towards EVE4 Clock Domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
21	EVE3_STATDEP	Static dependency towards EVE3 Clock Domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
20	EVE2_STATDEP	Static dependency towards EVE2CLOCK dOMAIN 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
19	EVE1_STATDEP	Static dependency towards EVE1 CLOCK dOMAIN 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
18	DSP2_STATDEP	Static dependency towards DSP2 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
17	CUSTEFUSE_STATDEP	Static dependency towards CUSTEFUSE clock domain 0x0: Dependency is disabled	R	0x0
16	COREAON_STATDEP	Static dependency towards COREAON clock domain 0x0: Dependency is disabled	R	0x0
15	WKUPAON_STATDEP	Static dependency towards WKUPAON clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
14	L4SEC_STATDEP	Static dependency towards L4SEC Clock Domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
13	L4PER_STATDEP	Static dependency towards L4PER1clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
12	L4CFG_STATDEP	Static dependency towards L4CFG clock domain 0x0: Dependency is disabled	R	0x0
11	RESERVED		R	0x0
10	GPU_STATDEP	Static dependency towards GPU Clock Domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
9	CAM_STATDEP	Static dependency towards CAM clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
8	DSS_STATDEP	Static dependency towards DSS clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
7	L3INIT_STATDEP	Static dependency towards L3INIT clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	RESERVED		R	0x0
5	L3MAIN1_STATDEP	Static dependency towards L3MAIN1 clock domain 0x1: Dependency is enabled	R	0x1

Bits	Field Name	Description	Type	Reset
4	EMIF_STATDEP	Static dependency towards EMIF clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	RESERVED		R	0x0
2	IVA_STATDEP	Static dependency towards IVA clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	RESERVED		R	0x0
0	IPU2_STATDEP	Static dependency towards IPU2 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-427. Register Call Summary for Register CM_DSP1_STATICDEP

Clock Management Functional Description • Clock Domain Dependency: [0][1][2][3][4][5][6][7][8][9][10][11][12][13][14][15][16][17][18]
PRCM Register Manual • DSP1_CM_CORE_AON Register Summary: [19]

Table 3-428. CM_DSP1_DYNAMICDEP

Address Offset	0x0000 0008		
Physical Address	0x4A00 5408	**Instance**	DSP1_CM_CORE_AON
Description	This register controls the dynamic domain depedencies from DSP domain towards 'target' domains. It is relevant only for domain having OCP master port(s).		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				WINDOWSIZE				RESERVED																		L3MAIN1_DYNDEP	RESERVED					

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:24	WINDOWSIZE	Size of sliding window used to monitor OCP interface activity for determination of auto-sleep feature. Time unit defined by CM_DYN_DEP_PRESCAL register.	RW	0x4
23:6	RESERVED		R	0x0
5	L3MAIN1_DYNDEP	Dynamic dependency towards L3MAIN1 clock domain 0x1: Dependency is enabled	R	0x1
4:0	RESERVED		R	0x0

Table 3-429. Register Call Summary for Register CM_DSP1_DYNAMICDEP

Clock Management Functional Description • Clock Domain Dependency: [0] • Clock Domain Dependency: [1]
PRCM Register Manual • DSP1_CM_CORE_AON Register Summary: [2]

Table 3-430. CM_DSP1_DSP1_CLKCTRL

Address Offset	0x0000 0020		
Physical Address	0x4A00 5420	**Instance**	DSP1_CM_CORE_AON
Description	This register manages the DSP clocks.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED													STBYST	IDLEST		RESERVED														MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:19	RESERVED		R	0x0
18	STBYST	Module standby status. [warm reset insensitive]	R	0x1
		0x0: Module is functional (not in standby)		
		0x1: Module is in standby		
17:16	IDLEST	Module idle status. [warm reset insensitive]	R	0x3
		0x0: Module is fully functional, including OCP		
		0x1: Module is performing transition: wakeup, or sleep, or sleep abortion		
		0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock		
		0x3: Module is disabled and cannot be accessed		
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed.	RW	0x0
		0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup).		
		0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state.		
		0x2: Reserved		
		0x3: Reserved		

Table 3-431. Register Call Summary for Register CM_DSP1_DSP1_CLKCTRL

Reset Management Functional Description
• DSP1 Subsystem Software Warm Reset Sequence: [0][1]
Clock Management Functional Description
• Clock Domain Module Attributes: [2][3][4]
PRCM Register Manual
• DSP1_CM_CORE_AON Register Summary: [5]

3.12.5 DSP2_CM_CORE_AON registers

3.12.5.1 DSP2_CM_CORE_AON Register Summary

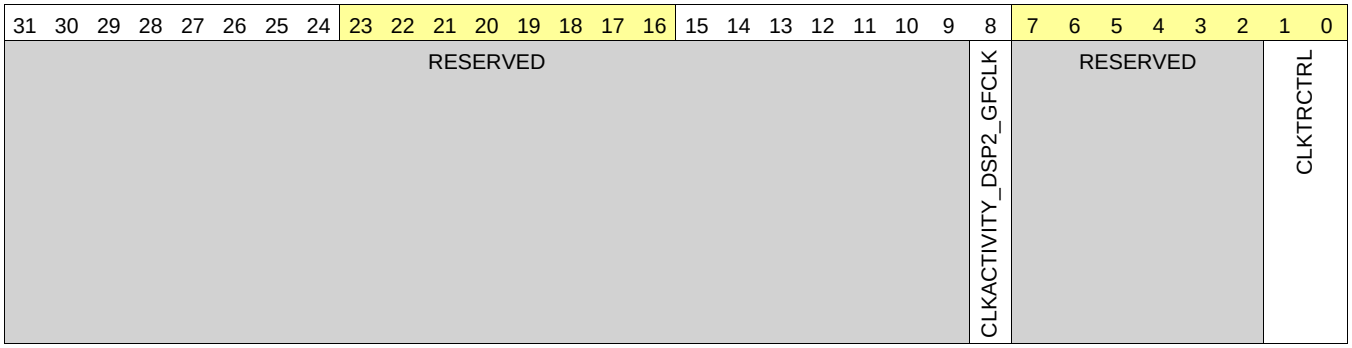
Table 3-432. DSP2_CM_CORE_AON Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	DSP2_CM_CORE_AON Physical Address
CM_DSP2_CLKSTCTRL	RW	32	0x0000 0000	0x4A00 5600
CM_DSP2_STATICDEP	RW	32	0x0000 0004	0x4A00 5604
CM_DSP2_DYNAMICDEP	RW	32	0x0000 0008	0x4A00 5608
CM_DSP2_DSP2_CLKCTRL	RW	32	0x0000 0020	0x4A00 5620

3.12.5.2 DSP2_CM_CORE_AON Register Description

Table 3-433. CM_DSP2_CLKSTCTRL

Address Offset	0x0000 0000		
Physical Address	0x4A00 5600	**Instance**	DSP2_CM_CORE_AON
Description	This register enables the DSP domain power state transition. It controls the HW supervised domain power state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain.		
Type	RW		



Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	CLKACTIVITY_DSP2_GFCLK	This field indicates the state of the DSP_ROOT_CLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
7:2	RESERVED		R	0x0
1:0	CLKTRCTRL	Controls the clock state transition of the DSP clock domain. 0x0: NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may however occur. 0x1: SW_SLEEP: Start a software forced sleep transition on the domain. 0x2: SW_WKUP: Start a software forced wake-up transition on the domain. 0x3: HW_AUTO: Automatic transition is enabled. Sleep and wakeup transition are based upon hardware conditions.	RW	0x3

Table 3-434. Register Call Summary for Register CM_DSP2_CLKSTCTRL

Reset Management Functional Description • DSP2 Subsystem Software Warm Reset Sequence: [0]
Clock Management Functional Description • Clock Domain Modes: [1][2]
PRCM Register Manual • DSP2_CM_CORE_AON Register Summary: [3]

Table 3-435. CM_DSP2_STATICDEP

Address Offset	0x0000 0004		
Physical Address	0x4A00 5604	**Instance**	DSP2_CM_CORE_AON
Description	This register controls the static domain depedencies from DSP domain towards 'target' domains. It is relevant only for domain having system initiator(s).		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	CRC_STATDEP	PCIE_STATDEP	ISS_STATDEP	L4PER3_STATDEP	L4PER2_STATDEP	GMAC_STATDEP	IPU_STATDEP	IPU1_STATDEP	EVE4_STATDEP	EVE3_STATDEP	EVE2_STATDEP	EVE1_STATDEP	RESERVED	CUSTEFUSE_STATDEP	COREAON_STATDEP	WKUPAON_STATDEP	L4SEC_STATDEP	L4PER_STATDEP	L4CFG_STATDEP	RESERVED	GPU_STATDEP	CAM_STATDEP	DSS_STATDEP	L3INIT_STATDEP	RESERVED	L3MAIN1_STATDEP	EMIF_STATDEP	RESERVED	IVA_STATDEP	DSP1_STATDEP	IPU2_STATDEP

Bits	Field Name	Description	Type	Reset
31	RESERVED		R	0x0
30	CRC_STATDEP	Static dependency towards L3INIT clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
29	PCIE_STATDEP	Static dependency towards PCIE cLOCK dOMAIN 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
28	ISS_STATDEP	Static dependency towards ISS Clock Domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
27	L4PER3_STATDEP	Static dependency towards L4PER3 Clock Domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
26	L4PER2_STATDEP	Static dependency towards L4PER2 Clock Domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
25	GMAC_STATDEP	Static dependency towards GMAC Clock Domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
24	IPU_STATDEP	Static dependency towards IPUClock Domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
23	IPU1_STATDEP	Static dependency towards IPU1 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
22	EVE4_STATDEP	Static dependency towards EVE4 Clock Domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
21	EVE3_STATDEP	Static dependency towards EVE3 Clock Domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
20	EVE2_STATDEP	Static dependency towards EVE2CLOCK dOMAIN 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
19	EVE1_STATDEP	Static dependency towards EVE1 CLOCK dOMAIN 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
18	RESERVED		R	0x0
17	CUSTEFUSE_STATDEP	Static dependency towards CUSTEFUSE clock domain 0x0: Dependency is disabled	R	0x0
16	COREAON_STATDEP	Static dependency towards COREAON clock domain 0x0: Dependency is disabled	R	0x0
15	WKUPAON_STATDEP	Static dependency towards WKUPAON clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
14	L4SEC_STATDEP	Static dependency towards L4SEC Clock Domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
13	L4PER_STATDEP	Static dependency towards L4PER1clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
12	L4CFG_STATDEP	Static dependency towards L4CFG clock domain 0x0: Dependency is disabled	R	0x0
11	RESERVED		R	0x0
10	GPU_STATDEP	Static dependency towards GPU Clock Domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
9	CAM_STATDEP	Static dependency towards CAM clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
8	DSS_STATDEP	Static dependency towards DSS clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
7	L3INIT_STATDEP	Static dependency towards L3INIT clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	RESERVED		R	0x0
5	L3MAIN1_STATDEP	Static dependency towards L3MAIN1 clock domain 0x1: Dependency is enabled	R	0x1

Bits	Field Name	Description	Type	Reset
4	EMIF_STATDEP	Static dependency towards EMIF clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	RESERVED		R	0x0
2	IVA_STATDEP	Static dependency towards IVA clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	DSP1_STATDEP	Static dependency towards DSP1 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	IPU2_STATDEP	Static dependency towards IPU2 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-436. Register Call Summary for Register CM_DSP2_STATICDEP

Clock Management Functional Description
- Clock Domain Dependency: [0][1][2][3][4][5][6][7][8][9][10][11][12][13][14][15][16][17][18]

PRCM Register Manual
- DSP2_CM_CORE_AON Register Summary: [19]

Table 3-437. CM_DSP2_DYNAMICDEP

Address Offset	0x0000 0008		
Physical Address	0x4A00 5608	**Instance**	DSP2_CM_CORE_AON
Description	This register controls the dynamic domain depedencies from DSP domain towards 'target' domains. It is relevant only for domain having OCP master port(s).		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				WINDOWSIZE				RESERVED																		L3MAIN1_DYNDEP	RESERVED					

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:24	WINDOWSIZE	Size of sliding window used to monitor OCP interface activity for determination of auto-sleep feature. Time unit defined by CM_DYN_DEP_PRESCAL register.	RW	0x4
23:6	RESERVED		R	0x0
5	L3MAIN1_DYNDEP	Dynamic dependency towards L3MAIN1 clock domain 0x1: Dependency is enabled	R	0x1
4:0	RESERVED		R	0x0

Table 3-438. Register Call Summary for Register CM_DSP2_DYNAMICDEP

PRCM Register Manual
- DSP2_CM_CORE_AON Register Summary: [0]

Table 3-439. CM_DSP2_DSP2_CLKCTRL

Address Offset	0x0000 0020		
Physical Address	0x4A00 5620	**Instance**	DSP2_CM_CORE_AON
Description	This register manages the DSP clocks.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED													STBYST	IDLEST		RESERVED														MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:19	RESERVED		R	0x0
18	STBYST	Module standby status. [warm reset insensitive]	R	0x1
		0x0: Module is functional (not in standby)		
		0x1: Module is in standby		
17:16	IDLEST	Module idle status. [warm reset insensitive]	R	0x3
		0x0: Module is fully functional, including OCP		
		0x1: Module is performing transition: wakeup, or sleep, or sleep abortion		
		0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock		
		0x3: Module is disabled and cannot be accessed		
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed.	RW	0x0
		0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup).		
		0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state.		
		0x2: Reserved		
		0x3: Reserved		

Table 3-440. Register Call Summary for Register CM_DSP2_DSP2_CLKCTRL

Reset Management Functional Description
• DSP2 Subsystem Software Warm Reset Sequence: [0][1]
Clock Management Functional Description
• Clock Domain Module Attributes: [2][3][4]
PRCM Register Manual
• DSP2_CM_CORE_AON Register Summary: [5]

3.12.6 EVE1_CM_CORE_AON registers

3.12.6.1 EVE1_CM_CORE_AON Register Summary

Table 3-441. EVE1_CM_CORE_AON Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	EVE1_CM_CORE_AON Physical Address
CM_EVE1_CLKSTCTRL	RW	32	0x0000 0000	0x4A00 5640
CM_EVE1_STATICDEP	RW	32	0x0000 0004	0x4A00 5644
CM_EVE1_EVE1_CLKCTRL	RW	32	0x0000 0020	0x4A00 5660

3.12.6.2 EVE1_CM_CORE_AON Register Description

Table 3-442. CM_EVE1_CLKSTCTRL

Address Offset	0x0000 0000		
Physical Address	0x4A00 5640	**Instance**	EVE1_CM_CORE_AON
Description	This register enables the EVE domain power state transition. It controls the HW supervised domain power state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							CLKACTIVITY_EVE1_GFCLK	RESERVED						CLKTRCTRL	

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	CLKACTIVITY_EVE1_GFCLK	This field indicates the state of the EVE1_GFCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
7:2	RESERVED		R	0x0
1:0	CLKTRCTRL	Controls the clock state transition of the DSP clock domain. 0x0: NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may however occur. 0x1: SW_SLEEP: Start a software forced sleep transition on the domain. 0x2: SW_WKUP: Start a software forced wake-up transition on the domain. 0x3: HW_AUTO: Automatic transition is enabled. Sleep and wakeup transition are based upon hardware conditions.	RW	0x3

Table 3-443. Register Call Summary for Register CM_EVE1_CLKSTCTRL

Clock Management Functional Description
• Clock Domain Modes: [0][1]
PRCM Register Manual
• EVE1_CM_CORE_AON Register Summary: [2]

Table 3-444. CM_EVE1_STATICDEP

Address Offset	0x0000 0004		
Physical Address	0x4A00 5644	**Instance**	EVE1_CM_CORE_AON
Description	This register controls the static domain depedencies from EVE1 domain towards 'target' domains. It is relevant only for domain having system initiator(s).		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								EVE4_STATDEP	EVE3_STATDEP	EVE2_STATDEP	RESERVED													L3MAIN1_STATDEP	EMIF_STATDEP	RESERVED	IVA_STATDEP	RESERVED			

Bits	Field Name	Description	Type	Reset
31:23	RESERVED		R	0x0
22	EVE4_STATDEP	Static dependency towards EVE4 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
21	EVE3_STATDEP	Static dependency towards EVE3 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
20	EVE2_STATDEP	Static dependency towards EVE2 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
19:6	RESERVED		R	0x0
5	L3MAIN1_STATDEP	Static dependency towards L3MAIN1 clock domain 0x1: Dependency is enabled	R	0x1
4	EMIF_STATDEP	Static dependency towards EMIF clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
3	RESERVED		R	0x0
2	IVA_STATDEP	Static dependency towards IVA clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1:0	RESERVED		R	0x0

Table 3-445. Register Call Summary for Register CM_EVE1_STATICDEP

Clock Management Functional Description
• Clock Domain Dependency: [0][1]
PRCM Register Manual
• EVE1_CM_CORE_AON Register Summary: [2]

Table 3-446. CM_EVE1_EVE1_CLKCTRL

Address Offset	0x0000 0020		
Physical Address	0x4A00 5660	**Instance**	EVE1_CM_CORE_AON
Description	This register manages the EVE clocks.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED													STBYST	IDLEST		RESERVED														MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:19	RESERVED		R	0x0
18	STBYST	Module standby status. [warm reset insensitive]	R	0x1
		0x0: Module is functional (not in standby)		
		0x1: Module is in standby		
17:16	IDLEST	Module idle status. [warm reset insensitive]	R	0x3
		0x0: Module is fully functional, including OCP		
		0x1: Module is performing transition: wakeup, or sleep, or sleep abortion		
		0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock		
		0x3: Module is disabled and cannot be accessed		
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed.	RW	0x0
		0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup).		
		0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state.		
		0x2: Reserved		
		0x3: Reserved		

Table 3-447. Register Call Summary for Register CM_EVE1_EVE1_CLKCTRL

Clock Management Functional Description

- Clock Domain Module Attributes: [0][1][2]

PRCM Register Manual

- EVE1_CM_CORE_AON Register Summary: [3]

3.12.7 EVE2_CM_CORE_AON registers

3.12.7.1 EVE2_CM_CORE_AON Register Summary

Table 3-448. EVE2_CM_CORE_AON Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	EVE2_CM_CORE_AON Physical Address
CM_EVE2_CLKSTCTRL	RW	32	0x0000 0000	0x4A00 5680
CM_EVE2_STATICDEP	RW	32	0x0000 0004	0x4A00 5684
CM_EVE2_EVE2_CLKCTRL	RW	32	0x0000 0020	0x4A00 56A0

3.12.7.2 EVE2_CM_CORE_AON Register Description

Table 3-449. CM_EVE2_CLKSTCTRL

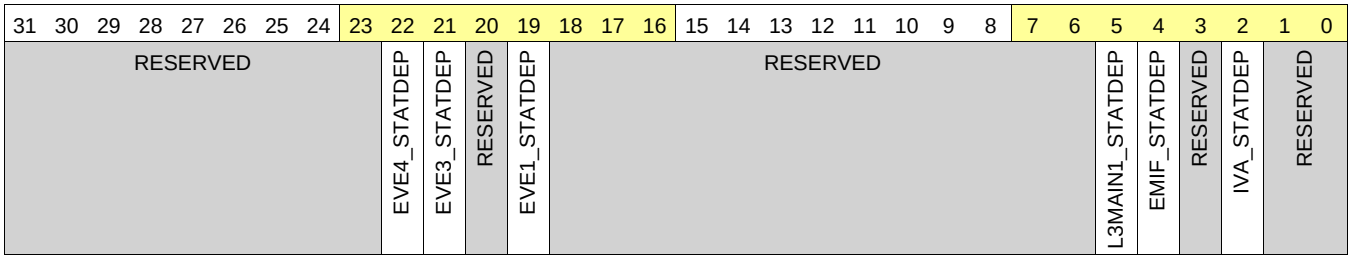
Address Offset	0x0000 0000		
Physical Address	0x4A00 5680	**Instance**	EVE2_CM_CORE_AON
Description	This register enables the EVE domain power state transition. It controls the HW supervised domain power state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							CLKACTIVITY_EVE2_GFCLK	RESERVED						CLKTRCTRL	

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	CLKACTIVITY_EVE2_GFCLK	This field indicates the state of the EVE2_GFCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
7:2	RESERVED		R	0x0
1:0	CLKTRCTRL	Controls the clock state transition of the DSP clock domain. 0x0: NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may however occur. 0x1: SW_SLEEP: Start a software forced sleep transition on the domain. 0x2: SW_WKUP: Start a software forced wake-up transition on the domain. 0x3: HW_AUTO: Automatic transition is enabled. Sleep and wakeup transition are based upon hardware conditions.	RW	0x3

Table 3-450. CM_EVE2_STATICDEP

Address Offset	0x0000 0004		
Physical Address	0x4A00 5684	**Instance**	EVE2_CM_CORE_AON
Description	This register controls the static domain depedencies from EVE2 domain towards 'target' domains. It is relevant only for domain having system initiator(s).		
Type	RW		



31:23	22	21	20	19	18:6	5	4	3	2	1:0
RESERVED	EVE4_STATDEP	EVE3_STATDEP	RESERVED	EVE1_STATDEP	RESERVED	L3MAIN1_STATDEP	EMIF_STATDEP	RESERVED	IVA_STATDEP	RESERVED

Bits	Field Name	Description	Type	Reset
31:23	RESERVED		R	0x0
22	EVE4_STATDEP	Static dependency towards EVE4 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
21	EVE3_STATDEP	Static dependency towards EVE3 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
20	RESERVED		R	0x0
19	EVE1_STATDEP	Static dependency towards EVE1 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
18:6	RESERVED		R	0x0
5	L3MAIN1_STATDEP	Static dependency towards L3MAIN1 clock domain 0x1: Dependency is enabled	R	0x1
4	EMIF_STATDEP	Static dependency towards EMIF clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
3	RESERVED		R	0x0
2	IVA_STATDEP	Static dependency towards IVA clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1:0	RESERVED		R	0x0

Table 3-451. CM_EVE2_EVE2_CLKCTRL

Address Offset	0x0000 0020		
Physical Address	0x4A00 56A0	**Instance**	EVE2_CM_CORE_AON
Description	This register manages the EVE clocks.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED													STBYST	IDLEST		RESERVED														MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:19	RESERVED		R	0x0
18	STBYST	Module standby status. [warm reset insensitive] 0x0: Module is functional (not in standby) 0x1: Module is in standby	R	0x1
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state. 0x2: Reserved 0x3: Reserved	RW	0x0

3.12.8 EVE3_CM_CORE_AON registers

3.12.8.1 EVE3_CM_CORE_AON Register Summary

Table 3-452. EVE3_CM_CORE_AON Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	EVE3_CM_CORE_AON Physical Address
CM_EVE3_CLKSTCTRL	RW	32	0x0000 0000	0x4A00 56C0
CM_EVE3_STATICDEP	RW	32	0x0000 0004	0x4A00 56C4
CM_EVE3_EVE3_CLKCTRL	RW	32	0x0000 0020	0x4A00 56E0

3.12.8.2 EVE3_CM_CORE_AON Register Description

Table 3-453. CM_EVE3_CLKSTCTRL

Address Offset	0x0000 0000		
Physical Address	0x4A00 56C0	**Instance**	EVE3_CM_CORE_AON
Description	This register enables the EVE domain power state transition. It controls the HW supervised domain power state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain.		

Table 3-453. CM_EVE3_CLKSTCTRL (continued)

Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							CLKACTIVITY_EVE3_GFCLK	RESERVED						CLKTRCTRL	

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	CLKACTIVITY_EVE3_GFCLK	This field indicates the state of the EVE3_GFCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
7:2	RESERVED		R	0x0
1:0	CLKTRCTRL	Controls the clock state transition of the DSP clock domain. 0x0: NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may however occur. 0x1: SW_SLEEP: Start a software forced sleep transition on the domain. 0x2: SW_WKUP: Start a software forced wake-up transition on the domain. 0x3: HW_AUTO: Automatic transition is enabled. Sleep and wakeup transition are based upon hardware conditions.	RW	0x3

Table 3-454. CM_EVE3_STATICDEP

Address Offset	0x0000 0004		
Physical Address	0x4A00 56C4	Instance	EVE3_CM_CORE_AON
Description	This register controls the static domain depedencies from EVE3 domain towards 'target' domains. It is relevant only for domain having system initiator(s).		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED									EVE4_STATDEP	RESERVED	EVE2_STATDEP	EVE1_STATDEP	RESERVED											L3MAIN1_STATDEP	EMIF_STATDEP	RESERVED	IVA_STATDEP	RESERVED			

Bits	Field Name	Description	Type	Reset
31:23	RESERVED		R	0x0
22	EVE4_STATDEP	Static dependency towards EVE4 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
21	RESERVED		R	0x0
20	EVE2_STATDEP	Static dependency towards EVE2 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
19	EVE1_STATDEP	Static dependency towards EVE1 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
18:6	RESERVED		R	0x0
5	L3MAIN1_STATDEP	Static dependency towards L3MAIN1 clock domain 0x1: Dependency is enabled	R	0x1
4	EMIF_STATDEP	Static dependency towards EMIF clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
3	RESERVED		R	0x0
2	IVA_STATDEP	Static dependency towards IVA clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1:0	RESERVED		R	0x0

Table 3-455. CM_EVE3_EVE3_CLKCTRL

Address Offset	0x0000 0020		
Physical Address	0x4A00 56E0	**Instance**	EVE3_CM_CORE_AON
Description	This register manages the EVE clocks.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED													STBYST	IDLEST		RESERVED								MODULEMODE							

Bits	Field Name	Description	Type	Reset
31:19	RESERVED		R	0x0
18	STBYST	Module standby status. [warm reset insensitive] 0x0: Module is functional (not in standby) 0x1: Module is in standby	R	0x1
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state. 0x2: Reserved 0x3: Reserved	RW	0x0

3.12.9 EVE4_CM_CORE_AON registers

3.12.9.1 EVE4_CM_CORE_AON Register Summary

Table 3-456. EVE4_CM_CORE_AON Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	EVE4_CM_CORE_AON Physical Address
CM_EVE4_CLKSTCTRL	RW	32	0x0000 0000	0x4A00 5700
CM_EVE4_STATICDEP	RW	32	0x0000 0004	0x4A00 5704
CM_EVE4_EVE4_CLKCTRL	RW	32	0x0000 0020	0x4A00 5720

3.12.9.2 EVE4_CM_CORE_AON Register Description

Table 3-457. CM_EVE4_CLKSTCTRL

Address Offset	0x0000 0000		
Physical Address	0x4A00 5700	**Instance**	EVE4_CM_CORE_AON
Description	This register enables the EVE domain power state transition. It controls the HW supervised domain power state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							CLKACTIVITY_EVE4_GFCLK	RESERVED						CLKTRCTRL	

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	CLKACTIVITY_EVE4_GFCLK	This field indicates the state of the EVE3_GFCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
7:2	RESERVED		R	0x0
1:0	CLKTRCTRL	Controls the clock state transition of the DSP clock domain. 0x0: NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may however occur. 0x1: SW_SLEEP: Start a software forced sleep transition on the domain. 0x2: SW_WKUP: Start a software forced wake-up transition on the domain. 0x3: HW_AUTO: Automatic transition is enabled. Sleep and wakeup transition are based upon hardware conditions.	RW	0x3

Table 3-458. CM_EVE4_STATICDEP

Address Offset	0x0000 0004		
Physical Address	0x4A00 5704	**Instance**	EVE4_CM_CORE_AON
Description	This register controls the static domain depedencies from EVE4 domain towards 'target' domains. It is relevant only for domain having system initiator(s).		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED										EVE3_STATDEP	EVE2_STATDEP	EVE1_STATDEP	RESERVED											L3MAIN1_STATDEP	EMIF_STATDEP	RESERVED	IVA_STATDEP	RESERVED			

Bits	Field Name	Description	Type	Reset
31:22	RESERVED		R	0x0
21	EVE3_STATDEP	Static dependency towards EVE3 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
20	EVE2_STATDEP	Static dependency towards EVE2 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
19	EVE1_STATDEP	Static dependency towards EVE1 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
18:6	RESERVED		R	0x0
5	L3MAIN1_STATDEP	Static dependency towards L3MAIN1 clock domain 0x1: Dependency is enabled	R	0x1
4	EMIF_STATDEP	Static dependency towards EMIF clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1

Bits	Field Name	Description	Type	Reset
3	RESERVED		R	0x0
2	IVA_STATDEP	Static dependency towards IVA clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1:0	RESERVED		R	0x0

Table 3-459. CM_EVE4_EVE4_CLKCTRL

Address Offset	0x0000 0020		
Physical Address	0x4A00 5720	**Instance**	EVE4_CM_CORE_AON
Description	This register manages the EVE clocks.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED													STBYST	IDLEST		RESERVED														MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:19	RESERVED		R	0x0
18	STBYST	Module standby status. [warm reset insensitive] 0x0: Module is functional (not in standby) 0x1: Module is in standby	R	0x1
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state. 0x2: Reserved 0x3: Reserved	RW	0x0

3.12.10 INSTR_CM_CORE_AON registers

3.12.10.1 INSTR_CM_CORE_AON Register Summary

Table 3-460. INSTR_CM_CORE_AON Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	INSTR_CM_CORE_AON Physical Address
CMI_IDENTICATION	R	32	0x0000 0000	0x4A00 5F00
CMI_SYS_CONFIG	RW	32	0x0000 0010	0x4A00 5F10
CMI_STATUS	R	32	0x0000 0014	0x4A00 5F14
CMI_CONFIGURATION	RW	32	0x0000 0024	0x4A00 5F24
CMI_CLASS_FILTERING	RW	32	0x0000 0028	0x4A00 5F28
CMI_TRIGGERING	RW	32	0x0000 002C	0x4A00 5F2C
CMI_SAMPLING	RW	32	0x0000 0030	0x4A00 5F30

3.12.10.2 INSTR_CM_CORE_AON Register Description

Table 3-461. CMI_IDENTICATION

Address Offset	0x0000 0000		
Physical Address	0x4A00 5F00	**Instance**	INSTR_CM_CORE_AON
Description	CM profiling identification register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCHEME		RESERVED		FUNC												RTL					MAJOR			CUSTOM				MINOR			

Bits	Field Name	Description	Type	Reset
31:30	SCHEME		R	0x1
29:28	RESERVED		R	0x1
27:16	FUNC	Indicates a software compatible module family	R	0x5
15:11	RTL	RTL version	R	0x9
10:8	MAJOR	Major revision	R	0x0
7:6	CUSTOM	Indicates a special version for a particular device	R	0x0
5:0	MINOR	Minor revision	R	0x0

Table 3-462. Register Call Summary for Register CMI_IDENTICATION

PRCM Register Manual
- INSTR_CM_CORE_AON Register Summary: [0]

Table 3-463. CMI_SYS_CONFIG

Address Offset	0x0000 0010		
Physical Address	0x4A00 5F10	**Instance**	INSTR_CM_CORE_AON
Description	CM profiling system configuartion register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								RESERVED				IDLEMODE		RESERVED	SOFTRESET

Bits	Field Name	Description	Type	Reset
31:4	RESERVED		R	0x0
3:2	IDLEMODE	Configuration of the local target state management mode	RW	0x2
1	RESERVED		R	0x0
0	SOFTRESET	Software reset	RW	0x0

Table 3-464. Register Call Summary for Register CMI_SYS_CONFIG

PRCM Register Manual
- INSTR_CM_CORE_AON Register Summary: [0]

Table 3-465. CMI_STATUS

Address Offset	0x0000 0014		
Physical Address	0x4A00 5F14	**Instance**	INSTR_CM_CORE_AON
Description	CM profiling status register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							FIFOEMPTY	RESERVED							

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	FIFOEMPTY	PM Profiling buffer empty	R	0x1
7:0	RESERVED		R	0x0

Table 3-466. Register Call Summary for Register CMI_STATUS

PRCM Register Manual
- INSTR_CM_CORE_AON Register Summary: [0]

Table 3-467. CMI_CONFIGURATION

Address Offset	0x0000 0024		
Physical Address	0x4A00 5F24	**Instance**	INSTR_CM_CORE_AON
Description	CM profiling configuration register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CLAIM_3		CLAIM_2	CLAIM_1	RESERVED												MOD_ACT_EN	RESERVED							EVT_CAPT_EN	RESERVED						

Bits	Field Name	Description	Type	Reset
31:30	CLAIM_3	Ownership	RW	0x0
29	CLAIM_2	Debugger override qualifier	RW	0x1
28	CLAIM_1	Current owner	R	0x0
27:16	RESERVED		R	0x0
15	MOD_ACT_EN	When HIGH the CM Module Activity collection is enabled	RW	0x0
14:8	RESERVED		R	0x0
7	EVT_CAPT_EN	When HIGH the CM events capture is enabled	RW	0x0
6:0	RESERVED		R	0x0

Table 3-468. Register Call Summary for Register CMI_CONFIGURATION

PRCM Register Manual

- INSTR_CM_CORE_AON Register Summary: [0]

Table 3-469. CMI_CLASS_FILTERING

Address Offset	0x0000 0028		
Physical Address	0x4A00 5F28	**Instance**	INSTR_CM_CORE_AON
Description	CM profiling class filtering register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SNAP_CAPT_EN_1F	SNAP_CAPT_EN_1E	SNAP_CAPT_EN_1D	SNAP_CAPT_EN_1C	SNAP_CAPT_EN_1B	SNAP_CAPT_EN_1A	SNAP_CAPT_EN_19	SNAP_CAPT_EN_18	SNAP_CAPT_EN_17	SNAP_CAPT_EN_16	SNAP_CAPT_EN_15	SNAP_CAPT_EN_14	SNAP_CAPT_EN_13	SNAP_CAPT_EN_12	SNAP_CAPT_EN_11	SNAP_CAPT_EN_10	RESERVED												SNAP_CAPT_EN_03	SNAP_CAPT_EN_02	SNAP_CAPT_EN_01	SNAP_CAPT_EN_00

Bits	Field Name	Description	Type	Reset
31	SNAP_CAPT_EN_1F	Snapshot capture enable - Class-ID = 0x1F	RW	0x0
30	SNAP_CAPT_EN_1E		RW	0x0
29	SNAP_CAPT_EN_1D		RW	0x0
28	SNAP_CAPT_EN_1C		RW	0x0
27	SNAP_CAPT_EN_1B		RW	0x0
26	SNAP_CAPT_EN_1A		RW	0x0
25	SNAP_CAPT_EN_19		RW	0x0
24	SNAP_CAPT_EN_18		RW	0x0
23	SNAP_CAPT_EN_17		RW	0x0
22	SNAP_CAPT_EN_16		RW	0x0
21	SNAP_CAPT_EN_15		RW	0x0
20	SNAP_CAPT_EN_14		RW	0x0

Bits	Field Name	Description	Type	Reset
19	SNAP_CAPT_EN_13		RW	0x0
18	SNAP_CAPT_EN_12		RW	0x0
17	SNAP_CAPT_EN_11		RW	0x0
16	SNAP_CAPT_EN_10	Snapshot capture enable - Class-ID = 0x10	RW	0x0
15:4	RESERVED		R	0x0
3	SNAP_CAPT_EN_03	Snapshot capture enable - Class-ID = 0x03 [0x23]	RW	0x0
2	SNAP_CAPT_EN_02	Snapshot capture enable - Class-ID = 0x02 [0x22]	RW	0x0
1	SNAP_CAPT_EN_01	Snapshot capture enable - Class-ID = 0x01 [0x21]	RW	0x0
0	SNAP_CAPT_EN_00	Snapshot capture enable - Class-ID = 0x00 [0x20]	RW	0x0

Table 3-470. Register Call Summary for Register CMI_CLASS_FILTERING

PRCM Register Manual
- INSTR_CM_CORE_AON Register Summary: [0]

Table 3-471. CMI_TRIGGERING

Address Offset	0x0000 002C		
Physical Address	0x4A00 5F2C	**Instance**	INSTR_CM_CORE_AON
Description	CM profiling triggering control register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														TRIG_STOP_EN	TRIG_START_EN

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1	TRIG_STOP_EN	Enable stop capturing CM events from external trigger detection	RW	0x0
0	TRIG_START_EN	Enable start capturing CM events from external trigger detection	RW	0x0

Table 3-472. Register Call Summary for Register CMI_TRIGGERING

PRCM Register Manual
- INSTR_CM_CORE_AON Register Summary: [0]

Table 3-473. CMI_SAMPLING

Address Offset	0x0000 0030		
Physical Address	0x4A00 5F30	**Instance**	INSTR_CM_CORE_AON
Description	CM profiling sampling window register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												FCLK_DIV_FACOR				RESERVED								SAMP_WIND_SIZE							

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19:16	FCLK_DIV_FACOR	FunClk divide factor ranging from 1 to 16	RW	0x0
15:8	RESERVED		R	0x0
7:0	SAMP_WIND_SIZE	CM events sampling window size	RW	0x0

Table 3-474. Register Call Summary for Register CMI_SAMPLING

PRCM Register Manual
- INSTR_CM_CORE_AON Register Summary: [0]

3.12.11 IPU_CM_CORE_AON registers

3.12.11.1 IPU_CM_CORE_AON Register Summary

Table 3-475. IPU_CM_CORE_AON Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	IPU_CM_CORE_AON Physical Address
CM_IPU1_CLKSTCTRL	RW	32	0x0000 0000	0x4A00 5500
CM_IPU1_STATICDEP	RW	32	0x0000 0004	0x4A00 5504
CM_IPU1_DYNAMICDEP	RW	32	0x0000 0008	0x4A00 5508
CM_IPU1_IPU1_CLKCTRL	RW	32	0x0000 0020	0x4A00 5520
CM_IPU_CLKSTCTRL	RW	32	0x0000 0040	0x4A00 5540
CM_IPU_MCASP1_CLKCTRL	RW	32	0x0000 0050	0x4A00 5550
CM_IPU_TIMER5_CLKCTRL	RW	32	0x0000 0058	0x4A00 5558
CM_IPU_TIMER6_CLKCTRL	RW	32	0x0000 0060	0x4A00 5560
CM_IPU_TIMER7_CLKCTRL	RW	32	0x0000 0068	0x4A00 5568
CM_IPU_TIMER8_CLKCTRL	RW	32	0x0000 0070	0x4A00 5570
CM_IPU_I2C5_CLKCTRL	RW	32	0x0000 0078	0x4A00 5578
CM_IPU_UART6_CLKCTRL	RW	32	0x0000 0080	0x4A00 5580

3.12.11.2 IPU_CM_CORE_AON Register Description

Table 3-476. CM_IPU1_CLKSTCTRL

Address Offset	0x0000 0000		
Physical Address	0x4A00 5500	**Instance**	IPU_CM_CORE_AON
Description	This register enables the domain power state transition. It controls the HW supervised domain power state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							CLKACTIVITY_IPU1_GFCLK	RESERVED						CLKTRCTRL	

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	CLKACTIVITY_IPU1_GFCLK	This field indicates the state of the IPU1_GFCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
7:2	RESERVED		R	0x0
1:0	CLKTRCTRL	Controls the clock state transition of the IPU1 clock domain. 0x0: NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may however occur. 0x1: SW_SLEEP: Start a software forced sleep transition on the domain. 0x2: SW_WKUP: Start a software forced wake-up transition on the domain. 0x3: HW_AUTO: Automatic transition is enabled. Sleep and wakeup transition are based upon hardware conditions.	RW	0x3

Table 3-477. Register Call Summary for Register CM_IPU1_CLKSTCTRL

Clock Management Functional Description • Clock Domain Modes: [0][1]
PRCM Register Manual • IPU_CM_CORE_AON Register Summary: [2]

Table 3-478. CM_IPU1_STATICDEP

Address Offset	0x0000 0004		
Physical Address	0x4A00 5504	**Instance**	IPU_CM_CORE_AON
Description	This register controls the static domain depedencies from IPU domain towards 'target' domains. It is relevant only for domain having system initiator(s).		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	CRC_STATDEP	PCIE_STATDEP	ISS_STATDEP	L4PER3_STATDEP	L4PER2_STATDEP	GMAC_STATDEP	IPU_STATDEP	RESERVED	EVE4_STATDEP	EVE3_STATDEP	EVE2_STATDEP	EVE1_STATDEP	DSP2_STATDEP	CUSTEFUSE_STATDEP	COREAON_STATDEP	WKUPAON_STATDEP	L4SEC_STATDEP	L4PER_STATDEP	L4CFG_STATDEP	SDMA_STATDEP	GPU_STATDEP	CAM_STATDEP	DSS_STATDEP	L3INIT_STATDEP	RESERVED	L3MAIN1_STATDEP	EMIF_STATDEP	RESERVED	IVA_STATDEP	DSP1_STATDEP	IPU2_STATDEP

Bits	Field Name	Description	Type	Reset
31	RESERVED		R	0x0
30	CRC_STATDEP	Static dependency towards CRC clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
29	PCIE_STATDEP	Static dependency towards PCIE clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
28	ISS_STATDEP	Static dependency towards ISS clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
27	L4PER3_STATDEP	Static dependency towards L4PER3 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
26	L4PER2_STATDEP	Static dependency towards L4PER2 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
25	GMAC_STATDEP	Static dependency towards GMAC clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
24	IPU_STATDEP	Static dependency towards IPU clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
23	RESERVED		R	0x0
22	EVE4_STATDEP	Static dependency towards EVE4 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
21	EVE3_STATDEP	Static dependency towards EVE3 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
20	EVE2_STATDEP	Static dependency towards EVE2 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
19	EVE1_STATDEP	Static dependency towards EVE1 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
18	DSP2_STATDEP	Static dependency towards DSP2 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
17	CUSTEFUSE_STATDEP	Static dependency towards CUSTEFUSE clock domain 0x0: Dependency is disabled	R	0x0
16	COREAON_STATDEP	Static dependency towards COREAON clock domain 0x0: Dependency is disabled	R	0x0
15	WKUPAON_STATDEP	Static dependency towards WKUPAON clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
14	L4SEC_STATDEP	Static dependency towards L4SEC clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
13	L4PER_STATDEP	Static dependency towards L4PER1 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
12	L4CFG_STATDEP	Static dependency towards L4CFG clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
11	SDMA_STATDEP	Static dependency towards DMA clock domain 0x0: Dependency is disabled	R	0x0
10	GPU_STATDEP	Static dependency towards GPU clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
9	CAM_STATDEP	Static dependency towards CAM clock domain 0x0: Dependency is disabled	R	0x0
8	DSS_STATDEP	Static dependency towards DSS clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
7	L3INIT_STATDEP	Static dependency towards L3INIT clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	RESERVED		R	0x0
5	L3MAIN1_STATDEP	Static dependency towards L3MAIN1 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
4	EMIF_STATDEP	Static dependency towards EMIF clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
3	RESERVED		R	0x0
2	IVA_STATDEP	Static dependency towards IVA clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	DSP1_STATDEP	Static dependency towards DSP clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	IPU2_STATDEP	Static dependency towards IPU2 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-479. Register Call Summary for Register CM_IPU1_STATICDEP

Clock Management Functional Description • Clock Domain Dependency: [0][1][2][3][4][5][6][7][8][9][10][11][12][13][14][15][16][17][18] • Clock Domain Dependency: [19][20][21][22][23][24][25][26][28][29][31][32][33][34][35][36][37][38][39]
PRCM Register Manual • IPU_CM_CORE_AON Register Summary: [40]

Table 3-480. CM_IPU1_DYNAMICDEP

Address Offset	0x0000 0008		
Physical Address	0x4A00 5508	**Instance**	IPU_CM_CORE_AON
Description	This register controls the dynamic domain depedencies from IPU domain towards 'target' domains. It is relevant only for domain having OCP master port(s).		

Table 3-480. CM_IPU1_DYNAMICDEP (continued)

Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				WINDOWSIZE				RESERVED																		L3MAIN1_DYNDEP	RESERVED						

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:24	WINDOWSIZE	Size of sliding window used to monitor OCP interface activity for determination of auto-sleep feature. Time unit defined by CM_DYN_DEP_PRESCAL register.	RW	0x4
23:6	RESERVED		R	0x0
5	L3MAIN1_DYNDEP	Dynamic dependency towards L3MAIN1 clock domain 0x1: Dependency is enabled	R	0x1
4:0	RESERVED		R	0x0

Table 3-481. Register Call Summary for Register CM_IPU1_DYNAMICDEP

Clock Management Functional Description
- Clock Domain Dependency: [0]
- Clock Domain Dependency: [1]

PRCM Register Manual
- IPU_CM_CORE_AON Register Summary: [2]

Table 3-482. CM_IPU1_IPU1_CLKCTRL

Address Offset	0x0000 0020		
Physical Address	0x4A00 5520	**Instance**	IPU_CM_CORE_AON
Description	This register manages the IPU1 clocks.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED							CLKSEL	RESERVED					STBYST	IDLEST		RESERVED															MODULEMODE

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24	CLKSEL	Selects the IPU functional clock 0x0: Selects DPLL_ABE_X2_CLK as the functional clock 0x1: Selects CORE_IPU_ISS_BOOST_CLK as the functional clock	RW	0x1
23:19	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
18	STBYST	Module standby status. [warm reset insensitive] 0x0: Module is functional (not in standby) 0x1: Module is in standby	R	0x1
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state. 0x2: Reserved 0x3: Reserved	RW	0x0

Table 3-483. Register Call Summary for Register CM_IPU1_IPU1_CLKCTRL

Clock Management Functional Description
- CM_CORE_AON Clock Generator: [0]
- Programming Guide For Control Module: [1]
- Clock Domain Module Attributes: [2][3][5]

PRCM Register Manual
- IPU_CM_CORE_AON Register Summary: [6]

Table 3-484. CM_IPU_CLKSTCTRL

Address Offset	0x0000 0040		
Physical Address	0x4A00 5540	**Instance**	IPU_CM_CORE_AON
Description	This register enables the IPU domain power state transition. It controls the HW supervised domain power state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED													CLKACTIVITY_MCASP1_AHCLKR	CLKACTIVITY_MCASP1_AHCLKX	CLKACTIVITY_MCASP1_AUX_GFCLK	RESERVED	CLKACTIVITY_UART6_GFCLK	CLKACTIVITY_IPU_96M_GFCLK	CLKACTIVITY_TIMER8_GFCLK	CLKACTIVITY_TIMER7_GFCLK	CLKACTIVITY_TIMER6_GFCLK	CLKACTIVITY_TIMER5_GFCLK	CLKACTIVITY_IPU_L3_GICLK	RESERVED						CLKTRCTRL	

Bits	Field Name	Description	Type	Reset
31:19	RESERVED		R	0x0
18	CLKACTIVITY_MCASP1_AHCLKR	This field indicates the state of the MCASP1_AHCLKR clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
17	CLKACTIVITY_MCASP1_AHCLKX	This field indicates the state of the MCASP1_AHCLKX clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
16	CLKACTIVITY_MCASP1_AUX_GFCLK	This field indicates the state of the MCASP1_AUX_GFCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
15	RESERVED		R	0x0
14	CLKACTIVITY_UART6_GFCLK	This field indicates the state of the UART6_GFCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
13	CLKACTIVITY_IPU_96M_GFCLK	This field indicates the state of the IPU_96M_GFCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
12	CLKACTIVITY_TIMER8_GFCLK	This field indicates the state of the TIMER8_GFCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
11	CLKACTIVITY_TIMER7_GFCLK	This field indicates the state of the TIMER7_GFCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0

Bits	Field Name	Description	Type	Reset
10	CLKACTIVITY_TIMER6_GFCLK	This field indicates the state of the TIMER6_GFCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
9	CLKACTIVITY_TIMER5_GFCLK	This field indicates the state of the TIMER5_GFCLK functional clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
8	CLKACTIVITY_IPU_L3_GICLK	This field indicates the state of the IPU_L3_GICLK interface clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
7:2	RESERVED		R	0x0
1:0	CLKTRCTRL	Controls the clock state transition of the IPU clock domain. 0x0: NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may however occur. 0x1: SW_SLEEP: Start a software forced sleep transition on the domain. 0x2: SW_WKUP: Start a software forced wake-up transition on the domain. 0x3: HW_AUTO: Automatic transition is enabled. Sleep and wakeup transition are based upon hardware conditions.	RW	0x3

Table 3-485. Register Call Summary for Register CM_IPU_CLKSTCTRL

Clock Management Functional Description • Clock Domain Modes: [0][1][2][3][4][5][6][7][8][9]
PRCM Register Manual • IPU_CM_CORE_AON Register Summary: [10]

Table 3-486. CM_IPU_MCASP1_CLKCTRL

Address Offset	0x0000 0050		
Physical Address	0x4A00 5550	**Instance**	IPU_CM_CORE_AON
Description	This register manages the McASP1 clocks.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CLKSEL_AHCLKR				CLKSEL_AHCLKX				CLKSEL_AUX_CLK		RESERVED				IDLEST		RESERVED														MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:28	CLKSEL_AHCLKR	Selects reference clock for AHCLKR 0x0: Selects ABE_24M_GFCLK 0x1: Selects ABE_SYS_CLK 0x2: Selects FUNC_24M_GFCLK 0x3: Selects ATL_CLK3 0x4: Selects ATL_CLK2 0x5: Selects ATL_CLK1 0x6: Selects ATL_CLK0 0x7: Selects SYS_CLK2 0x8: Selects XREF_CLK0 0x9: Selects XREF_CLK1 0xA: Selects XREF_CLK2 0xB: Selects XREF_CLK1 0xC: Selects divided version of SYS_CLK1. See CM_CLKSEL_MLB_MCASP 0xD: Selects divided version of SYS_CLK1. See CM_CLKSEL_MLBP_MCASP 0xE: RESERVED 0xF: RESERVED	RW	0x0
27:24	CLKSEL_AHCLKX	Selects reference clock for AHCLKX 0x0: Selects ABE_24M_GFCLK 0x1: Selects ABE_SYS_CLK 0x2: Selects FUNC_24M_GFCLK 0x3: Selects ATL_CLK3 0x4: Selects ATL_CLK2 0x5: Selects ATL_CLK1 0x6: Selects ATL_CLK0 0x7: Selects SYS_CLK2 0x8: Selects XREF_CLK0 0x9: Selects XREF_CLK1 0xA: Selects XREF_CLK2 0xB: Selects XREF_CLK1 0xC: Selects divided version of SYS_CLK1. See CM_CLKSEL_MLB_MCASP 0xD: Selects divided version of SYS_CLK1. See CM_CLKSEL_MLBP_MCASP 0xE: RESERVED 0xF: RESERVED	RW	0x0
23:22	CLKSEL_AUX_CLK	Selects the source of the AUX clock. 0x0: Selects divided version of SYS_CLK1. See CM_CLKSEL_PER_ABE_X1_GFCLK_MCASP_AUX 0x1: Selects divided version of SYS_CLK1. See CM_CLKSEL_VIDEO1_MCASP_AUX 0x2: Selects divided version of SYS_CLK1. See CM_CLKSEL_VIDEO2_MCASP_AUX 0x3: Selects divided version of L4_ICLK. See CM_CLKSEL_HDMI_MCASP_AUX	RW	0x0
21:18	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guaranteed to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-487. Register Call Summary for Register CM_IPU_MCASP1_CLKCTRL

Clock Management Functional Description
- CM_CORE_AON_MCASP1 Overview: [0][1][2]
- Clock Domain Module Attributes: [3][4]

PRCM Register Manual
- IPU_CM_CORE_AON Register Summary: [5]

Table 3-488. CM_IPU_TIMER5_CLKCTRL

Address Offset	0x0000 0058		
Physical Address	0x4A00 5558	**Instance**	IPU_CM_CORE_AON
Description	This register manages the TIMER5 clocks.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				CLKSEL				RESERVED						IDLEST		RESERVED														MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:24	CLKSEL	Select the source of the functional clock 0x0: Selects TIMER_SYS_CLK 0x1: Selects FUNC_32K_CLK 0x2: Selects SYS_CLK2 0x3: Selects XREF_CLK0 0x4: Selects XREF_CLK1 0x5: Selects XREF_CLK0 0x6: Selects XREF_CLK1 0x7: Selects ABE_GICLK 0x8: Selects divided version of SYS_CLK1. See CM_CLKSEL_VIDEO1_TIMER 0x9: Selects divided version of SYS_CLK1. See CM_CLKSEL_VIDEO2_TIMER 0xA: Selects divided version of SYS_CLK1. See CM_CLKSEL_HDMI_TIMER 0xB: Selects CLKOUTMUX0 Others: RESERVED	RW	0x0
23:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guaranteed to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-489. Register Call Summary for Register CM_IPU_TIMER5_CLKCTRL

Clock Management Functional Description
- CM_CORE_AON_CLKOUTMUX Overview: [0]
- CM_CORE_AON_TIMER Overview: [1]
- Clock Domain Module Attributes: [2][3]

PRCM Register Manual
- IPU_CM_CORE_AON Register Summary: [4]

Table 3-490. CM_IPU_TIMER6_CLKCTRL

Address Offset	0x0000 0060		
Physical Address	0x4A00 5560	**Instance**	IPU_CM_CORE_AON
Description	This register manages the TIMER6 clocks.		
Type	RW		

31 30 29 28	27 26 25 24	23 22 21 20 19 18	17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2	1 0
RESERVED	CLKSEL	RESERVED	IDLEST	RESERVED		MODULEMODE

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:24	CLKSEL	Select the source of the functional clock 0x0: Selects TIMER_SYS_CLK 0x1: Selects FUNC_32K_CLK 0x2: Selects SYS_CLK2 0x3: Selects XREF_CLK0 0x4: Selects XREF_CLK1 0x5: Selects XREF_CLK0 0x6: Selects XREF_CLK1 0x7: Selects ABE_GICLK 0x8: Selects divided version of SYS_CLK1. See CM_CLKSEL_VIDEO1_TIMER 0x9: Selects divided version of SYS_CLK1. See CM_CLKSEL_VIDEO2_TIMER 0xA: Selects divided version of SYS_CLK1. See CM_CLKSEL_HDMI_TIMER 0xB: Selects CLKOUTMUX0 Others: RESERVED	RW	0x0
23:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guaranteed to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-491. Register Call Summary for Register CM_IPU_TIMER6_CLKCTRL

Clock Management Functional Description
- CM_CORE_AON_CLKOUTMUX Overview: [0]
- CM_CORE_AON_TIMER Overview: [1]
- Clock Domain Module Attributes: [2][3]

PRCM Register Manual
- IPU_CM_CORE_AON Register Summary: [4]

Table 3-492. CM_IPU_TIMER7_CLKCTRL

Address Offset	0x0000 0068		
Physical Address	0x4A00 5568	**Instance**	IPU_CM_CORE_AON
Description	This register manages the TIMER7 clocks.		
Type	RW		

31 30 29 28	27 26 25 24	23 22 21 20 19 18	17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2	1 0
RESERVED	CLKSEL	RESERVED	IDLEST	RESERVED		MODULEMODE

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:24	CLKSEL	Select the source of the functional clock 0x0: Selects TIMER_SYS_CLK 0x1: Selects FUNC_32K_CLK 0x2: Selects SYS_CLK2 0x3: Selects XREF_CLK0 0x4: Selects XREF_CLK1 0x5: Selects XREF_CLK0 0x6: Selects XREF_CLK1 0x7: Selects ABE_GICLK 0x8: Selects divided version of SYS_CLK1. See CM_CLKSEL_VIDEO1_TIMER 0x9: Selects divided version of SYS_CLK1. See CM_CLKSEL_VIDEO2_TIMER 0xA: Selects divided version of SYS_CLK1. See CM_CLKSEL_HDMI_TIMER 0xB: Selects CLKOUTMUX0 Others: RESERVED	RW	0x0
23:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guaranteed to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-493. Register Call Summary for Register CM_IPU_TIMER7_CLKCTRL

Clock Management Functional Description • CM_CORE_AON_CLKOUTMUX Overview: [0] • CM_CORE_AON_TIMER Overview: [1] • Clock Domain Module Attributes: [2][3]
PRCM Register Manual • IPU_CM_CORE_AON Register Summary: [4]

Table 3-494. CM_IPU_TIMER8_CLKCTRL

Address Offset	0x0000 0070		
Physical Address	0x4A00 5570	**Instance**	IPU_CM_CORE_AON
Description	This register manages the TIMER8 clocks.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				CLKSEL				RESERVED						IDLEST		RESERVED														MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:24	CLKSEL	Select the source of the functional clock 0x0: Selects TIMER_SYS_CLK 0x1: Selects FUNC_32K_CLK 0x2: Selects SYS_CLK2 0x3: Selects XREF_CLK0 0x4: Selects XREF_CLK1 0x5: Selects XREF_CLK0 0x6: Selects XREF_CLK1 0x7: Selects ABE_GICLK 0x8: Selects divided version of SYS_CLK1. See CM_CLKSEL_VIDEO1_TIMER 0x9: Selects divided version of SYS_CLK1. See CM_CLKSEL_VIDEO2_TIMER 0xA: Selects divided version of SYS_CLK1. See CM_CLKSEL_HDMI_TIMER 0xB: Selects CLKOUTMUX0 Others: RESERVED	RW	0x0
23:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guaranteed to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-495. Register Call Summary for Register CM_IPU_TIMER8_CLKCTRL

Clock Management Functional Description
- CM_CORE_AON_CLKOUTMUX Overview: [0]
- CM_CORE_AON_TIMER Overview: [1]
- Clock Domain Module Attributes: [2][3]

PRCM Register Manual
- IPU_CM_CORE_AON Register Summary: [4]

Table 3-496. CM_IPU_I2C5_CLKCTRL

Address Offset	0x0000 0078		
Physical Address	0x4A00 5578	**Instance**	IPU_CM_CORE_AON
Description	This register manages the I2C5 clocks.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														IDLEST		RESERVED								MODULEMODE							

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guaranteed to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-497. CM_IPU_UART6_CLKCTRL

Address Offset	0x0000 0080		
Physical Address	0x4A00 5580	**Instance**	IPU_CM_CORE_AON
Description	This register manages the UART6 clocks.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED							CLKSEL	RESERVED						IDLEST		RESERVED														MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24	CLKSEL	Selects functional clock for UART between FUNCTION_48M_CLK and FUNC_192M_CLK 0x0: Selects FUNC_48M_CLK 0x1: Selects FUNC_192M_CLK	RW	0x0
23:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guaranteed to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

3.12.12 MPU_CM_CORE_AON registers

3.12.12.1 MPU_CM_CORE_AON Register Summary

Table 3-498. MPU_CM_CORE_AON Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	MPU_CM_CORE_AON Physical Address
CM_MPU_CLKSTCTRL	RW	32	0x0000 0000	0x4A00 5300
CM_MPU_STATICDEP	RW	32	0x0000 0004	0x4A00 5304
CM_MPU_DYNAMICDEP	RW	32	0x0000 0008	0x4A00 5308
CM_MPU_MPU_CLKCTRL	RW	32	0x0000 0020	0x4A00 5320
CM_MPU_MPU_MPU_DBG_CLKCTRL	R	32	0x0000 0028	0x4A00 5328

3.12.12.2 MPU_CM_CORE_AON Register Description

Table 3-499. CM_MPU_CLKSTCTRL

Address Offset	0x0000 0000		
Physical Address	0x4A00 5300	**Instance**	MPU_CM_CORE_AON
Description	This register enables the MPU domain power state transition. It controls the HW supervised domain power state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							CLKACTIVITY_MPU_GCLK	RESERVED						CLKTRCTRL	

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	CLKACTIVITY_MPU_GCLK	This field indicates the state of the MPU_DPLL_CLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
7:2	RESERVED		R	0x0
1:0	CLKTRCTRL	Controls the clock state transition of the MPU clock domain. 0x0: NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may however occur. 0x1: Reserved 0x2: SW_WKUP: Start a software forced wake-up transition on the domain. 0x3: HW_AUTO: Automatic transition is enabled. Sleep and wakeup transition are based upon hardware conditions.	RW	0x0

Table 3-500. CM_MPU_STATICDEP

Address Offset	0x0000 0004		
Physical Address	0x4A00 5304	**Instance**	MPU_CM_CORE_AON
Description	This register controls the static domain depedencies from MPU domain towards 'target' domains. It is relevant only for domain having system initiator(s).		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED		PCIE_STATDEP	ISS_STATDEP	L4PER3_STATDEP	L4PER2_STATDEP	GMAC_STATDEP	IPU_STATDEP	IPU1_STATDEP	EVE4_STATDEP	EVE3_STATDEP	EVE2_STATDEP	EVE1_STATDEP	DSP2_STATDEP	CUSTEFUSE_STATDEP	COREAON_STATDEP	WKUPAON_STATDEP	L4SEC_STATDEP	L4PER_STATDEP	L4CFG_STATDEP	SDMA_STATDEP	GPU_STATDEP	CAM_STATDEP	DSS_STATDEP	L3INIT_STATDEP	RESERVED	L3MAIN1_STATDEP	EMIF_STATDEP	RESERVED	IVA_STATDEP	DSP1_STATDEP	IPU2_STATDEP

Bits	Field Name	Description	Type	Reset
31:30	RESERVED		R	0x0
29	PCIE_STATDEP	Static dependency towards PCIE clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
28	ISS_STATDEP	Static dependency towards ISS clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
27	L4PER3_STATDEP	Static dependency towards L4PER3 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
26	L4PER2_STATDEP	Static dependency towards L4PER2 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
25	GMAC_STATDEP	Static dependency towards GMAC clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
24	IPU_STATDEP	Static dependency towards IPU clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
23	IPU1_STATDEP	Static dependency towards IPU1 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
22	EVE4_STATDEP	Static dependency towards EVE4 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
21	EVE3_STATDEP	Static dependency towards EVE3 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
20	EVE2_STATDEP	Static dependency towards EVE2 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
19	EVE1_STATDEP	Static dependency towards EVE1 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
18	DSP2_STATDEP	Static dependency towards DSP2 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
17	CUSTEFUSE_STATDEP	Static dependency towards CUSTEFUSE clock domain 0x0: Dependency is disabled	R	0x0
16	COREAON_STATDEP	Static dependency towards COREAON clock domain 0x0: Dependency is disabled	R	0x0
15	WKUPAON_STATDEP	Static dependency towards WKUPAON clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
14	L4SEC_STATDEP	Static dependency towards L4SEC clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
13	L4PER_STATDEP	Static dependency towards L4PER clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
12	L4CFG_STATDEP	Static dependency towards L4CFG clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
11	SDMA_STATDEP	Static dependency towards SDMA clock domain 0x0: Dependency is disabled	R	0x0
10	GPU_STATDEP	Static dependency towards GPU clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
9	CAM_STATDEP	Static dependency towards CAM clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
8	DSS_STATDEP	Static dependency towards DSS clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
7	L3INIT_STATDEP	Static dependency towards L3INIT clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
6	RESERVED		R	0x0
5	L3MAIN1_STATDEP	Static dependency towards L3MAIN clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
4	EMIF_STATDEP	Static dependency towards EMIF clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
3	RESERVED		R	0x0
2	IVA_STATDEP	Static dependency towards IVA clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
1	DSP1_STATDEP	Static dependency towards DSP1 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	IPU2_STATDEP	Static dependency towards IPU2 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-501. CM_MPU_DYNAMICDEP

Address Offset	0x0000 0008		
Physical Address	0x4A00 5308	**Instance**	MPU_CM_CORE_AON
Description	This register controls the dynamic domain depedencies from MPU domain towards 'target' domains. It is relevant only for domain having OCP master port(s).		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				WINDOWSIZE				RESERVED																		L3MAIN1_DYNDEP	EMIF_DYNDEP	RESERVED				

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:24	WINDOWSIZE	Size of sliding window used to monitor OCP interface activity for determination of auto-sleep feature. Time unit defined by CM_DYN_DEP_PRESCAL register.	RW	0x4
23:6	RESERVED		R	0x0
5	L3MAIN1_DYNDEP	Dynamic dependency towards L3MAIN1 clock domain 0x1: Dependency is enabled	R	0x1
4	EMIF_DYNDEP	Dynamic dependency towards EMIF clock domain 0x1: Dependency is enabled	R	0x1
3:0	RESERVED		R	0x0

Table 3-502. CM_MPU_MPU_CLKCTRL

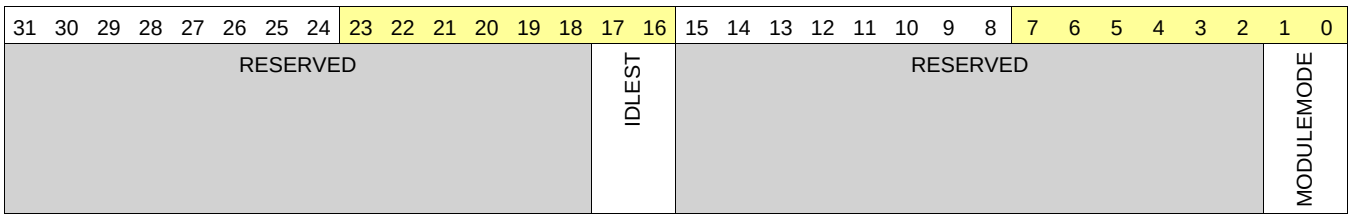
Address Offset	0x0000 0020		
Physical Address	0x4A00 5320	**Instance**	MPU_CM_CORE_AON
Description	This register manages the MPU clocks.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED					CLKSEL_ABE_DIV_MODE	CLKSEL_EMIF_DIV_MODE		RESERVED					STBYST	IDLEST		RESERVED																	MODULEMODE

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26	CLKSEL_ABE_DIV_MODE	Selects the ratio for MPU - ABE async bridge versus MPU DPLL clock 0x0: MPU DPLL clock divided by 8 0x1: MPU DPLL clock divided by 16	RW	0x0
25:24	CLKSEL_EMIF_DIV_MODE	Selects the ratio for MPU - L3 async bridge versus MPU DPLL clock 0x0: MPU DPLL clock divided by 4 0x1: MPU DPLL clock divided by 4 0x2: MPU DPLL clock divided by 8 0x3: MPU DPLL clock divided by 8	RW	0x0
23:19	RESERVED		R	0x0
18	STBYST	Module standby status. [warm reset insensitive] 0x0: Module is functional (not in standby) 0x1: Module is in standby	R	0x1
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. Module clocks may be gated according to the clock domain state.	R	0x1

Table 3-503. CM_MPU_MPU_MPU_DBG_CLKCTRL

Address Offset	0x0000 0028		
Physical Address	0x4A00 5328	**Instance**	MPU_CM_CORE_AON
Description	This register manages the MPU_MPU_DBG clocks. [warm reset insensitive]		
Type	R		



Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state.	R	0x1

3.12.13 OCP_SOCKET_CM_CORE_AON registers

3.12.13.1 OCP_SOCKET_CM_CORE_AON Register Summary

Table 3-504. OCP_SOCKET_CM_CORE_AON Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	OCP_SOCKET_CM_CORE_AON Physical Address
REVISION_CM_CORE_AON	R	32	0x0000 0000	0x4A00 5000
CM_CM_CORE_AON_PROFILING_CLKCTRL	RW	32	0x0000 0040	0x4A00 5040
CM_CORE_AON_DEBUG_OUT	R	32	0x0000 00EC	0x4A00 50EC
CM_CORE_AON_DEBUG_CFG0	RW	32	0x0000 00F0	0x4A00 50F0
CM_CORE_AON_DEBUG_CFG1	RW	32	0x0000 00F4	0x4A00 50F4
CM_CORE_AON_DEBUG_CFG2	RW	32	0x0000 00F8	0x4A00 50F8
CM_CORE_AON_DEBUG_CFG3	RW	32	0x0000 00FC	0x4A00 50FC

3.12.13.2 OCP_SOCKET_CM_CORE_AON Register Description

Table 3-505. REVISION_CM_CORE_AON

Address Offset	0x0000 0000		
Physical Address	0x4A00 5000	**Instance**	OCP_SOCKET_CM_CORE_AON
Description	This register contains the IP revision code for the CM_CORE_AON part of the PRCM		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCHEME								RESERVED								FUNC								R_RTL				X_MAJOR				CUSTOM						Y_MINOR			

Bits	Field Name	Description	Type	Reset
31:30	SCHEME		R	0x1
29:28	RESERVED		R	0x0
27:16	FUNC	Function indicates a software compatible module family.	R	0x0
15:11	R_RTL	RTL Version (R)	R	0x0
10:8	X_MAJOR	Major Revision (X)	R	0x3
7:6	CUSTOM	Indicates a special version for a particular device. 0x0: Non custom (standard) revision	R	0x0
5:0	Y_MINOR	Minor Revision (Y)	R	0x1

Table 3-506. Register Call Summary for Register REVISION_CM_CORE_AON

PRCM Register Manual
- OCP_SOCKET_CM_CORE_AON Register Summary: [0]

Table 3-507. CM_CM_CORE_AON_PROFILING_CLKCTRL

Address Offset	0x0000 0040		
Physical Address	0x4A00 5040	**Instance**	OCP_SOCKET_CM_CORE_AON
Description	This register manages the CM_CORE_AON_PROFILING clock. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														IDLEST		RESERVED														MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status 0x0: Module is fully functional 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle 0x3: Module is disabled	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. OCP configuration port is not accessible. 0x1: Module is managed automatically by HW along with CM_CORE_AON and EMU domain. OCP configuration port is accessible only when EMU domain is on. 0x2: Reserved 0x3: Reserved	RW	0x1

Table 3-508. Register Call Summary for Register CM_CM_CORE_AON_PROFILING_CLKCTRL

PRCM Register Manual
- OCP_SOCKET_CM_CORE_AON Register Summary: [0]
- RESTORE_CM_CORE_AON Register Description: [1][2]

Table 3-509. CM_CORE_AON_DEBUG_OUT

Address Offset	0x0000 00EC		
Physical Address	0x4A00 50EC	**Instance**	OCP_SOCKET_CM_CORE_AON
Description	This register is used to monitor the CM_COREAON's 32 bit HEDEBUG BUS [warm reset insensitive]		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OUTPUT																															

Bits	Field Name	Description	Type	Reset
31:0	OUTPUT	HW DEBUG OUTPUT	R	0x0

Table 3-510. Register Call Summary for Register CM_CORE_AON_DEBUG_OUT

PRCM Register Manual
- OCP_SOCKET_CM_CORE_AON Register Summary: [0]

Table 3-511. CM_CORE_AON_DEBUG_CFG0

Address Offset	0x0000 00F0		
Physical Address	0x4A00 50F0	**Instance**	OCP_SOCKET_CM_CORE_AON
Description	This register is used to configure the CM_CORE_AON's 32-bit debug output. There is one 7-bit source select field for selecting from a shared set of 8-bit internal signal blocks per byte. The signals included in each block are specified in the PRCM integration specification. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						SEL0									

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9:0	SEL0	Internal signal block select for debug word byte-0	RW	0x0

Table 3-512. Register Call Summary for Register CM_CORE_AON_DEBUG_CFG0

PRCM Register Manual
- OCP_SOCKET_CM_CORE_AON Register Summary: [0]

Table 3-513. CM_CORE_AON_DEBUG_CFG1

Address Offset	0x0000 00F4		
Physical Address	0x4A00 50F4	**Instance**	OCP_SOCKET_CM_CORE_AON
Description	This register is used to configure the CM_CORE_AON's 32-bit debug output. There is one 7-bit source select field for selecting from a shared set of 8-bit internal signal blocks per byte. The signals included in each block are specified in the PRCM integration specification. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						SEL1									

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9:0	SEL1	Internal signal block select for debug word byte-1	RW	0x0

Table 3-514. Register Call Summary for Register CM_CORE_AON_DEBUG_CFG1

PRCM Register Manual
- OCP_SOCKET_CM_CORE_AON Register Summary: [0]

Table 3-515. CM_CORE_AON_DEBUG_CFG2

Address Offset	0x0000 00F8		
Physical Address	0x4A00 50F8	**Instance**	OCP_SOCKET_CM_CORE_AON
Description	This register is used to configure the CM_CORE_AON's 32-bit debug output. There is one 7-bit source select field for selecting from a shared set of 8-bit internal signal blocks per byte. The signals included in each block are specified in the PRCM integration specification. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						SEL2									

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9:0	SEL2	Internal signal block select for debug word byte-2	RW	0x0

Table 3-516. Register Call Summary for Register CM_CORE_AON_DEBUG_CFG2

PRCM Register Manual
- OCP_SOCKET_CM_CORE_AON Register Summary: [0]

Table 3-517. CM_CORE_AON_DEBUG_CFG3

Address Offset	0x0000 00FC		
Physical Address	0x4A00 50FC	**Instance**	OCP_SOCKET_CM_CORE_AON
Description	This register is used to configure the CM_CORE_AON's 32-bit debug output. There is one 7-bit source select field for selecting from a shared set of 8-bit internal signal blocks per byte. The signals included in each block are specified in the PRCM integration specification. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						SEL3									

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9:0	SEL3	Internal signal block select for debug word byte-3	RW	0x0

Table 3-518. Register Call Summary for Register CM_CORE_AON_DEBUG_CFG3

PRCM Register Manual
- OCP_SOCKET_CM_CORE_AON Register Summary: [0]

3.12.14 RESTORE_CM_CORE_AON registers

3.12.14.1 RESTORE_CM_CORE_AON Register Summary

Table 3-519. RESTORE_CM_CORE_AON Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	RESTORE_CM_CORE_AON Physical Address
CM_CLKSEL_CORE_RESTORE	RW	32	0x0000 0000	0x4A00 5E00
CM_DIV_M2_DPLL_CORE_RESTORE	RW	32	0x0000 0004	0x4A00 5E04
CM_DIV_M3_DPLL_CORE_RESTORE	RW	32	0x0000 0008	0x4A00 5E08
CM_DIV_H11_DPLL_CORE_RESTORE	RW	32	0x0000 000C	0x4A00 5E0C
CM_DIV_H12_DPLL_CORE_RESTORE	RW	32	0x0000 0010	0x4A00 5E10
CM_DIV_H13_DPLL_CORE_RESTORE	RW	32	0x0000 0014	0x4A00 5E14
CM_DIV_H14_DPLL_CORE_RESTORE	RW	32	0x0000 0018	0x4A00 5E18
CM_DIV_H21_DPLL_CORE_RESTORE	RW	32	0x0000 001C	0x4A00 5E1C
CM_DIV_H22_DPLL_CORE_RESTORE	RW	32	0x0000 0020	0x4A00 5E20
CM_DIV_H23_DPLL_CORE_RESTORE	RW	32	0x0000 0024	0x4A00 5E24
CM_DIV_H24_DPLL_CORE_RESTORE	RW	32	0x0000 0028	0x4A00 5E28
CM_CLKSEL_DPLL_CORE_RESTORE	RW	32	0x0000 002C	0x4A00 5E2C
CM_SSC_DELTAMSTEP_DPLL_CORE_RESTORE	RW	32	0x0000 0030	0x4A00 5E30
CM_SSC_MODFREQDIV_DPLL_CORE_RESTORE	RW	32	0x0000 0034	0x4A00 5E34
CM_CLKMODE_DPLL_CORE_RESTORE	RW	32	0x0000 0038	0x4A00 5E38
CM_SHADOW_FREQ_CONFIG2_RESTORE	RW	32	0x0000 003C	0x4A00 5E3C
CM_SHADOW_FREQ_CONFIG1_RESTORE	RW	32	0x0000 0040	0x4A00 5E40
CM_AUTOIDLE_DPLL_CORE_RESTORE	RW	32	0x0000 0044	0x4A00 5E44
CM_MPU_CLKSTCTRL_RESTORE	RW	32	0x0000 0048	0x4A00 5E48
CM_CM_CORE_AON_PROFILING_CLKCTRL_RESTORE	RW	32	0x0000 004C	0x4A00 5E4C
CM_DYN_DEP_PRESCAL_RESTORE	RW	32	0x0000 0050	0x4A00 5E50

3.12.14.2 RESTORE_CM_CORE_AON Register Description

Table 3-520. CM_CLKSEL_CORE_RESTORE

Address Offset	0x0000 0000		
Physical Address	0x4A00 5E00	**Instance**	RESTORE_CM_CORE_AON
Description	Second address map for register CM_CLKSEL_CORE. Used only by automatic restore upon wakeup from device OFF mode.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESTORE																															

Bits	Field Name	Description	Type	Reset
31:0	RESTORE	See CM_CLKSEL_CORE register.	RW	0x0

Table 3-521. Register Call Summary for Register CM_CLKSEL_CORE_RESTORE

PRCM Register Manual • RESTORE_CM_CORE_AON Register Summary: [0]

Table 3-522. CM_DIV_M2_DPLL_CORE_RESTORE

Address Offset	0x0000 0004		
Physical Address	0x4A00 5E04	**Instance**	RESTORE_CM_CORE_AON
Description	Second address map for register CM_DIV_M2_DPLL_CORE. Used only by automatic restore upon wakeup from device OFF mode.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESTORE																															

Bits	Field Name	Description	Type	Reset
31:0	RESTORE	See CM_DIV_M2_DPLL_CORE register.	RW	0x1

Table 3-523. Register Call Summary for Register CM_DIV_M2_DPLL_CORE_RESTORE

PRCM Register Manual • RESTORE_CM_CORE_AON Register Summary: [0]

Table 3-524. CM_DIV_M3_DPLL_CORE_RESTORE

Address Offset	0x0000 0008		
Physical Address	0x4A00 5E08	**Instance**	RESTORE_CM_CORE_AON
Description	Second address map for register CM_DIV_M3_DPLL_CORE. Used only by automatic restore upon wakeup from device OFF mode.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESTORE																															

Bits	Field Name	Description	Type	Reset
31:0	RESTORE	See CM_DIV_M3_DPLL_CORE register.	RW	0x1

Table 3-525. Register Call Summary for Register CM_DIV_M3_DPLL_CORE_RESTORE

PRCM Register Manual • RESTORE_CM_CORE_AON Register Summary: [0]

Table 3-526. CM_DIV_H11_DPLL_CORE_RESTORE

Address Offset	0x0000 000C		
Physical Address	0x4A00 5E0C	**Instance**	RESTORE_CM_CORE_AON
Description	Second address map for register CM_DIV_H11_DPLL_CORE. Used only by automatic restore upon wakeup from device OFF mode.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESTORE																															

Bits	Field Name	Description	Type	Reset
31:0	RESTORE	See CM_DIV_H11_DPLL_CORE register.	RW	0x4

Table 3-527. Register Call Summary for Register CM_DIV_H11_DPLL_CORE_RESTORE

PRCM Register Manual
- RESTORE_CM_CORE_AON Register Summary: [0]

Table 3-528. CM_DIV_H12_DPLL_CORE_RESTORE

Address Offset	0x0000 0010		
Physical Address	0x4A00 5E10	**Instance**	RESTORE_CM_CORE_AON
Description	Second address map for register CM_DIV_H12_DPLL_CORE. Used only by automatic restore upon wakeup from device OFF mode.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESTORE																															

Bits	Field Name	Description	Type	Reset
31:0	RESTORE	See CM_DIV_H12_DPLL_CORE register.	RW	0x4

Table 3-529. Register Call Summary for Register CM_DIV_H12_DPLL_CORE_RESTORE

PRCM Register Manual
- RESTORE_CM_CORE_AON Register Summary: [0]

Table 3-530. CM_DIV_H13_DPLL_CORE_RESTORE

Address Offset	0x0000 0014		
Physical Address	0x4A00 5E14	**Instance**	RESTORE_CM_CORE_AON
Description	Second address map for register CM_DIV_H13_DPLL_CORE. Used only by automatic restore upon wakeup from device OFF mode.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESTORE																															

Bits	Field Name	Description	Type	Reset
31:0	RESTORE	See CM_DIV_H12_DPLL_CORE register.	RW	0x4

Table 3-531. Register Call Summary for Register CM_DIV_H13_DPLL_CORE_RESTORE

PRCM Register Manual
- RESTORE_CM_CORE_AON Register Summary: [0]

Table 3-532. CM_DIV_H14_DPLL_CORE_RESTORE

Address Offset	0x0000 0018		
Physical Address	0x4A00 5E18	**Instance**	RESTORE_CM_CORE_AON
Description	Second address map for register CM_DIV_H14_DPLL_CORE. Used only by automatic restore upon wakeup from device OFF mode.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESTORE																															

Bits	Field Name	Description	Type	Reset
31:0	RESTORE	See CM_DIV_H14_DPLL_CORE register.	RW	0x4

Table 3-533. Register Call Summary for Register CM_DIV_H14_DPLL_CORE_RESTORE

PRCM Register Manual
- RESTORE_CM_CORE_AON Register Summary: [0]

Table 3-534. CM_DIV_H21_DPLL_CORE_RESTORE

Address Offset	0x0000 001C		
Physical Address	0x4A00 5E1C	**Instance**	RESTORE_CM_CORE_AON
Description	Second address map for register CM_DIV_H21_DPLL_CORE. Used only by automatic restore upon wakeup from device OFF mode.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESTORE																															

Bits	Field Name	Description	Type	Reset
31:0	RESTORE	See CM_DIV_H21_DPLL_CORE register.	RW	0x4

Table 3-535. Register Call Summary for Register CM_DIV_H21_DPLL_CORE_RESTORE

PRCM Register Manual
- RESTORE_CM_CORE_AON Register Summary: [0]

Table 3-536. CM_DIV_H22_DPLL_CORE_RESTORE

Address Offset	0x0000 0020		
Physical Address	0x4A00 5E20	**Instance**	RESTORE_CM_CORE_AON
Description	Second address map for register CM_DIV_H22_DPLL_CORE. Used only by automatic restore upon wakeup from device OFF mode.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESTORE																															

Bits	Field Name	Description	Type	Reset
31:0	RESTORE	See CM_DIV_H22_DPLL_CORE register.	RW	0x4

Table 3-537. Register Call Summary for Register CM_DIV_H22_DPLL_CORE_RESTORE

PRCM Register Manual

- RESTORE_CM_CORE_AON Register Summary: [0]

Table 3-538. CM_DIV_H23_DPLL_CORE_RESTORE

Address Offset	0x0000 0024		
Physical Address	0x4A00 5E24	**Instance**	RESTORE_CM_CORE_AON
Description	Second address map for register CM_DIV_H23_DPLL_CORE. Used only by automatic restore upon wakeup from device OFF mode.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESTORE																															

Bits	Field Name	Description	Type	Reset
31:0	RESTORE	See CM_DIV_H23_DPLL_CORE register.	RW	0x8

Table 3-539. Register Call Summary for Register CM_DIV_H23_DPLL_CORE_RESTORE

PRCM Register Manual

- RESTORE_CM_CORE_AON Register Summary: [0]

Table 3-540. CM_DIV_H24_DPLL_CORE_RESTORE

Address Offset	0x0000 0028		
Physical Address	0x4A00 5E28	**Instance**	RESTORE_CM_CORE_AON
Description	Second address map for register CM_DIV_H24_DPLL_CORE. Used only by automatic restore upon wakeup from device OFF mode.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESTORE																															

Bits	Field Name	Description	Type	Reset
31:0	RESTORE	See CM_DIV_H24_DPLL_CORE register.	RW	0x8

Table 3-541. Register Call Summary for Register CM_DIV_H24_DPLL_CORE_RESTORE

PRCM Register Manual

- RESTORE_CM_CORE_AON Register Summary: [0]

Table 3-542. CM_CLKSEL_DPLL_CORE_RESTORE

Address Offset	0x0000 002C		
Physical Address	0x4A00 5E2C	**Instance**	RESTORE_CM_CORE_AON
Description	Second address map for register CM_CLKSEL_DPLL_CORE. Used only by automatic restore upon wakeup from device OFF mode.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESTORE																															

Bits	Field Name	Description	Type	Reset
31:0	RESTORE	See CM_CLKSEL_DPLL_CORE register.	RW	0x0

Table 3-543. Register Call Summary for Register CM_CLKSEL_DPLL_CORE_RESTORE

Clock Management Functional Description • DPLL_CORE Synthesized Clock Parameters: [0][1]
PRCM Register Manual • RESTORE_CM_CORE_AON Register Summary: [2]

Table 3-544. CM_SSC_DELTAMSTEP_DPLL_CORE_RESTORE

Address Offset	0x0000 0030		
Physical Address	0x4A00 5E30	**Instance**	RESTORE_CM_CORE_AON
Description	Second address map for register CM_SSC_DELTAMSTEP_DPLL_CORE. Used only by automatic restore upon wakeup from device OFF mode. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESTORE																															

Bits	Field Name	Description	Type	Reset
31:0	RESTORE	See CM_SSC_DELTAMSTEP_DPLL_CORE register.	RW	0x0

Table 3-545. CM_SSC_MODFREQDIV_DPLL_CORE_RESTORE

Address Offset	0x0000 0034		
Physical Address	0x4A00 5E34	**Instance**	RESTORE_CM_CORE_AON
Description	Second address map for register CM_SSC_MODFREQDIV_DPLL_CORE. Used only by automatic restore upon wakeup from device OFF mode. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESTORE																															

Bits	Field Name	Description	Type	Reset
31:0	RESTORE	See CM_SSC_MODFREQDIV_DPLL_CORE register.	RW	0x0

Table 3-546. CM_CLKMODE_DPLL_CORE_RESTORE

Address Offset	0x0000 0038		
Physical Address	0x4A00 5E38	**Instance**	RESTORE_CM_CORE_AON
Description	Second address map for register CM_CLKMODE_DPLL_CORE. Used only by automatic restore upon wakeup from device OFF mode.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESTORE																															

Bits	Field Name	Description	Type	Reset
31:0	RESTORE	See CM_CLKMODE_DPLL_CORE register.	RW	0x4

Table 3-547. Register Call Summary for Register CM_CLKMODE_DPLL_CORE_RESTORE

Clock Management Functional Description • DPLL_CORE Power Modes: [0][1]
PRCM Register Manual • RESTORE_CM_CORE_AON Register Summary: [2]

Table 3-548. CM_SHADOW_FREQ_CONFIG2_RESTORE

Address Offset	0x0000 003C		
Physical Address	0x4A00 5E3C	**Instance**	RESTORE_CM_CORE_AON
Description	Second address map for register CM_SHADOW_FREQ_CONFIG2. Used only by automatic restore upon wakeup from device OFF mode.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESTORE																															

Bits	Field Name	Description	Type	Reset
31:0	RESTORE	See CM_SHADOW_FREQ_CONFIG2 register.	RW	0x20

Table 3-549. Register Call Summary for Register CM_SHADOW_FREQ_CONFIG2_RESTORE

PRCM Register Manual • RESTORE_CM_CORE_AON Register Summary: [0]

Table 3-550. CM_SHADOW_FREQ_CONFIG1_RESTORE

Address Offset	0x0000 0040		
Physical Address	0x4A00 5E40	**Instance**	RESTORE_CM_CORE_AON
Description	Second address map for register CM_SHADOW_FREQ_CONFIG1. Used only by automatic restore upon wakeup from device OFF mode.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESTORE																															

Bits	Field Name	Description	Type	Reset
31:0	RESTORE	See CM_SHADOW_FREQ_CONFIG1 register.	RW	0xc0c

Table 3-551. Register Call Summary for Register CM_SHADOW_FREQ_CONFIG1_RESTORE

PRCM Register Manual • RESTORE_CM_CORE_AON Register Summary: [0]

Table 3-552. CM_AUTOIDLE_DPLL_CORE_RESTORE

Address Offset	0x0000 0044		
Physical Address	0x4A00 5E44	**Instance**	RESTORE_CM_CORE_AON
Description	Second address map for register CM_AUTOIDLE_DPLL_CORE. Used only by automatic restore upon wakeup from device OFF mode.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESTORE																															

Bits	Field Name	Description	Type	Reset
31:0	RESTORE	See CM_AUTOIDLE_DPLL_COREregister.	RW	0x0

Table 3-553. Register Call Summary for Register CM_AUTOIDLE_DPLL_CORE_RESTORE

Clock Management Functional Description
- DPLL_CORE Power Modes: [0]

PRCM Register Manual
- RESTORE_CM_CORE_AON Register Summary: [1]

Table 3-554. CM_MPU_CLKSTCTRL_RESTORE

Address Offset	0x0000 0048		
Physical Address	0x4A00 5E48	**Instance**	RESTORE_CM_CORE_AON
Description	Second address map for register CM_MPU_CLKSTCTRL. Used only by automatic restore upon wakeup from device OFF mode.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESTORE																															

Bits	Field Name	Description	Type	Reset
31:0	RESTORE	See CM_MPU_CLKSTCTRL register.	RW	0x0

Table 3-555. CM_CM_CORE_AON_PROFILING_CLKCTRL_RESTORE

Address Offset	0x0000 004C		
Physical Address	0x4A00 5E4C	**Instance**	RESTORE_CM_CORE_AON
Description	Second address map for register CM_CM_CORE_AON_PROFILING_CLKCTRL. Used only by automatic restore upon wakeup from device OFF mode. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESTORE																															

Bits	Field Name	Description	Type	Reset
31:0	RESTORE	See CM_CM_CORE_AON_PROFILING_CLKCTRL register.	RW	0x30001

Table 3-556. Register Call Summary for Register CM_CM_CORE_AON_PROFILING_CLKCTRL_RESTORE

PRCM Register Manual • RESTORE_CM_CORE_AON Register Summary: [0]

Table 3-557. CM_DYN_DEP_PRESCAL_RESTORE

Address Offset	0x0000 0050		
Physical Address	0x4A00 5E50	**Instance**	RESTORE_CM_CORE_AON
Description	Second address map for register CM_DYN_DEP_PRESCAL. Used only by automatic restore upon wakeup from device OFF mode.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESTORE																															

Bits	Field Name	Description	Type	Reset
31:0	RESTORE	See CCM_DYN_DEP_PRESCAL register.	RW	0x20

Table 3-558. Register Call Summary for Register CM_DYN_DEP_PRESCAL_RESTORE

PRCM Register Manual • RESTORE_CM_CORE_AON Register Summary: [0]

3.12.15 RTC_CM_CORE_AON registers

3.12.15.1 RTC_CM_CORE_AON Register Summary

Table 3-559. RTC_CM_CORE_AON Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	RTC_CM_CORE_AON Physical Address
CM_RTC_CLKSTCTRL	RW	32	0x0000 0000	0x4A00 5740
CM_RTC_RTCSS_CLKCTRL	RW	32	0x0000 0004	0x4A00 5744

3.12.15.2 RTC_CM_CORE_AON Register Description

Table 3-560. CM_RTC_CLKSTCTRL

Address Offset	0x0000 0000		
Physical Address	0x4A00 5740	**Instance**	RTC_CM_CORE_AON
Description	This register enables the domain power state transition. It controls the HW supervised domain power state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																					CLKACTIVITY_RTC_AUX_CLK	RESERVED	CLKACTIVITY_RTC_L4_GICLK	RESERVED						CLKTRCTRL	

Bits	Field Name	Description	Type	Reset
31:11	RESERVED		R	0x0
10	CLKACTIVITY_RTC_AUX_CLK	This field indicates the state of the RTC_AUX_CLK in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
9	RESERVED		R	0x0
8	CLKACTIVITY_RTC_L4_GICLK	This field indicates the state of the RTC_L4_GICLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
7:2	RESERVED		R	0x0
1:0	CLKTRCTRL	Controls the clock state transition of the WKUPAON clock domain. 0x0: NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may however occur. 0x1: Reserved 0x2: SW_WKUP: Start a software forced wake-up transition on the domain. 0x3: HW_AUTO: Automatic transition is enabled. Sleep and wakeup transition are based upon hardware conditions.	RW	0x0

Table 3-561. CM_RTC_RTCSS_CLKCTRL

Address Offset	0x0000 0004		
Physical Address	0x4A00 5744	**Instance**	RTC_CM_CORE_AON
Description	This register manages the RTC clocks.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														IDLEST		RESERVED														MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guaranteed to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

3.12.16 ISS_CM_CORE_AON registers

3.12.16.1 ISS_CM_CORE_AON Register Summary

Table 3-562. ISS_CM_CORE_AON Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	ISS_CM_CORE_AON Physical Address
CM_ISS_CLKSTCTRL	RW	32	0x0000 0000	0x4A00 5760
CM_ISS_ISS_CLKCTRL	RW	32	0x0000 0004	0x4A00 5764
CM_ISS_STATICDEP	RW	32	0x0000 0008	0x4A00 5768

3.12.16.2 ISS_CM_CORE_AON Register Description

Table 3-563. CM_ISS_CLKSTCTRL

Address Offset	0x0000 0000		
Physical Address	0x4A00 5760	**Instance**	ISS_CM_CORE_AON
Description	This register enables the ISS domain power state transition. It controls the HW supervised domain power state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							CLKACTIVITY_ISS_GCLK	RESERVED						CLKTRCTRL	

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	CLKACTIVITY_ISS_GCLK	This field indicates the state of the ISS_GCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
7:2	RESERVED		R	0x0
1:0	CLKTRCTRL	Controls the clock state transition of the DSP clock domain. 0x0: NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may however occur. 0x1: SW_SLEEP: Start a software forced sleep transition on the domain. 0x2: SW_WKUP: Start a software forced wake-up transition on the domain. 0x3: HW_AUTO: Automatic transition is enabled. Sleep and wakeup transition are based upon hardware conditions.	RW	0x3

Table 3-564. Register Call Summary for Register CM_ISS_CLKSTCTRL

Clock Management Functional Description • Clock Domain Modes: [0][1]
PRCM Register Manual • ISS_CM_CORE_AON Register Summary: [2]

Table 3-565. CM_ISS_ISS_CLKCTRL

Address Offset	0x0000 0004		
Physical Address	0x4A00 5764	**Instance**	ISS_CM_CORE_AON
Description	This register manages the ISS clocks.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED													STBYST	IDLEST		RESERVED														MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:19	RESERVED		R	0x0
18	STBYST	Module standby status. [warm reset insensitive] 0x0: Module is functional (not in standby) 0x1: Module is in standby	R	0x1
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state. 0x2: Reserved 0x3: Reserved	RW	0x0

Table 3-566. Register Call Summary for Register CM_ISS_ISS_CLKCTRL

Clock Management Functional Description
- Clock Domain Module Attributes: [0][1][2]

PRCM Register Manual
- ISS_CM_CORE_AON Register Summary: [3]

Table 3-567. CM_ISS_STATICDEP

Address Offset	0x0000 0008		
Physical Address	0x4A00 5768	**Instance**	ISS_CM_CORE_AON
Description	This register controls the static domain depedencies from ISS domain towards 'target' domains. It is relevant only for domain having system initiator(s).		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				L4PER3_STATDEP	RESERVED																					L3MAIN1_STATDEP	EMIF_STATDEP	RESERVED				

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27	L4PER3_STATDEP	Static dependency towards L4PER3 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
26:6	RESERVED		R	0x0
5	L3MAIN1_STATDEP	Static dependency towards L3MAIN1 clock domain 0x1: Dependency is enabled	R	0x1
4	EMIF_STATDEP	Static dependency towards EMIF clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3:0	RESERVED		R	0x0

Table 3-568. Register Call Summary for Register CM_ISS_STATICDEP

Clock Management Functional Description • Clock Domain Dependency: [0][1][2]
PRCM Register Manual • ISS_CM_CORE_AON Register Summary: [3]

3.12.17 CAM_CM_CORE registers

3.12.17.1 CAM_CM_CORE Register Summary

Table 3-569. CAM_CM_CORE Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	CAM_CM_CORE Physical Address
CM_CAM_CLKSTCTRL	RW	32	0x0000 0000	0x4A00 9000
CM_CAM_STATICDEP	RW	32	0x0000 0004	0x4A00 9004
CM_CAM_VIP1_CLKCTRL	RW	32	0x0000 0020	0x4A00 9020
CM_CAM_VIP2_CLKCTRL	RW	32	0x0000 0028	0x4A00 9028
CM_CAM_VIP3_CLKCTRL	RW	32	0x0000 0030	0x4A00 9030
CM_CAM_LVDSRX_CLKCTRL	RW	32	0x0000 0038	0x4A00 9038
CM_CAM_CSI1_CLKCTRL	R	32	0x0000 0040	0x4A00 9040
CM_CAM_CSI2_CLKCTRL	R	32	0x0000 0048	0x4A00 9048

3.12.17.2 CAM_CM_CORE Register Description

Table 3-570. CM_CAM_CLKSTCTRL

Address Offset	0x0000 0000		
Physical Address	0x4A00 9000	**Instance**	CAM_CM_CORE
Description	This register enables the domain power state transition. It controls the HW supervised domain power state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																			CLKACTIVITY_LVDSRX_96M_GFCLK	CLKACTIVITY_LVDSRX_L4_GICLK	CLKACTIVITY_VIP3_GCLK	CLKACTIVITY_VIP2_GCLK	CLKACTIVITY_VIP1_GCLK	RESERVED						CLKTRCTRL	

Bits	Field Name	Description	Type	Reset
31:13	RESERVED		R	0x0
12	CLKACTIVITY_LVDSRX_96M_GFCLK	This field indicates the state of the LVDSRX_96M_GFCLK clock input of the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
11	CLKACTIVITY_LVDSRX_L4_GICLK	This field indicates the state of the LVDSRX_L4_GICLK clock input of the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
10	CLKACTIVITY_VIP3_GCLK	This field indicates the state of the VIP3_GCLK clock input of the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
9	CLKACTIVITY_VIP2_GCLK	This field indicates the state of the VIP2_GCLK clock input of the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
8	CLKACTIVITY_VIP1_GCLK	This field indicates the state of the VIP1_GCLK clock input of the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
7:2	RESERVED		R	0x0
1:0	CLKTRCTRL	Controls the clock state transition of the CAM clock domain. 0x0: NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may however occur. 0x1: SW_SLEEP: Start a software forced sleep transition on the domain. 0x2: SW_WKUP: Start a software forced wake-up transition on the domain. 0x3: HW_AUTO: Automatic transition is enabled. Sleep and wakeup transition are based upon hardware conditions.	RW	0x3

Table 3-571. Register Call Summary for Register CM_CAM_CLKSTCTRL

Clock Management Functional Description
- Clock Domain Modes: [0][1]

PRCM Register Manual
- Not Supported Functionality (Registers and Bitfields): [2][3]
- CAM_CM_CORE Register Summary: [4]

Table 3-572. CM_CAM_STATICDEP

Address Offset	0x0000 0004		
Physical Address	0x4A00 9004	**Instance**	CAM_CM_CORE
Description	This register controls the static domain depedencies from CAM domain towards 'target' domains. It is relevant only for domain having system initiator(s).		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED			ISS_STATDEP	L4PER3_STATDEP	RESERVED	GMAC_STATDEP	RESERVED	EVE4_STATDEP	EVE3_STATDEP	EVE2_STATDEP	EVE1_STATDEP	RESERVED				L4CFG_STATDEP	RESERVED							L3MAIN1_STATDEP	EMIF_STATDEP	RESERVED	IVA_STATDEP	RESERVED			

Bits	Field Name	Description	Type	Reset
31:29	RESERVED		R	0x0
28	ISS_STATDEP	Static dependency towards ISS clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
27	L4PER3_STATDEP	Static dependency towards L4PER3 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
26	RESERVED		R	0x0
25	GMAC_STATDEP	Static dependency towards GMAC clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
24:23	RESERVED		R	0x0
22	EVE4_STATDEP	Static dependency towards EVE4 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
21	EVE3_STATDEP	Static dependency towards EVE3 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
20	EVE2_STATDEP	Static dependency towards EVE2 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
19	EVE1_STATDEP	Static dependency towards EVE1 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
18:13	RESERVED		R	0x0
12	L4CFG_STATDEP	Static dependency towards L4CFG clock domain 0x0: Dependency is disabled	R	0x0
11:6	RESERVED		R	0x0
5	L3MAIN1_STATDEP	Static dependency towards L3MAIN1 clock domain 0x1: Dependency is enabled	R	0x1
4	EMIF_STATDEP	Static dependency towards EMIF clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
3	RESERVED		R	0x0
2	IVA_STATDEP	Static dependency towards IVA clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1:0	RESERVED		R	0x0

Table 3-573. Register Call Summary for Register CM_CAM_STATICDEP

Clock Management Functional Description

- Clock Domain Dependency: [0][1][2][3][4][5][6]

PRCM Register Manual

- Not Supported Functionality (Registers and Bitfields): [7][8][9][10]
- CAM_CM_CORE Register Summary: [11]

Table 3-574. CM_CAM_VIP1_CLKCTRL

Address Offset	0x0000 0020		
Physical Address	0x4A00 9020	**Instance**	CAM_CM_CORE
Description	This register manages the VIP1 clocks.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED							CLKSEL	RESERVED					STBYST	IDLEST		RESERVED														MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24	CLKSEL	Selects functional clock for VIP between L3_ICLK and CORE_ISS_MAIN_CLK 0x0: Selects L3_ICLK 0x1: Selects CORE_ISS_MAIN_CLK	RW	0x0
23:19	RESERVED		R	0x0
18	STBYST	Module standby status. [warm reset insensitive] 0x0: Module is functional (not in standby) 0x1: Module is in standby	R	0x1
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state. 0x2: Reserved 0x3: Reserved	RW	0x0

Table 3-575. Register Call Summary for Register CM_CAM_VIP1_CLKCTRL

Clock Management Functional Description
- CM_CORE_AON Clock Generator: [0]
- Clock Domain Module Attributes: [1][2][3]

PRCM Register Manual
- CAM_CM_CORE Register Summary: [4]

Table 3-576. CM_CAM_VIP2_CLKCTRL

Address Offset	0x0000 0028		
Physical Address	0x4A00 9028	**Instance**	CAM_CM_CORE
Description	This register manages the VIP2 clocks.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED							CLKSEL	RESERVED					STBYST	IDLEST		RESERVED														MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24	CLKSEL	Selects functional clock for VIP between L3_ICLK and CORE_ISS_MAIN_CLK 0x0: Selects L3_ICLK 0x1: Selects CORE_ISS_MAIN_CLK	RW	0x0
23:19	RESERVED		R	0x0
18	STBYST	Module standby status. [warm reset insensitive] 0x0: Module is functional (not in standby) 0x1: Module is in standby	R	0x1
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state. 0x2: Reserved 0x3: Reserved	RW	0x0

Table 3-577. CM_CAM_VIP3_CLKCTRL

Address Offset	0x0000 0030		
Physical Address	0x4A00 9030	**Instance**	CAM_CM_CORE
Description	This register manages the VIP3 clocks.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED							CLKSEL	RESERVED					STBYST	IDLEST		RESERVED														MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24	CLKSEL	Selects functional clock for VIP between L3_ICLK and CORE_ISS_MAIN_CLK 0x0: Selects L3_ICLK 0x1: Selects CORE_ISS_MAIN_CLK	RW	0x0
23:19	RESERVED		R	0x0
18	STBYST	Module standby status. [warm reset insensitive] 0x0: Module is functional (not in standby) 0x1: Module is in standby	R	0x1
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state. 0x2: Reserved 0x3: Reserved	RW	0x0

Table 3-578. CM_CAM_LVDSRX_CLKCTRL

Address Offset	0x0000 0038		
Physical Address	0x4A00 9038	**Instance**	CAM_CM_CORE
Description	This register manages the LVDSRX clocks.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														IDLEST		RESERVED								MODULEMODE							

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guaranteed to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-579. Register Call Summary for Register CM_CAM_LVDSRX_CLKCTRL

PRCM Register Manual
- CAM_CM_CORE Register Summary: [0]

Table 3-580. CM_CAM_CSI1_CLKCTRL

Address Offset	0x0000 0040		
Physical Address	0x4A00 9040	**Instance**	CAM_CM_CORE
Description	This register manages the CSI1 clocks.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED													STBYST	IDLEST		RESERVED								MODULEMODE							

Bits	Field Name	Description	Type	Reset
31:19	RESERVED		R	0x0
18	STBYST	Module standby status. [warm reset insensitive] 0x0: Module is functional (not in standby) 0x1: Module is in standby	R	0x1

Bits	Field Name	Description	Type	Reset
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state.	R	0x1

Table 3-581. Register Call Summary for Register CM_CAM_CSI1_CLKCTRL

PRCM Register Manual
- CAM_CM_CORE Register Summary: [0]

Table 3-582. CM_CAM_CSI2_CLKCTRL

Address Offset	0x0000 0048		
Physical Address	0x4A00 9048	**Instance**	CAM_CM_CORE
Description	This register manages the CSI2 clocks.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED													STBYST	IDLEST		RESERVED														MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:19	RESERVED		R	0x0
18	STBYST	Module standby status. [warm reset insensitive] 0x0: Module is functional (not in standby) 0x1: Module is in standby	R	0x1
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state.	R	0x1

Table 3-583. Register Call Summary for Register CM_CAM_CSI2_CLKCTRL

PRCM Register Manual
- CAM_CM_CORE Register Summary: [0]

3.12.18 CKGEN_CM_CORE registers

3.12.18.1 CKGEN_CM_CORE Register Summary

Table 3-584. CKGEN_CM_CORE Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	CKGEN_CM_CORE Physical Address
CM_CLKSEL_USB_60MHZ	RW	32	0x0000 0000	0x4A00 8104
CM_CLKMODE_DPLL_PER	RW	32	0x0000 003C	0x4A00 8140
CM_IDLEST_DPLL_PER	R	32	0x0000 0040	0x4A00 8144
CM_AUTOIDLE_DPLL_PER	RW	32	0x0000 0044	0x4A00 8148
CM_CLKSEL_DPLL_PER	RW	32	0x0000 0048	0x4A00 814C
CM_DIV_M2_DPLL_PER	RW	32	0x0000 004C	0x4A00 8150
CM_DIV_M3_DPLL_PER	RW	32	0x0000 0050	0x4A00 8154
CM_DIV_H11_DPLL_PER	RW	32	0x0000 0054	0x4A00 8158
CM_DIV_H12_DPLL_PER	RW	32	0x0000 0058	0x4A00 815C
CM_DIV_H13_DPLL_PER	RW	32	0x0000 005C	0x4A00 8160
CM_DIV_H14_DPLL_PER	RW	32	0x0000 0060	0x4A00 8164
CM_SSC_DELTAMSTEP_DPLL_PER	RW	32	0x0000 0064	0x4A00 8168
CM_SSC_MODFREQDIV_DPLL_PER	RW	32	0x0000 0068	0x4A00 816C
CM_CLKMODE_DPLL_USB	RW	32	0x0000 007C	0x4A00 8180
CM_IDLEST_DPLL_USB	R	32	0x0000 0080	0x4A00 8184
CM_AUTOIDLE_DPLL_USB	RW	32	0x0000 0084	0x4A00 8188
CM_CLKSEL_DPLL_USB	RW	32	0x0000 0088	0x4A00 818C
CM_DIV_M2_DPLL_USB	RW	32	0x0000 008C	0x4A00 8190
CM_SSC_DELTAMSTEP_DPLL_USB	RW	32	0x0000 00A4	0x4A00 81A8
CM_SSC_MODFREQDIV_DPLL_USB	RW	32	0x0000 00A8	0x4A00 81AC
CM_CLKDCOLDO_DPLL_USB	R	32	0x0000 00B0	0x4A00 81B4
CM_CLKMODE_DPLL_PCIE_REF	RW	32	0x0000 00FC	0x4A00 8200
CM_IDLEST_DPLL_PCIE_REF	R	32	0x0000 0100	0x4A00 8204
CM_AUTOIDLE_DPLL_PCIE_REF	RW	32	0x0000 0104	0x4A00 8208
CM_CLKSEL_DPLL_PCIE_REF	RW	32	0x0000 0108	0x4A00 820C
CM_DIV_M2_DPLL_PCIE_REF	RW	32	0x0000 010C	0x4A00 8210
CM_SSC_DELTAMSTEP_DPLL_PCIE_REF	RW	32	0x0000 0110	0x4A00 8214
CM_SSC_MODFREQDIV_DPLL_PCIE_REF	RW	32	0x0000 0114	0x4A00 8218
CM_CLKMODE_APLL_PCIE	RW	32	0x0000 0118	0x4A00 821C
CM_IDLEST_APLL_PCIE	R	32	0x0000 011C	0x4A00 8220
CM_DIV_M2_APLL_PCIE	R	32	0x0000 0120	0x4A00 8224

Table 3-584. CKGEN_CM_CORE Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	CKGEN_CM_CORE Physical Address
CM_CLKVCOLDO_APLL_PCIE	R	32	0x0000 0124	0x4A00 8228

3.12.18.2 CKGEN_CM_CORE Register Description

Table 3-585. CM_CLKSEL_USB_60MHZ

Address Offset	0x0000 0000		
Physical Address	0x4A00 8104	**Instance**	CKGEN_CM_CORE
Description	Selects the configuration of the divider generating 60MHz clock for USB from the DPLL_USB o/p.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															CLKSEL

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	CLKSEL	Select the configuration of the divider 0x0: Set the divider in bypass mode to support bypass clock from DPLL_USB to pass through without division. 0x1: Set the divider to divide the DPLL o/p (480MHz typical) by 8 to generate 60MHz clock.	RW	0x1

Table 3-586. CM_CLKMODE_DPLL_PER

Address Offset	0x0000 003C		
Physical Address	0x4A00 8140	**Instance**	CKGEN_CM_CORE
Description	This register allows controlling the DPLL modes.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																DPLL_SSC_TYPE	DPLL_SSC_DOWNSPREAD	DPLL_SSC_ACK	DPLL_SSC_EN	DPLL_REGM4XEN	DPLL_LPMODE_EN	DPLL_RELOCK_RAMP_EN	DPLL_DRIFTGUARD_EN	DPLL_RAMP_RATE			DPLL_RAMP_LEVEL				DPLL_EN

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0
15	DPLL_SSC_TYPE	Select between Triangular and SquareWave Spread Spectrum Clocking 0x0: Triangular Spread Spectrum Clocking is selected 0x1: Square Wave Spread Spectrum Clocking is selected (only available under proper licensing agreement)	RW	0x0
14	DPLL_SSC_DOWNSPREAD	Control if only low frequency spread is required 0x0: When SSC is enabled, clock frequency is spread on both sides of the programmed frequency 0x1: When SSC is enabled, clock frequency is spread only on the lower side of the programmed frequency	RW	0x0
13	DPLL_SSC_ACK	Acknowledgement from the DPLL regarding start and stop of Spread Spectrum Clocking feature 0x0: SSC has been turned off on PLL o/ps 0x1: SSC has been turned on on PLL o/ps	R	0x0
12	DPLL_SSC_EN	Enable or disable Spread Spectrum Clocking 0x0: SSC disabled 0x1: SSC enabled	RW	0x0
11	DPLL_REGM4XEN	Enable the REGM4XEN mode of the DPLL. Please check the DPLL documentation to check when this mode can be enabled. 0x0: REGM4XEN mode of the DPLL is disabled	R	0x0
10	DPLL_LPMODE_EN	Set the DPLL in Low Power mode. Check the DPLL documentation to see when this can be enabled. 0x0: Low power mode of the DPLL is disabled 0x1: Low power mode of the DPLL is enabled	RW	0x0
9	DPLL_RELOCK_RAMP_EN	If enabled, the clock ramping feature is used applied during the lock process, as well as the relock process. If disabled, the clock ramping feature is used only during the first lock.	RW	0x0
8	DPLL_DRIFTGUARD_EN	This bit allows to enable or disable the automatic recalibration feature of the DPLL. The DPLL will automatically start a recalibration process upon assertion of the DPLL's RECAL flag if this bit is set. 0x0: DRIFTGUARD feature is disabled 0x1: DRIFTGUARD feature is enabled	RW	0x0
7:5	DPLL_RAMP_RATE	Selects the time in terms of DPLL REFCLKs spent at each stage of the clock ramping process 0x0: 2 REFCLKs 0x1: 4 REFCLKs 0x2: 8 REFCLKs 0x3: 16 REFCLKs 0x4: 32 REFCLKs 0x5: 64 REFCLKs 0x6: 128 REFCLKs 0x7: 512 REFCLKs	RW	0x0

Bits	Field Name	Description	Type	Reset
4:3	DPLL_RAMP_LEVEL	The DPLL provides an output clock frequency ramping feature when switching from bypass clock to normal clock during lock and re-lock. The frequency ramping will happen in a maximum of 4 steps in frequency before the DPLL's frequency lock indicator is asserted. This register is used to enable/disable the DPLL ramping feature. If enabled, it is also used to select the algorithm used for clock ramping 0x0: CLKOUT = No ramping CLKOUTX2 = No ramping 0x1: CLKOUT = Bypass clk - Fout/8 - Fout/4 - Fout/2 - Fout CLKOUTX2 = Bypass clk - Foutx2/8 - Foutx2/4 - Foutx2/2 - Foutx2 0x2: CLKOUT = Bypass clk - Fout/4 - Fout/2 - Fout/1.5 - Fout CLKOUTX2 = Bypass clk - Foutx2/4 - Foutx2/2 - Foutx2/1.5 - Foutx2 0x3: Reserved	RW	0x0
2:0	DPLL_EN	DPLL control. 0x0: Reserved 0x1: Reserved 0x2: Reserved 0x3: Reserved 0x4: Reserved 0x5: Put the DPLL in Idle Bypass Low Power mode. 0x6: Put the DPLL in Idle Bypass Fast Relock mode. 0x7: Enables the DPLL in Lock mode	RW	0x5

Table 3-587. Register Call Summary for Register CM_CLKMODE_DPLL_PER

Clock Management Functional Description
- DPLL_PER Power Modes: [0][1]
- DPLL_PER Recalibration: [2]

PRCM Register Manual
- Not Supported Functionality (Registers and Bitfields): [8][9][10][11]
- CKGEN_CM_CORE Register Summary: [12]

Table 3-588. CM_IDLEST_DPLL_PER

Address Offset	0x0000 0040		
Physical Address	0x4A00 8144	**Instance**	CKGEN_CM_CORE
Description	This register allows monitoring DPLL activity. This register is read only and automatically updated. [warm reset insensitive]		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																											ST_DPLL_INIT	ST_DPLL_MODE			ST_DPLL_CLK

Bits	Field Name	Description	Type	Reset
31:5	RESERVED		R	0x0
4	ST_DPLL_INIT	DPLL init status (for debug purpose). 0x0: DPLL is not init 0x1: DPLL has been init	R	0x0
3:1	ST_DPLL_MODE	DPLL mode status (for debug purpose). 0x0: Transient state. From reset to any LP idle state or from any power state to any power state (Power states are Low Power Stop mode, Fast Relock Stop mode, Idle Bypass Low Power mode and Idle Bypass Fast Relock mode). 0x1: The DPLL is in Low Power Stop mode. 0x2: The DPLL is in Fast Relock Stop mode. 0x3: Reserved 0x4: Reserved 0x5: The DPLL is in Idle Bypass Low Power mode. 0x6: The DPLL is in Idle Bypass Fast Relock mode. 0x7: Reserved	R	0x0
0	ST_DPLL_CLK	DPLL lock status 0x0: DPLL is either in bypass mode or in stop mode. 0x1: DPLL is LOCKED	R	0x0

Table 3-589. Register Call Summary for Register CM_IDLEST_DPLL_PER

PRCM Register Manual
- CKGEN_CM_CORE Register Summary: [0]

Table 3-590. CM_AUTOIDLE_DPLL_PER

Address Offset	0x0000 0044		
Physical Address	0x4A00 8148	**Instance**	CKGEN_CM_CORE
Description	This register provides automatic control over the DPLL activity.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														AUTO_DPLL_MODE	

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0
2:0	AUTO_DPLL_MODE	DPLL automatic control; 0x0: DPLL auto control disabled 0x1: The DPLL is automatically put in Low Power Stop mode when its DPLL generated clocks are not required anymore. It is also restarted automatically. 0x2: The DPLL is automatically put in Fast Relock Stop mode when its DPLL generated clocks are not required anymore. It is also restarted automatically. 0x3: Reserved 0x4: Reserved 0x5: The DPLL is automatically put in Idle Bypass Low Power mode when its DPLL generated clocks are not required anymore. It is also restarted automatically. 0x6: The DPLL is automatically put in Idle Bypass Fast Relock mode when its DPLL generated clocks are not required anymore. It is also restarted automatically. 0x7: Reserved	RW	0x0

Table 3-591. Register Call Summary for Register CM_AUTOIDLE_DPLL_PER

Clock Management Functional Description
- DPLL_PER Power Modes: [0]

PRCM Register Manual
- CKGEN_CM_CORE Register Summary: [1]

Table 3-592. CM_CLKSEL_DPLL_PER

Address Offset	0x0000 0048		
Physical Address	0x4A00 814C	**Instance**	CKGEN_CM_CORE
Description	This register provides controls over the DPLL.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DPLL_BYP_CLKSEL	DCC_EN	RESERVED						DPLL_MULT								RESERVED	DPLL_DIV						

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23	DPLL_BYP_CLKSEL	Allows control of the BYPASS clock of the PLL and the associated HSDIVIDER. Same as ULOWCLKEN on DPLL. In DPLL Locked mode, 0 - No impact 1 - No impact In DPLL Bypass mode, 0 - CLKINP is selected as the BYPASS clock for CLKOUT/CLKOUTX2 1 - CLKINPULOW is selected as the BYPASS clock for CLKOUT/CLKOUTX2	RW	0x0
22	DCC_EN	Duty-cycle corrector for high frequency clock 0x0: Duty-cycle corrector is disabled	R	0x0
21:19	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
18:8	DPLL_MULT	DPLL multiplier factor (2 to 2047). (equal to input M of DPLL; M=2 to 2047 = DPLL multiplies by M). [warm reset insensitive] 0x0: Reserved 0x1: Reserved	RW	0x0
7	RESERVED		R	0x0
6:0	DPLL_DIV	DPLL divider factor (0 to 127) (equal to input N of DPLL; actual division factor is N+1). [warm reset insensitive]	RW	0x0

Table 3-593. Register Call Summary for Register CM_CLKSEL_DPLL_PER

Clock Management Functional Description
- DPLL_PER Synthesized Clock Parameters: [0][1]

PRCM Register Manual
- CKGEN_CM_CORE Register Summary: [2]

Table 3-594. CM_DIV_M2_DPLL_PER

Address Offset	0x0000 004C		
Physical Address	0x4A00 8150	**Instance**	CKGEN_CM_CORE
Description	This register provides controls over the M2 divider of DPLL_PER.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																				CLKX2ST	RESERVED	CLKST	RESERVED					DIVHS			

Bits	Field Name	Description	Type	Reset
31:12	RESERVED		R	0x0
11	CLKX2ST	DPLL CLKOUTX2 status 0x0: The clock output is gated 0x1: The clock output is enabled	R	0x0
10	RESERVED		R	0x0
9	CLKST	DPLL CLKOUT status 0x0: The clock output is gated 0x1: The clock output is enabled	R	0x0
8:5	RESERVED		R	0x0
4:0	DIVHS	This field programs the M2 post-divider factor (1 to 31) of DPLL_PER. 0x0: Reserved 0x1: M2 = /1 0x2: M2 = /2 ... 0x1F: M2 = /31	RW	0x1

Table 3-595. Register Call Summary for Register CM_DIV_M2_DPLL_PER

Clock Management Functional Description • DPLL_PER Synthesized Clock Parameters: [0][1][2]
PRCM Register Manual • CKGEN_CM_CORE Register Summary: [3]

Table 3-596. CM_DIV_M3_DPLL_PER

Address Offset	0x0000 0050		
Physical Address	0x4A00 8154	**Instance**	CKGEN_CM_CORE
Description	This register provides controls over the M3 divider of the DPLL_PER.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						CLKST	RESERVED					DIVHS			

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	CLKST	DPLL CLKOUTHIF status 0x0: The clock output is gated 0x1: The clock output is enabled	R	0x0
8:5	RESERVED		R	0x0
4:0	DIVHS	This field programs the M3 post-divider factor (1 to 31) of DPLL_PER. 0x0: Reserved 0x1: M3 = /1 0x2: M3 = /2 ... 0x1F: M3 = /31	RW	0x1

Table 3-597. CM_DIV_H11_DPLL_PER

Address Offset	0x0000 0054		
Physical Address	0x4A00 8158	**Instance**	CKGEN_CM_CORE
Description	This register provides controls over the CLKOUT1 o/p of the HSDIVIDER1 DPLL_PER.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						CLKST	RESERVED					DIVHS			

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	CLKST	HSDIVIDER1 CLKOUT1 status 0x0: The clock output is gated 0x1: The clock output is enabled	R	0x0
8:6	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
5:0	DIVHS	This field programs the H11 post-divider factor (1 to 63) of DPLL_PER. 0x0: Reserved 0x1: H11 = /1 0x2: H11 = /2 ... 0x3F: H11 = /63	RW	0x1

Table 3-598. Register Call Summary for Register CM_DIV_H11_DPLL_PER

Clock Management Functional Description • DPLL_PER Synthesized Clock Parameters: [0][1]
PRCM Register Manual • CKGEN_CM_CORE Register Summary: [2]

Table 3-599. CM_DIV_H12_DPLL_PER

Address Offset	0x0000 0058		
Physical Address	0x4A00 815C	**Instance**	CKGEN_CM_CORE
Description	This register provides controls over the CLKOUT2 o/p of the HSDIVIDER1 DPLL_PER.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						CLKST	RESERVED					DIVHS			

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	CLKST	HSDIVIDER1 CLKOUT2 status 0x0: The clock output is gated 0x1: The clock output is enabled	R	0x0
8:6	RESERVED		R	0x0
5:0	DIVHS	This field programs the H12 post-divider factor (1 to 63) of DPLL_PER. 0x0: Reserved 0x1: H12 = /1 0x2: H12 = /2 ... 0x3F: H12 = /63	RW	0x1

Table 3-600. Register Call Summary for Register CM_DIV_H12_DPLL_PER

Clock Management Functional Description • DPLL_PER Synthesized Clock Parameters: [0][1]
PRCM Register Manual • CKGEN_CM_CORE Register Summary: [2]

Table 3-601. CM_DIV_H13_DPLL_PER

Address Offset	0x0000 005C		
Physical Address	0x4A00 8160	**Instance**	CKGEN_CM_CORE
Description	This register provides controls over the CLKOUT3 o/p of the HSDIVIDER1 DPLL_PER.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						CLKST	RESERVED					DIVHS			

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	CLKST	HSDIVIDER1 CLKOUT3 status 0x0: The clock output is gated 0x1: The clock output is enabled	R	0x0
8:6	RESERVED		R	0x0
5:0	DIVHS	This field programs the H13 post-divider factor (1 to 63) of DPLL_PER. 0x0: Reserved 0x1: H13 = /1 0x2: H13 = /2 ... 0x3F: H13 = /63	RW	0x1

Table 3-602. Register Call Summary for Register CM_DIV_H13_DPLL_PER

Clock Management Functional Description
- DPLL_PER Synthesized Clock Parameters: [0][1]

PRCM Register Manual
- CKGEN_CM_CORE Register Summary: [2]

Table 3-603. CM_DIV_H14_DPLL_PER

Address Offset	0x0000 0060		
Physical Address	0x4A00 8164	**Instance**	CKGEN_CM_CORE
Description	This register provides controls over the CLKOUT4 o/p of the HSDIVIDER1.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						CLKST	RESERVED					DIVHS			

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	CLKST	HSDIVIDER1 CLKOUT4 status 0x0: The clock output is gated 0x1: The clock output is enabled	R	0x0

Bits	Field Name	Description	Type	Reset
8:6	RESERVED		R	0x0
5:0	DIVHS	This field programs the H14 post-divider factor (1 to 63) of DPLL_PER. 0x0: Reserved 0x1: H14 = /1 0x2: H14 = /2 ... 0x3F: H14 = /63	RW	0x1

Table 3-604. CM_SSC_DELTAMSTEP_DPLL_PER

Address Offset	0x0000 0064		
Physical Address	0x4A00 8168	**Instance**	CKGEN_CM_CORE
Description	Control the DeltaMStep parameter for Spread Spectrum Clocking. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DELTAMSTEP																							

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19:0	DELTAMSTEP	DeltaMStep is split into fractional and integer part. For Triangular Spread Spectrum: [19:18] for integer part, [17:0] for fractional part For Square Wave Spread Spectrum [19:14] for integer part, [13:0] for fractional part	RW	0x0

Table 3-605. Register Call Summary for Register CM_SSC_DELTAMSTEP_DPLL_PER

Clock Management Functional Description
PRCM Register Manual • CKGEN_CM_CORE Register Summary: [1]

Table 3-606. CM_SSC_MODFREQDIV_DPLL_PER

Address Offset	0x0000 0068		
Physical Address	0x4A00 816C	**Instance**	CKGEN_CM_CORE
Description	Control the Modulation Frequency (Fm) for Spread Spectrum Clocking. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																					MODFREQDIV_EXPONENT			RESERVED	MODFREQDIV_MANTISSA						

Bits	Field Name	Description	Type	Reset
31:11	RESERVED		R	0x0
10:8	MODFREQDIV_EXPONENT	Set the Exponent component of MODFREQDIV factor	RW	0x0
7	RESERVED		R	0x0
6:0	MODFREQDIV_MANTISSA	Set the Mantissa component of MODFREQDIV factor	RW	0x0

Table 3-607. Register Call Summary for Register CM_SSC_MODFREQDIV_DPLL_PER

Clock Management Functional Description

PRCM Register Manual

- CKGEN_CM_CORE Register Summary: [2]

Table 3-608. CM_CLKMODE_DPLL_USB

Address Offset	0x0000 007C		
Physical Address	0x4A00 8180	**Instance**	CKGEN_CM_CORE
Description	This register allows controlling the DPLL modes.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																DPLL_SSC_TYPE	DPLL_SSC_DOWNSPREAD	DPLL_SSC_ACK	DPLL_SSC_EN	RESERVED											DPLL_EN

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0
15	DPLL_SSC_TYPE	Select between Triangular and SquareWave Spread Spectrum Clocking 0x0: Triangular Spread Spectrum Clocking is selected 0x1: SquareWave Spread Spectrum Clocking is selected (only available under proper licensing agreement)	RW	0x0
14	DPLL_SSC_DOWNSPREAD	Control if only low frequency spread is required 0x0: When SSC is enabled, clock frequency is spread on both sides of the programmed frequency 0x1: When SSC is enabled, clock frequency is spread only on the lower side of the programmed frequency	RW	0x0
13	DPLL_SSC_ACK	Acknowledgement from the DPLL regarding start and stop of Spread Spectrum Clocking feature 0x0: SSC has been turned off on PLL o/ps 0x1: SSC has been turned on on PLL o/ps	R	0x0
12	DPLL_SSC_EN	Enable or disable Spread Spectrum Clocking 0x0: SSC disabled 0x1: SSC enabled	RW	0x0
11:3	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
2:0	DPLL_EN	DPLL control. 0x0: Reserved 0x1: Put the DPLL in Low Power Stop mode 0x2: Reserved2 0x3: Reserved 0x4: Reserved 0x5: Put the DPLL in Idle Bypass Low Power mode. 0x6: Reserved 0x7: Enables the DPLL in Lock mode	RW	0x5

Table 3-609. CM_IDLEST_DPLL_USB

Address Offset	0x0000 0080		
Physical Address	0x4A00 8184	**Instance**	CKGEN_CM_CORE
Description	This register allows monitoring DPLL activity. This register is read only and automatically updated. [warm reset insensitive]		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																											ST_DPLL_INIT	ST_DPLL_MODE			ST_DPLL_CLK

Bits	Field Name	Description	Type	Reset
31:5	RESERVED		R	0x0
4	ST_DPLL_INIT	DPLL init status (for debug purpose). 0x0: DPLL is not init 0x1: DPLL has been init	R	0x0
3:1	ST_DPLL_MODE	DPLL mode status (for debug purpose). 0x0: Transient state. From reset to any LP idle state or from any power state to any power state (Power states are Low Power Stop mode, Fast Relock Stop mode, Idle Bypass Low Power mode and Idle Bypass Fast Relock mode). 0x1: The DPLL is in Low Power Stop mode. 0x2: Reserved 0x3: Reserved 0x4: Reserved 0x5: The DPLL is in Idle Bypass Low Power mode. 0x6: Reserved 0x7: Reserved	R	0x0
0	ST_DPLL_CLK	DPLL lock status 0x0: DPLL is either in bypass mode or in stop mode. 0x1: DPLL is LOCKED	R	0x0

Table 3-610. CM_AUTOIDLE_DPLL_USB

Address Offset	0x0000 0084		
Physical Address	0x4A00 8188	**Instance**	CKGEN_CM_CORE
Description	This register provides automatic control over the DPLL activity.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														AUTO_DPLL_MODE	

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0
2:0	AUTO_DPLL_MODE	DPLL automatic control; 0x0: DPLL auto control disabled 0x1: The DPLL is automatically put in Low Power Stop mode when its DPLL generated clocks are not required anymore. It is also restarted automatically. 0x2: Reserved 0x3: Reserved 0x4: Reserved 0x5: The DPLL is automatically put in Idle Bypass Low Power mode when its DPLL generated clocks are not required anymore. It is also restarted automatically. 0x6: Reserved 0x7: Reserved	RW	0x0

Table 3-611. CM_CLKSEL_DPLL_USB

Address Offset	0x0000 0088		
Physical Address	0x4A00 818C	**Instance**	CKGEN_CM_CORE
Description	This register provides controls over the DPLL.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DPLL_SD_DIV								DPLL_BYP_CLKSEL	DCC_EN	DPLL_SELFREQDCO	RESERVED	DPLL_MULT												DPLL_DIV							

Bits	Field Name	Description	Type	Reset
31:24	DPLL_SD_DIV	Sigma-Delta divider select (2-255). This factor must be set by s/w to ensure optimum jitter performance. DPLL_SD_DIV = CEILING ([DPLL_MULT/(DPLL_DIV+1)] * CLKINP / 250), where CLKINP is the input clock of the DPLL in MHz). Must be set with M and N factors, and must not be changed once DPLL is locked. 0x0: Reserved 0x1: Reserved	RW	0x4
23	DPLL_BYP_CLKSEL	Allows control of the BYPASS clock of the PLL and the associated HSDIVIDER. Same as ULOWCLKEN on DPLL. In DPLL Locked mode, 0 - No impact 1 - No impact In DPLL Bypass mode, 0 - CLKINP is selected as the BYPASS clock for CLKOUT 1 - CLKINPULOW is selected as the BYPASS clock for CLKOUT	RW	0x0
22	DCC_EN	Duty-cycle corrector for high frequency clock 0x0: Duty-cycle corrector is disabled	R	0x0
21	DPLL_SELFREQDCO	select DCO output according to required frequency. 0x0: DCO clock is 1500MHz SELFREQDCO input of DPLL is set to '010' 0x1: DCO clock is 1250MHz SELFREQDCO input of DPLL is set to '100'	RW	0x0
20	RESERVED		R	0x0
19:8	DPLL_MULT	DPLL multiplier factor (2 to 4095). (equal to input M of DPLL; M=2 to 4095 = DPLL multiplies by M). [warm reset insensitive] 0x0: Reserved 0x1: Reserved	RW	0x0
7:0	DPLL_DIV	DPLL divider factor (0 to 255) (equal to input N of DPLL; actual division factor is N+1). [warm reset insensitive]	RW	0x0

Table 3-612. CM_DIV_M2_DPLL_USB

Address Offset	0x0000 008C		
Physical Address	0x4A00 8190	**Instance**	CKGEN_CM_CORE
Description	This register provides controls over the M2 divider of the DPLL.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						CLKST	RESERVED	DIVHS							

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	CLKST	DPLL CLKOUT status 0x0: The clock output is gated 0x1: The clock output is enabled	R	0x0
8:7	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
6:0	DIVHS	This field programs the M2 post-divider factor (1 to 127) of DPLL_USB. 0x0: Reserved 0x1: M2 = /1 0x2: M2 = /2 ... 0x7F: M2 = /127	RW	0x1

Table 3-613. CM_SSC_DELTAMSTEP_DPLL_USB

Address Offset	0x0000 00A4		
Physical Address	0x4A00 81A8	**Instance**	CKGEN_CM_CORE
Description	Control the DeltaMStep parameter for Spread Spectrum Clocking. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																DELTAMSTEP															

Bits	Field Name	Description	Type	Reset
31:21	RESERVED		R	0x0
20:0	DELTAMSTEP	DeltaMStep is split into fractional and integer part. For Triangular Spread Spectrum: [20:18] for integer part, [17:0] for fractional part For Square Wave Spread Spectrum [19:14] for integer part, [13:0] for fractional part	RW	0x0

Table 3-614. CM_SSC_MODFREQDIV_DPLL_USB

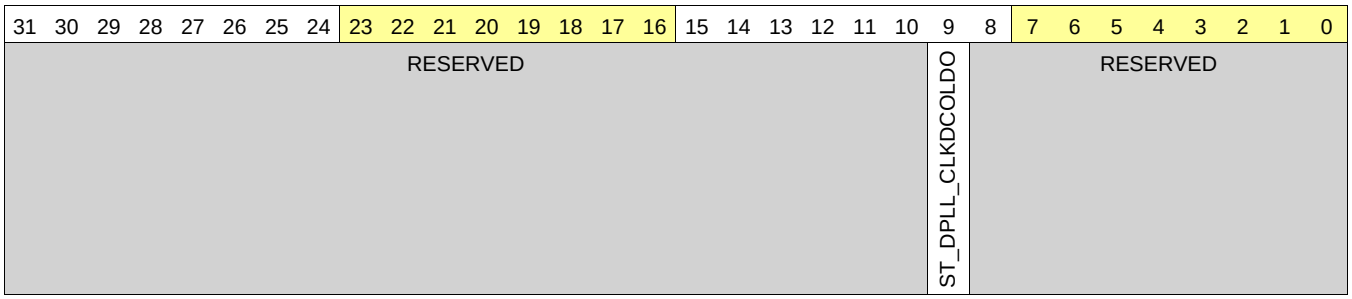
Address Offset	0x0000 00A8		
Physical Address	0x4A00 81AC	**Instance**	CKGEN_CM_CORE
Description	Control the Modulation Frequency (Fm) for Spread Spectrum Clocking. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																					MODFREQDIV_EXPONENT			RESERVED	MODFREQDIV_MANTISSA						

Bits	Field Name	Description	Type	Reset
31:11	RESERVED		R	0x0
10:8	MODFREQDIV_EXPONENT	Set the Exponent component of MODFREQDIV factor	RW	0x0
7	RESERVED		R	0x0
6:0	MODFREQDIV_MANTISSA	Set the Mantissa component of MODFREQDIV factor	RW	0x0

Table 3-615. CM_CLKDCOLDO_DPLL_USB

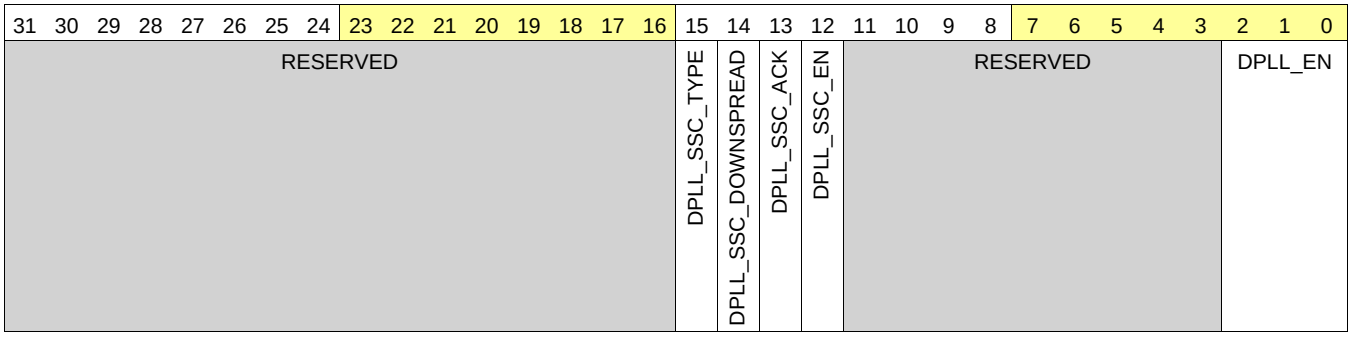
Address Offset	0x0000 00B0		
Physical Address	0x4A00 81B4	**Instance**	CKGEN_CM_CORE
Description	This register provides status over CLKDCOLDO output of the DPLL.		
Type	R		



Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	ST_DPLL_CLKDCOLDO	DPLL CLKDCOLDO status 0x0: The clock output is gated 0x1: The clock output is enabled	R	0x0
8:0	RESERVED		R	0x0

Table 3-616. CM_CLKMODE_DPLL_PCIE_REF

Address Offset	0x0000 00FC		
Physical Address	0x4A00 8200	**Instance**	CKGEN_CM_CORE
Description	This register allows controlling the DPLL modes.		
Type	RW		



Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0
15	DPLL_SSC_TYPE	Select between Triangular and SquareWave Spread Spectrum Clocking 0x0: Triangular Spread Spectrum Clocking is selected 0x1: SquareWave Spread Spectrum Clocking is selected (only available under proper licensing agreement)	RW	0x0

Bits	Field Name	Description	Type	Reset
14	DPLL_SSC_DOWNSPREAD	Control if only low frequency spread is required 0x0: When SSC is enabled, clock frequency is spread on both sides of the programmed frequency 0x1: When SSC is enabled, clock frequency is spread only on the lower side of the programmed frequency	RW	0x0
13	DPLL_SSC_ACK	Acknowledgement from the DPLL regarding start and stop of Spread Spectrum Clocking feature 0x0: SSC has been turned off on PLL o/ps 0x1: SSC has been turned on on PLL o/ps	R	0x0
12	DPLL_SSC_EN	Enable or disable Spread Spectrum Clocking 0x0: SSC disabled 0x1: SSC enabled	RW	0x0
11:3	RESERVED		R	0x0
2:0	DPLL_EN	DPLL control. 0x0: Reserved 0x1: Put the DPLL in Low Power Stop mode 0x2: Reserved2 0x3: Reserved 0x4: Reserved 0x5: Put the DPLL in Idle Bypass Low Power mode. 0x6: Reserved 0x7: Enables the DPLL in Lock mode	RW	0x5

Table 3-617. CM_IDLEST_DPLL_PCIE_REF

Address Offset	0x0000 0100		
Physical Address	0x4A00 8204	**Instance**	CKGEN_CM_CORE
Description	This register allows monitoring DPLL activity. This register is read only and automatically updated. [warm reset insensitive]		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																											ST_DPLL_INIT	ST_DPLL_MODE			ST_DPLL_CLK

Bits	Field Name	Description	Type	Reset
31:5	RESERVED		R	0x0
4	ST_DPLL_INIT	DPLL init status (for debug purpose). 0x0: DPLL is not init 0x1: DPLL has been init	R	0x0

Bits	Field Name	Description	Type	Reset
3:1	ST_DPLL_MODE	DPLL mode status (for debug purpose). 0x0: Transient state. From reset to any LP idle state or from any power state to any power state (Power states are Low Power Stop mode, Fast Relock Stop mode, Idle Bypass Low Power mode and Idle Bypass Fast Relock mode). 0x1: The DPLL is in Low Power Stop mode. 0x2: Reserved 0x3: Reserved 0x4: Reserved 0x5: The DPLL is in Idle Bypass Low Power mode. 0x6: Reserved 0x7: Reserved	R	0x0
0	ST_DPLL_CLK	DPLL lock status 0x0: DPLL is either in bypass mode or in stop mode. 0x1: DPLL is LOCKED	R	0x0

Table 3-618. CM_AUTOIDLE_DPLL_PCIE_REF

Address Offset	0x0000 0104		
Physical Address	0x4A00 8208	**Instance**	CKGEN_CM_CORE
Description	This register provides automatic control over the DPLL activity.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														AUTO_DPLL_MODE	

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0
2:0	AUTO_DPLL_MODE	DPLL automatic control; 0x0: DPLL auto control disabled 0x1: The DPLL is automatically put in Low Power Stop mode when its DPLL generated clocks are not required anymore. It is also restarted automatically. 0x2: Reserved 0x3: Reserved 0x4: Reserved 0x5: The DPLL is automatically put in Idle Bypass Low Power mode when its DPLL generated clocks are not required anymore. It is also restarted automatically. 0x6: Reserved 0x7: Reserved	RW	0x0

Table 3-619. CM_CLKSEL_DPLL_PCIE_REF

Address Offset	0x0000 0108		
Physical Address	0x4A00 820C	**Instance**	CKGEN_CM_CORE
Description	This register provides controls over the DPLL.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DPLL_SD_DIV								DPLL_BYP_CLKSEL	DCC_EN	DPLL_SELFREQDCO	RESERVED	DPLL_MULT												DPLL_DIV							

Bits	Field Name	Description	Type	Reset
31:24	DPLL_SD_DIV	Sigma-Delta divider select (2-255). This factor must be set by s/w to ensure optimum jitter performance. DPLL_SD_DIV = CEILING ([DPLL_MULT/(DPLL_DIV+1)] * CLKINP / 250), where CLKINP is the input clock of the DPLL in MHz). Must be set with M and N factors, and must not be changed once DPLL is locked. 0x0: Reserved 0x1: Reserved	RW	0x4
23	DPLL_BYP_CLKSEL	Allows control of the BYPASS clock of the PLL and the associated HSDIVIDER. Same as ULOWCLKEN on DPLL. In DPLL Locked mode, 0 - No impact 1 - No impact In DPLL Bypass mode, 0 - CLKINP is selected as the BYPASS clock for CLKOUT 1 - CLKINPULOW is selected as the BYPASS clock for CLKOUT	R	0x0
22	DCC_EN	Duty-cycle corrector for high frequency clock 0x0: Duty-cycle corrector is disabled	R	0x0
21	DPLL_SELFREQDCO	select DCO output according to required frequency. 0x0: DCO clock is 1500MHz SELFREQDCO input of DPLL is set to '010' 0x1: DCO clock is 1250MHz SELFREQDCO input of DPLL is set to '100'	RW	0x0
20	RESERVED		R	0x0
19:8	DPLL_MULT	DPLL multiplier factor (2 to 4095). (equal to input M of DPLL; M=2 to 4095 = DPLL multiplies by M). [warm reset insensitive] 0x0: Reserved 0x1: Reserved	RW	0x0
7:0	DPLL_DIV	DPLL divider factor (0 to 255) (equal to input N of DPLL; actual division factor is N+1). [warm reset insensitive]	RW	0x0

Table 3-620. CM_DIV_M2_DPLL_PCIE_REF

Address Offset	0x0000 010C		
Physical Address	0x4A00 8210	**Instance**	CKGEN_CM_CORE
Description	This register provides controls over the M2 divider of the DPLL.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																					CLKLDOST	CLKST	RESERVED	DIVHS							

Bits	Field Name	Description	Type	Reset
31:11	RESERVED		R	0x0
10	CLKLDOST	DPLL CLKOUTLDO status 0x0: Output clock is gated 0x1: Output clock is enabled	R	0x0
9	CLKST	DPLL CLKOUT status 0x0: The clock output is gated 0x1: The clock output is enabled	R	0x0
8:7	RESERVED		R	0x0
6:0	DIVHS	This field programs the M2 post-divider factor (1 to 127) of DPLL_PCIE_REF. 0x0: Reserved 0x1: M2 = /1 0x2: M2 = /2 **...** 0x7F: M2 = /127	RW	0x1

Table 3-621. CM_SSC_DELTAMSTEP_DPLL_PCIE_REF

Address Offset	0x0000 0110		
Physical Address	0x4A00 8214	**Instance**	CKGEN_CM_CORE
Description	Control the DeltaMStep parameter for Spread Spectrum Clocking. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																DELTAMSTEP															

Bits	Field Name	Description	Type	Reset
31:21	RESERVED		R	0x0
20:0	DELTAMSTEP	DeltaMStep is split into fractional and integer part. For Triangular Spread Spectrum: [20:18] for integer part, [17:0] for fractional part For Square Wave Spread Spectrum [19:14] for integer part, [13:0] for fractional part	RW	0x0

Table 3-622. CM_SSC_MODFREQDIV_DPLL_PCIE_REF

Address Offset	0x0000 0114		
Physical Address	0x4A00 8218	**Instance**	CKGEN_CM_CORE
Description	Control the Modulation Frequency (Fm) for Spread Spectrum Clocking. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																					MODFREQDIV_EXPONENT			RESERVED	MODFREQDIV_MANTISSA						

Bits	Field Name	Description	Type	Reset
31:11	RESERVED		R	0x0
10:8	MODFREQDIV_EXPONENT	Set the Exponent component of MODFREQDIV factor	RW	0x0
7	RESERVED		R	0x0
6:0	MODFREQDIV_MANTISSA	Set the Mantissa component of MODFREQDIV factor	RW	0x0

Table 3-623. CM_CLKMODE_APLL_PCIE

Address Offset	0x0000 0118		
Physical Address	0x4A00 821C	**Instance**	CKGEN_CM_CORE
Description	This register allows controlling the APLL modes.		
Type	RW		

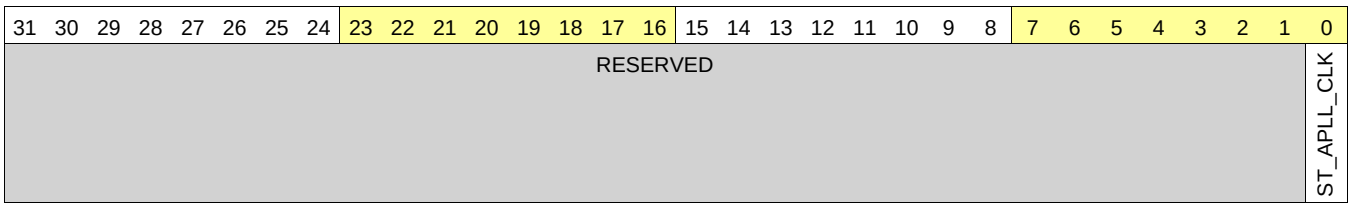
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							CLKDIV_BYPASS	REFSEL	RESERVED	INPSEL			MODE	MODE_SELECT	

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	CLKDIV_BYPASS	0x0: Division of CLKVCOLDO_DIV clock is controlled by OUTSEL pin driven by PCIE controlleur. If OUTSEL is '0', CLKVCOLDO_DIV is at same frequency than CLKVCOLDO output If OUTSEL is '1', CLKVCOLDO_DIV is at CLKVCOLDO divide by 2 frequency 0x1: CLKVCOLDO_DIV clock is not divided by 2 (CLKVCOLDO_DIV is at same frequency than CLKVCOLDO output)	RW	0x0
7	REFSEL	Select source of reference input clock 0x0: APLL reference input clock is from ADPLL 0x1: APLL reference input clock is from ACSPCIE	RW	0x0
6	RESERVED		R	0x0
5:3	INPSEL	Reference clock is 100MHz.	R	0x0
2	MODE	APLLPCIE Mode Status 0x0: APLLPCIE Mode Status	R	0x0

Bits	Field Name	Description	Type	Reset
1:0	MODE_SELECT	Control APLL mode. 0x0: RESERVED 0x1: Put the APLL in Force Lock mode 0x2: Put the APLL in Auto Idle mode 0x3: RESERVED	RW	0x0

Table 3-624. CM_IDLEST_APLL_PCIE

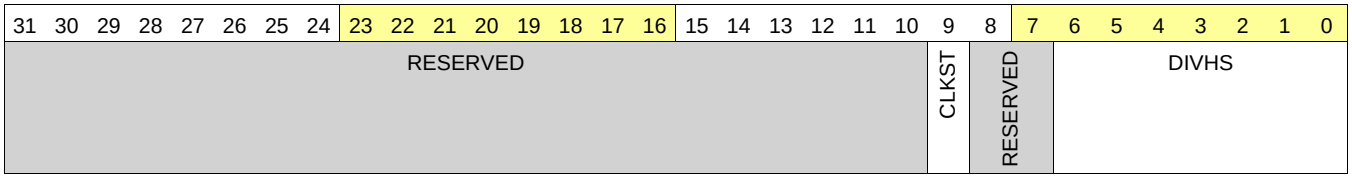
Address Offset	0x0000 011C		
Physical Address	0x4A00 8220	**Instance**	CKGEN_CM_CORE
Description	This register allows monitoring APLL activity. This register is read only and automatically updated. [warm reset insensitive]		
Type	R		



Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	ST_APLL_CLK	APLL lock status 0x0: APLL is either in bypass mode or in stop mode. 0x1: APLL is LOCKED	R	0x0

Table 3-625. CM_DIV_M2_APLL_PCIE

Address Offset	0x0000 0120		
Physical Address	0x4A00 8224	**Instance**	CKGEN_CM_CORE
Description	This register provides controls over the M2 divider of the DPLL.		
Type	R		



Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	CLKST	DPLL CLKOUT status 0x0: The clock output is gated 0x1: The clock output is enabled	R	0x0
8:7	RESERVED		R	0x0
6:0	DIVHS	DPLL M2 post-divider factor (1 to 127). (RESERVED) 0x0: Reserved	R	0x1

Table 3-626. CM_CLKVCOLDO_APLL_PCIE

Address Offset	0x0000 0124		
Physical Address	0x4A00 8228	**Instance**	CKGEN_CM_CORE
Description	This register provides status over CLKVCOLDO and CLKVCOLDO_DIV outputs of the APLL.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																					CLK_DIVST	CLKST	RESERVED								

Bits	Field Name	Description	Type	Reset
31:11	RESERVED		R	0x0
10	CLK_DIVST	APLL CLKVCOLDO_DIV status 0x0: The clock output is gated 0x1: The clock output is enabled	R	0x0
9	CLKST	APLL CLKVCOLDO status 0x0: The clock output is gated 0x1: The clock output is enabled	R	0x0
8:0	RESERVED		R	0x0

3.12.19 COREAON_CM_CORE registers

3.12.19.1 COREAON_CM_CORE Register Summary

Table 3-627. COREAON_CM_CORE Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	COREAON_CM_CORE Physical Address
CM_COREAON_CLKSTCTRL	RW	32	0x0000 0000	0x4A00 8600
CM_COREAON_SMARTREFLEX_MPU_CLKCTRL	RW	32	0x0000 0028	0x4A00 8628
CM_COREAON_SMARTREFLEX_CORE_CLKCTRL	RW	32	0x0000 0038	0x4A00 8638
CM_COREAON_USB_PHY1_CORE_CLKCTRL	RW	32	0x0000 0040	0x4A00 8640
CM_COREAON_IO_SRCOMP_CLKCTRL	RW	32	0x0000 0050	0x4A00 8650
CM_COREAON_SMARTREFLEX_GPU_CLKCTRL	RW	32	0x0000 0058	0x4A00 8658
CM_COREAON_SMARTREFLEX_DSPEVE_CLKCTRL	RW	32	0x0000 0068	0x4A00 8668
CM_COREAON_SMARTREFLEX_IVAHD_CLKCTRL	RW	32	0x0000 0078	0x4A00 8678
CM_COREAON_USB_PHY2_CORE_CLKCTRL	RW	32	0x0000 0088	0x4A00 8688
CM_COREAON_USB_PHY3_CORE_CLKCTRL	RW	32	0x0000 0098	0x4A00 8698
CM_COREAON_DUMMY_MODULE1_CLKCTRL	RW	32	0x0000 00A0	0x4A00 86A0
CM_COREAON_DUMMY_MODULE2_CLKCTRL	RW	32	0x0000 00B0	0x4A00 86B0
CM_COREAON_DUMMY_MODULE3_CLKCTRL	RW	32	0x0000 00C0	0x4A00 86C0
CM_COREAON_DUMMY_MODULE4_CLKCTRL	RW	32	0x0000 00D0	0x4A00 86D0

3.12.19.2 COREAON_CM_CORE Register Description

Table 3-628. CM_COREAON_CLKSTCTRL

Address Offset	0x0000 0000		
Physical Address	0x4A00 8600	**Instance**	COREAON_CM_CORE
Description	This register enables the domain power state transition. It controls the HW supervised domain power state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED															CLKACTIVITY_ABE_GICLK	CLKACTIVITY_SR_IVAHD_SYS_GFCLK	CLKACTIVITY_COREAON_IO_SRCOMP_GFCLK	CLKACTIVITY_SR_DSPEVE_SYS_GFCLK	CLKACTIVITY_COREAON_32K_GFCLK	CLKACTIVITY_SR_CORE_SYS_GFCLK	CLKACTIVITY_SR_GPU_SYS_GFCLK	CLKACTIVITY_SR_MPU_SYS_GFCLK	CLKACTIVITY_COREAON_L4_GICLK	RESERVED						CLKTRCTRL	

Bits	Field Name	Description	Type	Reset
31:17	RESERVED		R	0x0
16	CLKACTIVITY_ABE_GICLK	This field indicates the state of the ABE_GICLK clock input of the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
15	CLKACTIVITY_SR_IVAHD_SYS_GFCLK	This field indicates the state of the SR_IVAHD_SYS_GFCLK clock input of the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
14	CLKACTIVITY_COREAON_IO_SRCOMP_GFCLK	This field indicates the state of the COREAON_IO_SRCOMP_GFCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
13	CLKACTIVITY_SR_DSPEVE_SYS_GFCLK	This field indicates the state of the SR_DSPEVE_SYS_GFCLK clock input of the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0

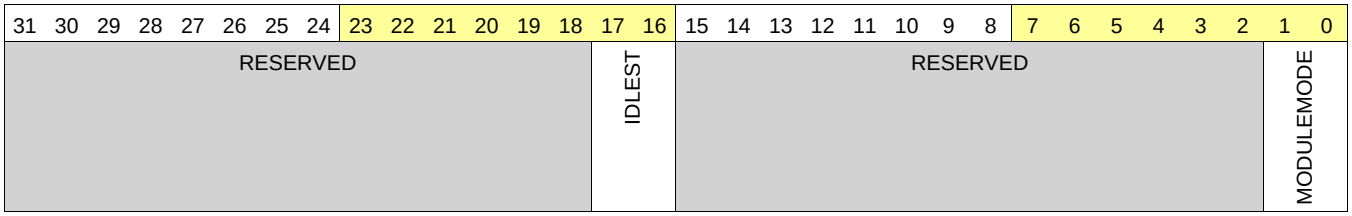
Bits	Field Name	Description	Type	Reset
12	CLKACTIVITY_COREAON_32K_GFCLK	This field indicates the state of the COREAON_32K_GFCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
11	CLKACTIVITY_SR_CORE_SYS_GFCLK	This field indicates the state of the SR_CORE_SYS_GFCLK clock input of the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
10	CLKACTIVITY_SR_GPU_SYS_GFCLK	This field indicates the state of the SR_GPU_SYS_GFCLK clock input of the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
9	CLKACTIVITY_SR_MPU_SYS_GFCLK	This field indicates the state of the SR_MPU_SYS_GFCLK clock input of the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
8	CLKACTIVITY_COREAON_L4_GICLK	This field indicates the state of the COREAON_L4_GICLK clock of the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
7:2	RESERVED		R	0x0
1:0	CLKTRCTRL	Controls the clock state transition of the COREAON clock domain. 0x0: NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may however occur. 0x1: Reserved 0x2: SW_WKUP: Start a software forced wake-up transition on the domain. 0x3: HW_AUTO: Automatic transition is enabled. Sleep and wakeup transition are based upon hardware conditions.	RW	0x3

Table 3-629. Register Call Summary for Register CM_COREAON_CLKSTCTRL

Clock Management Functional Description • Clock Domain Modes: [0][1][2][3][4][6]
PRCM Register Manual • Not Supported Functionality (Registers and Bitfields): [7] • COREAON_CM_CORE Register Summary: [8]

Table 3-630. CM_COREAON_SMARTREFLEX_MPU_CLKCTRL

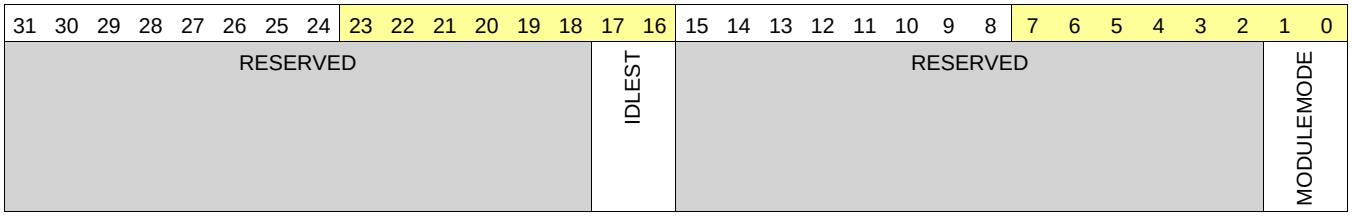
Address Offset	0x0000 0028		
Physical Address	0x4A00 8628	**Instance**	COREAON_CM_CORE
Description	This register manages the SR_MPU clocks.		
Type	RW		



Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guaranteed to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-631. CM_COREAON_SMARTREFLEX_CORE_CLKCTRL

Address Offset	0x0000 0038		
Physical Address	0x4A00 8638	**Instance**	COREAON_CM_CORE
Description	This register manages the SR_CORE clocks.		
Type	RW		



Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guaranteed to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-632. CM_COREAON_USB_PHY1_CORE_CLKCTRL

Address Offset	0x0000 0040		
Physical Address	0x4A00 8640	**Instance**	COREAON_CM_CORE
Description	This register manages the USB PHY 32KHz clock.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							OPTFCLKEN_CLK32K	RESERVED							

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	OPTFCLKEN_CLK32K	Optional functional clock control. 0x0: Optional functional clock is disabled 0x1: Optional functional clock is enabled	RW	0x0
7:0	RESERVED		R	0x0

Table 3-633. CM_COREAON_IO_SRCOMP_CLKCTRL

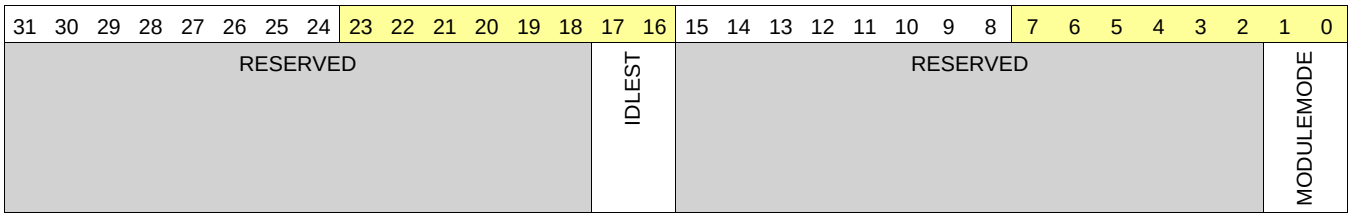
Address Offset	0x0000 0050		
Physical Address	0x4A00 8650	**Instance**	COREAON_CM_CORE
Description	This register manages the clock delivered to the IO Slew rate compensation cells. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							CLKEN_SRCOMP_FCLK	RESERVED							

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	CLKEN_SRCOMP_FCLK	Functional clock control. 0x0: Functional clock is disabled 0x1: Functional clock is enabled.	RW	0x0
7:0	RESERVED		R	0x0

Table 3-634. CM_COREAON_SMARTREFLEX_GPU_CLKCTRL

Address Offset	0x0000 0058		
Physical Address	0x4A00 8658	**Instance**	COREAON_CM_CORE
Description	This register manages the SR_GPU clocks.		
Type	RW		



Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guaranteed to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-635. CM_COREAON_SMARTREFLEX_DSPEVE_CLKCTRL

Address Offset	0x0000 0068		
Physical Address	0x4A00 8668	**Instance**	COREAON_CM_CORE
Description	This register manages the SR_DSPEVE clocks.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														IDLEST		RESERVED								MODULEMODE							

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guaranteed to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-636. CM_COREAON_SMARTREFLEX_IVAHD_CLKCTRL

Address Offset	0x0000 0078		
Physical Address	0x4A00 8678	**Instance**	COREAON_CM_CORE
Description	This register manages the SR_IVAHD clocks.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														IDLEST		RESERVED								MODULEMODE							

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guaranteed to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-637. CM_COREAON_USB_PHY2_CORE_CLKCTRL

Address Offset	0x0000 0088		
Physical Address	0x4A00 8688	**Instance**	COREAON_CM_CORE
Description	This register manages the USB PHY 32KHz clock.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							OPTFCLKEN_CLK32K	RESERVED							

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	OPTFCLKEN_CLK32K	Optional functional clock control. 0x0: Optional functional clock is disabled 0x1: Optional functional clock is enabled	RW	0x0
7:0	RESERVED		R	0x0

Table 3-638. CM_COREAON_USB_PHY3_CORE_CLKCTRL

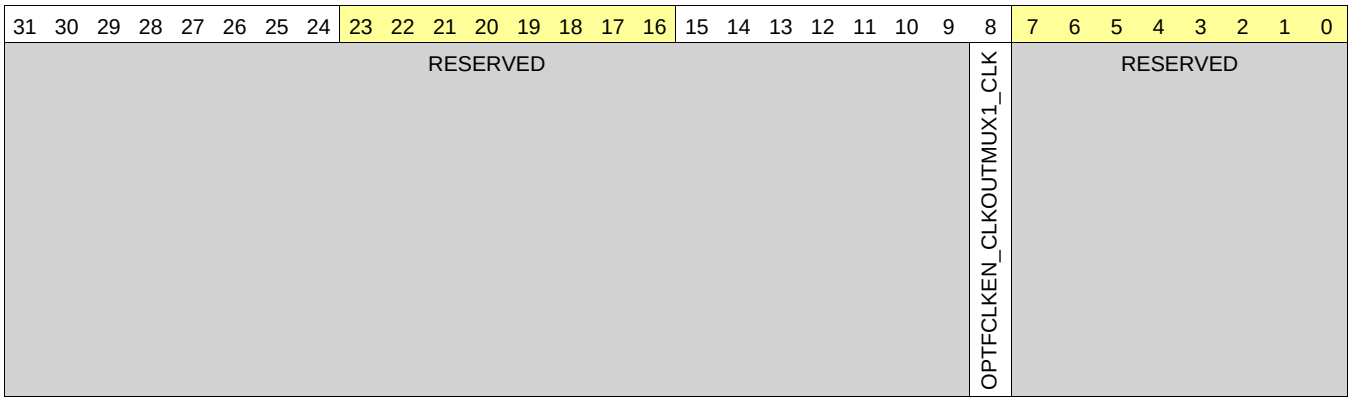
Address Offset	0x0000 0098		
Physical Address	0x4A00 8698	**Instance**	COREAON_CM_CORE
Description	This register manages the USB PHY 32KHz clock.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							OPTFCLKEN_CLK32K	RESERVED							

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	OPTFCLKEN_CLK32K	Optional functional clock control. 0x0: Optional functional clock is disabled 0x1: Optional functional clock is enabled	RW	0x0
7:0	RESERVED		R	0x0

Table 3-639. CM_COREAON_DUMMY_MODULE1_CLKCTRL

Address Offset	0x0000 00A0		
Physical Address	0x4A00 86A0	**Instance**	COREAON_CM_CORE
Description	Used for controlling the CLKOUTMUX1 gate.		
Type	RW		



Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	OPTFCLKEN_CLKOUTMUX1_CLK	Optional functional clock control. 0x0: Optional functional clock is disabled 0x1: Optional functional clock is enabled	RW	0x0
7:0	RESERVED		R	0x0

Table 3-640. Register Call Summary for Register CM_COREAON_DUMMY_MODULE1_CLKCTRL

Clock Management Functional Description
- CM_CORE_AON_CLKOUTMUX Overview: [0]

PRCM Register Manual
- COREAON_CM_CORE Register Summary: [1]

Table 3-641. CM_COREAON_DUMMY_MODULE2_CLKCTRL

Address Offset	0x0000 00B0		
Physical Address	0x4A00 86B0	**Instance**	COREAON_CM_CORE
Description	Used for controlling CLKOUTMUX2 gate.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							OPTFCLKEN_CLKOUTMUX2_CLK	RESERVED							

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	OPTFCLKEN_CLKOUTMUX2_CLK	Optional functional clock control. 0x0: Optional functional clock is disabled 0x1: Optional functional clock is enabled	RW	0x0
7:0	RESERVED		R	0x0

Table 3-642. Register Call Summary for Register CM_COREAON_DUMMY_MODULE2_CLKCTRL

Clock Management Functional Description • CM_CORE_AON_CLKOUTMUX Overview: [0]
PRCM Register Manual • COREAON_CM_CORE Register Summary: [1]

Table 3-643. CM_COREAON_DUMMY_MODULE3_CLKCTRL

Address Offset	0x0000 00C0		
Physical Address	0x4A00 86C0	**Instance**	COREAON_CM_CORE
Description	Used for controlling the L3INIT_60M_GFCLK gate.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							OPTFCLKEN_L3INIT_60M_GFCLK	RESERVED							

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	OPTFCLKEN_L3INIT_60M_GFCLK	Optional functional clock control. 0x0: Optional functional clock is disabled 0x1: Optional functional clock is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
7:0	RESERVED		R	0x0

Table 3-644. CM_COREAON_DUMMY_MODULE4_CLKCTRL

Address Offset	0x0000 00D0		
Physical Address	0x4A00 86D0	**Instance**	COREAON_CM_CORE
Description	Used for controlling ABE_GICLK gate.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							OPTFCLKEN_ABE_GICLK	RESERVED							

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	OPTFCLKEN_ABE_GICLK	Optional functional clock control. 0x0: Optional functional clock is disabled 0x1: Optional functional clock is enabled	RW	0x0
7:0	RESERVED		R	0x0

Table 3-645. Register Call Summary for Register CM_COREAON_DUMMY_MODULE4_CLKCTRL

PRCM Register Manual
- COREAON_CM_CORE Register Summary: [0]

3.12.20 CORE_CM_CORE registers

3.12.20.1 CORE_CM_CORE Register Summary

Table 3-646. CORE_CM_CORE Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	CORE_CM_CORE Physical Address
CM_L3MAIN1_CLKSTCTRL	RW	32	0x0000 0000	0x4A00 8700
CM_L3MAIN1_DYNAMICDEP	RW	32	0x0000 0008	0x4A00 8708
CM_L3MAIN1_L3_MAIN_1_CLKCTRL	R	32	0x0000 0020	0x4A00 8720
CM_L3MAIN1_GPMC_CLKCTRL	RW	32	0x0000 0028	0x4A00 8728
CM_L3MAIN1_MMU_EDMA_CLKCTRL	R	32	0x0000 0030	0x4A00 8730
CM_L3MAIN1_MMU_PCIESS_CLKCTRL	R	32	0x0000 0048	0x4A00 8748
CM_L3MAIN1_OCMC_RAM1_CLKCTRL	R	32	0x0000 0050	0x4A00 8750
CM_L3MAIN1_TESOC_CLKCTRL	R	32	0x0000 0058	0x4A00 8758
CM_L3MAIN1_OCMC_RAM3_CLKCTRL	R	32	0x0000 0060	0x4A00 8760
CM_L3MAIN1_OCMC_ROM_CLKCTRL	R	32	0x0000 0068	0x4A00 8768
CM_L3MAIN1_TPCC_CLKCTRL	R	32	0x0000 0070	0x4A00 8770
CM_L3MAIN1_TPTC1_CLKCTRL	RW	32	0x0000 0078	0x4A00 8778

Table 3-646. CORE_CM_CORE Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	CORE_CM_CORE Physical Address
CM_L3MAIN1_TPTC2_CLKCTRL	RW	32	0x0000 0080	0x4A00 8780
CM_L3MAIN1_VCP1_CLKCTRL	R	32	0x0000 0088	0x4A00 8788
CM_L3MAIN1_VCP2_CLKCTRL	R	32	0x0000 0090	0x4A00 8790
CM_L3MAIN1_SPARE_CME_CLKCTRL	R	32	0x0000 0098	0x4A00 8798
CM_L3MAIN1_SPARE_HDMI_CLKCTRL	R	32	0x0000 00A0	0x4A00 87A0
CM_L3MAIN1_SPARE_ICM_CLKCTRL	R	32	0x0000 00A8	0x4A00 87A8
CM_L3MAIN1_SPARE_IVA2_CLKCTRL	R	32	0x0000 00B0	0x4A00 87B0
CM_L3MAIN1_SPARE_SATA2_CLKCTRL	R	32	0x0000 00B8	0x4A00 87B8
CM_L3MAIN1_SPARE_UNKNOWN4_CLKCTRL	R	32	0x0000 00C0	0x4A00 87C0
CM_L3MAIN1_SPARE_UNKNOWN5_CLKCTRL	R	32	0x0000 00C8	0x4A00 87C8
CM_L3MAIN1_SPARE_UNKNOWN6_CLKCTRL	R	32	0x0000 00D0	0x4A00 87D0
CM_L3MAIN1_SPARE_VIDEOPLL1_CLKCTRL	R	32	0x0000 00D8	0x4A00 87D8
CM_L3MAIN1_SPARE_VIDEOPLL2_CLKCTRL	R	32	0x0000 00F0	0x4A00 87F0
CM_L3MAIN1_SPARE_VIDEOPLL3_CLKCTRL	R	32	0x0000 00F8	0x4A00 87F8
CM_IPU2_CLKSTCTRL	RW	32	0x0000 0200	0x4A00 8900
CM_IPU2_STATICDEP	RW	32	0x0000 0204	0x4A00 8904
CM_IPU2_DYNAMICDEP	RW	32	0x0000 0208	0x4A00 8908
CM_IPU2_IPU2_CLKCTRL	RW	32	0x0000 0220	0x4A00 8920
CM_DMA_CLKSTCTRL	RW	32	0x0000 0300	0x4A00 8A00
CM_DMA_STATICDEP	RW	32	0x0000 0304	0x4A00 8A04
CM_DMA_DYNAMICDEP	R	32	0x0000 0308	0x4A00 8A08
CM_DMA_DMA_SYSTEM_CLKCTRL	R	32	0x0000 0320	0x4A00 8A20
CM_EMIF_CLKSTCTRL	RW	32	0x0000 0400	0x4A00 8B00
CM_EMIF_DMM_CLKCTRL	R	32	0x0000 0420	0x4A00 8B20
CM_EMIF_EMIF_OCP_FW_CLKCTRL	R	32	0x0000 0428	0x4A00 8B28
CM_EMIF_EMIF1_CLKCTRL	RW	32	0x0000 0430	0x4A00 8B30
CM_EMIF_EMIF2_CLKCTRL	RW	32	0x0000 0438	0x4A00 8B38
CM_EMIF_EMIF_DLL_CLKCTRL	RW	32	0x0000 0440	0x4A00 8B40
CM_CRC_CRC_CLKCTRL	RW	32	0x0000 0500	0x4A00 8C00
CM_CRC_CLKSTCTRL	RW	32	0x0000 0520	0x4A00 8C20
CM_L4CFG_CLKSTCTRL	RW	32	0x0000 0600	0x4A00 8D00
CM_L4CFG_DYNAMICDEP	RW	32	0x0000 0608	0x4A00 8D08
CM_L4CFG_L4_CFG_CLKCTRL	R	32	0x0000 0620	0x4A00 8D20
CM_L4CFG_SPINLOCK_CLKCTRL	R	32	0x0000 0628	0x4A00 8D28
CM_L4CFG_MAILBOX1_CLKCTRL	R	32	0x0000 0630	0x4A00 8D30
CM_L4CFG_SAR_ROM_CLKCTRL	R	32	0x0000 0638	0x4A00 8D38
CM_L4CFG_OCP2SCP2_CLKCTRL	R	32	0x0000 0640	0x4A00 8D40
CM_L4CFG_MAILBOX2_CLKCTRL	R	32	0x0000 0648	0x4A00 8D48
CM_L4CFG_MAILBOX3_CLKCTRL	R	32	0x0000 0650	0x4A00 8D50
CM_L4CFG_MAILBOX4_CLKCTRL	R	32	0x0000 0658	0x4A00 8D58
CM_L4CFG_MAILBOX5_CLKCTRL	R	32	0x0000 0660	0x4A00 8D60
CM_L4CFG_MAILBOX6_CLKCTRL	R	32	0x0000 0668	0x4A00 8D68
CM_L4CFG_MAILBOX7_CLKCTRL	R	32	0x0000 0670	0x4A00 8D70
CM_L4CFG_MAILBOX8_CLKCTRL	R	32	0x0000 0678	0x4A00 8D78
CM_L4CFG_MAILBOX9_CLKCTRL	R	32	0x0000 0680	0x4A00 8D80
CM_L4CFG_MAILBOX10_CLKCTRL	R	32	0x0000 0688	0x4A00 8D88
CM_L4CFG_MAILBOX11_CLKCTRL	R	32	0x0000 0690	0x4A00 8D90

Table 3-646. CORE_CM_CORE Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	CORE_CM_CORE Physical Address
CM_L4CFG_MAILBOX12_CLKCTRL	R	32	0x0000 0698	0x4A00 8D98
CM_L4CFG_MAILBOX13_CLKCTRL	R	32	0x0000 06A0	0x4A00 8DA0
CM_L4CFG_SPARE_SMARTREFLEX_RTC_CLKCTRL	R	32	0x0000 06A8	0x4A00 8DA8
CM_L4CFG_SPARE_SMARTREFLEX_SDRAM_CLKCTRL	R	32	0x0000 06B0	0x4A00 8DB0
CM_L4CFG_SPARE_SMARTREFLEX_WKUP_CLKCTRL	R	32	0x0000 06B8	0x4A00 8DB8
CM_L4CFG_IO_DELAY_BLOCK_CLKCTRL	R	32	0x0000 06C0	0x4A00 8DC0
CM_L3INSTR_CLKSTCTRL	R	32	0x0000 0700	0x4A00 8E00
CM_L3INSTR_L3_MAIN_2_CLKCTRL	RW	32	0x0000 0720	0x4A00 8E20
CM_L3INSTR_L3_INSTR_CLKCTRL	RW	32	0x0000 0728	0x4A00 8E28
CM_L3INSTR_OCP_WP_NOC_CLKCTRL	RW	32	0x0000 0740	0x4A00 8E40
CM_L3INSTR_DLL_AGING_CLKCTRL	R	32	0x0000 0748	0x4A00 8E48
CM_L3INSTR_CTRL_MODULE_BANDGAP_CLKCTRL	RW	32	0x0000 0750	0x4A00 8E50

3.12.20.2 CORE_CM_CORE Register Description

Table 3-647. CM_L3MAIN1_CLKSTCTRL

Address Offset	0x0000 0000		
Physical Address	0x4A00 8700	**Instance**	CORE_CM_CORE
Description	This register enables the domain power state transition. It controls the HW supervised domain power state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						CLKACTIVITY_L3MAIN1_L4_GICLK	CLKACTIVITY_L3MAIN1_L3_GICLK	RESERVED							CLKTRCTRL

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	CLKACTIVITY_L3MAIN1_L4_GICLK	This field indicates the state of the L3MAIN1_L4_GICLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0

Bits	Field Name	Description	Type	Reset
8	CLKACTIVITY_L3MAIN1_L3_GICLK	This field indicates the state of the L3MAIN1_L3_GICLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
7:2	RESERVED		R	0x0
1:0	CLKTRCTRL	Controls the clock state transition of the L3MAIN1 clock domain. 0x0: NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may however occur. 0x1: Reserved 0x2: Reserved 0x3: HW_AUTO: Automatic transition is enabled. Sleep and wakeup transition are based upon hardware conditions.	RW	0x0

Table 3-648. Register Call Summary for Register CM_L3MAIN1_CLKSTCTRL

Clock Management Functional Description
- Clock Domain Modes: [0][1][2]

PRCM Register Manual
- CORE_CM_CORE Register Summary: [3]
- RESTORE_CM_CORE Register Description: [4][5]

Table 3-649. CM_L3MAIN1_DYNAMICDEP

Address Offset	0x0000 0008		
Physical Address	0x4A00 8708	**Instance**	CORE_CM_CORE
Description	This register controls the dynamic domain depedencies from L3MAIN1 domain towards 'target' domains. It is relevant only for domain having OCP master port(s).		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EVE4_DYNDEP	EVE3_DYNDEP	EVE2_DYNDEP	EVE1_DYNDEP	WINDOWSIZE				L4PER3_DYNDEP	L4PER2_DYNDEP	PCIE_DYNDEP	DSP2_DYNDEP	RESERVED	IPU1_DYNDEP	RESERVED		WKUPAON_DYNDEP	L4SEC_DYNDEP	L4PER_DYNDEP	L4CFG_DYNDEP	RESERVED	GPU_DYNDEP	RESERVED	DSS_DYNDEP	RESERVED			EMIF_DYNDEP	IPU_DYNDEP	IVA_DYNDEP	DSP1_DYNDEP	IPU2_DYNDEP

Bits	Field Name	Description	Type	Reset
31	EVE4_DYNDEP	Dynamic dependency towards EVE4 clock domain 0x1: Dependency is enabled	R	0x1
30	EVE3_DYNDEP	Dynamic dependency towards EVE3 clock domain 0x1: Dependency is enabled	R	0x1
29	EVE2_DYNDEP	Dynamic dependency towards EVE2 clock domain 0x1: Dependency is enabled	R	0x1
28	EVE1_DYNDEP	Dynamic dependency towards EVE1 clock domain 0x1: Dependency is enabled	R	0x1
27:24	WINDOWSIZE	Size of sliding window used to monitor OCP interface activity for determination of auto-sleep feature. Time unit defined by CM_DYN_DEP_PRESCAL register.	RW	0x4

Bits	Field Name	Description	Type	Reset
23	L4PER3_DYNDEP	Dynamic dependency towards L4PER3 clock domain 0x1: Dependency is enabled	R	0x1
22	L4PER2_DYNDEP	Dynamic dependency towards L4PER2 clock domain 0x1: Dependency is enabled	R	0x1
21	PCIE_DYNDEP	Dynamic dependency towards PCIE clock domain 0x1: Dependency is enabled	R	0x1
20	DSP2_DYNDEP	Dynamic dependency towards DSP2 clock domain 0x1: Dependency is enabled	R	0x1
19	RESERVED		R	0x0
18	IPU1_DYNDEP	Dynamic dependency towards IPU1 clock domain 0x1: Dependency is enabled	R	0x1
17:16	RESERVED		R	0x0
15	WKUPAON_DYNDEP	Dynamic dependency towards WKUPAON clock domain 0x1: Dependency is enabled	R	0x1
14	L4SEC_DYNDEP	Dynamic dependency towards L4SEC clock domain 0x1: Dependency is enabled	R	0x1
13	L4PER_DYNDEP	Dynamic dependency towards L4PER1 clock domain 0x1: Dependency is enabled	R	0x1
12	L4CFG_DYNDEP	Dynamic dependency towards L4CFG clock domain 0x1: Dependency is enabled	R	0x1
11	RESERVED		R	0x0
10	GPU_DYNDEP	Dynamic dependency towards GPU clock domain 0x1: Dependency is enabled	R	0x1
9	RESERVED		R	0x0
8	DSS_DYNDEP	Dynamic dependency towards DSS clock domain 0x1: Dependency is enabled	R	0x1
7:5	RESERVED		R	0x0
4	EMIF_DYNDEP	Dynamic dependency towards EMIF clock domain 0x1: Dependency is enabled	R	0x1
3	IPU_DYNDEP	Dynamic dependency towards IPU clock domain 0x1: Dependency is enabled	R	0x1
2	IVA_DYNDEP	Dynamic dependency towards IVA clock domain 0x1: Dependency is enabled	R	0x1
1	DSP1_DYNDEP	Dynamic dependency towards DSP1 clock domain 0x1: Dependency is enabled	R	0x1
0	IPU2_DYNDEP	Dynamic dependency towards IPU2 clock domain 0x1: Dependency is enabled	R	0x1

Table 3-650. Register Call Summary for Register CM_L3MAIN1_DYNAMICDEP

Clock Management Functional Description
- Clock Domain Dependency: [0][1][2][3][4][5][6][7][8][9][10][11]

PRCM Register Manual
- Not Supported Functionality (Registers and Bitfields): [12][13][14][15][16][17][18][19]
- CORE_CM_CORE Register Summary: [20]
- RESTORE_CM_CORE Register Description: [21][22]

Table 3-651. CM_L3MAIN1_L3_MAIN_1_CLKCTRL

Address Offset	0x0000 0020		
Physical Address	0x4A00 8720	**Instance**	CORE_CM_CORE
Description	This register manages the L3_MAIN_1 clocks.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														IDLEST		RESERVED														MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state.	R	0x1

Table 3-652. Register Call Summary for Register CM_L3MAIN1_L3_MAIN_1_CLKCTRL

Clock Management Functional Description
- Clock Domain Module Attributes: [0][1]

PRCM Register Manual
- CORE_CM_CORE Register Summary: [2]

Table 3-653. CM_L3MAIN1_GPMC_CLKCTRL

Address Offset	0x0000 0028		
Physical Address	0x4A00 8728	**Instance**	CORE_CM_CORE
Description	This register manages the GPMC clocks.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														IDLEST		RESERVED														MODULEMODE	

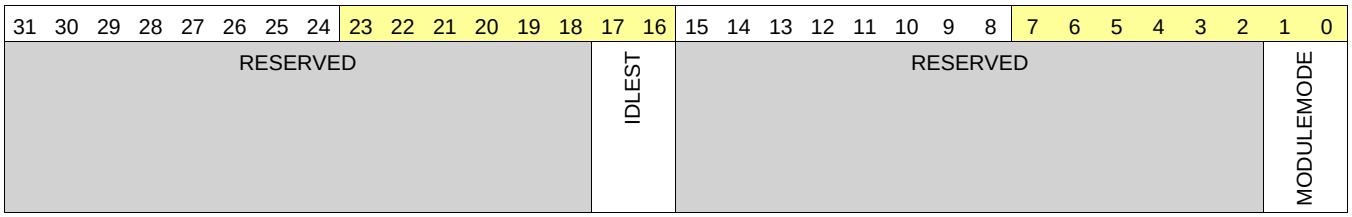
Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is temporarily disabled by SW. OCP access to module are stalled. Can be used to change timing parameter of GPMC module. 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state. 0x2: Reserved 0x3: Reserved	RW	0x1

Table 3-654. Register Call Summary for Register CM_L3MAIN1_GPMC_CLKCTRL

Clock Management Functional Description • Clock Domain Module Attributes: [0][1]
PRCM Register Manual • CORE_CM_CORE Register Summary: [2]

Table 3-655. CM_L3MAIN1_MMU_EDMA_CLKCTRL

Address Offset	0x0000 0030		
Physical Address	0x4A00 8730	**Instance**	CORE_CM_CORE
Description	This register manages the MMU_L4_EDMA clocks.		
Type	R		



Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state.	R	0x1

Table 3-656. Register Call Summary for Register CM_L3MAIN1_MMU_EDMA_CLKCTRL

Clock Management Functional Description • Clock Domain Module Attributes: [0][1]
PRCM Register Manual • CORE_CM_CORE Register Summary: [2]

Table 3-657. CM_L3MAIN1_MMU_PCIESS_CLKCTRL

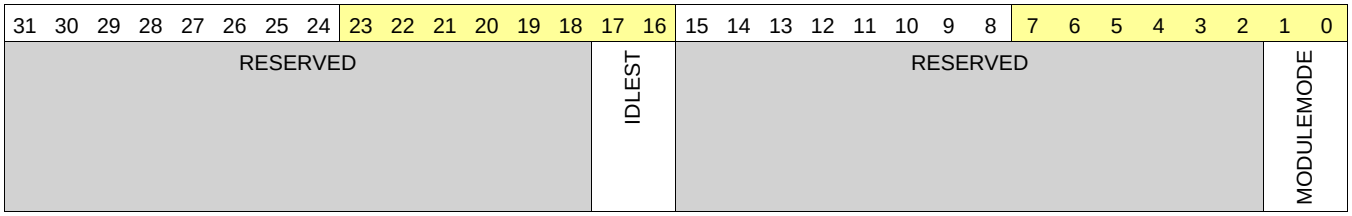
Address Offset	0x0000 0048		
Physical Address	0x4A00 8748	**Instance**	CORE_CM_CORE
Description	This register manages the MMU_L4_PCIESS clocks.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														IDLEST		RESERVED								MODULEMODE							

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state.	R	0x1

Table 3-658. CM_L3MAIN1_OCMC_RAM1_CLKCTRL

Address Offset	0x0000 0050		
Physical Address	0x4A00 8750	**Instance**	CORE_CM_CORE
Description	This register manages the OCMC_RAM1 clocks.		
Type	R		



Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state.	R	0x1

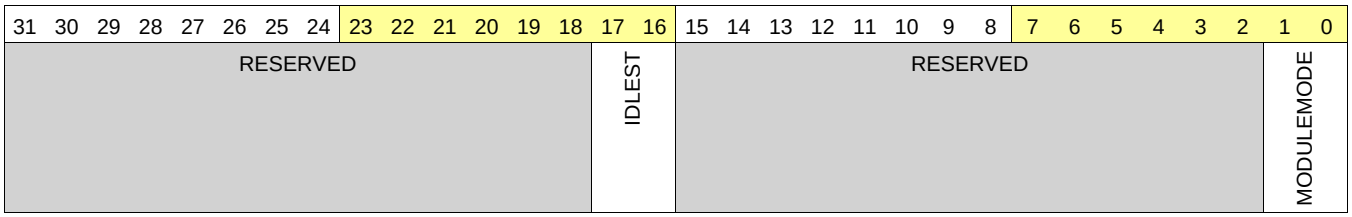
Table 3-659. Register Call Summary for Register CM_L3MAIN1_OCMC_RAM1_CLKCTRL

Clock Management Functional Description
- Clock Domain Module Attributes: [0][1]

PRCM Register Manual
- CORE_CM_CORE Register Summary: [2]

Table 3-660. CM_L3MAIN1_TESOC_CLKCTRL

Address Offset	0x0000 0058		
Physical Address	0x4A00 8758	**Instance**	CORE_CM_CORE
Description	This register manages the TESOC clocks.		
Type	R		



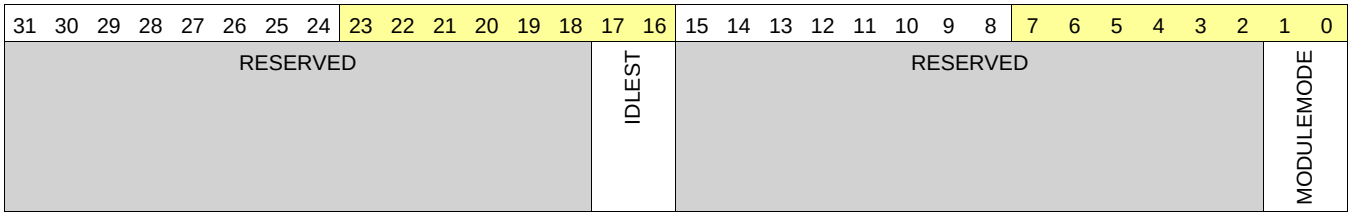
Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state.	R	0x1

Table 3-661. Register Call Summary for Register CM_L3MAIN1_TESOC_CLKCTRL

Clock Management Functional Description • Clock Domain Module Attributes: [0][1]
PRCM Register Manual • CORE_CM_CORE Register Summary: [2]

Table 3-662. CM_L3MAIN1_OCMC_RAM3_CLKCTRL

Address Offset	0x0000 0060		
Physical Address	0x4A00 8760	**Instance**	CORE_CM_CORE
Description	This register manages the OCMC_RAM3 clocks.		
Type	R		



Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state.	R	0x1

Table 3-663. CM_L3MAIN1_OCMC_ROM_CLKCTRL

Address Offset	0x0000 0068		
Physical Address	0x4A00 8768	**Instance**	CORE_CM_CORE
Description	This register manages the OCMC_RAM clocks.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														IDLEST		RESERVED														MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state.	R	0x1

Table 3-664. CM_L3MAIN1_TPCC_CLKCTRL

Address Offset	0x0000 0070		
Physical Address	0x4A00 8770	**Instance**	CORE_CM_CORE
Description	This register manages the TPCC clocks.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														IDLEST		RESERVED														MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state.	R	0x1

Table 3-665. Register Call Summary for Register CM_L3MAIN1_TPCC_CLKCTRL

Clock Management Functional Description • Clock Domain Module Attributes: [0][1]
PRCM Register Manual • CORE_CM_CORE Register Summary: [2]

Table 3-666. CM_L3MAIN1_TPTC1_CLKCTRL

Address Offset	0x0000 0078		
Physical Address	0x4A00 8778	**Instance**	CORE_CM_CORE
Description	This register manages the TPTC1 clocks.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED													STBYST	IDLEST		RESERVED														MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:19	RESERVED		R	0x0
18	STBYST	Module standby status. [warm reset insensitive] 0x0: Module is functional (not in standby) 0x1: Module is in standby	R	0x1
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state. 0x2: Reserved 0x3: Reserved	RW	0x0

Table 3-667. Register Call Summary for Register CM_L3MAIN1_TPTC1_CLKCTRL

Clock Management Functional Description • Clock Domain Module Attributes: [0][1][2]
PRCM Register Manual • CORE_CM_CORE Register Summary: [3]

Table 3-668. CM_L3MAIN1_TPTC2_CLKCTRL

Address Offset	0x0000 0080		
Physical Address	0x4A00 8780	**Instance**	CORE_CM_CORE
Description	This register manages the TPTC2 clocks.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED													STBYST	IDLEST		RESERVED														MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:19	RESERVED		R	0x0
18	STBYST	Module standby status. [warm reset insensitive] 0x0: Module is functional (not in standby) 0x1: Module is in standby	R	0x1
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state. 0x2: Reserved 0x3: Reserved	RW	0x0

Table 3-669. Register Call Summary for Register CM_L3MAIN1_TPTC2_CLKCTRL

Clock Management Functional Description • Clock Domain Module Attributes: [0][1][2]
PRCM Register Manual • CORE_CM_CORE Register Summary: [3]

Table 3-670. CM_L3MAIN1_VCP1_CLKCTRL

Address Offset	0x0000 0088		
Physical Address	0x4A00 8788	**Instance**	CORE_CM_CORE
Description	This register manages the VCP1 clocks.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														IDLEST		RESERVED														MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state.	R	0x1

Table 3-671. CM_L3MAIN1_VCP2_CLKCTRL

Address Offset	0x0000 0090		
Physical Address	0x4A00 8790	**Instance**	CORE_CM_CORE
Description	This register manages the VCP2 clocks.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														IDLEST		RESERVED								MODULEMODE							

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state.	R	0x1

Table 3-672. CM_L3MAIN1_SPARE_CME_CLKCTRL

Address Offset	0x0000 0098		
Physical Address	0x4A00 8798	**Instance**	CORE_CM_CORE
Description	This register manages the SPARE_CME clocks.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														IDLEST		RESERVED								MODULEMODE							

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state.	R	0x1

Table 3-673. CM_L3MAIN1_SPARE_HDMI_CLKCTRL

Address Offset	0x0000 00A0		
Physical Address	0x4A00 87A0	**Instance**	CORE_CM_CORE
Description	This register manages the SPARE_HDMI clocks.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														IDLEST		RESERVED														MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state.	R	0x1

Table 3-674. CM_L3MAIN1_SPARE_ICM_CLKCTRL

Address Offset	0x0000 00A8		
Physical Address	0x4A00 87A8	**Instance**	CORE_CM_CORE
Description	This register manages the SPARE_ICM clocks.		

Table 3-674. CM_L3MAIN1_SPARE_ICM_CLKCTRL (continued)

Type	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														IDLEST		RESERVED								MODULEMODE							

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state.	R	0x1

Table 3-675. CM_L3MAIN1_SPARE_IVA2_CLKCTRL

Address Offset	0x0000 00B0		
Physical Address	0x4A00 87B0	Instance	CORE_CM_CORE
Description	This register manages the SPARE_IVA2 clocks.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														IDLEST		RESERVED								MODULEMODE							

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state.	R	0x1

Table 3-676. CM_L3MAIN1_SPARE_SATA2_CLKCTRL

Address Offset	0x0000 00B8		
Physical Address	0x4A00 87B8	**Instance**	CORE_CM_CORE
Description	This register manages the SPARE_SATA2 clocks.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														IDLEST		RESERVED														MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state.	R	0x1

Table 3-677. CM_L3MAIN1_SPARE_UNKNOWN4_CLKCTRL

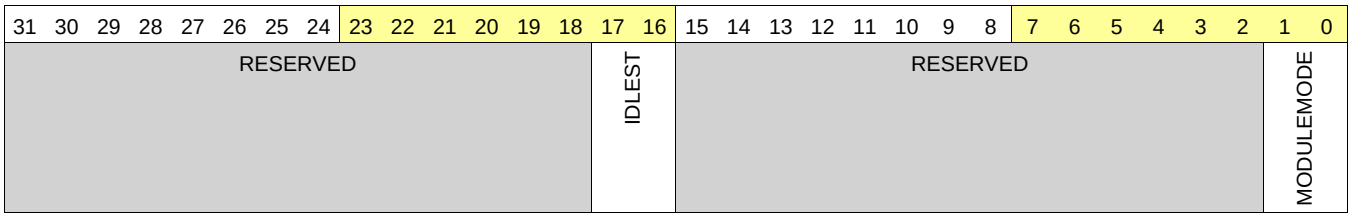
Address Offset	0x0000 00C0		
Physical Address	0x4A00 87C0	**Instance**	CORE_CM_CORE
Description	This register manages the SPARE_UNKNOWN4 clocks.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														IDLEST		RESERVED														MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state.	R	0x1

Table 3-678. CM_L3MAIN1_SPARE_UNKNOWN5_CLKCTRL

Address Offset	0x0000 00C8		
Physical Address	0x4A00 87C8	**Instance**	CORE_CM_CORE
Description	This register manages the SPARE_UNKNOWN5 clocks.		
Type	R		



Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state.	R	0x1

Table 3-679. CM_L3MAIN1_SPARE_UNKNOWN6_CLKCTRL

Address Offset	0x0000 00D0		
Physical Address	0x4A00 87D0	**Instance**	CORE_CM_CORE
Description	This register manages the SPARE_UNKNOWN6 clocks.		

Table 3-679. CM_L3MAIN1_SPARE_UNKNOWN6_CLKCTRL (continued)

Type	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														IDLEST		RESERVED														MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state.	R	0x1

Table 3-680. CM_L3MAIN1_SPARE_VIDEOPLL1_CLKCTRL

Address Offset	0x0000 00D8		
Physical Address	0x4A00 87D8	Instance	CORE_CM_CORE
Description	This register manages the SPARE_VIDEOPLL1 clocks.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														IDLEST		RESERVED														MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state.	R	0x1

Table 3-681. CM_L3MAIN1_SPARE_VIDEOPLL2_CLKCTRL

Address Offset	0x0000 00F0		
Physical Address	0x4A00 87F0	**Instance**	CORE_CM_CORE
Description	This register manages the SPARE_VIDEOPLL2 clocks.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														IDLEST		RESERVED														MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state.	R	0x1

Table 3-682. CM_L3MAIN1_SPARE_VIDEOPLL3_CLKCTRL

Address Offset	0x0000 00F8		
Physical Address	0x4A00 87F8	**Instance**	CORE_CM_CORE
Description	This register manages the SPARE_VIDEOPLL3 clocks.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														IDLEST		RESERVED														MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state.	R	0x1

Table 3-683. CM_IPU2_CLKSTCTRL

Address Offset	0x0000 0200		
Physical Address	0x4A00 8900	**Instance**	CORE_CM_CORE
Description	This register enables the domain power state transition. It controls the HW supervised domain power state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							CLKACTIVITY_IPU2_GFCLK	RESERVED						CLKTRCTRL	

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	CLKACTIVITY_IPU2_GFCLK	This field indicates the state of the IPU2_GCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
7:2	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
1:0	CLKTRCTRL	Controls the clock state transition of the IPU2 clock domain. 0x0: NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may however occur. 0x1: SW_SLEEP: Start a software forced sleep transition on the domain. 0x2: SW_WKUP: Start a software forced wake-up transition on the domain. 0x3: HW_AUTO: Automatic transition is enabled. Sleep and wakeup transition are based upon hardware conditions.	RW	0x3

Table 3-684. Register Call Summary for Register CM_IPU2_CLKSTCTRL

PRCM Register Manual
- CORE_CM_CORE Register Summary: [0]

Table 3-685. CM_IPU2_STATICDEP

Address Offset	0x0000 0204		
Physical Address	0x4A00 8904	**Instance**	CORE_CM_CORE
Description	This register controls the static domain depedencies from IPU domain towards 'target' domains. It is relevant only for domain having system initiator(s).		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	CRC_STATDEP	PCIE_STATDEP	ISS_STATDEP	L4PER3_STATDEP	L4PER2_STATDEP	GMAC_STATDEP	IPU_STATDEP	IPU1_STATDEP	EVE4_STATDEP	EVE3_STATDEP	EVE2_STATDEP	EVE1_STATDEP	DSP2_STATDEP	CUSTEFUSE_STATDEP	COREAON_STATDEP	WKUPAON_STATDEP	L4SEC_STATDEP	L4PER_STATDEP	L4CFG_STATDEP	SDMA_STATDEP	GPU_STATDEP	CAM_STATDEP	DSS_STATDEP	L3INIT_STATDEP	RESERVED	L3MAIN1_STATDEP	EMIF_STATDEP	RESERVED	IVA_STATDEP	DSP1_STATDEP	RESERVED

Bits	Field Name	Description	Type	Reset
31	RESERVED		R	0x0
30	CRC_STATDEP	Static dependency towards CRC clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
29	PCIE_STATDEP	Static dependency towards PCIE clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
28	ISS_STATDEP	Static dependency towards ISS clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
27	L4PER3_STATDEP	Static dependency towards L4PER3 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
26	L4PER2_STATDEP	Static dependency towards L4PER2 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
25	GMAC_STATDEP	Static dependency towards GMAC clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
24	IPU_STATDEP	Static dependency towards IPU clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
23	IPU1_STATDEP	Static dependency towards IPU1 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
22	EVE4_STATDEP	Static dependency towards EVE4 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
21	EVE3_STATDEP	Static dependency towards EVE3 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
20	EVE2_STATDEP	Static dependency towards EVE2 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
19	EVE1_STATDEP	Static dependency towards EVE1 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
18	DSP2_STATDEP	Static dependency towards DSP2 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
17	CUSTEFUSE_STATDEP	Static dependency towards CUSTEFUSE clock domain 0x0: Dependency is disabled	R	0x0
16	COREAON_STATDEP	Static dependency towards COREAON clock domain 0x0: Dependency is disabled	R	0x0
15	WKUPAON_STATDEP	Static dependency towards WKUPAON clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
14	L4SEC_STATDEP	Static dependency towards L4SEC clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
13	L4PER_STATDEP	Static dependency towards L4PER1 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
12	L4CFG_STATDEP	Static dependency towards L4CFG clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
11	SDMA_STATDEP	Static dependency towards DMA clock domain 0x0: Dependency is disabled	R	0x0
10	GPU_STATDEP	Static dependency towards GPU clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
9	CAM_STATDEP	Static dependency towards CAM clock domain 0x0: Dependency is disabled	R	0x0

Bits	Field Name	Description	Type	Reset
8	DSS_STATDEP	Static dependency towards DSS clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
7	L3INIT_STATDEP	Static dependency towards L3INIT clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	RESERVED		R	0x0
5	L3MAIN1_STATDEP	Static dependency towards L3MAIN1 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
4	EMIF_STATDEP	Static dependency towards EMIF clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
3	RESERVED		R	0x0
2	IVA_STATDEP	Static dependency towards IVA clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	DSP1_STATDEP	Static dependency towards DSP clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	RESERVED		R	0x0

Table 3-686. Register Call Summary for Register CM_IPU2_STATICDEP

PRCM Register Manual
- CORE_CM_CORE Register Summary: [0]

Table 3-687. CM_IPU2_DYNAMICDEP

Address Offset	0x0000 0208		
Physical Address	0x4A00 8908	**Instance**	CORE_CM_CORE
Description	This register controls the dynamic domain depedencies from IPU domain towards 'target' domains. It is relevant only for domain having OCP master port(s).		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				WINDOWSIZE				RESERVED														CAM_DYNDEP	RESERVED			L3MAIN1_DYNDEP	RESERVED				

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:24	WINDOWSIZE	Size of sliding window used to monitor OCP interface activity for determination of auto-sleep feature. Time unit defined by CM_DYN_DEP_PRESCAL register.	RW	0x4
23:10	RESERVED		R	0x0
9	CAM_DYNDEP	Dynamic dependency towards CAM clock domain 0x0: Dependency is disabled	R	0x0

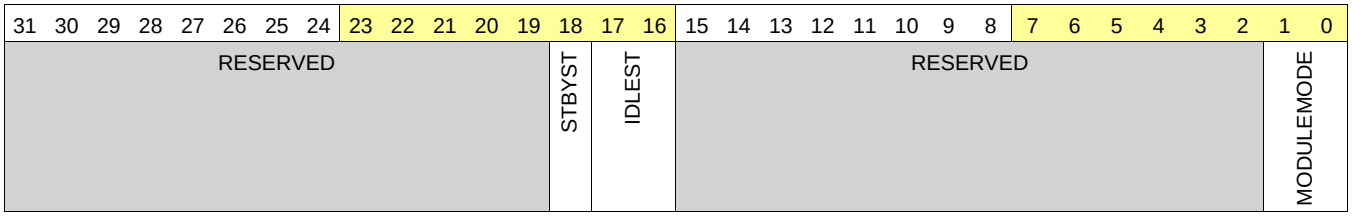
Bits	Field Name	Description	Type	Reset
8:6	RESERVED		R	0x0
5	L3MAIN1_DYNDEP	Dynamic dependency towards L3MAIN1 clock domain 0x1: Dependency is enabled	R	0x1
4:0	RESERVED		R	0x0

Table 3-688. Register Call Summary for Register CM_IPU2_DYNAMICDEP

PRCM Register Manual
- CORE_CM_CORE Register Summary: [0]

Table 3-689. CM_IPU2_IPU2_CLKCTRL

Address Offset	0x0000 0220		
Physical Address	0x4A00 8920	**Instance**	CORE_CM_CORE
Description	This register manages the IPU2 clocks.		
Type	RW		



Bits	Field Name	Description	Type	Reset
31:19	RESERVED		R	0x0
18	STBYST	Module standby status. [warm reset insensitive] 0x0: Module is functional (not in standby) 0x1: Module is in standby	R	0x1
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state. 0x2: Reserved 0x3: Reserved	RW	0x0

Table 3-690. Register Call Summary for Register CM_IPU2_IPU2_CLKCTRL

PRCM Register Manual

- CORE_CM_CORE Register Summary: [0]

Table 3-691. CM_DMA_CLKSTCTRL

Address Offset	0x0000 0300		
Physical Address	0x4A00 8A00	**Instance**	CORE_CM_CORE
Description	This register enables the domain power state transition. It controls the HW supervised domain power state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							CLKACTIVITY_DMA_L3_GICLK	RESERVED						CLKTRCTRL	

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	CLKACTIVITY_DMA_L3_GICLK	This field indicates the state of the DMA_L3_GICLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
7:2	RESERVED		R	0x0
1:0	CLKTRCTRL	Controls the clock state transition of the DMA clock domain. 0x0: NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may however occur. 0x1: Reserved 0x2: SW_WKUP: Start a software forced wake-up transition on the domain. 0x3: HW_AUTO: Automatic transition is enabled. Sleep and wakeup transition are based upon hardware conditions.	RW	0x3

Table 3-692. CM_DMA_STATICDEP

Address Offset	0x0000 0304		
Physical Address	0x4A00 8A04	**Instance**	CORE_CM_CORE
Description	This register controls the static domain depedencies from DMA domain towards 'target' domains. It is relevant only for domain having system initiator(s).		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED		PCIE_STATDEP	RESERVED	L4PER3_STATDEP	L4PER2_STATDEP	RESERVED	IPU_STATDEP	IPU1_STATDEP	RESERVED							WKUPAON_STATDEP	L4SEC_STATDEP	L4PER_STATDEP	L4CFG_STATDEP	RESERVED		CAM_STATDEP	DSS_STATDEP	L3INIT_STATDEP	RESERVED	L3MAIN1_STATDEP	EMIF_STATDEP	RESERVED	IVA_STATDEP	RESERVED	IPU2_STATDEP

Bits	Field Name	Description	Type	Reset
31:30	RESERVED		R	0x0
29	PCIE_STATDEP	Static dependency towards PCIE clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
28	RESERVED		R	0x0
27	L4PER3_STATDEP	Static dependency towards L4PER3 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
26	L4PER2_STATDEP	Static dependency towards L4PER2 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
25	RESERVED		R	0x0
24	IPU_STATDEP	Static dependency towards IPU clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
23	IPU1_STATDEP	Static dependency towards IPU1 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
22:16	RESERVED		R	0x0
15	WKUPAON_STATDEP	Static dependency towards WKUPAON clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
14	L4SEC_STATDEP	Static dependency towards L4SEC clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
13	L4PER_STATDEP	Static dependency towards L4PER1 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
12	L4CFG_STATDEP	Static dependency towards L4CFG clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
11:10	RESERVED		R	0x0
9	CAM_STATDEP	Static dependency towards CAM clock domain 0x0: Dependency is disabled	R	0x0
8	DSS_STATDEP	Static dependency towards DSS clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
7	L3INIT_STATDEP	Static dependency towards L3INIT clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1

Bits	Field Name	Description	Type	Reset
6	RESERVED		R	0x0
5	L3MAIN1_STATDEP	Static dependency towards L3MAIN1 clock domain 0x1: Dependency is enabled	R	0x1
4	EMIF_STATDEP	Static dependency towards EMIF clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
3	RESERVED		R	0x0
2	IVA_STATDEP	Static dependency towards IVA clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	RESERVED		R	0x0
0	IPU2_STATDEP	Static dependency towards IPU2 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-693. CM_DMA_DYNAMICDEP

Address Offset	0x0000 0308		
Physical Address	0x4A00 8A08	**Instance**	CORE_CM_CORE
Description	This register controls the dynamic domain depedencies from SDMA domain towards 'target' domains. It is relevant only for domain having OCP master port(s).		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																										L3MAIN1_DYNDEP	RESERVED				

Bits	Field Name	Description	Type	Reset
31:6	RESERVED		R	0x0
5	L3MAIN1_DYNDEP	Dynamic dependency towards L3MAIN1 clock domain 0x0: Dependency is disabled	R	0x0
4:0	RESERVED		R	0x0

Table 3-694. CM_DMA_DMA_SYSTEM_CLKCTRL

Address Offset	0x0000 0320		
Physical Address	0x4A00 8A20	**Instance**	CORE_CM_CORE
Description	This register manages the DMA_SYSTEM clocks.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED													STBYST	IDLEST		RESERVED														MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:19	RESERVED		R	0x0
18	STBYST	Module standby status. [warm reset insensitive] 0x0: Module is functional (not in standby) 0x1: Module is in standby	R	0x1
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state.	R	0x1

Table 3-695. CM_EMIF_CLKSTCTRL

Address Offset	0x0000 0400		
Physical Address	0x4A00 8B00	**Instance**	CORE_CM_CORE
Description	This register enables the domain power state transition. It controls the HW supervised domain power state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																					CLKACTIVITY_EMIF_PHY_GCLK	CLKACTIVITY_EMIF_DLL_GCLK	CLKACTIVITY_EMIF_L3_GICLK	RESERVED						CLKTRCTRL	

Bits	Field Name	Description	Type	Reset
31:11	RESERVED		R	0x0
10	CLKACTIVITY_EMIF_PHY_GCLK	This field indicates the state of the EMIF_PHY_GCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
9	CLKACTIVITY_EMIF_DLL_GCLK	This field indicates the state of the DLL_GCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0

Bits	Field Name	Description	Type	Reset
8	CLKACTIVITY_EMIF_L3_GICLK	This field indicates the state of the EMIF_L3_GICLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
7:2	RESERVED		R	0x0
1:0	CLKTRCTRL	Controls the clock state transition of the EMIF clock domain. 0x0: NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may however occur. 0x1: Reserved 0x2: SW_WKUP: Start a software forced wake-up transition on the domain. 0x3: HW_AUTO: Automatic transition is enabled. Sleep and wakeup transition are based upon hardware conditions.	RW	0x0

Table 3-696. Register Call Summary for Register CM_EMIF_CLKSTCTRL

Clock Management Functional Description
- Clock Domain Modes: [0][1][2][3]

PRCM Register Manual
- CORE_CM_CORE Register Summary: [4]

Table 3-697. CM_EMIF_DMM_CLKCTRL

Address Offset	0x0000 0420		
Physical Address	0x4A00 8B20	**Instance**	CORE_CM_CORE
Description	This register manages the DMM clocks.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														IDLEST		RESERVED														MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state.	R	0x1

Table 3-698. CM_EMIF_EMIF_OCP_FW_CLKCTRL

Address Offset	0x0000 0428		
Physical Address	0x4A00 8B28	**Instance**	CORE_CM_CORE
Description	This register manages the EMIF_OCP_FW clocks.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														IDLEST		RESERVED								MODULEMODE							

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state.	R	0x1

Table 3-699. Register Call Summary for Register CM_EMIF_EMIF_OCP_FW_CLKCTRL

Clock Management Functional Description • Clock Domain Module Attributes:
PRCM Register Manual • CORE_CM_CORE Register Summary: [2]

Table 3-700. CM_EMIF_EMIF1_CLKCTRL

Address Offset	0x0000 0430		
Physical Address	0x4A00 8B30	**Instance**	CORE_CM_CORE
Description	This register manages the EMIF1 clocks.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED							CLKSEL_LL	RESERVED						IDLEST		RESERVED														MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24	CLKSEL_LL	Source of EMIF1 External Low Latency interface clock EMIF_LL_GCLK Value is provided by LLI_C2C_SELECT input pin 0x0: EMIF_LL_GCLK is same as C2C clock 0x1: EMIF_LL_GCLK is same as LLI clock	R	0x0
23:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is temporarily disabled by SW. OCP access to module are stalled. Can be used to change timing parameter of EMIF_1 module. 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state. 0x2: Reserved 0x3: Reserved	RW	0x1

Table 3-701. Register Call Summary for Register CM_EMIF_EMIF1_CLKCTRL

Clock Management Functional Description • Clock Domain Module Attributes: [0][1]
PRCM Register Manual • CORE_CM_CORE Register Summary: [2]

Table 3-702. CM_EMIF_EMIF2_CLKCTRL

Address Offset	0x0000 0438		
Physical Address	0x4A00 8B38	**Instance**	CORE_CM_CORE
Description	This register manages the EMIF2 clocks.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														IDLEST		RESERVED														MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is temporarily disabled by SW. OCP access to module are stalled. Can be used to change timing parameter of EMIF_2 module. 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state. 0x2: Reserved 0x3: Reserved	RW	0x1

Table 3-703. CM_EMIF_EMIF_DLL_CLKCTRL

Address Offset	0x0000 0440		
Physical Address	0x4A00 8B40	**Instance**	CORE_CM_CORE
Description	This register manages the DLL clock.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							OPTFCLKEN_DLL_CLK	RESERVED							

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	OPTFCLKEN_DLL_CLK	Optional functional clock control. 0x0: Optional functional clock is disabled. DLL_CLK can be gated when EMIF domain performs sleep transition 0x1: Optional functional clock is enabled. DLL_CLK is garantied to not be gated if already running.	RW	0x0
7:0	RESERVED		R	0x0

Table 3-704. Register Call Summary for Register CM_EMIF_EMIF_DLL_CLKCTRL

Clock Management Functional Description • Clock Domain Modes: [0]
PRCM Register Manual • CORE_CM_CORE Register Summary: [1]

Table 3-705. CM_CRC_CRC_CLKCTRL

Address Offset	0x0000 0500		
Physical Address	0x4A00 8C00	**Instance**	CORE_CM_CORE
Description	This register manages the CRC clocks.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				CLKSEL_SOURCE2		CLKSEL_SOURCE1		RESERVED						IDLEST		RESERVED														MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:26	CLKSEL_SOURCE2	Selects source for CRC clock 0x0: Selects L3_ICLK 0x1: Selects PER_ABE_X1_CLK 0x2: Selects DPLL_CLK from SOURCE1 0x3: RESERVED	RW	0x0
25:24	CLKSEL_SOURCE1	Selects source for CRC clock 0x0: Selects FUNC_32K_CLK 0x1: Selects VIDEO1_CLK 0x2: Selects VIDEO2_CLK 0x3: Selects HDMI_CLK	RW	0x0
23:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guaranteed to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-706. Register Call Summary for Register CM_CRC_CRC_CLKCTRL

Clock Management Functional Description • Clock Domain Module Attributes: [0][3]
PRCM Register Manual • Not Supported Functionality (Registers and Bitfields): [4][5] • CORE_CM_CORE Register Summary: [6]

Table 3-707. CM_CRC_CLKSTCTRL

Address Offset	0x0000 0520		
Physical Address	0x4A00 8C20	**Instance**	CORE_CM_CORE
Description	This register enables the domain power state transition. It controls the HW supervised domain power state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						CLKACTIVITY_CRC_GFCLK	CLKACTIVITY_CRC_L3_GICLK	RESERVED						CLKTRCTRL	

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	CLKACTIVITY_CRC_GFCLK	This field indicates the state of the CRC_GFCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
8	CLKACTIVITY_CRC_L3_GICLK	This field indicates the state of the CRC_L3_GICLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
7:2	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
1:0	CLKTRCTRL	Controls the clock state transition of the C2C clock domain. 0x0: NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may however occur. 0x1: Reserved 0x2: SW_WKUP: Start a software forced wake-up transition on the domain. 0x3: HW_AUTO: Automatic transition is enabled. Sleep and wakeup transition are based upon hardware conditions.	RW	0x3

Table 3-708. Register Call Summary for Register CM_CRC_CLKSTCTRL

Clock Management Functional Description
- Clock Domain Modes: [0][2]

PRCM Register Manual
- Not Supported Functionality (Registers and Bitfields): [3]
- CORE_CM_CORE Register Summary: [4]

Table 3-709. CM_L4CFG_CLKSTCTRL

Address Offset	0x0000 0600		
Physical Address	0x4A00 8D00	**Instance**	CORE_CM_CORE
Description	This register enables the domain power state transition. It controls the HW supervised domain power state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						CLKACTIVITY_L4CFG_L3_GICLK	CLKACTIVITY_L4CFG_L4_GICLK	RESERVED							CLKTRCTRL

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	CLKACTIVITY_L4CFG_L3_GICLK	This field indicates the state of the L4CFG_L3_GICLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
8	CLKACTIVITY_L4CFG_L4_GICLK	This field indicates the state of the L4CFG_L4_GICLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
7:2	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
1:0	CLKTRCTRL	Controls the clock state transition of the L4CFG clock domain. 0x0: NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may however occur. 0x1: Reserved 0x2: Reserved 0x3: HW_AUTO: Automatic transition is enabled. Sleep and wakeup transition are based upon hardware conditions.	RW	0x0

Table 3-710. Register Call Summary for Register CM_L4CFG_CLKSTCTRL

Clock Management Functional Description
- Clock Domain Modes: [0][1][2]

PRCM Register Manual
- CORE_CM_CORE Register Summary: [3]
- RESTORE_CM_CORE Register Description: [4]

Table 3-711. CM_L4CFG_DYNAMICDEP

Address Offset	0x0000 0608		
Physical Address	0x4A00 8D08	**Instance**	CORE_CM_CORE
Description	This register controls the dynamic domain depedencies from L4CFG domain towards 'target' domains. It is relevant only for domain having OCP master port(s).		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				WINDOWSIZE				RESERVED				MPU_DYNDEP	RESERVED	CUSTEFUSE_DYNDEP	COREAON_DYNDEP	RESERVED				SDMA_DYNDEP	RESERVED			L3INIT_DYNDEP	RESERVED	L3MAIN1_DYNDEP	EMIF_DYNDEP	RESERVED			

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:24	WINDOWSIZE	Size of sliding window used to monitor OCP interface activity for determination of auto-sleep feature. Time unit defined by CM_DYN_DEP_PRESCAL register.	RW	0x4
23:20	RESERVED		R	0x0
19	MPU_DYNDEP	Dynamic dependency towards MPU clock domain 0x1: Dependency is enabled	R	0x1
18	RESERVED		R	0x0
17	CUSTEFUSE_DYNDEP	Dynamic dependency towards CUSTEFUSE clock domain 0x1: Dependency is enabled	R	0x1
16	COREAON_DYNDEP	Dynamic dependency towards COREAON clock domain 0x1: Dependency is enabled	R	0x1
15:12	RESERVED		R	0x0
11	SDMA_DYNDEP	Dynamic dependency towards DMA clock domain 0x1: Dependency is enabled	R	0x1

Bits	Field Name	Description	Type	Reset
10:8	RESERVED		R	0x0
7	L3INIT_DYNDEP	Dynamic dependency towards L3INIT clock domain 0x1: Dependency is enabled	R	0x1
6	RESERVED		R	0x0
5	L3MAIN1_DYNDEP	Dynamic dependency towards L3MAIN1 clock domain 0x1: Dependency is enabled	R	0x1
4	EMIF_DYNDEP	Dynamic dependency towards EMIF clock domain 0x1: Dependency is enabled	R	0x1
3:0	RESERVED		R	0x0

Table 3-712. Register Call Summary for Register CM_L4CFG_DYNAMICDEP

Clock Management Functional Description • Clock Domain Dependency: [0][1][2][3][4]
PRCM Register Manual • Not Supported Functionality (Registers and Bitfields): [5][6] • CORE_CM_CORE Register Summary: [7] • RESTORE_CM_CORE Register Description: [8][9]

Table 3-713. CM_L4CFG_L4_CFG_CLKCTRL

Address Offset	0x0000 0620		
Physical Address	0x4A00 8D20	**Instance**	CORE_CM_CORE
Description	This register manages the L4_CFG clocks.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														IDLEST		RESERVED								MODULEMODE							

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state.	R	0x1

Table 3-714. Register Call Summary for Register CM_L4CFG_L4_CFG_CLKCTRL

Clock Management Functional Description • Clock Domain Module Attributes: [0][1]
PRCM Register Manual • CORE_CM_CORE Register Summary: [2]

Table 3-715. CM_L4CFG_SPINLOCK_CLKCTRL

Address Offset	0x0000 0628		
Physical Address	0x4A00 8D28	**Instance**	CORE_CM_CORE
Description	This register manages the SPINLOCK clocks.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														IDLEST		RESERVED														MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state.	R	0x1

Table 3-716. Register Call Summary for Register CM_L4CFG_SPINLOCK_CLKCTRL

Clock Management Functional Description • Clock Domain Module Attributes: [0][1]
PRCM Register Manual • CORE_CM_CORE Register Summary: [2]

Table 3-717. CM_L4CFG_MAILBOX1_CLKCTRL

Address Offset	0x0000 0630		
Physical Address	0x4A00 8D30	**Instance**	CORE_CM_CORE
Description	This register manages the MAILBOX1 clocks.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														IDLEST		RESERVED								MODULEMODE							

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state.	R	0x1

Table 3-718. Register Call Summary for Register CM_L4CFG_MAILBOX1_CLKCTRL

Clock Management Functional Description • Clock Domain Module Attributes: [0][1]
PRCM Register Manual • CORE_CM_CORE Register Summary: [2]

Table 3-719. CM_L4CFG_SAR_ROM_CLKCTRL

Address Offset	0x0000 0638		
Physical Address	0x4A00 8D38	**Instance**	CORE_CM_CORE
Description	This register manages the SAR_ROM clocks.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														IDLEST		RESERVED								MODULEMODE							

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state.	R	0x1

Table 3-720. CM_L4CFG_OCP2SCP2_CLKCTRL

Address Offset	0x0000 0640		
Physical Address	0x4A00 8D40	**Instance**	CORE_CM_CORE
Description	This register manages the OCP2SCP2 clocks and the optional clock of USB PHY.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														IDLEST		RESERVED														MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state.	R	0x1

Table 3-721. CM_L4CFG_MAILBOX2_CLKCTRL

Address Offset	0x0000 0648		
Physical Address	0x4A00 8D48	**Instance**	CORE_CM_CORE
Description	This register manages the MAILBOX2 clocks.		

Table 3-721. CM_L4CFG_MAILBOX2_CLKCTRL (continued)

Type	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														IDLEST		RESERVED														MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state.	R	0x1

Table 3-722. Register Call Summary for Register CM_L4CFG_MAILBOX2_CLKCTRL

Clock Management Functional Description
- Clock Domain Module Attributes: [0][1]

PRCM Register Manual
- CORE_CM_CORE Register Summary: [2]

Table 3-723. CM_L4CFG_MAILBOX3_CLKCTRL

Address Offset	0x0000 0650		
Physical Address	0x4A00 8D50	Instance	CORE_CM_CORE
Description	This register manages the MAILBOX3 clocks.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														IDLEST		RESERVED														MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state.	R	0x1

Table 3-724. CM_L4CFG_MAILBOX4_CLKCTRL

Address Offset	0x0000 0658		
Physical Address	0x4A00 8D58	**Instance**	CORE_CM_CORE
Description	This register manages the MAILBOX4 clocks.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														IDLEST		RESERVED														MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state.	R	0x1

Table 3-725. CM_L4CFG_MAILBOX5_CLKCTRL

Address Offset	0x0000 0660		
Physical Address	0x4A00 8D60	**Instance**	CORE_CM_CORE
Description	This register manages the MAILBOX5 clocks.		

Table 3-725. CM_L4CFG_MAILBOX5_CLKCTRL (continued)

Type	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														IDLEST		RESERVED								MODULEMODE							

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state.	R	0x1

Table 3-726. CM_L4CFG_MAILBOX6_CLKCTRL

Address Offset	0x0000 0668		
Physical Address	0x4A00 8D68	Instance	CORE_CM_CORE
Description	This register manages the MAILBOX6 clocks.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														IDLEST		RESERVED								MODULEMODE							

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state.	R	0x1

Table 3-727. CM_L4CFG_MAILBOX7_CLKCTRL

Address Offset	0x0000 0670		
Physical Address	0x4A00 8D70	**Instance**	CORE_CM_CORE
Description	This register manages the MAILBOX7 clocks.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														IDLEST		RESERVED														MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state.	R	0x1

Table 3-728. CM_L4CFG_MAILBOX8_CLKCTRL

Address Offset	0x0000 0678		
Physical Address	0x4A00 8D78	**Instance**	CORE_CM_CORE
Description	This register manages the MAILBOX8 clocks.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														IDLEST		RESERVED														MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state.	R	0x1

Table 3-729. CM_L4CFG_MAILBOX9_CLKCTRL

Address Offset	0x0000 0680		
Physical Address	0x4A00 8D80	**Instance**	CORE_CM_CORE
Description	This register manages the MAILBOX9 clocks.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														IDLEST		RESERVED														MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state.	R	0x1

Table 3-730. CM_L4CFG_MAILBOX10_CLKCTRL

Address Offset	0x0000 0688		
Physical Address	0x4A00 8D88	**Instance**	CORE_CM_CORE
Description	This register manages the MAILBOX10 clocks.		

Table 3-730. CM_L4CFG_MAILBOX10_CLKCTRL (continued)

Type	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														IDLEST		RESERVED														MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state.	R	0x1

Table 3-731. CM_L4CFG_MAILBOX11_CLKCTRL

Address Offset	0x0000 0690		
Physical Address	0x4A00 8D90	Instance	CORE_CM_CORE
Description	This register manages the MAILBOX11 clocks.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														IDLEST		RESERVED														MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state.	R	0x1

Table 3-732. CM_L4CFG_MAILBOX12_CLKCTRL

Address Offset	0x0000 0698		
Physical Address	0x4A00 8D98	**Instance**	CORE_CM_CORE
Description	This register manages the MAILBOX12 clocks.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														IDLEST		RESERVED														MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state.	R	0x1

Table 3-733. CM_L4CFG_MAILBOX13_CLKCTRL

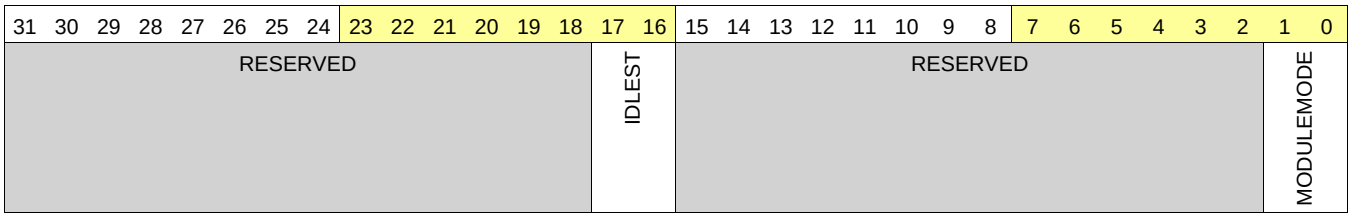
Address Offset	0x0000 06A0		
Physical Address	0x4A00 8DA0	**Instance**	CORE_CM_CORE
Description	This register manages the MAILBOX13 clocks.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														IDLEST		RESERVED														MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state.	R	0x1

Table 3-734. CM_L4CFG_SPARE_SMARTREFLEX_RTC_CLKCTRL

Address Offset	0x0000 06A8		
Physical Address	0x4A00 8DA8	**Instance**	CORE_CM_CORE
Description	This register manages the SPARE_SMARTREFLEX_RTC clocks.		
Type	R		



Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state.	R	0x1

Table 3-735. CM_L4CFG_SPARE_SMARTREFLEX_SDRAM_CLKCTRL

Address Offset	0x0000 06B0		
Physical Address	0x4A00 8DB0	**Instance**	CORE_CM_CORE
Description	This register manages the SPARE_SMARTREFLEX_SDRAM clocks.		

Table 3-735. CM_L4CFG_SPARE_SMARTREFLEX_SDRAM_CLKCTRL (continued)

Type	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														IDLEST		RESERVED														MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state.	R	0x1

Table 3-736. CM_L4CFG_SPARE_SMARTREFLEX_WKUP_CLKCTRL

Address Offset	0x0000 06B8		
Physical Address	0x4A00 8DB8	Instance	CORE_CM_CORE
Description	This register manages the SPARE_SMARTREFLEX_WKUP clocks.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														IDLEST		RESERVED														MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state.	R	0x1

Table 3-737. CM_L4CFG_IO_DELAY_BLOCK_CLKCTRL

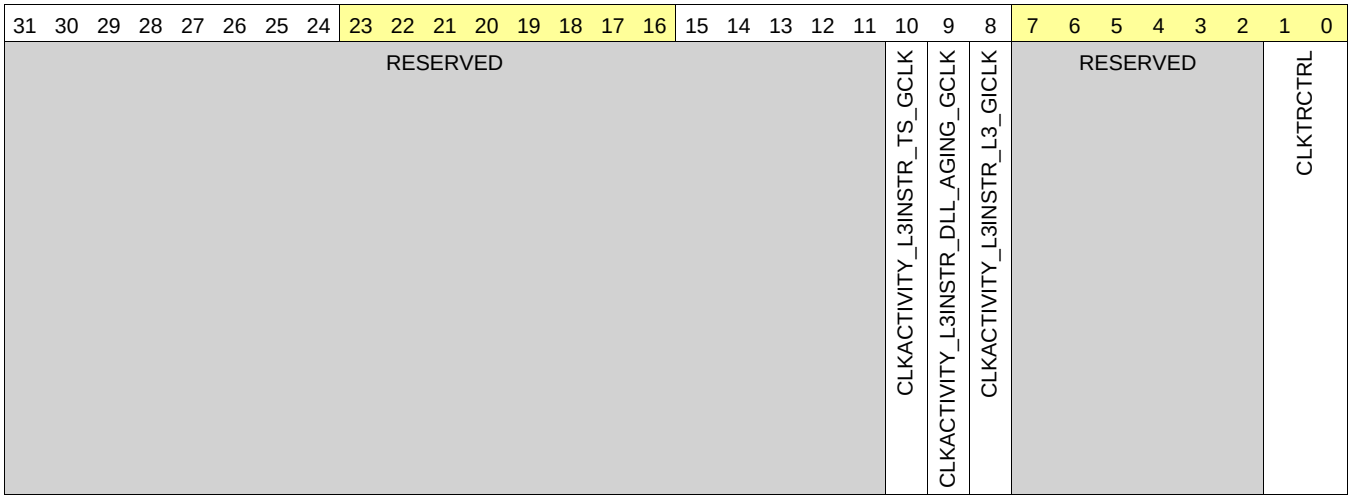
Address Offset	0x0000 06C0		
Physical Address	0x4A00 8DC0	**Instance**	CORE_CM_CORE
Description	This register manages the IO_DELAY_BLOCK clocks.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														IDLEST		RESERVED														MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state.	R	0x1

Table 3-738. CM_L3INSTR_CLKSTCTRL

Address Offset	0x0000 0700		
Physical Address	0x4A00 8E00	**Instance**	CORE_CM_CORE
Description	This register enables the domain power state transition. It controls the HW supervised domain power state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain.		
Type	R		



Bits	Field Name	Description	Type	Reset
31:11	RESERVED		R	0x0
10	CLKACTIVITY_L3INSTR_TS_GCLK	This field indicates the state of the L3INSTR_TS_GCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
9	CLKACTIVITY_L3INSTR_DLL_AGING_GCLK	This field indicates the state of the L3INSTR_DLL_AGING_GCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
8	CLKACTIVITY_L3INSTR_L3_GICLK	This field indicates the state of the L3INSTR_L3_GICLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
7:2	RESERVED		R	0x0
1:0	CLKTRCTRL	Controls the clock state transition of the L3INSTR clock domain. 0x3: HW_AUTO: Automatic transition is enabled. Sleep and wakeup transition are based upon hardware conditions.	R	0x3

Table 3-739. Register Call Summary for Register CM_L3INSTR_CLKSTCTRL

Clock Management Functional Description
- Clock Domain Modes: [0][1][2][3]

PRCM Register Manual
- CORE_CM_CORE Register Summary: [4]

Table 3-740. CM_L3INSTR_L3_MAIN_2_CLKCTRL

Address Offset	0x0000 0720		
Physical Address	0x4A00 8E20	**Instance**	CORE_CM_CORE
Description	This register manages the L3_MAIN_2 clocks. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														IDLEST		RESERVED								MODULEMODE							

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state. 0x2: Reserved 0x3: Reserved	RW	0x1

Table 3-741. Register Call Summary for Register CM_L3INSTR_L3_MAIN_2_CLKCTRL

Clock Management Functional Description • Clock Domain Module Attributes: [0][1]
PRCM Register Manual • CORE_CM_CORE Register Summary: [2] • RESTORE_CM_CORE Register Description: [3]

Table 3-742. CM_L3INSTR_L3_INSTR_CLKCTRL

Address Offset	0x0000 0728		
Physical Address	0x4A00 8E28	**Instance**	CORE_CM_CORE
Description	This register manages the L3 INSTRUMENTATION clocks. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														IDLEST		RESERVED								MODULEMODE							

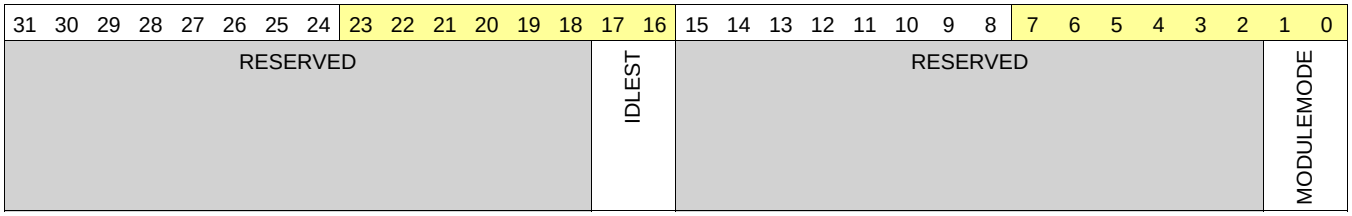
Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state. 0x2: Reserved 0x3: Reserved	RW	0x1

Table 3-743. Register Call Summary for Register CM_L3INSTR_L3_INSTR_CLKCTRL

Clock Management Functional Description • Clock Domain Module Attributes: [0][1]
PRCM Register Manual • CORE_CM_CORE Register Summary: [2] • RESTORE_CM_CORE Register Description: [3][4]

Table 3-744. CM_L3INSTR_OCP_WP_NOC_CLKCTRL

Address Offset	0x0000 0740		
Physical Address	0x4A00 8E40	**Instance**	CORE_CM_CORE
Description	This register manages the OCP_WP_NOC clocks. [warm reset insensitive]		
Type	RW		



Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3

Bits	Field Name	Description	Type	Reset
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state. 0x2: Reserved 0x3: Reserved	RW	0x1

Table 3-745. Register Call Summary for Register CM_L3INSTR_OCP_WP_NOC_CLKCTRL

Clock Management Functional Description
- Clock Domain Module Attributes: [0][1]

PRCM Register Manual
- CORE_CM_CORE Register Summary: [2]
- RESTORE_CM_CORE Register Description: [3][4]

Table 3-746. CM_L3INSTR_DLL_AGING_CLKCTRL

Address Offset	0x0000 0748		
Physical Address	0x4A00 8E48	**Instance**	CORE_CM_CORE
Description	This register manages the DLL_AGING clocks.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														IDLEST		RESERVED														MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state.	R	0x1

Table 3-747. Register Call Summary for Register CM_L3INSTR_DLL_AGING_CLKCTRL

Clock Management Functional Description • Clock Domain Module Attributes: [0][1]
PRCM Register Manual • CORE_CM_CORE Register Summary: [2]

Table 3-748. CM_L3INSTR_CTRL_MODULE_BANDGAP_CLKCTRL

Address Offset	0x0000 0750		
Physical Address	0x4A00 8E50	**Instance**	CORE_CM_CORE
Description	This register manages the CTRL_MODULE_BANDGAP clock.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED						CLKSEL		RESERVED						IDLEST		RESERVED														MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:24	CLKSEL	Selects the divider value for generating the Thermal Sensor clock from WKUPAON_ICLK source. The divider has to be selected so as to guarantee a frequency between 1MHz and 2MHz. 0x0: Divide by 8 0x1: Divide by 16 0x2: Divide by 32 0x3: Reserved	RW	0x2
23:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state.	R	0x1

Table 3-749. Register Call Summary for Register CM_L3INSTR_CTRL_MODULE_BANDGAP_CLKCTRL

Clock Management Functional Description • PRM Clock Source: [0] • Clock Domain Module Attributes: [1][2]

Table 3-749. Register Call Summary for Register CM_L3INSTR_CTRL_MODULE_BANDGAP_CLKCTRL (continued)

PRCM Register Manual • CORE_CM_CORE Register Summary: [3]

3.12.21 CUSTEFUSE_CM_CORE registers

3.12.21.1 CUSTEFUSE_CM_CORE Register Summary

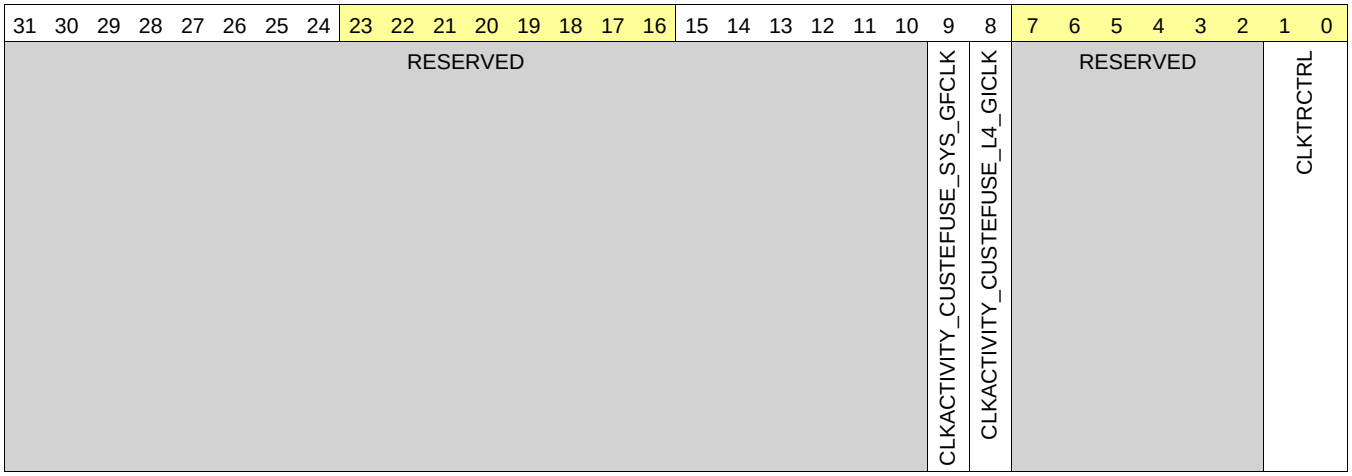
Table 3-750. CUSTEFUSE_CM_CORE Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	CUSTEFUSE_CM_CORE Physical Address
CM_CUSTEFUSE_CLKSTCTRL	RW	32	0x0000 0000	0x4A00 9600
CM_CUSTEFUSE_EFUSE_CTRL_CUST_CLKCTRL	RW	32	0x0000 0020	0x4A00 9620

3.12.21.2 CUSTEFUSE_CM_CORE Register Description

Table 3-751. CM_CUSTEFUSE_CLKSTCTRL

Address Offset	0x0000 0000		
Physical Address	0x4A00 9600	**Instance**	CUSTEFUSE_CM_CORE
Description	This register enables the domain power state transition. It controls the HW supervised domain power state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain.		
Type	RW		



Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	CLKACTIVITY_CUSTEFUSE_SYS_GFCLK	This field indicates the state of the Cust_Efuse_SYS_CLK clock input of the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0

Bits	Field Name	Description	Type	Reset
8	CLKACTIVITY_CUSTEFUSE_L4_GICLK	This field indicates the state of the L4_CUSTEFUSE_GICLK clock input of the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
7:2	RESERVED		R	0x0
1:0	CLKTRCTRL	Controls the clock state transition of the CUSTEFUSE clock domain. 0x0: NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may however occur. 0x1: Reserved 0x2: SW_WKUP: Start a software forced wake-up transition on the domain. 0x3: HW_AUTO: Automatic transition is enabled. Sleep and wakeup transition are based upon hardware conditions.	RW	0x3

Table 3-752. Register Call Summary for Register CM_CUSTEFUSE_CLKSTCTRL

Clock Management Functional Description • Clock Domain Modes: [0][1][2]
PRCM Register Manual • CUSTEFUSE_CM_CORE Register Summary: [3]

Table 3-753. CM_CUSTEFUSE_EFUSE_CTRL_CUST_CLKCTRL

Address Offset	0x0000 0020		
Physical Address	0x4A00 9620	**Instance**	CUSTEFUSE_CM_CORE
Description	This register manages the CUSTEFUSE clocks.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														IDLEST		RESERVED								MODULEMODE							

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guaranteed to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-754. Register Call Summary for Register CM_CUSTEFUSE_EFUSE_CTRL_CUST_CLKCTRL

Clock Management Functional Description
- Clock Domain Module Attributes: [0][1]

PRCM Register Manual
- CUSTEFUSE_CM_CORE Register Summary: [2]

3.12.22 DSS_CM_CORE registers

3.12.22.1 DSS_CM_CORE Register Summary

Table 3-755. DSS_CM_CORE Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	DSS_CM_CORE Physical Address
CM_DSS_CLKSTCTRL	RW	32	0x0000 0000	0x4A00 9100
CM_DSS_STATICDEP	RW	32	0x0000 0004	0x4A00 9104
CM_DSS_DYNAMICDEP	R	32	0x0000 0008	0x4A00 9108
CM_DSS_DSS_CLKCTRL	RW	32	0x0000 0020	0x4A00 9120
CM_DSS_BB2D_CLKCTRL	RW	32	0x0000 0030	0x4A00 9130
CM_DSS_SDVENC_CLKCTRL	RW	32	0x0000 003C	0x4A00 913C

3.12.22.2 DSS_CM_CORE Register Description

Table 3-756. CM_DSS_CLKSTCTRL

Address Offset	0x0000 0000		
Physical Address	0x4A00 9100	**Instance**	DSS_CM_CORE
Description	This register enables the DSS domain power state transition. It controls the HW supervised domain power state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED													CLKACTIVITY_HDMI_PHY_GFCLK	CLKACTIVITY_HDMI_CEC_GFCLK	CLKACTIVITY_DSS_SYS_GFCLK	CLKACTIVITY_DSS_L4_GICLK	CLKACTIVITY_SDVENC_GFCLK	CLKACTIVITY_BB2D_GFCLK	CLKACTIVITY_VIDEO2_DPLL_CLK	CLKACTIVITY_HDMI_DPLL_CLK	CLKACTIVITY_VIDEO1_DPLL_CLK	CLKACTIVITY_DSS_GFCLK	CLKACTIVITY_DSS_L3_GICLK	RESERVED						CLKTRCTRL	

Bits	Field Name	Description	Type	Reset
31:19	RESERVED		R	0x0
18	CLKACTIVITY_HDMI_PHY_GFCLK	This field indicates the state of the HDMI_PHY_GFCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
17	CLKACTIVITY_HDMI_CEC_GFCLK	This field indicates the state of the HDMI_CEC_GFCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
16	CLKACTIVITY_DSS_SYS_GFCLK	This field indicates the state of the DSS_SYS_GFCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
15	CLKACTIVITY_DSS_L4_GICLK	This field indicates the state of the DSS_L4_GICLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
14	CLKACTIVITY_SDVENC_GFCLK	This field indicates the state of the SDVENC_GFCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
13	CLKACTIVITY_BB2D_GFCLK	This field indicates the state of the BB2D_GFCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
12	CLKACTIVITY_VIDEO2_DPLL_CLK	This field indicates the state of the VIDEO2_PHY_GFCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
11	CLKACTIVITY_HDMI_DPLL_CLK	This field indicates the state of the HDMI_DPLL_CLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0

Bits	Field Name	Description	Type	Reset
10	CLKACTIVITY_VIDEO1_DPLL_CLK	This field indicates the state of the VIDEO1_DPLL_CLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
9	CLKACTIVITY_DSS_GFCLK	This field indicates the state of the DSS_GFCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
8	CLKACTIVITY_DSS_L3_GICLK	This field indicates the state of the DSS_L3_GICLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
7:2	RESERVED		R	0x0
1:0	CLKTRCTRL	Controls the clock state transition of the DSS clock domain. 0x0: NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may however occur. 0x1: SW_SLEEP: Start a software forced sleep transition on the domain. 0x2: SW_WKUP: Start a software forced wake-up transition on the domain. 0x3: HW_AUTO: Automatic transition is enabled. Sleep and wakeup transition are based upon hardware conditions.	RW	0x3

Table 3-757. Register Call Summary for Register CM_DSS_CLKSTCTRL

Clock Management Functional Description
- Clock Domain Modes: [0][4]

PRCM Register Manual
- Not Supported Functionality (Registers and Bitfields): [5][6][7][8][9][10][11]
- DSS_CM_CORE Register Summary: [12]

Table 3-758. CM_DSS_STATICDEP

Address Offset	0x0000 0004		
Physical Address	0x4A00 9104	**Instance**	DSS_CM_CORE
Description	This register controls the static domain depedencies from DSS domain towards 'target' domains. It is relevant only for domain having system initiator(s).		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																										L3MAIN1_STATDEP	EMIF_STATDEP	RESERVED	IVA_STATDEP	RESERVED	

Bits	Field Name	Description	Type	Reset
31:6	RESERVED		R	0x0
5	L3MAIN1_STATDEP	Static dependency towards L3MAIN1 clock domain 0x1: Dependency is enabled	R	0x1
4	EMIF_STATDEP	Static dependency towards EMIF clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	RESERVED		R	0x0
2	IVA_STATDEP	Static dependency towards IVA clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1:0	RESERVED		R	0x0

Table 3-759. Register Call Summary for Register CM_DSS_STATICDEP

Clock Management Functional Description • Clock Domain Dependency: [0][1]
PRCM Register Manual • DSS_CM_CORE Register Summary: [2]

Table 3-760. CM_DSS_DYNAMICDEP

Address Offset	0x0000 0008		
Physical Address	0x4A00 9108	**Instance**	DSS_CM_CORE
Description	This register controls the dynamic domain depedencies from DSS domain towards 'target' domains. It is relevant only for domain having OCP master port(s).		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																										L3MAIN1_DYNDEP	RESERVED				

Bits	Field Name	Description	Type	Reset
31:6	RESERVED		R	0x0
5	L3MAIN1_DYNDEP	Dynamic dependency towards L3MAIN1 domain 0x0: Dependency is disabled	R	0x0
4:0	RESERVED		R	0x0

Table 3-761. Register Call Summary for Register CM_DSS_DYNAMICDEP

Clock Management Functional Description • Clock Domain Dependency: [0]
PRCM Register Manual • DSS_CM_CORE Register Summary: [1]

Table 3-762. CM_DSS_DSS_CLKCTRL

Address Offset	0x0000 0020		
Physical Address	0x4A00 9120	**Instance**	DSS_CM_CORE
Description	This register manages the DSS clocks.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED													STBYST	IDLEST		RESERVED		OPTFCLKEN_VIDEO2_CLK	OPTFCLKEN_VIDEO1_CLK	OPTFCLKEN_32KHZ_CLK	OPTFCLKEN_HDMI_CLK	OPTFCLKEN_48MHZ_CLK	OPTFCLKEN_DSSCLK	RESERVED						MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:19	RESERVED		R	0x0
18	STBYST	Module standby status. [warm reset insensitive] 0x0: Module is functional (not in standby) 0x1: Module is in standby	R	0x1
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:14	RESERVED		R	0x0
13	OPTFCLKEN_VIDEO2_CLK	Optional functional clock control. 0x0: Optional functional clock is disabled 0x1: Optional functional clock is enabled	RW	0x0
12	OPTFCLKEN_VIDEO1_CLK	Optional functional clock control. 0x0: Optional functional clock is disabled 0x1: Optional functional clock is enabled	RW	0x0
11	OPTFCLKEN_32KHZ_CLK	Optional functional clock control. 0x0: Optional functional clock is disabled 0x1: Optional functional clock is enabled	RW	0x0
10	OPTFCLKEN_HDMI_CLK	Optional functional clock control. 0x0: Optional functional clock is disabled 0x1: Optional functional clock is enabled	RW	0x0
9	OPTFCLKEN_48MHZ_CLK	Optional functional clock control. 0x0: Optional functional clock is disabled 0x1: Optional functional clock is enabled	RW	0x0
8	OPTFCLKEN_DSSCLK	Optional functional clock control. 0x0: Optional functional clock is disabled 0x1: Optional functional clock is enabled	RW	0x0
7:2	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guaranteed to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-763. Register Call Summary for Register CM_DSS_DSS_CLKCTRL

Clock Management Functional Description
- Clock Domain Modes:
- Clock Domain Module Attributes: [1][2][3]

PRCM Register Manual
- Not Supported Functionality (Registers and Bitfields): [4][5][6]
- DSS_CM_CORE Register Summary: [7]

Table 3-764. CM_DSS_BB2D_CLKCTRL

Address Offset	0x0000 0030		
Physical Address	0x4A00 9130	**Instance**	DSS_CM_CORE
Description	This register manages the BB2D clocks.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED													STBYST	IDLEST		RESERVED														MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:19	RESERVED		R	0x0
18	STBYST	Module standby status. [warm reset insensitive] 0x0: Module is functional (not in standby) 0x1: Module is in standby	R	0x1
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guaranteed to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-765. CM_DSS_SDVENC_CLKCTRL

Address Offset	0x0000 003C		
Physical Address	0x4A00 913C	**Instance**	DSS_CM_CORE
Description	This register manages the SDVENC clocks.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														IDLEST		RESERVED														MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guaranteed to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-766. Register Call Summary for Register CM_DSS_SDVENC_CLKCTRL

Clock Management Functional Description
- Clock Domain Module Attributes:

PRCM Register Manual
- DSS_CM_CORE Register Summary: [2]

3.12.23 GPU_CM_CORE registers

3.12.23.1 GPU_CM_CORE Register Summary

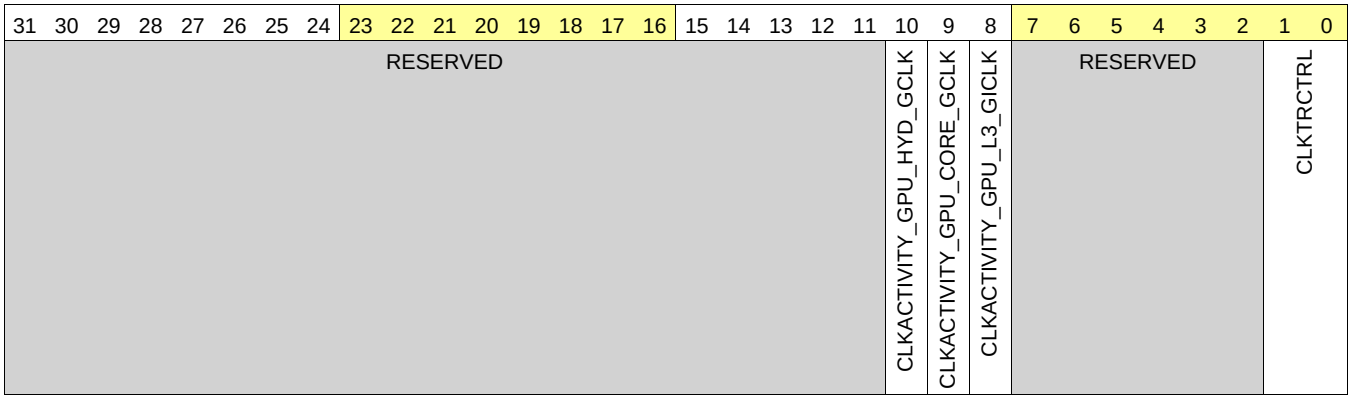
Table 3-767. GPU_CM_CORE Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	GPU_CM_CORE Physical Address
CM_GPU_CLKSTCTRL	RW	32	0x0000 0000	0x4A00 9200
CM_GPU_STATICDEP	RW	32	0x0000 0004	0x4A00 9204
CM_GPU_DYNAMICDEP	R	32	0x0000 0008	0x4A00 9208
CM_GPU_GPU_CLKCTRL	RW	32	0x0000 0020	0x4A00 9220

3.12.23.2 GPU_CM_CORE Register Description

Table 3-768. CM_GPU_CLKSTCTRL

Address Offset	0x0000 0000		
Physical Address	0x4A00 9200	**Instance**	GPU_CM_CORE
Description	This register enables the GPU domain power state transition. It controls the HW supervised domain power state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain.		
Type	RW		



Bits	Field Name	Description	Type	Reset
31:11	RESERVED		R	0x0
10	CLKACTIVITY_GPU_HYD_GCLK	This field indicates the state of the GPU_HYD_GCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
9	CLKACTIVITY_GPU_CORE_GCLK	This field indicates the state of the GPU_CORE_GCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
8	CLKACTIVITY_GPU_L3_GICLK	This field indicates the state of the GPU_L3_GICLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0

Bits	Field Name	Description	Type	Reset
7:2	RESERVED		R	0x0
1:0	CLKTRCTRL	Controls the clock state transition of the GPU clock domain. 0x0: NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may however occur. 0x1: SW_SLEEP: Start a software forced sleep transition on the domain. 0x2: SW_WKUP: Start a software forced wake-up transition on the domain. 0x3: HW_AUTO: Automatic transition is enabled. Sleep and wakeup transition are based upon hardware conditions.	RW	0x3

Table 3-769. CM_GPU_STATICDEP

Address Offset	0x0000 0004		
Physical Address	0x4A00 9204	**Instance**	GPU_CM_CORE
Description	This register controls the static domain depedencies from GPU domain towards 'target' domains. It is relevant only for domain having system initiator(s).		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																										L3MAIN1_STATDEP	EMIF_STATDEP	RESERVED	IVA_STATDEP	RESERVED	

Bits	Field Name	Description	Type	Reset
31:6	RESERVED		R	0x0
5	L3MAIN1_STATDEP	Static dependency towards L3MAIN1 clock domain 0x1: Dependency is enabled	R	0x1
4	EMIF_STATDEP	Static dependency towards EMIF clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	RESERVED		R	0x0
2	IVA_STATDEP	Static dependency towards IVA clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1:0	RESERVED		R	0x0

Table 3-770. CM_GPU_DYNAMICDEP

Address Offset	0x0000 0008		
Physical Address	0x4A00 9208	**Instance**	GPU_CM_CORE
Description	This register controls the dynamic domain depedencies from GPU domain towards 'target' domains. It is relevant only for domain having OCP master port(s).		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																									L3MAIN1_DYNDEP	RESERVED					

Bits	Field Name	Description	Type	Reset
31:7	RESERVED		R	0x0
6	L3MAIN1_DYNDEP	Dynamic dependency towards L3MAIN1 clock domain 0x0: Dependency is disabled	R	0x0
5:0	RESERVED		R	0x0

Table 3-771. CM_GPU_GPU_CLKCTRL

Address Offset	0x0000 0020		
Physical Address	0x4A00 9220	**Instance**	GPU_CM_CORE
Description	This register manages the GPU clocks.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				CLKSEL_HYD_CLK		CLKSEL_CORE_CLK		RESERVED					STBYST	IDLEST		RESERVED															MODULEMODE		

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:26	CLKSEL_HYD_CLK	Select the source of the functional clock 0x0: Selects the CORE_GPU_CLK as the source 0x1: Selects the PER_GPU_CLK 0x2: Selects GPU_GCLK 0x3: RESERVED	RW	0x0
25:24	CLKSEL_CORE_CLK	Select the source of the functional clock 0x0: Selects the CORE_GPU_CLK as the source 0x1: Selects the PER_GPU_CLK 0x2: Selects GPU_GCLK 0x3: RESERVED	RW	0x0
23:19	RESERVED		R	0x0
18	STBYST	Module standby status. [warm reset insensitive] 0x0: Module is functional (not in standby) 0x1: Module is in standby	R	0x1

Bits	Field Name	Description	Type	Reset
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guaranteed to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

3.12.24 IVA_CM_CORE registers

3.12.24.1 IVA_CM_CORE Register Summary

Table 3-772. IVA_CM_CORE Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	IVA_CM_CORE Physical Address
CM_IVA_CLKSTCTRL	RW	32	0x0000 0000	0x4A00 8F00
CM_IVA_STATICDEP	RW	32	0x0000 0004	0x4A00 8F04
CM_IVA_DYNAMICDEP	R	32	0x0000 0008	0x4A00 8F08
CM_IVA_IVA_CLKCTRL	RW	32	0x0000 0020	0x4A00 8F20
CM_IVA_SL2_CLKCTRL	RW	32	0x0000 0028	0x4A00 8F28

3.12.24.2 IVA_CM_CORE Register Description

Table 3-773. CM_IVA_CLKSTCTRL

Address Offset	0x0000 0000		
Physical Address	0x4A00 8F00	**Instance**	IVA_CM_CORE
Description	This register enables the IVA domain power state transition. It controls the HW supervised domain power state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							CLKACTIVITY_IVA_GCLK	RESERVED						CLKTRCTRL	

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	CLKACTIVITY_IVA_GCLK	This field indicates the state of the IVA_ROOT_CLK clock input of the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
7:2	RESERVED		R	0x0
1:0	CLKTRCTRL	Controls the clock state transition of the IVA clock domain. 0x0: NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may however occur. 0x1: SW_SLEEP: Start a software forced sleep transition on the domain. 0x2: SW_WKUP: Start a software forced wake-up transition on the domain. 0x3: HW_AUTO: Automatic transition is enabled. Sleep and wakeup transition are based upon hardware conditions.	RW	0x3

Table 3-774. CM_IVA_STATICDEP

Address Offset	0x0000 0004		
Physical Address	0x4A00 8F04	**Instance**	IVA_CM_CORE
Description	This register controls the static domain depedencies from IVA domain towards 'target' domains. It is relevant only for domain having system initiator(s).		
Type	RW		

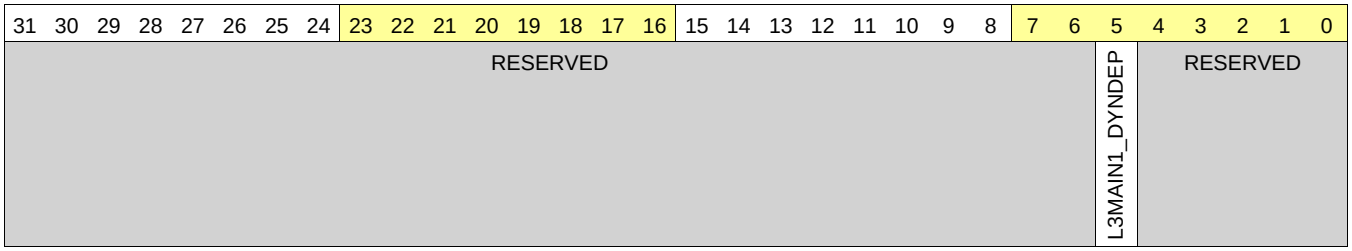
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																										L3MAIN1_STATDEP	EMIF_STATDEP	RESERVED			

Bits	Field Name	Description	Type	Reset
31:6	RESERVED		R	0x0
5	L3MAIN1_STATDEP	Static dependency towards L3MAIN1 clock domain 0x1: Dependency is enabled	R	0x1
4	EMIF_STATDEP	Static dependency towards EMIF clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
3:0	RESERVED		R	0x0

Table 3-775. CM_IVA_DYNAMICDEP

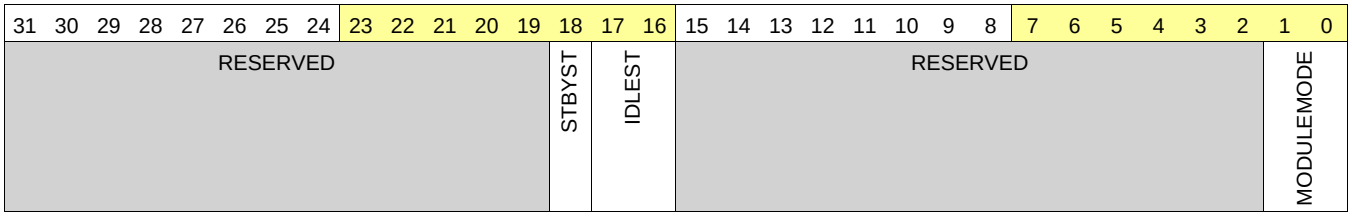
Address Offset	0x0000 0008		
Physical Address	0x4A00 8F08	**Instance**	IVA_CM_CORE
Description	This register controls the dynamic domain depedencies from IVA domain towards 'target' domains. It is relevant only for domain having OCP master port(s).		
Type	R		



Bits	Field Name	Description	Type	Reset
31:6	RESERVED		R	0x0
5	L3MAIN1_DYNDEP	Dynamic dependency towards L3MAIN1 clock domain 0x0: Dependency is disabled	R	0x0
4:0	RESERVED		R	0x0

Table 3-776. CM_IVA_IVA_CLKCTRL

Address Offset	0x0000 0020		
Physical Address	0x4A00 8F20	**Instance**	IVA_CM_CORE
Description	This register manages the IVA clocks.		
Type	RW		



Bits	Field Name	Description	Type	Reset
31:19	RESERVED		R	0x0
18	STBYST	Module standby status. [warm reset insensitive] 0x0: Module is functional (not in standby) 0x1: Module is in standby	R	0x1
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state. 0x2: Reserved 0x3: Reserved	RW	0x0

Table 3-777. CM_IVA_SL2_CLKCTRL

Address Offset	0x0000 0028		
Physical Address	0x4A00 8F28	**Instance**	IVA_CM_CORE
Description	This register manages the SL2 clocks.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														IDLEST		RESERVED								MODULEMODE							

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state. 0x2: Reserved 0x3: Reserved	RW	0x0

3.12.25 L3INIT_CM_CORE registers

3.12.25.1 L3INIT_CM_CORE Register Summary

Table 3-778. L3INIT_CM_CORE Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	L3INIT_CM_CORE Physical Address
CM_L3INIT_CLKSTCTRL	RW	32	0x0000 0000	0x4A00 9300
CM_L3INIT_STATICDEP	RW	32	0x0000 0004	0x4A00 9304
CM_L3INIT_DYNAMICDEP	R	32	0x0000 0008	0x4A00 9308
CM_L3INIT_MMC1_CLKCTRL	RW	32	0x0000 0028	0x4A00 9328
CM_L3INIT_MMC2_CLKCTRL	RW	32	0x0000 0030	0x4A00 9330
CM_L3INIT_USB_OTG_SS2_CLKCTRL	RW	32	0x0000 0040	0x4A00 9340
CM_L3INIT_USB_OTG_SS3_CLKCTRL	RW	32	0x0000 0048	0x4A00 9348
CM_L3INIT_USB_OTG_SS4_CLKCTRL	RW	32	0x0000 0050	0x4A00 9350
CM_L3INIT_MLB_SS_CLKCTRL	RW	32	0x0000 0058	0x4A00 9358
CM_L3INIT_IEEE1500_2_OCP_CLKCTRL	R	32	0x0000 0078	0x4A00 9378
CM_L3INIT_SATA_CLKCTRL	RW	32	0x0000 0088	0x4A00 9388
CM_PCIE_CLKSTCTRL	RW	32	0x0000 00A0	0x4A00 93A0
CM_PCIE_STATICDEP	RW	32	0x0000 00A4	0x4A00 93A4
CM_PCIE_PCIESS1_CLKCTRL	RW	32	0x0000 00B0	0x4A00 93B0
CM_PCIE_PCIESS2_CLKCTRL	RW	32	0x0000 00B8	0x4A00 93B8
CM_GMAC_CLKSTCTRL	RW	32	0x0000 00C0	0x4A00 93C0
CM_GMAC_STATICDEP	RW	32	0x0000 00C4	0x4A00 93C4
CM_GMAC_DYNAMICDEP	R	32	0x0000 00C8	0x4A00 93C8
CM_GMAC_GMAC_CLKCTRL	RW	32	0x0000 00D0	0x4A00 93D0
CM_L3INIT_OCP2SCP1_CLKCTRL	RW	32	0x0000 00E0	0x4A00 93E0
CM_L3INIT_OCP2SCP3_CLKCTRL	RW	32	0x0000 00E8	0x4A00 93E8
CM_L3INIT_USB_OTG_SS1_CLKCTRL	RW	32	0x0000 00F0	0x4A00 93F0

3.12.25.2 L3INIT_CM_CORE Register Description

Table 3-779. CM_L3INIT_CLKSTCTRL

Address Offset	0x0000 0000		
Physical Address	0x4A00 9300	**Instance**	L3INIT_CM_CORE
Description	This register enables the domain power state transition. It controls the HW supervised domain power state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED							CLKACTIVITY_SATA_REF_GFCLK	CLKACTIVITY_L3INIT_32K_GFCLK	CLKACTIVITY_L3INIT_960M_GFCLK	CLKACTIVITY_L3INIT_480M_GFCLK	CLKACTIVITY_USB_OTG_SS_REF_CLK	CLKACTIVITY_MLB_SYS_L3_GFCLK	CLKACTIVITY_MLB_SPB_L4_GICLK	CLKACTIVITY_MLB_SHB_L3_GICLK	CLKACTIVITY_MMC2_GFCLK	CLKACTIVITY_MMC1_GFCLK	CLKACTIVITY_HSI_GFCLK	CLKACTIVITY_USB_DPLL_HS_CLK	CLKACTIVITY_USB_DPLL_CLK	CLKACTIVITY_L3INIT_48M_GFCLK	CLKACTIVITY_L3INIT_USB_LFPS_TX_GFCLK	CLKACTIVITY_L3INIT_L4_GICLK	CLKACTIVITY_L3INIT_L3_GICLK	RESERVED						CLKTRCTRL

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24	CLKACTIVITY_SATA_REF_GFCLK	This field indicates the state of the SATA_REF_GFCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
23	CLKACTIVITY_L3INIT_32K_GFCLK	This field indicates the state of the L3INIT_32K_FCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
22	CLKACTIVITY_L3INIT_960M_GFCLK	This field indicates the state of the L3INIT_960M_GFCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
21	CLKACTIVITY_L3INIT_480M_GFCLK	This field indicates the state of the L3INIT_480M_GFCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
20	CLKACTIVITY_USB_OTG_SS_REF_CLK	This field indicates the state of the USB_OTG_SS_REF_CLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
19	CLKACTIVITY_MLB_SYS_L3_GFCLK	This field indicates the state of the MLB_SYS_L3_GFCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
18	CLKACTIVITY_MLB_SPB_L4_GICLK	This field indicates the state of the MLB_SPB_L4_GICLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0

Bits	Field Name	Description	Type	Reset
17	CLKACTIVITY_MLB_SHB_L3_GICLK	This field indicates the state of the MLB_SHB_L3_GICLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
16	CLKACTIVITY_MMC2_GFCLK	This field indicates the state of the MMC2 clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
15	CLKACTIVITY_MMC1_GFCLK	This field indicates the state of the MMC1_GFCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
14	CLKACTIVITY_HSI_GFCLK	This field indicates the state of the HSI_GFCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
13	CLKACTIVITY_USB_DPLL_HS_CLK	This field indicates the state of the USB_DPLL_HS_CLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
12	CLKACTIVITY_USB_DPLL_CLK	This field indicates the state of the USB_DPLL_CLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
11	CLKACTIVITY_L3INIT_48M_GFCLK	This field indicates the state of the INIT_48M_GFCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
10	CLKACTIVITY_L3INIT_USB_LFPS_TX_GFCLK	This field indicates the state of the L3INIT_USB_LFPS_TX_GFCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
9	CLKACTIVITY_L3INIT_L4_GICLK	This field indicates the state of the L3INIT_L4_GICLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
8	CLKACTIVITY_L3INIT_L3_GICLK	This field indicates the state of the L3INIT_L3_GICLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
7:2	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
1:0	CLKTRCTRL	Controls the clock state transition of the L3INIT clock domain. 0x0: NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may however occur. 0x1: SW_SLEEP: Start a software forced sleep transition on the domain. 0x2: SW_WKUP: Start a software forced wake-up transition on the domain. 0x3: HW_AUTO: Automatic transition is enabled. Sleep and wakeup transition are based upon hardware conditions.	RW	0x0

Table 3-780. Register Call Summary for Register CM_L3INIT_CLKSTCTRL

Clock Management Functional Description
- Clock Domain Modes: [0][1][2]

PRCM Register Manual
- Not Supported Functionality (Registers and Bitfields): [3][4][5][6][7][8][9][10][11][12][13][14][15]
- L3INIT_CM_CORE Register Summary: [16]
- RESTORE_CM_CORE Register Description: [17][18]

Table 3-781. CM_L3INIT_STATICDEP

Address Offset	0x0000 0004		
Physical Address	0x4A00 9304	**Instance**	L3INIT_CM_CORE
Description	This register controls the static domain depedencies from L3INIT domain towards 'target' domains. It is relevant only for domain having system initiator(s).		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				L4PER3_STATDEP	RESERVED											WKUPAON_STATDEP	L4SEC_STATDEP	L4PER_STATDEP	L4CFG_STATDEP	RESERVED				L3MAIN1_STATDEP	EMIF_STATDEP	RESERVED	IVA_STATDEP	RESERVED			

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27	L4PER3_STATDEP	Static dependency towards L4PER3 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
26:16	RESERVED		R	0x0
15	WKUPAON_STATDEP	Static dependency towards WKUPAON clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
14	L4SEC_STATDEP	Static dependency towards L4SEC clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
13	L4PER_STATDEP	Static dependency towards L4PER1 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
12	L4CFG_STATDEP	Static dependency towards L4CFG clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
11:6	RESERVED		R	0x0
5	L3MAIN1_STATDEP	Static dependency towards L3MAIN1 clock domain 0x1: Dependency is enabled	R	0x1
4	EMIF_STATDEP	Static dependency towards EMIF clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	RESERVED		R	0x0
2	IVA_STATDEP	Static dependency towards IVA clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1:0	RESERVED		R	0x0

Table 3-782. Register Call Summary for Register CM_L3INIT_STATICDEP

Clock Management Functional Description
- Clock Domain Dependency: [0][1][2][3][4][5]

PRCM Register Manual
- Not Supported Functionality (Registers and Bitfields): [6][7]
- L3INIT_CM_CORE Register Summary: [8]

Table 3-783. CM_L3INIT_DYNAMICDEP

Address Offset	0x0000 0008		
Physical Address	0x4A00 9308	**Instance**	L3INIT_CM_CORE
Description	This register controls the dynamic domain depedencies from L3INIT domain towards 'target' domains. It is relevant only for domain having OCP master port(s).		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																										L3MAIN1_DYNDEP	RESERVED				

Bits	Field Name	Description	Type	Reset
31:6	RESERVED		R	0x0
5	L3MAIN1_DYNDEP	Dynamic dependency towards L3MAIN1 clock domain 0x0: Dependency is disabled	R	0x0
4:0	RESERVED		R	0x0

Table 3-784. Register Call Summary for Register CM_L3INIT_DYNAMICDEP

Clock Management Functional Description
- Clock Domain Dependency: [0]

PRCM Register Manual
- L3INIT_CM_CORE Register Summary: [1]

Table 3-785. CM_L3INIT_MMC1_CLKCTRL

Address Offset	0x0000 0028		
Physical Address	0x4A00 9328	**Instance**	L3INIT_CM_CORE
Description	This register manages the MMC1 clocks.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED					CLKSEL_DIV		CLKSEL_SOURCE	RESERVED					STBYST	IDLEST		RESERVED							OPTFCLKEN_CLK32K	RESERVED						MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:25	CLKSEL_DIV	MMC1 clock divide ratio. 0x0: MMC1 clock is divided by 1. 0x1: MMC1 clock is divided by 2. 0x2: MMC1 clock is divided by 4. 0x3: RESERVED	RW	0x0
24	CLKSEL_SOURCE	Selects the source of the functional clock. 0x0: 128MHz clock derived from DPLL_PER is selected 0x1: 192MHz clock derived from DPLL_PER is selected	RW	0x0
23:19	RESERVED		R	0x0
18	STBYST	Module standby status. [warm reset insensitive] 0x0: Module is functional (not in standby) 0x1: Module is in standby	R	0x1
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:9	RESERVED		R	0x0
8	OPTFCLKEN_CLK32K	MMC optional clock control: 32K CLK 0x0: Optional functional clock is disabled 0x1: Optional functional clock is enabled	RW	0x0
7:2	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guaranteed to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-786. CM_L3INIT_MMC2_CLKCTRL

Address Offset	0x0000 0030		
Physical Address	0x4A00 9330	**Instance**	L3INIT_CM_CORE
Description	This register manages the MMC2 clocks.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED					CLKSEL_DIV		CLKSEL_SOURCE	RESERVED					STBYST	IDLEST		RESERVED							OPTFCLKEN_CLK32K	RESERVED						MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:25	CLKSEL_DIV	MMC2 clock divide ratio 0x0: MMC2 clock is divided by 1. 0x1: MMC2 clock is divided by 2. 0x2: MMC2 clock is divided by 4. 0x3: RESERVED	RW	0x0
24	CLKSEL_SOURCE	Selects the source of the functional clock. 0x0: 128MHz clock derived from DPLL_PER is selected 0x1: 192MHz clock derived from DPLL_PER is selected	RW	0x0
23:19	RESERVED		R	0x0
18	STBYST	Module standby status. [warm reset insensitive] 0x0: Module is functional (not in standby) 0x1: Module is in standby	R	0x1
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:9	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
8	OPTFCLKEN_CLK32K	MMC optional clock control: 32K CLK 0x0: Optional functional clock is disabled 0x1: Optional functional clock is enabled	RW	0x0
7:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guaranteed to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-787. CM_L3INIT_USB_OTG_SS2_CLKCTRL

Address Offset	0x0000 0040		
Physical Address	0x4A00 9340	**Instance**	L3INIT_CM_CORE
Description	This register manages the USB_OTG_SS2 clocks.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED													STBYST	IDLEST		RESERVED							OPTFCLKEN_REFCLK960M	RESERVED						MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:19	RESERVED		R	0x0
18	STBYST	Module standby status. [warm reset insensitive] 0x0: Module is functional (not in standby) 0x1: Module is in standby	R	0x1
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:9	RESERVED		R	0x0
8	OPTFCLKEN_REFCLK960M	USB_OTG_SS optional clock control: REFCLK960M (960MHz clock) 0x0: Optional functional clock is disabled 0x1: Optional functional clock is enabled	RW	0x0
7:2	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state. 0x2: Reserved 0x3: Reserved	RW	0x0

Table 3-788. CM_L3INIT_USB_OTG_SS3_CLKCTRL

Address Offset	0x0000 0048		
Physical Address	0x4A00 9348	**Instance**	L3INIT_CM_CORE
Description	This register manages the USB_OTG_SS3 clocks.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED													STBYST	IDLEST		RESERVED														MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:19	RESERVED		R	0x0
18	STBYST	Module standby status. [warm reset insensitive] 0x0: Module is functional (not in standby) 0x1: Module is in standby	R	0x1
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state. 0x2: Reserved 0x3: Reserved	RW	0x0

Table 3-789. CM_L3INIT_USB_OTG_SS4_CLKCTRL

Address Offset	0x0000 0050		
Physical Address	0x4A00 9350	**Instance**	L3INIT_CM_CORE
Description	This register manages the USB_OTG_SS4 clocks.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED													STBYST	IDLEST		RESERVED														MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:19	RESERVED		R	0x0
18	STBYST	Module standby status. [warm reset insensitive] 0x0: Module is functional (not in standby) 0x1: Module is in standby	R	0x1
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state. 0x2: Reserved 0x3: Reserved	RW	0x0

Table 3-790. CM_L3INIT_MLB_SS_CLKCTRL

Address Offset	0x0000 0058		
Physical Address	0x4A00 9358	**Instance**	L3INIT_CM_CORE
Description	This register manages the MLBSS clocks.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED													STBYST	IDLEST		RESERVED														MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:19	RESERVED		R	0x0
18	STBYST	Module standby status. [warm reset insensitive] 0x0: Module is functional (not in standby) 0x1: Module is in standby	R	0x1
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED	Reserved	R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guaranteed to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-791. CM_L3INIT_IEEE1500_2_OCP_CLKCTRL

Address Offset	0x0000 0078		
Physical Address	0x4A00 9378	**Instance**	L3INIT_CM_CORE
Description	This register manages the IEE1500_2_OCP clocks.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED													STBYST	IDLEST		RESERVED														MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:19	RESERVED	Reserved	R	0x0
18	STBYST	Module standby status. [warm reset insensitive] 0x0: Module is functional (not in standby) 0x1: Module is in standby	R	0x1
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED	Reserved	R	0x0

Bits	Field Name	Description	Type	Reset
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state.	R	0x1

Table 3-792. Register Call Summary for Register CM_L3INIT_IEEE1500_2_OCP_CLKCTRL

Clock Management Functional Description • Clock Domain Module Attributes: [0][1][2]
PRCM Register Manual • L3INIT_CM_CORE Register Summary: [3]

Table 3-793. CM_L3INIT_SATA_CLKCTRL

Address Offset	0x0000 0088		
Physical Address	0x4A00 9388	**Instance**	L3INIT_CM_CORE
Description	This register manages the SATA clocks.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED													STBYST	IDLEST		RESERVED							OPTFCLKEN_REF_CLK	RESERVED						MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:19	RESERVED	Reserved	R	0x0
18	STBYST	Module standby status. [warm reset insensitive] 0x0: Module is functional (not in standby) 0x1: Module is in standby	R	0x1
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:9	RESERVED		R	0x0
8	OPTFCLKEN_REF_CLK	SATA optional clock control: REF_CLK (from SYS_CLK clock) 0x0: Optional functional clock is disabled 0x1: Optional functional clock is enabled	RW	0x0
7:2	RESERVED	Reserved	R	0x0

Bits	Field Name	Description	Type	Reset
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guaranteed to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-794. CM_PCIE_CLKSTCTRL

Address Offset	0x0000 00A0		
Physical Address	0x4A00 93A0	**Instance**	L3INIT_CM_CORE
Description	This register enables the domain power state transition. It controls the HW supervised domain power state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																		CLKACTIVITY_PCIE_32K_GFCLK	CLKACTIVITY_PCIE_SYS_GFCLK	CLKACTIVITY_PCIE_REF_GFCLK	CLKACTIVITY_PCIE_PHY_DIV_GCLK	CLKACTIVITY_PCIE_PHY_GCLK	CLKACTIVITY_PCIE_L3_GICLK	RESERVED						CLKTRCTRL	

Bits	Field Name	Description	Type	Reset
31:14	RESERVED		R	0x0
13	CLKACTIVITY_PCIE_32K_GFCLK	This field indicates the state of the PCIE_32K_GFCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
12	CLKACTIVITY_PCIE_SYS_GFCLK	This field indicates the state of the PCIE_SYS_GFCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
11	CLKACTIVITY_PCIE_REF_GFCLK	This field indicates the state of the PCIE_REF_GFCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0

Bits	Field Name	Description	Type	Reset
10	CLKACTIVITY_PCIE_PHY_DIV_GCLK	This field indicates the state of the PCIE_PHY_DIV_GCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
9	CLKACTIVITY_PCIE_PHY_GCLK	This field indicates the state of the PCIE_PHY_GCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
8	CLKACTIVITY_PCIE_L3_GICLK	This field indicates the state of the PCIE_L3_GICLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
7:2	RESERVED		R	0x0
1:0	CLKTRCTRL	Controls the clock state transition of the L3INIT clock domain. 0x0: NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may however occur. 0x1: SW_SLEEP: Start a software forced sleep transition on the domain. 0x2: SW_WKUP: Start a software forced wake-up transition on the domain. 0x3: HW_AUTO: Automatic transition is enabled. Sleep and wakeup transition are based upon hardware conditions.	RW	0x0

Table 3-795. CM_PCIE_STATICDEP

Address Offset	0x0000 00A4		
Physical Address	0x4A00 93A4	**Instance**	L3INIT_CM_CORE
Description	This register controls the static domain depedencies from PCIE domain towards 'target' domains. It is relevant only for domain having system initiator(s).		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	CRC_STATDEP	RESERVED	ISS_STATDEP	L4PER3_STATDEP	L4PER2_STATDEP	GMAC_STATDEP	IPU_STATDEP	IPU1_STATDEP	EVE4_STATDEP	EVE3_STATDEP	EVE2_STATDEP	EVE1_STATDEP	DSP2_STATDEP	CUSTEFUSE_STATDEP	COREAON_STATDEP	RESERVED	L4SEC_STATDEP	L4PER_STATDEP	L4CFG_STATDEP	SDMA_STATDEP	GPU_STATDEP	CAM_STATDEP	DSS_STATDEP	L3INIT_STATDEP	RESERVED		EMIF_STATDEP	RESERVED	IVA_STATDEP	DSP1_STATDEP	RESERVED

Bits	Field Name	Description	Type	Reset
31	RESERVED		R	0x0
30	CRC_STATDEP	Static dependency towards CRC clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
29	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
28	ISS_STATDEP	Static dependency towards ISS clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
27	L4PER3_STATDEP	Static dependency towards L4PER3 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
26	L4PER2_STATDEP	Static dependency towards L4PER2 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
25	GMAC_STATDEP	Static dependency towards GMAC clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
24	IPU_STATDEP	Static dependency towards IPU clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
23	IPU1_STATDEP	Static dependency towards IPU1 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
22	EVE4_STATDEP	Static dependency towards EVE4 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
21	EVE3_STATDEP	Static dependency towards EVE3 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
20	EVE2_STATDEP	Static dependency towards EVE2 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
19	EVE1_STATDEP	Static dependency towards EVE1 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
18	DSP2_STATDEP	Static dependency towards DSP2 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
17	CUSTEFUSE_STATDEP	Static dependency towards CUSTEFUSE clock domain 0x0: Dependency is disabled	R	0x0
16	COREAON_STATDEP	Static dependency towards COREAON clock domain 0x0: Dependency is disabled	R	0x0
15	RESERVED		R	0x0
14	L4SEC_STATDEP	Static dependency towards L4SEC clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
13	L4PER_STATDEP	Static dependency towards L4PER clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
12	L4CFG_STATDEP	Static dependency towards L4CFG clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
11	SDMA_STATDEP	Static dependency towards SDMA clock domain 0x0: Dependency is disabled	R	0x0

Bits	Field Name	Description	Type	Reset
10	GPU_STATDEP	Static dependency towards GPU clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
9	CAM_STATDEP	Static dependency towards CAM clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
8	DSS_STATDEP	Static dependency towards DSS clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
7	L3INIT_STATDEP	Static dependency towards L3INIT clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6:5	RESERVED		R	0x0
4	EMIF_STATDEP	Static dependency towards EMIF clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	RESERVED		R	0x0
2	IVA_STATDEP	Static dependency towards IVA clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	DSP1_STATDEP	Static dependency towards DSP1 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	RESERVED		R	0x0

Table 3-796. CM_PCIE_PCIESS1_CLKCTRL

Address Offset	0x0000 00B0		
Physical Address	0x4A00 93B0	**Instance**	L3INIT_CM_CORE
Description	This register manages the PCESS1 clocks.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED													STBYST	IDLEST		RESERVED					OPTFCLKEN_PCIEPHY_CLK_DIV	OPTFCLKEN_PCIEPHY_CLK	OPTFCLKEN_32KHZ	RESERVED							MODULEMODE

Bits	Field Name	Description	Type	Reset
31:19	RESERVED		R	0x0
18	STBYST	Module standby status. [warm reset insensitive] 0x0: Module is functional (not in standby) 0x1: Module is in standby	R	0x1

Bits	Field Name	Description	Type	Reset
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:11	RESERVED		R	0x0
10	OPTFCLKEN_PCIEPHY_CLK_DIV	PCIE PHY optional clock control 0x0: Optional functional clock is disabled 0x1: Optional functional clock is enabled	RW	0x0
9	OPTFCLKEN_PCIEPHY_CLK	PCIE PHY optional clock control 0x0: Optional functional clock is disabled 0x1: Optional functional clock is enabled	RW	0x0
8	OPTFCLKEN_32KHZ	PCIE PHY optional clock control 0x0: Optional functional clock is disabled 0x1: Optional functional clock is enabled	RW	0x0
7:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guaranteed to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-797. CM_PCIE_PCIESS2_CLKCTRL

Address Offset	0x0000 00B8		
Physical Address	0x4A00 93B8	**Instance**	L3INIT_CM_CORE
Description	This register manages the PCESS2 clocks.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED													STBYST	IDLEST		RESERVED					OPTFCLKEN_PCIEPHY_CLK_DIV	OPTFCLKEN_PCIEPHY_CLK	OPTFCLKEN_32KHZ	RESERVED							MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:19	RESERVED		R	0x0
18	STBYST	Module standby status. [warm reset insensitive] 0x0: Module is functional (not in standby) 0x1: Module is in standby	R	0x1
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:11	RESERVED		R	0x0
10	OPTFCLKEN_PCIEPHY_CLK_DIV	PCIE PHY optional clock control 0x0: Optional functional clock is disabled 0x1: Optional functional clock is enabled	RW	0x0
9	OPTFCLKEN_PCIEPHY_CLK	PCIE PHY optional clock control 0x0: Optional functional clock is disabled 0x1: Optional functional clock is enabled	RW	0x0
8	OPTFCLKEN_32KHZ	PCIE PHY optional clock control 0x0: Optional functional clock is disabled 0x1: Optional functional clock is enabled	RW	0x0
7:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guaranteed to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-798. CM_GMAC_CLKSTCTRL

Address Offset	0x0000 00C0		
Physical Address	0x4A00 93C0	**Instance**	L3INIT_CM_CORE
Description	This register enables the domain power state transition. It controls the HW supervised domain power state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																			CLKACTIVITY_GMAC_MAIN_CLK	CLKACTIVITY_GMAC_RFT_CLK	CLKACTIVITY_RMII_50MHZ_CLK	CLKACTIVITY_RGMII_5MHZ_CLK	CLKACTIVITY_GMII_250MHZ_CLK	RESERVED						CLKTRCTRL	

Bits	Field Name	Description	Type	Reset
31:13	RESERVED		R	0x0
12	CLKACTIVITY_GMAC_MAIN_CLK	This field indicates the state of the GMAC_MAIN_CLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
11	CLKACTIVITY_GMAC_RFT_CLK	This field indicates the state of the GMAC_RFT_CLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
10	CLKACTIVITY_RMII_50MHZ_CLK	This field indicates the state of the RMII_50MHZ_CLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
9	CLKACTIVITY_RGMII_5MHZ_CLK	This field indicates the state of the RGMII_5MHZ_CLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
8	CLKACTIVITY_GMII_250MHZ_CLK	This field indicates the state of the GMII_250MHZ_CLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
7:2	RESERVED		R	0x0
1:0	CLKTRCTRL	**WARNING:** This bit field must not be programmed for SW_SLEEP or HW_AUTO when in EEE mode. Controls the clock state transition of the GMAC clock domain. 0x0: NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may however occur. 0x1: SW_SLEEP: Start a software forced sleep transition on the domain. 0x2: SW_WKUP: Start a software forced wake-up transition on the domain. 0x3: HW_AUTO: Automatic transition is enabled. Sleep and wakeup transition are based upon hardware conditions.	RW	0x0

Table 3-799. Register Call Summary for Register CM_GMAC_CLKSTCTRL

Clock Management Functional Description
• Clock Domain Modes: [0][1][2][3][4][5]
PRCM Register Manual
• L3INIT_CM_CORE Register Summary: [6]

Table 3-800. CM_GMAC_STATICDEP

Address Offset	0x0000 00C4		
Physical Address	0x4A00 93C4	**Instance**	L3INIT_CM_CORE
Description	This register controls the static domain depedencies from GMAC domain towards 'target' domains. It is relevant only for domain having system initiator(s).		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED					L4PER2_STATDEP	RESERVED																		L3MAIN1_STATDEP	EMIF_STATDEP	RESERVED							

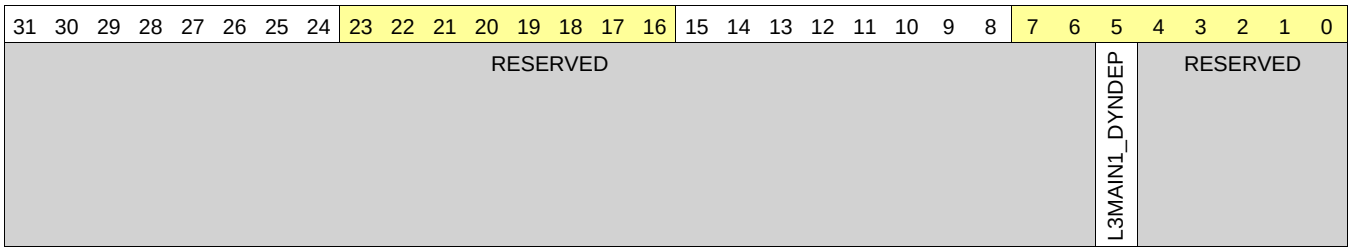
Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26	L4PER2_STATDEP	Static dependency towards L4PER2 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
25:6	RESERVED		R	0x0
5	L3MAIN1_STATDEP	Static dependency towards L3MAIN1 clock domain 0x1: Dependency is enabled	R	0x1
4	EMIF_STATDEP	Static dependency towards EMIF clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3:0	RESERVED		R	0x0

Table 3-801. Register Call Summary for Register CM_GMAC_STATICDEP

Clock Management Functional Description
• Clock Domain Dependency: [0][1][2]
PRCM Register Manual
• L3INIT_CM_CORE Register Summary: [3]

Table 3-802. CM_GMAC_DYNAMICDEP

Address Offset	0x0000 00C8		
Physical Address	0x4A00 93C8	**Instance**	L3INIT_CM_CORE
Description	This register controls the dynamic domain depedencies from GMAC domain towards 'target' domains. It is relevant only for domain having OCP master port(s).		
Type	R		



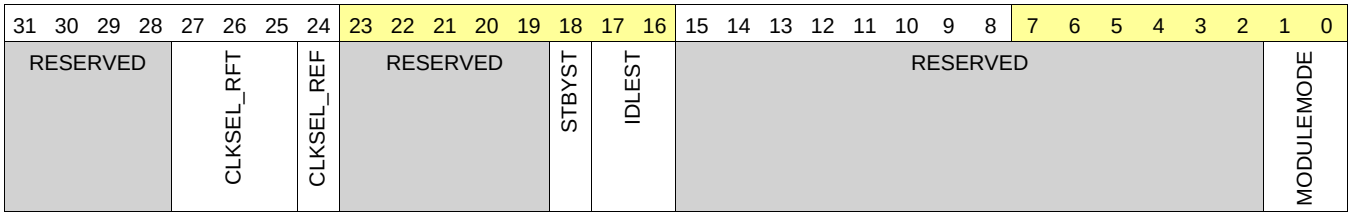
Bits	Field Name	Description	Type	Reset
31:6	RESERVED		R	0x0
5	L3MAIN1_DYNDEP	Dynamic dependency towards L3MAIN1 clock domain 0x0: Dependency is disabled	R	0x0
4:0	RESERVED		R	0x0

Table 3-803. Register Call Summary for Register CM_GMAC_DYNAMICDEP

Clock Management Functional Description • Clock Domain Dependency: [0]
PRCM Register Manual • L3INIT_CM_CORE Register Summary: [1]

Table 3-804. CM_GMAC_GMAC_CLKCTRL

Address Offset	0x0000 00D0		
Physical Address	0x4A00 93D0	**Instance**	L3INIT_CM_CORE
Description	This register manages the GMAC clocks.		
Type	RW		



Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:25	CLKSEL_RFT	Selects the source of the GMAC_RFT_CLK. [warm reset insensitive]. Repeated options are intentionally implemented for software compatibility with previous designs. 0x0: Selects SYS_CLK1 0x1: Selects SYS_CLK1 0x2: Selects SYS_CLK1 0x3: Selects SYS_CLK1 0x4: Selects L3_ICLK 0x5: RESERVED 0x6: RESERVED 0x7: RESERVED	RW	0x4

Bits	Field Name	Description	Type	Reset
24	CLKSEL_REF	Selects the source of the RMII_50MHZ_CLK functional clock. [warm reset insensitive] 0x0: GMAC_RMII_HS_CLK derived from DPLL_GMAC_DSP is selected 0x1: RESERVED	RW	0x0
23:19	RESERVED		R	0x0
18	STBYST	Module standby status. [warm reset insensitive] 0x0: Module is functional (not in standby) 0x1: Module is in standby	R	0x1
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED	Reserved	R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. [warm reset insensitive] 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guaranteed to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-805. Register Call Summary for Register CM_GMAC_GMAC_CLKCTRL

Clock Management Functional Description
- CM_CORE_AON Clock Generator: [0]
- Clock Domain Module Attributes: [1][2][3][4][5]

PRCM Register Manual
- Not Supported Functionality (Registers and Bitfields): [6][7][8][9][10]
- L3INIT_CM_CORE Register Summary: [11]

Table 3-806. CM_L3INIT_OCP2SCP1_CLKCTRL

Address Offset	0x0000 00E0		
Physical Address	0x4A00 93E0	**Instance**	L3INIT_CM_CORE
Description	This register manages the OCP2SCP1 clocks and the optional clock of USB PHY.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														IDLEST		RESERVED														MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state. 0x2: Reserved 0x3: Reserved	RW	0x0

Table 3-807. CM_L3INIT_OCP2SCP3_CLKCTRL

Address Offset	0x0000 00E8		
Physical Address	0x4A00 93E8	**Instance**	L3INIT_CM_CORE
Description	This register manages the OCP2SCP3 clocks and the optional clock of USB PHY.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														IDLEST		RESERVED														MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state. 0x2: Reserved 0x3: Reserved	RW	0x0

Table 3-808. CM_L3INIT_USB_OTG_SS1_CLKCTRL

Address Offset	0x0000 00F0		
Physical Address	0x4A00 93F0	**Instance**	L3INIT_CM_CORE
Description	This register manages the USB_OTG_SS1 clocks.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED													STBYST	IDLEST		RESERVED							OPTFCLKEN_REFCLK960M	RESERVED						MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:19	RESERVED		R	0x0
18	STBYST	Module standby status. [warm reset insensitive] 0x0: Module is functional (not in standby) 0x1: Module is in standby	R	0x1
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:9	RESERVED		R	0x0
8	OPTFCLKEN_REFCLK960M	USB_OTG_SS optional clock control: REFCLK960M (960MHz clock) 0x0: Optional functional clock is disabled 0x1: Optional functional clock is enabled	RW	0x0
7:2	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state. 0x2: Reserved 0x3: Reserved	RW	0x0

3.12.26 L4PER_CM_CORE registers

3.12.26.1 L4PER_CM_CORE Register Summary

Table 3-809. L4PER_CM_CORE Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	L4PER_CM_CORE Physical Address
CM_L4PER_CLKSTCTRL	RW	32	0x0000 0000	0x4A00 9700
CM_L4PER_DYNAMICDEP	RW	32	0x0000 0008	0x4A00 9708
CM_L4PER2_L4_PER2_CLKCTRL	R	32	0x0000 000C	0x4A00 970C
CM_L4PER3_L4_PER3_CLKCTRL	R	32	0x0000 0014	0x4A00 9714
CM_L4PER2_PRUSS1_CLKCTRL	RW	32	0x0000 0018	0x4A00 9718
CM_L4PER2_PRUSS2_CLKCTRL	RW	32	0x0000 0020	0x4A00 9720
CM_L4PER_DCC6_CLKCTRL	RW	32	0x0000 0028	0x4A00 9728
CM_L4PER_DCC7_CLKCTRL	RW	32	0x0000 0030	0x4A00 9730
CM_L4PER_TIMER2_CLKCTRL	RW	32	0x0000 0038	0x4A00 9738
CM_L4PER_TIMER3_CLKCTRL	RW	32	0x0000 0040	0x4A00 9740
CM_L4PER_TIMER4_CLKCTRL	RW	32	0x0000 0048	0x4A00 9748
CM_L4PER_DCC5_CLKCTRL	RW	32	0x0000 0050	0x4A00 9750
CM_L4PER_ELM_CLKCTRL	R	32	0x0000 0058	0x4A00 9758
CM_L4PER_GPIO2_CLKCTRL	RW	32	0x0000 0060	0x4A00 9760
CM_L4PER_GPIO3_CLKCTRL	RW	32	0x0000 0068	0x4A00 9768
CM_L4PER_GPIO4_CLKCTRL	RW	32	0x0000 0070	0x4A00 9770
CM_L4PER_GPIO5_CLKCTRL	RW	32	0x0000 0078	0x4A00 9778
CM_L4PER_GPIO6_CLKCTRL	RW	32	0x0000 0080	0x4A00 9780
CM_L4PER_ESM_CLKCTRL	RW	32	0x0000 0088	0x4A00 9788
CM_L4PER2_PWMSS2_CLKCTRL	RW	32	0x0000 0090	0x4A00 9790
CM_L4PER2_PWMSS3_CLKCTRL	RW	32	0x0000 0098	0x4A00 9798
CM_L4PER_I2C1_CLKCTRL	RW	32	0x0000 00A0	0x4A00 97A0
CM_L4PER_I2C2_CLKCTRL	RW	32	0x0000 00A8	0x4A00 97A8
CM_L4PER_I2C3_CLKCTRL	RW	32	0x0000 00B0	0x4A00 97B0
CM_L4PER_I2C4_CLKCTRL	RW	32	0x0000 00B8	0x4A00 97B8
CM_L4PER_L4_PER1_CLKCTRL	R	32	0x0000 00C0	0x4A00 97C0
CM_L4PER2_PWMSS1_CLKCTRL	RW	32	0x0000 00C4	0x4A00 97C4
CM_L4PER3_DCC1_CLKCTRL	RW	32	0x0000 00C8	0x4A00 97C8
CM_L4PER3_DCC2_CLKCTRL	RW	32	0x0000 00D0	0x4A00 97D0
CM_L4PER3_DCC3_CLKCTRL	RW	32	0x0000 00D8	0x4A00 97D8
CM_L4PER_MCSPI1_CLKCTRL	RW	32	0x0000 00F0	0x4A00 97F0

Table 3-809. L4PER_CM_CORE Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	L4PER_CM_CORE Physical Address
CM_L4PER_MCSPI2_CLKCTRL	RW	32	0x0000 00F8	0x4A00 97F8
CM_L4PER_MCSPI3_CLKCTRL	RW	32	0x0000 0100	0x4A00 9800
CM_L4PER_MCSPI4_CLKCTRL	RW	32	0x0000 0108	0x4A00 9808
CM_L4PER_GPIO7_CLKCTRL	RW	32	0x0000 0110	0x4A00 9810
CM_L4PER_GPIO8_CLKCTRL	RW	32	0x0000 0118	0x4A00 9818
CM_L4PER_MMC3_CLKCTRL	RW	32	0x0000 0120	0x4A00 9820
CM_L4PER_MMC4_CLKCTRL	RW	32	0x0000 0128	0x4A00 9828
CM_L4PER3_DCC4_CLKCTRL	RW	32	0x0000 0130	0x4A00 9830
CM_L4PER2_QSPI_CLKCTRL	RW	32	0x0000 0138	0x4A00 9838
CM_L4PER_UART1_CLKCTRL	RW	32	0x0000 0140	0x4A00 9840
CM_L4PER_UART2_CLKCTRL	RW	32	0x0000 0148	0x4A00 9848
CM_L4PER_UART3_CLKCTRL	RW	32	0x0000 0150	0x4A00 9850
CM_L4PER_UART4_CLKCTRL	RW	32	0x0000 0158	0x4A00 9858
CM_L4PER2_ADC_CLKCTRL	RW	32	0x0000 0160	0x4A00 9860
CM_L4PER2_ATL_CLKCTRL	RW	32	0x0000 0168	0x4A00 9868
CM_L4PER_UART5_CLKCTRL	RW	32	0x0000 0170	0x4A00 9870
CM_L4PER2_MCASP5_CLKCTRL	RW	32	0x0000 0178	0x4A00 9878
CM_L4SEC_CLKSTCTRL	RW	32	0x0000 0180	0x4A00 9880
CM_L4SEC_STATICDEP	RW	32	0x0000 0184	0x4A00 9884
CM_L4SEC_DYNAMICDEP	R	32	0x0000 0188	0x4A00 9888
CM_L4PER2_MCASP8_CLKCTRL	RW	32	0x0000 0190	0x4A00 9890
CM_L4PER2_MCASP4_CLKCTRL	RW	32	0x0000 0198	0x4A00 9898
CM_L4SEC_AES1_CLKCTRL	RW	32	0x0000 01A0	0x4A00 98A0
CM_L4SEC_AES2_CLKCTRL	RW	32	0x0000 01A8	0x4A00 98A8
CM_L4SEC_DES3DES_CLKCTRL	RW	32	0x0000 01B0	0x4A00 98B0
CM_L4SEC_FPKA_CLKCTRL	RW	32	0x0000 01B8	0x4A00 98B8
CM_L4SEC_RNG_CLKCTRL	RW	32	0x0000 01C0	0x4A00 98C0
CM_L4SEC_SHA2MD51_CLKCTRL	RW	32	0x0000 01C8	0x4A00 98C8
CM_L4PER2_UART7_CLKCTRL	RW	32	0x0000 01D0	0x4A00 98D0
CM_L4SEC_DMA_CRYPTO_CLKCTRL	R	32	0x0000 01D8	0x4A00 98D8
CM_L4PER2_UART8_CLKCTRL	RW	32	0x0000 01E0	0x4A00 98E0
CM_L4PER2_UART9_CLKCTRL	RW	32	0x0000 01E8	0x4A00 98E8
CM_L4PER2_DCAN2_CLKCTRL	RW	32	0x0000 01F0	0x4A00 98F0
CM_L4SEC_SHA2MD52_CLKCTRL	RW	32	0x0000 01F8	0x4A00 98F8
CM_L4PER2_CLKSTCTRL	RW	32	0x0000 01FC	0x4A00 98FC
CM_L4PER2_DYNAMICDEP	RW	32	0x0000 0200	0x4A00 9900
CM_L4PER2_MCASP6_CLKCTRL	RW	32	0x0000 0204	0x4A00 9904
CM_L4PER2_MCASP7_CLKCTRL	RW	32	0x0000 0208	0x4A00 9908
CM_L4PER2_STATICDEP	RW	32	0x0000 020C	0x4A00 990C
CM_L4PER3_CLKSTCTRL	RW	32	0x0000 0210	0x4A00 9910
CM_L4PER3_DYNAMICDEP	RW	32	0x0000 0214	0x4A00 9914

3.12.26.2 L4PER_CM_CORE Register Description

Table 3-810. CM_L4PER_CLKSTCTRL

Address Offset	0x0000 0000		
Physical Address	0x4A00 9700	**Instance**	L4PER_CM_CORE
Description	This register enables the domain power state transition. It controls the HW supervised domain power state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				CLKACTIVITY_L4PER_32K_GFCLK	CLKACTIVITY_UART5_GFCLK	CLKACTIVITY_PER_192M_GFCLK	CLKACTIVITY_GPIO_GFCLK	CLKACTIVITY_MMC4_GFCLK	CLKACTIVITY_MMC3_GFCLK	CLKACTIVITY_PER_96M_GFCLK	CLKACTIVITY_PER_48M_GFCLK	CLKACTIVITY_PER_12M_GFCLK	CLKACTIVITY_UART4_GFCLK	CLKACTIVITY_UART3_GFCLK	CLKACTIVITY_UART2_GFCLK	CLKACTIVITY_UART1_GFCLK	CLKACTIVITY_DCC5_GFCLK	CLKACTIVITY_TIMER4_GFCLK	CLKACTIVITY_TIMER3_GFCLK	CLKACTIVITY_TIMER2_GFCLK	CLKACTIVITY_DCC7_GFCLK	CLKACTIVITY_DCC6_GFCLK	CLKACTIVITY_L4PER_L3_GICLK	RESERVED						CLKTRCTRL	

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27	CLKACTIVITY_L4PER_32K_GFCLK	This field indicates the state of the L4PER_32K_FCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
26	CLKACTIVITY_UART5_GFCLK	This field indicates the state of the UART5_GFCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
25	CLKACTIVITY_PER_192M_GFCLK	This field indicates the state of the PER_192M_GFCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
24	CLKACTIVITY_GPIO_GFCLK	This field indicates the state of the GPIO_GFCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
23	CLKACTIVITY_MMC4_GFCLK	This field indicates the state of the MMC4_GFCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0

Bits	Field Name	Description	Type	Reset
22	CLKACTIVITY_MMC3_GFCLK	This field indicates the state of the MMC3_GFCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
21	CLKACTIVITY_PER_96M_GFCLK	This field indicates the state of the PER_96M_GFCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
20	CLKACTIVITY_PER_48M_GFCLK	This field indicates the state of the PER_48M_GFCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
19	CLKACTIVITY_PER_12M_GFCLK	This field indicates the state of the PER_12M_GFCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
18	CLKACTIVITY_UART4_GFCLK	This field indicates the state of the UART4_GFCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
17	CLKACTIVITY_UART3_GFCLK	This field indicates the state of the UART3_GFCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
16	CLKACTIVITY_UART2_GFCLK	This field indicates the state of the UART2_GFCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
15	CLKACTIVITY_UART1_GFCLK	This field indicates the state of the UART1_GFCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
14	CLKACTIVITY_DCC5_GFCLK	This field indicates the state of the DMT9_GFCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
13	CLKACTIVITY_TIMER4_GFCLK	This field indicates the state of the DMT4_GFCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
12	CLKACTIVITY_TIMER3_GFCLK	This field indicates the state of the DMT3_GFCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0

Bits	Field Name	Description	Type	Reset
11	CLKACTIVITY_TIMER2_GFCLK	This field indicates the state of the DMT2_GFCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
10	CLKACTIVITY_DCC7_GFCLK	This field indicates the state of the DMT11_GFCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
9	CLKACTIVITY_DCC6_GFCLK	This field indicates the state of the DMT10_GFCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
8	CLKACTIVITY_L4PER_L3_GICLK	This field indicates the state of the L4PER_L3_GICLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
7:2	RESERVED		R	0x0
1:0	CLKTRCTRL	Controls the clock state transition of the L4PER clock domain. 0x0: NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may however occur. 0x1: SW_SLEEP: Start a software forced sleep transition on the domain. 0x2: SW_WKUP: Start a software forced wake-up transition on the domain. 0x3: HW_AUTO: Automatic transition is enabled. Sleep and wakeup transition are based upon hardware conditions.	RW	0x0

Table 3-811. Register Call Summary for Register CM_L4PER_CLKSTCTRL

Clock Management Functional Description
- Clock Domain Modes: [0][1][2][3][4][5][6][7][8][9][10][11][12]

PRCM Register Manual
- Not Supported Functionality (Registers and Bitfields): [13][14][15][16][17][18]
- L4PER_CM_CORE Register Summary: [19]
- RESTORE_CM_CORE Register Description: [20][21]

Table 3-812. CM_L4PER_DYNAMICDEP

Address Offset	0x0000 0008		
Physical Address	0x4A00 9708	**Instance**	L4PER_CM_CORE
Description	This register controls the dynamic domain depedencies from L4PER domain towards 'target' domains. It is relevant only for domain having OCP master port(s).		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				WINDOWSIZE				RESERVED								L4SEC_DYNDEP	RESERVED						DSS_DYNDEP	L3INIT_DYNDEP	RESERVED			IPU_DYNDEP	RESERVED		

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:24	WINDOWSIZE	Size of sliding window used to monitor OCP interface activity for determination of auto-sleep feature. Time unit defined by CM_DYN_DEP_PRESCAL register.	RW	0x4
23:15	RESERVED		R	0x0
14	L4SEC_DYNDEP	Dynamic dependency towards L4SEC clock domain 0x1: Dependency is enabled	R	0x1
13:9	RESERVED		R	0x0
8	DSS_DYNDEP	Dynamic dependency towards DSS clock domain 0x1: Dependency is enabled	R	0x1
7	L3INIT_DYNDEP	Dynamic dependency towards L3INIT clock domain 0x1: Dependency is enabled	R	0x1
6:4	RESERVED		R	0x0
3	IPU_DYNDEP	Dynamic dependency towards IPU clock domain 0x1: Dependency is enabled	R	0x1
2:0	RESERVED		R	0x0

Table 3-813. Register Call Summary for Register CM_L4PER_DYNAMICDEP

Clock Management Functional Description
- Clock Domain Dependency: [0][1][2]

PRCM Register Manual
- Not Supported Functionality (Registers and Bitfields): [3]
- L4PER_CM_CORE Register Summary: [4]
- RESTORE_CM_CORE Register Description: [5][6]

Table 3-814. CM_L4PER2_L4_PER2_CLKCTRL

Address Offset	0x0000 000C		
Physical Address	0x4A00 970C	**Instance**	L4PER_CM_CORE
Description	This register manages the L4_PER2 clocks.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														IDLEST		RESERVED														MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state.	R	0x1

Table 3-815. Register Call Summary for Register CM_L4PER2_L4_PER2_CLKCTRL

Clock Management Functional Description • Clock Domain Module Attributes: [0][1]
PRCM Register Manual • L4PER_CM_CORE Register Summary: [2]

Table 3-816. CM_L4PER3_L4_PER3_CLKCTRL

Address Offset	0x0000 0014		
Physical Address	0x4A00 9714	**Instance**	L4PER_CM_CORE
Description	This register manages the L4_PER3 clocks.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														IDLEST		RESERVED														MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state.	R	0x1

Table 3-817. Register Call Summary for Register CM_L4PER3_L4_PER3_CLKCTRL

Clock Management Functional Description
- Clock Domain Module Attributes: [0][1]

PRCM Register Manual
- L4PER_CM_CORE Register Summary: [2]

Table 3-818. CM_L4PER2_PRUSS1_CLKCTRL

Address Offset	0x0000 0018		
Physical Address	0x4A00 9718	**Instance**	L4PER_CM_CORE
Description	This register manages the PRUSS clocks.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED													STBYST	IDLEST		RESERVED								MODULEMODE							

Bits	Field Name	Description	Type	Reset
31:19	RESERVED		R	0x0
18	STBYST	Module standby status. [warm reset insensitive] 0x0: Module is functional (not in standby) 0x1: Module is in standby	R	0x1
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guaranteed to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-819. CM_L4PER2_PRUSS2_CLKCTRL

Address Offset	0x0000 0020		
Physical Address	0x4A00 9720	**Instance**	L4PER_CM_CORE
Description	This register manages the PRUSS clocks.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED													STBYST	IDLEST		RESERVED														MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:19	RESERVED		R	0x0
18	STBYST	Module standby status. [warm reset insensitive] 0x0: Module is functional (not in standby) 0x1: Module is in standby	R	0x1
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guaranteed to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-820. CM_L4PER_DCC6_CLKCTRL

Address Offset	0x0000 0028		
Physical Address	0x4A00 9728	**Instance**	L4PER_CM_CORE
Description	This register manages the DCC6 clocks.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				CLKSEL				RESERVED						IDLEST		RESERVED														MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:24	CLKSEL	Select the source of the functional clock 0x0: Selects the TIMER_SYS_CLK as the source 0x1: Selects the SYS_CLK1_32K_CLK as the source 0x2: Selects SYS_CLK2 0x3: Selects XREF_CLK0 0x4: Selects XREF_CLK1 0x5: Selects XREF_CLK2 0x6: Selects XREF_CLK3 0x7: Selects ABE_GICLK 0x8: Selects VIDEO1 CLK 0x9: Selects VIDEO2 CLK 0xA: Selects HDMI CLK 0xB: RESERVED 0xC: RESERVED1 0xD: RESERVED2 0xE: RESERVED3 0xF: RESERVED4	RW	0x0
23:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guaranteed to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-821. Register Call Summary for Register CM_L4PER_DCC6_CLKCTRL

Clock Management Functional Description
- Clock Domain Module Attributes: [0][1]

PRCM Register Manual
- Not Supported Functionality (Registers and Bitfields): [2]
- L4PER_CM_CORE Register Summary: [3]

Table 3-822. CM_L4PER_DCC7_CLKCTRL

Address Offset	0x0000 0030		
Physical Address	0x4A00 9730	**Instance**	L4PER_CM_CORE
Description	This register manages the DCC7 clocks.		
Type	RW		

31 30 29 28	27 26 25 24	23 22 21 20 19 18	17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2	1 0
RESERVED	CLKSEL	RESERVED	IDLEST	RESERVED		MODULEMODE

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:24	CLKSEL	Select the source of the functional clock 0x0: Selects the TIMER_SYS_CLK as the source 0x1: Selects the SYS_CLK1_32K_CLK as the source 0x2: Selects SYS_CLK2 0x3: Selects XREF_CLK0 0x4: Selects XREF_CLK1 0x5: Selects XREF_CLK2 0x6: Selects XREF_CLK3 0x7: Selects ABE_GICLK 0x8: Selects VIDEO1 CLK 0x9: Selects VIDEO2 CLK 0xA: Selects HDMI CLK 0xB: RESERVED 0xC: RESERVED1 0xD: RESERVED2 0xE: RESERVED3 0xF: RESERVED4	RW	0x0
23:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guaranteed to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-823. Register Call Summary for Register CM_L4PER_DCC7_CLKCTRL

Clock Management Functional Description
- Clock Domain Module Attributes: [0][1]

PRCM Register Manual
- Not Supported Functionality (Registers and Bitfields): [2]
- L4PER_CM_CORE Register Summary: [3]

Table 3-824. CM_L4PER_TIMER2_CLKCTRL

Address Offset	0x0000 0038		
Physical Address	0x4A00 9738	**Instance**	L4PER_CM_CORE
Description	This register manages the TIMER2 clocks.		
Type	RW		

31 30 29 28	27 26 25 24	23 22 21 20 19 18	17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2	1 0
RESERVED	CLKSEL	RESERVED	IDLEST	RESERVED		MODULEMODE

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:24	CLKSEL	Select the source of the functional clock 0x0: Selects SYS_CLK1 0x1: Selects FUNC_32K_CLK 0x2: Selects SYS_CLK2 0x3: Selects XREF_CLK0 0x4: Selects XREF_CLK1 0x5: Selects XREF_CLK0 0x6: Selects XREF_CLK1 0x7: Selects ABE_GICLK 0x8: Selects divided version of SYS_CLK1. See CM_CLKSEL_VIDEO1_TIMER 0x9: Selects divided version of SYS_CLK1. See CM_CLKSEL_VIDEO2_TIMER 0xA: Selects divided version of SYS_CLK1. See CM_CLKSEL_HDMI_TIMER Others: RESERVED	RW	0x0
23:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guaranteed to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-825. Register Call Summary for Register CM_L4PER_TIMER2_CLKCTRL

Clock Management Functional Description • CM_CORE_AON_TIMER Overview: [0] • Clock Domain Module Attributes: [1][2]
PRCM Register Manual • Not Supported Functionality (Registers and Bitfields): [3][4][5][6][7] • L4PER_CM_CORE Register Summary: [8]

Table 3-826. CM_L4PER_TIMER3_CLKCTRL

Address Offset	0x0000 0040		
Physical Address	0x4A00 9740	**Instance**	L4PER_CM_CORE
Description	This register manages the TIMER3 clocks.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				CLKSEL				RESERVED						IDLEST		RESERVED														MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:24	CLKSEL	Select the source of the functional clock 0x0: Selects SYS_CLK1 0x1: Selects FUNC_32K_CLK 0x2: Selects SYS_CLK2 0x3: Selects XREF_CLK0 0x4: Selects XREF_CLK1 0x5: Selects XREF_CLK0 0x6: Selects XREF_CLK1 0x7: Selects ABE_GICLK 0x8: Selects divided version of SYS_CLK1. See CM_CLKSEL_VIDEO1_TIMER 0x9: Selects divided version of SYS_CLK1. See CM_CLKSEL_VIDEO2_TIMER 0xA: Selects divided version of SYS_CLK1. See CM_CLKSEL_HDMI_TIMER Others: RESERVED	RW	0x0
23:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guaranteed to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-827. Register Call Summary for Register CM_L4PER_TIMER3_CLKCTRL

Clock Management Functional Description
- CM_CORE_AON_TIMER Overview: [0]
- Clock Domain Module Attributes: [1][2]

PRCM Register Manual
- Not Supported Functionality (Registers and Bitfields): [3][4][5][6][7]
- L4PER_CM_CORE Register Summary: [8]

Table 3-828. CM_L4PER_TIMER4_CLKCTRL

Address Offset	0x0000 0048		
Physical Address	0x4A00 9748	**Instance**	L4PER_CM_CORE
Description	This register manages the TIMER4 clocks.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				CLKSEL				RESERVED						IDLEST		RESERVED														MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:24	CLKSEL	Select the source of the functional clock 0x0: Selects SYS_CLK1 0x1: Selects FUNC_32K_CLK 0x2: Selects SYS_CLK2 0x3: Selects XREF_CLK0 0x4: Selects XREF_CLK1 0x5: Selects XREF_CLK0 0x6: Selects XREF_CLK1 0x7: Selects ABE_GICLK 0x8: Selects divided version of SYS_CLK1. See CM_CLKSEL_VIDEO1_TIMER 0x9: Selects divided version of SYS_CLK1. See CM_CLKSEL_VIDEO2_TIMER 0xA: Selects divided version of SYS_CLK1. See CM_CLKSEL_HDMI_TIMER Others: RESERVED	RW	0x0

Bits	Field Name	Description	Type	Reset
23:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guaranteed to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-829. Register Call Summary for Register CM_L4PER_TIMER4_CLKCTRL

Clock Management Functional Description
- CM_CORE_AON_TIMER Overview: [0]
- Clock Domain Module Attributes: [1][2]

PRCM Register Manual
- Not Supported Functionality (Registers and Bitfields): [3][4][5][6][7]
- L4PER_CM_CORE Register Summary: [8]

Table 3-830. CM_L4PER_DCC5_CLKCTRL

Address Offset	0x0000 0050		
Physical Address	0x4A00 9750	**Instance**	L4PER_CM_CORE
Description	This register manages the DCC5 clocks.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				CLKSEL				RESERVED						IDLEST		RESERVED														MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:24	CLKSEL	Select the source of the functional clock 0x0: Selects the TIMER_SYS_CLK as the source 0x1: Selects the SYS_CLK1_32K_CLK as the source 0x2: Selects SYS_CLK2 0x3: Selects XREF_CLK0 0x4: Selects XREF_CLK1 0x5: Selects XREF_CLK2 0x6: Selects XREF_CLK3 0x7: Selects PER_ABE_X1 CLK 0x8: Selects VIDEO1 CLK 0x9: Selects VIDEO2 CLK 0xA: Selects HDMI CLK 0xB: RESERVED 0xC: RESERVED1 0xD: RESERVED2 0xE: RESERVED3 0xF: RESERVED4	RW	0x0
23:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guaranteed to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-831. Register Call Summary for Register CM_L4PER_DCC5_CLKCTRL

Clock Management Functional Description
- Clock Domain Module Attributes: [0][1]

PRCM Register Manual
- Not Supported Functionality (Registers and Bitfields): [2]
- L4PER_CM_CORE Register Summary: [3]

Table 3-832. CM_L4PER_ELM_CLKCTRL

Address Offset	0x0000 0058		
Physical Address	0x4A00 9758	**Instance**	L4PER_CM_CORE
Description	This register manages the ELM clocks.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														IDLEST		RESERVED								MODULEMODE							

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state.	R	0x1

Table 3-833. Register Call Summary for Register CM_L4PER_ELM_CLKCTRL

Clock Management Functional Description • Clock Domain Module Attributes: [0][1]
PRCM Register Manual • L4PER_CM_CORE Register Summary: [2]

Table 3-834. CM_L4PER_GPIO2_CLKCTRL

Address Offset	0x0000 0060		
Physical Address	0x4A00 9760	**Instance**	L4PER_CM_CORE
Description	This register manages the GPIO2 clocks.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														IDLEST		RESERVED							OPTFCLKEN_DBCLK	RESERVED						MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:9	RESERVED		R	0x0
8	OPTFCLKEN_DBCLK	Optional functional clock control. 0x0: Optional functional clock is disabled 0x1: Optional functional clock is enabled	RW	0x0
7:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state. 0x2: Reserved 0x3: Reserved	RW	0x0

Table 3-835. Register Call Summary for Register CM_L4PER_GPIO2_CLKCTRL

Clock Management Functional Description
- Clock Domain Module Attributes: [0][1]

PRCM Register Manual
- L4PER_CM_CORE Register Summary: [2]

Table 3-836. CM_L4PER_GPIO3_CLKCTRL

Address Offset	0x0000 0068		
Physical Address	0x4A00 9768	**Instance**	L4PER_CM_CORE
Description	This register manages the GPIO3 clocks.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														IDLEST		RESERVED							OPTFCLKEN_DBCLK	RESERVED						MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:9	RESERVED		R	0x0
8	OPTFCLKEN_DBCLK	Optional functional clock control. 0x0: Optional functional clock is disabled 0x1: Optional functional clock is enabled	RW	0x0
7:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state. 0x2: Reserved 0x3: Reserved	RW	0x0

Table 3-837. Register Call Summary for Register CM_L4PER_GPIO3_CLKCTRL

Clock Management Functional Description
- Clock Domain Module Attributes: [0][1]

PRCM Register Manual
- L4PER_CM_CORE Register Summary: [2]

Table 3-838. CM_L4PER_GPIO4_CLKCTRL

Address Offset	0x0000 0070		
Physical Address	0x4A00 9770	**Instance**	L4PER_CM_CORE
Description	This register manages the GPIO4 clocks.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														IDLEST		RESERVED							OPTFCLKEN_DBCLK	RESERVED						MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:9	RESERVED		R	0x0
8	OPTFCLKEN_DBCLK	Optional functional clock control. 0x0: Optional functional clock is disabled 0x1: Optional functional clock is enabled	RW	0x0
7:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state. 0x2: Reserved 0x3: Reserved	RW	0x0

Table 3-839. Register Call Summary for Register CM_L4PER_GPIO4_CLKCTRL

Clock Management Functional Description • Clock Domain Module Attributes: [0][1]
PRCM Register Manual • L4PER_CM_CORE Register Summary: [2]

Table 3-840. CM_L4PER_GPIO5_CLKCTRL

Address Offset	0x0000 0078		
Physical Address	0x4A00 9778	**Instance**	L4PER_CM_CORE
Description	This register manages the GPIO5 clocks.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														IDLEST		RESERVED							OPTFCLKEN_DBCLK	RESERVED						MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:9	RESERVED		R	0x0
8	OPTFCLKEN_DBCLK	Optional functional clock control. 0x0: Optional functional clock is disabled 0x1: Optional functional clock is enabled	RW	0x0
7:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state. 0x2: Reserved 0x3: Reserved	RW	0x0

Table 3-841. CM_L4PER_GPIO6_CLKCTRL

Address Offset	0x0000 0080		
Physical Address	0x4A00 9780	**Instance**	L4PER_CM_CORE
Description	This register manages the GPIO6 clocks.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														IDLEST		RESERVED							OPTFCLKEN_DBCLK	RESERVED						MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:9	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
8	OPTFCLKEN_DBCLK	Optional functional clock control. 0x0: Optional functional clock is disabled 0x1: Optional functional clock is enabled	RW	0x0
7:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state. 0x2: Reserved 0x3: Reserved	RW	0x0

Table 3-842. CM_L4PER_ESM_CLKCTRL

Address Offset	0x0000 0088		
Physical Address	0x4A00 9788	**Instance**	L4PER_CM_CORE
Description	This register manages the ESM clocks.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														IDLEST		RESERVED								MODULEMODE							

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guaranteed to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-843. Register Call Summary for Register CM_L4PER_ESM_CLKCTRL

Clock Management Functional Description • Clock Domain Module Attributes: [0][1]
PRCM Register Manual • L4PER_CM_CORE Register Summary: [2]

Table 3-844. CM_L4PER2_PWMSS2_CLKCTRL

Address Offset	0x0000 0090		
Physical Address	0x4A00 9790	**Instance**	L4PER_CM_CORE
Description	This register manages the PWMSS1 clocks.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														IDLEST		RESERVED														MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guaranteed to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-845. CM_L4PER2_PWMSS3_CLKCTRL

Address Offset	0x0000 0098		
Physical Address	0x4A00 9798	**Instance**	L4PER_CM_CORE
Description	This register manages the PWMSS2 clocks.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														IDLEST		RESERVED								MODULEMODE							

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guaranteed to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-846. CM_L4PER_I2C1_CLKCTRL

Address Offset	0x0000 00A0		
Physical Address	0x4A00 97A0	**Instance**	L4PER_CM_CORE
Description	This register manages the I2C1 clocks.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														IDLEST		RESERVED								MODULEMODE							

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guaranteed to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-847. Register Call Summary for Register CM_L4PER_I2C1_CLKCTRL

Clock Management Functional Description
- Clock Domain Module Attributes: [0][1]
- Clock Domain Module Attributes: [2]

PRCM Register Manual
- L4PER_CM_CORE Register Summary: [3]

Table 3-848. CM_L4PER_I2C2_CLKCTRL

Address Offset	0x0000 00A8		
Physical Address	0x4A00 97A8	**Instance**	L4PER_CM_CORE
Description	This register manages the I2C2 clocks.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														IDLEST		RESERVED								MODULEMODE							

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guaranteed to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-849. Register Call Summary for Register CM_L4PER_I2C2_CLKCTRL

Clock Management Functional Description
- Clock Domain Module Attributes: [0][1]
- Clock Domain Module Attributes: [2]

PRCM Register Manual
- L4PER_CM_CORE Register Summary: [3]

Table 3-850. CM_L4PER_I2C3_CLKCTRL

Address Offset	0x0000 00B0		
Physical Address	0x4A00 97B0	**Instance**	L4PER_CM_CORE
Description	This register manages the I2C3 clocks.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														IDLEST		RESERVED								MODULEMODE							

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guaranteed to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-851. CM_L4PER_I2C4_CLKCTRL

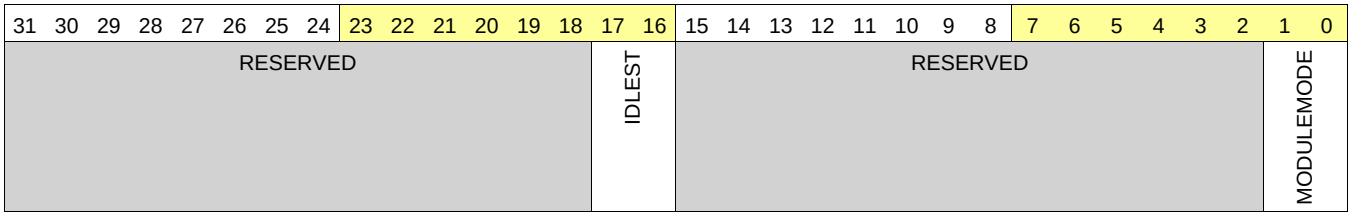
Address Offset	0x0000 00B8		
Physical Address	0x4A00 97B8	**Instance**	L4PER_CM_CORE
Description	This register manages the I2C4 clocks.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														IDLEST		RESERVED								MODULEMODE							

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guaranteed to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-852. CM_L4PER_L4_PER1_CLKCTRL

Address Offset	0x0000 00C0		
Physical Address	0x4A00 97C0	**Instance**	L4PER_CM_CORE
Description	This register manages the L4_PER1 clocks.		
Type	R		



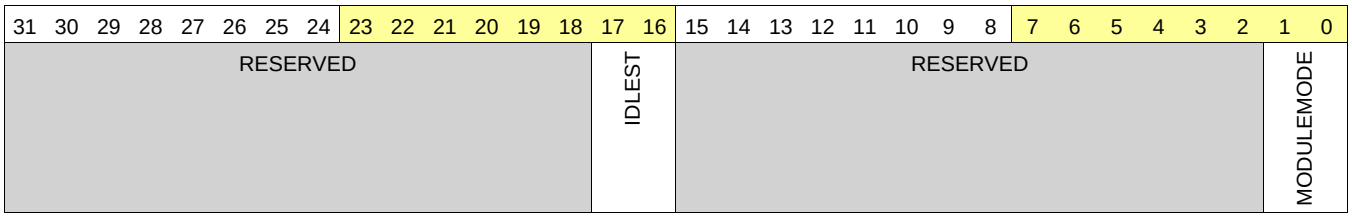
Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state.	R	0x1

Table 3-853. Register Call Summary for Register CM_L4PER_L4_PER1_CLKCTRL

Clock Management Functional Description • Clock Domain Module Attributes: [0][1]
PRCM Register Manual • L4PER_CM_CORE Register Summary: [2]

Table 3-854. CM_L4PER2_PWMSS1_CLKCTRL

Address Offset	0x0000 00C4		
Physical Address	0x4A00 97C4	**Instance**	L4PER_CM_CORE
Description	This register manages the PWMSS1 clocks.		
Type	RW		



Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guaranteed to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-855. Register Call Summary for Register CM_L4PER2_PWMSS1_CLKCTRL

Clock Management Functional Description • Clock Domain Module Attributes: [0][1]
PRCM Register Manual • L4PER_CM_CORE Register Summary: [2]

Table 3-856. CM_L4PER3_DCC1_CLKCTRL

Address Offset	0x0000 00C8		
Physical Address	0x4A00 97C8	**Instance**	L4PER_CM_CORE
Description	This register manages the DCC1 clocks.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				CLKSEL				RESERVED						IDLEST		RESERVED														MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:24	CLKSEL	Select the source of the functional clock 0x0: Selects the TIMER_SYS_CLK as the source 0x1: Selects the SYS_CLK1_32K_CLK as the source 0x2: Selects SYS_CLK2 0x3: Selects XREF_CLK0 0x4: Selects XREF_CLK1 0x5: Selects XREF_CLK2 0x6: Selects XREF_CLK3 0x7: Selects PER_ABE_X1 CLK 0x8: Selects VIDEO1 CLK 0x9: Selects VIDEO2 CLK 0xA: Selects HDMI CLK 0xB: RESERVED 0xC: RESERVED1 0xD: RESERVED2 0xE: RESERVED3 0xF: RESERVED4	RW	0x0
23:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guaranteed to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-857. Register Call Summary for Register CM_L4PER3_DCC1_CLKCTRL

Clock Management Functional Description
- Clock Domain Module Attributes: [0][1]

PRCM Register Manual
- Not Supported Functionality (Registers and Bitfields): [2]
- L4PER_CM_CORE Register Summary: [3]

Table 3-858. CM_L4PER3_DCC2_CLKCTRL

Address Offset	0x0000 00D0		
Physical Address	0x4A00 97D0	**Instance**	L4PER_CM_CORE
Description	This register manages the DCC2 clocks.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				CLKSEL				RESERVED						IDLEST		RESERVED														MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:24	CLKSEL	Select the source of the functional clock 0x0: Selects the TIMER_SYS_CLK as the source 0x1: Selects the SYS_CLK1_32K_CLK as the source 0x2: Selects SYS_CLK2 0x3: Selects XREF_CLK0 0x4: Selects XREF_CLK1 0x5: Selects XREF_CLK2 0x6: Selects XREF_CLK3 0x7: Selects PER_ABE_X1 CLK 0x8: Selects VIDEO1 CLK 0x9: Selects VIDEO2 CLK 0xA: Selects HDMI CLK 0xB: RESERVED 0xC: RESERVED1 0xD: RESERVED2 0xE: RESERVED3 0xF: RESERVED4	RW	0x0
23:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guaranteed to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-859. Register Call Summary for Register CM_L4PER3_DCC2_CLKCTRL

Clock Management Functional Description
- Clock Domain Module Attributes: [0][1]

PRCM Register Manual
- Not Supported Functionality (Registers and Bitfields): [2]
- L4PER_CM_CORE Register Summary: [3]

Table 3-860. CM_L4PER3_DCC3_CLKCTRL

Address Offset	0x0000 00D8		
Physical Address	0x4A00 97D8	**Instance**	L4PER_CM_CORE
Description	This register manages the DCC3 clocks.		
Type	RW		

31 30 29 28	27 26 25 24	23 22 21 20 19 18	17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2	1 0
RESERVED	CLKSEL	RESERVED	IDLEST	RESERVED		MODULEMODE

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:24	CLKSEL	Select the source of the functional clock 0x0: Selects the TIMER_SYS_CLK as the source 0x1: Selects the SYS_CLK1_32K_CLK as the source 0x2: Selects SYS_CLK2 0x3: Selects XREF_CLK0 0x4: Selects XREF_CLK1 0x5: Selects XREF_CLK2 0x6: Selects XREF_CLK3 0x7: Selects PER_ABE_X1 CLK 0x8: Selects VIDEO1 CLK 0x9: Selects VIDEO2 CLK 0xA: Selects HDMI CLK 0xB: RESERVED 0xC: RESERVED1 0xD: RESERVED2 0xE: RESERVED3 0xF: RESERVED4	RW	0x0
23:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guaranteed to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-861. Register Call Summary for Register CM_L4PER3_DCC3_CLKCTRL

Clock Management Functional Description • Clock Domain Module Attributes: [0][1]
PRCM Register Manual • Not Supported Functionality (Registers and Bitfields): [2] • L4PER_CM_CORE Register Summary: [3]

Table 3-862. CM_L4PER_MCSPI1_CLKCTRL

Address Offset	0x0000 00F0		
Physical Address	0x4A00 97F0	**Instance**	L4PER_CM_CORE
Description	This register manages the McSPI1 clocks.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														IDLEST		RESERVED														MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guaranteed to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-863. Register Call Summary for Register CM_L4PER_MCSPI1_CLKCTRL

Clock Management Functional Description • Clock Domain Module Attributes: [0][1]
PRCM Register Manual • L4PER_CM_CORE Register Summary: [2]

Table 3-864. CM_L4PER_MCSPI2_CLKCTRL

Address Offset	0x0000 00F8		
Physical Address	0x4A00 97F8	**Instance**	L4PER_CM_CORE
Description	This register manages the McSPI2 clocks.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														IDLEST		RESERVED								MODULEMODE							

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guaranteed to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

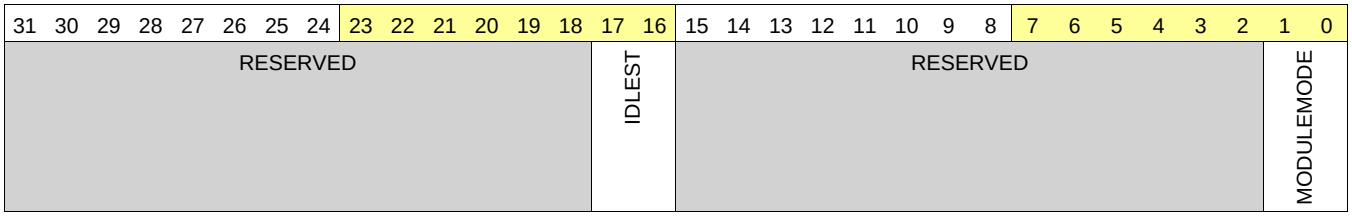
Table 3-865. Register Call Summary for Register CM_L4PER_MCSPI2_CLKCTRL

Clock Management Functional Description
- Clock Domain Module Attributes: [0][1]

PRCM Register Manual
- L4PER_CM_CORE Register Summary: [2]

Table 3-866. CM_L4PER_MCSPI3_CLKCTRL

Address Offset	0x0000 0100		
Physical Address	0x4A00 9800	**Instance**	L4PER_CM_CORE
Description	This register manages the McSPI3 clocks.		
Type	RW		



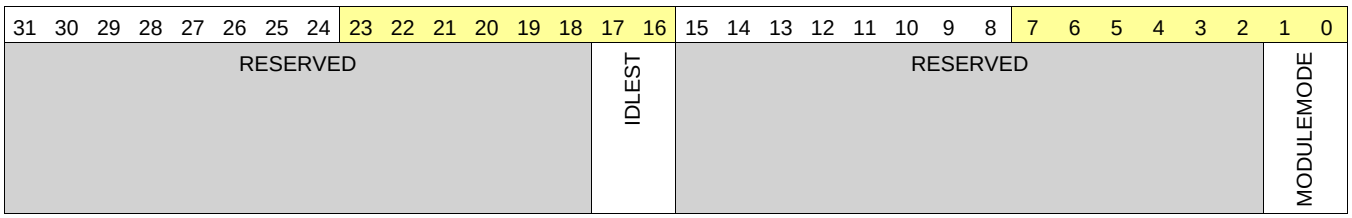
Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guaranteed to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-867. Register Call Summary for Register CM_L4PER_MCSPI3_CLKCTRL

Clock Management Functional Description • Clock Domain Module Attributes: [0][1]
PRCM Register Manual • L4PER_CM_CORE Register Summary: [2]

Table 3-868. CM_L4PER_MCSPI4_CLKCTRL

Address Offset	0x0000 0108		
Physical Address	0x4A00 9808	**Instance**	L4PER_CM_CORE
Description	This register manages the McSPI4 clocks.		
Type	RW		



Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guaranteed to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-869. Register Call Summary for Register CM_L4PER_MCSPI4_CLKCTRL

Clock Management Functional Description • Clock Domain Module Attributes: [0][1]
PRCM Register Manual • L4PER_CM_CORE Register Summary: [2]

Table 3-870. CM_L4PER_GPIO7_CLKCTRL

Address Offset	0x0000 0110		
Physical Address	0x4A00 9810	**Instance**	L4PER_CM_CORE
Description	This register manages the GPIO7 clocks.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														IDLEST		RESERVED							OPTFCLKEN_DBCLK	RESERVED						MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3

Bits	Field Name	Description	Type	Reset
15:9	RESERVED		R	0x0
8	OPTFCLKEN_DBCLK	Optional functional clock control. 0x0: Optional functional clock is disabled 0x1: Optional functional clock is enabled	RW	0x0
7:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state. 0x2: Reserved 0x3: Reserved	RW	0x0

Table 3-871. CM_L4PER_GPIO8_CLKCTRL

Address Offset	0x0000 0118		
Physical Address	0x4A00 9818	**Instance**	L4PER_CM_CORE
Description	This register manages the GPIO8 clocks.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														IDLEST		RESERVED							OPTFCLKEN_DBCLK	RESERVED						MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:9	RESERVED		R	0x0
8	OPTFCLKEN_DBCLK	Optional functional clock control. 0x0: Optional functional clock is disabled 0x1: Optional functional clock is enabled	RW	0x0
7:2	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state. 0x2: Reserved 0x3: Reserved	RW	0x0

Table 3-872. CM_L4PER_MMC3_CLKCTRL

Address Offset	0x0000 0120		
Physical Address	0x4A00 9820	**Instance**	L4PER_CM_CORE
Description	This register manages the MMC3 clocks.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED					CLKSEL_DIV		CLKSEL_MUX	RESERVED						IDLEST		RESERVED							OPTFCLKEN_CLK32K	RESERVED						MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:25	CLKSEL_DIV	Selects the divider value 0x0: Select MMC CLK divided by 1 0x1: Select MMC CLK divided by 2 0x2: Selects MMC CLK divided by 4 0x3: RESERVED	RW	0x0
24	CLKSEL_MUX	Select the clock for the MMC from DPLL_PER. 0x0: Selects FUNC_48M_CLK 0x1: Selects FUNC_192M_CLK	RW	0x0
23:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:9	RESERVED		R	0x0
8	OPTFCLKEN_CLK32K	MMC optional clock control: 32K CLK 0x0: Optional functional clock is disabled 0x1: Optional functional clock is enabled	RW	0x0
7:2	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guaranteed to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-873. CM_L4PER_MMC4_CLKCTRL

Address Offset	0x0000 0128		
Physical Address	0x4A00 9828	**Instance**	L4PER_CM_CORE
Description	This register manages the MMC4 clocks.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED					CLKSEL_DIV		CLKSEL_MUX	RESERVED						IDLEST		RESERVED							OPTFCLKEN_CLK32K	RESERVED						MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:25	CLKSEL_DIV	Selects the divider value 0x0: Selects MMC CLK divided by 1 0x1: Selects MMC CLK divided by 2 0x2: Selects MMC CLK divided by 4 0x3: RESERVED	RW	0x0
24	CLKSEL_MUX	Select the clock for the MMC from DPLL_PER. 0x0: Selects FUNC_48M_CLK 0x1: Selects FUNC_192M_CLK	RW	0x0
23:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:9	RESERVED		R	0x0
8	OPTFCLKEN_CLK32K	MMC optional clock control: 32K CLK 0x0: Optional functional clock is disabled 0x1: Optional functional clock is enabled	RW	0x0
7:2	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guaranteed to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-874. Register Call Summary for Register CM_L4PER_MMC4_CLKCTRL

Clock Management Functional Description
- CM_CORE_AON Clock Generator: [0][1]
- Clock Domain Module Attributes: [2][3]

PRCM Register Manual
- L4PER_CM_CORE Register Summary: [4]

Table 3-875. CM_L4PER3_DCC4_CLKCTRL

Address Offset	0x0000 0130		
Physical Address	0x4A00 9830	**Instance**	L4PER_CM_CORE
Description	This register manages the DCC4 clocks.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				CLKSEL				RESERVED						IDLEST		RESERVED														MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:24	CLKSEL	Select the source of the functional clock 0x0: Selects the TIMER_SYS_CLK as the source 0x1: Selects the SYS_CLK1_32K_CLK as the source 0x2: Selects SYS_CLK2 0x3: Selects XREF_CLK0 0x4: Selects XREF_CLK1 0x5: Selects XREF_CLK2 0x6: Selects XREF_CLK3 0x7: Selects PER_ABE_X1 CLK 0x8: Selects VIDEO1 CLK 0x9: Selects VIDEO2 CLK 0xA: Selects HDMI CLK 0xB: RESERVED 0xC: RESERVED1 0xD: RESERVED2 0xE: RESERVED3 0xF: RESERVED4	RW	0x0
23:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guaranteed to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-876. Register Call Summary for Register CM_L4PER3_DCC4_CLKCTRL

Clock Management Functional Description
- Clock Domain Module Attributes: [0][1]

PRCM Register Manual
- Not Supported Functionality (Registers and Bitfields): [2]
- L4PER_CM_CORE Register Summary: [3]

Table 3-877. CM_L4PER2_QSPI_CLKCTRL

Address Offset	0x0000 0138		
Physical Address	0x4A00 9838	**Instance**	L4PER_CM_CORE
Description	This register manages the QSPI clocks.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED					CLKSEL_DIV		CLKSEL_SOURCE	RESERVED						IDLEST		RESERVED														MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:25	CLKSEL_DIV	QSPI clock divide ratio. 0x0: QSPI clock is divided by 1. 0x1: QSPI clock is divided by 2. 0x2: QSPI clock is divided by 4. 0x3: RESERVED	RW	0x0
24	CLKSEL_SOURCE	Selects the source of the functional clock. 0x0: FUNC_128M_CLK from DPLL_PER is selected 0x1: Selects PER_QSPI_CLK from DPLL_PER	RW	0x0
23:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED	Reserved	R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guaranteed to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-878. Register Call Summary for Register CM_L4PER2_QSPI_CLKCTRL

Clock Management Functional Description
- CM_CORE_AON Clock Generator: [0][1]
- Clock Domain Module Attributes: [2][3]

PRCM Register Manual
- L4PER_CM_CORE Register Summary: [4]

Table 3-879. CM_L4PER_UART1_CLKCTRL

Address Offset	0x0000 0140		
Physical Address	0x4A00 9840	**Instance**	L4PER_CM_CORE
Description	This register manages the UART1 clocks.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED							CLKSEL	RESERVED						IDLEST		RESERVED														MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24	CLKSEL	Selects functional clock for UART between FUNC_48M_CLK and FUNC_192M_CLK 0x0: Selects FUNC_48M_CLK 0x1: Selects FUNC_192M_CLK	RW	0x0
23:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guaranteed to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-880. Register Call Summary for Register CM_L4PER_UART1_CLKCTRL

Clock Management Functional Description • CM_CORE_AON Clock Generator: [0] • Clock Domain Module Attributes: [1][2]
PRCM Register Manual • L4PER_CM_CORE Register Summary: [3]

Table 3-881. CM_L4PER_UART2_CLKCTRL

Address Offset	0x0000 0148		
Physical Address	0x4A00 9848	**Instance**	L4PER_CM_CORE
Description	This register manages the UART2 clocks.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED							CLKSEL	RESERVED						IDLEST		RESERVED														MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24	CLKSEL	Selects functional clock for UART between FUNC_48M_CLK and FUNC_192M_CLK 0x0: Selects FUNC_48M_CLK 0x1: Selects FUNC_192M_CLK	RW	0x0
23:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guaranteed to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-882. Register Call Summary for Register CM_L4PER_UART2_CLKCTRL

Clock Management Functional Description • CM_CORE_AON Clock Generator: [0] • Clock Domain Module Attributes: [1][2]
PRCM Register Manual • L4PER_CM_CORE Register Summary: [3]

Table 3-883. CM_L4PER_UART3_CLKCTRL

Address Offset	0x0000 0150		
Physical Address	0x4A00 9850	**Instance**	L4PER_CM_CORE
Description	This register manages the UART3 clocks.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED							CLKSEL	RESERVED						IDLEST		RESERVED														MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24	CLKSEL	Selects functional clock for UART between FUNC_48M_CLK and FUNC_192M_CLK 0x0: Selects FUNC_48M_CLK 0x1: Selects FUNC_192M_CLK	RW	0x0
23:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guaranteed to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-884. Register Call Summary for Register CM_L4PER_UART3_CLKCTRL

Clock Management Functional Description
- CM_CORE_AON Clock Generator: [0]
- Clock Domain Module Attributes: [1][2]

PRCM Register Manual
- L4PER_CM_CORE Register Summary: [3]

Table 3-885. CM_L4PER_UART4_CLKCTRL

Address Offset	0x0000 0158		
Physical Address	0x4A00 9858	**Instance**	L4PER_CM_CORE
Description	This register manages the UART4 clocks.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED							CLKSEL	RESERVED						IDLEST		RESERVED														MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24	CLKSEL	Selects functional clock for UART between FUNCTION_48M_CLK and FUNC_192M_CLK 0x0: Selects FUNC_48M_CLK 0x1: Selects FUNC_192M_CLK	RW	0x0
23:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guaranteed to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-886. Register Call Summary for Register CM_L4PER_UART4_CLKCTRL

PRCM Register Manual
- L4PER_CM_CORE Register Summary: [0]

Table 3-887. CM_L4PER2_ADC_CLKCTRL

Address Offset	0x0000 0160		
Physical Address	0x4A00 9860	**Instance**	L4PER_CM_CORE
Description	This register manages the ADC clocks. In TDA2x, this register is CM_L4PER2_MCASP2_CLKCTRL		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CLKSEL_AHCLKR				CLKSEL_AHCLKX				CLKSEL_AUX_CLK		RESERVED				IDLEST		RESERVED														MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:28	CLKSEL_AHCLKR	Selects reference clock for AHCLKR 0x0: Selects ABE_24M_GFCLK 0x1: Selects ABE_SYS_CLK 0x2: Selects FUNC_24M_GFCLK 0x3: Selects ATL CLK3 0x4: Selects ATL CLK2 0x5: Selects ATL CLK1 0x6: Selects ATL CLK0 0x7: Selects SYS_CLK2 0x8: Selects XREF_CLK0 0x9: Selects XREF_CLK1 0xA: Selects XREF_CLK2 0xB: Selects XREF_CLK3 0xC: Selects MLB_CLK 0xD: Selects MLBP_CLK 0xE: RESERVED 0xF: RESERVED	RW	0x0
27:24	CLKSEL_AHCLKX	Selects reference clock for AHCLKX 0x0: Selects ABE_24M_GFCLK 0x1: Selects ABE_SYS_CLK 0x2: Selects FUNC_24M_GFCLK 0x3: Selects ATL CLK3 0x4: Selects ATL CLK2 0x5: Selects ATL CLK1 0x6: Selects ATL CLK0 0x7: Selects SYS_CLK2 0x8: Selects XREF_CLK0 0x9: Selects XREF_CLK1 0xA: Selects XREF_CLK2 0xB: Selects XREF_CLK3 0xC: Selects MLB_CLK 0xD: Selects MLBP_CLK 0xE: RESERVED 0xF: RESERVED	RW	0x0
23:22	CLKSEL_AUX_CLK	Selects the source of the AUX clock 0x0: Selects PER_ABE_X1_GFCLK 0x1: Selects VIDEO1 CLK 0x2: Selects VIDEO2 CLK 0x3: Selects HDMI CLK	RW	0x0
21:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guaranteed to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-888. Register Call Summary for Register CM_L4PER2_ADC_CLKCTRL

Clock Management Functional Description • Clock Domain Module Attributes: [0][1]
PRCM Register Manual • Not Supported Functionality (Registers and Bitfields): [2][3][4] • L4PER_CM_CORE Register Summary: [5]

Table 3-889. CM_L4PER2_ATL_CLKCTRL

Address Offset	0x0000 0168		
Physical Address	0x4A00 9868	**Instance**	L4PER_CM_CORE
Description	This register manages the ATL module mode.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				CLKSEL_AHCLKX				CLKSEL_AUX_CLK		RESERVED				IDLEST		RESERVED														MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:24	CLKSEL_AHCLKX	Selects reference clock for AHCLKX 0x0: Selects ABE_24M_GFCLK 0x1: Selects ABE_SYS_CLK 0x2: Selects FUNC_24M_GFCLK 0x3: Selects ATL CLK3 0x4: Selects ATL CLK2 0x5: Selects ATL CLK1 0x6: Selects ATL CLK0 0x7: Selects SYS_CLK2 0x8: Selects XREF_CLK0 0x9: Selects XREF_CLK1 0xA: Selects XREF_CLK2 0xB: Selects XREF_CLK3 0xC: Selects MLB_CLK 0xD: Selects MLBP_CLK 0xE: RESERVED 0xF: RESERVED	RW	0x0
23:22	CLKSEL_AUX_CLK	Selects the source of the AUX clock 0x0: Selects PER_ABE_X1_GFCLK 0x1: Selects VIDEO1 CLK 0x2: Selects VIDEO2 CLK 0x3: Selects HDMI CLK	RW	0x0
21:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guaranteed to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-890. Register Call Summary for Register CM_L4PER2_ATL_CLKCTRL

Clock Management Functional Description • Clock Domain Module Attributes: [0][1]
PRCM Register Manual • L4PER_CM_CORE Register Summary: [2]

Table 3-891. CM_L4PER_UART5_CLKCTRL

Address Offset	0x0000 0170		
Physical Address	0x4A00 9870	**Instance**	L4PER_CM_CORE
Description	This register manages the UART5 clocks.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED							CLKSEL	RESERVED						IDLEST		RESERVED														MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24	CLKSEL	Selects functional clock for UART between FUNCTION_48M_CLK and FUNC_192M_CLK 0x0: Selects FUNC_48M_CLK 0x1: Selects FUNC_192M_CLK	RW	0x0
23:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guaranteed to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-892. CM_L4PER2_MCASP5_CLKCTRL

Address Offset	0x0000 0178		
Physical Address	0x4A00 9878	**Instance**	L4PER_CM_CORE
Description	This register manages the McASP3 clocks.		
Type	RW		

31 30 29 28	27 26 25 24	23 22	21 20 19 18	17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2	1 0
RESERVED	CLKSEL_AHCLKX	CLKSEL_AUX_CLK	RESERVED	IDLEST	RESERVED		MODULEMODE

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:24	CLKSEL_AHCLKX	Selects reference clock for AHCLKX 0x0: Selects ABE_24M_GFCLK 0x1: Selects ABE_SYS_CLK 0x2: Selects FUNC_24M_GFCLK 0x3: Selects ATL_CLK3 0x4: Selects ATL_CLK2 0x5: Selects ATL_CLK1 0x6: Selects ATL_CLK0 0x7: Selects SYS_CLK2 0x8: Selects XREF_CLK0 0x9: Selects XREF_CLK1 0xA: Selects XREF_CLK2 0xB: Selects XREF_CLK1 0xC: Selects divided version of SYS_CLK1. See CM_CLKSEL_MLB_MCASP 0xD: Selects divided version of SYS_CLK1. See CM_CLKSEL_MLBP_MCASP 0xE: RESERVED 0xF: RESERVED	RW	0x0
23:22	CLKSEL_AUX_CLK	Selects the source of the AUX clock. 0x0: Selects divided version of SYS_CLK1. See CM_CLKSEL_PER_ABE_X1_GFCLK_MCASP_AUX 0x1: Selects divided version of SYS_CLK1. See CM_CLKSEL_VIDEO1_MCASP_AUX 0x2: Selects divided version of SYS_CLK1. See CM_CLKSEL_VIDEO2_MCASP_AUX 0x3: Selects divided version of L4_ICLK. See CM_CLKSEL_HDMI_MCASP_AUX	RW	0x0
21:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guaranteed to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-893. Register Call Summary for Register CM_L4PER2_MCASP5_CLKCTRL

Clock Management Functional Description
- CM_CORE_AON_MCASP1 Overview: [0][1]
- Clock Domain Module Attributes: [2][3]

PRCM Register Manual
- L4PER_CM_CORE Register Summary: [4]

Table 3-894. CM_L4SEC_CLKSTCTRL

Address Offset	0x0000 0180		
Physical Address	0x4A00 9880	**Instance**	L4PER_CM_CORE
Description	This register enables the domain power state transition. It controls the HW supervised domain power state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							CLKACTIVITY_L4SEC_L3_GICLK	RESERVED						CLKTRCTRL	

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	CLKACTIVITY_L4SEC_L3_GICLK	This field indicates the state of the L3_SECURE_GICLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
7:2	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
1:0	CLKTRCTRL	Controls the clock state transition of the L4PER clock domain. 0x0: NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may however occur. 0x1: SW_SLEEP: Start a software forced sleep transition on the domain. 0x2: SW_WKUP: Start a software forced wake-up transition on the domain. 0x3: HW_AUTO: Automatic transition is enabled. Sleep and wakeup transition are based upon hardware conditions.	RW	0x0

Table 3-895. CM_L4SEC_STATICDEP

Address Offset	0x0000 0184		
Physical Address	0x4A00 9884	**Instance**	L4PER_CM_CORE
Description	This register controls the static domain depedencies from L4SEC domain towards 'target' domains. It is relevant only for domain having system initiator(s).		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																		L4PER_STATDEP	RESERVED							L3MAIN1_STATDEP	EMIF_STATDEP	RESERVED			

Bits	Field Name	Description	Type	Reset
31:14	RESERVED		R	0x0
13	L4PER_STATDEP	Static dependency towards L4PER1 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
12:6	RESERVED		R	0x0
5	L3MAIN1_STATDEP	Static dependency towards L3MAIN1 clock domain 0x1: Dependency is enabled	R	0x1
4	EMIF_STATDEP	Static dependency towards EMIF clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3:0	RESERVED		R	0x0

Table 3-896. CM_L4SEC_DYNAMICDEP

Address Offset	0x0000 0188		
Physical Address	0x4A00 9888	**Instance**	L4PER_CM_CORE
Description	This register controls the dynamic domain depedencies from L4SEC domain towards 'target' domains. It is relevant only for domain having OCP master port(s).		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								RESERVED		L3MAIN1_DYNDEP	RESERVED				

Bits	Field Name	Description	Type	Reset
31:6	RESERVED		R	0x0
5	L3MAIN1_DYNDEP	Dynamic dependency towards L3MAIN1 clock domain 0x0: Dependency is disabled	R	0x0
4:0	RESERVED		R	0x0

Table 3-897. CM_L4PER2_MCASP8_CLKCTRL

Address Offset	0x0000 0190		
Physical Address	0x4A00 9890	**Instance**	L4PER_CM_CORE
Description	This register manages the McASP8 clocks.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				CLKSEL_AHCLKX				CLKSEL_AUX_CLK		RESERVED				IDLEST		RESERVED														MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:24	CLKSEL_AHCLKX	Selects reference clock for AHCLKX 0x0: Selects ABE_24M_GFCLK 0x1: Selects ABE_SYS_CLK 0x2: Selects FUNC_24M_GFCLK 0x3: Selects CRC CLK3 0x4: Selects CRC CLK2 0x5: Selects CRC CLK1 0x6: Selects CRC CLK0 0x7: Selects SYS_CLK2 0x8: Selects XREF_CLK0 0x9: Selects XREF_CLK1 0xA: Selects XREF_CLK2 0xB: Selects XREF_CLK3 0xC: Selects MLB_CLK 0xD: Selects MLBP_CLK 0xE: RESERVED 0xF: RESERVED	RW	0x0

Bits	Field Name	Description	Type	Reset
23:22	CLKSEL_AUX_CLK	Selects the source of the AUX clock 0x0: Selects PER_ABE_X1_GFCLK 0x1: Selects VIDEO1 CLK 0x2: Selects VIDEO2 CLK 0x3: Selects HDMI CLK	RW	0x0
21:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guaranteed to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-898. CM_L4PER2_MCASP4_CLKCTRL

Address Offset	0x0000 0198		
Physical Address	0x4A00 9898	**Instance**	L4PER_CM_CORE
Description	This register manages the McASP2 clocks.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				CLKSEL_AHCLKX				CLKSEL_AUX_CLK		RESERVED				IDLEST		RESERVED														MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:24	CLKSEL_AHCLKX	Selects reference clock for AHCLKX 0x0: Selects ABE_24M_GFCLK 0x1: Selects ABE_SYS_CLK 0x2: Selects FUNC_24M_GFCLK 0x3: Selects ATL_CLK3 0x4: Selects ATL_CLK2 0x5: Selects ATL_CLK1 0x6: Selects ATL_CLK0 0x7: Selects SYS_CLK2 0x8: Selects XREF_CLK0 0x9: Selects XREF_CLK1 0xA: Selects XREF_CLK2 0xB: Selects XREF_CLK1 0xC: Selects divided version of SYS_CLK1. See CM_CLKSEL_MLB_MCASP 0xD: Selects divided version of SYS_CLK1. See CM_CLKSEL_MLBP_MCASP 0xE: RESERVED 0xF: RESERVED	RW	0x0
23:22	CLKSEL_AUX_CLK	Selects the source of the AUX clock. 0x0: Selects divided version of SYS_CLK1. See CM_CLKSEL_PER_ABE_X1_GFCLK_MCASP_AUX 0x1: Selects divided version of SYS_CLK1. See CM_CLKSEL_VIDEO1_MCASP_AUX 0x2: Selects divided version of SYS_CLK1. See CM_CLKSEL_VIDEO2_MCASP_AUX 0x3: Selects divided version of L4_ICLK. See CM_CLKSEL_HDMI_MCASP_AUX	RW	0x0
21:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guaranteed to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-899. Register Call Summary for Register CM_L4PER2_MCASP4_CLKCTRL

Clock Management Functional Description
- CM_CORE_AON_MCASP1 Overview: [0][1]
- Clock Domain Module Attributes: [2][3]

PRCM Register Manual
- L4PER_CM_CORE Register Summary: [4]

Table 3-900. CM_L4SEC_AES1_CLKCTRL

Address Offset	0x0000 01A0		
Physical Address	0x4A00 98A0	**Instance**	L4PER_CM_CORE
Description	This register manages the AES1 clocks.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														IDLEST		RESERVED								MODULEMODE							

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state. 0x2: Reserved 0x3: Reserved	RW	0x0

Table 3-901. CM_L4SEC_AES2_CLKCTRL

Address Offset	0x0000 01A8		
Physical Address	0x4A00 98A8	**Instance**	L4PER_CM_CORE
Description	This register manages the AES2 clocks.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														IDLEST		RESERVED														MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state. 0x2: Reserved 0x3: Reserved	RW	0x0

Table 3-902. CM_L4SEC_DES3DES_CLKCTRL

Address Offset	0x0000 01B0		
Physical Address	0x4A00 98B0	**Instance**	L4PER_CM_CORE
Description	This register manages the DES3DES clocks.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														IDLEST		RESERVED														MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state. 0x2: Reserved 0x3: Reserved	RW	0x0

Table 3-903. CM_L4SEC_FPKA_CLKCTRL

Address Offset	0x0000 01B8		
Physical Address	0x4A00 98B8	**Instance**	L4PER_CM_CORE
Description	This register manages the FPKA clocks.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														IDLEST		RESERVED								MODULEMODE							

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guaranteed to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-904. CM_L4SEC_RNG_CLKCTRL

Address Offset	0x0000 01C0		
Physical Address	0x4A00 98C0	**Instance**	L4PER_CM_CORE
Description	This register manages the RNG clocks.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														IDLEST		RESERVED														MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state. 0x2: Reserved 0x3: Reserved	RW	0x0

Table 3-905. CM_L4SEC_SHA2MD51_CLKCTRL

Address Offset	0x0000 01C8		
Physical Address	0x4A00 98C8	**Instance**	L4PER_CM_CORE
Description	This register manages the SHA2MD51 clocks.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														IDLEST		RESERVED														MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state. 0x2: Reserved 0x3: Reserved	RW	0x0

Table 3-906. CM_L4PER2_UART7_CLKCTRL

Address Offset	0x0000 01D0		
Physical Address	0x4A00 98D0	**Instance**	L4PER_CM_CORE
Description	This register manages the UART7 clocks.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED							CLKSEL	RESERVED						IDLEST		RESERVED														MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24	CLKSEL	Selects functional clock for UART between FUNCTION_48M_CLK and FUNC_192M_CLK 0x0: Selects FUNC_48M_CLK 0x1: Selects FUNC_192M_CLK	RW	0x0
23:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guaranteed to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-907. CM_L4SEC_DMA_CRYPTO_CLKCTRL

Address Offset	0x0000 01D8		
Physical Address	0x4A00 98D8	**Instance**	L4PER_CM_CORE
Description	This register manages the DMA_CRYPTO clocks.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED													STBYST	IDLEST		RESERVED														MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:19	RESERVED		R	0x0
18	STBYST	Module standby status. [warm reset insensitive] 0x0: Module is functional (not in standby) 0x1: Module is in standby	R	0x1
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state.	R	0x1

Table 3-908. CM_L4PER2_UART8_CLKCTRL

Address Offset	0x0000 01E0		
Physical Address	0x4A00 98E0	**Instance**	L4PER_CM_CORE
Description	This register manages the UART8 clocks.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED							CLKSEL	RESERVED						IDLEST		RESERVED														MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24	CLKSEL	Selects functional clock for UART between FUNCTION_48M_CLK and FUNC_192M_CLK 0x0: Selects FUNC_48M_CLK 0x1: Selects FUNC_192M_CLK	RW	0x0
23:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guaranteed to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-909. CM_L4PER2_UART9_CLKCTRL

Address Offset	0x0000 01E8		
Physical Address	0x4A00 98E8	**Instance**	L4PER_CM_CORE
Description	This register manages the UART9 clocks.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED							CLKSEL	RESERVED						IDLEST		RESERVED															MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24	CLKSEL	Selects functional clock for UART between FUNCTION_48M_CLK and FUNC_192M_CLK 0x0: Selects FUNC_48M_CLK 0x1: Selects FUNC_192M_CLK	RW	0x0
23:18	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guaranteed to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-910. CM_L4PER2_DCAN2_CLKCTRL

Address Offset	0x0000 01F0		
Physical Address	0x4A00 98F0	**Instance**	L4PER_CM_CORE
Description	This register manages the MCAN clocks.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														IDLEST		RESERVED								MODULEMODE							

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guaranteed to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-911. Register Call Summary for Register CM_L4PER2_DCAN2_CLKCTRL

Clock Management Functional Description • Clock Domain Module Attributes: [0][1]
PRCM Register Manual • L4PER_CM_CORE Register Summary: [2]

Table 3-912. CM_L4SEC_SHA2MD52_CLKCTRL

Address Offset	0x0000 01F8		
Physical Address	0x4A00 98F8	**Instance**	L4PER_CM_CORE
Description	This register manages the SHA2MD52 clocks.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														IDLEST		RESERVED								MODULEMODE							

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state. 0x2: Reserved 0x3: Reserved	RW	0x0

Table 3-913. CM_L4PER2_CLKSTCTRL

Address Offset	0x0000 01FC		
Physical Address	0x4A00 98FC	**Instance**	L4PER_CM_CORE
Description	This register enables the domain power state transition. It controls the HW supervised domain power state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CLKACTIVITY_MCASP8_AUX_GFCLK	CLKACTIVITY_MCASP8_AHCLKX	CLKACTIVITY_MCASP7_AUX_GFCLK	CLKACTIVITY_MCASP7_AHCLKX	CLKACTIVITY_MCASP6_AUX_GFCLK	CLKACTIVITY_MCASP6_AHCLKX	CLKACTIVITY_MCASP5_AHCLKX	CLKACTIVITY_MCASP5_AUX_GFCLK	CLKACTIVITY_MCASP4_AUX_GFCLK	CLKACTIVITY_MCASP4_AHCLKX	CLKACTIVITY_MCASP3_AUX_GFCLK	CLKACTIVITY_MCASP3_AHCLKX	CLKACTIVITY_ADC_AUX_GFCLK	CLKACTIVITY_ADC_AHCLKR	CLKACTIVITY_ADC_AHCLKX	CLKACTIVITY_L4PER2_L3_GICLK	CLKACTIVITY_DCAN2_SYS_CLK	CLKACTIVITY_ICSS_IEP_CLK	CLKACTIVITY_PER_192M_GFCLK	CLKACTIVITY_QSPI_GFCLK	CLKACTIVITY_UART9_GFCLK	CLKACTIVITY_UART8_GFCLK	CLKACTIVITY_UART7_GFCLK	CLKACTIVITY_ICSS_CLK	RESERVED	RESERVED			RESERVED		CLKTRCTRL	

Bits	Field Name	Description	Type	Reset
31	CLKACTIVITY_MCASP8_AUX_GFCLK	This field indicates the state of the MCASP8_AUX_GFCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
30	CLKACTIVITY_MCASP8_AHCLKX	This field indicates the state of the MCASP8_AHCLKX clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
29	CLKACTIVITY_MCASP7_AUX_GFCLK	This field indicates the state of the MCASP7_AUX_GFCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
28	CLKACTIVITY_MCASP7_AHCLK	This field indicates the state of the MCASP7_AHCLKX clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
27	CLKACTIVITY_MCASP6_AUX_GFCLK	This field indicates the state of the MCASP6_AUX_GFCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0

Bits	Field Name	Description	Type	Reset
26	CLKACTIVITY_MCASP6_AHCLKX	This field indicates the state of the MCASP6_AUX_GFCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
25	CLKACTIVITY_MCASP5_AHCLKX	This field indicates the state of the MCASP5_AHCLKX clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
24	CLKACTIVITY_MCASP5_AUX_GFCLK	This field indicates the state of the MCASP5_AUX_GFCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
23	CLKACTIVITY_MCASP4_AUX_GFCLK	This field indicates the state of the MCASP4_AUX_GFCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
22	CLKACTIVITY_MCASP4_AHCLKX	This field indicates the state of the MCASP4_AHCLKX clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
21	CLKACTIVITY_MCASP3_AUX_GFCLK	This field indicates the state of the MCASP3_AUX_GFCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
20	CLKACTIVITY_MCASP3_AHCLKX	This field indicates the state of the MCASP3_AHCLKX clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
19	CLKACTIVITY_MCASP2_AUX_GFCLK	This field indicates the state of the ADC_AUX_GFCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
18	CLKACTIVITY_MCASP2_AHCLKR	This field indicates the state of the ADC_AHCLKR clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
17	CLKACTIVITY_MCASP2_AHCLKX	This field indicates the state of the ADC_AHCLKX clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0

Bits	Field Name	Description	Type	Reset
16	CLKACTIVITY_L4PER2_L3_GICLK	This field indicates the state of the L4PER2_L3_GICLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
15	CLKACTIVITY_DCAN2_SYS_CLK	This field indicates the state of the DCAN2_SYS_CLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
14	CLKACTIVITY_ICSS_IEP_CLK	This field indicates the state of the ICSS_IEP_CLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
13	CLKACTIVITY_PER_192M_GFCLK	This field indicates the state of the PER_192M_GFCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
12	CLKACTIVITY_QSPI_GFCLK	This field indicates the state of the QSPI_GFCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
11	CLKACTIVITY_UART9_GFCLK	This field indicates the state of the UART9_GFCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
10	CLKACTIVITY_UART8_GFCLK	This field indicates the state of the UART8_GFCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
9	CLKACTIVITY_UART7_GFCLK	This field indicates the state of the UART7_GFCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
8	CLKACTIVITY_ICSS_CLK	This field indicates the state of the ICSS_CLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
7:2	RESERVED		R	0x0
1:0	CLKTRCTRL	Controls the clock state transition of the L4PER clock domain. 0x0: NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may however occur. 0x1: SW_SLEEP: Start a software forced sleep transition on the domain. 0x2: SW_WKUP: Start a software forced wake-up transition on the domain. 0x3: HW_AUTO: Automatic transition is enabled. Sleep and wakeup transition are based upon hardware conditions.	RW	0x0

Table 3-914. Register Call Summary for Register CM_L4PER2_CLKSTCTRL

Clock Management Functional Description
• Clock Domain Modes: [0][1][3][4][5][6][7][8][9][10]
PRCM Register Manual
• Not Supported Functionality (Registers and Bitfields): [11][12][13][14][15][16][17][18][19][20][21][22][23][24]
• L4PER_CM_CORE Register Summary: [25]

Table 3-915. CM_L4PER2_DYNAMICDEP

Address Offset	0x0000 0200		
Physical Address	0x4A00 9900	**Instance**	L4PER_CM_CORE
Description	This register controls the dynamic domain depedencies from L4PER2 domain towards 'target' domains. It is relevant only for domain having OCP master port(s).		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				WINDOWSIZE				RESERVED	GMAC_DYNDEP	RESERVED									L4CFG_DYNDEP	RESERVED				L3INIT_DYNDEP	CRC_DYNDEP	RESERVED		IPU_DYNDEP	RESERVED		

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:24	WINDOWSIZE	Size of sliding window used to monitor OCP interface activity for determination of auto-sleep feature. Time unit defined by CM_DYN_DEP_PRESCAL register.	RW	0x4
23	RESERVED		R	0x0
22	GMAC_DYNDEP	Dynamic dependency towards GMAC clock domain 0x1: Dependency is enabled	R	0x1
21:13	RESERVED		R	0x0
12	L4CFG_DYNDEP	Dynamic dependency towards L4CFG clock domain 0x1: Dependency is enabled	R	0x1
11:8	RESERVED		R	0x0
7	L3INIT_DYNDEP	Dynamic dependency towards L3INIT clock domain 0x1: Dependency is enabled	R	0x1
6	CRC_DYNDEP	Dynamic dependency towards CRC clock domain 0x1: Dependency is enabled	R	0x1
5:4	RESERVED		R	0x0
3	IPU_DYNDEP	Dynamic dependency towards IPU clock domain 0x1: Dependency is enabled	R	0x1
2:0	RESERVED		R	0x0

Table 3-916. Register Call Summary for Register CM_L4PER2_DYNAMICDEP

Clock Management Functional Description
• Clock Domain Dependency: [0][1][2][3][4]
PRCM Register Manual
• L4PER_CM_CORE Register Summary: [5]

Table 3-917. CM_L4PER2_MCASP6_CLKCTRL

Address Offset	0x0000 0204		
Physical Address	0x4A00 9904	**Instance**	L4PER_CM_CORE
Description	This register manages the McASP2 clocks.		
Type	RW		

31 30 29 28	27 26 25 24	23 22	21 20 19 18	17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2	1 0
RESERVED	CLKSEL_AHCLKX	CLKSEL_AUX_CLK	RESERVED	IDLEST	RESERVED		MODULEMODE

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:24	CLKSEL_AHCLKR	Selects reference clock for AHCLKR 0x0: Selects ABE_24M_GFCLK 0x1: Selects ABE_SYS_CLK 0x2: Selects FUNC_24M_GFCLK 0x3: Selects ATL_CLK3 0x4: Selects ATL_CLK2 0x5: Selects ATL_CLK1 0x6: Selects ATL_CLK0 0x7: Selects SYS_CLK2 0x8: Selects XREF_CLK0 0x9: Selects XREF_CLK1 0xA: Selects XREF_CLK2 0xB: Selects XREF_CLK1 0xC: Selects divided version of SYS_CLK1. See CM_CLKSEL_MLB_MCASP 0xD: Selects divided version of SYS_CLK1. See CM_CLKSEL_MLBP_MCASP 0xE: RESERVED 0xF: RESERVED	RW	0x0
23:22	CLKSEL_AUX_CLK	Selects the source of the AUX clock. 0x0: Selects divided version of SYS_CLK1. See CM_CLKSEL_PER_ABE_X1_GFCLK_MCASP_AUX 0x1: Selects divided version of SYS_CLK1. See CM_CLKSEL_VIDEO1_MCASP_AUX 0x2: Selects divided version of SYS_CLK1. See CM_CLKSEL_VIDEO2_MCASP_AUX 0x3: Selects divided version of SYS_CLK1. See CM_CLKSEL_HDMI_MCASP_AUX	RW	0x0
21:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x2

Bits	Field Name	Description	Type	Reset
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guaranteed to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	R	0x2

Table 3-918. Register Call Summary for Register CM_L4PER2_MCASP6_CLKCTRL

Clock Management Functional Description
- CM_CORE_AON_MCASP1 Overview: [0]

PRCM Register Manual
- L4PER_CM_CORE Register Summary: [1]

Table 3-919. CM_L4PER2_MCASP7_CLKCTRL

Address Offset	0x0000 0208		
Physical Address	0x4A00 9908	**Instance**	L4PER_CM_CORE
Description	This register manages the McASP3 clocks.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				CLKSEL_AHCLKX				CLKSEL_AUX_CLK		RESERVED				IDLEST		RESERVED														MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:24	CLKSEL_AHCLKR	Selects reference clock for AHCLKR 0x0: Selects ABE_24M_GFCLK 0x1: Selects ABE_SYS_CLK 0x2: Selects FUNC_24M_GFCLK 0x3: Selects ATL_CLK3 0x4: Selects ATL_CLK2 0x5: Selects ATL_CLK1 0x6: Selects ATL_CLK0 0x7: Selects SYS_CLK2 0x8: Selects XREF_CLK0 0x9: Selects XREF_CLK1 0xA: Selects XREF_CLK2 0xB: Selects XREF_CLK1 0xC: Selects divided version of SYS_CLK1. See CM_CLKSEL_MLB_MCASP 0xD: Selects divided version of SYS_CLK1. See CM_CLKSEL_MLBP_MCASP 0xE: RESERVED 0xF: RESERVED	RW	0x0
23:22	CLKSEL_AUX_CLK	Selects the source of the AUX clock. 0x0: Selects divided version of SYS_CLK1. See CM_CLKSEL_PER_ABE_X1_GFCLK_MCASP_AUX 0x1: Selects divided version of SYS_CLK1. See CM_CLKSEL_VIDEO1_MCASP_AUX 0x2: Selects divided version of SYS_CLK1. See CM_CLKSEL_VIDEO2_MCASP_AUX 0x3: Selects divided version of SYS_CLK1. See CM_CLKSEL_HDMI_MCASP_AUX	RW	0x0
21:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x2
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guaranteed to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	R	0x2

Table 3-920. Register Call Summary for Register CM_L4PER2_MCASP7_CLKCTRL

Clock Management Functional Description
• CM_CORE_AON_MCASP1 Overview: [0]
PRCM Register Manual
• L4PER_CM_CORE Register Summary: [1]

Table 3-921. CM_L4PER2_STATICDEP

Address Offset	0x0000 020C		
Physical Address	0x4A00 990C	**Instance**	L4PER_CM_CORE
Description	This register controls the static domain depedencies from L4PER2 domain towards 'target' domains. It is relevant only for domain having system initiator(s).		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								IPU1_STATDEP	RESERVED				DSP2_STATDEP	RESERVED												L3MAIN1_STATDEP	RESERVED				DSP1_STATDEP	IPU2_STATDEP

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23	IPU1_STATDEP	Static dependency towards IPU1 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
22:19	RESERVED		R	0x0
18	DSP2_STATDEP	Static dependency towards DSP2 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
17:6	RESERVED		R	0x0
5	L3MAIN1_STATDEP	Static dependency towards L3MAIN1 clock domain 0x1: Dependency is enabled	R	0x1
4:2	RESERVED		R	0x0
1	DSP1_STATDEP	Static dependency towards DSP1 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	IPU2_STATDEP	Static dependency towards IPU2 clock domain 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-922. Register Call Summary for Register CM_L4PER2_STATICDEP

PRCM Register Manual
• L4PER_CM_CORE Register Summary: [0]

Table 3-923. CM_L4PER3_CLKSTCTRL

Address Offset	0x0000 0210		
Physical Address	0x4A00 9910	**Instance**	L4PER_CM_CORE
Description	This register enables the domain power state transition. It controls the HW supervised domain power state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																			CLKACTIVITY_DCC4_GFCLK	CLKACTIVITY_DCC3_GFCLK	CLKACTIVITY_DCC2_GFCLK	CLKACTIVITY_DCC1_GFCLK	CLKACTIVITY_L4PER3_L3_GICLK	RESERVED		RESERVED					CLKTRCTRL	

Bits	Field Name	Description	Type	Reset
31:13	RESERVED		R	0x0
12	CLKACTIVITY_TIMER16_GFCLK	This field indicates the state of the DMT16_GFCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
11	CLKACTIVITY_TIMER15_GFCLK	This field indicates the state of the DMT15_GFCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
10	CLKACTIVITY_TIMER14_GFCLK	This field indicates the state of the DMT14_GFCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
9	CLKACTIVITY_TIMER13_GFCLK	This field indicates the state of the DMT13_GFCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
8	CLKACTIVITY_L4PER3_L3_GICLK	This field indicates the state of the L4PER2_L3_GICLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
7:2	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
1:0	CLKTRCTRL	Controls the clock state transition of the L4PER clock domain. 0x0: NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may however occur. 0x1: SW_SLEEP: Start a software forced sleep transition on the domain. 0x2: SW_WKUP: Start a software forced wake-up transition on the domain. 0x3: HW_AUTO: Automatic transition is enabled. Sleep and wakeup transition are based upon hardware conditions.	RW	0x0

Table 3-924. Register Call Summary for Register CM_L4PER3_CLKSTCTRL

Clock Management Functional Description
- Clock Domain Modes: [0][1]

PRCM Register Manual
- Not Supported Functionality (Registers and Bitfields): [2][3][4][5]
- L4PER_CM_CORE Register Summary: [6]

Table 3-925. CM_L4PER3_DYNAMICDEP

Address Offset	0x0000 0214		
Physical Address	0x4A00 9914	**Instance**	L4PER_CM_CORE
Description	This register controls the dynamic domain depedencies from L4PER3 domain towards 'target' domains. It is relevant only for domain having OCP master port(s).		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ISS_DYNDEP	RESERVED			WINDOWSIZE				RTC_DYNDEP	RESERVED										L4CFG_DYNDEP	RESERVED		CAM_DYNDEP	RESERVED	L3INIT_DYNDEP	RESERVED	L3MAIN1_DYNDEP	RESERVED	IPU_DYNDEP	RESERVED		

Bits	Field Name	Description	Type	Reset
31	ISS_DYNDEP	Dynamic dependency towards ISS clock domain 0x1: Dependency is enabled	R	0x1
30:28	RESERVED		R	0x0
27:24	WINDOWSIZE	Size of sliding window used to monitor OCP interface activity for determination of auto-sleep feature. Time unit defined by CM_DYN_DEP_PRESCAL register.	RW	0x4
23	RTC_DYNDEP	Dynamic dependency towards RTC clock domain 0x1: Dependency is enabled	R	0x1
22:13	RESERVED		R	0x0
12	L4CFG_DYNDEP	Dynamic dependency towards L4CFG clock domain 0x1: Dependency is enabled	R	0x1
11:10	RESERVED		R	0x0
9	CAM_DYNDEP	Dynamic dependency towards CAM clock domain 0x1: Dependency is enabled	R	0x1
8	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
7	L3INIT_DYNDEP	Dynamic dependency towards L3INIT clock domain 0x1: Dependency is enabled	R	0x1
6	RESERVED		R	0x0
5	L3MAIN1_DYNDEP	Dynamic dependency towards L3MAIN1 clock domain 0x1: Dependency is enabled	R	0x1
4	RESERVED		R	0x0
3	IPU_DYNDEP	Dynamic dependency towards IPU clock domain 0x1: Dependency is enabled	R	0x1
2:0	RESERVED		R	0x0

Table 3-926. Register Call Summary for Register CM_L4PER3_DYNAMICDEP

Clock Management Functional Description • Clock Domain Dependency: [0][1][2][3][4][5]
PRCM Register Manual • L4PER_CM_CORE Register Summary: [6]

3.12.27 OCP_SOCKET_CM_CORE registers

3.12.27.1 OCP_SOCKET_CM_CORE Register Summary

Table 3-927. OCP_SOCKET_CM_CORE Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	OCP_SOCKET_CM_CORE Physical Address
REVISION_CM_CORE	R	32	0x0000 0000	0x4A00 8000
CM_CM_CORE_PROFILING_CLKCTRL	RW	32	0x0000 0040	0x4A00 8040
CM_CORE_DEBUG_CFG	RW	32	0x0000 00F0	0x4A00 80F0

3.12.27.2 OCP_SOCKET_CM_CORE Register Description

Table 3-928. REVISION_CM_CORE

Address Offset	0x0000 0000		
Physical Address	0x4A00 8000	**Instance**	OCP_SOCKET_CM_CORE
Description	This register contains the IP revision code for the CM_CORE part of the PRCM		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCHEME		RESERVED		FUNC												R_RTL					X_MAJOR			CUSTOM				Y_MINOR			

Bits	Field Name	Description	Type	Reset
31:30	SCHEME		R	0x1
29:28	RESERVED		R	0x0
27:16	FUNC	Function indicates a software compatible module family.	R	0x0
15:11	R_RTL	RTL Version (R)	R	0x0
10:8	X_MAJOR	Major Revision (X)	R	0x3

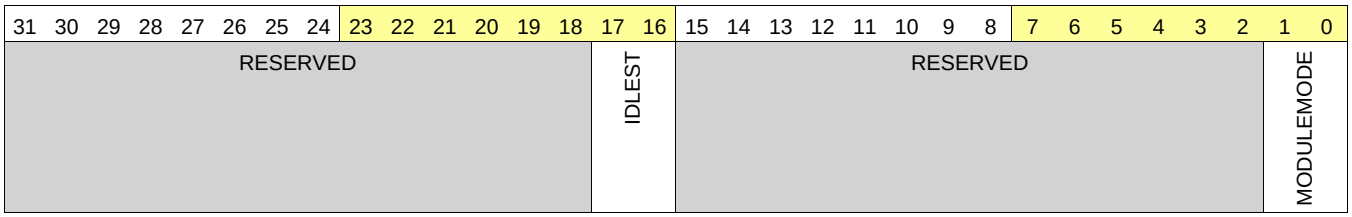
Bits	Field Name	Description	Type	Reset
7:6	CUSTOM	Indicates a special version for a particular device. 0x0: Non custom (standard) revision	R	0x0
5:0	Y_MINOR	Minor Revision (Y) 0x0: ES1.0 and similar versions 0x1: ES2.0 and similar versions	R	0x1

Table 3-929. Register Call Summary for Register REVISION_CM_CORE

PRCM Register Manual
- OCP_SOCKET_CM_CORE Register Summary: [0]

Table 3-930. CM_CM_CORE_PROFILING_CLKCTRL

Address Offset	0x0000 0040		
Physical Address	0x4A00 8040	**Instance**	OCP_SOCKET_CM_CORE
Description	This register manages the CM_CORE_PROFILING clocks. [warm reset insensitive]		
Type	RW		



Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status 0x0: Module is fully functional 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle 0x3: Module is disabled	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. OCP configuration port is not accessible. 0x1: Module is managed automatically by HW along with L3INSTR domain. 0x2: Reserved 0x3: Reserved	RW	0x1

Table 3-931. Register Call Summary for Register CM_CM_CORE_PROFILING_CLKCTRL

PRCM Register Manual
- OCP_SOCKET_CM_CORE Register Summary: [0]
- RESTORE_CM_CORE Register Description: [1][2]

Table 3-932. CM_CORE_DEBUG_CFG

Address Offset	0x0000 00F0		
Physical Address	0x4A00 80F0	**Instance**	OCP_SOCKET_CM_CORE
Description	This register is used to configure the CM_CORE's 32-bit debug output. There is one 8-bit source select field for selecting from a shared set of 8-bit internal signal blocks per byte. The signals included in each block are specified in the PRCM integration specification. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SEL3								SEL2								SEL1								SEL0							

Bits	Field Name	Description	Type	Reset
31:24	SEL3	Internal signal block select for debug word byte-3	RW	0x3
23:16	SEL2	Internal signal block select for debug word byte-2	RW	0x2
15:8	SEL1	Internal signal block select for debug word byte-1	RW	0x1
7:0	SEL0	Internal signal block select for debug word byte-0	RW	0x0

Table 3-933. Register Call Summary for Register CM_CORE_DEBUG_CFG

PRCM Register Manual
- OCP_SOCKET_CM_CORE Register Summary: [0]

3.12.28 RESTORE_CM_CORE registers

3.12.28.1 RESTORE_CM_CORE Register Summary

Table 3-934. RESTORE_CM_CORE Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	RESTORE_CM_CORE Physical Address
CM_L3MAIN1_CLKSTCTRL_RESTORE	RW	32	0x0000 0000	0x4A00 9E18
CM_L4CFG_CLKSTCTRL_RESTORE	RW	32	0x0000 0008	0x4A00 9E20
CM_L4PER_CLKSTCTRL_RESTORE	RW	32	0x0000 0010	0x4A00 9E28
CM_L3INIT_CLKSTCTRL_RESTORE	RW	32	0x0000 0014	0x4A00 9E2C
CM_L3INSTR_L3_MAIN_2_CLKCTRL_RESTORE	RW	32	0x0000 0018	0x4A00 9E30
CM_L3INSTR_L3_INSTR_CLKCTRL_RESTORE	RW	32	0x0000 001C	0x4A00 9E34
CM_L3INSTR_OCP_WP_NOC_CLKCTRL_RESTORE	RW	32	0x0000 0020	0x4A00 9E38
CM_CM_CORE_PROFILING_CLKCTRL_RESTORE	RW	32	0x0000 0024	0x4A00 9E3C
CM_L3MAIN1_DYNAMICDEP_RESTORE	RW	32	0x0000 0030	0x4A00 9E48
CM_L4CFG_DYNAMICDEP_RESTORE	RW	32	0x0000 0040	0x4A00 9E58
CM_L4PER_DYNAMICDEP_RESTORE	RW	32	0x0000 0044	0x4A00 9E5C
CM_COREAON_IO_SRCOMP_CLKCTRL_RESTORE	RW	32	0x0000 0048	0x4A00 9E60
CM_DMA_STATICDEP_RESTORE	RW	32	0x0000 0054	0x4A00 9E6C

3.12.28.2 RESTORE_CM_CORE Register Description

Table 3-935. CM_L3MAIN1_CLKSTCTRL_RESTORE

Address Offset	0x0000 0000		
Physical Address	0x4A00 9E18	**Instance**	RESTORE_CM_CORE
Description	Second address map for register CM_L3MAIN1_CLKSTCTRL. Used only by automatic restore upon wakeup from device OFF mode.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESTORE																															

Bits	Field Name	Description	Type	Reset
31:0	RESTORE	See CM_L3MAIN1_CLKSTCTRL register.	RW	0x0

Table 3-936. Register Call Summary for Register CM_L3MAIN1_CLKSTCTRL_RESTORE

PRCM Register Manual
- RESTORE_CM_CORE Register Summary: [0]

Table 3-937. CM_L4CFG_CLKSTCTRL_RESTORE

Address Offset	0x0000 0008		
Physical Address	0x4A00 9E20	**Instance**	RESTORE_CM_CORE
Description	Second address map for register CM_L4CFG_CLKSTCTRL. Used only by automatic restore upon wakeup from device OFF mode.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESTORE																															

Bits	Field Name	Description	Type	Reset
31:0	RESTORE	See CM_L4CFG_CLKSTCTRLregister.	RW	0x0

Table 3-938. Register Call Summary for Register CM_L4CFG_CLKSTCTRL_RESTORE

PRCM Register Manual
- RESTORE_CM_CORE Register Summary: [0]

Table 3-939. CM_L4PER_CLKSTCTRL_RESTORE

Address Offset	0x0000 0010		
Physical Address	0x4A00 9E28	**Instance**	RESTORE_CM_CORE
Description	Second address map for register CM_L4PER_CLKSTCTRL. Used only by automatic restore upon wakeup from device OFF mode.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESTORE																															

Bits	Field Name	Description	Type	Reset
31:0	RESTORE	See CM_L4PER_CLKSTCTRL register.	RW	0x0

Table 3-940. Register Call Summary for Register CM_L4PER_CLKSTCTRL_RESTORE

PRCM Register Manual
- RESTORE_CM_CORE Register Summary: [0]

Table 3-941. CM_L3INIT_CLKSTCTRL_RESTORE

Address Offset	0x0000 0014		
Physical Address	0x4A00 9E2C	**Instance**	RESTORE_CM_CORE
Description	Second address map for register CM_L3INIT_CLKSTCTRL. Used only by automatic restore upon wakeup from device OFF mode.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESTORE																															

Bits	Field Name	Description	Type	Reset
31:0	RESTORE	See CM_L3INIT_CLKSTCTRL register.	RW	0x0

Table 3-942. Register Call Summary for Register CM_L3INIT_CLKSTCTRL_RESTORE

PRCM Register Manual
- RESTORE_CM_CORE Register Summary: [0]

Table 3-943. CM_L3INSTR_L3_MAIN_2_CLKCTRL_RESTORE

Address Offset	0x0000 0018		
Physical Address	0x4A00 9E30	**Instance**	RESTORE_CM_CORE
Description	Second address map for register CM_L3INSTR_L3_MAIN_2_CLKCTRL. Used only by automatic restore upon wakeup from device OFF mode. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESTORE																															

Bits	Field Name	Description	Type	Reset
31:0	RESTORE	See CM_L3INSTR_L3_MAIN_3_CLKCTRL register.	RW	0x30001

Table 3-944. Register Call Summary for Register CM_L3INSTR_L3_MAIN_2_CLKCTRL_RESTORE

Clock Management Functional Description
- Clock Domain Module Attributes: [0][1]

PRCM Register Manual
- RESTORE_CM_CORE Register Summary: [2]

Table 3-945. CM_L3INSTR_L3_INSTR_CLKCTRL_RESTORE

Address Offset	0x0000 001C		
Physical Address	0x4A00 9E34	**Instance**	RESTORE_CM_CORE
Description	Second address map for register CM_L3INSTR_L3_INSTR_CLKCTRL. Used only by automatic restore upon wakeup from device OFF mode. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESTORE																															

Bits	Field Name	Description	Type	Reset
31:0	RESTORE	See CM_L3INSTR_L3_INSTR_CLKCTRL register.	RW	0x30001

Table 3-946. Register Call Summary for Register CM_L3INSTR_L3_INSTR_CLKCTRL_RESTORE

Clock Management Functional Description
- Clock Domain Module Attributes: [0][1]

PRCM Register Manual
- RESTORE_CM_CORE Register Summary: [2]

Table 3-947. CM_L3INSTR_OCP_WP_NOC_CLKCTRL_RESTORE

Address Offset	0x0000 0020		
Physical Address	0x4A00 9E38	**Instance**	RESTORE_CM_CORE
Description	Second address map for register CM_L3INSTR_OCP_WP_NOC_CLKCTRL. Used only by automatic restore upon wakeup from device OFF mode. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESTORE																															

Bits	Field Name	Description	Type	Reset
31:0	RESTORE	See CM_L3INSTR_OCP_WP_NOC_CLKCTRL register.	RW	0x30001

Table 3-948. Register Call Summary for Register CM_L3INSTR_OCP_WP_NOC_CLKCTRL_RESTORE

Clock Management Functional Description
- Clock Domain Module Attributes: [0][1]

PRCM Register Manual
- RESTORE_CM_CORE Register Summary: [2]

Table 3-949. CM_CM_CORE_PROFILING_CLKCTRL_RESTORE

Address Offset	0x0000 0024		
Physical Address	0x4A00 9E3C	**Instance**	RESTORE_CM_CORE
Description	Second address map for register CM_CM_CORE_PROFILING_CLKCTRL. Used only by automatic restore upon wakeup from device OFF mode. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESTORE																															

Bits	Field Name	Description	Type	Reset
31:0	RESTORE	See CM_CM_CORE_PROFILING_CLKCTRL register.	RW	0x30001

Table 3-950. Register Call Summary for Register CM_CM_CORE_PROFILING_CLKCTRL_RESTORE

PRCM Register Manual
- RESTORE_CM_CORE Register Summary: [0]

Table 3-951. CM_L3MAIN1_DYNAMICDEP_RESTORE

Address Offset	0x0000 0030		
Physical Address	0x4A00 9E48	**Instance**	RESTORE_CM_CORE
Description	Second address map for register CM_L3MAIN1_DYNAMICDEP. Used only by automatic restore upon wakeup from device OFF mode.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESTORE																															

Bits	Field Name	Description	Type	Reset
31:0	RESTORE	See CM_L3MAIN1_DYNAMICDEP register.	RW	0x4001058

Table 3-952. Register Call Summary for Register CM_L3MAIN1_DYNAMICDEP_RESTORE

PRCM Register Manual
- RESTORE_CM_CORE Register Summary: [0]

Table 3-953. CM_L4CFG_DYNAMICDEP_RESTORE

Address Offset	0x0000 0040		
Physical Address	0x4A00 9E58	**Instance**	RESTORE_CM_CORE
Description	Second address map for register CM_L4CFG_DYNAMICDEP. Used only by automatic restore upon wakeup from device OFF mode.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESTORE																															

Bits	Field Name	Description	Type	Reset
31:0	RESTORE	See CM_L4CFG_DYNAMICDEP register.	RW	0x40789f2

Table 3-954. Register Call Summary for Register CM_L4CFG_DYNAMICDEP_RESTORE

PRCM Register Manual
- RESTORE_CM_CORE Register Summary: [0]

Table 3-955. CM_L4PER_DYNAMICDEP_RESTORE

Address Offset	0x0000 0044		
Physical Address	0x4A00 9E5C	**Instance**	RESTORE_CM_CORE
Description	Second address map for register CM_L4PER_DYNAMICDEP. Used only by automatic restore upon wakeup from device OFF mode.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESTORE																															

Bits	Field Name	Description	Type	Reset
31:0	RESTORE	See CM_L4PER_DYNAMICDEP register.	RW	0x4004180

Table 3-956. Register Call Summary for Register CM_L4PER_DYNAMICDEP_RESTORE

PRCM Register Manual
- RESTORE_CM_CORE Register Summary: [0]

Table 3-957. CM_COREAON_IO_SRCOMP_CLKCTRL_RESTORE

Address Offset	0x0000 0048		
Physical Address	0x4A00 9E60	**Instance**	RESTORE_CM_CORE
Description	Second address map for register CM_COREAON_IO_SRCOMP_CLKCTRL. Used only by automatic restore upon wakeup from device OFF mode.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESTORE																															

Bits	Field Name	Description	Type	Reset
31:0	RESTORE	See CCM_DYN_DEP_PRESCAL register.	RW	0x20

Table 3-958. CM_DMA_STATICDEP_RESTORE

Address Offset	0x0000 0054		
Physical Address	0x4A00 9E6C	**Instance**	RESTORE_CM_CORE
Description	Second address map for register CM_DMA_STATICDEP. Used only by automatic restore upon wakeup from device OFF mode.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESTORE																															

Bits	Field Name	Description	Type	Reset
31:0	RESTORE	See CM_DMA_STATICDEP register.	RW	0xb0f0

Table 3-959. Register Call Summary for Register CM_DMA_STATICDEP_RESTORE

PRCM Register Manual
- RESTORE_CM_CORE Register Summary: [0]

3.12.29 SMARTREFLEX_CORE registers

3.12.29.1 SMARTREFLEX_CORE Register Summary

Table 3-960. SMARTREFLEX_CORE Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	SMARTREFLEX_CORE Physical Address	SMARTREFLEX_DSPEVE Physical Address
SRCONFIG	RW	32	0x0000 0000	0x4A0D D000	0x4A18 3000
SRSTATUS	R	32	0x0000 0004	0x4A0D D004	0x4A18 3004
SENVAL	R	32	0x0000 0008	0x4A0D D008	0x4A18 3008
SENMIN	R	32	0x0000 000C	0x4A0D D00C	0x4A18 300C
SENMAX	R	32	0x0000 0010	0x4A0D D010	0x4A18 3010
SENAVG	R	32	0x0000 0014	0x4A0D D014	0x4A18 3014
AVGWEIGHT	RW	32	0x0000 0018	0x4A0D D018	0x4A18 3018
NVALUERECIPROCAL	RW	32	0x0000 001C	0x4A0D D01C	0x4A18 301C
IRQ_EOI	RW	32	0x0000 0020	0x4A0D D020	0x4A18 3020
IRQSTATUS_RAW	RW	32	0x0000 0024	0x4A0D D024	0x4A18 3024
IRQSTATUS	RW	32	0x0000 0028	0x4A0D D028	0x4A18 3028
IRQENABLE_SET	RW	32	0x0000 002C	0x4A0D D02C	0x4A18 302C
IRQENABLE_CLR	RW	32	0x0000 0030	0x4A0D D030	0x4A18 3030
SENERROR	R	32	0x0000 0034	0x4A0D D034	0x4A18 3034
ERRCONFIG	RW	32	0x0000 0038	0x4A0D D038	0x4A18 3038

3.12.29.2 SMARTREFLEX_CORE Register Description

Table 3-961. SRCONFIG

Address Offset	0x0000 0000		
Physical Address	0x4A0D D000 0x4A18 3000	**Instance**	SMARTREFLEX_CORE SMARTREFLEX_DSPEVE
Description	Configuration bits for the Sensor Core and the Digital Processing		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ACCUMDATA								SRCLKLENGTH												SRENABLE	SENENABLE	ERRORGENERATORENABLE	MINMAXAVGENABLE	RESERVED						SENNENABLE	SENPENABLE

Bits	Field Name	Description	Type	Reset
31:22	ACCUMDATA	Number of values to accumulate.	RW	0x80
21:12	SRCLKLENGTH	Determines the frequency of SRClk.	RW	0x200
11	SRENABLE	0: Syncronously resets MinMaxAvgAccumValid, MinMaxAvgValid, ErrorGeneratorValid, AccumData sensor, SRClk counter, and MinMaxAvg registers. Also gates the clock for power savings and disables all of the digital logic. 1: Enables the module	RW	0x0
10	SENENABLE	0: Both N and P sensors disabled (SVT) 1: Sensors enabled per SenNEnable SenPEnable	RW	0x1
9	ERRORGENERATORENABLE	0: Error Generator Module disabled 1: Error Generator Module enabled	RW	0x0
8	MINMAXAVGENABLE	0: Min/Max/Avg Detector Module disabled 1: Min/Max/Avg Detector Module enabled	RW	0x0
7:2	RESERVED		R	0x0
1	SENNENABLE	0: Disables SenN sensor 1: Enables SenN sensor	RW	0x1
0	SENPENABLE	0: Disables SenP sensor 1: Enables SenP sensor	RW	0x0

Table 3-962. SRSTATUS

Address Offset	0x0000 0004		
Physical Address	0x4A0D D004 0x4A18 3004	**Instance**	SMARTREFLEX_CORE SMARTREFLEX_DSPEVE
Description	Status bits that indicate that the values in the register are valid or events have occurred		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												AVGERRVALID	MINMAXAVGVALID	ERRORGENERATORVALID	MINMAXAVGACCUMVALID

Bits	Field Name	Description	Type	Reset
31:4	RESERVED		R	0x0
3	AVGERRVALID	0: AvgError registers are not valid 1: AvgError registers are valid	R	0x0
2	MINMAXAVGVALID	0: SenVal, SenMin, SenMax, SenAvg registers are not valid 1: SenVal, SenMin, SenMax, SenAvg registers are valid, but not necessarily fully accumulated	R	0x0
1	ERRORGENERATORVALID	0: SenError register do not have valid data 1: SenError registers have valid data	R	0x0
0	MINMAXAVGACCUMVALID	0: SenVal, SenMin, SenMax, SenAvg registers are not valid 1: SenVal, SenMin, SenMax, SenAvg registers have valid, final data	R	0x0

Table 3-963. SENVAL

Address Offset	0x0000 0008		
Physical Address	0x4A0D D008 0x4A18 3008	**Instance**	SMARTREFLEX_CORE SMARTREFLEX_DSPEVE
Description	The current sensor values from the Sensor Core(SVT)		

Table 3-963. SENVAL (continued)

Type	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SENPVAL																SENNVAL															

Bits	Field Name	Description	Type	Reset
31:16	SENPVAL	The latest value of the SenPVal from the SVT sensor core	R	0x0
15:0	SENNVAL	The latest value of the SenNVal from the SVT sensor core	R	0x0

Table 3-964. SENMIN

Address Offset	0x0000 000C		
Physical Address	0x4A0D D00C 0x4A18 300C	Instance	SMARTREFLEX_CORE SMARTREFLEX_DSPEVE
Description	The minimum sensor values(SVT)		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SENPMIN																SENNMIN															

Bits	Field Name	Description	Type	Reset
31:16	SENPMIN	The minimum value of the SenPVal from the SVT sensor core since the last restart operation	R	0xffff
15:0	SENNMIN	The minimum value of the SenNVal from the SVT sensor core since the last restart operation	R	0xffff

Table 3-965. SENMAX

Address Offset	0x0000 0010		
Physical Address	0x4A0D D010 0x4A18 3010	Instance	SMARTREFLEX_CORE SMARTREFLEX_DSPEVE
Description	The maximum sensor values(SVT)		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SENPMAX																SENNMAX															

Bits	Field Name	Description	Type	Reset
31:16	SENPMAX	The maximum value of the SenPVal from the SVT sensor core since the last restart operation	R	0x0
15:0	SENNMAX	The maximum value of the SenNVal from the SVT sensor core since the last restart operation	R	0x0

Table 3-966. SENAVG

Address Offset	0x0000 0014		
Physical Address	0x4A0D D014 0x4A18 3014	Instance	SMARTREFLEX_CORE SMARTREFLEX_DSPEVE
Description	The average sensor values(SVT)		

Table 3-966. SENAVG (continued)

Type	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SENPAVG																SENNAVG															

Bits	Field Name	Description	Type	Reset
31:16	SENPAVG	The running average of the SenPVal from the SVT sensor core since the last restart operation	R	0x0
15:0	SENNAVG	The running average of the SenNVal from the SVT sensor core since the last restart operation	R	0x0

Table 3-967. AVGWEIGHT

Address Offset	0x0000 0018		
Physical Address	0x4A0D D018 0x4A18 3018	Instance	SMARTREFLEX_CORE SMARTREFLEX_DSPEVE
Description	The weighting factor in the average computation		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SENPAVGWEIGHT1																SENNAVGWEIGHT															

Bits	Field Name	Description	Type	Reset
31:16	SENPAVGWEIGHT1	The weighting factor for the SenP averager	RW	0x0
15:0	SENNAVGWEIGHT	The weighting factor for the SenN averager	RW	0x0

Table 3-968. NVALUERECIPROCAL

Address Offset	0x0000 001C		
Physical Address	0x4A0D D01C 0x4A18 301C	Instance	SMARTREFLEX_CORE SMARTREFLEX_DSPEVE
Description	The reciprocal of the SenN and SenP values used in error generation(SVT)		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								SENPGAIN				SENNGAIN				SENPRN								SENNRN							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:20	SENPGAIN	The gain value for the SVT SenP reciprocal	RW	0x0
19:16	SENNGAIN	The gain value for the SVT SenN reciprocal	RW	0x0
15:8	SENPRN	The scale value for the SVT SenP reciprocal	RW	0x0
7:0	SENNRN	The scale value for the SVT SenN reciprocal	RW	0x0

Table 3-969. IRQ_EOI

Address Offset	0x0000 0020		
Physical Address	0x4A0D D020 0x4A18 3020	Instance	SMARTREFLEX_CORE SMARTREFLEX_DSPEVE
Description	EOI protocol re-trigger		

Table 3-969. IRQ_EOI (continued)

Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															EOI

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	EOI	The value read is always '0' Write: 0: re-evaluate pending sources re-send intr* 1: No change to interrupt	RW	0x0

Table 3-970. IRQSTATUS_RAW

Address Offset	0x0000 0024		
Physical Address	0x4A0D D024 0x4A18 3024	Instance	SMARTREFLEX_CORE SMARTREFLEX_DSPEVE
Description	MCU raw interrupt raw status and set		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												MCUACCUMINTSTATRAW	MCUVALIDINTSTATRAW	MCUBOUNDSINTSTATRAW	MCUDISABLEACKINTSTATRAW

Bits	Field Name	Description	Type	Reset
31:4	RESERVED		R	0x0
3	MCUACCUMINTSTATRAW	Read:Accum interrupt status Write: 0: Accum interrupt status is unchanged 1: Accum interrupt status is set	RW	0x0
2	MCUVALIDINTSTATRAW	Read:Valid interrupt status Write: 0: Valid status is unchanged 1: Valid status is set	RW	0x0
1	MCUBOUNDSINTSTATRAW	Read:Bounds interrupt status Write: 0: Bounds interrupt status is unchanged 1: Bounds interrupt status is set	RW	0x0
0	MCUDISABLEACKINTSTATRAW	Read:MCUDisable acknowledge interrupt status Write: 0: MCUDisable acknowledge status is unchanged 1: MCUDisable acknowledge status is set	RW	0x0

Table 3-971. IRQSTATUS

Address Offset	0x0000 0028		
Physical Address	0x4A0D D028 0x4A18 3028	Instance	SMARTREFLEX_CORE SMARTREFLEX_DSPEVE
Description	MCU masked interrupt status and clear		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												MCUACCUMINTSTATENA	MCUVALIDINTSTATENA	MCUBOUNDSINTSTATENA	MCUDISABLEACKINTSTATENA

Bits	Field Name	Description	Type	Reset
31:4	RESERVED		R	0x0
3	MCUACCUMINTSTATENA	Read:Accum interrupt status if enabled Write: 0: Accum interrupt status is unchanged 1: Accum interrupt status is cleared	RW	0x0
2	MCUVALIDINTSTATENA	Read:Valid interrupt status if enabled Write: 0: Valid interrupt status is unchanged 1: Valid interrupt status is cleared	RW	0x0
1	MCUBOUNDSINTSTATENA	Read:Bounds interrupt status if enabled Write: 0: Bounds interrupt status is unchanged 1: Bounds interrupt status is cleared	RW	0x0
0	MCUDISABLEACKINTSTATENA	Read:MCUDisable acknowledge interrupt status if enabled Write: 0: MCUDisable acknowledge status is unchanged 1: MCUDisable acknowledge status is cleared	RW	0x0

Table 3-972. IRQENABLE_SET

Address Offset	0x0000 002C		
Physical Address	0x4A0D D02C 0x4A18 302C	**Instance**	SMARTREFLEX_CORE SMARTREFLEX_DSPEVE
Description	MCU interrupt enable flag set		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												MCUACCUMINTENASET	MCUVALIDINTENASET	MCUBOUNDSINTENASET	MCUDISABLEACKINTENASET

Bits	Field Name	Description	Type	Reset
31:4	RESERVED		R	0x0
3	MCUACCUMINTENASET	Read: 0: Accum interrupt generation is disabled/masked 1: Accum interrupt generation is enabled Write: 0: No change to Accum interrupt enable 1: Enable Accum interrupt generation	RW	0x0

Bits	Field Name	Description	Type	Reset
2	MCUVALIDINTENASET	Read: 0: Valid interrupt generation is disabled/masked 1: Valid interrupt generation is enabled Write: 0: No change to Valid interrupt enable 1: Enable Valid interrupt generation	RW	0x0
1	MCUBOUNDSINTENASET	Read: 0: Bounds interrupt generation is disabled/masked 1: Bounds interrupt generation is enabled Write: 0: No change to Bounds interrupt enable 1: Enable Bounds interrupt generation	RW	0x0
0	MCUDISABLEACKINTENASET	Read: 0: MCUDisableAck interrupt generation is disabled/masked 1: MCUDisableAck interrupt generation is enabled Write: 0: No change to MCUDisAck interrupt enable 1: Enable MCUDisableAck interrupt generation	RW	0x0

Table 3-973. IRQENABLE_CLR

Address Offset	0x0000 0030		
Physical Address	0x4A0D D030 0x4A18 3030	**Instance**	SMARTREFLEX_CORE SMARTREFLEX_DSPEVE
Description	MCU interrupt enable flag clear		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												MCUACCUMINTENACLR	MCUVALIDINTENACLR	MCUBOUNDSINTENACLR	MCUDISABLEACKINTENACLR

Bits	Field Name	Description	Type	Reset
31:4	RESERVED		R	0x0
3	MCUACCUMINTENACLR	Read: 0: Accum interrupt generation is disabled/masked 1: Accum interrupt generation is enabled Write: 0: No change to Disable Accum interrupt enable 1: Disable Accum interrupt generation	RW	0x0
2	MCUVALIDINTENACLR	Read: 0: Valid interrupt generation is disabled/masked 1: Valid interrupt generation is enabled Write: 0: No change to Disable Valid interrupt enable 1: Disable Valid interrupt generation	RW	0x0
1	MCUBOUNDSINTENACLR	Read: 0: Bounds interrupt generation is disabled/masked 1: Bounds interrupt generation is enabled Write: 0: No change to Bounds interrupt enable 1: Disable Bounds interrupt generation	RW	0x0
0	MCUDISABLEACKINTENACLR	Read: 0: MCUDisableAck interrupt generation is disabled/masked 1: MCUDisableAck interrupt generation is enabled Write: 0: No change to MCUDisAck interrupt enable 1: Disable MCUDisableAck interrupt generation	RW	0x0

Table 3-974. SENERROR

Address Offset	0x0000 0034		
Physical Address	0x4A0D D034 0x4A18 3034	**Instance**	SMARTREFLEX_CORE SMARTREFLEX_DSPEVE
Description	The sensor error from the error generator		

Table 3-974. SENERROR (continued)

Type	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																AVGERROR								SENERROR							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0
15:8	AVGERROR	The average sensor error	R	0x0
7:0	SENERROR	The percentage of sensor error	R	0x0

Table 3-975. ERRCONFIG

Address Offset	0x0000 0038		
Physical Address	0x4A0D D038 0x4A18 3038	Instance	SMARTREFLEX_CORE SMARTREFLEX_DSPEVE
Description	The sensor error configuration		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED					WAKEUPENABLE	IDLEMODE		VPBOUNDSINTSTATENA	VPBOUNDSINTENABLE	RESERVED			ERRWEIGHT			ERRMAXLIMIT								ERRMINLIMIT							

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26	WAKEUPENABLE	Wakeup from MCU Interrupts enable	RW	0x0
25:24	IDLEMODE	0b00: Force-Idle Mode 0b01: No Idle Mode 0b10: SmartIdle Mode #2 0b11: Smart-Idle-Wkup mode	RW	0x2
23	VPBOUNDSINTSTATENA	Read: Bounds interrupt status if enabled Write: 0: Bounds interrupt status is unchanged 1: Bounds interrupt status is cleared	RW	0x0
22	VPBOUNDSINTENABLE	0: Bounds interrupt disabled 1: Bounds interrupt enabled	RW	0x0
21:19	RESERVED		R	0x0
18:16	ERRWEIGHT	The AvgSenError weight.	RW	0x0
15:8	ERRMAXLIMIT	The upper limit of SenError for interrupt generation	RW	0x7f
7:0	ERRMINLIMIT	The lower limit of SenError for interrupt generation	RW	0x80

3.12.30 CAM_PRM registers

3.12.30.1 CAM_PRM Register Summary

Table 3-976. CAM_PRM Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	CAM_PRM Physical Address
PM_CAM_PWRSTCTRL	RW	32	0x0000 0000	0x4AE0 7000
PM_CAM_PWRSTST	RW	32	0x0000 0004	0x4AE0 7004
PM_CAM_VIP1_WKDEP	RW	32	0x0000 0020	0x4AE0 7020
RM_CAM_VIP1_CONTEXT	RW	32	0x0000 0024	0x4AE0 7024
PM_CAM_VIP2_WKDEP	RW	32	0x0000 0028	0x4AE0 7028
RM_CAM_VIP2_CONTEXT	RW	32	0x0000 002C	0x4AE0 702C
PM_CAM_VIP3_WKDEP	RW	32	0x0000 0030	0x4AE0 7030
RM_CAM_VIP3_CONTEXT	RW	32	0x0000 0034	0x4AE0 7034
RM_CAM_LVDSRX_CONTEXT	RW	32	0x0000 003C	0x4AE0 703C
RM_CAM_CSI1_CONTEXT	RW	32	0x0000 0044	0x4AE0 7044
RM_CAM_CSI2_CONTEXT	RW	32	0x0000 004C	0x4AE0 704C

3.12.30.2 CAM_PRM Register Description

Table 3-977. PM_CAM_PWRSTCTRL

Address Offset	0x0000 0000		
Physical Address	0x4AE0 7000	**Instance**	CAM_PRM
Description	This register controls the CAM power state to reach upon a domain sleep transition		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														VIP_BANK_ONSTATE		RESERVED											LOWPOWERSTATECHANGE	RESERVED		POWERSTATE	

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	VIP_BANK_ONSTATE	VIP_BANK memory state when domain is ON. 0x3: Memory bank is on when the domain is ON.	R	0x3
15:5	RESERVED		R	0x0
4	LOWPOWERSTATECHANGE	Power state change request when domain has already performed a sleep transition. Allows going into deeper low power state without waking up the power domain. 0x0: Do not request a low power state change. 0x1: Request a low power state change. This bit is automatically cleared when the power state is effectively changed or when power state is ON.	RW	0x0

Bits	Field Name	Description	Type	Reset
3:2	RESERVED		R	0x0
1:0	POWERSTATE	Power state control 0x0: OFF state 0x1: Reserved 0x2: INACTIVE state 0x3: ON State	RW	0x0

Table 3-978. Register Call Summary for Register PM_CAM_PWRSTCTRL

Power Management Functional Description • Logic and Memory Area Power Mode Control and Status: [0][1][2]
PRCM Register Manual • CAM_PRM Register Summary: [3]

Table 3-979. PM_CAM_PWRSTST

Address Offset	0x0000 0004		
Physical Address	0x4AE0 7004	**Instance**	CAM_PRM
Description	This register provides a status on the current CAM power domain state. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED						LASTPOWERSTATEENTERED		RESERVED			INTRANSITION	RESERVED												VIP_BANK_STATEST		RESERVED		LOGICSTATEST		POWERSTATEST	

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:24	LASTPOWERSTATEENTERED	Last low power state entered. Set to 0x3 upon write of the same only. This register is intended for debug purpose only. 0x0: Power domain was previously OFF 0x1: Power domain was previously in RETENTION 0x2: Power domain was previously ON-INACTIVE 0x3: Power domain was previously ON-ACTIVE	RW	0x0
23:21	RESERVED		R	0x0
20	INTRANSITION	Domain transition status 0x0: No on-going transition on power domain 0x1: Power domain transition is in progress.	R	0x0
19:6	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
5:4	VIP_BANK_STATEST	VIP_BANK memory state status 0x0: Memory is OFF 0x1: Reserved 0x2: Reserved 0x3: Memory is ON	R	0x3
3	RESERVED		R	0x0
2	LOGICSTATEST	Logic state status 0x0: Logic in domain is OFF 0x1: Logic in domain is ON	R	0x1
1:0	POWERSTATEST	Current power state status 0x0: Power domain is OFF 0x1: Power domain is in RETENTION 0x2: Power domain is ON-INACTIVE 0x3: Power domain is ON-ACTIVE	R	0x3

Table 3-980. Register Call Summary for Register PM_CAM_PWRSTST

Power Management Functional Description
- Logic and Memory Area Power Mode Control and Status: [0][1][2][3][4]

PRCM Register Manual
- Not Supported Functionality (Registers and Bitfields): [5][6][7][8][9]
- CAM_PRM Register Summary: [10]

Table 3-981. PM_CAM_VIP1_WKDEP

Address Offset	0x0000 0020		
Physical Address	0x4AE0 7020	**Instance**	CAM_PRM
Description	This register controls wakeup dependency based on VIP1 service requests.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						WKUPDEP_VIP1_EVE4	WKUPDEP_VIP1_EVE3	WKUPDEP_VIP1_EVE2	WKUPDEP_VIP1_EVE1	WKUPDEP_VIP1_DSP2	WKUPDEP_VIP1_IPU1	RESERVED	WKUPDEP_VIP1_DSP1	WKUPDEP_VIP1_IPU2	WKUPDEP_VIP1_MPU

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	WKUPDEP_VIP1_EVE4	Wakeup dependency from VIP1 module (SWakeup signal) towards EVE4 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
8	WKUPDEP_VIP1_EVE3	Wakeup dependency from VIP1 module (SWakeup signal) towards EVE3 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
7	WKUPDEP_VIP1_EVE2	Wakeup dependency from VIP1 module (SWakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_VIP1_EVE1	Wakeup dependency from VIP1 module (SWakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_VIP1_DSP2	Wakeup dependency from VIP1 module (SWakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_VIP1_IPU1	Wakeup dependency from VIP1 module (SWakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	RESERVED		R	0x0
2	WKUPDEP_VIP1_DSP1	Wakeup dependency from VIP1 module (SWakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_VIP1_IPU2	Wakeup dependency from vip1 module (SWakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_VIP1_MPU	Wakeup dependency from VIP1 module (SWakeup signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-982. Register Call Summary for Register PM_CAM_VIP1_WKDEP

PRCM Register Manual
- Not Supported Functionality (Registers and Bitfields): [0][1][2][3][4]
- CAM_PRM Register Summary: [5]

Table 3-983. RM_CAM_VIP1_CONTEXT

Address Offset	0x0000 0024		
Physical Address	0x4AE0 7024	**Instance**	CAM_PRM
Description	This register contains dedicated VIP1 context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							LOSTMEM_VIP_BANK	RESERVED							LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	LOSTMEM_VIP_BANK	Specify if memory-based context in VIP_MEM memory bank has been lost due to a previous power transition or other reset source. 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
7:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CAM_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-984. Register Call Summary for Register RM_CAM_VIP1_CONTEXT

Power Management Functional Description • PD_CAM Description: [0]
PRCM Register Manual • CAM_PRM Register Summary: [1]

Table 3-985. PM_CAM_VIP2_WKDEP

Address Offset	0x0000 0028		
Physical Address	0x4AE0 7028	**Instance**	CAM_PRM
Description	This register controls wakeup dependency based on VIP2 service requests.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						WKUPDEP_VIP2_EVE4	WKUPDEP_VIP2_EVE3	WKUPDEP_VIP2_EVE2	WKUPDEP_VIP2_EVE1	WKUPDEP_VIP2_DSP2	WKUPDEP_VIP2_IPU1	RESERVED	WKUPDEP_VIP2_DSP1	WKUPDEP_VIP2_IPU2	WKUPDEP_VIP2_MPU

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	WKUPDEP_VIP2_EVE4	Wakeup dependency from VIP2 module (SWakeup signal) towards EVE4 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
8	WKUPDEP_VIP2_EVE3	Wakeup dependency from VIP2 module (SWakeup signal) towards EVE3 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
7	WKUPDEP_VIP2_EVE2	Wakeup dependency from VIP2 module (SWakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_VIP2_EVE1	Wakeup dependency from VIP2 module (SWakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_VIP2_DSP2	Wakeup dependency from VIP1 module (SWakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_VIP2_IPU1	Wakeup dependency from VIP2 module (SWakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	RESERVED		R	0x0
2	WKUPDEP_VIP2_DSP1	Wakeup dependency from VIP2 module (SWakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_VIP2_IPU2	Wakeup dependency from VIP2 module (SWakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_VIP2_MPU	Wakeup dependency from VIP2 module (SWakeup signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-986. Register Call Summary for Register PM_CAM_VIP2_WKDEP

PRCM Register Manual
- CAM_PRM Register Summary: [0]

Table 3-987. RM_CAM_VIP2_CONTEXT

Address Offset	0x0000 002C		
Physical Address	0x4AE0 702C	**Instance**	CAM_PRM
Description	This register contains dedicated VIP2 context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							LOSTMEM_VIP_BANK	RESERVED							LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	LOSTMEM_VIP_BANK	Specify if memory-based context in VIP_MEM memory bank has been lost due to a previous power transition or other reset source. 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
7:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CAM_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-988. Register Call Summary for Register RM_CAM_VIP2_CONTEXT

PRCM Register Manual
- CAM_PRM Register Summary: [0]

Table 3-989. PM_CAM_VIP3_WKDEP

Address Offset	0x0000 0030		
Physical Address	0x4AE0 7030	**Instance**	CAM_PRM
Description	This register controls wakeup dependency based on VIP3 service requests.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						WKUPDEP_VIP3_EVE4	WKUPDEP_VIP3_EVE3	WKUPDEP_VIP3_EVE2	WKUPDEP_VIP3_EVE1	WKUPDEP_VIP3_DSP2	WKUPDEP_VIP3_IPU1	RESERVED	WKUPDEP_VIP3_DSP1	WKUPDEP_VIP3_IPU2	WKUPDEP_VIP3_MPU

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	WKUPDEP_VIP3_EVE4	Wakeup dependency from VIP3 module (SWakeup signal) towards EVE4 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
8	WKUPDEP_VIP3_EVE3	Wakeup dependency from VIP3 module (SWakeup signal) towards EVE3 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
7	WKUPDEP_VIP3_EVE2	Wakeup dependency from VIP3 module (SWakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_VIP3_EVE1	Wakeup dependency from VIP3 module (SWakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_VIP3_DSP2	Wakeup dependency from VIP3 module (SWakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_VIP3_IPU1	Wakeup dependency from VIP3 module (SWakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	RESERVED		R	0x0
2	WKUPDEP_VIP3_DSP1	Wakeup dependency from VIP3 module (SWakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_VIP3_IPU2	Wakeup dependency from vip3 module (SWakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_VIP3_MPU	Wakeup dependency from VIP3 module (SWakeup signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-990. Register Call Summary for Register PM_CAM_VIP3_WKDEP

PRCM Register Manual
- CAM_PRM Register Summary: [0]

Table 3-991. RM_CAM_VIP3_CONTEXT

Address Offset	0x0000 0034		
Physical Address	0x4AE0 7034	**Instance**	CAM_PRM
Description	This register contains dedicated VIP3 context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							LOSTMEM_VIP_BANK	RESERVED							LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	LOSTMEM_VIP_BANK	Specify if memory-based context in VIP_MEM memory bank has been lost due to a previous power transition or other reset source. 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
7:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CAM_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-992. Register Call Summary for Register RM_CAM_VIP3_CONTEXT

PRCM Register Manual
- CAM_PRM Register Summary: [0]

Table 3-993. RM_CAM_LVDSRX_CONTEXT

Address Offset	0x0000 003C		
Physical Address	0x4AE0 703C	**Instance**	CAM_PRM
Description	This register contains dedicated LVDSRX context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CAM_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-994. Register Call Summary for Register RM_CAM_LVDSRX_CONTEXT

PRCM Register Manual
- CAM_PRM Register Summary: [0]

Table 3-995. RM_CAM_CSI1_CONTEXT

Address Offset	0x0000 0044		
Physical Address	0x4AE0 7044	**Instance**	CAM_PRM
Description	This register contains dedicated CSI1 context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CAM_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-996. Register Call Summary for Register RM_CAM_CSI1_CONTEXT

PRCM Register Manual
- CAM_PRM Register Summary: [0]

Table 3-997. RM_CAM_CSI2_CONTEXT

Address Offset	0x0000 004C		
Physical Address	0x4AE0 704C	**Instance**	CAM_PRM
Description	This register contains dedicated CSI2 context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CAM_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-998. Register Call Summary for Register RM_CAM_CSI2_CONTEXT

PRCM Register Manual
- CAM_PRM Register Summary: [0]

3.12.31 CKGEN_PRM registers

3.12.31.1 CKGEN_PRM Register Summary

Table 3-999. CKGEN_PRM Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	CKGEN_PRM Physical Address
CM_CLKSEL_SYSCLK1	RW	32	0x0000 0000	0x4AE0 6100
CM_CLKSEL_WKUPAON	RW	32	0x0000 0008	0x4AE0 6108
CM_CLKSEL_ABE_PLL_REF	RW	32	0x0000 000C	0x4AE0 610C
CM_CLKSEL_SYS	RW	32	0x0000 0010	0x4AE0 6110
CM_CLKSEL_ABE_PLL_BYPAS	RW	32	0x0000 0014	0x4AE0 6114
CM_CLKSEL_ABE_PLL_SYS	RW	32	0x0000 0018	0x4AE0 6118
CM_CLKSEL_ABE_24M	RW	32	0x0000 001C	0x4AE0 611C
CM_CLKSEL_ABE_SYS	RW	32	0x0000 0020	0x4AE0 6120
CM_CLKSEL_HDMI_MCASP_AUX	RW	32	0x0000 0024	0x4AE0 6124
CM_CLKSEL_HDMI_TIMER	RW	32	0x0000 0028	0x4AE0 6128
CM_CLKSEL_MCASP_SYS	RW	32	0x0000 002C	0x4AE0 612C
CM_CLKSEL_MLBP_MCASP	RW	32	0x0000 0030	0x4AE0 6130
CM_CLKSEL_MLB_MCASP	RW	32	0x0000 0034	0x4AE0 6134
CM_CLKSEL_PER_ABE_X1_GFCLK_MCASP_AUX	RW	32	0x0000 0038	0x4AE0 6138
CM_CLKSEL_SYS_CLK1_32K	RW	32	0x0000 0040	0x4AE0 6140
CM_CLKSEL_TIMER_SYS	RW	32	0x0000 0044	0x4AE0 6144
CM_CLKSEL_VIDEO1_MCASP_AUX	RW	32	0x0000 0048	0x4AE0 6148
CM_CLKSEL_VIDEO1_TIMER	RW	32	0x0000 004C	0x4AE0 614C
CM_CLKSEL_VIDEO2_MCASP_AUX	RW	32	0x0000 0050	0x4AE0 6150
CM_CLKSEL_VIDEO2_TIMER	RW	32	0x0000 0054	0x4AE0 6154
CM_CLKSEL_CLKOUTMUX0	RW	32	0x0000 0058	0x4AE0 6158
CM_CLKSEL_CLKOUTMUX1	RW	32	0x0000 005C	0x4AE0 615C
CM_CLKSEL_CLKOUTMUX2	RW	32	0x0000 0060	0x4AE0 6160
CM_CLKSEL_HDMI_PLL_SYS	RW	32	0x0000 0064	0x4AE0 6164
CM_CLKSEL_VIDEO1_PLL_SYS	RW	32	0x0000 0068	0x4AE0 6168
CM_CLKSEL_VIDEO2_PLL_SYS	RW	32	0x0000 006C	0x4AE0 616C
CM_CLKSEL_ABE_CLK_DIV	RW	32	0x0000 0070	0x4AE0 6170
CM_CLKSEL_ABE_GICLK_DIV	RW	32	0x0000 0074	0x4AE0 6174
CM_CLKSEL_AESS_FCLK_DIV	RW	32	0x0000 0078	0x4AE0 6178
CM_CLKSEL_EVE_CLK	RW	32	0x0000 0080	0x4AE0 6180
CM_CLKSEL_USB_OTG_CLK_CLKOUTMUX	RW	32	0x0000 0084	0x4AE0 6184

Table 3-999. CKGEN_PRM Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	CKGEN_PRM Physical Address
CM_CLKSEL_CORE_DPLL_OUT_CLK_CLKOUTMUX	RW	32	0x0000 0088	0x4AE0 6188
CM_CLKSEL_DSP_GFCLK_CLKOUTMUX	RW	32	0x0000 008C	0x4AE0 618C
CM_CLKSEL_EMIF_PHY_GCLK_CLKOUTMUX	RW	32	0x0000 0090	0x4AE0 6190
CM_CLKSEL_EMU_CLK_CLKOUTMUX	RW	32	0x0000 0094	0x4AE0 6194
CM_CLKSEL_FUNC_96M_AON_CLK_CLKOUTMUX	RW	32	0x0000 0098	0x4AE0 6198
CM_CLKSEL_GMAC_250M_CLK_CLKOUTMUX	RW	32	0x0000 009C	0x4AE0 619C
CM_CLKSEL_GPU_GCLK_CLKOUTMUX	RW	32	0x0000 00A0	0x4AE0 61A0
CM_CLKSEL_HDMI_CLK_CLKOUTMUX	RW	32	0x0000 00A4	0x4AE0 61A4
CM_CLKSEL_IVA_GCLK_CLKOUTMUX	RW	32	0x0000 00A8	0x4AE0 61A8
CM_CLKSEL_L3INIT_480M_GFCLK_CLKOUTMUX	RW	32	0x0000 00AC	0x4AE0 61AC
CM_CLKSEL_MPU_GCLK_CLKOUTMUX	RW	32	0x0000 00B0	0x4AE0 61B0
CM_CLKSEL_PCIE1_CLK_CLKOUTMUX	RW	32	0x0000 00B4	0x4AE0 61B4
CM_CLKSEL_PCIE2_CLK_CLKOUTMUX	RW	32	0x0000 00B8	0x4AE0 61B8
CM_CLKSEL_PER_ABE_X1_CLK_CLKOUTMUX	RW	32	0x0000 00BC	0x4AE0 61BC
CM_CLKSEL_SATA_CLK_CLKOUTMUX	RW	32	0x0000 00C0	0x4AE0 61C0
CM_CLKSEL_SECURE_32K_CLK_CLKOUTMUX	RW	32	0x0000 00C4	0x4AE0 61C4
CM_CLKSEL_SYS_CLK1_CLKOUTMUX	RW	32	0x0000 00C8	0x4AE0 61C8
CM_CLKSEL_SYS_CLK2_CLKOUTMUX	RW	32	0x0000 00CC	0x4AE0 61CC
CM_CLKSEL_VIDEO1_CLK_CLKOUTMUX	RW	32	0x0000 00D0	0x4AE0 61D0
CM_CLKSEL_VIDEO2_CLK_CLKOUTMUX	RW	32	0x0000 00D4	0x4AE0 61D4
CM_CLKSEL_ABE_LP_CLK	RW	32	0x0000 00D8	0x4AE0 61D8
CM_CLKSEL_ADC_GFCLK	RW	32	0x0000 00DC	0x4AE0 61DC
CM_CLKSEL_EVE_GFCLK_CLKOUTMUX	RW	32	0x0000 00E0	0x4AE0 61E0

3.12.31.2 CKGEN_PRM Register Description

Table 3-1000. CM_CLKSEL_SYSCLK1

Address Offset	0x0000 0000		
Physical Address	0x4AE0 6100	**Instance**	CKGEN_PRM
Description	Select the SYS CLK for SYSCLK1_32K_CLK. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															CLKSEL

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	CLKSEL	Selects the divider value 0x0: Select SYS_CLK divided by 6 0x1: Select SYS_CLK divided by 10	RW	0x0

Table 3-1001. Register Call Summary for Register CM_CLKSEL_SYSCLK1

PRCM Register Manual
- CKGEN_PRM Register Summary: [0]

Table 3-1002. CM_CLKSEL_WKUPAON

Address Offset	0x0000 0008		
Physical Address	0x4AE0 6108	**Instance**	CKGEN_PRM
Description	Control the functional clock source of WKUPAON, PRM and Smart Reflex functional clock.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															CLKSEL

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	CLKSEL	Select the clock source for WKUPAON_ICLK clock 0x0: Selects SYS_CLK for WKUPAON_ICLK 0x1: Selects ABE_LP_CLK for WKUPAON_ICLK	RW	0x0

Table 3-1003. Register Call Summary for Register CM_CLKSEL_WKUPAON

Clock Management Functional Description
- PRM Clock Source: [0]

PRCM Register Manual
- CKGEN_PRM Register Summary: [1]

Table 3-1004. CM_CLKSEL_ABE_PLL_REF

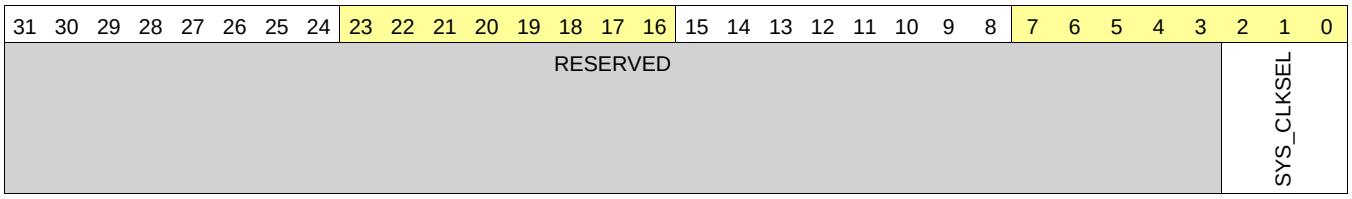
Address Offset	0x0000 000C		
Physical Address	0x4AE0 610C	**Instance**	CKGEN_PRM
Description	Control the source of the reference clock for DPLL_ABE		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															CLKSEL

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	CLKSEL	Select the source for the DPLL_ABE reference clock. 0x0: Selects SYS_CLK 0x1: Selects SYS_CLK1_32K_CLK	RW	0x0

Table 3-1005. CM_CLKSEL_SYS

Address Offset	0x0000 0010		
Physical Address	0x4AE0 6110	**Instance**	CKGEN_PRM
Description	Software (ROM Code) sets the SYS_CLK configuration corresponding to the frequency of SYS_CLK. [warm reset insensitive]		
Type	RW		



Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0
2:0	SYS_CLKSEL	System clock input selection. 0x0: Uninitialized 0x1: Input clock is 12 MHz 0x2: Input clock is 20 MHz 0x3: Input clock is 16.8 MHz 0x4: Input clock is 19.2 MHz 0x5: Input clock is 26 MHz 0x6: Input clock is 27 MHz 0x7: Input clock is 38.4 MHz	RW	0x0

Table 3-1006. Register Call Summary for Register CM_CLKSEL_SYS

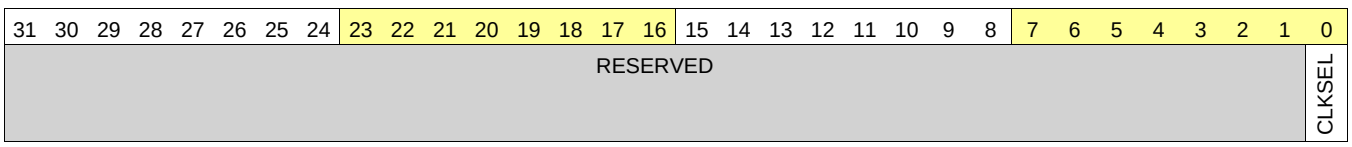
Clock Management Functional Description

PRCM Register Manual

- Not Supported Functionality (Registers and Bitfields): [5][6][7][8]
- CKGEN_PRM Register Summary: [9]

Table 3-1007. CM_CLKSEL_ABE_PLL_BYPAS

Address Offset	0x0000 0014		
Physical Address	0x4AE0 6114	**Instance**	CKGEN_PRM
Description	Control the source of the bypass clock for DPLL_ABE		
Type	RW		



Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	CLKSEL	Control the source of the bypass clock for DPLL_ABE 0x0: Selects SYS_CLK for ABE_DPLL_BYPASS_CLK 0x1: Selects SYS_CLK1_32K_CLK for ABE_DPLL_BYPASS_CLK	RW	0x0

Table 3-1008. CM_CLKSEL_ABE_PLL_SYS

Address Offset	0x0000 0018		
Physical Address	0x4AE0 6118	**Instance**	CKGEN_PRM
Description	Control the source of the SYS clock for DPLL_ABE		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															CLKSEL

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	CLKSEL	Select the SYS clock for the DPLL_ABE reference and bypass clock. 0x0: Selects SYS_CLK1 0x1: Selects SYS_CLK2	RW	0x0

Table 3-1009. CM_CLKSEL_ABE_24M

Address Offset	0x0000 001C		
Physical Address	0x4AE0 611C	**Instance**	CKGEN_PRM
Description	Select the ABE_24M_FCLK for TIMERS subsystems. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															CLKSEL

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	CLKSEL	Selects the divider value 0x0: Select SYS_CLK divided by 8 0x1: Select SYS_CLK divided by 16	RW	0x0

Table 3-1010. Register Call Summary for Register CM_CLKSEL_ABE_24M

Clock Management Functional Description
- CM_CORE_AON Clock Generator: [0]

PRCM Register Manual
- CKGEN_PRM Register Summary: [1]

Table 3-1011. CM_CLKSEL_ABE_SYS

Address Offset	0x0000 0020		
Physical Address	0x4AE0 6120	**Instance**	CKGEN_PRM
Description	Select the ABE_SYS_CLK. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															CLKSEL

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	CLKSEL	Selects the divider value 0x0: Select SYS_CLK1 divided by 1 0x1: Select SYS_CLK1 divided by 2	RW	0x0

Table 3-1012. Register Call Summary for Register CM_CLKSEL_ABE_SYS

Clock Management Functional Description • PRM Clock Source: [0]
PRCM Register Manual • CKGEN_PRM Register Summary: [1]

Table 3-1013. CM_CLKSEL_HDMI_MCASP_AUX

Address Offset	0x0000 0024		
Physical Address	0x4AE0 6124	**Instance**	CKGEN_PRM
Description	Select divider for L4_ICLK(SR2.0)/SYS_CLK1(SR1.0) for McASP modules. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														CLKSEL	

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0
2:0	CLKSEL	Selects the divider value 0x0: Select clock divided by 1 0x1: Select clock divided by 2 0x2: Select clock divided by 4 0x3: Select clock divided by 8 0x4: Select clock divided by 16 0x5: RESERVED 0x6: RESERVED 0x7: RESERVED	RW	0x0

Table 3-1014. Register Call Summary for Register CM_CLKSEL_HDMI_MCASP_AUX

Clock Management Functional Description • CM_CORE_AON_MCASP1 Overview: [0]
PRCM Register Manual • IPU_CM_CORE_AON Register Description: [1] • L4PER_CM_CORE Register Description: [2][3][4][5] • CKGEN_PRM Register Summary: [6]

Table 3-1015. CM_CLKSEL_HDMI_TIMER

Address Offset	0x0000 0028		
Physical Address	0x4AE0 6128	**Instance**	CKGEN_PRM
Description	Select the SYS_CLK1 for TIMER subsystems. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														CLKSEL	

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0
2:0	CLKSEL	Selects the divider value 0x0: Select SYS_CLK1 divided by 1 0x1: Select SYS_CLK1 divided by 2 0x2: Select SYS_CLK1 divided by 4 0x3: Select SYS_CLK1 divided by 8 0x4: Select SYS_CLK1 divided by 16 0x5: Select SYS_CLK1 divided by 22 0x6: Select SYS_CLK1 divided by 32 0x7: Reserved	RW	0x0

Table 3-1016. Register Call Summary for Register CM_CLKSEL_HDMI_TIMER

Clock Management Functional Description
- CM_CORE_AON_TIMER Overview: [0]

PRCM Register Manual
- IPU_CM_CORE_AON Register Description: [1][2][3][4]
- L4PER_CM_CORE Register Description: [5][6][7]
- CKGEN_PRM Register Summary: [8]
- WKUPAON_CM Register Description: [9]

Table 3-1017. CM_CLKSEL_MCASP_SYS

Address Offset	0x0000 002C		
Physical Address	0x4AE0 612C	**Instance**	CKGEN_PRM
Description	Select the SYS CLK for ABE_24M_FCLK. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															CLKSEL

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	CLKSEL	Selects the divider value 0x0: Select SYS_CLK divided by 8 0x1: Select SYS_CLK divided by 16	RW	0x0

Table 3-1018. Register Call Summary for Register CM_CLKSEL_MCASP_SYS

PRCM Register Manual • CKGEN_PRM Register Summary: [0]

Table 3-1019. CM_CLKSEL_MLBP_MCASP

Address Offset	0x0000 0030		
Physical Address	0x4AE0 6130	**Instance**	CKGEN_PRM
Description	Select the MLBP_CLK for McASP subsystems. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														CLKSEL	

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0
2:0	CLKSEL	Selects the divider value 0x0: Select SYS_CLK1 divided by 1 0x1: Select SYS_CLK1 divided by 2 0x2: Select SYS_CLK1 divided by 4 0x3: Select SYS_CLK1 divided by 8 0x4: Select SYS_CLK1 divided by 16 0x5: RESERVED 0x6: RESERVED 0x7: RESERVED	RW	0x0

Table 3-1020. Register Call Summary for Register CM_CLKSEL_MLBP_MCASP

Clock Management Functional Description • CM_CORE_AON_MCASP1 Overview: [0]
PRCM Register Manual • IPU_CM_CORE_AON Register Description: [1][2] • L4PER_CM_CORE Register Description: [3][4][5][6] • CKGEN_PRM Register Summary: [7]

Table 3-1021. CM_CLKSEL_MLB_MCASP

Address Offset	0x0000 0034		
Physical Address	0x4AE0 6134	**Instance**	CKGEN_PRM
Description	Select the SYS_CLK1 for McASP. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														CLKSEL	

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0
2:0	CLKSEL	Selects the divider value 0x0: Select SYS_CLK1 divided by 1 0x1: Select SYS_CLK1 divided by 2 0x2: Select SYS_CLK1 divided by 4 0x3: Select SYS_CLK1 divided by 8 0x4: Select SYS_CLK1 divided by 16 0x5: RESERVED 0x6: RESERVED 0x7: RESERVED	RW	0x0

Table 3-1022. Register Call Summary for Register CM_CLKSEL_MLB_MCASP

Clock Management Functional Description • CM_CORE_AON_MCASP1 Overview: [0]
PRCM Register Manual • IPU_CM_CORE_AON Register Description: [1][2] • L4PER_CM_CORE Register Description: [3][4][5][6] • CKGEN_PRM Register Summary: [7]

Table 3-1023. CM_CLKSEL_PER_ABE_X1_GFCLK_MCASP_AUX

Address Offset	0x0000 0038		
Physical Address	0x4AE0 6138	**Instance**	CKGEN_PRM
Description	Select the SYS_CLK1 for McASP modules. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																													CLKSEL		

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0
2:0	CLKSEL	Selects the divider value 0x0: Select SYS_CLK1 divided by 1 0x1: Select SYS_CLK1 divided by 2 0x2: Select SYS_CLK1 divided by 4 0x3: Select SYS_CLK1 divided by 8 0x4: Select SYS_CLK1 divided by 16 0x5: RESERVED 0x6: RESERVED 0x7: RESERVED	RW	0x0

Table 3-1024. Register Call Summary for Register CM_CLKSEL_PER_ABE_X1_GFCLK_MCASP_AUX

Clock Management Functional Description • CM_CORE_AON_MCASP1 Overview: [0]
PRCM Register Manual • IPU_CM_CORE_AON Register Description: [1] • L4PER_CM_CORE Register Description: [2][3][4][5] • CKGEN_PRM Register Summary: [6]

Table 3-1025. CM_CLKSEL_SYS_CLK1_32K

Address Offset	0x0000 0040		
Physical Address	0x4AE0 6140	**Instance**	CKGEN_PRM
Description	Control the source of the FUNC_32K_CLK clock for GPIO, WD_TIMER, KBD, etc.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															CLKSEL

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	CLKSEL	Select the FUNC_32K_CLK clock 0x0: Selects SYS_CLK1 divided by 610 0x1: RESERVED	RW	0x0

Table 3-1026. Register Call Summary for Register CM_CLKSEL_SYS_CLK1_32K

PRCM Register Manual
- CKGEN_PRM Register Summary: [0]

Table 3-1027. CM_CLKSEL_TIMER_SYS

Address Offset	0x0000 0044		
Physical Address	0x4AE0 6144	**Instance**	CKGEN_PRM
Description	Select the TIMER_SYS_CLK for TIMER modules. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															CLKSEL

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	CLKSEL	Selects the divider value 0x0: Select SYS_CLK1 divided by 1 0x1: Select SYS_CLK1 divided by 2	RW	0x0

Table 3-1028. Register Call Summary for Register CM_CLKSEL_TIMER_SYS

Clock Management Functional Description
- PRM Clock Source: [0]

PRCM Register Manual
- CKGEN_PRM Register Summary: [1]

Table 3-1029. CM_CLKSEL_VIDEO1_MCASP_AUX

Address Offset	0x0000 0048		
Physical Address	0x4AE0 6148	**Instance**	CKGEN_PRM
Description	Select the SYS_CLK1 for McASP modules. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														CLKSEL	

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0
2:0	CLKSEL	Selects the divider value	RW	0x0
		0x0: Select SYS_CLK1 divided by 1		
		0x1: Select SYS_CLK1 divided by 2		
		0x2: Select SYS_CLK1 divided by 4		
		0x3: Select SYS_CLK1 divided by 8		
		0x4: Select SYS_CLK1 divided by 16		
		0x5: RESERVED		
		0x6: RESERVED		
		0x7: RESERVED		

Table 3-1030. Register Call Summary for Register CM_CLKSEL_VIDEO1_MCASP_AUX

Clock Management Functional Description
- CM_CORE_AON_MCASP1 Overview: [0]

PRCM Register Manual
- IPU_CM_CORE_AON Register Description: [1]
- L4PER_CM_CORE Register Description: [2][3][4][5]
- CKGEN_PRM Register Summary: [6]

Table 3-1031. CM_CLKSEL_VIDEO1_TIMER

Address Offset	0x0000 004C		
Physical Address	0x4AE0 614C	**Instance**	CKGEN_PRM
Description	Select the SYS_CLK1 for TIMER subsystems. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														CLKSEL	

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0
2:0	CLKSEL	Selects the divider value 0x0: Select SYS_CLK1 divided by 1 0x1: Select SYS_CLK1 divided by 2 0x2: Select SYS_CLK1 divided by 4 0x3: Select SYS_CLK1 divided by 8 0x4: Select SYS_CLK1 divided by 16 0x5: Select SYS_CLK1 divided by 22 0x6: Select SYS_CLK1 divided by 32 0x7: Reserved	RW	0x0

Table 3-1032. Register Call Summary for Register CM_CLKSEL_VIDEO1_TIMER

Clock Management Functional Description • CM_CORE_AON_TIMER Overview: [0]
PRCM Register Manual • IPU_CM_CORE_AON Register Description: [1][2][3][4] • L4PER_CM_CORE Register Description: [5][6][7] • CKGEN_PRM Register Summary: [8] • WKUPAON_CM Register Description: [9]

Table 3-1033. CM_CLKSEL_VIDEO2_MCASP_AUX

Address Offset	0x0000 0050		
Physical Address	0x4AE0 6150	**Instance**	CKGEN_PRM
Description	Select the SYS_CLK1 for McASP modules. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														CLKSEL	

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0
2:0	CLKSEL	Selects the divider value 0x0: Select SYS_CLK1 divided by 1 0x1: Select SYS_CLK1 divided by 2 0x2: Select SYS_CLK1 divided by 4 0x3: Select SYS_CLK1 divided by 8 0x4: Select SYS_CLK1 divided by 16 0x5: RESERVED 0x6: RESERVED 0x7: RESERVED	RW	0x0

Table 3-1034. Register Call Summary for Register CM_CLKSEL_VIDEO2_MCASP_AUX

Clock Management Functional Description • CM_CORE_AON_MCASP1 Overview: [0]
PRCM Register Manual • IPU_CM_CORE_AON Register Description: [1] • L4PER_CM_CORE Register Description: [2][3][4][5] • CKGEN_PRM Register Summary: [6]

Table 3-1035. CM_CLKSEL_VIDEO2_TIMER

Address Offset	0x0000 0054		
Physical Address	0x4AE0 6154	**Instance**	CKGEN_PRM
Description	Select the SYS_CLK1 for TIMER subsystems. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														CLKSEL	

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0
2:0	CLKSEL	Selects the divider value	RW	0x0
		0x0: Select SYS_CLK1 divided by 1		
		0x1: Select SYS_CLK1 divided by 2		
		0x2: Select SYS_CLK1 divided by 4		
		0x3: Select SYS_CLK1 divided by 8		
		0x4: Select SYS_CLK1 divided by 16		
		0x5: Select SYS_CLK1 divided by 22		
		0x6: Select SYS_CLK1 divided by 32		
		0x7: Reserved		

Table 3-1036. Register Call Summary for Register CM_CLKSEL_VIDEO2_TIMER

Clock Management Functional Description
- CM_CORE_AON_TIMER Overview: [0]

PRCM Register Manual
- IPU_CM_CORE_AON Register Description: [1][2][3][4]
- L4PER_CM_CORE Register Description: [5][6][7]
- CKGEN_PRM Register Summary: [8]
- WKUPAON_CM Register Description: [9]

Table 3-1037. CM_CLKSEL_CLKOUTMUX0

Address Offset	0x0000 0058		
Physical Address	0x4AE0 6158	**Instance**	CKGEN_PRM
Description	Control the source of the CLKOUTMUX0		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														CLKSEL	

Bits	Field Name	Description	Type	Reset
31:5	RESERVED		R	0x0
4:0	CLKSEL	Select the source of the CLKOUTMUX0. 0x0: Selects divided version of SYS_CLK1. See CM_CLKSEL_SYS_CLK1_CLKOUTMUX 0x1: Selects divided version of EVE_GFCLK. See CM_CLKSEL_EVE_GFCLK_CLKOUTMUX 0x2: Selects divided version of SYS_CLK1. See CM_CLKSEL_PER_ABE_X1_CLK_CLKOUTMUX 0x3: Selects divided version of SYS_CLK1. See CM_CLKSEL_MPU_GCLK_CLKOUTMUX 0x4: Selects divided version of DSP_GCLK. See CM_CLKSEL_DSP_GFCLK_CLKOUTMUX 0x5: Selects divided version of SYS_CLK1. See CM_CLKSEL_IVA_GCLK_CLKOUTMUX 0x6: Selects divided version of SYS_CLK1. See CM_CLKSEL_GPU_GCLK_CLKOUTMUX 0x7: Selects divided version of CORE_DPLL_OUT_CLK. See CM_CLKSEL_CORE_DPLL_OUT_CLK_CLKOUTMUX 0x8: Selects divided version of EMIF_PHY_GCLK. See CM_CLKSEL_EMIF_PHY_GCLK_CLKOUTMUX 0x9: Selects divided version of GMAC_250M_CLK. See CM_CLKSEL_GMAC_250M_CLK_CLKOUTMUX 0xA: Selects divided version of SYS_CLK1. See CM_CLKSEL_VIDEO2_CLK_CLKOUTMUX 0xB: Selects divided version of SYS_CLK1. See CM_CLKSEL_VIDEO1_CLK_CLKOUTMUX 0xC: Selects divided version of SYS_CLK1. See CM_CLKSEL_HDMI_CLK_CLKOUTMUX 0xD: Selects divided version of FUNC_96M_AON_CLK. See CM_CLKSEL_FUNC_96M_AON_CLK_CLKOUTMUX 0xE: Selects divided version of SYS_CLK1. See CM_CLKSEL_L3INIT_480M_GFCLK_CLKOUTMUX 0xF: Selects divided version of SYS_CLK1. See CM_CLKSEL_USB_OTG_CLK_CLKOUTMUX 0x10: Selects divided version of SYS_CLK1. See CM_CLKSEL_SATA_CLK_CLKOUTMUX 0x11: Selects divided version of SYS_CLK1. See CM_CLKSEL_PCIE2_CLK_CLKOUTMUX 0x12: Selects divided version of SYS_CLK1. See CM_CLKSEL_PCIE1_CLK_CLKOUTMUX 0x13: Selects divided version of EMU_GCLK. See CM_CLKSEL_EMU_CLK_CLKOUTMUX 0x14: Selects divided version of RCOSC_32K_GCLK. See CM_CLKSEL_SECURE_32K_CLK_CLKOUTMUX Note: The RCOSC_32K_CLK clock, provided by the On-die 32K RC OSC is not accurate 32KHz clock. The frequency may vary with temperature and silicon characteristics. 0x15-0x1F: RESERVED	RW	0x0

Table 3-1038. Register Call Summary for Register CM_CLKSEL_CLKOUTMUX0

Clock Management Functional Description • CM_CORE_AON_CLKOUTMUX Overview: [0]
PRCM Register Manual • CKGEN_PRM Register Summary: [1]

Table 3-1039. CM_CLKSEL_CLKOUTMUX1

Address Offset	0x0000 005C		
Physical Address	0x4AE0 615C	**Instance**	CKGEN_PRM
Description	Control the source of the CLKOUTMUX1		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																											CLKSEL			

Bits	Field Name	Description	Type	Reset
31:5	RESERVED		R	0x0
4:0	CLKSEL	Select the source of the CLKOUTMUX1 0x0: Selects divided version of SYS_CLK1. See CM_CLKSEL_SYS_CLK1_CLKOUTMUX 0x1: Selects divided version of SYS_CLK2. See CM_CLKSEL_SYS_CLK2_CLKOUTMUX 0x2: Selects divided version of SYS_CLK1. See CM_CLKSEL_PER_ABE_X1_CLK_CLKOUTMUX 0x3: Selects divided version of SYS_CLK1. See CM_CLKSEL_MPU_GCLK_CLKOUTMUX 0x4: Selects divided version of DSP_GCLK. See CM_CLKSEL_DSP_GFCLK_CLKOUTMUX 0x5: Selects divided version of SYS_CLK1. See CM_CLKSEL_IVA_GCLK_CLKOUTMUX 0x6: Selects divided version of SYS_CLK1. See CM_CLKSEL_GPU_GCLK_CLKOUTMUX 0x7: Selects divided version of CORE_DPLL_OUT_CLK. See CM_CLKSEL_CORE_DPLL_OUT_CLK_CLKOUTMUX 0x8: Selects divided version of EMIF_PHY_GCLK. See CM_CLKSEL_EMIF_PHY_GCLK_CLKOUTMUX 0x9: Selects divided version of GMAC_250M_CLK. See CM_CLKSEL_GMAC_250M_CLK_CLKOUTMUX 0xA: Selects divided version of SYS_CLK1. See CM_CLKSEL_VIDEO2_CLK_CLKOUTMUX 0xB: Selects divided version of SYS_CLK1. See CM_CLKSEL_VIDEO1_CLK_CLKOUTMUX 0xC: Selects divided version of SYS_CLK1. See CM_CLKSEL_HDMI_CLK_CLKOUTMUX 0xD: Selects divided version of FUNC_96M_AON_CLK. See CM_CLKSEL_FUNC_96M_AON_CLK_CLKOUTMUX 0xE: Selects divided version of SYS_CLK1. See CM_CLKSEL_L3INIT_480M_GFCLK_CLKOUTMUX 0xF: Selects divided version of SYS_CLK1. See CM_CLKSEL_USB_OTG_CLK_CLKOUTMUX 0x10: Selects divided version of SYS_CLK1. See CM_CLKSEL_SATA_CLK_CLKOUTMUX 0x11: Selects divided version of SYS_CLK1. See CM_CLKSEL_PCIE2_CLK_CLKOUTMUX 0x12: Selects divided version of SYS_CLK1. See CM_CLKSEL_PCIE1_CLK_CLKOUTMUX 0x13: Selects divided version of EMU_GCLK. See CM_CLKSEL_EMU_CLK_CLKOUTMUX 0x14: Selects divided version of RCOSC_32K_GCLK. See CM_CLKSEL_SECURE_32K_CLK_CLKOUTMUX Note: The RCOSC_32K_CLK clock, provided by the On-die 32K RC OSC is not accurate 32KHz clock. The frequency may vary with temperature and silicon characteristics. 0x15: Selects EVE_GFCLK. See CM_CLKSEL_EVE_GFCLK_CLKOUTMUX 0x16-0x1F: RESERVED	RW	0x0

Table 3-1040. Register Call Summary for Register CM_CLKSEL_CLKOUTMUX1

Clock Management Functional Description • CM_CORE_AON_CLKOUTMUX Overview: [0]
PRCM Register Manual • CKGEN_PRM Register Summary: [1]

Table 3-1041. CM_CLKSEL_CLKOUTMUX2

Address Offset	0x0000 0060		
Physical Address	0x4AE0 6160	**Instance**	CKGEN_PRM
Description	Control the source of the CLKOUTMUX2		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																											CLKSEL			

Bits	Field Name	Description	Type	Reset
31:5	RESERVED		R	0x0
4:0	CLKSEL	Select the source of the CLKOUTMUX2 0x0: Selects divided version of SYS_CLK1. See CM_CLKSEL_SYS_CLK1_CLKOUTMUX 0x1: Selects divided version of SYS_CLK2. See CM_CLKSEL_SYS_CLK2_CLKOUTMUX 0x2: Selects divided version of SYS_CLK1. See CM_CLKSEL_PER_ABE_X1_CLK_CLKOUTMUX 0x3: Selects divided version of SYS_CLK1. See CM_CLKSEL_MPU_GCLK_CLKOUTMUX 0x4: Selects divided version of DSP_GCLK. See CM_CLKSEL_DSP_GFCLK_CLKOUTMUX 0x5: Selects divided version of SYS_CLK1. See CM_CLKSEL_IVA_GCLK_CLKOUTMUX 0x6: Selects divided version of SYS_CLK1. See CM_CLKSEL_GPU_GCLK_CLKOUTMUX 0x7: Selects divided version of CORE_DPLL_OUT_CLK. See CM_CLKSEL_CORE_DPLL_OUT_CLK_CLKOUTMUX 0x8: Selects divided version of EMIF_PHY_GCLK. See CM_CLKSEL_EMIF_PHY_GCLK_CLKOUTMUX 0x9: Selects divided version of GMAC_250M_CLK. See CM_CLKSEL_GMAC_250M_CLK_CLKOUTMUX 0xA: Selects divided version of SYS_CLK1. See CM_CLKSEL_VIDEO2_CLK_CLKOUTMUX 0xB: Selects divided version of SYS_CLK1. See CM_CLKSEL_VIDEO1_CLK_CLKOUTMUX 0xC: Selects divided version of SYS_CLK1. See CM_CLKSEL_HDMI_CLK_CLKOUTMUX 0xD: Selects divided version of FUNC_96M_AON_CLK. See CM_CLKSEL_FUNC_96M_AON_CLK_CLKOUTMUX 0xE: Selects divided version of SYS_CLK1. See CM_CLKSEL_L3INIT_480M_GFCLK_CLKOUTMUX 0xF: Selects divided version of SYS_CLK1. See CM_CLKSEL_USB_OTG_CLK_CLKOUTMUX 0x10: Selects divided version of SYS_CLK1. See CM_CLKSEL_SATA_CLK_CLKOUTMUX 0x11: Selects divided version of SYS_CLK1. See CM_CLKSEL_PCIE2_CLK_CLKOUTMUX 0x12: Selects divided version of SYS_CLK1. See CM_CLKSEL_PCIE1_CLK_CLKOUTMUX 0x13: Selects divided version of EMU_GCLK. See CM_CLKSEL_EMU_CLK_CLKOUTMUX 0x14: Selects divided version of RCOSC_32K_GCLK. See CM_CLKSEL_SECURE_32K_CLK_CLKOUTMUX Note: The RCOSC_32K_CLK clock, provided by the On-die 32K RC OSC is not accurate 32KHz clock. The frequency may vary with temperature and silicon characteristics. 0x15: Selects EVE_GFCLK. See CM_CLKSEL_EVE_GFCLK_CLKOUTMUX 0x16-0x1F: RESERVED	RW	0x0

Table 3-1042. Register Call Summary for Register CM_CLKSEL_CLKOUTMUX2

Clock Management Functional Description • CM_CORE_AON_CLKOUTMUX Overview: [0]
PRCM Register Manual • CKGEN_PRM Register Summary: [1]

Table 3-1043. CM_CLKSEL_HDMI_PLL_SYS

Address Offset	0x0000 0064		
Physical Address	0x4AE0 6164	**Instance**	CKGEN_PRM
Description	Control the source of the SYS clock for DPLL_HDMI		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															CLKSEL

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	CLKSEL	Select the SYS clock for the DPLL_HDMI 0x0: Selects SYS_CLK1 0x1: Selects SYS_CLK2	RW	0x0

Table 3-1044. CM_CLKSEL_VIDEO1_PLL_SYS

Address Offset	0x0000 0068		
Physical Address	0x4AE0 6168	**Instance**	CKGEN_PRM
Description	Control the source of the SYS clock for DPLL_VIDEO1		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															CLKSEL

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	CLKSEL	Select the SYS clock for the DPLL_VIDEO1. 0x0: Selects SYS_CLK1 0x1: Selects SYS_CLK2	RW	0x0

Table 3-1045. CM_CLKSEL_VIDEO2_PLL_SYS

Address Offset	0x0000 006C		
Physical Address	0x4AE0 616C	**Instance**	CKGEN_PRM
Description	Control the source of the SYS clock for DPLL_VIDEO1		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															CLKSEL

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	CLKSEL	Select the SYS clock for the DPLL_VIDEO2. 0x0: Selects SYS_CLK1 0x1: Selects SYS_CLK2	RW	0x0

Table 3-1046. CM_CLKSEL_ABE_CLK_DIV

Address Offset	0x0000 0070		
Physical Address	0x4AE0 6170	**Instance**	CKGEN_PRM
Description	Select the ABE_CLK. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														CLKSEL	

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0
2:0	CLKSEL	Selects the divider value 0x0: Select CLK divided by 1 0x1: Select CLK divided by 2 0x2: Select CLK divided by 4 0x3: Select CLK divided by 8 0x4: Select CLK divided by 16 0x5: RESERVED 0x6: RESERVED 0x7: RESERVED	RW	0x0

Table 3-1047. Register Call Summary for Register CM_CLKSEL_ABE_CLK_DIV

Clock Management Functional Description
- CM_CORE_AON Clock Generator: [0]

PRCM Register Manual
- CKGEN_PRM Register Summary: [1]

Table 3-1048. CM_CLKSEL_ABE_GICLK_DIV

Address Offset	0x0000 0074		
Physical Address	0x4AE0 6174	**Instance**	CKGEN_PRM
Description	Select the ABE_GICLK. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															CLKSEL

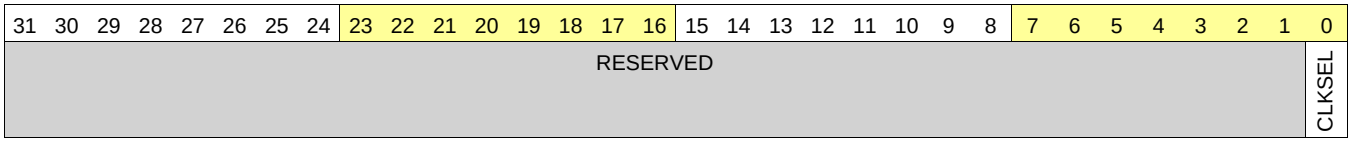
Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	CLKSEL	Selects the divider value 0x0: Select CLK divided by 1 0x1: Select CLK divided by 2	RW	0x0

Table 3-1049. Register Call Summary for Register CM_CLKSEL_ABE_GICLK_DIV

Clock Management Functional Description • CM_CORE_AON Clock Generator: [0]
PRCM Register Manual • CKGEN_PRM Register Summary: [1]

Table 3-1050. CM_CLKSEL_AESS_FCLK_DIV

Address Offset	0x0000 0078		
Physical Address	0x4AE0 6178	**Instance**	CKGEN_PRM
Description	Select the AESS_FCLK. [warm reset insensitive]		
Type	RW		



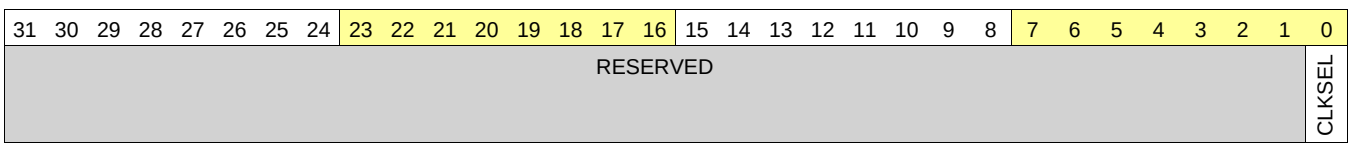
Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	CLKSEL	Selects the divider value 0x0: Select CLK divided by 1 0x1: Select CLK divided by 2	RW	0x0

Table 3-1051. Register Call Summary for Register CM_CLKSEL_AESS_FCLK_DIV

Clock Management Functional Description • CM_CORE_AON Clock Generator: [0]
PRCM Register Manual • CKGEN_PRM Register Summary: [1]

Table 3-1052. CM_CLKSEL_EVE_CLK

Address Offset	0x0000 0080		
Physical Address	0x4AE0 6180	**Instance**	CKGEN_PRM
Description	Controls the source of the EVE_CLK for EVE		
Type	RW		



Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	CLKSEL	Selects the EVE_CLK for EVE 0x0: Selects clock from DPLL_EVE_VID_DSP 0x1: Selects clock from DPLL_CORE or DPLL_GMAC_DSP (selected in the control module).	RW	0x0

Table 3-1053. Register Call Summary for Register CM_CLKSEL_EVE_CLK

Clock Management Functional Description • CM_CORE_AON Clock Generator: [0]
PRCM Register Manual • CKGEN_PRM Register Summary: [1]

Table 3-1054. CM_CLKSEL_USB_OTG_CLK_CLKOUTMUX

Address Offset	0x0000 0084		
Physical Address	0x4AE0 6184	**Instance**	CKGEN_PRM
Description	Select the USB_OTG_CLK. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														CLKSEL	

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0
2:0	CLKSEL	Selects the divider value 0x0: Select CLK divided by 1 0x1: Select CLK divided by 2 0x2: Select CLK divided by 4 0x3: Select CLK divided by 8 0x4: Select CLK divided by 16 0x5: SELECT CLK divided by 32 0x6: RESERVED 0x7: RESERVED	RW	0x0

Table 3-1055. Register Call Summary for Register CM_CLKSEL_USB_OTG_CLK_CLKOUTMUX

Clock Management Functional Description • CM_CORE_AON_CLKOUTMUX Overview: [0]
PRCM Register Manual • CKGEN_PRM Register Summary: [1] • CKGEN_PRM Register Description: [2][3][4]

Table 3-1056. CM_CLKSEL_CORE_DPLL_OUT_CLK_CLKOUTMUX

Address Offset	0x0000 0088		
Physical Address	0x4AE0 6188	**Instance**	CKGEN_PRM
Description	Select the CORE_DPLL_OUT_CLK. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																													CLKSEL		

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0
2:0	CLKSEL	Selects the divider value 0x0: Select CLK divided by 1 0x1: Select CLK divided by 2 0x2: Select CLK divided by 4 0x3: Select CLK divided by 8 0x4: Select CLK divided by 16 0x5: Select CLK divided by 32 0x6: RESERVED 0x7: RESERVED	RW	0x0

Table 3-1057. Register Call Summary for Register CM_CLKSEL_CORE_DPLL_OUT_CLK_CLKOUTMUX

Clock Management Functional Description
- CM_CORE_AON_CLKOUTMUX Overview: [0]

PRCM Register Manual
- CKGEN_PRM Register Summary: [1]
- CKGEN_PRM Register Description: [2][3][4]

Table 3-1058. CM_CLKSEL_DSP_GFCLK_CLKOUTMUX

Address Offset	0x0000 008C		
Physical Address	0x4AE0 618C	**Instance**	CKGEN_PRM
Description	Select the DSP_GFCLK. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																													CLKSEL		

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0
2:0	CLKSEL	Selects the divider value 0x0: Select CLK divided by 1 0x1: Select CLK divided by 2 0x2: Select CLK divided by 4 0x3: Select CLK divided by 8 0x4: Select CLK divided by 16 0x5: SELECT CLK divided by 32 0x6: RESERVED 0x7: RESERVED	RW	0x0

Table 3-1059. Register Call Summary for Register CM_CLKSEL_DSP_GFCLK_CLKOUTMUX

Clock Management Functional Description • CM_CORE_AON_CLKOUTMUX Overview: [0]
PRCM Register Manual • CKGEN_PRM Register Summary: [1] • CKGEN_PRM Register Description: [2][3][4]

Table 3-1060. CM_CLKSEL_EMIF_PHY_GCLK_CLKOUTMUX

Address Offset	0x0000 0090		
Physical Address	0x4AE0 6190	**Instance**	CKGEN_PRM
Description	Select the EMIF_PHY_GCLK. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																													CLKSEL		

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0
2:0	CLKSEL	Selects the divider value 0x0: Select CLK divided by 1 0x1: Select CLK divided by 2 0x2: Select CLK divided by 4 0x3: Select CLK divided by 8 0x4: Select CLK divided by 16 0x5: SELECT CLK divided by 32 0x6: RESERVED 0x7: RESERVED	RW	0x0

Table 3-1061. Register Call Summary for Register CM_CLKSEL_EMIF_PHY_GCLK_CLKOUTMUX

Clock Management Functional Description • CM_CORE_AON_CLKOUTMUX Overview: [0]
PRCM Register Manual • CKGEN_PRM Register Summary: [1] • CKGEN_PRM Register Description: [2][3][4]

Table 3-1062. CM_CLKSEL_EMU_CLK_CLKOUTMUX

Address Offset	0x0000 0094		
Physical Address	0x4AE0 6194	**Instance**	CKGEN_PRM
Description	Select the EMU_CLK. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																													CLKSEL		

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0
2:0	CLKSEL	Selects the divider value 0x0: Select CLK divided by 1 0x1: Select CLK divided by 2 0x2: Select CLK divided by 4 0x3: Select CLK divided by 8 0x4: Select CLK divided by 16 0x5: Select CLK divided by 32 0x6: RESERVED 0x7: RESERVED	RW	0x0

Table 3-1063. Register Call Summary for Register CM_CLKSEL_EMU_CLK_CLKOUTMUX

Clock Management Functional Description
- CM_CORE_AON_CLKOUTMUX Overview: [0]

PRCM Register Manual
- CKGEN_PRM Register Summary: [1]
- CKGEN_PRM Register Description: [2][3][4]

Table 3-1064. CM_CLKSEL_FUNC_96M_AON_CLK_CLKOUTMUX

Address Offset	0x0000 0098		
Physical Address	0x4AE0 6198	**Instance**	CKGEN_PRM
Description	Select the FUNC_96M_AON_CLK. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																													CLKSEL		

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0
2:0	CLKSEL	Selects the divider value 0x0: Select CLK divided by 1 0x1: Select CLK divided by 2 0x2: Select CLK divided by 4 0x3: Select CLK divided by 8 0x4: Select CLK divided by 16 0x5: SELECT CLK divided by 32 0x6: RESERVED 0x7: RESERVED	RW	0x0

Table 3-1065. Register Call Summary for Register CM_CLKSEL_FUNC_96M_AON_CLK_CLKOUTMUX

Clock Management Functional Description
- CM_CORE_AON_CLKOUTMUX Overview: [0]

PRCM Register Manual
- CKGEN_PRM Register Summary: [1]
- CKGEN_PRM Register Description: [2][3][4]

Table 3-1066. CM_CLKSEL_GMAC_250M_CLK_CLKOUTMUX

Address Offset	0x0000 009C		
Physical Address	0x4AE0 619C	**Instance**	CKGEN_PRM
Description	Select the GMAC_250M_CLK. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														CLKSEL	

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0
2:0	CLKSEL	Selects the divider value 0x0: Select CLK divided by 1 0x1: Select CLK divided by 2 0x2: Select CLK divided by 4 0x3: Select CLK divided by 8 0x4: Select CLK divided by 16 0x5: SELECT CLK divided by 32 0x6: RESERVED 0x7: RESERVED	RW	0x0

Table 3-1067. Register Call Summary for Register CM_CLKSEL_GMAC_250M_CLK_CLKOUTMUX

Clock Management Functional Description
- CM_CORE_AON_CLKOUTMUX Overview: [0]

PRCM Register Manual
- CKGEN_PRM Register Summary: [1]
- CKGEN_PRM Register Description: [2][3][4]

Table 3-1068. CM_CLKSEL_GPU_GCLK_CLKOUTMUX

Address Offset	0x0000 00A0		
Physical Address	0x4AE0 61A0	**Instance**	CKGEN_PRM
Description	Select the GPU_GCLK. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														CLKSEL	

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0
2:0	CLKSEL	Selects the divider value 0x0: Select CLK divided by 1 0x1: Select CLK divided by 2 0x2: Select CLK divided by 4 0x3: Select CLK divided by 8 0x4: Select CLK divided by 16 0x5: Select CLK divided by 32 0x6: RESERVED 0x7: RESERVED	RW	0x0

Table 3-1069. Register Call Summary for Register CM_CLKSEL_GPU_GCLK_CLKOUTMUX

Clock Management Functional Description
• CM_CORE_AON_CLKOUTMUX Overview: [0]
PRCM Register Manual
• CKGEN_PRM Register Summary: [1]
• CKGEN_PRM Register Description: [2][3][4]

Table 3-1070. CM_CLKSEL_HDMI_CLK_CLKOUTMUX

Address Offset	0x0000 00A4		
Physical Address	0x4AE0 61A4	**Instance**	CKGEN_PRM
Description	Select the HDMI_CLK. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																													CLKSEL		

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0
2:0	CLKSEL	Selects the divider value 0x0: Select CLK divided by 1 0x1: Select CLK divided by 2 0x2: Select CLK divided by 4 0x3: Select CLK divided by 8 0x4: Select CLK divided by 16 0x5: Select CLK divided by 32 0x6: RESERVED 0x7: RESERVED	RW	0x0

Table 3-1071. Register Call Summary for Register CM_CLKSEL_HDMI_CLK_CLKOUTMUX

Clock Management Functional Description
• CM_CORE_AON_CLKOUTMUX Overview: [0]
PRCM Register Manual
• CKGEN_PRM Register Summary: [1]
• CKGEN_PRM Register Description: [2][3][4]

Table 3-1072. CM_CLKSEL_IVA_GCLK_CLKOUTMUX

Address Offset	0x0000 00A8		
Physical Address	0x4AE0 61A8	**Instance**	CKGEN_PRM
Description	Select the IVA_GCLK. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																													CLKSEL		

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0
2:0	CLKSEL	Selects the divider value 0x0: Select CLK divided by 1 0x1: Select CLK divided by 2 0x2: Select CLK divided by 4 0x3: Select CLK divided by 8 0x4: Select CLK divided by 16 0x5: Select CLK divided by 32 0x6: RESERVED 0x7: RESERVED	RW	0x0

Table 3-1073. Register Call Summary for Register CM_CLKSEL_IVA_GCLK_CLKOUTMUX

Clock Management Functional Description
- CM_CORE_AON_CLKOUTMUX Overview: [0]

PRCM Register Manual
- CKGEN_PRM Register Summary: [1]
- CKGEN_PRM Register Description: [2][3][4]

Table 3-1074. CM_CLKSEL_L3INIT_480M_GFCLK_CLKOUTMUX

Address Offset	0x0000 00AC		
Physical Address	0x4AE0 61AC	**Instance**	CKGEN_PRM
Description	Select the L3INIT_480M_GFCLK. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																													CLKSEL		

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0
2:0	CLKSEL	Selects the divider value 0x0: Select CLK divided by 1 0x1: Select CLK divided by 2 0x2: Select CLK divided by 4 0x3: Select CLK divided by 8 0x4: Select CLK divided by 16 0x5: Select CLK divided by 32 0x6: RESERVED 0x7: RESERVED	RW	0x0

Table 3-1075. Register Call Summary for Register CM_CLKSEL_L3INIT_480M_GFCLK_CLKOUTMUX

Clock Management Functional Description
- CM_CORE_AON_CLKOUTMUX Overview: [0]

PRCM Register Manual
- CKGEN_PRM Register Summary: [1]
- CKGEN_PRM Register Description: [2][3][4]

Table 3-1076. CM_CLKSEL_MPU_GCLK_CLKOUTMUX

Address Offset	0x0000 00B0		
Physical Address	0x4AE0 61B0	**Instance**	CKGEN_PRM
Description	Select the MPU_GCLK. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																													CLKSEL		

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0
2:0	CLKSEL	Selects the divider value 0x0: Select CLK divided by 1 0x1: Select CLK divided by 2 0x2: Select CLK divided by 4 0x3: Select CLK divided by 8 0x4: Select CLK divided by 16 0x5: Select CLK divided by 32 0x6: RESERVED 0x7: RESERVED	RW	0x0

Table 3-1077. Register Call Summary for Register CM_CLKSEL_MPU_GCLK_CLKOUTMUX

Clock Management Functional Description
- CM_CORE_AON_CLKOUTMUX Overview: [0]

PRCM Register Manual
- CKGEN_PRM Register Summary: [1]
- CKGEN_PRM Register Description: [2][3][4]

Table 3-1078. CM_CLKSEL_PCIE1_CLK_CLKOUTMUX

Address Offset	0x0000 00B4		
Physical Address	0x4AE0 61B4	**Instance**	CKGEN_PRM
Description	Select the PCIE1_CLK. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																													CLKSEL		

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0
2:0	CLKSEL	Selects the divider value 0x0: Select CLK divided by 1 0x1: Select CLK divided by 2 0x2: Select CLK divided by 4 0x3: Select CLK divided by 8 0x4: Select CLK divided by 16 0x5: Select CLK divided by 32 0x6: RESERVED 0x7: RESERVED	RW	0x0

Table 3-1079. Register Call Summary for Register CM_CLKSEL_PCIE1_CLK_CLKOUTMUX

Clock Management Functional Description • CM_CORE_AON_CLKOUTMUX Overview: [0]
PRCM Register Manual • CKGEN_PRM Register Summary: [1] • CKGEN_PRM Register Description: [2][3][4]

Table 3-1080. CM_CLKSEL_PCIE2_CLK_CLKOUTMUX

Address Offset	0x0000 00B8		
Physical Address	0x4AE0 61B8	**Instance**	CKGEN_PRM
Description	Select the PCIE2_CLK. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																													CLKSEL		

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0
2:0	CLKSEL	Selects the divider value 0x0: Select CLK divided by 1 0x1: Select CLK divided by 2 0x2: Select CLK divided by 4 0x3: Select CLK divided by 8 0x4: Select CLK divided by 16 0x5: Select CLK divided by 32 0x6: RESERVED 0x7: RESERVED	RW	0x0

Table 3-1081. Register Call Summary for Register CM_CLKSEL_PCIE2_CLK_CLKOUTMUX

Clock Management Functional Description • CM_CORE_AON_CLKOUTMUX Overview: [0]
PRCM Register Manual • CKGEN_PRM Register Summary: [1] • CKGEN_PRM Register Description: [2][3][4]

Table 3-1082. CM_CLKSEL_PER_ABE_X1_CLK_CLKOUTMUX

Address Offset	0x0000 00BC		
Physical Address	0x4AE0 61BC	**Instance**	CKGEN_PRM
Description	Select the PER_ABE_X1_CLK. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																													CLKSEL		

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0
2:0	CLKSEL	Selects the divider value 0x0: Select CLK divided by 1 0x1: Select CLK divided by 2 0x2: Select CLK divided by 4 0x3: Select CLK divided by 8 0x4: Select CLK divided by 16 0x5: Select CLK divided by 32 0x6: RESERVED 0x7: RESERVED	RW	0x0

Table 3-1083. Register Call Summary for Register CM_CLKSEL_PER_ABE_X1_CLK_CLKOUTMUX

Clock Management Functional Description
- CM_CORE_AON_CLKOUTMUX Overview: [0]

PRCM Register Manual
- CKGEN_PRM Register Summary: [1]
- CKGEN_PRM Register Description: [2][3][4]

Table 3-1084. CM_CLKSEL_SATA_CLK_CLKOUTMUX

Address Offset	0x0000 00C0		
Physical Address	0x4AE0 61C0	**Instance**	CKGEN_PRM
Description	Select the SATA_CLK. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																													CLKSEL		

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0
2:0	CLKSEL	Selects the divider value 0x0: Select CLK divided by 1 0x1: Select CLK divided by 2 0x2: Select CLK divided by 4 0x3: Select CLK divided by 8 0x4: Select CLK divided by 16 0x5: Select CLK divided by 32 0x6: RESERVED 0x7: RESERVED	RW	0x0

Table 3-1085. Register Call Summary for Register CM_CLKSEL_SATA_CLK_CLKOUTMUX

Clock Management Functional Description
- CM_CORE_AON_CLKOUTMUX Overview: [0]

PRCM Register Manual
- CKGEN_PRM Register Summary: [1]
- CKGEN_PRM Register Description: [2][3][4]

Table 3-1086. CM_CLKSEL_SECURE_32K_CLK_CLKOUTMUX

Address Offset	0x0000 00C4		
Physical Address	0x4AE0 61C4	**Instance**	CKGEN_PRM
Description	Select the RCOSC_32K_CLK. [warm reset insensitive] Note: The RCOSC_32K_CLK clock, provided by the On-die 32K RC OSC is not accurate 32KHz clock. The frequency may vary with temperature and silicon characteristics.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																													CLKSEL		

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0
2:0	CLKSEL	Selects the divider value 0x0: Select CLK divided by 1 0x1: Select CLK divided by 2 0x2: Select CLK divided by 4 0x3: Select CLK divided by 8 0x4: Select CLK divided by 16 0x5: Select CLK divided by 32 0x6: RESERVED 0x7: RESERVED	RW	0x0

Table 3-1087. Register Call Summary for Register CM_CLKSEL_SECURE_32K_CLK_CLKOUTMUX

Clock Management Functional Description
- CM_CORE_AON_CLKOUTMUX Overview: [0]

PRCM Register Manual
- CKGEN_PRM Register Summary: [1]
- CKGEN_PRM Register Description: [2][3][4]

Table 3-1088. CM_CLKSEL_SYS_CLK1_CLKOUTMUX

Address Offset	0x0000 00C8		
Physical Address	0x4AE0 61C8	**Instance**	CKGEN_PRM
Description	Select the SYS_CLK1. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																													CLKSEL		

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0
2:0	CLKSEL	Selects the divider value 0x0: Select CLK divided by 1 0x1: Select CLK divided by 2 0x2: Select CLK divided by 4 0x3: Select CLK divided by 8 0x4: Select CLK divided by 16 0x5: SELECT CLK divided by 32 0x6: RESERVED 0x7: RESERVED	RW	0x0

Table 3-1089. Register Call Summary for Register CM_CLKSEL_SYS_CLK1_CLKOUTMUX

Clock Management Functional Description
- CM_CORE_AON_CLKOUTMUX Overview: [0]

PRCM Register Manual
- CKGEN_PRM Register Summary: [1]
- CKGEN_PRM Register Description: [2][3][4]

Table 3-1090. CM_CLKSEL_SYS_CLK2_CLKOUTMUX

Address Offset	0x0000 00CC		
Physical Address	0x4AE0 61CC	**Instance**	CKGEN_PRM
Description	Select the SYS_CLK2. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																													CLKSEL		

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0
2:0	CLKSEL	Selects the divider value 0x0: Select CLK divided by 1 0x1: Select CLK divided by 2 0x2: Select CLK divided by 4 0x3: Select CLK divided by 8 0x4: Select CLK divided by 16 0x5: SELECT CLK divided by 32 0x6: RESERVED 0x7: RESERVED	RW	0x0

Table 3-1091. Register Call Summary for Register CM_CLKSEL_SYS_CLK2_CLKOUTMUX

PRCM Register Manual
- CKGEN_PRM Register Summary: [0]
- CKGEN_PRM Register Description: [1][2]

Table 3-1092. CM_CLKSEL_VIDEO1_CLK_CLKOUTMUX

Address Offset	0x0000 00D0		
Physical Address	0x4AE0 61D0	**Instance**	CKGEN_PRM
Description	Select the VIDEO1_CLK. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																													CLKSEL		

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0
2:0	CLKSEL	Selects the divider value	RW	0x0
		0x0: Select CLK divided by 1		
		0x1: Select CLK divided by 2		
		0x2: Select CLK divided by 4		
		0x3: Select CLK divided by 8		
		0x4: Select CLK divided by 16		
		0x5: Select CLK divided by 32		
		0x6: RESERVED		
		0x7: RESERVED		

Table 3-1093. Register Call Summary for Register CM_CLKSEL_VIDEO1_CLK_CLKOUTMUX

Clock Management Functional Description
- CM_CORE_AON_CLKOUTMUX Overview: [0]

PRCM Register Manual
- CKGEN_PRM Register Summary: [1]
- CKGEN_PRM Register Description: [2][3][4]

Table 3-1094. CM_CLKSEL_VIDEO2_CLK_CLKOUTMUX

Address Offset	0x0000 00D4		
Physical Address	0x4AE0 61D4	**Instance**	CKGEN_PRM
Description	Select the VIDEO2_CLK. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																													CLKSEL		

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0
2:0	CLKSEL	Selects the divider value	RW	0x0
		0x0: Select CLK divided by 1		
		0x1: Select CLK divided by 2		
		0x2: Select CLK divided by 4		
		0x3: Select CLK divided by 8		
		0x4: Select CLK divided by 16		
		0x5: Select CLK divided by 32		
		0x6: RESERVED		
		0x7: RESERVED		

Table 3-1095. Register Call Summary for Register CM_CLKSEL_VIDEO2_CLK_CLKOUTMUX

Clock Management Functional Description • CM_CORE_AON_CLKOUTMUX Overview: [0]
PRCM Register Manual • CKGEN_PRM Register Summary: [1] • CKGEN_PRM Register Description: [2][3][4]

Table 3-1096. CM_CLKSEL_ABE_LP_CLK

Address Offset	0x0000 00D8		
Physical Address	0x4AE0 61D8	**Instance**	CKGEN_PRM
Description	Select the ABE_LP_CLK. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															CLKSEL

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	CLKSEL	Selects the divider value 0x0: Select CLK divided by 16 0x1: Select CLK divided by 32	RW	0x0

Table 3-1097. Register Call Summary for Register CM_CLKSEL_ABE_LP_CLK

Clock Management Functional Description • CM_CORE_AON Clock Generator: [0]
PRCM Register Manual • CKGEN_PRM Register Summary: [1]

Table 3-1098. CM_CLKSEL_ADC_GFCLK

Address Offset	0x0000 00DC		
Physical Address	0x4AE0 61DC	**Instance**	CKGEN_PRM
Description	Control the source of the ADC_GFCLK clock for		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															CLKSEL

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1:0	CLKSEL	Select the SYS clock for the DPLL_ABE reference and bypass clock. 0x0: Selects SYS_CLK1 0x1: Selects SYS_CLK2 0x2: Selects SYS_CLK1_32K_CLK 0x3: RESERVED	RW	0x0

Table 3-1099. CM_CLKSEL_EVE_GFCLK_CLKOUTMUX

Address Offset	0x0000 00E0		
Physical Address	0x4AE0 61E0	**Instance**	CKGEN_PRM
Description	Select the EVE_GFCLK. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														CLKSEL	

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0
2:0	CLKSEL	Selects the divider value 0x0: Select CLK divided by 1 0x1: Select CLK divided by 2 0x2: Select CLK divided by 4 0x3: Select CLK divided by 8 0x4: Select CLK divided by 16 0x5: SELECT CLK divided by 32 0x6: RESERVED 0x7: RESERVED	RW	0x0

Table 3-1100. Register Call Summary for Register CM_CLKSEL_EVE_GFCLK_CLKOUTMUX

Clock Management Functional Description
- CM_CORE_AON_CLKOUTMUX Overview: [0]

PRCM Register Manual
- CKGEN_PRM Register Summary: [1]
- CKGEN_PRM Register Description: [2][3][4]

3.12.32 COREAON_PRM registers

3.12.32.1 COREAON_PRM Register Summary

Table 3-1101. COREAON_PRM Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	COREAON_PRM Physical Address
PM_COREAON_SMARTREFLEX_MPU_WKDEP	RW	32	0x0000 0000	0x4AE0 6628
RM_COREAON_SMARTREFLEX_MPU_CONTEXT	RW	32	0x0000 0004	0x4AE0 662C
PM_COREAON_SMARTREFLEX_CORE_WKDEP	RW	32	0x0000 0010	0x4AE0 6638
RM_COREAON_SMARTREFLEX_CORE_CONTEXT	RW	32	0x0000 0014	0x4AE0 663C
PM_COREAON_SMARTREFLEX_GPU_WKDEP	RW	32	0x0000 0030	0x4AE0 6658

Table 3-1101. COREAON_PRM Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	COREAON_PRM Physical Address
RM_COREAON_SMARTREFLEX_GPU_CONTEXT	RW	32	0x0000 0034	0x4AE0 665C
PM_COREAON_SMARTREFLEX_DSPEVE_WKDEP	RW	32	0x0000 0040	0x4AE0 6668
RM_COREAON_SMARTREFLEX_DSPEVE_CONTEXT	RW	32	0x0000 0044	0x4AE0 666C
PM_COREAON_SMARTREFLEX_IVAHD_WKDEP	RW	32	0x0000 0050	0x4AE0 6678
RM_COREAON_SMARTREFLEX_IVAHD_CONTEXT	RW	32	0x0000 0054	0x4AE0 667C
RM_COREAON_DUMMY_MODULE1_CONTEXT	RW	32	0x0000 0084	0x4AE0 66AC
RM_COREAON_DUMMY_MODULE2_CONTEXT	RW	32	0x0000 0094	0x4AE0 66BC
RM_COREAON_DUMMY_MODULE3_CONTEXT	RW	32	0x0000 00A4	0x4AE0 66CC
RM_COREAON_DUMMY_MODULE4_CONTEXT	RW	32	0x0000 00B4	0x4AE0 66DC

3.12.32.2 COREAON_PRM Register Description

Table 3-1102. PM_COREAON_SMARTREFLEX_MPU_WKDEP

Address Offset	0x0000 0000		
Physical Address	0x4AE0 6628	**Instance**	COREAON_PRM
Description	This register controls wakeup dependency based on SR_MPU service requests.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						WKUPDEP_SMARTREFLEX_MPU_EVE4	WKUPDEP_SMARTREFLEX_MPU_EVE3	WKUPDEP_SMARTREFLEX_MPU_EVE2	WKUPDEP_SMARTREFLEX_MPU_EVE1	WKUPDEP_SMARTREFLEX_MPU_DSP2	WKUPDEP_SMARTREFLEX_MPU_IPU1	RESERVED	WKUPDEP_SMARTREFLEX_MPU_DSP1	WKUPDEP_SMARTREFLEX_MPU_IPU2	WKUPDEP_SMARTREFLEX_MPU_MPU								

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	WKUPDEP_SMARTREFLEX_MPU_EVE4	Wakeup dependency from SMARTREFLEX_MPU module (SWakeup signal) towards EVE4 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
8	WKUPDEP_SMARTREFLEX_MPU_EVE3	Wakeup dependency from SMARTREFLEX_MPU module (SWakeup signal) towards EVE3 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
7	WKUPDEP_SMARTREFLEX_MPU_EVE2	Wakeup dependency from SMARTREFLEX_MPU module (SWakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_SMARTREFLEX_MPU_EVE1	Wakeup dependency from SMARTREFLEX_MPU module (SWakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_SMARTREFLEX_MPU_DSP2	Wakeup dependency from SMARTREFLEX_MPU module (SWakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_SMARTREFLEX_MPU_IPU1	Wakeup dependency from SMARTREFLEX_MPU module (SWakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	RESERVED		R	0x0
2	WKUPDEP_SMARTREFLEX_MPU_DSP1	Wakeup dependency from SMARTREFLEX_MPU module (SWakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_SMARTREFLEX_MPU_IPU2	Wakeup dependency from SMARTREFLEX_MPU module (SWakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_SMARTREFLEX_MPU_MPU	Wakeup dependency from SMARTREFLEX_MPU module (SWakeup signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1103. RM_COREAON_SMARTREFLEX_MPU_CONTEXT

Address Offset	0x0000 0004		
Physical Address	0x4AE0 662C	**Instance**	COREAON_PRM
Description	This register contains dedicated SR_MPU context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of COREAON_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1104. PM_COREAON_SMARTREFLEX_CORE_WKDEP

Address Offset	0x0000 0010		
Physical Address	0x4AE0 6638	**Instance**	COREAON_PRM
Description	This register controls wakeup dependency based on SR_CORE service requests.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						WKUPDEP_SMARTREFLEX_CORE_EVE4	WKUPDEP_SMARTREFLEX_CORE_EVE3	WKUPDEP_SMARTREFLEX_CORE_EVE2	WKUPDEP_SMARTREFLEX_CORE_EVE1	WKUPDEP_SMARTREFLEX_CORE_DSP2	WKUPDEP_SMARTREFLEX_CORE_IPU1	RESERVED	WKUPDEP_SMARTREFLEX_CORE_DSP1	WKUPDEP_SMARTREFLEX_CORE_IPU2	WKUPDEP_SMARTREFLEX_CORE_MPU

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	WKUPDEP_SMARTREFLEX_CORE_EVE4	Wakeup dependency from SMARTREFLEX_CORE module (SWakeup signal) towards EVE4 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
8	WKUPDEP_SMARTREFLEX_CORE_EVE3	Wakeup dependency from SMARTREFLEX_CORE module (SWakeup signal) towards EVE3 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
7	WKUPDEP_SMARTREFLEX_CORE_EVE2	Wakeup dependency from SMARTREFLEX_CORE module (SWakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_SMARTREFLEX_CORE_EVE1	Wakeup dependency from SMARTREFLEX_CORE module (SWakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
5	WKUPDEP_SMARTREFLEX_CORE_DSP2	Wakeup dependency from SMARTREFLEX_CORE module (SWakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_SMARTREFLEX_CORE_IPU1	Wakeup dependency from SMARTREFLEX_CORE module (SWakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	RESERVED		R	0x0
2	WKUPDEP_SMARTREFLEX_CORE_DSP1	Wakeup dependency from SMARTREFLEX_CORE module (SWakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_SMARTREFLEX_CORE_IPU2	Wakeup dependency from SMARTREFLEX_CORE module (SWakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_SMARTREFLEX_CORE_MPU	Wakeup dependency from SMARTREFLEX_CORE module (SWakeup signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1105. RM_COREAON_SMARTREFLEX_CORE_CONTEXT

Address Offset	0x0000 0014		
Physical Address	0x4AE0 663C	**Instance**	COREAON_PRM
Description	This register contains dedicated SR_CORE context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of COREAON_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1106. PM_COREAON_SMARTREFLEX_GPU_WKDEP

Address Offset	0x0000 0030		
Physical Address	0x4AE0 6658	**Instance**	COREAON_PRM
Description	This register controls wakeup dependency based on SMARTREFLEX_GPU service requests.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						WKUPDEP_SMARTREFLEX_GPU_EVE4	WKUPDEP_SMARTREFLEX_GPU_EVE3	WKUPDEP_SMARTREFLEX_GPU_EVE2	WKUPDEP_SMARTREFLEX_GPU_EVE1	WKUPDEP_SMARTREFLEX_GPU_DSP2	WKUPDEP_SMARTREFLEX_GPU_IPU1	RESERVED	WKUPDEP_SMARTREFLEX_GPU_DSP1	WKUPDEP_SMARTREFLEX_GPU_IPU2	WKUPDEP_SMARTREFLEX_GPU_MPU

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	WKUPDEP_SMARTREFLEX_GPU_EVE4	Wakeup dependency from SMARTREFLEX_GPU module (SWakeup signal) towards EVE4 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
8	WKUPDEP_SMARTREFLEX_GPU_EVE3	Wakeup dependency from SMARTREFLEX_GPU module (SWakeup signal) towards EVE3 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
7	WKUPDEP_SMARTREFLEX_GPU_EVE2	Wakeup dependency from SMARTREFLEX_GPU module (SWakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_SMARTREFLEX_GPU_EVE1	Wakeup dependency from SMARTREFLEX_GPU module (SWakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_SMARTREFLEX_GPU_DSP2	Wakeup dependency from SMARTREFLEX_GPU module (SWakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_SMARTREFLEX_GPU_IPU1	Wakeup dependency from SMARTREFLEX_GPU module (SWakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
2	WKUPDEP_SMARTREFLEX_GPU_DSP1	Wakeup dependency from SMARTREFLEX_GPU module (SWakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_SMARTREFLEX_GPU_IPU2	Wakeup dependency from SMARTREFLEX_GPU module (SWakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_SMARTREFLEX_GPU_MPU	Wakeup dependency from SMARTREFLEX_GPU module (SWakeup signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1107. RM_COREAON_SMARTREFLEX_GPU_CONTEXT

Address Offset	0x0000 0034		
Physical Address	0x4AE0 665C	**Instance**	COREAON_PRM
Description	This register contains dedicated SR_GPU context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of COREAON_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1108. PM_COREAON_SMARTREFLEX_DSPEVE_WKDEP

Address Offset	0x0000 0040		
Physical Address	0x4AE0 6668	**Instance**	COREAON_PRM
Description	This register controls wakeup dependency based on SMARTREFLEX_DSPEVE service requests.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						WKUPDEP_SMARTREFLEX_DSPEVE_EVE4	WKUPDEP_SMARTREFLEX_DSPEVE_EVE3	WKUPDEP_SMARTREFLEX_DSPEVE_EVE2	WKUPDEP_SMARTREFLEX_DSPEVE_EVE1	WKUPDEP_SMARTREFLEX_DSPEVE_DSP2	WKUPDEP_SMARTREFLEX_DSPEVE_IPU1	WKUPDEP_SMARTREFLEX_DSPEVE_SDMA	WKUPDEP_SMARTREFLEX_DSPEVE_DSP1	WKUPDEP_SMARTREFLEX_DSPEVE_IPU2	WKUPDEP_SMARTREFLEX_DSPEVE_MPU

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	WKUPDEP_SMARTREFLEX_DSPEVE_EVE4	Wakeup dependency from SMARTREFLEX_DSPEVE module (SWakeup signal) towards EVE4 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
8	WKUPDEP_SMARTREFLEX_DSPEVE_EVE3	Wakeup dependency from SMARTREFLEX_DSPEVE module (SWakeup signal) towards EVE3 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
7	WKUPDEP_SMARTREFLEX_DSPEVE_EVE2	Wakeup dependency from SMARTREFLEX_DSPEVE module (SWakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_SMARTREFLEX_DSPEVE_EVE1	Wakeup dependency from SMARTREFLEX_DSPEVE module (SWakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_SMARTREFLEX_DSPEVE_DSP2	Wakeup dependency from SMARTREFLEX_DSPEVE module (SWakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_SMARTREFLEX_DSPEVE_IPU1	Wakeup dependency from SMARTREFLEX_DSPEVE module (SWakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	WKUPDEP_SMARTREFLEX_DSPEVE_SDMA	Wakeup dependency from SMARTREFLEX_DSPEVE module (SWakeup signal) towards SDMA + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
2	WKUPDEP_SMARTREFLEX_DSPEVE_DSP1	Wakeup dependency from SMARTREFLEX_GPU module (SWakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_SMARTREFLEX_DSPEVE_IPU2	Wakeup dependency from SMARTREFLEX_DSPEVE module (SWakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_SMARTREFLEX_DSPEVE_MPU	Wakeup dependency from SMARTREFLEX_DSPEVE module (SWakeup signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1109. RM_COREAON_SMARTREFLEX_DSPEVE_CONTEXT

Address Offset	0x0000 0044		
Physical Address	0x4AE0 666C	**Instance**	COREAON_PRM
Description	This register contains dedicated SR_DSPEVE context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of COREAON_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1110. PM_COREAON_SMARTREFLEX_IVAHD_WKDEP

Address Offset	0x0000 0050		
Physical Address	0x4AE0 6678	**Instance**	COREAON_PRM
Description	This register controls wakeup dependency based on SMARTREFLEX_IVAHD service requests.		
Type	RW		

31–10	9	8	7	6	5	4	3	2	1	0
RESERVED	WKUPDEP_SMARTREFLEX_IVAHD_EVE4	WKUPDEP_SMARTREFLEX_IVAHD_EVE3	WKUPDEP_SMARTREFLEX_IVAHD_EVE2	WKUPDEP_SMARTREFLEX_IVAHD_EVE1	WKUPDEP_SMARTREFLEX_IVAHD_DSP2	WKUPDEP_SMARTREFLEX_IVAHD_IPU1	RESERVED	WKUPDEP_SMARTREFLEX_IVAHD_DSP1	WKUPDEP_SMARTREFLEX_IVAHD_IPU2	WKUPDEP_SMARTREFLEX_IVAHD_MPU

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	WKUPDEP_SMARTREFLEX_IVAHD_EVE4	Wakeup dependency from SMARTREFLEX_IVAHD module (SWakeup signal) towards EVE4 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
8	WKUPDEP_SMARTREFLEX_IVAHD_EVE3	Wakeup dependency from SMARTREFLEX_IVAHD module (SWakeup signal) towards EVE3 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
7	WKUPDEP_SMARTREFLEX_IVAHD_EVE2	Wakeup dependency from SMARTREFLEX_IVAHD module (SWakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_SMARTREFLEX_IVAHD_EVE1	Wakeup dependency from SMARTREFLEX_IVAHD module (SWakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_SMARTREFLEX_IVAHD_DSP2	Wakeup dependency from SMARTREFLEX_IVAHD module (SWakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_SMARTREFLEX_IVAHD_IPU1	Wakeup dependency from SMARTREFLEX_IVAHD module (SWakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	RESERVED		R	0x0
2	WKUPDEP_SMARTREFLEX_IVAHD_DSP1	Wakeup dependency from SMARTREFLEX_IVAHD module (SWakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
1	WKUPDEP_SMARTREFLEX_IVAHD_IPU2	Wakeup dependency from SMARTREFLEX_IVAHD module (SWakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_SMARTREFLEX_IVAHD_MPU	Wakeup dependency from SMARTREFLEX_IVAHD module (SWakeup signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1111. RM_COREAON_SMARTREFLEX_IVAHD_CONTEXT

Address Offset	0x0000 0054		
Physical Address	0x4AE0 667C	**Instance**	COREAON_PRM
Description	This register contains dedicated SR_IVA context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of COREAON_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1112. RM_COREAON_DUMMY_MODULE1_CONTEXT

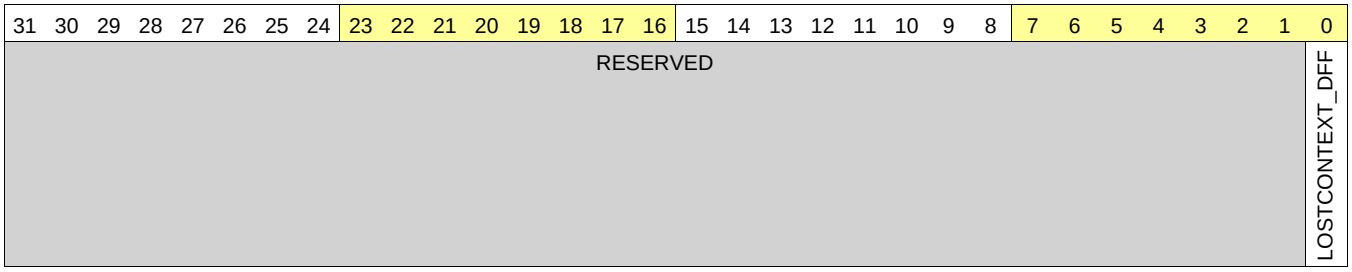
Address Offset	0x0000 0084		
Physical Address	0x4AE0 66AC	**Instance**	COREAON_PRM
Description	This register contains dedicated DUMMY MODULE1 context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of COREAON_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1113. RM_COREAON_DUMMY_MODULE2_CONTEXT

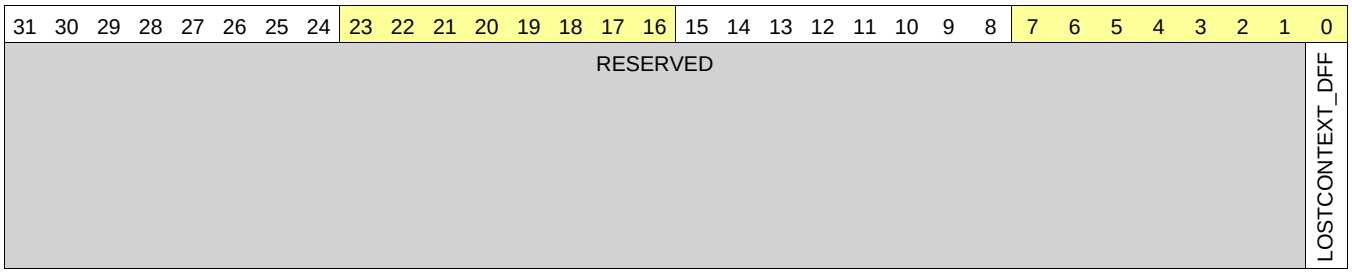
Address Offset	0x0000 0094		
Physical Address	0x4AE0 66BC	**Instance**	COREAON_PRM
Description	This register contains dedicated DUMMY MODULE2 context statuses. [warm reset insensitive]		
Type	RW		



Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of COREAON_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1114. RM_COREAON_DUMMY_MODULE3_CONTEXT

Address Offset	0x0000 00A4		
Physical Address	0x4AE0 66CC	**Instance**	COREAON_PRM
Description	This register contains DUMMY MODULE USBSTUB context statuses. [warm reset insensitive]		
Type	RW		



Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of COREAON_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1115. RM_COREAON_DUMMY_MODULE4_CONTEXT

Address Offset	0x0000 00B4		
Physical Address	0x4AE0 66DC	**Instance**	COREAON_PRM
Description	This register contains dedicated DUMMY MODULE context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of COREAON_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

3.12.33 CORE_PRM registers

3.12.33.1 CORE_PRM Register Summary

Table 3-1116. CORE_PRM Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	CORE_PRM Physical Address
PM_CORE_PWRSTCTRL	RW	32	0x0000 0000	0x4AE0 6700
PM_CORE_PWRSTST	RW	32	0x0000 0004	0x4AE0 6704
RM_L3MAIN1_L3_MAIN_1_CONTEXT	RW	32	0x0000 0024	0x4AE0 6724
RM_L3MAIN1_GPMC_CONTEXT	RW	32	0x0000 002C	0x4AE0 672C
RM_L3MAIN1_MMU_EDMA_CONTEXT	RW	32	0x0000 0034	0x4AE0 6734
RM_L3MAIN1_MMU_PCIESS_CONTEXT	RW	32	0x0000 004C	0x4AE0 674C
PM_L3MAIN1_OCMC_RAM1_WKDEP	RW	32	0x0000 0050	0x4AE0 6750
RM_L3MAIN1_OCMC_RAM1_CONTEXT	RW	32	0x0000 0054	0x4AE0 6754
PM_L3MAIN1_TESOC_WKDEP	RW	32	0x0000 0058	0x4AE0 6758
RM_L3MAIN1_TESOC_CONTEXT	RW	32	0x0000 005C	0x4AE0 675C
PM_L3MAIN1_OCMC_RAM3_WKDEP	RW	32	0x0000 0060	0x4AE0 6760
RM_L3MAIN1_OCMC_RAM3_CONTEXT	RW	32	0x0000 0064	0x4AE0 6764
RM_L3MAIN1_OCMC_ROM_CONTEXT	RW	32	0x0000 006C	0x4AE0 676C

Table 3-1116. CORE_PRM Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	CORE_PRM Physical Address
PM_L3MAIN1_TPCC_WKDEP	RW	32	0x0000 0070	0x4AE0 6770
RM_L3MAIN1_TPCC_CONTEXT	RW	32	0x0000 0074	0x4AE0 6774
PM_L3MAIN1_TPTC1_WKDEP	RW	32	0x0000 0078	0x4AE0 6778
RM_L3MAIN1_TPTC1_CONTEXT	RW	32	0x0000 007C	0x4AE0 677C
PM_L3MAIN1_TPTC2_WKDEP	RW	32	0x0000 0080	0x4AE0 6780
RM_L3MAIN1_TPTC2_CONTEXT	RW	32	0x0000 0084	0x4AE0 6784
RM_L3MAIN1_VCP1_CONTEXT	RW	32	0x0000 008C	0x4AE0 678C
RM_L3MAIN1_VCP2_CONTEXT	RW	32	0x0000 0094	0x4AE0 6794
RM_L3MAIN1_SPARE_CME_CONTEXT	RW	32	0x0000 009C	0x4AE0 679C
RM_L3MAIN1_SPARE_HDMI_CONTEXT	RW	32	0x0000 00A4	0x4AE0 67A4
RM_L3MAIN1_SPARE_ICM_CONTEXT	RW	32	0x0000 00AC	0x4AE0 67AC
RM_L3MAIN1_SPARE_IVA2_CONTEXT	RW	32	0x0000 00B4	0x4AE0 67B4
RM_L3MAIN1_SPARE_SATA2_CONTEXT	RW	32	0x0000 00BC	0x4AE0 67BC
RM_L3MAIN1_SPARE_UNKNOWN4_CONTEXT	RW	32	0x0000 00C4	0x4AE0 67C4
RM_L3MAIN1_SPARE_UNKNOWN5_CONTEXT	RW	32	0x0000 00CC	0x4AE0 67CC
RM_L3MAIN1_SPARE_UNKNOWN6_CONTEXT	RW	32	0x0000 00D4	0x4AE0 67D4
RM_L3MAIN1_SPARE_VIDEOPLL1_CONTEXT	RW	32	0x0000 00DC	0x4AE0 67DC
RM_L3MAIN1_SPARE_VIDEOPLL2_CONTEXT	RW	32	0x0000 00F4	0x4AE0 67F4
RM_L3MAIN1_SPARE_VIDEOPLL3_CONTEXT	RW	32	0x0000 00FC	0x4AE0 67FC
RM_IPU2_RSTCTRL	RW	32	0x0000 0210	0x4AE0 6910
RM_IPU2_RSTST	RW	32	0x0000 0214	0x4AE0 6914
RM_IPU2_IPU2_CONTEXT	RW	32	0x0000 0224	0x4AE0 6924
RM_DMA_DMA_SYSTEM_CONTEXT	RW	32	0x0000 0324	0x4AE0 6A24
RM_EMIF_DMM_CONTEXT	RW	32	0x0000 0424	0x4AE0 6B24
RM_EMIF_EMIF_OCP_FW_CONTEXT	RW	32	0x0000 042C	0x4AE0 6B2C
RM_EMIF_EMIF1_CONTEXT	RW	32	0x0000 0434	0x4AE0 6B34
RM_EMIF_EMIF2_CONTEXT	RW	32	0x0000 043C	0x4AE0 6B3C
RM_EMIF_EMIF_DLL_CONTEXT	RW	32	0x0000 0444	0x4AE0 6B44
RM_CRC_CRC_CONTEXT	RW	32	0x0000 0524	0x4AE0 6C24
RM_L4CFG_L4_CFG_CONTEXT	RW	32	0x0000 0624	0x4AE0 6D24
RM_L4CFG_SPINLOCK_CONTEXT	RW	32	0x0000 062C	0x4AE0 6D2C
RM_L4CFG_MAILBOX1_CONTEXT	RW	32	0x0000 0634	0x4AE0 6D34
RM_L4CFG_SAR_ROM_CONTEXT	RW	32	0x0000 063C	0x4AE0 6D3C
RM_L4CFG_OCP2SCP2_CONTEXT	RW	32	0x0000 0644	0x4AE0 6D44
RM_L4CFG_MAILBOX2_CONTEXT	RW	32	0x0000 064C	0x4AE0 6D4C
RM_L4CFG_MAILBOX3_CONTEXT	RW	32	0x0000 0654	0x4AE0 6D54
RM_L4CFG_MAILBOX4_CONTEXT	RW	32	0x0000 065C	0x4AE0 6D5C
RM_L4CFG_MAILBOX5_CONTEXT	RW	32	0x0000 0664	0x4AE0 6D64
RM_L4CFG_MAILBOX6_CONTEXT	RW	32	0x0000 066C	0x4AE0 6D6C
RM_L4CFG_MAILBOX7_CONTEXT	RW	32	0x0000 0674	0x4AE0 6D74
RM_L4CFG_MAILBOX8_CONTEXT	RW	32	0x0000 067C	0x4AE0 6D7C
RM_L4CFG_MAILBOX9_CONTEXT	RW	32	0x0000 0684	0x4AE0 6D84
RM_L4CFG_MAILBOX10_CONTEXT	RW	32	0x0000 068C	0x4AE0 6D8C
RM_L4CFG_MAILBOX11_CONTEXT	RW	32	0x0000 0694	0x4AE0 6D94
RM_L4CFG_MAILBOX12_CONTEXT	RW	32	0x0000 069C	0x4AE0 6D9C
RM_L4CFG_MAILBOX13_CONTEXT	RW	32	0x0000 06A4	0x4AE0 6DA4

Table 3-1116. CORE_PRM Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	CORE_PRM Physical Address
RM_L4CFG_SPARE_SMARTREFLEX_RTC_CONTEXT	RW	32	0x0000 06AC	0x4AE0 6DAC
RM_L4CFG_SPARE_SMARTREFLEX_SDRAM_CONTEXT	RW	32	0x0000 06B4	0x4AE0 6DB4
RM_L4CFG_SPARE_SMARTREFLEX_WKUP_CONTEXT	RW	32	0x0000 06BC	0x4AE0 6DBC
RM_L4CFG_IO_DELAY_BLOCK_CONTEXT	RW	32	0x0000 06C4	0x4AE0 6DC4
RM_L3INSTR_L3_MAIN_2_CONTEXT	RW	32	0x0000 0724	0x4AE0 6E24
RM_L3INSTR_L3_INSTR_CONTEXT	RW	32	0x0000 072C	0x4AE0 6E2C
RM_L3INSTR_OCP_WP_NOC_CONTEXT	RW	32	0x0000 0744	0x4AE0 6E44

3.12.33.2 CORE_PRM Register Description

Table 3-1117. PM_CORE_PWRSTCTRL

Address Offset	0x0000 0000		
Physical Address	0x4AE0 6700	**Instance**	CORE_PRM
Description	This register controls the CORE power state to reach upon a domain sleep transition		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED						OCP_NRET_BANK_ONSTATE		IPU_UNICACHE_ONSTATE		IPU_L2RAM_ONSTATE		CORE_OCMRAM_ONSTATE		CORE_OTHER_BANK_ONSTATE		RESERVED			OCP_NRET_BANK_RETSTATE	IPU_UNICACHE_RETSTATE	IPU_L2RAM_RETSTATE	CORE_OCMRAM_RETSTATE	CORE_OTHER_BANK_RETSTATE	RESERVED			LOWPOWERSTATECHANGE	RESERVED	LOGICRETSTATE	POWERSTATE	

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:24	OCP_NRET_BANK_ONSTATE	OCP_WP bank and DMM bank2 state when domain is ON. 0x3: Memory bank is on when the domain is ON.	R	0x3
23:22	IPU_UNICACHE_ONSTATE	IPU UNICACHE bank state when domain is ON. 0x3: Memory bank is on when the domain is ON.	R	0x3
21:20	IPU_L2RAM_ONSTATE	IPU L2 bank state when domain is ON. 0x3: Memory bank is on when the domain is ON.	R	0x3
19:18	CORE_OCMRAM_ONSTATE	OCMRAM bank state when domain is ON. 0x3: Memory bank is on when the domain is ON.	R	0x3
17:16	CORE_OTHER_BANK_ONSTATE	DMA/ICR bank and DMM bank1 state when domain is ON. 0x3: Memory bank is on when the domain is ON.	R	0x3
15:13	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
12	OCP_NRET_BANK_RETSTATE	OCP_WP bank and DMM bank2 state when domain is RETENTION. 0x0: Memory bank is off when the domain is in the RETENTION state.	R	0x0
11	IPU_UNICACHE_RETSTATE	IPU UNICACHE bank state when domain is RETENTION. 0x0: Memory bank is off when the domain is in the RETENTION state. 0x1: Memory bank is retained when domain is in RETENTION state.	RW	0x1
10	IPU_L2RAM_RETSTATE	IPU L2 bank state when domain is RETENTION. 0x0: Memory bank is off when the domain is in the RETENTION state. 0x1: Memory bank is retained when domain is in RETENTION state.	RW	0x1
9	CORE_OCMRAM_RETSTATE	OCMRAM bank state when domain is RETENTION. 0x1: Memory bank is retained when domain is in RETENTION state.	R	0x1
8	CORE_OTHER_BANK_RETSTATE	DMA/ICR bank and DMM bank1 state when domain is RETENTION. 0x1: Memory bank is retained when domain is in RETENTION state.	R	0x1
7:5	RESERVED		R	0x0
4	LOWPOWERSTATECHANGE	Power state change request when domain has already performed a sleep transition. Allows going into deeper low power state without waking up the power domain. 0x0: Do not request a low power state change. 0x1: Request a low power state change. This bit is automatically cleared when the power state is effectively changed or when power state is ON.	RW	0x0
3	RESERVED		R	0x0
2	LOGICRETSTATE	Logic state when power domain is RETENTION 0x0: Only retention registers are retained and remaing logic is off when the domain is in RETENTION state. 0x1: Whole logic is retained when domain is in RETENTION state.	RW	0x1
1:0	POWERSTATE	Power state control 0x0: Reserved 0x1: RETENTION state 0x2: INACTIVE state 0x3: ON State	RW	0x3

Table 3-1118. Register Call Summary for Register PM_CORE_PWRSTCTRL

Power Management Functional Description
PRCM Register Manual • Not Supported Functionality (Registers and Bitfields): [13][14][15][16][17] • CORE_PRM Register Summary: [18]

Table 3-1119. PM_CORE_PWRSTST

Address Offset	0x0000 0004		
Physical Address	0x4AE0 6704	**Instance**	CORE_PRM
Description	This register provides a status on the current CORE power domain state. [warm reset insensitive]		
Type	RW		

31 30 29 28 27 26	25 24	23 22 21	20	19 18 17 16	15 14	13 12	11 10	9 8	7 6	5 4	3	2	1 0
RESERVED	LASTPOWERSTATEENTERED	RESERVED	INTRANSITION	RESERVED		OCP_NRET_BANK_STATEST	IPU_UNICACHE_STATEST	IPU_L2RAM_STATEST	CORE_OCMRAM_STATEST	CORE_OTHER_BANK_STATEST	RESERVED	LOGICSTATEST	POWERSTATEST

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:24	LASTPOWERSTATEENTERED	Last low power state entered. Set to 0x3 upon write of the same only. This register is intended for debug purpose only. 0x0: Power domain was previously OFF 0x1: Power domain was previously in RETENTION 0x2: Power domain was previously ON-INACTIVE 0x3: Power domain was previously ON-ACTIVE	RW	0x0
23:21	RESERVED		R	0x0
20	INTRANSITION	Domain transition status 0x0: No on-going transition on power domain 0x1: Power domain transition is in progress.	R	0x0
19:14	RESERVED		R	0x0
13:12	OCP_NRET_BANK_STATEST	OCP_WP bank and DMM bank2 state status 0x0: Memory is OFF 0x1: Reserved 0x2: Reserved 0x3: Memory is ON	R	0x3
11:10	IPU_UNICACHE_STATEST	IPU UNICACHE bank state status 0x0: Memory is OFF 0x1: Memory is RETENTION 0x2: Reserved 0x3: Memory is ON	R	0x3
9:8	IPU_L2RAM_STATEST	IPU L2 bank state status 0x0: Memory is OFF 0x1: Memory is RETENTION 0x2: Reserved 0x3: Memory is ON	R	0x3
7:6	CORE_OCMRAM_STATEST	OCMRAM bank state status 0x0: Memory is OFF 0x1: Memory is RETENTION 0x2: Reserved 0x3: Memory is ON	R	0x3
5:4	CORE_OTHER_BANK_STATEST	DMA/ICR bank and DMM bank1 state status 0x0: Memory is OFF 0x1: Memory is RETENTION 0x2: Reserved 0x3: Memory is ON	R	0x3

Bits	Field Name	Description	Type	Reset
3	RESERVED		R	0x0
2	LOGICSTATEST	Logic state status 0x0: Logic in domain is OFF 0x1: Logic in domain is ON	R	0x1
1:0	POWERSTATEST	Current power state status 0x0: Reserved 0x1: Power domain is in RETENTION 0x2: Power domain is ON-INACTIVE 0x3: Power domain is ON-ACTIVE	R	0x3

Table 3-1120. Register Call Summary for Register PM_CORE_PWRSTST

Power Management Functional Description
PRCM Register Manual • Not Supported Functionality (Registers and Bitfields): [9][10][11][12][13][14][15] • CORE_PRM Register Summary: [16]

Table 3-1121. RM_L3MAIN1_L3_MAIN_1_CONTEXT

Address Offset	0x0000 0024		
Physical Address	0x4AE0 6724	**Instance**	CORE_PRM
Description	This register contains dedicated L3_MAIN_1 context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														LOSTCONTEXT_RFF	LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_PWRON_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1122. Register Call Summary for Register RM_L3MAIN1_L3_MAIN_1_CONTEXT

Power Management Functional Description
PRCM Register Manual • CORE_PRM Register Summary: [2]

Table 3-1123. RM_L3MAIN1_GPMC_CONTEXT

Address Offset	0x0000 002C		
Physical Address	0x4AE0 672C	**Instance**	CORE_PRM
Description	This register contains dedicated GPMC context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														LOSTCONTEXT_RFF	RESERVED

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
0	RESERVED		R	0x0

Table 3-1124. Register Call Summary for Register RM_L3MAIN1_GPMC_CONTEXT

Power Management Functional Description
PRCM Register Manual • CORE_PRM Register Summary: [1]

Table 3-1125. RM_L3MAIN1_MMU_EDMA_CONTEXT

Address Offset	0x0000 0034		
Physical Address	0x4AE0 6734	**Instance**	CORE_PRM
Description	This register contains dedicated MMU context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														LOSTCONTEXT_RFF	RESERVED

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_PWRON_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
0	RESERVED		R	0x0

Table 3-1126. Register Call Summary for Register RM_L3MAIN1_MMU_EDMA_CONTEXT

Power Management Functional Description
PRCM Register Manual • CORE_PRM Register Summary: [1]

Table 3-1127. RM_L3MAIN1_MMU_PCIESS_CONTEXT

Address Offset	0x0000 004C		
Physical Address	0x4AE0 674C	**Instance**	CORE_PRM
Description	This register contains dedicated MMU context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														LOSTCONTEXT_RFF	RESERVED

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_PWRON_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
0	RESERVED		R	0x0

Table 3-1128. PM_L3MAIN1_OCMC_RAM1_WKDEP

Address Offset	0x0000 0050		
Physical Address	0x4AE0 6750	**Instance**	CORE_PRM
Description	This register controls wakeup dependency based on OCMC_RAM1 service requests.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						WKUPDEP_OCMC_RAM1_EVE4	WKUPDEP_OCMC_RAM1_EVE3	WKUPDEP_OCMC_RAM1_EVE2	WKUPDEP_OCMC_RAM1_EVE1	WKUPDEP_OCMC_RAM1_DSP2	WKUPDEP_OCMC_RAM1_IPU1	RESERVED	WKUPDEP_OCMC_RAM1_DSP1	WKUPDEP_OCMC_RAM1_IPU2	WKUPDEP_OCMC_RAM1_MPU

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	WKUPDEP_OCMC_RAM1_EVE4	Wakeup dependency from OCMC_RAM1 module (SWakeup signal) towards EVE4 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
8	WKUPDEP_OCMC_RAM1_EVE3	Wakeup dependency from OCMC_RAM1 module (SWakeup signal) towards EVE3 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
7	WKUPDEP_OCMC_RAM1_EVE2	Wakeup dependency from OCMC_RAM1 module (SWakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_OCMC_RAM1_EVE1	Wakeup dependency from OCMC_RAM1 module (SWakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_OCMC_RAM1_DSP2	Wakeup dependency from OCMC_RAM1 module (SWakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_OCMC_RAM1_IPU1	Wakeup dependency from OCMC_RAM1 module (SWakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	RESERVED		R	0x0
2	WKUPDEP_OCMC_RAM1_DSP1	Wakeup dependency from OCMC_RAM1 module (SWakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_OCMC_RAM1_IPU2	Wakeup dependency from OCMC_RAM1 module (SWakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
0	WKUPDEP_OCMC_RAM1_MPU	Wakeup dependency from OCMC_RAM1 module (SWakeup signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1129. Register Call Summary for Register PM_L3MAIN1_OCMC_RAM1_WKDEP

PRCM Register Manual
- Not Supported Functionality (Registers and Bitfields): [0][1][2][3][4]
- CORE_PRM Register Summary: [5]

Table 3-1130. RM_L3MAIN1_OCMC_RAM1_CONTEXT

Address Offset	0x0000 0054		
Physical Address	0x4AE0 6754	**Instance**	CORE_PRM
Description	This register contains dedicated OCMC_RAM context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							LOSTMEM_CORE_OCMRAM	RESERVED							LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	LOSTMEM_CORE_OCMRAM	Specify if memory-based context in CORE_OCMRAM memory bank has been lost due to a previous power transition or other reset source (not affected by a global warm reset). 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
7:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1131. Register Call Summary for Register RM_L3MAIN1_OCMC_RAM1_CONTEXT

Power Management Functional Description

PRCM Register Manual
- CORE_PRM Register Summary: [1]

Table 3-1132. PM_L3MAIN1_TESOC_WKDEP

Address Offset	0x0000 0058		
Physical Address	0x4AE0 6758	**Instance**	CORE_PRM
Description	This register controls wakeup dependency based on TESOC service requests.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						WKUPDEP_TESOC_EVE4	WKUPDEP_TESOC_EVE3	WKUPDEP_TESOC_EVE2	WKUPDEP_TESOC_EVE1	WKUPDEP_TESOC_DSP2	WKUPDEP_TESOC_IPU1	RESERVED	WKUPDEP_TESOC_DSP1	WKUPDEP_TESOC_IPU2	WKUPDEP_TESOC_MPU

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	WKUPDEP_TESOC_EVE4	Wakeup dependency from TESOC module (SWakeup signal) towards EVE4 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
8	WKUPDEP_TESOC_EVE3	Wakeup dependency from TESOC module (SWakeup signal) towards EVE3 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
7	WKUPDEP_TESOC_EVE2	Wakeup dependency from TESOC module (SWakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_TESOC_EVE1	Wakeup dependency from TESOC module (SWakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_TESOC_DSP2	Wakeup dependency from TESOC module (SWakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_TESOC_IPU1	Wakeup dependency from TESOC module (SWakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	RESERVED		R	0x0
2	WKUPDEP_TESOC_DSP1	Wakeup dependency from TESOC module (SWakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
1	WKUPDEP_TESOC_IPU2	Wakeup dependency from TESOC module (SWakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_TESOC_MPU	Wakeup dependency from TESOC module (SWakeup signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1133. Register Call Summary for Register PM_L3MAIN1_TESOC_WKDEP

PRCM Register Manual
- Not Supported Functionality (Registers and Bitfields): [0][1][2][3][4]
- CORE_PRM Register Summary: [5]

Table 3-1134. RM_L3MAIN1_TESOC_CONTEXT

Address Offset	0x0000 005C		
Physical Address	0x4AE0 675C	**Instance**	CORE_PRM
Description	This register contains dedicated TESOC context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							LOSTMEM_CORE_OCMRAM	RESERVED							LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	LOSTMEM_CORE_OCMRAM	Specify if memory-based context in CORE_OCMRAM memory bank has been lost due to a previous power transition or other reset source (not affected by a global warm reset). 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
7:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1135. Register Call Summary for Register RM_L3MAIN1_TESOC_CONTEXT

Power Management Functional Description
PRCM Register Manual • CORE_PRM Register Summary: [1]

Table 3-1136. PM_L3MAIN1_OCMC_RAM3_WKDEP

Address Offset	0x0000 0060		
Physical Address	0x4AE0 6760	**Instance**	CORE_PRM
Description	This register controls wakeup dependency based on OCMC_RAM3 service requests.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						WKUPDEP_OCMC_RAM3_EVE4	WKUPDEP_OCMC_RAM3_EVE3	WKUPDEP_OCMC_RAM3_EVE2	WKUPDEP_OCMC_RAM3_EVE1	WKUPDEP_OCMC_RAM3_DSP2	WKUPDEP_OCMC_RAM3_IPU1	RESERVED	WKUPDEP_OCMC_RAM3_DSP1	WKUPDEP_OCMC_RAM3_IPU2	WKUPDEP_OCMC_RAM3_MPU

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	WKUPDEP_OCMC_RAM3_EVE4	Wakeup dependency from OCMC_RAM3 module (SWakeup signal) towards EVE4 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
8	WKUPDEP_OCMC_RAM3_EVE3	Wakeup dependency from OCMC_RAM3 module (SWakeup signal) towards EVE3 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
7	WKUPDEP_OCMC_RAM3_EVE2	Wakeup dependency from OCMC_RAM3 module (SWakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_OCMC_RAM3_EVE1	Wakeup dependency from OCMC_RAM3 module (SWakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_OCMC_RAM3_DSP2	Wakeup dependency from OCMC_RAM3 module (SWakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
4	WKUPDEP_OCMC_RAM3_IPU1	Wakeup dependency from OCMC_RAM3 module (SWakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	RESERVED		R	0x0
2	WKUPDEP_OCMC_RAM3_DSP1	Wakeup dependency from OCMC_RAM3 module (SWakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_OCMC_RAM3_IPU2	Wakeup dependency from OCMC_RAM3 module (SWakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_OCMC_RAM3_MPU	Wakeup dependency from OCMC_RAM3 module (SWakeup signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1137. RM_L3MAIN1_OCMC_RAM3_CONTEXT

Address Offset	0x0000 0064		
Physical Address	0x4AE0 6764	**Instance**	CORE_PRM
Description	This register contains dedicated OCMC_RAM3 context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							LOSTMEM_CORE_OCMRAM	RESERVED							LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	LOSTMEM_CORE_OCMRAM	Specify if memory-based context in CORE_OCMRAM memory bank has been lost due to a previous power transition or other reset source (not affected by a global warm reset). 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
7:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1138. RM_L3MAIN1_OCMC_ROM_CONTEXT

Address Offset	0x0000 006C		
Physical Address	0x4AE0 676C	**Instance**	CORE_PRM
Description	This register contains dedicated OCMC_ROM context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							LOSTMEM_CORE_OCMROM	RESERVED							LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	LOSTMEM_CORE_OCMROM	Specify if memory-based context in CORE_OCMROM memory bank has been lost due to a previous power transition or other reset source (not affected by a global warm reset). 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
7:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1139. PM_L3MAIN1_TPCC_WKDEP

Address Offset	0x0000 0070		
Physical Address	0x4AE0 6770	**Instance**	CORE_PRM
Description	This register controls wakeup dependency based on TPCC service requests.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						WKUPDEP_TPCC_EVE4	WKUPDEP_TPCC_EVE3	WKUPDEP_TPCC_EVE2	WKUPDEP_TPCC_EVE1	WKUPDEP_TPCC_DSP2	WKUPDEP_TPCC_IPU1	RESERVED	WKUPDEP_TPCC_DSP1	WKUPDEP_TPCC_IPU2	WKUPDEP_TPCC_MPU

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	WKUPDEP_TPCC_EVE4	Wakeup dependency from TPCC module (SWakeup signal) towards EVE4 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
8	WKUPDEP_TPCC_EVE3	Wakeup dependency from TPCC module (SWakeup signal) towards EVE3 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
7	WKUPDEP_TPCC_EVE2	Wakeup dependency from TPCC module (SWakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_TPCC_EVE1	Wakeup dependency from TPCC module (SWakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_TPCC_DSP2	Wakeup dependency from TPCC module (SWakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_TPCC_IPU1	Wakeup dependency from TPCC module (SWakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	RESERVED		R	0x0
2	WKUPDEP_TPCC_DSP1	Wakeup dependency from TPCC module (SWakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_TPCC_IPU2	Wakeup dependency from TPCC module (SWakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_TPCC_MPU	Wakeup dependency from TPCC module (SWakeup signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1140. Register Call Summary for Register PM_L3MAIN1_TPCC_WKDEP

PRCM Register Manual

- Not Supported Functionality (Registers and Bitfields): [0][1][2][3][4]
- CORE_PRM Register Summary: [5]

Table 3-1141. RM_L3MAIN1_TPCC_CONTEXT

Address Offset	0x0000 0074		
Physical Address	0x4AE0 6774	**Instance**	CORE_PRM
Description	This register contains dedicated TPCC context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							LOSTMEM_TPCC_BANK	RESERVED						LOSTCONTEXT_RFF	RESERVED

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	LOSTMEM_TPCC_BANK	Specify if memory-based context in TPCC_MEM memory bank has been lost due to a previous power transition or other reset source (not affected by a global warm reset). 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
7:2	RESERVED		R	0x0
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
0	RESERVED		R	0x0

Table 3-1142. Register Call Summary for Register RM_L3MAIN1_TPCC_CONTEXT

Power Management Functional Description
PRCM Register Manual • CORE_PRM Register Summary: [1]

Table 3-1143. PM_L3MAIN1_TPTC1_WKDEP

Address Offset	0x0000 0078		
Physical Address	0x4AE0 6778	**Instance**	CORE_PRM
Description	This register controls wakeup dependency based on TPTC service requests.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						WKUPDEP_TPTC1_EVE4	WKUPDEP_TPTC1_EVE3	WKUPDEP_TPTC1_EVE2	WKUPDEP_TPTC1_EVE1	WKUPDEP_TPTC1_DSP2	WKUPDEP_TPTC1_IPU1	RESERVED	WKUPDEP_TPTC1_DSP1	WKUPDEP_TPTC1_IPU2	WKUPDEP_TPTC1_MPU

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	WKUPDEP_TPTC1_EVE4	Wakeup dependency from TPTC module (SWakeup signal) towards EVE4 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
8	WKUPDEP_TPTC1_EVE3	Wakeup dependency from TPTC module (SWakeup signal) towards EVE3 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
7	WKUPDEP_TPTC1_EVE2	Wakeup dependency from TPTC module (SWakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_TPTC1_EVE1	Wakeup dependency from TPTC module (SWakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_TPTC1_DSP2	Wakeup dependency from TPTC module (SWakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_TPTC1_IPU1	Wakeup dependency from TPTC module (SWakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	RESERVED		R	0x0
2	WKUPDEP_TPTC1_DSP1	Wakeup dependency from TPTC module (SWakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_TPTC1_IPU2	Wakeup dependency from TPTC module (SWakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
0	WKUPDEP_TPTC1_MPU	Wakeup dependency from TPTC module (SWakeup signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1144. Register Call Summary for Register PM_L3MAIN1_TPTC1_WKDEP

PRCM Register Manual
• Not Supported Functionality (Registers and Bitfields): [0][1][2][3][4] • CORE_PRM Register Summary: [5]

Table 3-1145. RM_L3MAIN1_TPTC1_CONTEXT

Address Offset	0x0000 007C		
Physical Address	0x4AE0 677C	**Instance**	CORE_PRM
Description	This register contains dedicated TPTC1 context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							LOSTMEM_TPTC_BANK	RESERVED						LOSTCONTEXT_RFF	RESERVED

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	LOSTMEM_TPTC_BANK	Specify if memory-based context in TPTC_MEM memory bank has been lost due to a previous power transition or other reset source (not affected by a global warm reset). 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
7:2	RESERVED		R	0x0
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
0	RESERVED		R	0x0

Table 3-1146. Register Call Summary for Register RM_L3MAIN1_TPTC1_CONTEXT

Power Management Functional Description
PRCM Register Manual
• CORE_PRM Register Summary: [1]

Table 3-1147. PM_L3MAIN1_TPTC2_WKDEP

Address Offset	0x0000 0080		
Physical Address	0x4AE0 6780	**Instance**	CORE_PRM
Description	This register controls wakeup dependency based on TPTC service requests.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						WKUPDEP_TPTC2_EVE4	WKUPDEP_TPTC2_EVE3	WKUPDEP_TPTC2_EVE2	WKUPDEP_TPTC2_EVE1	WKUPDEP_TPTC2_DSP2	WKUPDEP_TPTC2_IPU1	RESERVED	WKUPDEP_TPTC2_DSP1	WKUPDEP_TPTC2_IPU2	WKUPDEP_TPTC2_MPU

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	WKUPDEP_TPTC2_EVE4	Wakeup dependency from TPTC module (SWakeup signal) towards EVE4 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
8	WKUPDEP_TPTC2_EVE3	Wakeup dependency from TPTC module (SWakeup signal) towards EVE3 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
7	WKUPDEP_TPTC2_EVE2	Wakeup dependency from TPTC module (SWakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_TPTC2_EVE1	Wakeup dependency from TPTC module (SWakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_TPTC2_DSP2	Wakeup dependency from TPTC module (SWakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_TPTC2_IPU1	Wakeup dependency from TPTC module (SWakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	RESERVED		R	0x0
2	WKUPDEP_TPTC2_DSP1	Wakeup dependency from TPTC module (SWakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
1	WKUPDEP_TPTC2_IPU2	Wakeup dependency from TPTC module (SWakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_TPTC2_MPU	Wakeup dependency from TPTC module (SWakeup signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1148. Register Call Summary for Register PM_L3MAIN1_TPTC2_WKDEP

PRCM Register Manual
• Not Supported Functionality (Registers and Bitfields): [0][1][2][3][4] • CORE_PRM Register Summary: [5]

Table 3-1149. RM_L3MAIN1_TPTC2_CONTEXT

Address Offset	0x0000 0084		
Physical Address	0x4AE0 6784	**Instance**	CORE_PRM
Description	This register contains dedicated TPTC2 context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							LOSTMEM_TPTC_BANK	RESERVED						LOSTCONTEXT_RFF	RESERVED

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	LOSTMEM_TPTC_BANK	Specify if memory-based context in TPTC_MEM memory bank has been lost due to a previous power transition or other reset source (not affected by a global warm reset). 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
7:2	RESERVED		R	0x0
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
0	RESERVED		R	0x0

Table 3-1150. Register Call Summary for Register RM_L3MAIN1_TPTC2_CONTEXT

Power Management Functional Description
PRCM Register Manual
• CORE_PRM Register Summary: [1]

Table 3-1151. RM_L3MAIN1_VCP1_CONTEXT

Address Offset	0x0000 008C		
Physical Address	0x4AE0 678C	**Instance**	CORE_PRM
Description	This register contains dedicated VCP1 context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							LOSTMEM_VCP_BANK	RESERVED							LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	LOSTMEM_VCP_BANK	Specify if memory-based context in VCP memory bank has been lost due to a previous power transition or other reset source. 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
7:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1152. Register Call Summary for Register RM_L3MAIN1_VCP1_CONTEXT

Power Management Functional Description

PRCM Register Manual

- CORE_PRM Register Summary: [1]

Table 3-1153. RM_L3MAIN1_VCP2_CONTEXT

Address Offset	0x0000 0094		
Physical Address	0x4AE0 6794	**Instance**	CORE_PRM
Description	This register contains dedicated VCP2 context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							LOSTMEM_VCP_BANK	RESERVED							LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	LOSTMEM_VCP_BANK	Specify if memory-based context in VCP memory bank has been lost due to a previous power transition or other reset source. 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
7:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1154. Register Call Summary for Register RM_L3MAIN1_VCP2_CONTEXT

PRCM Register Manual
- CORE_PRM Register Summary: [0]

Table 3-1155. RM_L3MAIN1_SPARE_CME_CONTEXT

Address Offset	0x0000 009C		
Physical Address	0x4AE0 679C	**Instance**	CORE_PRM
Description	This register contains dedicated SPARE_CME context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1156. RM_L3MAIN1_SPARE_HDMI_CONTEXT

Address Offset	0x0000 00A4		
Physical Address	0x4AE0 67A4	**Instance**	CORE_PRM
Description	This register contains dedicated SPARE_HDMI context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1157. RM_L3MAIN1_SPARE_ICM_CONTEXT

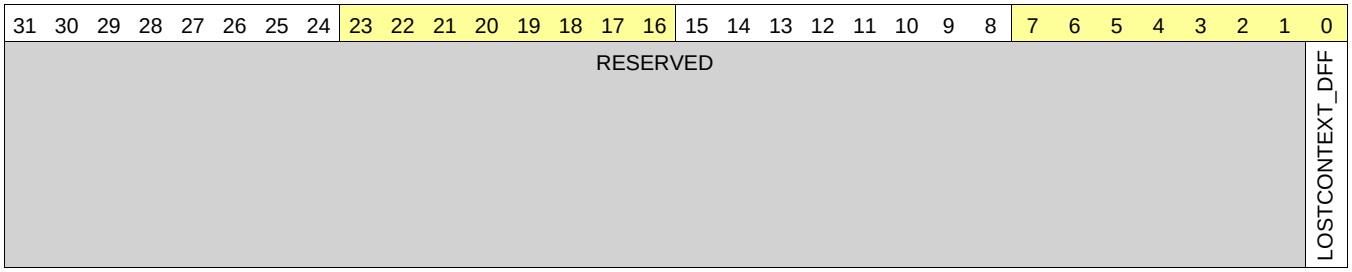
Address Offset	0x0000 00AC		
Physical Address	0x4AE0 67AC	**Instance**	CORE_PRM
Description	This register contains dedicated SPARE_ICM context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1158. RM_L3MAIN1_SPARE_IVA2_CONTEXT

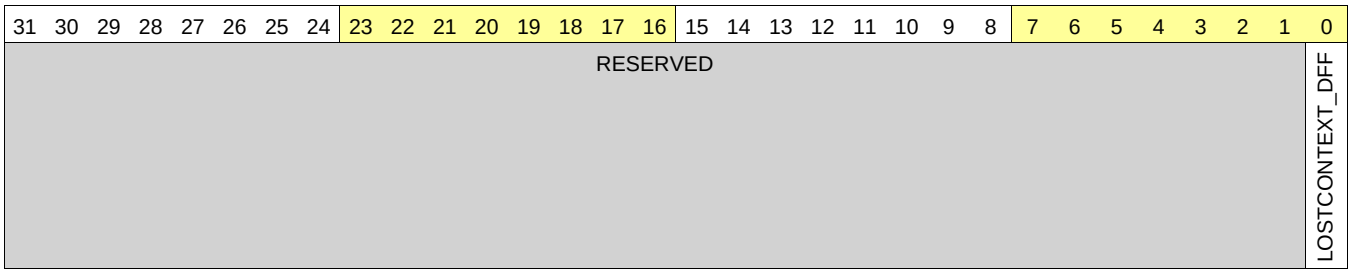
Address Offset	0x0000 00B4		
Physical Address	0x4AE0 67B4	**Instance**	CORE_PRM
Description	This register contains dedicated SPARE_IVA2 context statuses. [warm reset insensitive]		
Type	RW		



Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1159. RM_L3MAIN1_SPARE_SATA2_CONTEXT

Address Offset	0x0000 00BC		
Physical Address	0x4AE0 67BC	**Instance**	CORE_PRM
Description	This register contains dedicated SPARE_SATA2 context statuses. [warm reset insensitive]		
Type	RW		



Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1160. RM_L3MAIN1_SPARE_UNKNOWN4_CONTEXT

Address Offset	0x0000 00C4		
Physical Address	0x4AE0 67C4	**Instance**	CORE_PRM
Description	This register contains dedicated SPARE_UNKNOWN4 context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1161. RM_L3MAIN1_SPARE_UNKNOWN5_CONTEXT

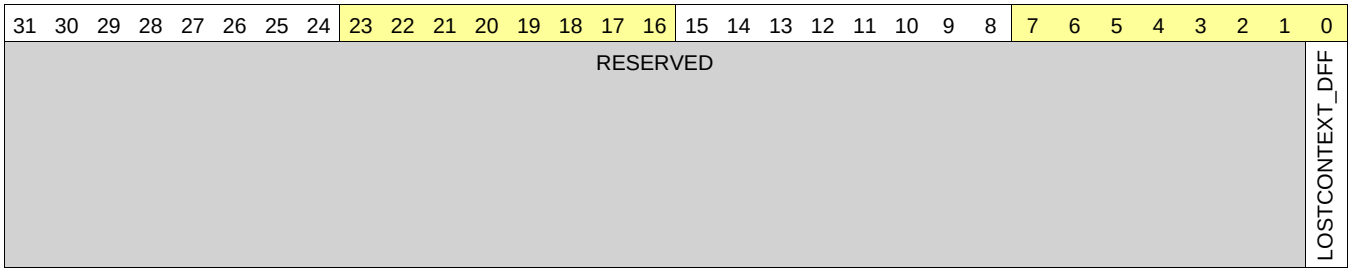
Address Offset	0x0000 00CC		
Physical Address	0x4AE0 67CC	**Instance**	CORE_PRM
Description	This register contains dedicated SPARE_UNKNOWN5 context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1162. RM_L3MAIN1_SPARE_UNKNOWN6_CONTEXT

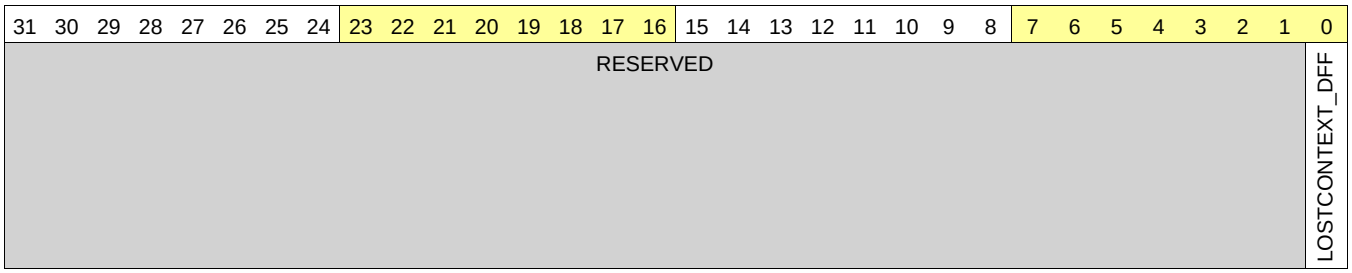
Address Offset	0x0000 00D4		
Physical Address	0x4AE0 67D4	**Instance**	CORE_PRM
Description	This register contains dedicated SPARE_UNKNOWN6 context statuses. [warm reset insensitive]		
Type	RW		



Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1163. RM_L3MAIN1_SPARE_VIDEOPLL1_CONTEXT

Address Offset	0x0000 00DC		
Physical Address	0x4AE0 67DC	**Instance**	CORE_PRM
Description	This register contains dedicated SPARE_VIDEOPLL1 context statuses. [warm reset insensitive]		
Type	RW		



Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1164. RM_L3MAIN1_SPARE_VIDEOPLL2_CONTEXT

Address Offset	0x0000 00F4		
Physical Address	0x4AE0 67F4	**Instance**	CORE_PRM
Description	This register contains dedicated SPARE_VIDEOPLL2 context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1165. RM_L3MAIN1_SPARE_VIDEOPLL3_CONTEXT

Address Offset	0x0000 00FC		
Physical Address	0x4AE0 67FC	**Instance**	CORE_PRM
Description	This register contains dedicated SPARE_VIDEOPLL3 context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1166. RM_IPU2_RSTCTRL

Address Offset	0x0000 0210		
Physical Address	0x4AE0 6910	**Instance**	CORE_PRM
Description	This register controls the release of the IPU2 sub-system resets.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																													RST_IPU	RST_CPU1	RST_CPU0

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0
2	RST_IPU	IPU system reset control. 0x0: Reset is cleared for IPU CACHE MMU 0x1: Reset is asserted for the IPU CACHE MMU	RW	0x1
1	RST_CPU1	IPU Cortex M3 CPU1 reset control 0x0: Reset is cleared for the IPU Cortex M3 CPU1 0x1: Reset is asserted for the IPU Cortex M3 CPU1	RW	0x1
0	RST_CPU0	IPU Cortex M3 CPU0 reset control. 0x0: Reset is cleared for the IPU Cortex M3 CPU0 0x1: Reset is asserted for the IPU Cortex M3 CPU0	RW	0x1

Table 3-1167. RM_IPU2_RSTST

Address Offset	0x0000 0214		
Physical Address	0x4AE0 6914	**Instance**	CORE_PRM
Description	This register logs the different reset sources of the IPU2 SS. Each bit is set upon release of the domain reset signal. Must be cleared by software. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								RST_ICECRUSHER_CPU1	RST_ICECRUSHER_CPU0	RST_EMULATION_CPU1	RST_EMULATION_CPU0	RST_IPU	RST_CPU1	RST_CPU0	

Bits	Field Name	Description	Type	Reset
31:7	RESERVED		R	0x0
6	RST_ICECRUSHER_CPU1	Cortex M3 CPU1 has been reset due to IPU ICECRUSHER1 reset source 0x0: No icecrusher reset 0x1: CPU1 has been reset upon icecrusher reset	RW	0x0
5	RST_ICECRUSHER_CPU0	Cortex M3 CPU0 has been reset due to IPU ICECRUSHER0 reset source 0x0: No icecrusher reset 0x1: CPU0 has been reset upon icecrusher reset	RW	0x0
4	RST_EMULATION_CPU1	Cortex M3 CPU1 has been reset due to emulation reset source e.g. assert reset command initiated by the icepick module 0x0: No emulation reset 0x1: CPU1 has been reset upon emulation reset	RW	0x0
3	RST_EMULATION_CPU0	Cortex M3 CPU0 has been reset due to emulation reset source e.g. assert reset command initiated by the icepick module 0x0: No emulation reset 0x1: CPU0 has been reset upon emulation reset	RW	0x0

Bits	Field Name	Description	Type	Reset
2	RST_IPU	IPU system SW reset status 0x0: No SW reset occurred 0x1: IPU MMU and CACHE interface has been reset upon SW reset	RW	0x0
1	RST_CPU1	IPU Cortex-M3 CPU1 SW reset status 0x0: No SW reset occurred 0x1: Cortex M3 CPU1 has been reset upon SW reset	RW	0x0
0	RST_CPU0	IPU Cortex-M3 CPU0 SW reset status 0x0: No SW reset occurred 0x1: Cortex M3 CPU0 has been reset upon SW reset	RW	0x0

Table 3-1168. RM_IPU2_IPU2_CONTEXT

Address Offset	0x0000 0224		
Physical Address	0x4AE0 6924	**Instance**	CORE_PRM
Description	This register contains dedicated BELLINI1 context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						LOSTMEM_IPU_L2RAM	LOSTMEM_IPU_UNICACHE	RESERVED						LOSTCONTEXT_RFF	LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	LOSTMEM_IPU_L2RAM	Specify if memory-based context in IPU_L2RAM memory bank has been lost due to a previous power transition or other reset source. 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
8	LOSTMEM_IPU_UNICACHE	Specify if memory-based context in IPU_UNICACHE memory bank has been lost due to a previous power transition or other reset source. 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
7:2	RESERVED		R	0x0
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of IPU_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of IPU_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1169. RM_DMA_DMA_SYSTEM_CONTEXT

Address Offset	0x0000 0324		
Physical Address	0x4AE0 6A24	**Instance**	CORE_PRM
Description	This register contains dedicated SDMA context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							LOSTMEM_CORE_OTHER_BANK	RESERVED						LOSTCONTEXT_RFF	RESERVED

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	LOSTMEM_CORE_OTHER_BANK	Specify if memory-based context in CORE_OTHER_BANK memory bank has been lost due to a previous power transition or other reset source. 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
7:2	RESERVED		R	0x0
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of DMA_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
0	RESERVED		R	0x0

Table 3-1170. RM_EMIF_DMM_CONTEXT

Address Offset	0x0000 0424		
Physical Address	0x4AE0 6B24	**Instance**	CORE_PRM
Description	This register contains dedicated DMM context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														LOSTCONTEXT_RFF	LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_PWRON_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_PWRON_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1171. RM_EMIF_EMIF_OCP_FW_CONTEXT

Address Offset	0x0000 042C		
Physical Address	0x4AE0 6B2C	**Instance**	CORE_PRM
Description	This register contains dedicated EMIF_OCP_FW context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														LOSTCONTEXT_RFF	LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_PWRON_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1172. Register Call Summary for Register RM_EMIF_EMIF_OCP_FW_CONTEXT

Power Management Functional Description
PRCM Register Manual • CORE_PRM Register Summary: [2]

Table 3-1173. RM_EMIF_EMIF1_CONTEXT

Address Offset	0x0000 0434		
Physical Address	0x4AE0 6B34	**Instance**	CORE_PRM
Description	This register contains dedicated EMIF_1 context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														LOSTCONTEXT_RFF	LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_PWRON_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_PWRON_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1174. Register Call Summary for Register RM_EMIF_EMIF1_CONTEXT

Power Management Functional Description
PRCM Register Manual • CORE_PRM Register Summary: [2]

Table 3-1175. RM_EMIF_EMIF2_CONTEXT

Address Offset	0x0000 043C		
Physical Address	0x4AE0 6B3C	**Instance**	CORE_PRM
Description	This register contains dedicated EMIF_2 context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														LOSTCONTEXT_RFF	LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_PWRON_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Bits	Field Name	Description	Type	Reset
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_PWRON_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1176. Register Call Summary for Register RM_EMIF_EMIF2_CONTEXT

PRCM Register Manual • CORE_PRM Register Summary: [0]

Table 3-1177. RM_EMIF_EMIF_DLL_CONTEXT

Address Offset	0x0000 0444		
Physical Address	0x4AE0 6B44	**Instance**	CORE_PRM
Description	This register contains dedicated DLL context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of DLL_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1178. Register Call Summary for Register RM_EMIF_EMIF_DLL_CONTEXT

Power Management Functional Description
PRCM Register Manual • CORE_PRM Register Summary: [1]

Table 3-1179. RM_CRC_CRC_CONTEXT

Address Offset	0x0000 0524		
Physical Address	0x4AE0 6C24	**Instance**	CORE_PRM
Description	This register contains dedicated CRC context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							LOSTMEM_CRC_BANK	RESERVED							LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	LOSTMEM_CRC_BANK	Specify if memory-based context in CRC_MEM memory bank has been lost due to a previous power transition or other reset source. 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
7:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1180. Register Call Summary for Register RM_CRC_CRC_CONTEXT

Power Management Functional Description
PRCM Register Manual • CORE_PRM Register Summary: [1]

Table 3-1181. RM_L4CFG_L4_CFG_CONTEXT

Address Offset	0x0000 0624		
Physical Address	0x4AE0 6D24	**Instance**	CORE_PRM
Description	This register contains dedicated L4_CFG context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														LOSTCONTEXT_RFF	LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_PWRON_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Bits	Field Name	Description	Type	Reset
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1182. Register Call Summary for Register RM_L4CFG_L4_CFG_CONTEXT

Power Management Functional Description
PRCM Register Manual • CORE_PRM Register Summary: [2]

Table 3-1183. RM_L4CFG_SPINLOCK_CONTEXT

Address Offset	0x0000 062C		
Physical Address	0x4AE0 6D2C	**Instance**	CORE_PRM
Description	This register contains dedicated HW_SEM context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														LOSTCONTEXT_RFF	RESERVED

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
0	RESERVED		R	0x0

Table 3-1184. Register Call Summary for Register RM_L4CFG_SPINLOCK_CONTEXT

Power Management Functional Description
PRCM Register Manual • CORE_PRM Register Summary: [1]

Table 3-1185. RM_L4CFG_MAILBOX1_CONTEXT

Address Offset	0x0000 0634		
Physical Address	0x4AE0 6D34	**Instance**	CORE_PRM
Description	This register contains dedicated MAILBOX1 context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														LOSTCONTEXT_RFF	RESERVED

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
0	RESERVED		R	0x0

Table 3-1186. Register Call Summary for Register RM_L4CFG_MAILBOX1_CONTEXT

Power Management Functional Description
PRCM Register Manual • CORE_PRM Register Summary: [1]

Table 3-1187. RM_L4CFG_SAR_ROM_CONTEXT

Address Offset	0x0000 063C		
Physical Address	0x4AE0 6D3C	**Instance**	CORE_PRM
Description	This register contains dedicated SAR_ROM context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1188. RM_L4CFG_OCP2SCP2_CONTEXT

Address Offset	0x0000 0644		
Physical Address	0x4AE0 6D44	**Instance**	CORE_PRM
Description	This register contains dedicated OCP2SCP2 context statuses. [warm reset insensitive]		

Table 3-1188. RM_L4CFG_OCP2SCP2_CONTEXT (continued)

Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1189. RM_L4CFG_MAILBOX2_CONTEXT

Address Offset	0x0000 064C		
Physical Address	0x4AE0 6D4C	**Instance**	CORE_PRM
Description	This register contains dedicated MAILBOX2 context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														LOSTCONTEXT_RFF	RESERVED

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
0	RESERVED		R	0x0

Table 3-1190. Register Call Summary for Register RM_L4CFG_MAILBOX2_CONTEXT

Power Management Functional Description
PRCM Register Manual • CORE_PRM Register Summary: [1]

Table 3-1191. RM_L4CFG_MAILBOX3_CONTEXT

Address Offset	0x0000 0654		
Physical Address	0x4AE0 6D54	**Instance**	CORE_PRM
Description	This register contains dedicated MAILBOX3 context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														LOSTCONTEXT_RFF	RESERVED

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
0	RESERVED		R	0x0

Table 3-1192. RM_L4CFG_MAILBOX4_CONTEXT

Address Offset	0x0000 065C		
Physical Address	0x4AE0 6D5C	**Instance**	CORE_PRM
Description	This register contains dedicated MAILBOX4 context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														LOSTCONTEXT_RFF	RESERVED

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
0	RESERVED		R	0x0

Table 3-1193. RM_L4CFG_MAILBOX5_CONTEXT

Address Offset	0x0000 0664		
Physical Address	0x4AE0 6D64	**Instance**	CORE_PRM
Description	This register contains dedicated MAILBOX5 context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														LOSTCONTEXT_RFF	RESERVED

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
0	RESERVED		R	0x0

Table 3-1194. RM_L4CFG_MAILBOX6_CONTEXT

Address Offset	0x0000 066C		
Physical Address	0x4AE0 6D6C	**Instance**	CORE_PRM
Description	This register contains dedicated MAILBOX6 context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														LOSTCONTEXT_RFF	RESERVED

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
0	RESERVED		R	0x0

Table 3-1195. RM_L4CFG_MAILBOX7_CONTEXT

Address Offset	0x0000 0674		
Physical Address	0x4AE0 6D74	**Instance**	CORE_PRM
Description	This register contains dedicated MAILBOX7 context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														LOSTCONTEXT_RFF	RESERVED

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
0	RESERVED		R	0x0

Table 3-1196. RM_L4CFG_MAILBOX8_CONTEXT

Address Offset	0x0000 067C		
Physical Address	0x4AE0 6D7C	**Instance**	CORE_PRM
Description	This register contains dedicated MAILBOX8 context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														LOSTCONTEXT_RFF	RESERVED

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
0	RESERVED		R	0x0

Table 3-1197. RM_L4CFG_MAILBOX9_CONTEXT

Address Offset	0x0000 0684		
Physical Address	0x4AE0 6D84	**Instance**	CORE_PRM
Description	This register contains dedicated MAILBOX9 context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														LOSTCONTEXT_RFF	RESERVED

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
0	RESERVED		R	0x0

Table 3-1198. RM_L4CFG_MAILBOX10_CONTEXT

Address Offset	0x0000 068C		
Physical Address	0x4AE0 6D8C	**Instance**	CORE_PRM
Description	This register contains dedicated MAILBOX10 context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														LOSTCONTEXT_RFF	RESERVED

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
0	RESERVED		R	0x0

Table 3-1199. RM_L4CFG_MAILBOX11_CONTEXT

Address Offset	0x0000 0694		
Physical Address	0x4AE0 6D94	**Instance**	CORE_PRM
Description	This register contains dedicated MAILBOX11 context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														LOSTCONTEXT_RFF	RESERVED

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
0	RESERVED		R	0x0

Table 3-1200. RM_L4CFG_MAILBOX12_CONTEXT

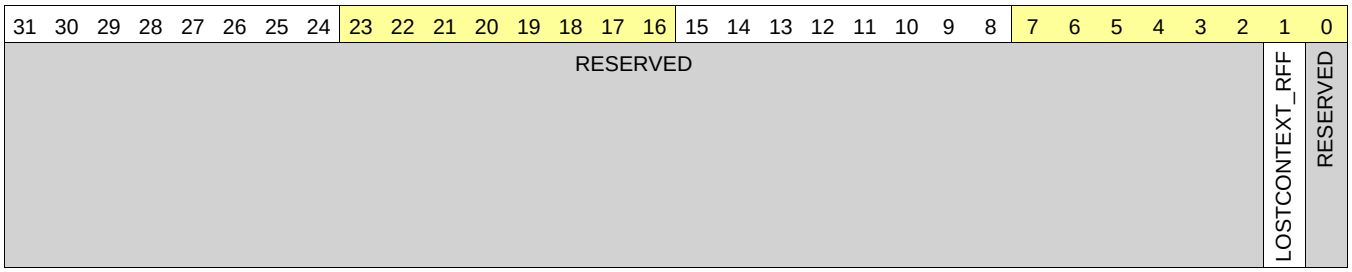
Address Offset	0x0000 069C		
Physical Address	0x4AE0 6D9C	**Instance**	CORE_PRM
Description	This register contains dedicated MAILBOX12 context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														LOSTCONTEXT_RFF	RESERVED

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
0	RESERVED		R	0x0

Table 3-1201. RM_L4CFG_MAILBOX13_CONTEXT

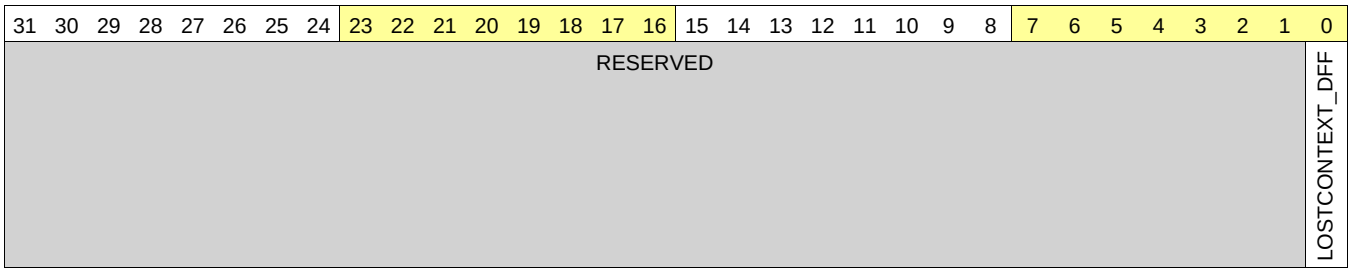
Address Offset	0x0000 06A4		
Physical Address	0x4AE0 6DA4	**Instance**	CORE_PRM
Description	This register contains dedicated MAILBOX13 context statuses. [warm reset insensitive]		
Type	RW		



Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
0	RESERVED		R	0x0

Table 3-1202. RM_L4CFG_SPARE_SMARTREFLEX_RTC_CONTEXT

Address Offset	0x0000 06AC		
Physical Address	0x4AE0 6DAC	**Instance**	CORE_PRM
Description	This register contains dedicated SPARE_SMARTREFLEX_RTC context statuses. [warm reset insensitive]		
Type	RW		



Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1203. RM_L4CFG_SPARE_SMARTREFLEX_SDRAM_CONTEXT

Address Offset	0x0000 06B4		
Physical Address	0x4AE0 6DB4	**Instance**	CORE_PRM
Description	This register contains dedicated SPARE_SMARTREFLEX_SDRAM context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1204. RM_L4CFG_SPARE_SMARTREFLEX_WKUP_CONTEXT

Address Offset	0x0000 06BC		
Physical Address	0x4AE0 6DBC	**Instance**	CORE_PRM
Description	This register contains dedicated SPARE_SMARTREFLEX_WKUP context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1205. RM_L4CFG_IO_DELAY_BLOCK_CONTEXT

Address Offset	0x0000 06C4		
Physical Address	0x4AE0 6DC4	**Instance**	CORE_PRM
Description	This register contains dedicated IO_DELAY_BLOCK context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1206. RM_L3INSTR_L3_MAIN_2_CONTEXT

Address Offset	0x0000 0724		
Physical Address	0x4AE0 6E24	**Instance**	CORE_PRM
Description	This register contains dedicated L3_3 context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														LOSTCONTEXT_RFF	LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_PWRON_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1207. Register Call Summary for Register RM_L3INSTR_L3_MAIN_2_CONTEXT

Power Management Functional Description
PRCM Register Manual • CORE_PRM Register Summary: [2]

Table 3-1208. RM_L3INSTR_L3_INSTR_CONTEXT

Address Offset	0x0000 072C		
Physical Address	0x4AE0 6E2C	**Instance**	CORE_PRM
Description	This register contains dedicated L3_INSTR context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1209. Register Call Summary for Register RM_L3INSTR_L3_INSTR_CONTEXT

Power Management Functional Description
PRCM Register Manual • CORE_PRM Register Summary: [1]

Table 3-1210. RM_L3INSTR_OCP_WP_NOC_CONTEXT

Address Offset	0x0000 0744		
Physical Address	0x4AE0 6E44	**Instance**	CORE_PRM
Description	This register contains dedicated OCP_WP1 context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							LOSTMEM_CORE_NRET_BANK	RESERVED						LOSTCONTEXT_RFF	LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	LOSTMEM_CORE_NRET_BANK	Specify if memory-based context in CORE_NRET_BANK memory bank has been lost due to a previous power transition or other reset source. 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
7:2	RESERVED		R	0x0
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_PWRON_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1211. Register Call Summary for Register RM_L3INSTR_OCP_WP_NOC_CONTEXT

Power Management Functional Description
PRCM Register Manual • CORE_PRM Register Summary: [2]

3.12.34 CUSTEFUSE_PRM registers

3.12.34.1 CUSTEFUSE_PRM Register Summary

Table 3-1212. CUSTEFUSE_PRM Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	CUSTEFUSE_PRM Physical Address
PM_CUSTEFUSE_PWRSTCTRL	RW	32	0x0000 0000	0x4AE0 7600
PM_CUSTEFUSE_PWRSTST	RW	32	0x0000 0004	0x4AE0 7604
RM_CUSTEFUSE_EFUSE_CTRL_CUST_CONTEXT	RW	32	0x0000 0024	0x4AE0 7624

3.12.34.2 CUSTEFUSE_PRM Register Description

Table 3-1213. PM_CUSTEFUSE_PWRSTCTRL

Address Offset	0x0000 0000		
Physical Address	0x4AE0 7600	**Instance**	CUSTEFUSE_PRM
Description	This register controls the CUSTEFUSE power state to reach upon a domain sleep transition		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																											LOWPOWERSTATECHANGE	RESERVED		POWERSTATE	

Bits	Field Name	Description	Type	Reset
31:5	RESERVED		R	0x0
4	LOWPOWERSTATECHANGE	Power state change request when domain has already performed a sleep transition. Allows going into deeper low power state without waking up the power domain. 0x0: Do not request a low power state change. 0x1: Request a low power state change. This bit is automatically cleared when the power state is effectively changed or when power state is ON.	RW	0x0
3:2	RESERVED		R	0x0
1:0	POWERSTATE	Power state control 0x0: OFF state 0x1: Reserved 0x2: INACTIVE state 0x3: ON State	RW	0x0

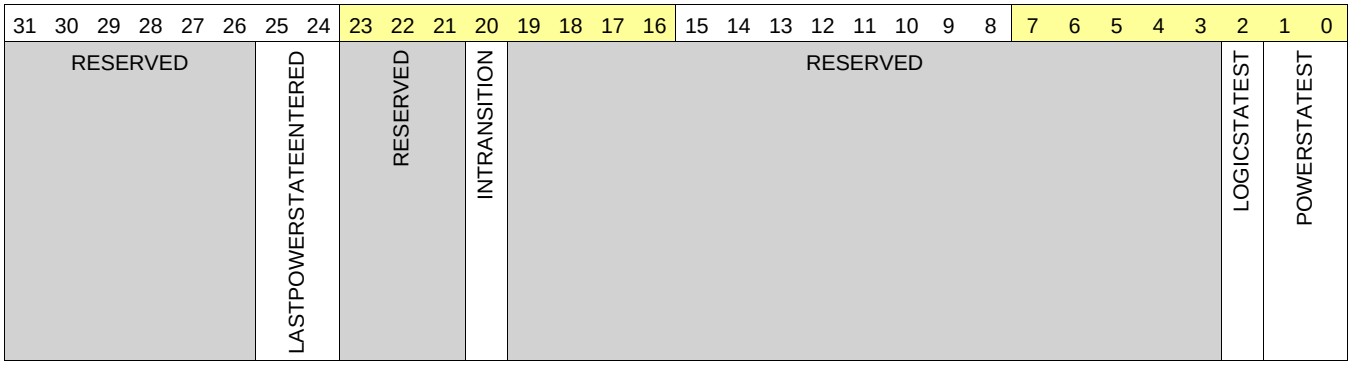
Table 3-1214. Register Call Summary for Register PM_CUSTEFUSE_PWRSTCTRL

Power Management Functional Description
- Logic and Memory Area Power Modes Control and Status: [0][1]

PRCM Register Manual
- Not Supported Functionality (Registers and Bitfields): [2]
- CUSTEFUSE_PRM Register Summary: [3]

Table 3-1215. PM_CUSTEFUSE_PWRSTST

Address Offset	0x0000 0004		
Physical Address	0x4AE0 7604	**Instance**	CUSTEFUSE_PRM
Description	This register provides a status on the current CUSTEFUSE power domain state. [warm reset insensitive]		
Type	RW		



Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:24	LASTPOWERSTATEENTERED	Last low power state entered. Set to 0x3 upon write of the same only. This register is intended for debug purpose only. 0x0: Power domain was previously OFF 0x1: Power domain was previously in RETENTION 0x2: Power domain was previously ON-INACTIVE 0x3: Power domain was previously ON-ACTIVE	RW	0x0
23:21	RESERVED		R	0x0
20	INTRANSITION	Domain transition status 0x0: No on-going transition on power domain 0x1: Power domain transition is in progress.	R	0x0
19:3	RESERVED		R	0x0
2	LOGICSTATEST	Logic state status 0x0: Logic in domain is OFF 0x1: Logic in domain is ON	R	0x1
1:0	POWERSTATEST	Current power state status 0x0: Power domain is OFF 0x1: Power domain is in RETENTION 0x2: Power domain is ON-INACTIVE 0x3: Power domain is ON-ACTIVE	R	0x3

Table 3-1216. Register Call Summary for Register PM_CUSTEFUSE_PWRSTST

Power Management Functional Description
- Logic and Memory Area Power Modes Control and Status: [0][1][2][3]

PRCM Register Manual
- Not Supported Functionality (Registers and Bitfields): [4][5][6][7][8]
- CUSTEFUSE_PRM Register Summary: [9]

Table 3-1217. RM_CUSTEFUSE_EFUSE_CTRL_CUST_CONTEXT

Address Offset	0x0000 0024		
Physical Address	0x4AE0 7624	**Instance**	CUSTEFUSE_PRM
Description	This register contains dedicated CUSTEFUSE module context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CUSTEFUSE_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1218. Register Call Summary for Register RM_CUSTEFUSE_EFUSE_CTRL_CUST_CONTEXT

Power Management Functional Description • PD_CUSTEFUSE Description: [0]
PRCM Register Manual • CUSTEFUSE_PRM Register Summary: [1]

3.12.35 DEVICE_PRM registers

3.12.35.1 DEVICE_PRM Register Summary

Table 3-1219. DEVICE_PRM Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	DEVICE_PRM Physical Address
PRM_RSTCTRL	RW	32	0x0000 0000	0x4AE0 7D00
PRM_RSTST	RW	32	0x0000 0004	0x4AE0 7D04
PRM_RSTTIME	RW	32	0x0000 0008	0x4AE0 7D08
PRM_CLKREQCTRL	RW	32	0x0000 000C	0x4AE0 7D0C
PRM_VOLTCTRL	RW	32	0x0000 0010	0x4AE0 7D10
PRM_PWRREQCTRL	RW	32	0x0000 0014	0x4AE0 7D14
PRM_PSCON_COUNT	RW	32	0x0000 0018	0x4AE0 7D18
PRM_IO_COUNT	RW	32	0x0000 001C	0x4AE0 7D1C
PRM_IO_PMCTRL	RW	32	0x0000 0020	0x4AE0 7D20
PRM_VOLTSETUP_WARMRESET	RW	32	0x0000 0024	0x4AE0 7D24
PRM_VOLTSETUP_CORE_OFF	RW	32	0x0000 0028	0x4AE0 7D28
PRM_VOLTSETUP_MPU_OFF	RW	32	0x0000 002C	0x4AE0 7D2C
PRM_VOLTSETUP_MM_OFF	RW	32	0x0000 0030	0x4AE0 7D30
PRM_VOLTSETUP_CORE_RET_SLEEP	RW	32	0x0000 0034	0x4AE0 7D34
PRM_VOLTSETUP_MPU_RET_SLEEP	RW	32	0x0000 0038	0x4AE0 7D38
PRM_VOLTSETUP_MM_RET_SLEEP	RW	32	0x0000 003C	0x4AE0 7D3C
PRM_VP_CORE_CONFIG	RW	32	0x0000 0040	0x4AE0 7D40
PRM_VP_CORE_STATUS	R	32	0x0000 0044	0x4AE0 7D44
PRM_VP_CORE_VLIMITTO	RW	32	0x0000 0048	0x4AE0 7D48
PRM_VP_CORE_VOLTAGE	RW	32	0x0000 004C	0x4AE0 7D4C

Table 3-1219. DEVICE_PRM Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	DEVICE_PRM Physical Address
PRM_VP_CORE_VSTEPMAX	RW	32	0x0000 0050	0x4AE0 7D50
PRM_VP_CORE_VSTEPMIN	RW	32	0x0000 0054	0x4AE0 7D54
PRM_VP_MPU_CONFIG	RW	32	0x0000 0058	0x4AE0 7D58
PRM_VP_MPU_STATUS	R	32	0x0000 005C	0x4AE0 7D5C
PRM_VP_MPU_VLIMITTO	RW	32	0x0000 0060	0x4AE0 7D60
PRM_VP_MPU_VOLTAGE	RW	32	0x0000 0064	0x4AE0 7D64
PRM_VP_MPU_VSTEPMAX	RW	32	0x0000 0068	0x4AE0 7D68
PRM_VP_MPU_VSTEPMIN	RW	32	0x0000 006C	0x4AE0 7D6C
PRM_VP_MM_CONFIG	RW	32	0x0000 0070	0x4AE0 7D70
PRM_VP_MM_STATUS	R	32	0x0000 0074	0x4AE0 7D74
PRM_VP_MM_VLIMITTO	RW	32	0x0000 0078	0x4AE0 7D78
PRM_VP_MM_VOLTAGE	RW	32	0x0000 007C	0x4AE0 7D7C
PRM_VP_MM_VSTEPMAX	RW	32	0x0000 0080	0x4AE0 7D80
PRM_VP_MM_VSTEPMIN	RW	32	0x0000 0084	0x4AE0 7D84
PRM_VC_SMPS_CORE_CONFIG	RW	32	0x0000 0088	0x4AE0 7D88
PRM_VC_SMPS_MM_CONFIG	RW	32	0x0000 008C	0x4AE0 7D8C
PRM_VC_SMPS_MPU_CONFIG	RW	32	0x0000 0090	0x4AE0 7D90
PRM_VC_VAL_CMD_VDD_CORE_L	RW	32	0x0000 0094	0x4AE0 7D94
PRM_VC_VAL_CMD_VDD_MM_L	RW	32	0x0000 0098	0x4AE0 7D98
PRM_VC_VAL_CMD_VDD_MPU_L	RW	32	0x0000 009C	0x4AE0 7D9C
PRM_VC_VAL_BYPASS	RW	32	0x0000 00A0	0x4AE0 7DA0
PRM_VC_CORE_ERRST	RW	32	0x0000 00A4	0x4AE0 7DA4
PRM_VC_MM_ERRST	RW	32	0x0000 00A8	0x4AE0 7DA8
PRM_VC_MPU_ERRST	RW	32	0x0000 00AC	0x4AE0 7DAC
PRM_VC_BYPASS_ERRST	RW	32	0x0000 00B0	0x4AE0 7DB0
PRM_VC_CFG_I2C_MODE	RW	32	0x0000 00B4	0x4AE0 7DB4
PRM_VC_CFG_I2C_CLK	RW	32	0x0000 00B8	0x4AE0 7DB8
PRM_SRAM_COUNT	RW	32	0x0000 00BC	0x4AE0 7DBC
PRM_SRAM_WKUP_SETUP	RW	32	0x0000 00C0	0x4AE0 7DC0
PRM_SLDO_CORE_SETUP	RW	32	0x0000 00C4	0x4AE0 7DC4
PRM_SLDO_CORE_CTRL	R	32	0x0000 00C8	0x4AE0 7DC8
PRM_SLDO_MPU_SETUP	RW	32	0x0000 00CC	0x4AE0 7DCC
PRM_SLDO_MPU_CTRL	RW	32	0x0000 00D0	0x4AE0 7DD0
PRM_SLDO_GPU_SETUP	RW	32	0x0000 00D4	0x4AE0 7DD4
PRM_SLDO_GPU_CTRL	RW	32	0x0000 00D8	0x4AE0 7DD8
PRM_ABBLDO_MPU_SETUP	RW	32	0x0000 00DC	0x4AE0 7DDC
PRM_ABBLDO_MPU_CTRL	RW	32	0x0000 00E0	0x4AE0 7DE0
PRM_ABBLDO_GPU_SETUP	RW	32	0x0000 00E4	0x4AE0 7DE4
PRM_ABBLDO_GPU_CTRL	RW	32	0x0000 00E8	0x4AE0 7DE8
PRM_BANDGAP_SETUP	RW	32	0x0000 00EC	0x4AE0 7DEC
PRM_DEVICE_OFF_CTRL	RW	32	0x0000 00F0	0x4AE0 7DF0
PRM_PHASE1_CNDP	R	32	0x0000 00F4	0x4AE0 7DF4
PRM_PHASE2A_CNDP	R	32	0x0000 00F8	0x4AE0 7DF8
PRM_PHASE2B_CNDP	R	32	0x0000 00FC	0x4AE0 7DFC
PRM_MODEM_IF_CTRL	RW	32	0x0000 0100	0x4AE0 7E00
PRM_VOLTST_MPU	R	32	0x0000 0110	0x4AE0 7E10
PRM_VOLTST_MM	R	32	0x0000 0114	0x4AE0 7E14

Table 3-1219. DEVICE_PRM Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	DEVICE_PRM Physical Address
PRM_SLDO_DSPEVE_SETUP	RW	32	0x0000 0118	0x4AE0 7E18
PRM_SLDO_IVA_SETUP	RW	32	0x0000 011C	0x4AE0 7E1C
PRM_ABBLDO_DSPEVE_CTRL	RW	32	0x0000 0120	0x4AE0 7E20
PRM_ABBLDO_IVA_CTRL	RW	32	0x0000 0124	0x4AE0 7E24
PRM_SLDO_DSPEVE_CTRL	RW	32	0x0000 0128	0x4AE0 7E28
PRM_SLDO_IVA_CTRL	RW	32	0x0000 012C	0x4AE0 7E2C
PRM_ABBLDO_DSPEVE_SETUP	RW	32	0x0000 0130	0x4AE0 7E30
PRM_ABBLDO_IVA_SETUP	RW	32	0x0000 0134	0x4AE0 7E34

3.12.35.2 DEVICE_PRM Register Description

Table 3-1220. PRM_RSTCTRL

Address Offset	0x0000 0000		
Physical Address	0x4AE0 7D00	**Instance**	DEVICE_PRM
Description	Global software cold and warm reset control. This register is auto-cleared. Only write 1 is possible. A read returns 0 only.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														RST_GLOBAL_COLD_SW	RST_GLOBAL_WARM_SW

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1	RST_GLOBAL_COLD_SW	Global COLD software reset control. This bit is reset only upon a global cold source of reset. 0x0: Global COLD software reset is cleared. 0x1: Triggers a global COLD software reset. The software must ensure the SDRAM is properly put in sef-refresh mode before applying this reset.	RW	0x0
0	RST_GLOBAL_WARM_SW	Global WARM software reset control. This bit is reset upon any global source of reset (warm and cold). 0x0: Global warm software reset is cleared. 0x1: Triggers a global warm software reset.	RW	0x0

Table 3-1221. Register Call Summary for Register PRM_RSTCTRL

Reset Management Functional Description • Global Reset Sources: [0][1]
PRCM Register Manual • DEVICE_PRM Register Summary: [2]

Table 3-1222. PRM_RSTST

Address Offset	0x0000 0004		
Physical Address	0x4AE0 7D04	**Instance**	DEVICE_PRM
Description	This register logs the global reset sources. Each bit is set upon release of the domain reset signal. Must be cleared by software. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED															TSHUT_IVA_RST	TSHUT_DSPEVE_RST	LLI_RST	TSHUT_CORE_RST	TSHUT_MM_RST	TSHUT_MPU_RST	C2C_RST	ICEPICK_RST	VDD_CORE_VOLT_MGR_RST	VDD_MM_VOLT_MGR_RST	VDD_MPU_VOLT_MGR_RST	EXTERNAL_WARM_RST	RESERVED	MPU_WDT_RST	RESERVED	GLOBAL_WARM_SW_RST	GLOBAL_COLD_RST

Bits	Field Name	Description	Type	Reset
31:17	RESERVED		R	0x0
16	TSHUT_IVA_RST	TSHUT_IVA warm reset event. This is a source of global WARM reset. 0x0: No TSHUT_MM reset. 0x1: TSHUT_MM reset has occurred.	RW	0x0
15	TSHUT_DSPEVE_RST	TSHUT_DSPEVE warm reset event. This is a source of global WARM reset. 0x0: No TSHUT_MM reset. 0x1: TSHUT_MM reset has occurred.	RW	0x0
14	LLI_RST	LLI warm reset event. This is a source of global WARM reset. 0x0: No LLI warm reset. 0x1: LLI warm reset has occurred.	RW	0x0
13	TSHUT_CORE_RST	TSHUT_CORE warm reset event. This is a source of global WARM reset. 0x0: No TSHUT_CORE reset. 0x1: TSHUT_CORE reset has occurred.	RW	0x0
12	TSHUT_MM_RST	TSHUT_GPU warm reset event. This is a source of global WARM reset. 0x0: No TSHUT_MM reset. 0x1: TSHUT_MM reset has occurred.	RW	0x0
11	TSHUT_MPU_RST	TSHUT_MPU warm reset event. This is a source of global WARM reset. 0x0: No TSHUT_MPU reset. 0x1: TSHUT_MPU reset has occurred.	RW	0x0
10	C2C_RST	C2C warm reset event. This is a source of global WARM reset. 0x0: No C2C warm reset. 0x1: C2C warm reset has occurred.	RW	0x0
9	ICEPICK_RST	IcePick reset event. This is a source of global warm reset initiated by the emulation. 0x0: No ICEPICK reset. 0x1: IcePick reset has occurred.	RW	0x0

Bits	Field Name	Description	Type	Reset
8	VDD_CORE_VOLT_MGR_RST	VDD_CORE voltage manager reset event This is a source of global WARM reset. 0x0: No VDD_CORE voltage manager reset. 0x1: VDD_CORE voltage manager reset has occurred.	RW	0x0
7	VDD_MM_VOLT_MGR_RST	VDD_MM voltage manager reset event This is a source of global WARM reset. 0x0: No VDD_MM voltage manager reset. 0x1: VDD_MM voltage manager reset has occurred.	RW	0x0
6	VDD_MPU_VOLT_MGR_RST	VDD_MPU voltage manager reset event This is a source of global WARM reset. 0x0: No VDD_MPU voltage manager reset. 0x1: VDD_MPU voltage manager reset has occurred.	RW	0x0
5	EXTERNAL_WARM_RST	External warm reset event 0x0: No global warm reset. 0x1: Global external warm reset has occurred.	RW	0x0
4	RESERVED	Reserved	RW	0x0
3	MPU_WDT_RST	MPU Watchdog timer reset event. This is a source of global WARM reset. 0x0: No MPU watchdog reset. 0x1: MPU wachtdog reset has occurred.	RW	0x0
2	RESERVED	Reserved	RW	0x0
1	GLOBAL_WARM_SW_RST	Global warm software reset event 0x0: No global warm SW reset 0x1: Global warm SW reset has occurred.	RW	0x0
0	GLOBAL_COLD_RST	Power-on (cold) reset event 0x0: No power-on reset. 0x1: Power-on reset has occurred.	RW	0x1

Table 3-1223. Register Call Summary for Register PRM_RSTST

Reset Management Functional Description
- Reset Logging: [0][1]

PRCM Register Manual
- Not Supported Functionality (Registers and Bitfields): [2][3][4][5][6][7][8][9]
- DEVICE_PRM Register Summary: [10]

Table 3-1224. PRM_RSTTIME

Address Offset	0x0000 0008		
Physical Address	0x4AE0 7D08	**Instance**	DEVICE_PRM
Description	Reset duration control. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RSTTIME2								RSTTIME1							

Bits	Field Name	Description	Type	Reset
31:15	RESERVED		R	0x0
14:10	RSTTIME2	Power domain reset duration 2 in number of RM.SYSCLK clock cycles. 0x0: Reserved	RW	0x10

Bits	Field Name	Description	Type	Reset
9:0	RSTTIME1	Global reset duration 1 in number of Func_32k_clk clock cycles. This bit-field is only sensitive to the external power-on reset (WKUPAON_SYS_PWRON_RST reset line) 0x0: Reserved	RW	0x6

Table 3-1225. Register Call Summary for Register PRM_RSTTIME

Reset Management Functional Description
- Reset Domains: [0]
- IPU1 Subsystem Power-On Reset Sequence: [1][2]
- Global Warm Reset Sequence: [3]

Device Low-Power States
- Wakeup Upon Global Warm Reset: [4]

PRCM Register Manual
- DEVICE_PRM Register Summary: [5]

Table 3-1226. PRM_CLKREQCTRL

Address Offset	0x0000 000C		
Physical Address	0x4AE0 7D0C	**Instance**	DEVICE_PRM
Description	This register allows controlling the CLKREQ signal towards SCRM.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														CLKREQ_COND	

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0
2:0	CLKREQ_COND	Control upon which condition CLKREQ signal is de-asserted. 0x0: CLKREQ is never de-asserted 0x1: CLKREQ is de-asserted when system clock is not required by any function in the device and if all voltage domains are in OFF state. 0x2: CLKREQ is de-asserted when system clock is not required by any function in the device and if all voltage domains are in RET or OFF state. 0x3: CLKREQ is de-asserted when system clock is not required by any function in the device and if all voltage domains are in SLEEP or RET or OFF state. 0x4: CLKREQ is de-asserted when system clock is not required by any function in the device. This is designed for low-power use-cases using the DPLL cascading scheme (ex: MP3) 0x5: Reserved 0x6: Reserved 0x7: Reserved	RW	0x0

Table 3-1227. PRM_VOLTCTRL

Address Offset	0x0000 0010		
Physical Address	0x4AE0 7D10	**Instance**	DEVICE_PRM
Description	This register provides voltage domain management controls.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																VDD_MM_I2C_DISABLE	VDD_MPU_I2C_DISABLE	VDD_CORE_I2C_DISABLE	RESERVED			VDD_MM_PRESENCE	VDD_MPU_PRESENCE	RESERVED		AUTO_CTRL_VDD_MM_L		AUTO_CTRL_VDD_MPU_L		AUTO_CTRL_VDD_CORE_L	

Bits	Field Name	Description	Type	Reset
31:15	RESERVED		R	0x0
14	VDD_MM_I2C_DISABLE	This bit allows disabling I2C interface with powerIC for MM voltage (for debug purpose only). [warm reset insensitive] 0x0: Normal mode: I2C is enabled. 0x1: Debug mode: I2C is disabled.	RW	0x0
13	VDD_MPU_I2C_DISABLE	This bit allows disabling I2C interface with powerIC for MPU voltage (for debug purpose only). [warm reset insensitive] 0x0: Normal mode: I2C is enabled. 0x1: Debug mode: I2C is disabled.	RW	0x0
12	VDD_CORE_I2C_DISABLE	This bit allows disabling I2C interface with powerIC for CORE voltage (for debug purpose only). [warm reset insensitive] 0x0: Normal mode: I2C is enabled. 0x1: Debug mode: I2C is disabled.	RW	0x0
11:10	RESERVED		R	0x0
9	VDD_MM_PRESENCE	This bit control the presence of MM voltage in device. [warm reset insensitive] 0x0: MM voltage is not present as an individual voltage: MM voltage is merged with MPU voltage if VDD_MPU_presence=1. MM voltage is merged with CORE voltage if VDD_MPU_presence=0. 0x1: MM voltage is present on the device.	RW	0x1
8	VDD_MPU_PRESENCE	This bit control the presence of MPU voltage in device. [warm reset insensitive] 0x0: MPU voltage is not present as an individual voltage: MPU voltage is merged with MM voltage if VDD_MM_presence=1. MPU voltage is merged with CORE voltage if VDD_MM_presence=0. 0x1: MPU voltage is present on the device.	RW	0x1
7:6	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
5:4	AUTO_CTRL_VDD_MM_L	This bit field specifies the state to which the hardware can automatically transition the VDD_MM_L voltage domain. 0x0: Voltage domain transitions are disabled. 0x1: Voltage domain transitions to SLEEP are enabled. 0x2: Voltage domain transitions to RET are enabled. 0x3: reserved	RW	0x0
3:2	AUTO_CTRL_VDD_MPU_L	This bit field specifies the state to which the hardware can automatically transition the VDD_MPU_L voltage domain. 0x0: Voltage domain transitions are disabled. 0x1: Voltage domain transitions to SLEEP are enabled. 0x2: Voltage domain transitions to RET are enabled. 0x3: reserved	RW	0x0
1:0	AUTO_CTRL_VDD_CORE_L	This bit field specifies the state to which the hardware can automatically transition the VDD_CORE_L voltage domain. 0x0: Voltage domain transitions are disabled. 0x1: Voltage domain transitions to SLEEP are enabled. 0x2: Voltage domain transitions to RET are enabled. 0x3: reserved	RW	0x0

Table 3-1228. PRM_PWRREQCTRL

Address Offset	0x0000 0014		
Physical Address	0x4AE0 7D14	**Instance**	DEVICE_PRM
Description	This register allows controlling the PWRREQ signal towards power IC.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														PWRREQ_COND	

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1:0	PWRREQ_COND	Control upon which condition from MPU, MM and CORE voltage domains PWRREQ is de-asserted. 0x0: PWRREQ is never de-asserted 0x1: PWRREQ is de-asserted if all voltage domain are in SLEEP, RET or OFF state. Conversely, PWRREQ is asserted upon any voltage domain entering or staying in ON state. 0x2: PWRREQ is de-asserted if all voltage domain are in RET or OFF state. Conversely, PWRREQ is asserted upon any voltage domain entering or staying in ON or SLEEP state. 0x3: PWRREQ is de-asserted if all voltage domain are in OFF state. Conversely, PWRREQ is asserted upon any voltage domain entering or staying in ON or SLEEP or RET state.	RW	0x0

Table 3-1229. PRM_PSCON_COUNT

Address Offset	0x0000 0018		
Physical Address	0x4AE0 7D18	**Instance**	DEVICE_PRM
Description	This register allows controlling 2 parameters for power state controller. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								HG_PONOUT_2_PGOODIN_TIME								PONOUT_2_PGOODIN_TIME								PCHARGE_TIME							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	HG_PONOUT_2_PGOODIN_TIME	The value 'NbCycles' set in this field determines the duration of the PONOUT to PGOODIN transition for power domain without DPS. The duration is computed as 8 x NbCycles of system clock cycles. Target is 10us.	RW	0x30
15:8	PONOUT_2_PGOODIN_TIME	The value 'NbCycles' set in this field determines the duration of the PONOUT to PGOODIN transition for power domain without DPS. The duration is computed as 8 x NbCycles of system clock cycles. Target is 10us.	RW	0x30
7:0	PCHARGE_TIME	Number of system clock cycles for the SRAM pre-charge duration. Target is 600ns.	RW	0x17

Table 3-1230. Register Call Summary for Register PRM_PSCON_COUNT

PRCM Register Manual

- DEVICE_PRM Register Summary: [0]

Table 3-1231. PRM_IO_COUNT

Address Offset	0x0000 001C		
Physical Address	0x4AE0 7D1C	**Instance**	DEVICE_PRM
Description	This register allows controlling DDR IO isolation removal setup. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								ISO_2_ON_TIME							

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x0
7:0	ISO_2_ON_TIME	Determines the setup time of the DDR IOs going out of isolation. Counting on the system clock. Target is 1.5us.	RW	0x3a

Table 3-1232. Register Call Summary for Register PRM_IO_COUNT

PRCM Register Manual

- DEVICE_PRM Register Summary: [0]

Table 3-1233. PRM_IO_PMCTRL

Address Offset	0x0000 0020		
Physical Address	0x4AE0 7D20	**Instance**	DEVICE_PRM
Description	This register allows controlling power management features of the IOs.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED															GLOBAL_WUEN	RESERVED						WUCLK_STATUS	WUCLK_CTRL	RESERVED		IO_ON_STATUS	ISOOVR_EXTEND	RESERVED		ISOCLK_STATUS	ISOCLK_OVERRIDE

Bits	Field Name	Description	Type	Reset
31:17	RESERVED		R	0x0
16	GLOBAL_WUEN	Global IO wakeup enable. This is a gating condition to all individual IO WUEN coming from control module. Gating is done in the Spinner logic. 0x0: All individual IO WUEN are gated in the Spinner logic (overriden to 0). 0x1: All individual IO WUEN from control module are going to IOs.	RW	0x0
15:10	RESERVED		R	0x0
9	WUCLK_STATUS	Gives value of WUCLKOUT signal coming back from IO pad ring.	R	0x0
8	WUCLK_CTRL	Direct control on WUCLKIN signal to IO pad ring. 0x0: WUCLKIN signal is driven to 0. IO wakeup daisy chain is functional as well as IO whose wakeup feature is enabled. 0x1: WUCLKIN signal is driven to 1. IO wakeup daisy chain is reset and is latching current pad states and WUEN inputs.	RW	0x0
7:6	RESERVED		R	0x0
5	IO_ON_STATUS	Gives the functional status of the IO ring. 0x0: Part or all of the IOs are not in the ON state, that is are in isolation state. 0x1: All IOs are in the ON state.	R	0x1
4	ISOOVR_EXTEND	Control non-EMIF IO isolation extension upon a device wakeup from OFF mode. 0x0: Non-EMIF IO isolation is not extended. 'EMIF_ON' IO transition happens as soon as automatic restore is completed. 0x1: Non-EMIF IO isolation is extended. 'EMIF_ON' IO transition is stalled.	RW	0x0
3:2	RESERVED		R	0x0
1	ISOCLK_STATUS	Gives value of ISOCLKOUT signal coming back from IO pad ring.	R	0x0

Bits	Field Name	Description	Type	Reset
0	ISOCLK_OVERRIDE	Override control on ISOCLKIN signal to IO pad ring. Used at boot time when it is needed to change the mode of an IO from 1.8V default mode to 1.2V mode. When not overriden, this signal is controlled by hardware only. 0x0: ISOCLKIN signal is not overriden. 0x1: ISOCLKIN signal is overriden to active value ('1').	RW	0x0

Table 3-1234. Register Call Summary for Register PRM_IO_PMCTRL

Device Low-Power States
- Isolation / Wakeup Sequence: [0][1][2][3][4][5]
- Software-Controlled I/O Isolation: [6][7]

PRCM Register Manual
- DEVICE_PRM Register Summary: [8]

Table 3-1235. PRM_VOLTSETUP_WARMRESET

Address Offset	0x0000 0024		
Physical Address	0x4AE0 7D24	**Instance**	DEVICE_PRM
Description	This register provides bit-fields for specifying voltage stabilization duration upon a global warm reset. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						STABLE_PRESCAL		RESERVED						STABLE_COUNT							

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9:8	STABLE_PRESCAL	Determines prescaler for stabilization duration counting. 0x0: Ramp-up counter is incremented every 32 system clock cycles 0x1: Ramp-up counter is incremented every 256 system clock cycles 0x2: Ramp-up counter is incremented every 2048 system clock cycles 0x3: Ramp-up counter is incremented every 16384 system clock cycles	RW	0x0
7:6	RESERVED		R	0x0
5:0	STABLE_COUNT	Determines the stabilization duration of all VDD_xxx_L regulators upon a global warm reset assertion. The duration is computed according to Stable_Prescal.	RW	0x0

Table 3-1236. PRM_VOLTSETUP_CORE_OFF

Address Offset	0x0000 0028		
Physical Address	0x4AE0 7D28	**Instance**	DEVICE_PRM
Description	This register provides bit-fields for specifying voltage ramp-up and ramp-down times for PRM managed external regulators. These values are used for VDD_CORE_L domain transitions with OFF state. [warm reset insensitive]		

Table 3-1236. PRM_VOLTSETUP_CORE_OFF (continued)

Type	RW

31 30 29 28 27 26	25 24	23 22	21 20 19 18 17 16	15 14 13 12 11 10	9 8	7 6	5 4 3 2 1 0
RESERVED	RAMP_DOWN_PRESCAL	RESERVED	RAMP_DOWN_COUNT	RESERVED	RAMP_UP_PRESCAL	RESERVED	RAMP_UP_COUNT

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:24	RAMP_DOWN_PRESCAL	Determines prescaler for ramp-down duration counting. 0x0: Ramp-down counter is incremented every 64 system clock cycles 0x1: Ramp-down counter is incremented every 256 system clock cycles 0x2: Ramp-down counter is incremented every 512 system clock cycles 0x3: Ramp-down counter is incremented every 2048 system clock cycles	RW	0x0
23:22	RESERVED		R	0x0
21:16	RAMP_DOWN_COUNT	Determines the ramp-down duration of VDD_CORE_L regulators. The duration is computed according to Ramp_Down_Prescal.	RW	0x0
15:10	RESERVED		R	0x0
9:8	RAMP_UP_PRESCAL	Determines prescaler for ramp-up duration counting. 0x0: Ramp-up counter is incremented every 64 system clock cycles 0x1: Ramp-up counter is incremented every 256 system clock cycles 0x2: Ramp-up counter is incremented every 512 system clock cycles 0x3: Ramp-up counter is incremented every 2048 system clock cycles	RW	0x0
7:6	RESERVED		R	0x0
5:0	RAMP_UP_COUNT	Determines the ramp-up duration of VDD_CORE_L regulators. The duration is computed according to Ramp_Up_Prescal. At cold reset, PRCM assumes that VDD_CORE_L will be at a valid ON voltage before SYS_NRESPWRON is de-asserted.	RW	0x0

Table 3-1237. PRM_VOLTSETUP_MPU_OFF

Address Offset	0x0000 002C		
Physical Address	0x4AE0 7D2C	Instance	DEVICE_PRM
Description	This register provides bit-fields for specifying voltage ramp-up and ramp-down times for PRM managed external regulators. These values are used for VDD_MPU_L domain transitions to or from OFF state. [warm reset insensitive]		
Type	RW		

31 30 29 28 27 26	25 24	23 22	21 20 19 18 17 16	15 14 13 12 11 10	9 8	7 6	5 4 3 2 1 0
RESERVED	RAMP_DOWN_PRESCAL	RESERVED	RAMP_DOWN_COUNT	RESERVED	RAMP_UP_PRESCAL	RESERVED	RAMP_UP_COUNT

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:24	RAMP_DOWN_PRESCAL	Determines prescaler for ramp-down duration counting. 0x0: Ramp-down counter is incremented every 64 system clock cycles 0x1: Ramp-down counter is incremented every 256 system clock cycles 0x2: Ramp-down counter is incremented every 512 system clock cycles 0x3: Ramp-down counter is incremented every 2048 system clock cycles	RW	0x0
23:22	RESERVED		R	0x0
21:16	RAMP_DOWN_COUNT	Determines the ramp-down duration of VDD_MPU_L regulators. The duration is computed according to Ramp_Down_Prescal.	RW	0x0
15:10	RESERVED		R	0x0
9:8	RAMP_UP_PRESCAL	Determines prescaler for ramp-up duration counting. 0x0: Ramp-up counter is incremented every 64 system clock cycles 0x1: Ramp-up counter is incremented every 256 system clock cycles 0x2: Ramp-up counter is incremented every 512 system clock cycles 0x3: Ramp-up counter is incremented every 2048 system clock cycles	RW	0x0
7:6	RESERVED		R	0x0
5:0	RAMP_UP_COUNT	Determines the ramp-up duration of VDD_MPU_L regulators. The duration is computed according to Ramp_Up_Prescal. At cold reset, PRCM assumes that VDD_MPU_L will be at a valid ON voltage before SYS_NRESPWRON is de-asserted.	RW	0x0

Table 3-1238. PRM_VOLTSETUP_MM_OFF

Address Offset	0x0000 0030		
Physical Address	0x4AE0 7D30	**Instance**	DEVICE_PRM
Description	This register provides bit-fields for specifying voltage ramp-up and ramp-down times for PRM managed external regulators. These values are used for VDD_MM_L domain transitions to or from OFF state. [warm reset insensitive]		
Type	RW		

31 30 29 28 27 26	25 24	23 22	21 20 19 18 17 16	15 14 13 12 11 10	9 8	7 6	5 4 3 2 1 0
RESERVED	RAMP_DOWN_PRESCAL	RESERVED	RAMP_DOWN_COUNT	RESERVED	RAMP_UP_PRESCAL	RESERVED	RAMP_UP_COUNT

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:24	RAMP_DOWN_PRESCAL	Determines prescaler for ramp-down duration counting. 0x0: Ramp-down counter is incremented every 64 system clock cycles 0x1: Ramp-down counter is incremented every 256 system clock cycles 0x2: Ramp-down counter is incremented every 512 system clock cycles 0x3: Ramp-down counter is incremented every 2048 system clock cycles	RW	0x0
23:22	RESERVED		R	0x0
21:16	RAMP_DOWN_COUNT	Determines the ramp-down duration of VDD_MM_L regulators. The duration is computed according to Ramp_Down_Prescal.	RW	0x0
15:10	RESERVED		R	0x0
9:8	RAMP_UP_PRESCAL	Determines prescaler for ramp-up duration counting. 0x0: Ramp-up counter is incremented every 64 system clock cycles 0x1: Ramp-up counter is incremented every 256 system clock cycles 0x2: Ramp-up counter is incremented every 512 system clock cycles 0x3: Ramp-up counter is incremented every 2048 system clock cycles	RW	0x0
7:6	RESERVED		R	0x0
5:0	RAMP_UP_COUNT	Determines the ramp-up duration of VDD_MM_L regulators. The duration is computed according to Ramp_Up_Prescal. At cold reset, PRCM assumes that VDD_MM_L will be at a valid ON voltage before SYS_NRESPWRON is de-asserted.	RW	0x0

Table 3-1239. PRM_VOLTSETUP_CORE_RET_SLEEP

Address Offset	0x0000 0034		
Physical Address	0x4AE0 7D34	**Instance**	DEVICE_PRM
Description	This register provides bit-fields for specifying voltage ramp-up and ramp-down times for PRM managed external regulators. These values are used for VDD_CORE_L domain transitions between ON and RET or SLEEP state. [warm reset insensitive]		
Type	RW		

31 30 29 28 27 26	25 24	23 22	21 20 19 18 17 16	15 14 13 12 11 10	9 8	7 6	5 4 3 2 1 0
RESERVED	RAMP_DOWN_PRESCAL	RESERVED	RAMP_DOWN_COUNT	RESERVED	RAMP_UP_PRESCAL	RESERVED	RAMP_UP_COUNT

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:24	RAMP_DOWN_PRESCAL	Determines prescaler for ramp-down duration counting. 0x0: Ramp-down counter is incremented every 64 system clock cycles 0x1: Ramp-down counter is incremented every 256 system clock cycles 0x2: Ramp-down counter is incremented every 512 system clock cycles 0x3: Ramp-down counter is incremented every 2048 system clock cycles	RW	0x0
23:22	RESERVED		R	0x0
21:16	RAMP_DOWN_COUNT	Determines the ramp-down duration of VDD_CORE_L regulators. The duration is computed according to Ramp_Down_Prescal.	RW	0x0
15:10	RESERVED		R	0x0
9:8	RAMP_UP_PRESCAL	Determines prescaler for ramp-up duration counting. 0x0: Ramp-up counter is incremented every 64 system clock cycles 0x1: Ramp-up counter is incremented every 256 system clock cycles 0x2: Ramp-up counter is incremented every 512 system clock cycles 0x3: Ramp-up counter is incremented every 2048 system clock cycles	RW	0x0
7:6	RESERVED		R	0x0
5:0	RAMP_UP_COUNT	Determines the ramp-up duration of VDD_CORE_L regulators. The duration is computed according to Ramp_Up_Prescal.	RW	0x0

Table 3-1240. PRM_VOLTSETUP_MPU_RET_SLEEP

Address Offset	0x0000 0038		
Physical Address	0x4AE0 7D38	**Instance**	DEVICE_PRM
Description	This register provides bit-fields for specifying voltage ramp-up and ramp-down times for PRM managed external regulators. These values are used for VDD_MPU_L domain transitions between ON and RET or SLEEP state. [warm reset insensitive]		
Type	RW		

31 30 29 28 27 26	25 24	23 22	21 20 19 18 17 16	15 14 13 12 11 10	9 8	7 6	5 4 3 2 1 0
RESERVED	RAMP_DOWN_PRESCAL	RESERVED	RAMP_DOWN_COUNT	RESERVED	RAMP_UP_PRESCAL	RESERVED	RAMP_UP_COUNT

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:24	RAMP_DOWN_PRESCAL	Determines prescaler for ramp-down duration counting. 0x0: Ramp-down counter is incremented every 64 system clock cycles 0x1: Ramp-down counter is incremented every 256 system clock cycles 0x2: Ramp-down counter is incremented every 512 system clock cycles 0x3: Ramp-down counter is incremented every 2048 system clock cycles	RW	0x0
23:22	RESERVED		R	0x0
21:16	RAMP_DOWN_COUNT	Determines the ramp-down duration of VDD_MPU_L regulators. The duration is computed according to Ramp_Down_Prescal.	RW	0x0
15:10	RESERVED		R	0x0
9:8	RAMP_UP_PRESCAL	Determines prescaler for ramp-up duration counting. 0x0: Ramp-up counter is incremented every 64 system clock cycles 0x1: Ramp-up counter is incremented every 265 system clock cycles 0x2: Ramp-up counter is incremented every 512 system clock cycles 0x3: Ramp-up counter is incremented every 2048 system clock cycles	RW	0x0
7:6	RESERVED		R	0x0
5:0	RAMP_UP_COUNT	Determines the ramp-up duration of VDD_MPU_L regulators. The duration is computed according to Ramp_Up_Prescal.	RW	0x0

Table 3-1241. PRM_VOLTSETUP_MM_RET_SLEEP

Address Offset	0x0000 003C		
Physical Address	0x4AE0 7D3C	**Instance**	DEVICE_PRM
Description	This register provides bit-fields for specifying voltage ramp-up and ramp-down times for PRM managed external regulators. These values are used for VDD_MM_L domain transitions between ON and RET or SLEEP state. [warm reset insensitive]		
Type	RW		

31 30 29 28 27 26	25 24	23 22	21 20 19 18 17 16	15 14 13 12 11 10	9 8	7 6	5 4 3 2 1 0
RESERVED	RAMP_DOWN_PRESCAL	RESERVED	RAMP_DOWN_COUNT	RESERVED	RAMP_UP_PRESCAL	RESERVED	RAMP_UP_COUNT

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:24	RAMP_DOWN_PRESCAL	Determines prescaler for ramp-down duration counting. 0x0: Ramp-down counter is incremented every 64 system clock cycles 0x1: Ramp-down counter is incremented every 256 system clock cycles 0x2: Ramp-down counter is incremented every 512 system clock cycles 0x3: Ramp-down counter is incremented every 2048 system clock cycles	RW	0x0
23:22	RESERVED		R	0x0
21:16	RAMP_DOWN_COUNT	Determines the ramp-down duration of VDD_MM_L regulators. The duration is computed according to Ramp_Down_Prescal.	RW	0x0
15:10	RESERVED		R	0x0
9:8	RAMP_UP_PRESCAL	Determines prescaler for ramp-up duration counting. 0x0: Ramp-up counter is incremented every 64 system clock cycles 0x1: Ramp-up counter is incremented every 256 system clock cycles 0x2: Ramp-up counter is incremented every 512 system clock cycles 0x3: Ramp-up counter is incremented every 2048 system clock cycles	RW	0x0
7:6	RESERVED		R	0x0
5:0	RAMP_UP_COUNT	Determines the ramp-up duration of VDD_MM_L regulators. The duration is computed according to Ramp_Up_Prescal.	RW	0x0

Table 3-1242. PRM_VP_CORE_CONFIG

Address Offset	0x0000 0040		
Physical Address	0x4AE0 7D40	**Instance**	DEVICE_PRM
Description	This register allows the configuration of the Voltage Processor dedicated to CORE Voltage Domain (VDD_CORE_L).		
Type	RW		

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4	3	2	1	0
ERROROFFSET	ERRORGAIN	INITVOLTAGE	RESERVED	TIMEOUTEN	INITVDD	FORCEUPDATE	ISSNABLE

Bits	Field Name	Description	Type	Reset
31:24	ERROROFFSET	Offset value in the Error to Voltage converter (two's complement number).	RW	0x0
23:16	ERRORGAIN	Gain value in the Error to Voltage converter (two's complement number).	RW	0x0
15:8	INITVOLTAGE	Set the initial voltage level of the SMPS.	RW	0x0
7:4	RESERVED	Write 0's for future compatibility. Read is undefined.	R	0x0
3	TIMEOUTEN	Enable or disable the timeout capability of the Voltage Controller State Machine. 0x0: Timeout is disabled. Loop will wait indefinitely. 0x1: Timeout will occur when TIMEOUT cycles have elapsed.	RW	0x0
2	INITVDD	Initializes the voltage in the Voltage Processor. 0x0: Reset the initialization bit. 0x1: The positive edge of InitVdd triggers a write of the value in the InitVoltage into the Voltage Processor.	RW	0x0
1	FORCEUPDATE	Forces an update of the SMPS. 0x0: Reset the force bit. 0x1: The positive edge of ForceUpdate triggers an update of the voltage to the SMPS.	RW	0x0
0	ISSNABLE	Enables or disables the Voltage Processor updates on SR_SInterruptz. 0x0: Disables the Voltage Processor. 0x1: Enables the Voltage Processor.	RW	0x0

Table 3-1243. PRM_VP_CORE_STATUS

Address Offset	0x0000 0044		
Physical Address	0x4AE0 7D44	**Instance**	DEVICE_PRM
Description	This register reflects the idle state of the Voltage Processor dedicated to the CORE Voltage Domain (VDD_CORE_L. This register is read only and automatically updated.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															VPINIDLE

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Write 0's for future compatibility. Read is undefined.	R	0x0
0	VPINIDLE	CORE Voltage Processor idle status. 0x0: The Voltage Processor for CORE is processing. Warm reset sensitive 0x1: The Voltage Processor for CORE is in idle state.	R	0x1

Table 3-1244. PRM_VP_CORE_VLIMITTO

Address Offset	0x0000 0048		
Physical Address	0x4AE0 7D48	**Instance**	DEVICE_PRM
Description	This register allows the configuration of the voltage limits and timeout values of the Voltage Processor dedicated to the CORE Voltage Domain (VDD_CORE_L).		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VDDMAX								VDDMIN								TIMEOUT															

Bits	Field Name	Description	Type	Reset
31:24	VDDMAX	Defines the maximum voltage supply level.	RW	0x0
23:16	VDDMIN	Defines the minimum voltage supply level.	RW	0x0
15:0	TIMEOUT	Defines Voltage Controller maximum wait time for responses.	RW	0x0

Table 3-1245. PRM_VP_CORE_VOLTAGE

Address Offset	0x0000 004C		
Physical Address	0x4AE0 7D4C	**Instance**	DEVICE_PRM
Description	This register indicates the current value of the SMPS voltage for the Voltage Processor dedicated to the CORE Voltage Domain (VDD_CORE_L).		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FORCEUPDATEWAIT																								VPVOLTAGE							

Bits	Field Name	Description	Type	Reset
31:8	FORCEUPDATEWAIT	The time voltage processor needs to wait for SMPS to be settled after receiving SMPS acknowledge. This wait can only be used during force_update operation.	RW	0x111
7:0	VPVOLTAGE	Indicates the current SMPS programmed voltage.	R	0x0

Table 3-1246. PRM_VP_CORE_VSTEPMAX

Address Offset	0x0000 0050		
Physical Address	0x4AE0 7D50	**Instance**	DEVICE_PRM
Description	This register allows the programming of the maximum voltage step and waiting time of the Voltage Processor dedicated to CORE Voltage Domain (VDD_CORE_L).		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								SMPSWAITTIMEMAX																VSTEPMAX							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED	Write 0's for future compatibility. Read is undefined.	R	0x0
23:8	SMPSWAITTIMEMAX	Slew rate for positive voltage step (in number of cycles per step).	RW	0x0
7:0	VSTEPMAX	Maximum voltage step	RW	0x0

Table 3-1247. PRM_VP_CORE_VSTEPMIN

Address Offset	0x0000 0054		
Physical Address	0x4AE0 7D54	**Instance**	DEVICE_PRM
Description	This register allows the programming of the minimum voltage step and waiting time of the Voltage Processor dedicated to the CORE Voltage Domain (VDD_CORE_L).		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								SMPSWAITTIMEMIN																VSTEPMIN							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED	Write 0's for future compatibility. Read is undefined.	R	0x0
23:8	SMPSWAITTIMEMIN	Slew rate for negative voltage step (in number of cycles per step).	RW	0x0
7:0	VSTEPMIN	Minimum voltage step	RW	0x0

Table 3-1248. PRM_VP_MPU_CONFIG

Address Offset	0x0000 0058		
Physical Address	0x4AE0 7D58	**Instance**	DEVICE_PRM
Description	This register allows the configuration of the Voltage Processor dedicated to MPU Voltage Domain (VDD_MPU_L).		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ERROROFFSET								ERRORGAIN								INITVOLTAGE								RESERVED				TIMEOUTEN	INITVDD	FORCEUPDATE	ISSNABLE

Bits	Field Name	Description	Type	Reset
31:24	ERROROFFSET	Offset value in the Error to Voltage converter (two's complement number).	RW	0x0
23:16	ERRORGAIN	Gain value in the Error to Voltage converter (two's complement number).	RW	0x0
15:8	INITVOLTAGE	Set the initial voltage level of the SMPS.	RW	0x0
7:4	RESERVED	Write 0's for future compatibility. Read is undefined.	R	0x0
3	TIMEOUTEN	Enable or disable the timeout capability of the Voltage Controller State Machine. 0x0: Timeout is disabled. Loop will wait indefinitely. 0x1: Timeout will occur when TIMEOUT cycles have elapsed.	RW	0x0
2	INITVDD	Initializes the voltage in the Voltage Processor. 0x0: Reset the initialization bit. 0x1: The positive edge of InitVdd triggers a write of the value in the InitVoltage into the Voltage Processor.	RW	0x0
1	FORCEUPDATE	Forces an update of the SMPS. 0x0: Reset the force bit. 0x1: The positive edge of ForceUpdate triggers an update of the voltage to the SMPS.	RW	0x0
0	ISSNABLE	Enables or disables the Voltage Processor updates on SR_SInterruptz. 0x0: Disables the Voltage Processor. 0x1: Enables the Voltage Processor.	RW	0x0

Table 3-1249. PRM_VP_MPU_STATUS

Address Offset	0x0000 005C		
Physical Address	0x4AE0 7D5C	**Instance**	DEVICE_PRM
Description	This register reflects the idle state of the Voltage Processor dedicated to the MPU Voltage Domain (VDD_MPU_L. This register is read only and automatically updated.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															VPINIDLE

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Write 0's for future compatibility. Read is undefined.	R	0x0
0	VPINIDLE	Voltage Processor 1 idle status. 0x0: The Voltage Processor 1 is processing. 0x1: The Voltage Processor 1 is in idle state.	R	0x1

Table 3-1250. PRM_VP_MPU_VLIMITTO

Address Offset	0x0000 0060		
Physical Address	0x4AE0 7D60	**Instance**	DEVICE_PRM
Description	This register allows the configuration of the voltage limits and timeout values of the Voltage Processor dedicated to the MPU Voltage Domain (VDD_MPU_L).		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VDDMAX								VDDMIN								TIMEOUT															

Bits	Field Name	Description	Type	Reset
31:24	VDDMAX	Defines the maximum voltage supply level.	RW	0x0
23:16	VDDMIN	Defines the minimum voltage supply level.	RW	0x0
15:0	TIMEOUT	Defines Voltage Controller maximum wait time for responses.	RW	0x0

Table 3-1251. PRM_VP_MPU_VOLTAGE

Address Offset	0x0000 0064		
Physical Address	0x4AE0 7D64	**Instance**	DEVICE_PRM
Description	This register indicates the current value of the SMPS voltage for the Voltage Processor dedicated to the MPU Voltage Domain (VDD_MPU_L).		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FORCEUPDATEWAIT																								VPVOLTAGE							

Bits	Field Name	Description	Type	Reset
31:8	FORCEUPDATEWAIT	The time voltage processor needs to wait for SMPS to be settled after receiving SMPS acknowledge. This wait only be used during force_update operation.	RW	0x111
7:0	VPVOLTAGE	Indicates the current SMPS programmed voltage.	R	0x0

Table 3-1252. PRM_VP_MPU_VSTEPMAX

Address Offset	0x0000 0068		
Physical Address	0x4AE0 7D68	**Instance**	DEVICE_PRM
Description	This register allows the programming of the maximum voltage step and waiting time of the Voltage Processor dedicated to MPU Voltage Domain (VDD_MPU_L).		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								SMPSWAITTIMEMAX																VSTEPMAX							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED	Write 0's for future compatibility. Read is undefined.	R	0x0
23:8	SMPSWAITTIMEMAX	Slew rate for positive voltage step (in number of cycles per step).	RW	0x0
7:0	VSTEPMAX	Maximum voltage step	RW	0x0

Table 3-1253. PRM_VP_MPU_VSTEPMIN

Address Offset	0x0000 006C		
Physical Address	0x4AE0 7D6C	**Instance**	DEVICE_PRM
Description	This register allows the programming of the minimum voltage step and waiting time of the Voltage Processor dedicated to the MPU Voltage Domain (VDD_MPU_L).		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								SMPSWAITTIMEMIN																VSTEPMIN							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED	Write 0's for future compatibility. Read is undefined.	R	0x0
23:8	SMPSWAITTIMEMIN	Slew rate for negative voltage step (in number of cycles per step).	RW	0x0
7:0	VSTEPMIN	Minimum voltage step	RW	0x0

Table 3-1254. PRM_VP_MM_CONFIG

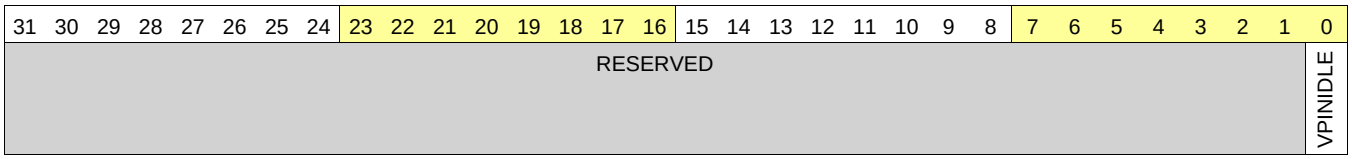
Address Offset	0x0000 0070		
Physical Address	0x4AE0 7D70	**Instance**	DEVICE_PRM
Description	This register allows the configuration of the Voltage Processor dedicated to MM Voltage Domain (VDD_MM_L).		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ERROROFFSET								ERRORGAIN								INITVOLTAGE								RESERVED				TIMEOUTEN	INITVDD	FORCEUPDATE	ISSNABLE

Bits	Field Name	Description	Type	Reset
31:24	ERROROFFSET	Offset value in the Error to Voltage converter (two's complement number).	RW	0x0
23:16	ERRORGAIN	Gain value in the Error to Voltage converter (two's complement number).	RW	0x0
15:8	INITVOLTAGE	Set the initial voltage level of the SMPS.	RW	0x0
7:4	RESERVED	Write 0's for future compatibility. Read is undefined.	R	0x0
3	TIMEOUTEN	Enable or disable the timeout capability of the Voltage Controller State Machine. 0x0: Timeout is disabled. Loop will wait indefinitely. 0x1: Timeout will occur when TIMEOUT cycles have elapsed.	RW	0x0
2	INITVDD	Initializes the voltage in the Voltage Processor. 0x0: Reset the initialization bit. 0x1: The positive edge of InitVdd triggers a write of the value in the InitVoltage into the Voltage Processor.	RW	0x0
1	FORCEUPDATE	Forces an update of the SMPS. 0x0: Reset the force bit. 0x1: The positive edge of ForceUpdate triggers an update of the voltage to the SMPS.	RW	0x0
0	ISSNABLE	Enables or disables the Voltage Processor updates on SR_SInterruptz. 0x0: Disables the Voltage Processor. 0x1: Enables the Voltage Processor.	RW	0x0

Table 3-1255. PRM_VP_MM_STATUS

Address Offset	0x0000 0074		
Physical Address	0x4AE0 7D74	**Instance**	DEVICE_PRM
Description	This register reflects the idle state of the Voltage Processor dedicated to the MPU Voltage Domain (VDD_MM_L. This register is read only and automatically updated.		
Type	R		



Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Write 0's for future compatibility. Read is undefined.	R	0x0
0	VPINIDLE	Voltage Processor 1 idle status. 0x0: The Voltage Processor 1 is processing. 0x1: The Voltage Processor 1 is in idle state.	R	0x1

Table 3-1256. PRM_VP_MM_VLIMITTO

Address Offset	0x0000 0078		
Physical Address	0x4AE0 7D78	**Instance**	DEVICE_PRM
Description	This register allows the configuration of the voltage limits and timeout values of the Voltage Processor dedicated to the MM voltage Domain (VDD_MM_L).		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VDDMAX								VDDMIN								TIMEOUT															

Bits	Field Name	Description	Type	Reset
31:24	VDDMAX	Defines the maximum voltage supply level.	RW	0x0
23:16	VDDMIN	Defines the minimum voltage supply level.	RW	0x0
15:0	TIMEOUT	Defines Voltage Controller maximum wait time for responses.	RW	0x0

Table 3-1257. PRM_VP_MM_VOLTAGE

Address Offset	0x0000 007C		
Physical Address	0x4AE0 7D7C	**Instance**	DEVICE_PRM
Description	This register indicates the current value of the SMPS voltage for the Voltage Processor dedicated to the MM voltage Domain (VDD_MM_L).		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FORCEUPDATEWAIT																								VPVOLTAGE							

Bits	Field Name	Description	Type	Reset
31:8	FORCEUPDATEWAIT	The time voltage processor needs to wait for SMPS to be settled after receiving SMPS acknowledge. This wait only be used during force_update operation.	RW	0x111
7:0	VPVOLTAGE	Indicates the current SMPS programmed voltage.	R	0x0

Table 3-1258. PRM_VP_MM_VSTEPMAX

Address Offset	0x0000 0080		
Physical Address	0x4AE0 7D80	**Instance**	DEVICE_PRM
Description	This register allows the programming of the maximum voltage step and waiting time of the Voltage Processor dedicated to MM voltage Domain (VDD_MM_L).		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								SMPSWAITTIMEMAX																VSTEPMAX							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED	Write 0's for future compatibility. Read is undefined.	R	0x0
23:8	SMPSWAITTIMEMAX	Slew rate for positive voltage step (in number of cycles per step).	RW	0x0
7:0	VSTEPMAX	Maximum voltage step	RW	0x0

Table 3-1259. PRM_VP_MM_VSTEPMIN

Address Offset	0x0000 0084		
Physical Address	0x4AE0 7D84	**Instance**	DEVICE_PRM
Description	This register allows the programming of the minimum voltage step and waiting time of the Voltage Processor dedicated to the MM voltage Domain (VDD_MM_L).		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								SMPSWAITTIMEMIN																VSTEPMIN							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED	Write 0's for future compatibility. Read is undefined.	R	0x0
23:8	SMPSWAITTIMEMIN	Slew rate for negative voltage step (in number of cycles per step).	RW	0x0
7:0	VSTEPMIN	Minimum voltage step	RW	0x0

Table 3-1260. PRM_VC_SMPS_CORE_CONFIG

Address Offset	0x0000 0088		
Physical Address	0x4AE0 7D88	**Instance**	DEVICE_PRM
Description	This register allows the setting of the I2C slave address of the Power IC device, the setting of the voltage configuration register address for the CORE VDD and the Command (ON/ON-Low-Power/Retention/OFF) configuration register address values for CORE VDD (if used SMPS chips have different command configuration register than voltage configuration register). [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED			CMD_VDD_CORE_L	RACEN_VDD_CORE_L	RAC_VDD_CORE_L	RAV_VDD_CORE_L	SEL_SA_VDD_CORE_L	CMDRA_VDD_CORE_L								VOLRA_VDD_CORE_L								RESERVED	SA_VDD_CORE_L						

Bits	Field Name	Description	Type	Reset
31:29	RESERVED	Write 0's for future compatibility. Read is undefined.	R	0x0
28	CMD_VDD_CORE_L	Command values (ON/ON-Low-Power/Retention/OFF voltage values) set selection for VDD_CORE_L channel 0x0: VDD_CORE_L channel use VC_VAL_CMD_VDD_MPU_L set for command values 0x1: VDD_CORE_L channel use VC_VAL_CMD_VDD_CORE_L set for command values	RW	0x1
27	RACEN_VDD_CORE_L	Enable bit for usage of RAC_VDD_CORE_L 0x0: VDD_CORE_L channel uses VOLRA values for register address of VFSM-s commands. VFSM-s commands goes also to voltage configuration register. 0x1: VDD_CORE_L channel uses CMDRA values for register address of VFSM-s commands. VFSM-s commands goes to different command configuration register.	RW	0x0
26	RAC_VDD_CORE_L	Command (ON/ON-Low-Power/Retention/OFF) configuration register address pointer for VDD_CORE_L channel 0x0: Select CMDRA_VDD_MPU_L for VDD_CORE_L channel 0x1: Select CMDRA_VDD_CORE_L for VDD_CORE_L channel	RW	0x1

Bits	Field Name	Description	Type	Reset
25	RAV_VDD_CORE_L	Voltage configuration register address pointer for VDD_CORE_L channel. 0x0: Select VOLRA_VDD_MPU_L for VDD_CORE_L channel 0x1: Select VOLRA_VDD_CORE_L for VDD_CORE_L channel	RW	0x1
24	SEL_SA_VDD_CORE_L	Slave address pointer for VDD_CORE_L channel. 0x0: Select SA_VDD_MPU_L for VDD_CORE_L channel 0x1: Select SA_VDD_CORE_L for VDD_CORE_L channel	RW	0x0
23:16	CMDRA_VDD_CORE_L	Command (ON/ON-Low-Power /Retention/OFF) configuration register address value for VDD_CORE_L channel.(if VDD_CORE_L source has different command configuration register than voltage VDD_MPU_L)	RW	0x0
15:8	VOLRA_VDD_CORE_L	Set the voltage configuration register address value for the VDD_CORE_L channel (if VDD_CORE_L source is placed in same chip as VDD_MPU_L source and have different voltage configuration register)	RW	0x0
7	RESERVED	Write 0's for future compatibility. Read is undefined.	R	0x0
6:0	SA_VDD_CORE_L	Set the I2C slave address value for the first Power IC device.	RW	0x0

Table 3-1261. PRM_VC_SMPS_MM_CONFIG

Address Offset	0x0000 008C		
Physical Address	0x4AE0 7D8C	**Instance**	DEVICE_PRM
Description	This register allows the setting of the I2C slave address of the Power IC device, the setting of the voltage configuration register address for the MM VDD and the Command (ON/ON-Low-Power/Retention/OFF) configuration register address values for MM VDD (if used SMPS chips have different command configuration register than voltage configuration register).. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED			CMD_VDD_MM_L	RACEN_VDD_MM_L	RAC_VDD_MM_L	RAV_VDD_MM_L	SEL_SA_VDD_MM_L	CMDRA_VDD_MM_L								VOLRA_VDD_MM_L								RESERVED	SA_VDD_MM_L						

Bits	Field Name	Description	Type	Reset
31:29	RESERVED	Write 0's for future compatibility. Read is undefined.	R	0x0
28	CMD_VDD_MM_L	Command values (ON/ON-Low-Power/Retention/OFF voltage values) set selection for VDD_MM_L channel 0x0: VDD_MM_L channel use VC_VAL_CMD_VDD_MPU_L set for command values 0x1: VDD_MM_L channel use VC_VAL_CMD_VDD_MM_L set for command values	RW	0x1

Bits	Field Name	Description	Type	Reset
27	RACEN_VDD_MM_L	Enable bit for usage of RAC_VDD_MM_L 0x0: VDD_MM_L channel uses VOLRA values for register address of VFSM-s commands. VFSM-s commands goes also to voltage configuration register. 0x1: VDD_MM_L channel uses CMDRA values for register address of VFSM-s commands. VFSM-s commands goes to different command configuration register.	RW	0x0
26	RAC_VDD_MM_L	Command (ON/ON-Low-Power/Retention/OFF) configuration register address pointer for VDD_MM_L channel 0x0: Select CMDRA_VDD_MPU_L for VDD_MM_L channel 0x1: Select CMDRA_VDD_MM_L for VDD_MM_L channel	RW	0x1
25	RAV_VDD_MM_L	Voltage configuration register address pointer for VDD_MM_L channel. 0x0: Select VOLRA_VDD_MPU_L for VDD_MM_L channel 0x1: Select VOLRA_VDD_MM_L for VDD_MM_L channel	RW	0x1
24	SEL_SA_VDD_MM_L	Slave address pointer for VDD_MM_L channel. 0x0: Select SA_VDD_MPU_L for VDD_MM_L channel 0x1: Select SA_VDD_MM_L for VDD_MM_L channel	RW	0x0
23:16	CMDRA_VDD_MM_L	Command (ON/ON-Low-Power /Retention/OFF) configuration register address value for VDD_MM_L channel (if VDD_MM_L source has different command configuration register than voltage VDD_MPU_L)	RW	0x0
15:8	VOLRA_VDD_MM_L	Voltage configuration register address value for VDD_MM_L channel (if VDD_MM_L source is placed in same chip as VDD_MPU_L source and have different voltage configuration register)	RW	0x0
7	RESERVED	Write 0's for future compatibility. Read is undefined.	R	0x0
6:0	SA_VDD_MM_L	Set the I2C slave address value for the second (if any) Power IC device.	RW	0x0

Table 3-1262. PRM_VC_SMPS_MPU_CONFIG

Address Offset	0x0000 0090		
Physical Address	0x4AE0 7D90	**Instance**	DEVICE_PRM
Description	This register allows the setting of the I2C slave address of the Power IC device, the setting of the voltage configuration register address for the MPU VDD and the Command (ON/ON-Low-Power/Retention/OFF) configuration register address values for MPU VDD (if used SMPS chips have different command configuration register than voltage configuration register). [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED			CMD_VDD_MPU_L	RACEN_VDD_MPU_L	RAC_VDD_MPU_L	RAV_VDD_MPU_L	SEL_SA_VDD_MPU_L	CMDRA_VDD_MPU_L								VOLRA_VDD_MPU_L								RESERVED	SA_VDD_MPU_L						

Bits	Field Name	Description	Type	Reset
31:29	RESERVED	Write 0's for future compatibility. Read is undefined.	R	0x0
28	CMD_VDD_MPU_L	Command values (ON/ON-Low-Power/Retention/OFF voltage values) set selection for VDD_MPU_L channel (This bit has no influence on VDD_MPU_L channel)	RW	0x0
27	RACEN_VDD_MPU_L	Enable bit for usage of RAC_VDD_MPU_L 0x0: VDD_MPU_L channel uses VOLRA values for register address of VFSM-s commands. VFSM-s commands goes also to voltage configuration register. 0x1: VDD_MPU_L channel uses CMDRA values for register address of VFSM-s commands. VFSM-s commands goes to different command configuration register.	RW	0x0
26	RAC_VDD_MPU_L	Command (ON/ON-Low-Power/Retention/OFF) configuration register address pointer for VDD_MPU_L channel. (This bit has no influence on first VDD_MPU_L channel)	RW	0x0
25	RAV_VDD_MPU_L	Voltage configuration register address pointer for VDD_MPU_L channel. (This bit has no influence on first VDD_MPU_L channel)	RW	0x0
24	SEL_SA_VDD_MPU_L	Slave address pointer for VDD_MPU_L channel. (This bit has no influence on first VDD_MPU_L channel)	RW	0x0
23:16	CMDRA_VDD_MPU_L	Command (ON/ON-Low-Power /Retention/OFF) configuration register address value for VDD_MPU_L channel.	RW	0x0
15:8	VOLRA_VDD_MPU_L	Voltage configuration register address value for VDD_MPU_L channel.	RW	0x0
7	RESERVED	Write 0's for future compatibility. Read is undefined.	R	0x0
6:0	SA_VDD_MPU_L	Set the I2C slave address value for the third (if any) Power IC device.	RW	0x0

Table 3-1263. PRM_VC_VAL_CMD_VDD_CORE_L

Address Offset	0x0000 0094		
Physical Address	0x4AE0 7D94	**Instance**	DEVICE_PRM
Description	This register allows the setting of the ON/ON-Low-Power/Retention/OFF command values for VDD_CORE_L channel. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ON								ONLP								RET								OFF							

Bits	Field Name	Description	Type	Reset
31:24	ON	Set the ON command value.	RW	0x0
23:16	ONLP	Set the ON-Low-Power command value.	RW	0x0
15:8	RET	Set the RET command value.	RW	0x0
7:0	OFF	Set the OFF command value.	RW	0x0

Table 3-1264. PRM_VC_VAL_CMD_VDD_MM_L

Address Offset	0x0000 0098		
Physical Address	0x4AE0 7D98	**Instance**	DEVICE_PRM
Description	This register allows the setting of the ON/ON-Low-Power/Retention/OFF command values for VDD_MM_L channel. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ON								ONLP								RET								OFF							

Bits	Field Name	Description	Type	Reset
31:24	ON	Set the ON command value.	RW	0x0
23:16	ONLP	Set the ON-Low-Power command value.	RW	0x0
15:8	RET	Set the RET command value.	RW	0x0
7:0	OFF	Set the OFF command value.	RW	0x0

Table 3-1265. PRM_VC_VAL_CMD_VDD_MPU_L

Address Offset	0x0000 009C		
Physical Address	0x4AE0 7D9C	**Instance**	DEVICE_PRM
Description	This register allows the setting of the ON/ON-Low-Power/Retention/OFF command values for VDD_MPU_L channel. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ON								ONLP								RET								OFF							

Bits	Field Name	Description	Type	Reset
31:24	ON	Set the ON command value.	RW	0x0
23:16	ONLP	Set the ON-Low-Power command value.	RW	0x0
15:8	RET	Set the RET command value.	RW	0x0
7:0	OFF	Set the OFF command value.	RW	0x0

Table 3-1266. PRM_VC_VAL_BYPASS

Address Offset	0x0000 00A0		
Physical Address	0x4AE0 7DA0	**Instance**	DEVICE_PRM
Description	Bypass data values register used for bypass command channel to send other configuration information (other then voltage configuration parameters) for SMPS chips which have no other configuration interface then this I2C interface and flag to indicate OPP change to EMIF to allow read/write leveling. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED						OPP_CHANGE_EMIF_LVL	VALID	DATA								REGADDR								RESERVED	SLAVEADDR						

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	OPP_CHANGE_EMIF_LVL	This bit controls read-write leveling of EMIF memories (DDR3). It must be set in case OPP voltage change is done through Voltage Controller whithout passing through Voltage processor. 0x0: Enable leveling 0x1: disable leveling	RW	0x0
24	VALID	This bit validates the bypass command. It is automatically cleared by HW either after getting the acknowledge back from the SMPS or if an error occurred. 0x0: The last command send has been acknowledged 0x1: Pending command is being process	RW	0x0
23:16	DATA	Data to send to the Power IC device.	RW	0x0
15:8	REGADDR	Set the address of Power IC device register to configure.	RW	0x0
7	RESERVED		R	0x0
6:0	SLAVEADDR	Set the I2C slave address value.	RW	0x0

Table 3-1267. PRM_VC_CORE_ERRST

Address Offset	0x0000 00A4		
Physical Address	0x4AE0 7DA4	**Instance**	DEVICE_PRM
Description	This debug register logs CORE related error status coming from Voltage Controller. Must be cleared by software.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																										VFSM_TIMEOUT_ERR_CORE	VFSM_RA_ERR_CORE	VFSM_SA_ERR_CORE	SMPS_TIMEOUT_ERR_CORE	SMPS_RA_ERR_CORE	SMPS_SA_ERR_CORE

Bits	Field Name	Description	Type	Reset
31:6	RESERVED		R	0x0
5	VFSM_TIMEOUT_ERR_CORE	CORE voltage FSM command frame is finished but is not acknowledged by the slave, or (I2C multimaster) arbitration lost. 0x0: No error 0x1: An error has been logged	RW	0x0
4	VFSM_RA_ERR_CORE	Wrong register address error for CORE voltage FSM 0x0: No error 0x1: An error has been logged	RW	0x0
3	VFSM_SA_ERR_CORE	Wrong slave address error for CORE voltage FSM 0x0: No error 0x1: An error has been logged	RW	0x0

Bits	Field Name	Description	Type	Reset
2	SMPS_TIMEOUT_ERR_CORE	CORE voltage processor command frame is finished but is not acknowledged by the slave, or (I2C multimaster) arbitration lost. 0x0: No error 0x1: An error has been logged	RW	0x0
1	SMPS_RA_ERR_CORE	Wrong register address error for CORE voltage processor 0x0: No error 0x1: An error has been logged	RW	0x0
0	SMPS_SA_ERR_CORE	Wrong slave address error for CORE voltage processor 0x0: No error 0x1: An error has been logged	RW	0x0

Table 3-1268. PRM_VC_MM_ERRST

Address Offset	0x0000 00A8		
Physical Address	0x4AE0 7DA8	**Instance**	DEVICE_PRM
Description	This debug register logs MM related error status coming from Voltage Controller. Must be cleared by software.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																										VFSM_TIMEOUT_ERR_MM	VFSM_RA_ERR_MM	VFSM_SA_ERR_MM	SMPS_TIMEOUT_ERR_MM	SMPS_RA_ERR_MM	SMPS_SA_ERR_MM

Bits	Field Name	Description	Type	Reset
31:6	RESERVED		R	0x0
5	VFSM_TIMEOUT_ERR_MM	MM voltage FSM command frame is finished but is not acknowledged by the slave, or (I2C multimaster) arbitration lost. 0x0: No error 0x1: An error has been logged	RW	0x0
4	VFSM_RA_ERR_MM	Wrong register address error for MM voltage FSM 0x0: No error 0x1: An error has been logged	RW	0x0
3	VFSM_SA_ERR_MM	Wrong slave address error for MM voltage FSM 0x0: No error 0x1: An error has been logged	RW	0x0
2	SMPS_TIMEOUT_ERR_MM	MM voltage processor command frame is finished but is not acknowledged by the slave, or (I2C multimaster) arbitration lost. 0x0: No error 0x1: An error has been logged	RW	0x0
1	SMPS_RA_ERR_MM	Wrong register address error for MM voltage processor 0x0: No error 0x1: An error has been logged	RW	0x0

Bits	Field Name	Description	Type	Reset
0	SMPS_SA_ERR_MM	Wrong slave address error for MM voltage processor 0x0: No error 0x1: An error has been logged	RW	0x0

Table 3-1269. PRM_VC_MPU_ERRST

Address Offset	0x0000 00AC		
Physical Address	0x4AE0 7DAC	**Instance**	DEVICE_PRM
Description	This debug register logs MPU related error status coming from Voltage Controller. Must be cleared by software.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																										VFSM_TIMEOUT_ERR_MPU	VFSM_RA_ERR_MPU	VFSM_SA_ERR_MPU	SMPS_TIMEOUT_ERR_MPU	SMPS_RA_ERR_MPU	SMPS_SA_ERR_MPU

Bits	Field Name	Description	Type	Reset
31:6	RESERVED		R	0x0
5	VFSM_TIMEOUT_ERR_MPU	MPU voltage FSM command frame is finished but is not acknowledged by the slave, or (I2C multimaster) arbitration lost. 0x0: No error 0x1: An error has been logged	RW	0x0
4	VFSM_RA_ERR_MPU	Wrong register address error for MPU voltage FSM 0x0: No error 0x1: An error has been logged	RW	0x0
3	VFSM_SA_ERR_MPU	Wrong slave address error for MPU voltage FSM 0x0: No error 0x1: An error has been logged	RW	0x0
2	SMPS_TIMEOUT_ERR_MPU	MPU voltage processor command frame is finished but is not acknowledged by the slave, or (I2C multimaster) arbitration lost. 0x0: No error 0x1: An error has been logged	RW	0x0
1	SMPS_RA_ERR_MPU	Wrong register address error for MPU voltage processor 0x0: No error 0x1: An error has been logged	RW	0x0
0	SMPS_SA_ERR_MPU	Wrong slave address error for MPU voltage processor 0x0: No error 0x1: An error has been logged	RW	0x0

Table 3-1270. PRM_VC_BYPASS_ERRST

Address Offset	0x0000 00B0		
Physical Address	0x4AE0 7DB0	**Instance**	DEVICE_PRM
Description	This debug register logs BYPASS related error status coming from Voltage Controller. Must be cleared by software.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																													BYPS_TIMEOUT_ERR	BYPS_RA_ERR	BYPS_SA_ERR

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0
2	BYPS_TIMEOUT_ERR	BYPASS command frame is finished but is not acknowledged by the slave, or (I2C multimaster) arbitration lost. 0x0: No error 0x1: An error has been logged	RW	0x0
1	BYPS_RA_ERR	Wrong register address error for BYPASS command 0x0: No error 0x1: An error has been logged	RW	0x0
0	BYPS_SA_ERR	Wrong slave address error for BYPASS command 0x0: No error 0x1: An error has been logged	RW	0x0

Table 3-1271. PRM_VC_CFG_I2C_MODE

Address Offset	0x0000 00B4		
Physical Address	0x4AE0 7DB4	**Instance**	DEVICE_PRM
Description	I2C configuration register. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								DFILTEREN	RESERVED	SRMODEEN	HSMODEEN	HSMCODE			

Bits	Field Name	Description	Type	Reset
31:7	RESERVED		R	0x0
6	DFILTEREN	This field enables double filter procedure for I2C input lines 0x0: I2C bus digital filter rejects all glitches smaller than 1 sytem clock cycle 0x1: I2C bus digital filter rejects all glitches smaller than 2 sytem clock cycle	RW	0x0

Bits	Field Name	Description	Type	Reset
5	RESERVED		R	0x0
4	SRMODEEN	Enables the I2C repeated start operation mode (effect of holding the SCL and SDA lines low, in effect blocking the I2C bus from losing arbitration between repeated start points). Use of this feature results from a trade-off between speed and power consumption of I2C interface. 0x0: Disables the repeated start operation mode 0x1: Enables the repeated start operation mode	RW	0x1
3	HSMODEEN	Enables I2C bus High Speed mode. 0x0: Disables the I2C high speed mode 0x1: Enables the I2C high speed mode	RW	0x1
2:0	HSMCODE	Master code value for I2C High Speed preamble transmission.	RW	0x0

Table 3-1272. PRM_VC_CFG_I2C_CLK

Address Offset	0x0000 00B8		
Physical Address	0x4AE0 7DB8	**Instance**	DEVICE_PRM
Description	I2C Interface clock configuration parameters. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HSSCLL								HSSCLH								SCLL								SCLH							

Bits	Field Name	Description	Type	Reset
31:24	HSSCLL	Number of the system clock cycles, necessary to count the low period of the I2C clock signal, when the I2C interface runs in High-Speed mode of operation.	RW	0x0
23:16	HSSCLH	Number of the system clock cycles, necessary to count the high period of the I2C clock signal, when the I2C interface runs in High-Speed mode of operation.	RW	0x0
15:8	SCLL	Number of the system clock cycles, necessary to count the low period of the I2C clock signal, when the I2C interface runs in Fast mode of operation.	RW	0x0
7:0	SCLH	Number of the system clock cycles, necessary to count the high period of the I2C clock signal, when the I2C interface runs in Fast mode of operation.	RW	0x0

Table 3-1273. PRM_SRAM_COUNT

Address Offset	0x0000 00BC		
Physical Address	0x4AE0 7DBC	**Instance**	DEVICE_PRM
Description	Common setup for SRAM LDO transition counters. Applies to all voltage domains. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STARTUP_COUNT								SLPCNT_VALUE								VSETUPCNT_VALUE								RESERVED	PCHARGECNT_VALUE						

Bits	Field Name	Description	Type	Reset
31:24	STARTUP_COUNT	Determines the start-up duration of SRAM and ABB LDO. The duration is computed as 16 x NbCycles of system clock cycles. Target is 50us.	RW	0x78
23:16	SLPCNT_VALUE	Delay between retention/off assertion of last SRAM bank and SRAMALLRET signal to LDO is driven high. Counting on system clock. Target is 2us.	RW	0x0
15:8	VSETUPCNT_VALUE	SRAM LDO rampup time from retention to active mode. The duration is computed as 8 x NbCycles of system clock cycles. Target is 30us.	RW	0x0
7:6	RESERVED		R	0x0
5:0	PCHARGECNT_VALUE	Delay between de-assertion of standby_rta_ret_on and standby_rta_ret_good. Counting on system clock. Target is 600ns.	RW	0x17

Table 3-1274. Register Call Summary for Register PRM_SRAM_COUNT

PRCM Register Manual
- DEVICE_PRM Register Summary: [0]

Table 3-1275. PRM_SRAM_WKUP_SETUP

Address Offset	0x0000 00C0		
Physical Address	0x4AE0 7DC0	**Instance**	DEVICE_PRM
Description	Setup of memory in WKUP voltage domain. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															ENABLE_RTA

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	ENABLE_RTA	Control for HD memory RTA feature. After PowerOn reset and Efuse sensing, this bitfield is automatically loaded with an Efuse value from control module. Bitfield remains writable after this. 0x0: HD memory RTA feature is disabled 0x1: HD memory RTA feature is enabled	RW	0x0

Table 3-1276. PRM_SLDO_CORE_SETUP

Address Offset	0x0000 00C4		
Physical Address	0x4AE0 7DC4	**Instance**	DEVICE_PRM
Description	Setup of the SRAM LDO for CORE voltage domain. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							AIPOFF	ENFUNC5	ENFUNC4	ENFUNC3	ENFUNC2	ENFUNC1	ABBOFF_SLEEP	ABBOFF_ACT	ENABLE_RTA

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	AIPOFF	Override on AIPOFF input of SRAM LDO. 0x0: AIPOFF signal is not overriden 0x1: AIPOFF signal is overriden to '1'. Corresponding SRAM LDO is disabled and in HZ mode.	RW	0x0
7	ENFUNC5	ENFUNC5 input of SRAM LDO. 0x0: Active to retention is a one step transfer 0x1: Active to retention is a two steps transfer	RW	0x0
6	ENFUNC4	ENFUNC4 input of SRAM LDO. 0x0: One external clock is supplied 0x1: No external clock is supplied	RW	0x0
5	ENFUNC3	ENFUNC3 input of SRAM LDO. After PowerOn reset and Efuse sensing, this bitfield is automatically loaded with an Efuse value from control module. Bitfield remains writable after this. 0x0: Sub regulation is disabled 0x1: Sub regulation is enabled	RW	0x0
4	ENFUNC2	ENFUNC2 input of SRAM LDO. After PowerOn reset and Efuse sensing, this bitfield is automatically loaded with an Efuse value from control module. Bitfield remains writable after this. 0x0: External cap is used 0x1: External cap is not used	RW	0x0
3	ENFUNC1	ENFUNC1 input of SRAM LDO. After PowerOn reset and Efuse sensing, this bitfield is automatically loaded with an Efuse value from control module. Bitfield remains writable after this. 0x0: Short circuit protection is disabled 0x1: Short circuit protection is enabled	RW	0x0
2	ABBOFF_SLEEP	Determines whether SRAMNWA is supplied by VDDS or VDDAR during deep-sleep. After PowerOn reset and Efuse sensing, this bitfield is automatically loaded with an Efuse value from control module. Bitfield remains writable after this. 0x0: SRAMNWA supplied with VDDS 0x1: SRAMNWA supplied with VDDAR	RW	0x0
1	ABBOFF_ACT	Determines whether SRAMNWA is supplied by VDDS or VDDAR during active mode. After PowerOn reset and Efuse sensing, this bitfield is automatically loaded with an Efuse value from control module. Bitfield remains writable after this. 0x0: SRAMNWA supplied with VDDS 0x1: SRAMNWA supplied with VDDAR	RW	0x0

Bits	Field Name	Description	Type	Reset
0	ENABLE_RTA	Control for HD memory RTA feature. After PowerOn reset and Efuse sensing, this bitfield is automatically loaded with an Efuse value from control module. Bitfield remains writable after this. 0x0: HD memory RTA feature is disabled 0x1: HD memory RTA feature is enabled	RW	0x0

Table 3-1277. Register Call Summary for Register PRM_SLDO_CORE_SETUP

PRCM Register Manual
- DEVICE_PRM Register Summary: [0]

Table 3-1278. PRM_SLDO_CORE_CTRL

Address Offset	0x0000 00C8		
Physical Address	0x4AE0 7DC8	**Instance**	DEVICE_PRM
Description	Control and status of the SRAM LDO for CORE voltage domain. [warm reset insensitive]		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						SRAM_IN_TRANSITION	SRAMLDO_STATUS	RESERVED							RETMODE_ENABLE

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	SRAM_IN_TRANSITION	Status indicating SRAM LDO state machine state. 0x0: SRAM LDO state machine is stable 0x1: SRAM LDO state machine is in transition state	R	0x0
8	SRAMLDO_STATUS	SRAMLDO status 0x0: SRAMLDO is in ACTIVE mode. 0x1: SRAMLDO is on RETENTION mode.	R	0x0
7:1	RESERVED		R	0x0
0	RETMODE_ENABLE	Control if the SRAM LDO retention mode is used or not. 0x0: SRAM LDO is not allowed to go to RET mode	R	0x0

Table 3-1279. Register Call Summary for Register PRM_SLDO_CORE_CTRL

PRCM Register Manual
- DEVICE_PRM Register Summary: [0]

Table 3-1280. PRM_SLDO_MPU_SETUP

Address Offset	0x0000 00CC		
Physical Address	0x4AE0 7DCC	**Instance**	DEVICE_PRM
Description	Setup of the SRAM LDO for MPU voltage domain. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							AIPOFF	ENFUNC5	ENFUNC4	ENFUNC3	ENFUNC2	ENFUNC1	ABBOFF_SLEEP	ABBOFF_ACT	ENABLE_RTA

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	AIPOFF	Override on AIPOFF input of SRAM LDO. 0x0: AIPOFF signal is not overriden 0x1: AIPOFF signal is overriden to '1'. Corresponding SRAM LDO is disabled and in HZ mode.	RW	0x0
7	ENFUNC5	ENFUNC5 input of SRAM LDO. 0x0: Active to retention is a one step transfer 0x1: Active to retention is a two steps transfer	RW	0x0
6	ENFUNC4	ENFUNC4 input of SRAM LDO. 0x0: One external clock is supplied 0x1: No external clock is supplied	RW	0x0
5	ENFUNC3	ENFUNC3 input of SRAM LDO. After PowerOn reset and Efuse sensing, this bitfield is automatically loaded with an Efuse value from control module. Bitfield remains writable after this. 0x0: Sub regulation is disabled 0x1: Sub regulation is enabled	RW	0x0
4	ENFUNC2	ENFUNC2 input of SRAM LDO. After PowerOn reset and Efuse sensing, this bitfield is automatically loaded with an Efuse value from control module. Bitfield remains writable after this. 0x0: External cap is used 0x1: External cap is not used	RW	0x0
3	ENFUNC1	ENFUNC1 input of SRAM LDO. After PowerOn reset and Efuse sensing, this bitfield is automatically loaded with an Efuse value from control module. Bitfield remains writable after this. 0x0: Short circuit protection is disabled 0x1: Short circuit protection is enabled	RW	0x0
2	ABBOFF_SLEEP	Determines whether SRAMNWA is supplied by VDDS or VDDAR during deep-sleep. After PowerOn reset and Efuse sensing, this bitfield is automatically loaded with an Efuse value from control module. Bitfield remains writable after this. 0x0: SRAMNWA supplied with VDDS 0x1: SRAMNWA supplied with VDDAR	RW	0x0
1	ABBOFF_ACT	Determines whether SRAMNWA is supplied by VDDS or VDDAR during active mode. After PowerOn reset and Efuse sensing, this bitfield is automatically loaded with an Efuse value from control module. Bitfield remains writable after this. 0x0: SRAMNWA supplied with VDDS 0x1: SRAMNWA supplied with VDDAR	RW	0x0

Bits	Field Name	Description	Type	Reset
0	ENABLE_RTA	Control for HD memory RTA feature. After PowerOn reset and Efuse sensing, this bitfield is automatically loaded with an Efuse value from control module. Bitfield remains writable after this. 0x0: HD memory RTA feature is disabled 0x1: HD memory RTA feature is enabled	RW	0x0

Table 3-1281. PRM_SLDO_MPU_CTRL

Address Offset	0x0000 00D0		
Physical Address	0x4AE0 7DD0	**Instance**	DEVICE_PRM
Description	Control and status of the SRAM LDO for MPU voltage domain. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						SRAM_IN_TRANSITION	SRAMLDO_STATUS	RESERVED							RETMODE_ENABLE

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	SRAM_IN_TRANSITION	Status indicating SRAM LDO state machine state. 0x0: SRAM LDO state machine is stable 0x1: SRAM LDO state machine is in transition state	R	0x0
8	SRAMLDO_STATUS	SRAMLDO status 0x0: SRAMLDO is in ACTIVE mode. 0x1: SRAMLDO is on RETENTION mode.	R	0x0
7:1	RESERVED		R	0x0
0	RETMODE_ENABLE	Control if the SRAM LDO retention mode is used or not. 0x0: SRAM LDO is not allowed to go to RET mode 0x1: SRAM LDO go to RET mode when all memory of voltage domain are OFF or RET	RW	0x0

Table 3-1282. PRM_SLDO_GPU_SETUP

Address Offset	0x0000 00D4		
Physical Address	0x4AE0 7DD4	**Instance**	DEVICE_PRM
Description	Setup of the SRAM LDO for GPU voltage domain. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							AIPOFF	ENFUNC5	ENFUNC4	ENFUNC3	ENFUNC2	ENFUNC1	ABBOFF_SLEEP	ABBOFF_ACT	ENABLE_RTA

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	AIPOFF	Override on AIPOFF input of SRAM LDO. 0x0: AIPOFF signal is not overriden 0x1: AIPOFF signal is overriden to '1'. Corresponding SRAM LDO is disabled and in HZ mode.	RW	0x0
7	ENFUNC5	ENFUNC5 input of SRAM LDO. 0x0: Active to retention is a one step transfer 0x1: Active to retention is a two steps transfer	RW	0x0
6	ENFUNC4	ENFUNC4 input of SRAM LDO. 0x0: One external clock is supplied 0x1: No external clock is supplied	RW	0x0
5	ENFUNC3	ENFUNC3 input of SRAM LDO. After PowerOn reset and Efuse sensing, this bitfield is automatically loaded with an Efuse value from control module. Bitfield remains writable after this. 0x0: Sub regulation is disabled 0x1: Sub regulation is enabled	RW	0x0
4	ENFUNC2	ENFUNC2 input of SRAM LDO. After PowerOn reset and Efuse sensing, this bitfield is automatically loaded with an Efuse value from control module. Bitfield remains writable after this. 0x0: External cap is used 0x1: External cap is not used	RW	0x0
3	ENFUNC1	ENFUNC1 input of SRAM LDO. After PowerOn reset and Efuse sensing, this bitfield is automatically loaded with an Efuse value from control module. Bitfield remains writable after this. 0x0: Short circuit protection is disabled 0x1: Short circuit protection is enabled	RW	0x0
2	ABBOFF_SLEEP	Determines whether SRAMNWA is supplied by VDDS or VDDAR during deep-sleep. After PowerOn reset and Efuse sensing, this bitfield is automatically loaded with an Efuse value from control module. Bitfield remains writable after this. 0x0: SRAMNWA supplied with VDDS 0x1: SRAMNWA supplied with VDDAR	RW	0x0
1	ABBOFF_ACT	Determines whether SRAMNWA is supplied by VDDS or VDDAR during active mode. After PowerOn reset and Efuse sensing, this bitfield is automatically loaded with an Efuse value from control module. Bitfield remains writable after this. 0x0: SRAMNWA supplied with VDDS 0x1: SRAMNWA supplied with VDDAR	RW	0x0
0	ENABLE_RTA	Control for HD memory RTA feature. After PowerOn reset and Efuse sensing, this bitfield is automatically loaded with an Efuse value from control module. Bitfield remains writable after this. 0x0: HD memory RTA feature is disabled 0x1: HD memory RTA feature is enabled	RW	0x0

Table 3-1283. PRM_SLDO_GPU_CTRL

Address Offset	0x0000 00D8		
Physical Address	0x4AE0 7DD8	**Instance**	DEVICE_PRM
Description	Control and status of the SRAM LDO for GPU voltage domain. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						SRAM_IN_TRANSITION	SRAMLDO_STATUS	RESERVED							RETMODE_ENABLE

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	SRAM_IN_TRANSITION	Status indicating SRAM LDO state machine state. 0x0: SRAM LDO state machine is stable 0x1: SRAM LDO state machine is in transition state	R	0x0
8	SRAMLDO_STATUS	SRAMLDO status 0x0: SRAMLDO is in ACTIVE mode. 0x1: SRAMLDO is on RETENTION mode.	R	0x0
7:1	RESERVED		R	0x0
0	RETMODE_ENABLE	Control if the SRAM LDO retention mode is used or not. 0x0: SRAM LDO is not allowed to go to RET mode 0x1: SRAM LDO go to RET mode when all memory of voltage domain are OFF or RET	RW	0x0

Table 3-1284. PRM_ABBLDO_MPU_SETUP

Address Offset	0x0000 00DC		
Physical Address	0x4AE0 7DDC	**Instance**	DEVICE_PRM
Description	Selects the MPU_ABB LDO mode.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																SR2_WTCNT_VALUE								RESERVED			NOCAP	RESERVED	ACTIVE_FBB_SEL	RESERVED	SR2EN

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0
15:8	SR2_WTCNT_VALUE	LDO settling time for active-mode OPP change. Counting at a 16 system clock cycles rate. Target is 50us. [warm reset insensitive]	RW	0x0
7:5	RESERVED		R	0x0
4	NOCAP	Defines whether ABB LDO is cap-less or not. After PowerOn reset and Efuse sensing, this bitfield is automatically loaded with an Efuse value from control module. Bitfield remains writable after this. [warm reset insensitive] 0x0: ABB LDO uses an external cap 0x1: ABB LDO does not use an external cap	RW	0x0
3	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
2	ACTIVE_FBB_SEL	Defines ABB LDO mode when voltage is in slow fast OPP. [warm reset insensitive] 0x0: ABB LDO is in bypass mode 0x1: ABB LDO is in FBB mode	RW	0x0
1	RESERVED		R	0x0
0	SR2EN	Enable ABB power management 0x0: ABB LDO is put in bypass mode 0x1: ABB LDO will operate accordingly to settings	RW	0x0

Table 3-1285. PRM_ABBLDO_MPU_CTRL

Address Offset	0x0000 00E0		
Physical Address	0x4AE0 7DE0	**Instance**	DEVICE_PRM
Description	Control and Status of ABB on MPU voltage domain. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																									SR2_IN_TRANSITION	RESERVED	SR2_STATUS			OPP_CHANGE	OPP_SEL	

Bits	Field Name	Description	Type	Reset
31:7	RESERVED		R	0x0
6	SR2_IN_TRANSITION	Indicates VBBLDO_CON is or is not in transition state. This output should be used by programming interface to clear OPP_CHANGE bit as an indication of OPP change completion. 0x0: IDLE 0x1: Indicates that VBBLDO_CON is in transition and SR2_STATUS bits are not stable to read.	R	0x0
5	RESERVED		R	0x0
4:3	SR2_STATUS	Indicate ABB LDO current operation status 0x0: ABB LDO is placed in bypass mode. 0x1: Reserved 0x2: ABB LDO is placed in FBB active mode. 0x3: Reserved	R	0x0
2	OPP_CHANGE	When OPP_CHANGE is set to 1, VBBLDO_CON samples OPP_SEL ACTIVE_FBB_SEL upon detecting rising edge. VBBLDO_CON asserts signal SR2_IN_TRANSITION in response to OPP_CHANGE. OPP_CHANGE should be cleared to 0 when SR2_IN_TRANSITION from VBBLDO_CON is de-asserted.	RW	0x0
1:0	OPP_SEL	Selects the OPP at which the MPU voltage domain is operating 0x0: default : Nominal 0x1: Fast OPP 0x2: Nominal 0x3: Slow OPP	RW	0x0

Table 3-1286. PRM_ABBLDO_GPU_SETUP

Address Offset	0x0000 00E4		
Physical Address	0x4AE0 7DE4	**Instance**	DEVICE_PRM
Description	Selects the GPU_ABB LDO mode.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																SR2_WTCNT_VALUE								RESERVED			NOCAP	RESERVED	ACTIVE_FBB_SEL	RESERVED	SR2EN

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0
15:8	SR2_WTCNT_VALUE	LDO settling time for active-mode OPP change. Counting at a 16 system clock cycles rate. Target is 50us. [warm reset insensitive]	RW	0x0
7:5	RESERVED		R	0x0
4	NOCAP	Defines whether ABB LDO is cap-less or not. After PowerOn reset and Efuse sensing, this bitfield is automatically loaded with an Efuse value from control module. Bitfield remains writable after this. [warm reset insensitive] 0x0: ABB LDO uses an external cap 0x1: ABB LDO does not use an external cap	RW	0x0
3	RESERVED		R	0x0
2	ACTIVE_FBB_SEL	Defines ABB LDO mode when voltage is in slow fast OPP. [warm reset insensitive] 0x0: ABB LDO is in bypass mode 0x1: ABB LDO is in FBB mode	RW	0x0
1	RESERVED		R	0x0
0	SR2EN	Enable ABB power management 0x0: ABB LDO is put in bypass mode 0x1: ABB LDO will operate accordingly to settings	RW	0x0

Table 3-1287. PRM_ABBLDO_GPU_CTRL

Address Offset	0x0000 00E8		
Physical Address	0x4AE0 7DE8	**Instance**	DEVICE_PRM
Description	Control and Status of ABB on GPU voltage domain. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								SR2_IN_TRANSITION	RESERVED	SR2_STATUS			OPP_CHANGE	OPP_SEL	

Bits	Field Name	Description	Type	Reset
31:7	RESERVED		R	0x0
6	SR2_IN_TRANSITION	Indicates VBBLDO_CON is or is not in transition state. This output should be used by programming interface to clear OPP_CHANGE bit as an indication of OPP change completion. 0x0: IDLE 0x1: Indicates that VBBLDO_CON is in transition and SR2_STATUS bits are not stable to read.	R	0x0
5	RESERVED		R	0x0
4:3	SR2_STATUS	Indicate ABB LDO current operation status 0x0: ABB LDO is placed in bypass mode. 0x1: Reserved 0x2: ABB LDO is placed in FBB active mode. 0x3: Reserved	R	0x0
2	OPP_CHANGE	When OPP_CHANGE is set to 1, VBBLDO_CON samples OPP_SEL ACTIVE_FBB_SEL upon detecting rising edge. VBBLDO_CON asserts signal SR2_IN_TRANSITION in response to OPP_CHANGE. OPP_CHANGE should be cleared to 0 when SR2_IN_TRANSITION from VBBLDO_CON is de-asserted.	RW	0x0
1:0	OPP_SEL	Selects the OPP at which the MM voltage domain is operating (Fast OPP, Nominal OPP or Slow OPP) 0x0: default : Nominal 0x1: Fast OPP 0x2: Nominal 0x3: Slow OPP	RW	0x0

Table 3-1288. PRM_BANDGAP_SETUP

Address Offset	0x0000 00EC		
Physical Address	0x4AE0 7DEC	**Instance**	DEVICE_PRM
Description	Setup of the bandgap. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								STARTUP_COUNT							

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x0
7:0	STARTUP_COUNT	Determines the start-up duration of BANDGAP. The duration is computed as 32 x NbCycles of system clock cycles. Target is 100us.	RW	0x78

Table 3-1289. Register Call Summary for Register PRM_BANDGAP_SETUP

Voltage-Management Functional Description
- BANDGAP Control: [0]

PRCM Register Manual
- DEVICE_PRM Register Summary: [1]

Table 3-1290. PRM_DEVICE_OFF_CTRL

Address Offset	0x0000 00F0		
Physical Address	0x4AE0 7DF0	**Instance**	DEVICE_PRM
Description	This register is used to control device OFF transition.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						EMIF2_OFFWKUP_DISABLE	EMIF1_OFFWKUP_DISABLE	RESERVED							DEVICE_OFF_ENABLE

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	EMIF2_OFFWKUP_DISABLE	Controls the EMIF2_DEVICE_OFFWKUP_CORESRTACTST notifier sent to EMIF1 upon a device wakeup from OFF mode. [warm reset insensitive] 0x0: Notifier is activated. 0x1: Notifier is not activated - stays low	RW	0x0
8	EMIF1_OFFWKUP_DISABLE	Controls the EMIF1_DEVICE_OFFWKUP_CORESRTACTST notifier sent to EMIF2 upon a device wakeup from OFF mode. [warm reset insensitive] 0x0: Notifier is activated. 0x1: Notifier is not activated - stays low	RW	0x0
7:1	RESERVED		R	0x0
0	DEVICE_OFF_ENABLE	Controls transition to device OFF mode. 0x0: Device is not allowed to perform transition to OFF mode 0x1: Device is allowed to perform transition to OFF mode as soon as all power domains in MPU, MM and CORE voltage are in OFF or OSWRET state (open switch retention)	RW	0x0

Table 3-1291. Register Call Summary for Register PRM_DEVICE_OFF_CTRL

PRCM Register Manual
- DEVICE_PRM Register Summary: [0]

Table 3-1292. PRM_PHASE1_CNDP

Address Offset	0x0000 00F4		
Physical Address	0x4AE0 7DF4	**Instance**	DEVICE_PRM
Description	This register stores the start descriptor address of automatic restore phase1. [warm reset insensitive]		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PHASE1_CNDP																															

Bits	Field Name	Description	Type	Reset
31:0	PHASE1_CNDP	Start descriptor address of automatic restore phase1. Hard-coded to SAR_ROM base address.	R	0x4a05e000

Table 3-1293. PRM_PHASE2A_CNDP

Address Offset	0x0000 00F8		
Physical Address	0x4AE0 7DF8	**Instance**	DEVICE_PRM
Description	This register stores the start descriptor address of automatic restore phase2A. [warm reset insensitive]		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PHASE2A_CNDP																															

Bits	Field Name	Description	Type	Reset
31:0	PHASE2A_CNDP	Start descriptor address of automatic restore phase2A. Hard-coded to SAR_ROM base address + 0x30.	R	0x4a05e030

Table 3-1294. PRM_PHASE2B_CNDP

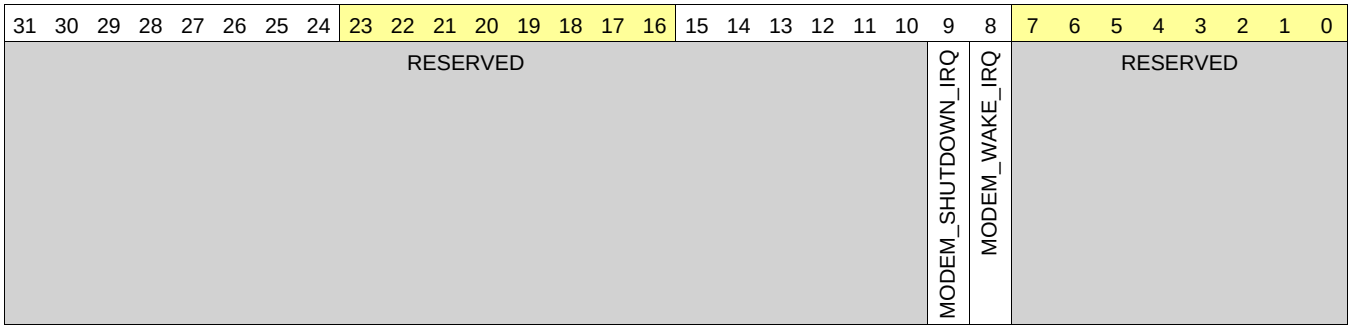
Address Offset	0x0000 00FC		
Physical Address	0x4AE0 7DFC	**Instance**	DEVICE_PRM
Description	This register stores the start descriptor address of automatic restore phase2B. [warm reset insensitive]		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PHASE2B_CNDP																															

Bits	Field Name	Description	Type	Reset
31:0	PHASE2B_CNDP	Start descriptor address of automatic restore phase2B. Hard-coded to SAR_ROM base address + 0x60.	R	0x4a05e060

Table 3-1295. PRM_MODEM_IF_CTRL

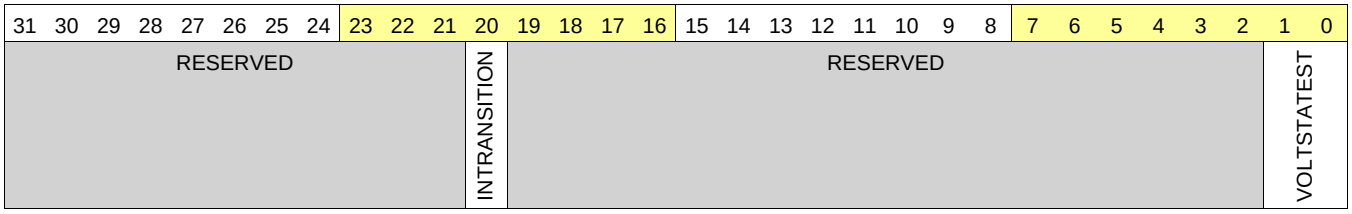
Address Offset	0x0000 0100		
Physical Address	0x4AE0 7E00	**Instance**	DEVICE_PRM
Description	This register is used to control dedicated interfaces between on-chip modem and APE.		
Type	RW		



Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	MODEM_SHUTDOWN_IRQ	Controls an interrupt signal to shutdown modem. 0x0: Interrupt is inactive 0x1: Interrupt is active	RW	0x0
8	MODEM_WAKE_IRQ	Controls an interrupt signal to wakeup modem. 0x0: Interrupt is inactive 0x1: Interrupt is active	RW	0x0
7:0	RESERVED		R	0x0

Table 3-1296. PRM_VOLTST_MPU

Address Offset	0x0000 0110		
Physical Address	0x4AE0 7E10	**Instance**	DEVICE_PRM
Description	This register provides a status on the current MPU voltage domain state. [warm reset insensitive]		
Type	R		



Bits	Field Name	Description	Type	Reset
31:21	RESERVED		R	0x0
20	INTRANSITION	Domain transition status 0x0: No on-going transition on voltage domain 0x1: Voltage domain transition is in progress.	R	0x0
19:2	RESERVED		R	0x0
1:0	VOLTSTATEST	Current voltage state status 0x0: Voltage domain is OFF 0x1: Voltage domain is in RETENTION 0x2: Voltage domain is SLEEP 0x3: Voltage domain is ON	R	0x3

Table 3-1297. PRM_VOLTST_MM

Address Offset	0x0000 0114		
Physical Address	0x4AE0 7E14	**Instance**	DEVICE_PRM
Description	This register provides a status on the current MM voltage domain state. [warm reset insensitive]		

Table 3-1297. PRM_VOLTST_MM (continued)

Type	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED											INTRANSITION	RESERVED																		VOLTSTATEST	

Bits	Field Name	Description	Type	Reset
31:21	RESERVED		R	0x0
20	INTRANSITION	Domain transition status 0x0: No on-going transition on voltage domain 0x1: Voltage domain transition is in progress.	R	0x0
19:2	RESERVED		R	0x0
1:0	VOLTSTATEST	Current voltage state status 0x0: Voltage domain is OFF 0x1: Voltage domain is in RETENTION 0x2: Voltage domain is SLEEP 0x3: Voltage domain is ON	R	0x3

Table 3-1298. PRM_SLDO_DSPEVE_SETUP

Address Offset	0x0000 0118		
Physical Address	0x4AE0 7E18	Instance	DEVICE_PRM
Description	Setup of the SRAM LDO for DSPEVE voltage domain. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							AIPOFF	ENFUNC5	ENFUNC4	ENFUNC3	ENFUNC2	ENFUNC1	ABBOFF_SLEEP	ABBOFF_ACT	ENABLE_RTA

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	AIPOFF	Override on AIPOFF input of SRAM LDO. 0x0: AIPOFF signal is not overriden 0x1: AIPOFF signal is overriden to '1'. Corresponding SRAM LDO is disabled and in HZ mode.	RW	0x0
7	ENFUNC5	ENFUNC5 input of SRAM LDO. 0x0: Active to retention is a one step transfer 0x1: Active to retention is a two steps transfer	RW	0x0
6	ENFUNC4	ENFUNC4 input of SRAM LDO. 0x0: One external clock is supplied 0x1: No external clock is supplied	RW	0x0

Bits	Field Name	Description	Type	Reset
5	ENFUNC3	ENFUNC3 input of SRAM LDO. After PowerOn reset and Efuse sensing, this bitfield is automatically loaded with an Efuse value from control module. Bitfield remains writable after this. 0x0: Sub regulation is disabled 0x1: Sub regulation is enabled	RW	0x0
4	ENFUNC2	ENFUNC2 input of SRAM LDO. After PowerOn reset and Efuse sensing, this bitfield is automatically loaded with an Efuse value from control module. Bitfield remains writable after this. 0x0: External cap is used 0x1: External cap is not used	RW	0x0
3	ENFUNC1	ENFUNC1 input of SRAM LDO. After PowerOn reset and Efuse sensing, this bitfield is automatically loaded with an Efuse value from control module. Bitfield remains writable after this. 0x0: Short circuit protection is disabled 0x1: Short circuit protection is enabled	RW	0x0
2	ABBOFF_SLEEP	Determines whether SRAMNWA is supplied by VDDS or VDDAR during deep-sleep. After PowerOn reset and Efuse sensing, this bitfield is automatically loaded with an Efuse value from control module. Bitfield remains writable after this. 0x0: SRAMNWA supplied with VDDS 0x1: SRAMNWA supplied with VDDAR	RW	0x0
1	ABBOFF_ACT	Determines whether SRAMNWA is supplied by VDDS or VDDAR during active mode. After PowerOn reset and Efuse sensing, this bitfield is automatically loaded with an Efuse value from control module. Bitfield remains writable after this. 0x0: SRAMNWA supplied with VDDS 0x1: SRAMNWA supplied with VDDAR	RW	0x0
0	ENABLE_RTA	Control for HD memory RTA feature. After PowerOn reset and Efuse sensing, this bitfield is automatically loaded with an Efuse value from control module. Bitfield remains writable after this. 0x0: HD memory RTA feature is disabled 0x1: HD memory RTA feature is enabled	RW	0x0

Table 3-1299. Register Call Summary for Register PRM_SLDO_DSPEVE_SETUP

PRCM Register Manual
- DEVICE_PRM Register Summary: [0]

Table 3-1300. PRM_SLDO_IVA_SETUP

Address Offset	0x0000 011C		
Physical Address	0x4AE0 7E1C	**Instance**	DEVICE_PRM
Description	Setup of the SRAM LDO for IVA voltage domain. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							AIPOFF	ENFUNC5	ENFUNC4	ENFUNC3	ENFUNC2	ENFUNC1	ABBOFF_SLEEP	ABBOFF_ACT	ENABLE_RTA

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	AIPOFF	Override on AIPOFF input of SRAM LDO. 0x0: AIPOFF signal is not overriden 0x1: AIPOFF signal is overriden to '1'. Corresponding SRAM LDO is disabled and in HZ mode.	RW	0x0
7	ENFUNC5	ENFUNC5 input of SRAM LDO. 0x0: Active to retention is a one step transfer 0x1: Active to retention is a two steps transfer	RW	0x0
6	ENFUNC4	ENFUNC4 input of SRAM LDO. 0x0: One external clock is supplied 0x1: No external clock is supplied	RW	0x0
5	ENFUNC3	ENFUNC3 input of SRAM LDO. After PowerOn reset and Efuse sensing, this bitfield is automatically loaded with an Efuse value from control module. Bitfield remains writable after this. 0x0: Sub regulation is disabled 0x1: Sub regulation is enabled	RW	0x0
4	ENFUNC2	ENFUNC2 input of SRAM LDO. After PowerOn reset and Efuse sensing, this bitfield is automatically loaded with an Efuse value from control module. Bitfield remains writable after this. 0x0: External cap is used 0x1: External cap is not used	RW	0x0
3	ENFUNC1	ENFUNC1 input of SRAM LDO. After PowerOn reset and Efuse sensing, this bitfield is automatically loaded with an Efuse value from control module. Bitfield remains writable after this. 0x0: Short circuit protection is disabled 0x1: Short circuit protection is enabled	RW	0x0
2	ABBOFF_SLEEP	Determines whether SRAMNWA is supplied by VDDS or VDDAR during deep-sleep. After PowerOn reset and Efuse sensing, this bitfield is automatically loaded with an Efuse value from control module. Bitfield remains writable after this. 0x0: SRAMNWA supplied with VDDS 0x1: SRAMNWA supplied with VDDAR	RW	0x0
1	ABBOFF_ACT	Determines whether SRAMNWA is supplied by VDDS or VDDAR during active mode. After PowerOn reset and Efuse sensing, this bitfield is automatically loaded with an Efuse value from control module. Bitfield remains writable after this. 0x0: SRAMNWA supplied with VDDS 0x1: SRAMNWA supplied with VDDAR	RW	0x0

Bits	Field Name	Description	Type	Reset
0	ENABLE_RTA	Control for HD memory RTA feature. After PowerOn reset and Efuse sensing, this bitfield is automatically loaded with an Efuse value from control module. Bitfield remains writable after this. 0x0: HD memory RTA feature is disabled 0x1: HD memory RTA feature is enabled	RW	0x0

Table 3-1301. PRM_ABBLDO_DSPEVE_CTRL

Address Offset	0x0000 0120		
Physical Address	0x4AE0 7E20	**Instance**	DEVICE_PRM
Description	Control and Status of ABB on DSPEVE voltage domain. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								SR2_IN_TRANSITION	RESERVED	SR2_STATUS			OPP_CHANGE	OPP_SEL	

Bits	Field Name	Description	Type	Reset
31:7	RESERVED		R	0x0
6	SR2_IN_TRANSITION	Indicates VBBLDO_CON is or is not in transition state. This output should be used by programming interface to clear OPP_CHANGE bit as an indication of OPP change completion. 0x0: IDLE 0x1: Indicates that VBBLDO_CON is in transition and SR2_STATUS bits are not stable to read.	R	0x0
5	RESERVED		R	0x0
4:3	SR2_STATUS	Indicate ABB LDO current operation status 0x0: ABB LDO is placed in bypass mode. 0x1: Reserved 0x2: ABB LDO is placed in FBB active mode. 0x3: Reserved	R	0x0
2	OPP_CHANGE	When OPP_CHANGE is set to 1, VBBLDO_CON samples OPP_SEL ACTIVE_FBB_SEL upon detecting rising edge. VBBLDO_CON asserts signal SR2_IN_TRANSITION in response to OPP_CHANGE. OPP_CHANGE should be cleared to 0 when SR2_IN_TRANSITION from VBBLDO_CON is de-asserted.	RW	0x0
1:0	OPP_SEL	Selects the OPP at which the MM voltage domain is operating (Fast OPP, Nominal OPP or Slow OPP) 0x0: default : Nominal 0x1: Fast OPP 0x2: Nominal 0x3: Slow OPP	RW	0x0

Table 3-1302. PRM_ABBLDO_IVA_CTRL

Address Offset	0x0000 0124		
Physical Address	0x4AE0 7E24	**Instance**	DEVICE_PRM
Description	Control and Status of ABB on IVA voltage domain. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																									SR2_IN_TRANSITION	RESERVED	SR2_STATUS		OPP_CHANGE	OPP_SEL	

Bits	Field Name	Description	Type	Reset
31:7	RESERVED		R	0x0
6	SR2_IN_TRANSITION	Indicates VBBLDO_CON is or is not in transition state. This output should be used by programming interface to clear OPP_CHANGE bit as an indication of OPP change completion. 0x0: IDLE 0x1: Indicates that VBBLDO_CON is in transition and SR2_STATUS bits are not stable to read.	R	0x0
5	RESERVED		R	0x0
4:3	SR2_STATUS	Indicate ABB LDO current operation status 0x0: ABB LDO is placed in bypass mode. 0x1: Reserved 0x2: ABB LDO is placed in FBB active mode. 0x3: Reserved	R	0x0
2	OPP_CHANGE	When OPP_CHANGE is set to 1, VBBLDO_CON samples OPP_SEL ACTIVE_FBB_SEL upon detecting rising edge. VBBLDO_CON asserts signal SR2_IN_TRANSITION in response to OPP_CHANGE. OPP_CHANGE should be cleared to 0 when SR2_IN_TRANSITION from VBBLDO_CON is de-asserted.	RW	0x0
1:0	OPP_SEL	Selects the OPP at which the MM voltage domain is operating (Fast OPP, Nominal OPP or Slow OPP) 0x0: default : Nominal 0x1: Fast OPP 0x2: Nominal 0x3: Slow OPP	RW	0x0

Table 3-1303. PRM_SLDO_DSPEVE_CTRL

Address Offset	0x0000 0128		
Physical Address	0x4AE0 7E28	**Instance**	DEVICE_PRM
Description	Control and status of the SRAM LDO for CORE voltage domain. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						SRAM_IN_TRANSITION	SRAMLDO_STATUS	RESERVED							RETMODE_ENABLE

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	SRAM_IN_TRANSITION	Status indicating SRAM LDO state machine state. 0x0: SRAM LDO state machine is stable 0x1: SRAM LDO state machine is in transition state	R	0x0
8	SRAMLDO_STATUS	SRAMLDO status 0x0: SRAMLDO is in ACTIVE mode. 0x1: SRAMLDO is on RETENTION mode.	R	0x0
7:1	RESERVED		R	0x0
0	RETMODE_ENABLE	Control if the SRAM LDO retention mode is used or not. 0x0: SRAM LDO is not allowed to go to RET mode 0x1: SRAM LDO go to RET mode when all memory of voltage domain are OFF or RET	RW	0x0

Table 3-1304. Register Call Summary for Register PRM_SLDO_DSPEVE_CTRL

PRCM Register Manual
- DEVICE_PRM Register Summary: [0]

Table 3-1305. PRM_SLDO_IVA_CTRL

Address Offset	0x0000 012C		
Physical Address	0x4AE0 7E2C	**Instance**	DEVICE_PRM
Description	Control and status of the SRAM LDO for CORE voltage domain. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						SRAM_IN_TRANSITION	SRAMLDO_STATUS	RESERVED							RETMODE_ENABLE

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	SRAM_IN_TRANSITION	Status indicating SRAM LDO state machine state. 0x0: SRAM LDO state machine is stable 0x1: SRAM LDO state machine is in transition state	R	0x0

Bits	Field Name	Description	Type	Reset
8	SRAMLDO_STATUS	SRAMLDO status 0x0: SRAMLDO is in ACTIVE mode. 0x1: SRAMLDO is on RETENTION mode.	R	0x0
7:1	RESERVED		R	0x0
0	RETMODE_ENABLE	Control if the SRAM LDO retention mode is used or not. 0x0: SRAM LDO is not allowed to go to RET mode 0x1: SRAM LDO go to RET mode when all memory of voltage domain are OFF or RET	RW	0x0

Table 3-1306. PRM_ABBLDO_DSPEVE_SETUP

Address Offset	0x0000 0130		
Physical Address	0x4AE0 7E30	**Instance**	DEVICE_PRM
Description	Selects the GPU_ABB LDO mode.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																SR2_WTCNT_VALUE								RESERVED			NOCAP	RESERVED	ACTIVE_FBB_SEL	RESERVED	SR2EN

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0
15:8	SR2_WTCNT_VALUE	LDO settling time for active-mode OPP change. Counting at a 16 system clock cycles rate. Target is 50us. [warm reset insensitive]	RW	0x0
7:5	RESERVED		R	0x0
4	NOCAP	Defines whether ABB LDO is cap-less or not. After PowerOn reset and Efuse sensing, this bitfield is automatically loaded with an Efuse value from control module. Bitfield remains writable after this. [warm reset insensitive] 0x0: ABB LDO uses an external cap 0x1: ABB LDO does not use an external cap	RW	0x0
3	RESERVED		R	0x0
2	ACTIVE_FBB_SEL	Defines ABB LDO mode when voltage is in slow fast OPP. [warm reset insensitive] 0x0: ABB LDO is in bypass mode 0x1: ABB LDO is in FBB mode	RW	0x0
1	RESERVED		R	0x0
0	SR2EN	Enable ABB power management 0x0: ABB LDO is put in bypass mode 0x1: ABB LDO will operate accordingly to settings	RW	0x0

Table 3-1307. PRM_ABBLDO_IVA_SETUP

Address Offset	0x0000 0134		
Physical Address	0x4AE0 7E34	**Instance**	DEVICE_PRM
Description	Selects the GPU_ABB LDO mode.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																SR2_WTCNT_VALUE								RESERVED			NOCAP	RESERVED	ACTIVE_FBB_SEL	RESERVED	SR2EN

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0
15:8	SR2_WTCNT_VALUE	LDO settling time for active-mode OPP change. Counting at a 16 system clock cycles rate. Target is 50us. [warm reset insensitive]	RW	0x0
7:5	RESERVED		R	0x0
4	NOCAP	Defines whether ABB LDO is cap-less or not. After PowerOn reset and Efuse sensing, this bitfield is automatically loaded with an Efuse value from control module. Bitfield remains writable after this. [warm reset insensitive] 0x0: ABB LDO uses an external cap 0x1: ABB LDO does not use an external cap	RW	0x0
3	RESERVED		R	0x0
2	ACTIVE_FBB_SEL	Defines ABB LDO mode when voltage is in slow fast OPP. [warm reset insensitive] 0x0: ABB LDO is in bypass mode 0x1: ABB LDO is in FBB mode	RW	0x0
1	RESERVED		R	0x0
0	SR2EN	Enable ABB power management 0x0: ABB LDO is put in bypass mode 0x1: ABB LDO will operate accordingly to settings	RW	0x0

3.12.36 DSP1_PRM registers

3.12.36.1 DSP1_PRM Register Summary

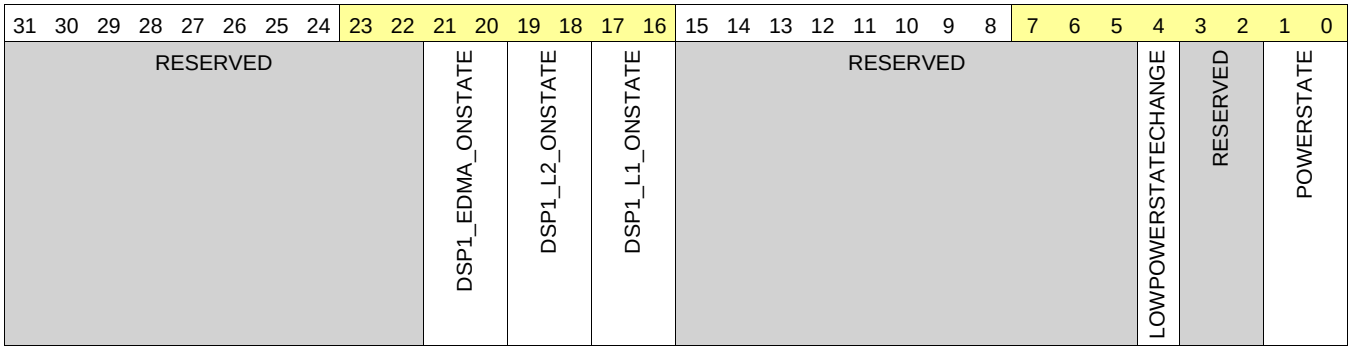
Table 3-1308. DSP1_PRM Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	DSP1_PRM Physical Address
PM_DSP1_PWRSTCTRL	RW	32	0x0000 0000	0x4AE0 6400
PM_DSP1_PWRSTST	RW	32	0x0000 0004	0x4AE0 6404
RM_DSP1_RSTCTRL	RW	32	0x0000 0010	0x4AE0 6410
RM_DSP1_RSTST	RW	32	0x0000 0014	0x4AE0 6414
RM_DSP1_DSP1_CONTEXT	RW	32	0x0000 0024	0x4AE0 6424

3.12.36.2 DSP1_PRM Register Description

Table 3-1309. PM_DSP1_PWRSTCTRL

Address Offset	0x0000 0000		
Physical Address	0x4AE0 6400	**Instance**	DSP1_PRM
Description	This register controls the DSP power state to reach upon a domain sleep transition		
Type	RW		



Bits	Field Name	Description	Type	Reset
31:22	RESERVED		R	0x0
21:20	DSP1_EDMA_ONSTATE	DSP_EDMA state when domain is ON. 0x3: Memory bank is on when the domain is ON.	R	0x3
19:18	DSP1_L2_ONSTATE	DSP_L2 state when domain is ON. 0x3: Memory bank is on when the domain is ON.	R	0x3
17:16	DSP1_L1_ONSTATE	DSP_L1 state when domain is ON. 0x3: Memory bank is on when the domain is ON.	R	0x3
15:5	RESERVED		R	0x0
4	LOWPOWERSTATECHANGE	Power state change request when domain has already performed a sleep transition. Allows going into deeper low power state without waking up the power domain. 0x0: Do not request a low power state change. 0x1: Request a low power state change. This bit is automatically cleared when the power state is effectively changed or when power state is ON.	RW	0x0
3:2	RESERVED		R	0x0
1:0	POWERSTATE	Power state control 0x0: OFF state 0x1: RESERVED 0x2: INACTIVE state 0x3: ON State	RW	0x3

Table 3-1310. Register Call Summary for Register PM_DSP1_PWRSTCTRL

Power Management Functional Description • Logic and Memory Area Power Modes Control and Status: [0][1][2][3][4]
PRCM Register Manual • Not Supported Functionality (Registers and Bitfields): [5] • DSP1_PRM Register Summary: [6]

Table 3-1311. PM_DSP1_PWRSTST

Address Offset	0x0000 0004		
Physical Address	0x4AE0 6404	**Instance**	DSP1_PRM
Description	This register provides a status on the DSP domain current power state. [warm reset insensitive]		
Type	RW		

31 30 29 28 27 26	25 24	23 22 21	20	19 18 17 16	15 14 13 12 11 10	9 8	7 6	5 4	3	2	1 0
RESERVED	LASTPOWERSTATEENTERED	RESERVED	INTRANSITION	RESERVED		DSP1_EDMA_STATEST	DSP1_L2_STATEST	DSP1_L1_STATEST	RESERVED	LOGICSTATEST	POWERSTATEST

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:24	LASTPOWERSTATEENTERED	Last low power state entered. Set to 0x3 upon write of the same only. This register is intended for debug purpose only. 0x0: Power domain was previously OFF 0x1: Power domain was previously in RETENTION 0x2: Power domain was previously ON-INACTIVE 0x3: Power domain was previously ON-ACTIVE	RW	0x0
23:21	RESERVED		R	0x0
20	INTRANSITION	Domain transition status 0x0: No on-going transition on power domain 0x1: Power domain transition is in progress.	R	0x0
19:10	RESERVED		R	0x0
9:8	DSP1_EDMA_STATEST	DSP_EDMA memory state status 0x0: Memory is OFF 0x1: RESERVED 0x2: Reserved 0x3: Memory is ON	R	0x3
7:6	DSP1_L2_STATEST	DSP_L2 memory state status 0x0: Memory is OFF 0x1: RESERVED 0x2: Reserved 0x3: Memory is ON	R	0x3
5:4	DSP1_L1_STATEST	DSP_L1 memory state status 0x0: Memory is OFF 0x1: RESERVED 0x2: Reserved 0x3: Memory is ON	R	0x3
3	RESERVED		R	0x0
2	LOGICSTATEST	Logic state status 0x0: Logic in domain is OFF 0x1: Logic in domain is ON	R	0x1
1:0	POWERSTATEST	Current power state status 0x0: Power domain is OFF 0x1: Power domain is in RETENTION 0x2: Power domain is ON-INACTIVE 0x3: Power domain is ON-ACTIVE	R	0x3

Table 3-1312. Register Call Summary for Register PM_DSP1_PWRSTST

Power Management Functional Description
• Logic and Memory Area Power Modes Control and Status: [0][1][2][3][4][5][6]
PRCM Register Manual
• Not Supported Functionality (Registers and Bitfields): [7][8][9][10][11]
• DSP1_PRM Register Summary: [12]

Table 3-1313. RM_DSP1_RSTCTRL

Address Offset	0x0000 0010		
Physical Address	0x4AE0 6410	**Instance**	DSP1_PRM
Description	This register controls the release of the DSP sub-system resets.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														RST_DSP1	RST_DSP1_LRST

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1	RST_DSP1	DSP reset control 0x0: Reset is cleared for the MMU, cache and slave interface 0x1: Reset is asserted for the MMU, cache and slave interface	RW	0x1
0	RST_DSP1_LRST	DSP Local reset control 0x0: Reset is cleared for the DSP - DSP 0x1: Reset is asserted for the DSP - DSP	RW	0x1

Table 3-1314. Register Call Summary for Register RM_DSP1_RSTCTRL

Reset Management Functional Description
• Reset Domains: [0][1][2][3]
• DSP1 Subsystem Power-On Reset Sequence: [4][5]
• DSP1 Subsystem Software Warm Reset Sequence: [6][7][8][9]
PRCM Register Manual
• DSP1_PRM Register Summary: [10]

Table 3-1315. RM_DSP1_RSTST

Address Offset	0x0000 0014		
Physical Address	0x4AE0 6414	**Instance**	DSP1_PRM
Description	This register logs the different reset sources of the DSP domain. Each bit is set upon release of the domain reset signal. Must be cleared by software. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												RST_DSP1_EMU_REQ	RST_DSP1_EMU	RST_DSP1	RST_DSP1_LRST

Bits	Field Name	Description	Type	Reset
31:4	RESERVED		R	0x0
3	RST_DSP1_EMU_REQ	DSP processor has been reset due to DSP emulation reset request driven from DSP-SS 0x0: No emulation reset 0x1: DSP DSP has been reset upon emulation reset request	RW	0x0
2	RST_DSP1_EMU	DSP domain has been reset due to emulation reset source e.g. assert reset command initiated by the icepick module 0x0: No emulation reset 0x1: DSP has been reset upon emulation reset	RW	0x0
1	RST_DSP1	DSP SW reset status 0x0: No SW reset occurred 0x1: MMU, cache and slave interface has been reset upon SW reset	RW	0x0
0	RST_DSP1_LRST	DSP Local SW reset 0x0: No SW reset occurred 0x1: DSP has been reset upon SW reset	RW	0x0

Table 3-1316. Register Call Summary for Register RM_DSP1_RSTST

PRCM Register Manual
- DSP1_PRM Register Summary: [0]

Table 3-1317. RM_DSP1_DSP1_CONTEXT

Address Offset	0x0000 0024		
Physical Address	0x4AE0 6424	**Instance**	DSP1_PRM
Description	This register contains dedicated DSP context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																					LOSTMEM_DSP_EDMA	LOSTMEM_DSP_L2	LOSTMEM_DSP_L1	RESERVED							LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:11	RESERVED		R	0x0
10	LOSTMEM_DSP_EDMA	Specify if memory-based context in DSP_EDMA memory bank has been lost due to a previous power transition or other reset source. 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
9	LOSTMEM_DSP_L2	Specify if memory-based context in DSP_L2 memory bank has been lost due to a previous power transition or other reset source. 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
8	LOSTMEM_DSP_L1	Specify if memory-based context in DSP_L1 memory bank has been lost due to a previous power transition or other reset source. 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
7:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of DSP_SYS_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1318. Register Call Summary for Register RM_DSP1_DSP1_CONTEXT

Power Management Functional Description • PD_DSP1 Description: [0]
PRCM Register Manual • DSP1_PRM Register Summary: [1]

3.12.37 DSP2_PRM registers

3.12.37.1 DSP2_PRM Register Summary

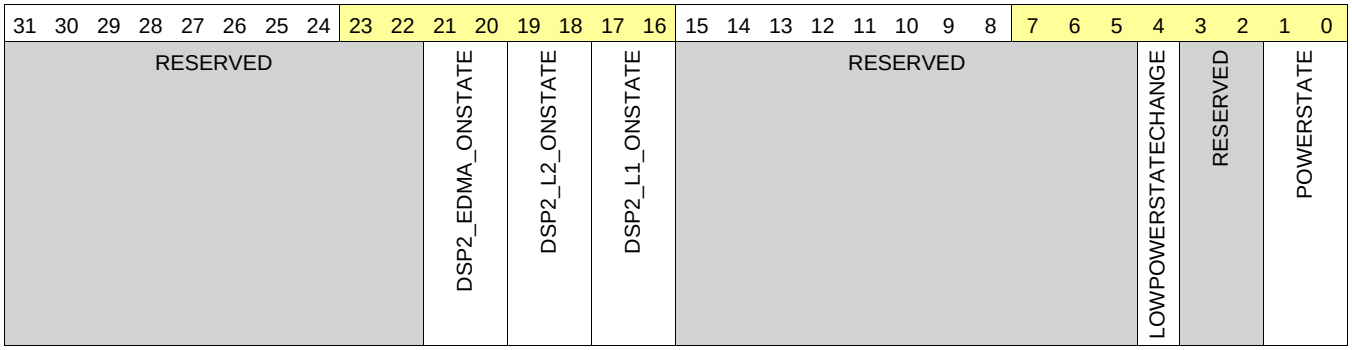
Table 3-1319. DSP2_PRM Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	DSP2_PRM Physical Address
PM_DSP2_PWRSTCTRL	RW	32	0x0000 0000	0x4AE0 7B00
PM_DSP2_PWRSTST	RW	32	0x0000 0004	0x4AE0 7B04
RM_DSP2_RSTCTRL	RW	32	0x0000 0010	0x4AE0 7B10
RM_DSP2_RSTST	RW	32	0x0000 0014	0x4AE0 7B14
RM_DSP2_DSP2_CONTEXT	RW	32	0x0000 0024	0x4AE0 7B24

3.12.37.2 DSP2_PRM Register Description

Table 3-1320. PM_DSP2_PWRSTCTRL

Address Offset	0x0000 0000		
Physical Address	0x4AE0 7B00	**Instance**	DSP2_PRM
Description	This register controls the DSP power state to reach upon a domain sleep transition		
Type	RW		



Bits	Field Name	Description	Type	Reset
31:22	RESERVED		R	0x0
21:20	DSP2_EDMA_ONSTATE	DSP_EDMA state when domain is ON. 0x3: Memory bank is on when the domain is ON.	R	0x3
19:18	DSP2_L2_ONSTATE	DSP_L2 state when domain is ON. 0x3: Memory bank is on when the domain is ON.	R	0x3
17:16	DSP2_L1_ONSTATE	DSP_L1 state when domain is ON. 0x3: Memory bank is on when the domain is ON.	R	0x3
15:5	RESERVED		R	0x0
4	LOWPOWERSTATECHANGE	Power state change request when domain has already performed a sleep transition. Allows going into deeper low power state without waking up the power domain. 0x0: Do not request a low power state change. 0x1: Request a low power state change. This bit is automatically cleared when the power state is effectively changed or when power state is ON.	RW	0x0
3:2	RESERVED		R	0x0
1:0	POWERSTATE	Power state control 0x0: OFF 0x1: RESERVED 0x2: INACTIVE state 0x3: ON State	RW	0x3

Table 3-1321. Register Call Summary for Register PM_DSP2_PWRSTCTRL

Power Management Functional Description • Logic and Memory Area Power Modes Control and Status: [0][1][2][3][4]
PRCM Register Manual • Not Supported Functionality (Registers and Bitfields): [5] • DSP2_PRM Register Summary: [6]

Table 3-1322. PM_DSP2_PWRSTST

Address Offset	0x0000 0004		
Physical Address	0x4AE0 7B04	**Instance**	DSP2_PRM
Description	This register provides a status on the DSP domain current power state. [warm reset insensitive]		
Type	RW		

31 30 29 28 27 26	25 24	23 22 21	20	19 18 17 16	15 14 13 12 11 10	9 8	7 6	5 4	3	2	1 0
RESERVED	LASTPOWERSTATEENTERED	RESERVED	INTRANSITION	RESERVED		DSP2_EDMA_STATEST	DSP2_L2_STATEST	DSP2_L1_STATEST	RESERVED	LOGICSTATEST	POWERSTATEST

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:24	LASTPOWERSTATEENTERED	Last low power state entered. Set to 0x3 upon write of the same only. This register is intended for debug purpose only. 0x0: Power domain was previously OFF 0x1: Power domain was previously in RETENTION 0x2: Power domain was previously ON-INACTIVE 0x3: Power domain was previously ON-ACTIVE	RW	0x0
23:21	RESERVED		R	0x0
20	INTRANSITION	Domain transition status 0x0: No on-going transition on power domain 0x1: Power domain transition is in progress.	R	0x0
19:10	RESERVED		R	0x0
9:8	DSP2_EDMA_STATEST	DSP_EDMA memory state status 0x0: Memory is OFF 0x1: Memory is RETENTION 0x2: Reserved 0x3: Memory is ON	R	0x3
7:6	DSP2_L2_STATEST	DSP_L2 memory state status 0x0: Memory is OFF 0x1: RESERVED 0x2: Reserved 0x3: Memory is ON	R	0x3
5:4	DSP2_L1_STATEST	DSP_L1 memory state status 0x0: Memory is OFF 0x1: RESERVED 0x2: Reserved 0x3: Memory is ON	R	0x3
3	RESERVED		R	0x0
2	LOGICSTATEST	Logic state status 0x0: Logic in domain is OFF 0x1: Logic in domain is ON	R	0x1
1:0	POWERSTATEST	Current power state status 0x0: Power domain is OFF 0x1: Power domain is in RETENTION 0x2: Power domain is ON-INACTIVE 0x3: Power domain is ON-ACTIVE	R	0x3

Table 3-1323. Register Call Summary for Register PM_DSP2_PWRSTST

Power Management Functional Description • Logic and Memory Area Power Modes Control and Status: [0][1][2][3][4][5][6]
PRCM Register Manual • Not Supported Functionality (Registers and Bitfields): [7][8][9][10][11][12] • DSP2_PRM Register Summary: [13]

Table 3-1324. RM_DSP2_RSTCTRL

Address Offset	0x0000 0010		
Physical Address	0x4AE0 7B10	**Instance**	DSP2_PRM
Description	This register controls the release of the DSP sub-system resets.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														RST_DSP2	RST_DSP2_LRST

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1	RST_DSP2	DSP SW reset control 0x0: Reset is cleared for the MMU, cache and slave interface 0x1: Reset is asserted for the MMU, cache and slave interface	RW	0x1
0	RST_DSP2_LRST	DSP Local reset control 0x0: Reset is cleared for the DSP - DSP 0x1: Reset is asserted for the DSP - DSP	RW	0x1

Table 3-1325. Register Call Summary for Register RM_DSP2_RSTCTRL

Reset Management Functional Description • Reset Domains: [0][1][2][3] • DSP2 Subsystem Power-On Reset Sequence: [4][5] • DSP2 Subsystem Software Warm Reset Sequence: [6][7][8][9]
PRCM Register Manual • DSP2_PRM Register Summary: [10]

Table 3-1326. RM_DSP2_RSTST

Address Offset	0x0000 0014		
Physical Address	0x4AE0 7B14	**Instance**	DSP2_PRM
Description	This register logs the different reset sources of the DSP domain. Each bit is set upon release of the domain reset signal. Must be cleared by software. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												RST_DSP2_EMU_REQ	RST_DSP2_EMU	RST_DSP2	RST_DSP2_LRST

Bits	Field Name	Description	Type	Reset
31:4	RESERVED		R	0x0
3	RST_DSP2_EMU_REQ	DSP processor has been reset due to DSP emulation reset request driven from DSP-SS 0x0: No emulation reset 0x1: DSP DSP has been reset upon emulation reset request	RW	0x0
2	RST_DSP2_EMU	DSP domain has been reset due to emulation reset source e.g. assert reset command initiated by the icepick module 0x0: No emulation reset 0x1: DSP has been reset upon emulation reset	RW	0x0
1	RST_DSP2	DSP SW reset status 0x0: No SW reset occurred 0x1: MMU, cache and slave interface has been reset upon SW reset	RW	0x0
0	RST_DSP2_LRST	DSP Local SW reset 0x0: No SW reset occurred 0x1: DSP has been reset upon SW reset	RW	0x0

Table 3-1327. Register Call Summary for Register RM_DSP2_RSTST

PRCM Register Manual
- DSP2_PRM Register Summary: [0]

Table 3-1328. RM_DSP2_DSP2_CONTEXT

Address Offset	0x0000 0024		
Physical Address	0x4AE0 7B24	**Instance**	DSP2_PRM
Description	This register contains dedicated DSP context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																					LOSTMEM_DSP_EDMA	LOSTMEM_DSP_L2	LOSTMEM_DSP_L1	RESERVED							LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:11	RESERVED		R	0x0
10	LOSTMEM_DSP_EDMA	Specify if memory-based context in DSP_EDMA memory bank has been lost due to a previous power transition or other reset source. 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
9	LOSTMEM_DSP_L2	Specify if memory-based context in DSP_L2 memory bank has been lost due to a previous power transition or other reset source. 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
8	LOSTMEM_DSP_L1	Specify if memory-based context in DSP_L1 memory bank has been lost due to a previous power transition or other reset source. 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
7:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of DSP_SYS_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1329. Register Call Summary for Register RM_DSP2_DSP2_CONTEXT

Power Management Functional Description • PD_DSP2 Description: [0]
PRCM Register Manual • DSP2_PRM Register Summary: [1]

3.12.38 DSS_PRM registers

3.12.38.1 DSS_PRM Register Summary

Table 3-1330. DSS_PRM Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	DSS_PRM Physical Address
PM_DSS_PWRSTCTRL	RW	32	0x0000 0000	0x4AE0 7100
PM_DSS_PWRSTST	RW	32	0x0000 0004	0x4AE0 7104
PM_DSS_DSS_WKDEP	RW	32	0x0000 0020	0x4AE0 7120
RM_DSS_DSS_CONTEXT	RW	32	0x0000 0024	0x4AE0 7124
PM_DSS_DSS2_WKDEP	RW	32	0x0000 0028	0x4AE0 7128
RM_DSS_BB2D_CONTEXT	RW	32	0x0000 0034	0x4AE0 7134
RM_DSS_SDVENC_CONTEXT	RW	32	0x0000 003C	0x4AE0 713C

3.12.38.2 DSS_PRM Register Description

Table 3-1331. PM_DSS_PWRSTCTRL

Address Offset	0x0000 0000		
Physical Address	0x4AE0 7100	**Instance**	DSS_PRM
Description	This register controls the DSS power state to reach upon a domain sleep transition		

Table 3-1331. PM_DSS_PWRSTCTRL (continued)

Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														DSS_MEM_ONSTATE		RESERVED							DSS_MEM_RETSTATE	RESERVED			LOWPOWERSTATECHANGE	RESERVED	LOGICRETSTATE	POWERSTATE	

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	DSS_MEM_ONSTATE	DSS_MEM state when domain is ON. 0x3: Memory bank is on when the domain is ON.	R	0x3
15:9	RESERVED		R	0x0
8	DSS_MEM_RETSTATE	DSS_MEM state when domain is RETENTION. 0x0: Memory bank is off when the domain is in the RETENTION state.	R	0x0
7:5	RESERVED		R	0x0
4	LOWPOWERSTATECHANGE	Power state change request when domain has already performed a sleep transition. Allows going into deeper low power state without waking up the power domain. 0x0: Do not request a low power state change. 0x1: Request a low power state change. This bit is automatically cleared when the power state is effectively changed or when power state is ON.	RW	0x0
3	RESERVED		R	0x0
2	LOGICRETSTATE	Logic state when power domain is RETENTION 0x0: Whole logic is off when the domain is in RETENTION state.	R	0x0
1:0	POWERSTATE	Power state control 0x0: OFF state 0x1: RETENTION state 0x2: INACTIVE state 0x3: ON State	RW	0x0

Table 3-1332. Register Call Summary for Register PM_DSS_PWRSTCTRL

Power Management Functional Description

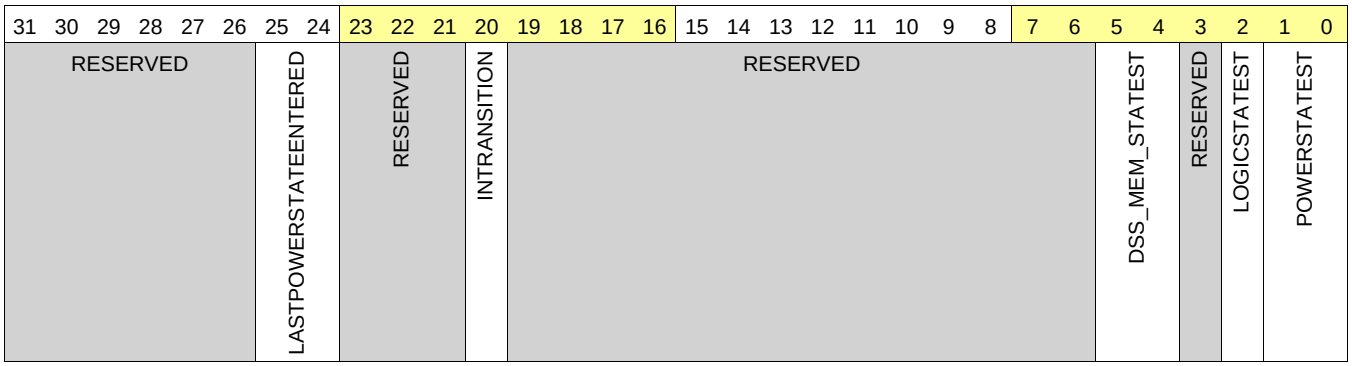
- Logic and Memory Area Power Mode Control and Status: [0][1][2][3][4]

PRCM Register Manual

- Not Supported Functionality (Registers and Bitfields): [5][6][7][8]
- DSS_PRM Register Summary: [9]

Table 3-1333. PM_DSS_PWRSTST

Address Offset	0x0000 0004		
Physical Address	0x4AE0 7104	**Instance**	DSS_PRM
Description	This register provides a status on the current DSS power domain state. [warm reset insensitive]		
Type	RW		



Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:24	LASTPOWERSTATEENTERED	Last low power state entered. Set to 0x3 upon write of the same only. This register is intended for debug purpose only. 0x0: Power domain was previously OFF 0x1: Power domain was previously in RETENTION 0x2: Power domain was previously ON-INACTIVE 0x3: Power domain was previously ON-ACTIVE	RW	0x0
23:21	RESERVED		R	0x0
20	INTRANSITION	Domain transition status 0x0: No on-going transition on power domain 0x1: Power domain transition is in progress.	R	0x0
19:6	RESERVED		R	0x0
5:4	DSS_MEM_STATEST	DSS_MEM state status 0x0: Memory is OFF 0x1: Reserved 0x2: Reserved 0x3: Memory is ON	R	0x3
3	RESERVED		R	0x0
2	LOGICSTATEST	Logic state status 0x0: Logic in domain is OFF 0x1: Logic in domain is ON	R	0x1
1:0	POWERSTATEST	Current power state status 0x0: Power domain is OFF 0x1: Power domain is in RETENTION 0x2: Power domain is ON-INACTIVE 0x3: Power domain is ON-ACTIVE	R	0x3

Table 3-1334. Register Call Summary for Register PM_DSS_PWRSTST

Power Management Functional Description

- Logic and Memory Area Power Mode Control and Status: [0][1][2][3][4]

PRCM Register Manual

- Not Supported Functionality (Registers and Bitfields): [5][6][7][8][9]
- DSS_PRM Register Summary: [10]

Table 3-1335. PM_DSS_DSS_WKDEP

Address Offset	0x0000 0020		
Physical Address	0x4AE0 7120	**Instance**	DSS_PRM
Description	This register controls wakeup dependency based on DSS service requests.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED		WKUPDEP_DSI1_B_EVE4	WKUPDEP_DSI1_B_EVE3	WKUPDEP_DSI1_B_EVE2	WKUPDEP_DSI1_B_EVE1	WKUPDEP_DSI1_B_DSP2	WKUPDEP_DSI1_B_IPU1	WKUPDEP_DSI1_B_SDMA	WKUPDEP_DSI1_B_DSP1	WKUPDEP_DSI1_B_IPU2	WKUPDEP_DSI1_B_MPU	WKUPDEP_DSI1_A_EVE4	WKUPDEP_DSI1_A_EVE3	WKUPDEP_DSI1_A_EVE2	WKUPDEP_DSI1_A_EVE1	WKUPDEP_DSI1_A_DSP2	WKUPDEP_DSI1_A_IPU1	WKUPDEP_DSI1_A_SDMA	WKUPDEP_DSI1_A_DSP1	WKUPDEP_DSI1_A_IPU2	WKUPDEP_DSI1_A_MPU	WKUPDEP_DISPC_EVE4	WKUPDEP_DISPC_EVE3	WKUPDEP_DISPC_EVE2	WKUPDEP_DISPC_EVE1	WKUPDEP_DISPC_DSP2	WKUPDEP_DISPC_IPU1	WKUPDEP_DISPC_SDMA	WKUPDEP_DISPC_DSP1	WKUPDEP_DISPC_IPU2	WKUPDEP_DISPC_MPU

Bits	Field Name	Description	Type	Reset
31:30	RESERVED		R	0x0
29	WKUPDEP_DSI1_B_EVE4	Wakeup dependency from DSS module (SWakeup signal) towards EVE4 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
28	WKUPDEP_DSI1_B_EVE3	Wakeup dependency from DSS module (SWakeup signal) towards EVE3 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
27	WKUPDEP_DSI1_B_EVE2	Wakeup dependency from DSS module (SWakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
26	WKUPDEP_DSI1_B_EVE1	Wakeup dependency from DSS module (SWakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
25	WKUPDEP_DSI1_B_DSP2	Wakeup dependency from DSS module (SWakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
24	WKUPDEP_DSI1_B_IPU1	Wakeup dependency from DSS module (SWakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
23	WKUPDEP_DSI1_B_SDMA	Wakeup dependency from DSS module (SWakeup signal) towards SDMA + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
22	WKUPDEP_DSI1_B_DSP1	Wakeup dependency from DSS module (SWakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
21	WKUPDEP_DSI1_B_IPU2	Wakeup dependency from DSS module (SWakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
20	WKUPDEP_DSI1_B_MPU	Wakeup dependency from DSS module (SWakeup signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
19	WKUPDEP_DSI1_A_EVE4	Wakeup dependency from DSS module (SWakeup signal) towards EVE4 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
18	WKUPDEP_DSI1_A_EVE3	Wakeup dependency from DSS module (SWakeup signal) towards EVE3 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
17	WKUPDEP_DSI1_A_EVE2	Wakeup dependency from DSS module (SWakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
16	WKUPDEP_DSI1_A_EVE1	Wakeup dependency from DSS module (SWakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
15	WKUPDEP_DSI1_A_DSP2	Wakeup dependency from DSS module (SWakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
14	WKUPDEP_DSI1_A_IPU1	Wakeup dependency from DSS module (SWakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
13	WKUPDEP_DSI1_A_SDMA	Wakeup dependency from DSS module (SWakeup signal) towards SDMA + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
12	WKUPDEP_DSI1_A_DSP1	Wakeup dependency from DSS module (SWakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
11	WKUPDEP_DSI1_A_IPU2	Wakeup dependency from DSS module (SWakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
10	WKUPDEP_DSI1_A_MPU	Wakeup dependency from DSS module (SWakeup signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
9	WKUPDEP_DISPC_EVE4	Wakeup dependency from DSS module (SWakeup signal) towards EVE4 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
8	WKUPDEP_DISPC_EVE3	Wakeup dependency from DSS module (SWakeup signal) towards EVE3 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
7	WKUPDEP_DISPC_EVE2	Wakeup dependency from DSS module (SWakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_DISPC_EVE1	Wakeup dependency from DSS module (SWakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_DISPC_DSP2	Wakeup dependency from DSS module (SWakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_DISPC_IPU1	Wakeup dependency from DSS module (SWakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	WKUPDEP_DISPC_SDMA	Wakeup dependency from DSS module (SWakeup signal) towards SDMA + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
2	WKUPDEP_DISPC_DSP1	Wakeup dependency from DSS module (SWakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_DISPC_IPU2	Wakeup dependency from DSS module (SWakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_DISPC_MPU	Wakeup dependency from DSS module (SWakeup signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1336. Register Call Summary for Register PM_DSS_DSS_WKDEP

Clock Management Functional Description
- Clock Domain Dependency: [0][1][2][3]

PRCM Register Manual
- Not Supported Functionality (Registers and Bitfields): [4][5][6][7][8][9][10][11][12][13][14][15][16][17][18][19][20][21][22][23][24][25][26][27][28][29]
- DSS_PRM Register Summary: [30]

Table 3-1337. RM_DSS_DSS_CONTEXT

Address Offset	0x0000 0024		
Physical Address	0x4AE0 7124	**Instance**	DSS_PRM
Description	This register contains dedicated DSS context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							LOSTMEM_DSS_MEM	RESERVED						LOSTCONTEXT_RFF	LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	LOSTMEM_DSS_MEM	Specify if memory-based context in DSS_MEM memory bank has been lost due to a previous power transition or other reset source. 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
7:2	RESERVED		R	0x0
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of DSS_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Bits	Field Name	Description	Type	Reset
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of DSS_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1338. Register Call Summary for Register RM_DSS_DSS_CONTEXT

Power Management Functional Description
- PD_DSS Description: [0][1]

PRCM Register Manual
- DSS_PRM Register Summary: [2]

Table 3-1339. PM_DSS_DSS2_WKDEP

Address Offset	0x0000 0028		
Physical Address	0x4AE0 7128	**Instance**	DSS_PRM
Description	This register controls wakeup dependency based on DSS service requests.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED						WKUPDEP_HDMIDMA_DSP2	RESERVED	WKUPDEP_HDMIDMA_SDMA	WKUPDEP_HDMIDMA_DSP1	RESERVED		WKUPDEP_DSI1_C_EVE4	WKUPDEP_DSI1_C_EVE3	WKUPDEP_DSI1_C_EVE2	WKUPDEP_DSI1_C_EVE1	WKUPDEP_DSI1_C_DSP2	WKUPDEP_DSI1_C_IPU1	WKUPDEP_DSI1_C_SDMA	WKUPDEP_DSI1_C_DSP1	WKUPDEP_DSI1_C_IPU2	WKUPDEP_DSI1_C_MPU	WKUPDEP_HDMIIRQ_EVE4	WKUPDEP_HDMIIRQ_EVE3	WKUPDEP_HDMIIRQ_EVE2	WKUPDEP_HDMIIRQ_EVE1	WKUPDEP_HDMIIRQ_DSP2	WKUPDEP_HDMIIRQ_IPU1	RESERVED	WKUPDEP_HDMIIRQ_DSP1	WKUPDEP_HDMIIRQ_IPU2	WKUPDEP_HDMIIRQ_MPU

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	WKUPDEP_HDMIDMA_DSP2	Wakeup dependency from DSS module (SWakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
24	RESERVED		R	0x0
23	WKUPDEP_HDMIDMA_SDMA	Wakeup dependency from DSS module (SWakeup signal) towards SDMA + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
22	WKUPDEP_HDMIDMA_DSP1	Wakeup dependency from DSS module (SWakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
21:20	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
19	WKUPDEP_DSI1_C_EVE4	Wakeup dependency from DSS module (SWakeup signal) towards EVE4 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
18	WKUPDEP_DSI1_C_EVE3	Wakeup dependency from DSS module (SWakeup signal) towards EVE3 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
17	WKUPDEP_DSI1_C_EVE2	Wakeup dependency from DSS module (SWakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
16	WKUPDEP_DSI1_C_EVE1	Wakeup dependency from DSS module (SWakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
15	WKUPDEP_DSI1_C_DSP2	Wakeup dependency from DSS module (SWakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
14	WKUPDEP_DSI1_C_IPU1	Wakeup dependency from DSS module (SWakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
13	WKUPDEP_DSI1_C_SDMA	Wakeup dependency from DSS module (SWakeup signal) towards SDMA + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
12	WKUPDEP_DSI1_C_DSP1	Wakeup dependency from DSS module (SWakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
11	WKUPDEP_DSI1_C_IPU2	Wakeup dependency from DSS module (SWakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
10	WKUPDEP_DSI1_C_MPU	Wakeup dependency from DSS module (SWakeup signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
9	WKUPDEP_HDMIIRQ_EVE4	Wakeup dependency from DSS module (SWakeup signal) towards EVE4 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
8	WKUPDEP_HDMIIRQ_EVE3	Wakeup dependency from DSS module (SWakeup signal) towards EVE3 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
7	WKUPDEP_HDMIIRQ_EVE2	Wakeup dependency from DSS module (SWakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_HDMIIRQ_EVE1	Wakeup dependency from DSS module (SWakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_HDMIIRQ_DSP2	Wakeup dependency from DSS module (SWakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_HDMIIRQ_IPU1	Wakeup dependency from DSS module (SWakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	RESERVED		R	0x0
2	WKUPDEP_HDMIIRQ_DSP1	Wakeup dependency from DSS module (SWakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_HDMIIRQ_IPU2	Wakeup dependency from DSS module (SWakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_HDMIIRQ_MPU	Wakeup dependency from DSS module (SWakeup signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1340. RM_DSS_BB2D_CONTEXT

Address Offset	0x0000 0034		
Physical Address	0x4AE0 7134	**Instance**	DSS_PRM
Description	This register contains dedicated BB2B context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							LOSTMEM_DSS_MEM	RESERVED							LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	LOSTMEM_DSS_MEM	Specify if memory-based context in DSS_MEM memory bank has been lost due to a previous power transition or other reset source. 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
7:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of DSS_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1341. RM_DSS_SDVENC_CONTEXT

Address Offset	0x0000 003C		
Physical Address	0x4AE0 713C	**Instance**	DSS_PRM
Description	This register contains dedicated SDVENC context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of DSS_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

3.12.39 EMU_CM registers

3.12.39.1 EMU_CM Register Summary

Table 3-1342. EMU_CM Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	EMU_CM Physical Address
CM_EMU_CLKSTCTRL	RW	32	0x0000 0000	0x4AE0 7A00
CM_EMU_DEBUGSS_CLKCTRL	R	32	0x0000 0004	0x4AE0 7A04
CM_EMU_DYNAMICDEP	RW	32	0x0000 0008	0x4AE0 7A08
CM_EMU_MPU_EMU_DBG_CLKCTRL	R	32	0x0000 000C	0x4AE0 7A0C

3.12.39.2 EMU_CM Register Description

Table 3-1343. CM_EMU_CLKSTCTRL

Address Offset	0x0000 0000		
Physical Address	0x4AE0 7A00	**Instance**	EMU_CM
Description	This register enables the EMU domain power state transition. It controls the HW supervised domain power state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							CLKACTIVITY_EMU_SYS_CLK	RESERVED						CLKTRCTRL	

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	CLKACTIVITY_EMU_SYS_CLK	This field indicates the state of the EMU_SYS_CLK clock in the domain. 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
7:2	RESERVED		R	0x0
1:0	CLKTRCTRL	Controls the clock state transition of the EMU clock domain. 0x0: Reserved 0x1: Reserved 0x2: SW_WKUP: Start a software forced wake-up transition on the domain. 0x3: HW_AUTO: Automatic transition is enabled. Sleep and wakeup transition are based upon hardware conditions.	RW	0x2

Table 3-1344. Register Call Summary for Register CM_EMU_CLKSTCTRL

Clock Management Functional Description • Clock Domain Modes: [0][1]
PRCM Register Manual • EMU_CM Register Summary: [2]

Table 3-1345. CM_EMU_DEBUGSS_CLKCTRL

Address Offset	0x0000 0004		
Physical Address	0x4AE0 7A04	**Instance**	EMU_CM
Description	This register manages the DEBUGSS clocks. [warm reset insensitive]		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED													STBYST	IDLEST		RESERVED														MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:19	RESERVED		R	0x0
18	STBYST	Module standby status 0x0: Module is functional (not in standby) 0x1: Module is in standby	R	0x1
17:16	IDLEST	Module idle status 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state.	R	0x1

Table 3-1346. Register Call Summary for Register CM_EMU_DEBUGSS_CLKCTRL

PRCM Register Manual • EMU_CM Register Summary: [0]

Table 3-1347. CM_EMU_DYNAMICDEP

Address Offset	0x0000 0008		
Physical Address	0x4AE0 7A08	**Instance**	EMU_CM
Description	This register controls the dynamic domain depedencies from EMU domain towards 'target' domains. It is relevant only for domain having OCP master port(s).		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				WINDOWSIZE				RESERVED																		L3MAIN1_DYNDEP	RESERVED					

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:24	WINDOWSIZE	Size of sliding window used to monitor OCP interface activity for determination of auto-sleep feature. Time unit defined by CM_DYN_DEP_PRESCAL register.	RW	0x4
23:6	RESERVED		R	0x0
5	L3MAIN1_DYNDEP	Dynamic dependency towards L3MAIN1 clock domain 0x1: Dependency is enabled	R	0x1
4:0	RESERVED		R	0x0

Table 3-1348. Register Call Summary for Register CM_EMU_DYNAMICDEP

Clock Management Functional Description • Clock Domain Dependency: [0]
PRCM Register Manual • EMU_CM Register Summary: [1]

Table 3-1349. CM_EMU_MPU_EMU_DBG_CLKCTRL

Address Offset	0x0000 000C		
Physical Address	0x4AE0 7A0C	**Instance**	EMU_CM
Description	This register manages the MPU_EMU_DBG clocks. [warm reset insensitive]		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														IDLEST		RESERVED														MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state.	R	0x1

Table 3-1350. Register Call Summary for Register CM_EMU_MPU_EMU_DBG_CLKCTRL

PRCM Register Manual
- EMU_CM Register Summary: [0]

3.12.40 EMU_PRM Registers

3.12.40.1 EMU_PRM Register Summary

Table 3-1351. EMU_PRM Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	EMU_PRM Physical Address
PM_EMU_PWRSTCTRL	R	32	0x0000 0000	0x4AE0 7900
PM_EMU_PWRSTST	RW	32	0x0000 0004	0x4AE0 7904
RM_EMU_DEBUGSS_CONTEXT	RW	32	0x0000 0024	0x4AE0 7924

3.12.40.2 EMU_PRM Register Description

Table 3-1352. PM_EMU_PWRSTCTRL

Address Offset	0x0000 0000		
Physical Address	0x4AE0 7900	**Instance**	EMU_PRM
Description	This register controls the EMU power state to reach upon a domain sleep transition		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														EMU_BANK_ONSTATE		RESERVED														POWERSTATE	

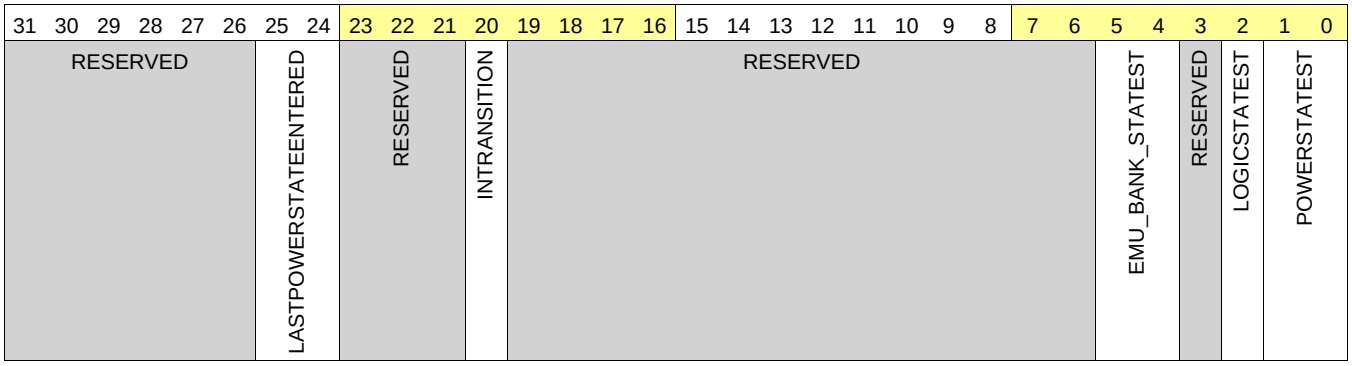
Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	EMU_BANK_ONSTATE	EMU memory state when domain is ON. 0x3: Memory bank is on when the domain is ON.	R	0x3
15:2	RESERVED		R	0x0
1:0	POWERSTATE	Power state control 0x0: OFF state	R	0x0

Table 3-1353. Register Call Summary for Register PM_EMU_PWRSTCTRL

Power Management Functional Description
PRCM Register Manual • EMU_PRM Register Summary: [2]

Table 3-1354. PM_EMU_PWRSTST

Address Offset	0x0000 0004		
Physical Address	0x4AE0 7904	**Instance**	EMU_PRM
Description	This register provides a status on the EMU domain current power state. [warm reset insensitive]		
Type	RW		



Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:24	LASTPOWERSTATEENTERED	Last low power state entered. Set to 0x3 upon write of the same only. This register is intended for debug purpose only. 0x0: Power domain was previously OFF 0x1: Power domain was previously in RETENTION 0x2: Power domain was previously ON-INACTIVE 0x3: Power domain was previously ON-ACTIVE	RW	0x0
23:21	RESERVED		R	0x0
20	INTRANSITION	Domain transition status 0x0: No on-going transition on power domain 0x1: Power domain transition is in progress.	R	0x0
19:6	RESERVED		R	0x0
5:4	EMU_BANK_STATEST	EMU memory bank state status 0x0: Memory is OFF 0x1: Reserved 0x2: Reserved 0x3: Memory is ON It is supplied by WKUP LDO	R	0x3
3	RESERVED		R	0x0
2	LOGICSTATEST	Logic state status 0x0: Logic in domain is OFF 0x1: Logic in domain is ON	R	0x1
1:0	POWERSTATEST	Current power state status 0x0: Power domain is OFF 0x3: Power domain is ON-ACTIVE	R	0x3

Table 3-1355. Register Call Summary for Register PM_EMU_PWRSTST

Power Management Functional Description
PRCM Register Manual • EMU_PRM Register Summary: [5]

Table 3-1356. RM_EMU_DEBUGSS_CONTEXT

Address Offset	0x0000 0024		
Physical Address	0x4AE0 7924	**Instance**	EMU_PRM
Description	This register contains dedicated DEBUGSS context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							LOSTMEM_EMU_BANK	RESERVED							LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	LOSTMEM_EMU_BANK	Specify if memory-based context in EMU_BANK memory bank has been lost due to a previous power transition or other reset source. 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
7:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of EMU_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1357. Register Call Summary for Register RM_EMU_DEBUGSS_CONTEXT

Power Management Functional Description
PRCM Register Manual • EMU_PRM Register Summary: [1]

3.12.41 EVE1_PRM registers

3.12.41.1 EVE1_PRM Register Summary

Table 3-1358. EVE1_PRM Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	EVE1_PRM Physical Address
PM_EVE1_PWRSTCTRL	RW	32	0x0000 0000	0x4AE0 7B40
PM_EVE1_PWRSTST	RW	32	0x0000 0004	0x4AE0 7B44
RM_EVE1_RSTCTRL	RW	32	0x0000 0010	0x4AE0 7B50

Table 3-1358. EVE1_PRM Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	EVE1_PRM Physical Address
RM_EVE1_RSTST	RW	32	0x0000 0014	0x4AE0 7B54
PM_EVE1_EVE1_WKDEP	RW	32	0x0000 0020	0x4AE0 7B60
RM_EVE1_EVE1_CONTEXT	RW	32	0x0000 0024	0x4AE0 7B64

3.12.41.2 EVE1_PRM Register Description

Table 3-1359. PM_EVE1_PWRSTCTRL

Address Offset	0x0000 0000		
Physical Address	0x4AE0 7B40	**Instance**	EVE1_PRM
Description	This register controls the EVE power state to reach upon a domain sleep transition		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														EVE1_BANK_ONSTATE		RESERVED								LOWPOWERSTATECHANGE	RESERVED			POWERSTATE			

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	EVE1_BANK_ONSTATE	EVE1 state when domain is ON. 0x3: Memory bank is on when the domain is ON.	R	0x3
15:5	RESERVED		R	0x0
4	LOWPOWERSTATECHANGE	Power state change request when domain has already performed a sleep transition. Allows going into deeper low power state without waking up the power domain. 0x0: Do not request a low power state change. 0x1: Request a low power state change. This bit is automatically cleared when the power state is effectively changed or when power state is ON.	RW	0x0
3:2	RESERVED		R	0x0
1:0	POWERSTATE	Power state control 0x0: OFF state 0x1: Reserved 0x2: INACTIVE state 0x3: ON State	RW	0x3

Table 3-1360. Register Call Summary for Register PM_EVE1_PWRSTCTRL

Power Management Functional Description • Logic and Memory Area Power Modes Control and Status: [0][1][2]
PRCM Register Manual • Not Supported Functionality (Registers and Bitfields): [3] • EVE1_PRM Register Summary: [4]

Table 3-1361. PM_EVE1_PWRSTST

Address Offset	0x0000 0004		
Physical Address	0x4AE0 7B44	**Instance**	EVE1_PRM
Description	This register provides a status on the EVE domain current power state. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED						LASTPOWERSTATEENTERED		RESERVED			INTRANSITION	RESERVED												EVE1_BANK_STATEST		RESERVED		LOGICSTATEST		POWERSTATEST	

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:24	LASTPOWERSTATEENTERED	Last low power state entered. Set to 0x3 upon write of the same only. This register is intended for debug purpose only. 0x0: Power domain was previously OFF 0x1: Power domain was previously in RETENTION 0x2: Power domain was previously ON-INACTIVE 0x3: Power domain was previously ON-ACTIVE	RW	0x0
23:21	RESERVED		R	0x0
20	INTRANSITION	Domain transition status 0x0: No on-going transition on power domain 0x1: Power domain transition is in progress.	R	0x0
19:6	RESERVED		R	0x0
5:4	EVE1_BANK_STATEST	EVE0 memory state status 0x0: Memory is OFF 0x1: RESERVED 0x2: Reserved 0x3: Memory is ON	R	0x3
3	RESERVED		R	0x0
2	LOGICSTATEST	Logic state status 0x0: Logic in domain is OFF 0x1: Logic in domain is ON	R	0x1

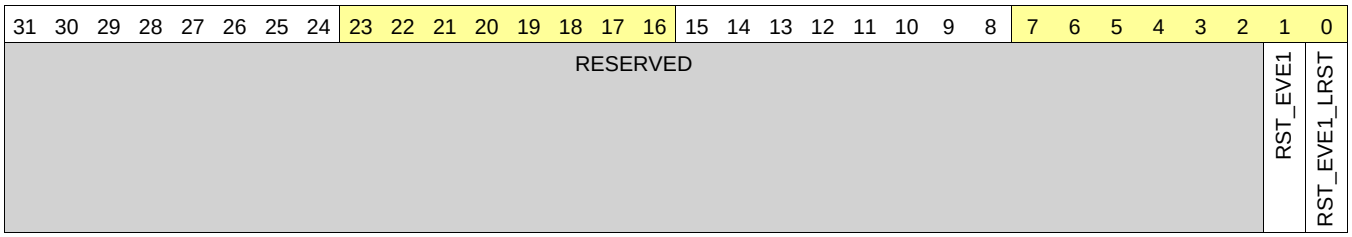
Bits	Field Name	Description	Type	Reset
1:0	POWERSTATEST	Current power state status 0x0: Power domain is OFF 0x1: Power domain is in RETENTION 0x2: Power domain is ON-INACTIVE 0x3: Power domain is ON-ACTIVE	R	0x3

Table 3-1362. Register Call Summary for Register PM_EVE1_PWRSTST

Power Management Functional Description • Logic and Memory Area Power Modes Control and Status: [0][1][2][3][4]
PRCM Register Manual • Not Supported Functionality (Registers and Bitfields): [5][6][7][8][9] • EVE1_PRM Register Summary: [10]

Table 3-1363. RM_EVE1_RSTCTRL

Address Offset	0x0000 0010		
Physical Address	0x4AE0 7B50	**Instance**	EVE1_PRM
Description	This register controls the release of the EVE sub-system resets.		
Type	RW		



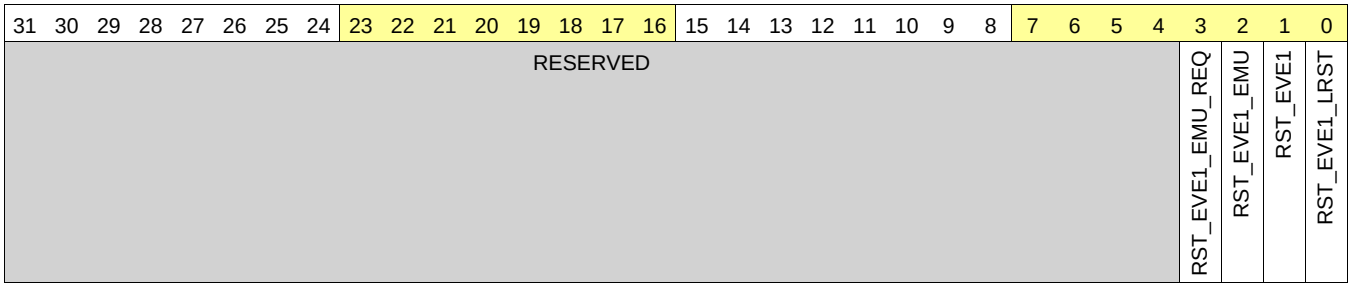
Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1	RST_EVE1	EVE reset control 0x0: Reset is cleared 0x1: Local Reset is asserted	RW	0x1
0	RST_EVE1_LRST	EVE Local reset control 0x0: Reset is cleared 0x1: Reset is asserted	RW	0x1

Table 3-1364. Register Call Summary for Register RM_EVE1_RSTCTRL

Reset Management Functional Description • Reset Domains: [0][1] • EVE Subsystem Power-On Reset Sequence: [2][3] • EVE Subsystem Software Warm Reset Sequence: [4][5][6]
PRCM Register Manual • EVE1_PRM Register Summary: [7]

Table 3-1365. RM_EVE1_RSTST

Address Offset	0x0000 0014		
Physical Address	0x4AE0 7B54	**Instance**	EVE1_PRM
Description	This register logs the different reset sources of the EVE domain. Each bit is set upon release of the domain reset signal. Must be cleared by software. [warm reset insensitive]		
Type	RW		



Bits	Field Name	Description	Type	Reset
31:4	RESERVED		R	0x0
3	RST_EVE1_EMU_REQ	EVE1 processor has been reset due to EVE emulation reset request driven from EVE1-SS 0x0: No emulation reset 0x1: EVE has been reset upon emulation reset request	RW	0x0
2	RST_EVE1_EMU	EVE1 domain has been reset due to emulation reset source e.g. assert reset command initiated by the icepick module 0x0: No emulation reset 0x1: EVE has been reset upon emulation reset	RW	0x0
1	RST_EVE1	EVE0 SW reset status 0x0: No SW reset occurred 0x1: Local reset upon SW reset	RW	0x0
0	RST_EVE1_LRST	EVE0 Local SW reset 0x0: No SW reset occurred 0x1: EVE has been reset upon SW reset	RW	0x0

Table 3-1366. Register Call Summary for Register RM_EVE1_RSTST

PRCM Register Manual
- EVE1_PRM Register Summary: [0]

Table 3-1367. PM_EVE1_EVE1_WKDEP

Address Offset	0x0000 0020		
Physical Address	0x4AE0 7B60	**Instance**	EVE1_PRM
Description	This register controls wakeup dependency based on EVE1 service requests.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						WKUPDEP_EVE1_EVE4	WKUPDEP_EVE1_EVE3	WKUPDEP_EVE1_EVE2	RESERVED	WKUPDEP_EVE1_DSP2	WKUPDEP_EVE1_IPU1	WKUPDEP_EVE1_SDMA	WKUPDEP_EVE1_DSP1	WKUPDEP_EVE1_IPU2	WKUPDEP_EVE1_MPU

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	WKUPDEP_EVE1_EVE4	Wakeup dependency from EVE1 module (SWakeup signal) towards EVE4 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
8	WKUPDEP_EVE1_EVE3	Wakeup dependency from EVE1 module (SWakeup signal) towards EVE3 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
7	WKUPDEP_EVE1_EVE2	Wakeup dependency from EVE1 module (SWakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	RESERVED		R	0x0
5	WKUPDEP_EVE1_DSP2	Wakeup dependency from EVE1 module (SWakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_EVE1_IPU1	Wakeup dependency from EVE1 module (SWakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	WKUPDEP_EVE1_SDMA	Wakeup dependency from EVE1 module (SWakeup signal) towards SDMA + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
2	WKUPDEP_EVE1_DSP1	Wakeup dependency from EVE1 module (SWakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_EVE1_IPU2	Wakeup dependency from EVE1 module (SWakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
0	WKUPDEP_EVE1_MPU	Wakeup dependency from EVE1 module (SWakeup signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1368. Register Call Summary for Register PM_EVE1_EVE1_WKDEP

Clock Management Functional Description • Clock Domain Dependency: [0][1][2]
PRCM Register Manual • Not Supported Functionality (Registers and Bitfields): [3][4][5][6][7][8] • EVE1_PRM Register Summary: [9]

Table 3-1369. RM_EVE1_EVE1_CONTEXT

Address Offset	0x0000 0024		
Physical Address	0x4AE0 7B64	**Instance**	EVE1_PRM
Description	This register contains dedicated EVE context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							LOSTMEM_EVE_BANK	RESERVED							LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	LOSTMEM_EVE_BANK	Specify if memory-based context in EVE1 memory bank has been lost due to a previous power transition or other reset source. 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
7:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of EVE0_SYS_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1370. Register Call Summary for Register RM_EVE1_EVE1_CONTEXT

Power Management Functional Description • PD_EVE1 Description: [0]
PRCM Register Manual • EVE1_PRM Register Summary: [1]

3.12.42 EVE2_PRM registers

3.12.42.1 EVE2_PRM Register Summary

Table 3-1371. EVE2_PRM Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	EVE2_PRM Physical Address
PM_EVE2_PWRSTCTRL	RW	32	0x0000 0000	0x4AE0 7B80
PM_EVE2_PWRSTST	RW	32	0x0000 0004	0x4AE0 7B84
RM_EVE2_RSTCTRL	RW	32	0x0000 0010	0x4AE0 7B90
RM_EVE2_RSTST	RW	32	0x0000 0014	0x4AE0 7B94
PM_EVE2_EVE2_WKDEP	RW	32	0x0000 0020	0x4AE0 7BA0
RM_EVE2_EVE2_CONTEXT	RW	32	0x0000 0024	0x4AE0 7BA4

3.12.42.2 EVE2_PRM Register Description

Table 3-1372. PM_EVE2_PWRSTCTRL

Address Offset	0x0000 0000		
Physical Address	0x4AE0 7B80	**Instance**	EVE2_PRM
Description	This register controls the EVE power state to reach upon a domain sleep transition		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														EVE2_BANK_ONSTATE		RESERVED											LOWPOWERSTATECHANGE	RESERVED		POWERSTATE	

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	EVE2_BANK_ONSTATE	EVE2 state when domain is ON. 0x3: Memory bank is on when the domain is ON.	R	0x3
15:5	RESERVED		R	0x0
4	LOWPOWERSTATECHANGE	Power state change request when domain has already performed a sleep transition. Allows going into deeper low power state without waking up the power domain. 0x0: Do not request a low power state change. 0x1: Request a low power state change. This bit is automatically cleared when the power state is effectively changed or when power state is ON.	RW	0x0
3:2	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
1:0	POWERSTATE	Power state control 0x0: OFF state 0x1: RESERVED 0x2: INACTIVE state 0x3: ON State	RW	0x3

Table 3-1373. PM_EVE2_PWRSTST

Address Offset	0x0000 0004		
Physical Address	0x4AE0 7B84	**Instance**	EVE2_PRM
Description	This register provides a status on the EVE domain current power state. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED						LASTPOWERSTATEENTERED		RESERVED			INTRANSITION	RESERVED												EVE2_BANK_STATEST		RESERVED		LOGICSTATEST		POWERSTATEST	

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:24	LASTPOWERSTATEENTERED	Last low power state entered. Set to 0x3 upon write of the same only. This register is intended for debug purpose only. 0x0: Power domain was previously OFF 0x1: Power domain was previously in RETENTION 0x2: Power domain was previously ON-INACTIVE 0x3: Power domain was previously ON-ACTIVE	RW	0x0
23:21	RESERVED		R	0x0
20	INTRANSITION	Domain transition status 0x0: No on-going transition on power domain 0x1: Power domain transition is in progress.	R	0x0
19:6	RESERVED		R	0x0
5:4	EVE2_BANK_STATEST	EVE2 memory state status 0x0: Memory is OFF 0x1: RESERVED 0x2: Reserved 0x3: Memory is ON	R	0x3
3	RESERVED		R	0x0
2	LOGICSTATEST	Logic state status 0x0: Logic in domain is OFF 0x1: Logic in domain is ON	R	0x1

Bits	Field Name	Description	Type	Reset
1:0	POWERSTATEST	Current power state status 0x0: Power domain is OFF 0x1: Power domain is in RETENTION 0x2: Power domain is ON-INACTIVE 0x3: Power domain is ON-ACTIVE	R	0x3

Table 3-1374. RM_EVE2_RSTCTRL

Address Offset	0x0000 0010		
Physical Address	0x4AE0 7B90	**Instance**	EVE2_PRM
Description	This register controls the release of the EVE sub-system resets.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														RST_EVE2	RST_EVE2_LRST

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1	RST_EVE2	EVE SW reset control 0x0: Reset is cleared 0x1: Local Reset is asserted	RW	0x1
0	RST_EVE2_LRST	EVE Local reset control 0x0: Reset is cleared for the DSP - DSP 0x1: Reset is asserted for the DSP - DSP	RW	0x1

Table 3-1375. RM_EVE2_RSTST

Address Offset	0x0000 0014		
Physical Address	0x4AE0 7B94	**Instance**	EVE2_PRM
Description	This register logs the different reset sources of the EVE domain. Each bit is set upon release of the domain reset signal. Must be cleared by software. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												RST_EVE2_EMU_REQ	RST_EVE2_EMU	RST_EVE2	RST_EVE2_LRST

Bits	Field Name	Description	Type	Reset
31:4	RESERVED		R	0x0
3	RST_EVE2_EMU_REQ	EVE2 processor has been reset due to EVE emulation reset request driven from EVE2-SS 0x0: No emulation reset 0x1: EVE has been reset upon emulation reset request	RW	0x0
2	RST_EVE2_EMU	EVE2 domain has been reset due to emulation reset source e.g. assert reset command initiated by the icepick module 0x0: No emulation reset 0x1: EVE has been reset upon emulation reset	RW	0x0
1	RST_EVE2	EVE SW reset status 0x0: No SW reset occurred 0x1: Local reset upon SW reset	RW	0x0
0	RST_EVE2_LRST	EVE Local SW reset 0x0: No SW reset occurred 0x1: EVE has been reset upon SW reset	RW	0x0

Table 3-1376. PM_EVE2_EVE2_WKDEP

Address Offset	0x0000 0020		
Physical Address	0x4AE0 7BA0	**Instance**	EVE2_PRM
Description	This register controls wakeup dependency based on EVE2 service requests.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						WKUPDEP_EVE2_EVE4	WKUPDEP_EVE2_EVE3	RESERVED	WKUPDEP_EVE2_EVE1	WKUPDEP_EVE2_DSP2	WKUPDEP_EVE2_IPU1	WKUPDEP_EVE2_SDMA	WKUPDEP_EVE2_DSP1	WKUPDEP_EVE2_IPU2	WKUPDEP_EVE2_MPU

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	WKUPDEP_EVE2_EVE4	Wakeup dependency from EVE2 module (SWakeup signal) towards EVE4 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
8	WKUPDEP_EVE2_EVE3	Wakeup dependency from EVE2 module (SWakeup signal) towards EVE3 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
7	RESERVED		R	0x0
6	WKUPDEP_EVE2_EVE1	Wakeup dependency from EVE2 module (SWakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
5	WKUPDEP_EVE2_DSP2	Wakeup dependency from EVE2 module (SWakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_EVE2_IPU1	Wakeup dependency from EVE2 module (SWakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	WKUPDEP_EVE2_SDMA	Wakeup dependency from EVE2 module (SWakeup signal) towards SDMA + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
2	WKUPDEP_EVE2_DSP1	Wakeup dependency from EVE2 module (SWakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_EVE2_IPU2	Wakeup dependency from EVE2 module (SWakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_EVE2_MPU	Wakeup dependency from EVE2 module (SWakeup signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1377. RM_EVE2_EVE2_CONTEXT

Address Offset	0x0000 0024		
Physical Address	0x4AE0 7BA4	**Instance**	EVE2_PRM
Description	This register contains dedicated EVE context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							LOSTMEM_EVE_BANK	RESERVED							LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	LOSTMEM_EVE_BANK	Specify if memory-based context in EVE2 memory bank has been lost due to a previous power transition or other reset source. 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Bits	Field Name	Description	Type	Reset
7:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of EVE1_SYS_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

3.12.43 EVE3_PRM registers

3.12.43.1 EVE3_PRM Register Summary

Table 3-1378. EVE3_PRM Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	EVE3_PRM Physical Address
PM_EVE3_PWRSTCTRL	RW	32	0x0000 0000	0x4AE0 7BC0
PM_EVE3_PWRSTST	RW	32	0x0000 0004	0x4AE0 7BC4
RM_EVE3_RSTCTRL	RW	32	0x0000 0010	0x4AE0 7BD0
RM_EVE3_RSTST	RW	32	0x0000 0014	0x4AE0 7BD4
PM_EVE3_EVE3_WKDEP	RW	32	0x0000 0020	0x4AE0 7BE0
RM_EVE3_EVE3_CONTEXT	RW	32	0x0000 0024	0x4AE0 7BE4

3.12.43.2 EVE3_PRM Register Description

Table 3-1379. PM_EVE3_PWRSTCTRL

Address Offset	0x0000 0000		
Physical Address	0x4AE0 7BC0	**Instance**	EVE3_PRM
Description	This register controls the EVE power state to reach upon a domain sleep transition		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														EVE3_BANK_ONSTATE		RESERVED											LOWPOWERSTATECHANGE	RESERVED			POWERSTATE

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	EVE3_BANK_ONSTATE	EVE3 state when domain is ON. 0x3: Memory bank is on when the domain is ON.	R	0x3
15:5	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
4	LOWPOWERSTATECHANGE	Power state change request when domain has already performed a sleep transition. Allows going into deeper low power state without waking up the power domain. 0x0: Do not request a low power state change. 0x1: Request a low power state change. This bit is automatically cleared when the power state is effectively changed or when power state is ON.	RW	0x0
3:2	RESERVED		R	0x0
1:0	POWERSTATE	Power state control 0x0: OFF state 0x1: RESERVED 0x2: INACTIVE state 0x3: ON State	RW	0x3

Table 3-1380. PM_EVE3_PWRSTST

Address Offset	0x0000 0004		
Physical Address	0x4AE0 7BC4	**Instance**	EVE3_PRM
Description	This register provides a status on the EVE domain current power state. [warm reset insensitive]		
Type	RW		

31 30 29 28 27 26	25 24	23 22 21	20	19 18 17 16	15 14 13 12 11 10 9 8	7 6	5 4	3	2	1 0
RESERVED	LASTPOWERSTATEENTERED	RESERVED	INTRANSITION	RESERVED			EVE3_BANK_STATEST	RESERVED	LOGICSTATEST	POWERSTATEST

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:24	LASTPOWERSTATEENTERED	Last low power state entered. Set to 0x3 upon write of the same only. This register is intended for debug purpose only. 0x0: Power domain was previously OFF 0x1: Power domain was previously in RETENTION 0x2: Power domain was previously ON-INACTIVE 0x3: Power domain was previously ON-ACTIVE	RW	0x0
23:21	RESERVED		R	0x0
20	INTRANSITION	Domain transition status 0x0: No on-going transition on power domain 0x1: Power domain transition is in progress.	R	0x0
19:6	RESERVED		R	0x0
5:4	EVE3_BANK_STATEST	EVE3 memory state status 0x0: Memory is OFF 0x1: RESERVED 0x2: Reserved 0x3: Memory is ON	R	0x3

Bits	Field Name	Description	Type	Reset
3	RESERVED		R	0x0
2	LOGICSTATEST	Logic state status 0x0: Logic in domain is OFF 0x1: Logic in domain is ON	R	0x1
1:0	POWERSTATEST	Current power state status 0x0: Power domain is OFF 0x1: Power domain is in RETENTION 0x2: Power domain is ON-INACTIVE 0x3: Power domain is ON-ACTIVE	R	0x3

Table 3-1381. RM_EVE3_RSTCTRL

Address Offset	0x0000 0010		
Physical Address	0x4AE0 7BD0	**Instance**	EVE3_PRM
Description	This register controls the release of the EVE sub-system resets.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														RST_EVE3	RST_EVE3_LRST

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1	RST_EVE3	EVE SW reset control 0x0: Reset is cleared 0x1: Local Reset is asserted	RW	0x1
0	RST_EVE3_LRST	EVE3 Local reset control 0x0: Reset is cleared for the DSP - DSP 0x1: Reset is asserted for the DSP - DSP	RW	0x1

Table 3-1382. RM_EVE3_RSTST

Address Offset	0x0000 0014		
Physical Address	0x4AE0 7BD4	**Instance**	EVE3_PRM
Description	This register logs the different reset sources of the EVE domain. Each bit is set upon release of the domain reset signal. Must be cleared by software. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												RST_EVE3_EMU_REQ	RST_EVE3_EMU	RST_EVE3	RST_EVE3_LRST

Bits	Field Name	Description	Type	Reset
31:4	RESERVED		R	0x0
3	RST_EVE3_EMU_REQ	EVE3 processor has been reset due to EVE emulation reset request driven from EVE3-SS 0x0: No emulation reset 0x1: EVE has been reset upon emulation reset request	RW	0x0
2	RST_EVE3_EMU	EVE3 domain has been reset due to emulation reset source e.g. assert reset command initiated by the icepick module 0x0: No emulation reset 0x1: EVE has been reset upon emulation reset	RW	0x0
1	RST_EVE3	EVE3 SW reset status 0x0: No SW reset occurred 0x1: Local reset upon SW reset	RW	0x0
0	RST_EVE3_LRST	EVE3 Local SW reset 0x0: No SW reset occurred 0x1: EVE has been reset upon SW reset	RW	0x0

Table 3-1383. PM_EVE3_EVE3_WKDEP

Address Offset	0x0000 0020		
Physical Address	0x4AE0 7BE0	**Instance**	EVE3_PRM
Description	This register controls wakeup dependency based on EVE3 service requests.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						WKUPDEP_EVE3_EVE4	RESERVED	WKUPDEP_EVE3_EVE2	WKUPDEP_EVE3_EVE1	WKUPDEP_EVE3_DSP2	WKUPDEP_EVE3_IPU1	WKUPDEP_EVE3_SDMA	WKUPDEP_EVE3_DSP1	WKUPDEP_EVE3_IPU2	WKUPDEP_EVE3_MPU

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	WKUPDEP_EVE3_EVE4	Wakeup dependency from EVE3 module (SWakeup signal) towards EVE4 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
8	RESERVED		R	0x0
7	WKUPDEP_EVE3_EVE2	Wakeup dependency from EVE3 module (SWakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_EVE3_EVE1	Wakeup dependency from EVE3 module (SWakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
5	WKUPDEP_EVE3_DSP2	Wakeup dependency from EVE3 module (SWakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_EVE3_IPU1	Wakeup dependency from EVE3 module (SWakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	WKUPDEP_EVE3_SDMA	Wakeup dependency from EVE3 module (SWakeup signal) towards SDMA + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
2	WKUPDEP_EVE3_DSP1	Wakeup dependency from EVE3 module (SWakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_EVE3_IPU2	Wakeup dependency from EVE3 module (SWakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_EVE3_MPU	Wakeup dependency from EVE3 module (SWakeup signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1384. RM_EVE3_EVE3_CONTEXT

Address Offset	0x0000 0024		
Physical Address	0x4AE0 7BE4	**Instance**	EVE3_PRM
Description	This register contains dedicated EVE context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							LOSTMEM_EVE_BANK	RESERVED							LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	LOSTMEM_EVE_BANK	Specify if memory-based context in EVE3 memory bank has been lost due to a previous power transition or other reset source. 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Bits	Field Name	Description	Type	Reset
7:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of EVE1_SYS_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

3.12.44 EVE4_PRM registers

3.12.44.1 EVE4_PRM Register Summary

Table 3-1385. EVE4_PRM Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	EVE4_PRM Physical Address
PM_EVE4_PWRSTCTRL	RW	32	0x0000 0000	0x4AE0 7C00
PM_EVE4_PWRSTST	RW	32	0x0000 0004	0x4AE0 7C04
RM_EVE4_RSTCTRL	RW	32	0x0000 0010	0x4AE0 7C10
RM_EVE4_RSTST	RW	32	0x0000 0014	0x4AE0 7C14
PM_EVE4_EVE4_WKDEP	RW	32	0x0000 0020	0x4AE0 7C20
RM_EVE4_EVE4_CONTEXT	RW	32	0x0000 0024	0x4AE0 7C24

3.12.44.2 EVE4_PRM Register Description

Table 3-1386. PM_EVE4_PWRSTCTRL

Address Offset	0x0000 0000		
Physical Address	0x4AE0 7C00	**Instance**	EVE4_PRM
Description	This register controls the EVE power state to reach upon a domain sleep transition		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														EVE4_BANK_ONSTATE		RESERVED											LOWPOWERSTATECHANGE	RESERVED		POWERSTATE	

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	EVE4_BANK_ONSTATE	EVE4 state when domain is ON. 0x3: Memory bank is on when the domain is ON.	R	0x3
15:5	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
4	LOWPOWERSTATECHANGE	Power state change request when domain has already performed a sleep transition. Allows going into deeper low power state without waking up the power domain. 0x0: Do not request a low power state change. 0x1: Request a low power state change. This bit is automatically cleared when the power state is effectively changed or when power state is ON.	RW	0x0
3:2	RESERVED		R	0x0
1:0	POWERSTATE	Power state control 0x0: OFF state 0x1: RESERVED 0x2: INACTIVE state 0x3: ON State	RW	0x3

Table 3-1387. PM_EVE4_PWRSTST

Address Offset	0x0000 0004		
Physical Address	0x4AE0 7C04	**Instance**	EVE4_PRM
Description	This register provides a status on the EVE domain current power state. [warm reset insensitive]		
Type	RW		

31 30 29 28 27 26	25 24	23 22 21	20	19 18 17 16	15 14 13 12 11 10 9 8	7 6	5 4	3	2	1 0
RESERVED	LASTPOWERSTATEENTERED	RESERVED	INTRANSITION	RESERVED			EVE4_BANK_STATEST	RESERVED	LOGICSTATEST	POWERSTATEST

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:24	LASTPOWERSTATEENTERED	Last low power state entered. Set to 0x3 upon write of the same only. This register is intended for debug purpose only. 0x0: Power domain was previously OFF 0x1: Power domain was previously in RETENTION 0x2: Power domain was previously ON-INACTIVE 0x3: Power domain was previously ON-ACTIVE	RW	0x0
23:21	RESERVED		R	0x0
20	INTRANSITION	Domain transition status 0x0: No on-going transition on power domain 0x1: Power domain transition is in progress.	R	0x0
19:6	RESERVED		R	0x0
5:4	EVE4_BANK_STATEST	EVE4 memory state status 0x0: Memory is OFF 0x1: Reserved 0x2: Reserved 0x3: Memory is ON	R	0x3

Bits	Field Name	Description	Type	Reset
3	RESERVED		R	0x0
2	LOGICSTATEST	Logic state status 0x0: Logic in domain is OFF 0x1: Logic in domain is ON	R	0x1
1:0	POWERSTATEST	Current power state status 0x0: Power domain is OFF 0x1: Power domain is in RETENTION 0x2: Power domain is ON-INACTIVE 0x3: Power domain is ON-ACTIVE	R	0x3

Table 3-1388. RM_EVE4_RSTCTRL

Address Offset	0x0000 0010		
Physical Address	0x4AE0 7C10	**Instance**	EVE4_PRM
Description	This register controls the release of the EVE sub-system resets.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														RST_EVE4	RST_EVE4_LRST

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1	RST_EVE4	EVE SW reset control 0x0: Reset is cleared 0x1: Local Reset is asserted	RW	0x1
0	RST_EVE4_LRST	EVE4 Local reset control 0x0: Reset is cleared for the DSP - DSP 0x1: Reset is asserted for the DSP - DSP	RW	0x1

Table 3-1389. RM_EVE4_RSTST

Address Offset	0x0000 0014		
Physical Address	0x4AE0 7C14	**Instance**	EVE4_PRM
Description	This register logs the different reset sources of the EVE domain. Each bit is set upon release of the domain reset signal. Must be cleared by software. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												RST_EVE4_EMU_REQ	RST_EVE4_EMU	RST_EVE4	RST_EVE4_LRST

Bits	Field Name	Description	Type	Reset
31:4	RESERVED		R	0x0
3	RST_EVE4_EMU_REQ	EVE4 processor has been reset due to EVE emulation reset request driven from EVE4-SS 0x0: No emulation reset 0x1: EVE has been reset upon emulation reset request	RW	0x0
2	RST_EVE4_EMU	EVE4 domain has been reset due to emulation reset source e.g. assert reset command initiated by the icepick module 0x0: No emulation reset 0x1: EVE has been reset upon emulation reset	RW	0x0
1	RST_EVE4	EVE4 SW reset status 0x0: No SW reset occurred 0x1: Local reset upon SW reset	RW	0x0
0	RST_EVE4_LRST	EVE4 Local SW reset 0x0: No SW reset occurred 0x1: EVE has been reset upon SW reset	RW	0x0

Table 3-1390. PM_EVE4_EVE4_WKDEP

Address Offset	0x0000 0020		
Physical Address	0x4AE0 7C20	**Instance**	EVE4_PRM
Description	This register controls wakeup dependency based on EVE4 service requests.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							WKUPDEP_EVE4_EVE3	WKUPDEP_EVE4_EVE2	WKUPDEP_EVE4_EVE1	WKUPDEP_EVE4_DSP2	WKUPDEP_EVE4_IPU1	WKUPDEP_EVE4_SDMA	WKUPDEP_EVE4_DSP1	WKUPDEP_EVE4_IPU2	WKUPDEP_EVE4_MPU

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	WKUPDEP_EVE4_EVE3	Wakeup dependency from EVE4 module (SWakeup signal) towards EVE3 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
7	WKUPDEP_EVE4_EVE2	Wakeup dependency from EVE4 module (SWakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_EVE4_EVE1	Wakeup dependency from EVE4 module (SWakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
5	WKUPDEP_EVE4_DSP2	Wakeup dependency from EVE4 module (SWakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_EVE4_IPU1	Wakeup dependency from EVE4 module (SWakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	WKUPDEP_EVE4_SDMA	Wakeup dependency from EVE4 module (SWakeup signal) towards SDMA + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
2	WKUPDEP_EVE4_DSP1	Wakeup dependency from EVE4 module (SWakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_EVE4_IPU2	Wakeup dependency from EVE4 module (SWakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_EVE4_MPU	Wakeup dependency from EVE4 module (SWakeup signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1391. RM_EVE4_EVE4_CONTEXT

Address Offset	0x0000 0024		
Physical Address	0x4AE0 7C24	**Instance**	EVE4_PRM
Description	This register contains dedicated EVE context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							LOSTMEM_EVE_BANK	RESERVED							LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	LOSTMEM_EVE_BANK	Specify if memory-based context in EVE4 memory bank has been lost due to a previous power transition or other reset source. 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Bits	Field Name	Description	Type	Reset
7:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of EVE1_SYS_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

3.12.45 GPU_PRM registers

3.12.45.1 GPU_PRM Register Summary

Table 3-1392. GPU_PRM Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	GPU_PRM Physical Address
PM_GPU_PWRSTCTRL	RW	32	0x0000 0000	0x4AE0 7200
PM_GPU_PWRSTST	RW	32	0x0000 0004	0x4AE0 7204
RM_GPU_GPU_CONTEXT	RW	32	0x0000 0024	0x4AE0 7224

3.12.45.2 GPU_PRM Register Description

Table 3-1393. PM_GPU_PWRSTCTRL

Address Offset	0x0000 0000		
Physical Address	0x4AE0 7200	**Instance**	GPU_PRM
Description	This register controls the GPU power state to reach upon a domain sleep transition		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														GPU_MEM_ONSTATE		RESERVED											LOWPOWERSTATECHANGE	RESERVED		POWERSTATE	

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	GPU_MEM_ONSTATE	GPU_MEM memory bank state when domain is ON. 0x3: Memory bank is on when the domain is ON.	R	0x3
15:5	RESERVED		R	0x0
4	LOWPOWERSTATECHANGE	Power state change request when domain has already performed a sleep transition. Allows going into deeper low power state without waking up the power domain. 0x0: Do not request a low power state change. 0x1: Request a low power state change. This bit is automatically cleared when the power state is effectively changed or when power state is ON.	RW	0x0

Bits	Field Name	Description	Type	Reset
3:2	RESERVED		R	0x0
1:0	POWERSTATE	Power state control 0x0: OFF state 0x1: Reserved 0x2: INACTIVE state 0x3: ON State	RW	0x0

Table 3-1394. PM_GPU_PWRSTST

Address Offset	0x0000 0004		
Physical Address	0x4AE0 7204	**Instance**	GPU_PRM
Description	This register provides a status on the current GPU power domain state. [warm reset insensitive]		
Type	RW		

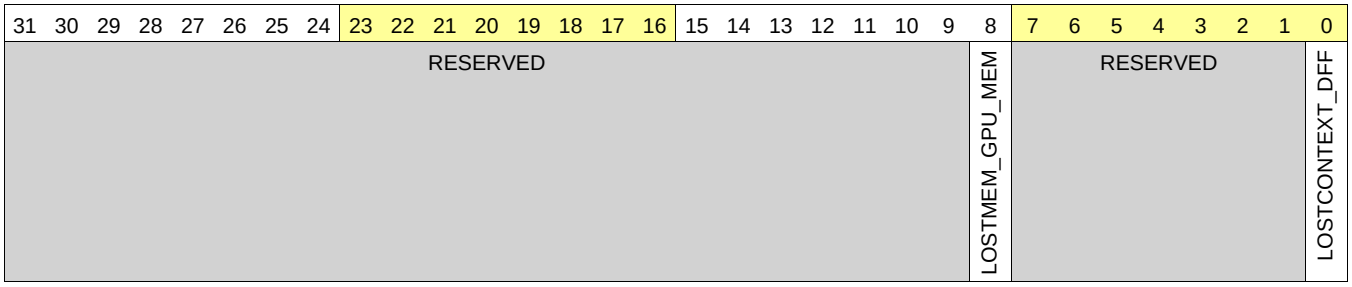
31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4	3	2	1 0
RESERVED	LASTPOWERSTATEENTERED	RESERVED	INTRANSITION	RESERVED	GPU_MEM_STATEST	RESERVED	LOGICSTATEST	POWERSTATEST

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:24	LASTPOWERSTATEENTERED	Last low power state entered. Set to 0x3 upon write of the same only. This register is intended for debug purpose only. 0x0: Power domain was previously OFF 0x1: Power domain was previously in RETENTION 0x2: Power domain was previously ON-INACTIVE 0x3: Power domain was previously ON-ACTIVE	RW	0x0
23:21	RESERVED		R	0x0
20	INTRANSITION	Domain transition status 0x0: No on-going transition on power domain 0x1: Power domain transition is in progress.	R	0x0
19:6	RESERVED		R	0x0
5:4	GPU_MEM_STATEST	GPU_MEM memory bank state status 0x0: Memory is OFF 0x1: Reserved 0x2: Reserved 0x3: Memory is ON	R	0x3
3	RESERVED		R	0x0
2	LOGICSTATEST	Logic state status 0x0: Logic in domain is OFF 0x1: Logic in domain is ON	R	0x1

Bits	Field Name	Description	Type	Reset
1:0	POWERSTATEST	Current power state status 0x0: Power domain is OFF 0x1: Power domain is in RETENTION 0x2: Power domain is ON-INACTIVE 0x3: Power domain is ON-ACTIVE	R	0x3

Table 3-1395. RM_GPU_GPU_CONTEXT

Address Offset	0x0000 0024		
Physical Address	0x4AE0 7224	**Instance**	GPU_PRM
Description	This register contains dedicated GPU context statuses. [warm reset insensitive]		
Type	RW		



Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	LOSTMEM_GPU_MEM	Specify if memory-based context in GPU_MEM memory bank has been lost due to a previous power transition or other reset source. 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
7:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of GPU_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

3.12.46 INSTR_PRM registers

3.12.46.1 INSTR_PRM Register Summary

Table 3-1396. INSTR_PRM Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	INSTR_PRM Physical Address
PMI_IDENTICATION	R	32	0x0000 0000	0x4AE0 7F00
PMI_SYS_CONFIG	RW	32	0x0000 0010	0x4AE0 7F10
PMI_STATUS	R	32	0x0000 0014	0x4AE0 7F14
PMI_CONFIGURATION	RW	32	0x0000 0024	0x4AE0 7F24
PMI_CLASS_FILTERING	RW	32	0x0000 0028	0x4AE0 7F28
PMI_TRIGGERING	RW	32	0x0000 002C	0x4AE0 7F2C
PMI_SAMPLING	RW	32	0x0000 0030	0x4AE0 7F30

3.12.46.2 INSTR_PRM Register Description

Table 3-1397. PMI_IDENTICATION

Address Offset	0x0000 0000		
Physical Address	0x4AE0 7F00	**Instance**	INSTR_PRM
Description	PM profiling identification register		
Type	R		

31 30	29 28	27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11	10 9 8	7 6	5 4 3 2 1 0
SCHEME	RESERVED	FUNC		RTL	MAJOR	CUSTOM	MINOR

Bits	Field Name	Description	Type	Reset
31:30	SCHEME		R	0x1
29:28	RESERVED		R	0x1
27:16	FUNC	Indicates a software compatible module family	R	0x5
15:11	RTL	RTL version	R	0x9
10:8	MAJOR	Major revision	R	0x0
7:6	CUSTOM	Indicates a special version for a particular device	R	0x0
5:0	MINOR	Minor revision	R	0x0

Table 3-1398. Register Call Summary for Register PMI_IDENTICATION

PRCM Register Manual
- INSTR_PRM Register Summary: [0]

Table 3-1399. PMI_SYS_CONFIG

Address Offset	0x0000 0010		
Physical Address	0x4AE0 7F10	**Instance**	INSTR_PRM
Description	PM profiling system configuartion register		
Type	RW		

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6	5 4	3 2	1	0
RESERVED				RESERVED	IDLEMODE	RESERVED	SOFTRESET

Bits	Field Name	Description	Type	Reset
31:4	RESERVED		R	0x0
3:2	IDLEMODE	Configuration of the local target state management mode	RW	0x2
1	RESERVED		R	0x0
0	SOFTRESET	Software reset	RW	0x0

Table 3-1400. Register Call Summary for Register PMI_SYS_CONFIG

PRCM Register Manual
- INSTR_PRM Register Summary: [0]

Table 3-1401. PMI_STATUS

Address Offset	0x0000 0014		
Physical Address	0x4AE0 7F14	**Instance**	INSTR_PRM
Description	PM profiling status register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							FIFOEMPTY	RESERVED							

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	FIFOEMPTY	PM Profiling buffer empty	R	0x1
7:0	RESERVED		R	0x0

Table 3-1402. Register Call Summary for Register PMI_STATUS

PRCM Register Manual
- INSTR_PRM Register Summary: [0]

Table 3-1403. PMI_CONFIGURATION

Address Offset	0x0000 0024		
Physical Address	0x4AE0 7F24	**Instance**	INSTR_PRM
Description	PM profiling configuration register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CLAIM_3		CLAIM_2	CLAIM_1	RESERVED				RESERVED								RESERVED								EVT_CAPT_EN	RESERVED						

Bits	Field Name	Description	Type	Reset
31:30	CLAIM_3	Ownership	RW	0x0
29	CLAIM_2	Debugger override qualifier	RW	0x1
28	CLAIM_1	Current owner	R	0x0
27:8	RESERVED		R	0x0
7	EVT_CAPT_EN	When HIGH the PM events capture is enabled	RW	0x0
6:0	RESERVED		R	0x0

Table 3-1404. Register Call Summary for Register PMI_CONFIGURATION

PRCM Register Manual
- INSTR_PRM Register Summary: [0]

Table 3-1405. PMI_CLASS_FILTERING

Address Offset	0x0000 0028		
Physical Address	0x4AE0 7F28	**Instance**	INSTR_PRM
Description	PM profiling class filtering register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												SNAP_CAPT_EN_03	SNAP_CAPT_EN_02	SNAP_CAPT_EN_01	SNAP_CAPT_EN_00

Bits	Field Name	Description	Type	Reset
31:4	RESERVED		R	0x0
3	SNAP_CAPT_EN_03	Snapshot capture enable - Class-ID = 0x03	RW	0x0
2	SNAP_CAPT_EN_02	Snapshot capture enable - Class-ID = 0x02	RW	0x0
1	SNAP_CAPT_EN_01	Snapshot capture enable - Class-ID = 0x01	RW	0x0
0	SNAP_CAPT_EN_00	Snapshot capture enable - Class-ID = 0x00	RW	0x0

Table 3-1406. Register Call Summary for Register PMI_CLASS_FILTERING

PRCM Register Manual
- INSTR_PRM Register Summary: [0]

Table 3-1407. PMI_TRIGGERING

Address Offset	0x0000 002C		
Physical Address	0x4AE0 7F2C	**Instance**	INSTR_PRM
Description	PM profiling triggering control register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														TRIG_STOP_EN	TRIG_START_EN

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1	TRIG_STOP_EN	Enable stop capturing PM events from external trigger detection	RW	0x0
0	TRIG_START_EN	Enable start capturing PM events from external trigger detection	RW	0x0

Table 3-1408. Register Call Summary for Register PMI_TRIGGERING

PRCM Register Manual
- INSTR_PRM Register Summary: [0]

Table 3-1409. PMI_SAMPLING

Address Offset	0x0000 0030		
Physical Address	0x4AE0 7F30	**Instance**	INSTR_PRM
Description	PM profiling sampling window register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												FCLK_DIV_FACOR				RESERVED								SAMP_WIND_SIZE							

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19:16	FCLK_DIV_FACOR	FunClk divide factor ranging from 1 to 16	RW	0x0
15:8	RESERVED		R	0x0
7:0	SAMP_WIND_SIZE	PM events sampling window size	RW	0x0

Table 3-1410. Register Call Summary for Register PMI_SAMPLING

PRCM Register Manual
- INSTR_PRM Register Summary: [0]

3.12.47 IPU_PRM registers

3.12.47.1 IPU_PRM Register Summary

Table 3-1411. IPU_PRM Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	IPU_PRM Physical Address
PM_IPU_PWRSTCTRL	RW	32	0x0000 0000	0x4AE0 6500
PM_IPU_PWRSTST	RW	32	0x0000 0004	0x4AE0 6504
RM_IPU1_RSTCTRL	RW	32	0x0000 0010	0x4AE0 6510
RM_IPU1_RSTST	RW	32	0x0000 0014	0x4AE0 6514
RM_IPU1_IPU1_CONTEXT	RW	32	0x0000 0024	0x4AE0 6524
PM_IPU_MCASP1_WKDEP	RW	32	0x0000 0050	0x4AE0 6550
RM_IPU_MCASP1_CONTEXT	RW	32	0x0000 0054	0x4AE0 6554
PM_IPU_TIMER5_WKDEP	RW	32	0x0000 0058	0x4AE0 6558
RM_IPU_TIMER5_CONTEXT	RW	32	0x0000 005C	0x4AE0 655C
PM_IPU_TIMER6_WKDEP	RW	32	0x0000 0060	0x4AE0 6560
RM_IPU_TIMER6_CONTEXT	RW	32	0x0000 0064	0x4AE0 6564
PM_IPU_TIMER7_WKDEP	RW	32	0x0000 0068	0x4AE0 6568
RM_IPU_TIMER7_CONTEXT	RW	32	0x0000 006C	0x4AE0 656C

Table 3-1411. IPU_PRM Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	IPU_PRM Physical Address
PM_IPU_TIMER8_WKDEP	RW	32	0x0000 0070	0x4AE0 6570
RM_IPU_TIMER8_CONTEXT	RW	32	0x0000 0074	0x4AE0 6574
PM_IPU_I2C5_WKDEP	RW	32	0x0000 0078	0x4AE0 6578
RM_IPU_I2C5_CONTEXT	RW	32	0x0000 007C	0x4AE0 657C
PM_IPU_UART6_WKDEP	RW	32	0x0000 0080	0x4AE0 6580
RM_IPU_UART6_CONTEXT	RW	32	0x0000 0084	0x4AE0 6584

3.12.47.2 IPU_PRM Register Description

Table 3-1412. PM_IPU_PWRSTCTRL

Address Offset	0x0000 0000		
Physical Address	0x4AE0 6500	**Instance**	IPU_PRM
Description	This register controls the IPU domain power state to reach upon a domain sleep transition		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED										PERIPHMEM_ONSTATE		RESERVED		AESSMEM_ONSTATE		RESERVED					PERIPHMEM_RETSTATE	RESERVED	AESSMEM_RETSTATE	RESERVED			LOWPOWERSTATECHANGE	RESERVED	LOGICRETSTATE	POWERSTATE	

Bits	Field Name	Description	Type	Reset
31:22	RESERVED		R	0x0
21:20	PERIPHMEM_ONSTATE	PERIPHMEM memory state when domain is ON. 0x3: Memory bank is on when the domain is ON.	R	0x3
19:18	RESERVED		R	0x0
17:16	AESSMEM_ONSTATE	AESSMEM memory state when domain is ON. 0x3: Memory bank is on when the domain is ON.	R	0x3
15:11	RESERVED		R	0x0
10	PERIPHMEM_RETSTATE	PERIPHMEM memory state when domain is RETENTION. 0x0: Memory bank is off when the domain is in the RETENTION state. 0x1: Memory bank is retained when domain is in RETENTION state.	RW	0x0
9	RESERVED		R	0x0
8	AESSMEM_RETSTATE	AESSMEM memory state when domain is RETENTION. 0x0: Memory bank is off when the domain is in the RETENTION state. 0x1: Memory bank is retained when domain is in RETENTION state.	RW	0x1
7:5	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
4	LOWPOWERSTATECHANGE	Power state change request when domain has already performed a sleep transition. Allows going into deeper low power state without waking up the power domain. 0x0: Do not request a low power state change. 0x1: Request a low power state change. This bit is automatically cleared when the power state is effectively changed or when power state is ON.	RW	0x0
3	RESERVED		R	0x0
2	LOGICRETSTATE	Logic state when power domain is RETENTION 0x0: Whole logic is off when the domain is in RETENTION state.	R	0x0
1:0	POWERSTATE	Power state control 0x0: OFF state 0x1: RETENTION state 0x2: INACTIVE state 0x3: ON State	RW	0x0

Table 3-1413. Register Call Summary for Register PM_IPU_PWRSTCTRL

Power Management Functional Description
- Logic and Memory Area Power Modes Control and Status: [0][1][2][3][4][5][6]

PRCM Register Manual
- Not Supported Functionality (Registers and Bitfields): [7][8][9][10][11]
- IPU_PRM Register Summary: [12]

Table 3-1414. PM_IPU_PWRSTST

Address Offset	0x0000 0004		
Physical Address	0x4AE0 6504	**Instance**	IPU_PRM
Description	This register provides a status on the IPU domain current power domain state. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED							LASTPOWERSTATEENTERED	RESERVED			INTRANSITION	RESERVED										PERIPHMEM_STATEST		RESERVED		AESSMEM_STATEST		RESERVED		LOGICSTATEST		POWERSTATEST

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:24	LASTPOWERSTATEENTERED	Last low power state entered. Set to 0x3 upon write of the same only. This register is intended for debug purpose only. 0x0: Power domain was previously OFF 0x1: Power domain was previously in RETENTION 0x2: Power domain was previously ON-INACTIVE 0x3: Power domain was previously ON-ACTIVE	RW	0x0
23:21	RESERVED		R	0x0
20	INTRANSITION	Domain transition status 0x0: No on-going transition on power domain 0x1: Power domain transition is in progress.	R	0x0
19:10	RESERVED		R	0x0
9:8	PERIPHMEM_STATEST	PERIPHMEM memory state status 0x0: Memory is OFF 0x1: Reserved 0x2: Reserved 0x3: Memory is ON	R	0x3
7:6	RESERVED		R	0x0
5:4	AESSMEM_STATEST	AESSMEM memory state status 0x0: Memory is OFF 0x1: Memory is RETENTION 0x2: Reserved 0x3: Memory is ON	R	0x3
3	RESERVED		R	0x0
2	LOGICSTATEST	Logic state status 0x0: Logic in domain is OFF 0x1: Logic in domain is ON	R	0x1
1:0	POWERSTATEST	Current power state status 0x0: Power domain is OFF 0x1: Power domain is in RETENTION 0x2: Power domain is ON-INACTIVE 0x3: Power domain is ON-ACTIVE	R	0x3

Table 3-1415. Register Call Summary for Register PM_IPU_PWRSTST

Power Management Functional Description • Logic and Memory Area Power Modes Control and Status: [0][1][2][3][4][5]
PRCM Register Manual • Not Supported Functionality (Registers and Bitfields): [6][7][8][9][10][11] • IPU_PRM Register Summary: [12]

Table 3-1416. RM_IPU1_RSTCTRL

Address Offset	0x0000 0010		
Physical Address	0x4AE0 6510	**Instance**	IPU_PRM
Description	This register controls the release of the IPU1 sub-system resets.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																													RST_IPU	RST_CPU1	RST_CPU0

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0
2	RST_IPU	BELLINI system reset control. 0x0: Reset is cleared for IPU CACHE MMU 0x1: Reset is asserted for the IPU CACHE MMU	RW	0x1
1	RST_CPU1	BELLINI Cortex M3 CPU1 reset control 0x0: Reset is cleared for the IPU Cortex M3 CPU1 0x1: Reset is asserted for the IPU Cortex M3 CPU1	RW	0x1
0	RST_CPU0	BELLINI Cortex M3 CPU0 reset control. 0x0: Reset is cleared for the IPU Cortex M3 CPU0 0x1: Reset is asserted for the IPU Cortex M3 CPU0	RW	0x1

Table 3-1417. RM_IPU1_RSTST

Address Offset	0x0000 0014		
Physical Address	0x4AE0 6514	**Instance**	IPU_PRM
Description	This register logs the different reset sources of the IPU1 SS. Each bit is set upon release of the domain reset signal. Must be cleared by software. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								RST_ICECRUSHER_CPU1	RST_ICECRUSHER_CPU0	RST_EMULATION_CPU1	RST_EMULATION_CPU0	RST_IPU	RST_CPU1	RST_CPU0

Bits	Field Name	Description	Type	Reset
31:7	RESERVED		R	0x0
6	RST_ICECRUSHER_CPU1	CPU1 has been reset due to IPU ICECRUSHER1 reset source Read 0x0: No icecrusher reset Read 0x1: CPU1 has been reset upon icecrusher reset Write 0x0: No effect Write 0x1: Clear Reset	RW	0x0
5	RST_ICECRUSHER_CPU0	CPU0 has been reset due to IPU ICECRUSHER0 reset source Read 0x0: No icecrusher reset Read 0x1: CPU0 has been reset upon icecrusher reset Write 0x0: No effect Write 0x1: Clear Reset	RW	0x0

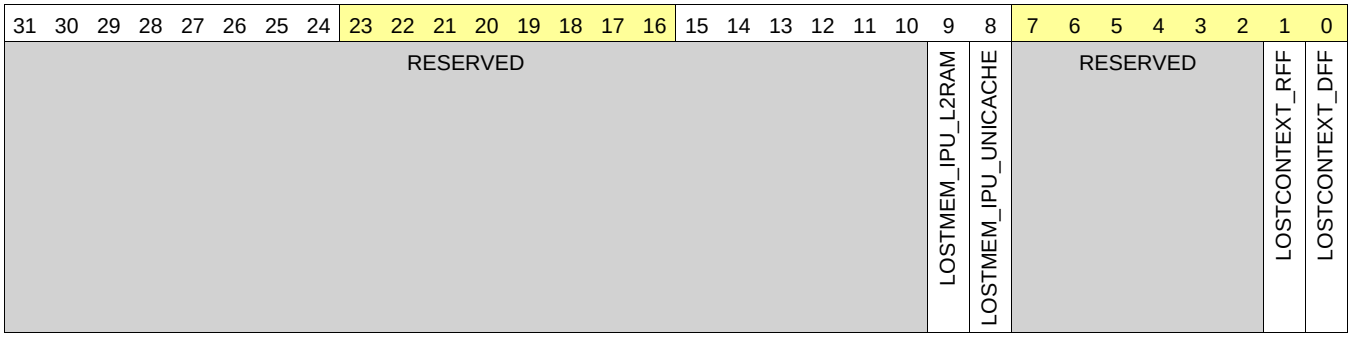
Bits	Field Name	Description	Type	Reset
4	RST_EMULATION_CPU1	CPU1 has been reset due to emulation reset source e.g. assert reset command initiated by the icepick module Read 0x0: No emulation reset Read 0x1: CPU1 has been reset upon emulation reset Write 0x0: No effect Write 0x1: Clear Reset	RW	0x0
3	RST_EMULATION_CPU0	CPU0 has been reset due to emulation reset source e.g. assert reset command initiated by the icepick module Read 0x0: No emulation reset Read 0x1: CPU0 has been reset upon emulation reset Write 0x0: No effect Write 0x1: Clear Reset	RW	0x0
2	RST_IPU	IPU system SW reset status Read 0x0: No software reset occurred Read 0x1: IPU MMU and CACHE interface has been reset upon SW reset Write 0x0: No effect Write 0x1: Clear Reset	RW	0x0
1	RST_CPU1	CPU1 SW reset status Read 0x0: No software reset occurred Read 0x1: Cortex M4 CPU1 has been reset upon software reset Write 0x0: No effect Write 0x1: Clear Reset	RW	0x0
0	RST_CPU0	CPU0 SW reset status Read 0x0: No software reset occurred Read 0x1: Cortex M4 CPU0 has been reset upon software reset Write 0x0: No effect Write 0x1: Clear Reset	RW	0x0

Table 3-1418. Register Call Summary for Register RM_IPU1_RSTST

PRCM Register Manual
- IPU_PRM Register Summary: [0]

Table 3-1419. RM_IPU1_IPU1_CONTEXT

Address Offset	0x0000 0024		
Physical Address	0x4AE0 6524	**Instance**	IPU_PRM
Description	This register contains dedicated IPU1 context statuses. [warm reset insensitive]		
Type	RW		



Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	LOSTMEM_IPU_L2RAM	Specify if memory-based context in IPU_L2RAM memory bank has been lost due to a previous power transition or other reset source. 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
8	LOSTMEM_IPU_UNICACHE	Specify if memory-based context in IPU_UNICACHE memory bank has been lost due to a previous power transition or other reset source. 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
7:2	RESERVED		R	0x0
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of IPU_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of IPU_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1420. Register Call Summary for Register RM_IPU1_IPU1_CONTEXT

Power Management Functional Description • PD_IPU Description: [0][1]
PRCM Register Manual • IPU_PRM Register Summary: [2]

Table 3-1421. PM_IPU_MCASP1_WKDEP

Address Offset	0x0000 0050		
Physical Address	0x4AE0 6550	**Instance**	IPU_PRM
Description	This register controls wakeup dependency based on McASP1 service requests.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																WKUPDEP_MCASP1_DMA_DSP2	RESERVED	WKUPDEP_MCASP1_DMA_SDMA	WKUPDEP_MCASP1_DMA_DSP1	RESERVED		WKUPDEP_MCASP1_IRQ_EVE4	WKUPDEP_MCASP1_IRQ_EVE3	WKUPDEP_MCASP1_IRQ_EVE2	WKUPDEP_MCASP1_IRQ_EVE1	WKUPDEP_MCASP1_IRQ_DSP2	WKUPDEP_MCASP1_IRQ_IPU1	RESERVED	WKUPDEP_MCASP1_IRQ_DSP1	WKUPDEP_MCASP1_IRQ_IPU2	WKUPDEP_MCASP1_IRQ_MPU

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0
15	WKUPDEP_MCASP1_DMA_DSP2	Wakeup dependency from McASP1 module (SWakeup_dma signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
14	RESERVED		R	0x0
13	WKUPDEP_MCASP1_DMA_SDMA	Wakeup dependency from McASP1 module (SWakeup_dma signal) towards SDMA + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
12	WKUPDEP_MCASP1_DMA_DSP1	Wakeup dependency from McASP1 module (SWakeup_dma signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
11:10	RESERVED		R	0x0
9	WKUPDEP_MCASP1_IRQ_EVE4	Wakeup dependency from McASP1 module (SWakeup IRQ signal) towards EVE4 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
8	WKUPDEP_MCASP1_IRQ_EVE3	Wakeup dependency from McASP1 module (SWakeup IRQ signal) towards EVE3 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
7	WKUPDEP_MCASP1_IRQ_EVE2	Wakeup dependency from McASP1 module (SWakeup IRQ signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_MCASP1_IRQ_EVE1	Wakeup dependency from McASP1 module (SWakeup IRQ signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
5	WKUPDEP_MCASP1_IRQ_DSP2	Wakeup dependency from McASP1 module (SWakeup IRQ signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_MCASP1_IRQ_IPU1	Wakeup dependency from McASP1 module (SWakeup IRQ signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	RESERVED		R	0x0
2	WKUPDEP_MCASP1_IRQ_DSP1	Wakeup dependency from McASP1 module (SWakeup IRQ signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_MCASP1_IRQ_IPU2	Wakeup dependency from McASP1 module (SWakeup IRQ signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_MCASP1_IRQ_MPU	Wakeup dependency from McASP1 module (SWakeup IRQ signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1422. Register Call Summary for Register PM_IPU_MCASP1_WKDEP

PRCM Register Manual
- Not Supported Functionality (Registers and Bitfields): [0][1][2][3][4][5]
- IPU_PRM Register Summary: [6]

Table 3-1423. RM_IPU_MCASP1_CONTEXT

Address Offset	0x0000 0054		
Physical Address	0x4AE0 6554	**Instance**	IPU_PRM
Description	This register contains dedicated McASP context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of ABE_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1424. Register Call Summary for Register RM_IPU_MCASP1_CONTEXT

PRCM Register Manual
- IPU_PRM Register Summary: [0]

Table 3-1425. PM_IPU_TIMER5_WKDEP

Address Offset	0x0000 0058		
Physical Address	0x4AE0 6558	**Instance**	IPU_PRM
Description	This register controls wakeup dependency based on TIMER5 service requests.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						WKUPDEP_TIMER5_EVE4	WKUPDEP_TIMER5_EVE3	WKUPDEP_TIMER5_EVE2	WKUPDEP_TIMER5_EVE1	WKUPDEP_TIMER5_DSP2	WKUPDEP_TIMER5_IPU1	RESERVED	WKUPDEP_TIMER5_DSP1	WKUPDEP_TIMER5_IPU2	WKUPDEP_TIMER5_MPU

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	WKUPDEP_TIMER5_EVE4	Wakeup dependency from TIMER5 module (SWakeup signal) towards EVE4 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
8	WKUPDEP_TIMER5_EVE3	Wakeup dependency from TIMER5 module (SWakeup signal) towards EVE3 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
7	WKUPDEP_TIMER5_EVE2	Wakeup dependency from TIMER5 module (SWakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_TIMER5_EVE1	Wakeup dependency from TIMER5 module (SWakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
5	WKUPDEP_TIMER5_DSP2	Wakeup dependency from TIMER5 module (SWakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_TIMER5_IPU1	Wakeup dependency from TIMER5 module (SWakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	RESERVED		R	0x0
2	WKUPDEP_TIMER5_DSP1	Wakeup dependency from TIMER5 module (SWakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_TIMER5_IPU2	Wakeup dependency from TIMER5 module (SWakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_TIMER5_MPU	Wakeup dependency from TIMER5 module (SWakeup signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1426. Register Call Summary for Register PM_IPU_TIMER5_WKDEP

PRCM Register Manual
- Not Supported Functionality (Registers and Bitfields): [0][1][2][3][4]
- IPU_PRM Register Summary: [5]

Table 3-1427. RM_IPU_TIMER5_CONTEXT

Address Offset	0x0000 005C		
Physical Address	0x4AE0 655C	**Instance**	IPU_PRM
Description	This register contains dedicated TIMER5 context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of ABE_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1428. Register Call Summary for Register RM_IPU_TIMER5_CONTEXT

PRCM Register Manual
- IPU_PRM Register Summary: [0]

Table 3-1429. PM_IPU_TIMER6_WKDEP

Address Offset	0x0000 0060		
Physical Address	0x4AE0 6560	**Instance**	IPU_PRM
Description	This register controls wakeup dependency based on TIMER6 service requests.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						WKUPDEP_TIMER6_EVE4	WKUPDEP_TIMER6_EVE3	WKUPDEP_TIMER6_EVE2	WKUPDEP_TIMER6_EVE1	WKUPDEP_TIMER6_DSP2	WKUPDEP_TIMER6_IPU1	RESERVED	WKUPDEP_TIMER6_DSP1	WKUPDEP_TIMER6_IPU2	WKUPDEP_TIMER6_MPU

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	WKUPDEP_TIMER6_EVE4	Wakeup dependency from TIMER6 module (SWakeup signal) towards EVE4 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
8	WKUPDEP_TIMER6_EVE3	Wakeup dependency from TIMER6 (SWakeup signal) towards EVE3 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
7	WKUPDEP_TIMER6_EVE2	Wakeup dependency from TIMER6 module (SWakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_TIMER6_EVE1	Wakeup dependency from TIMER6 module (SWakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
5	WKUPDEP_TIMER6_DSP2	Wakeup dependency from TIMER6 module (SWakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_TIMER6_IPU1	Wakeup dependency from TIMER6 module (SWakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	RESERVED		R	0x0
2	WKUPDEP_TIMER6_DSP1	Wakeup dependency from TIMER6 module (SWakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_TIMER6_IPU2	Wakeup dependency from TIMER6 module (SWakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_TIMER6_MPU	Wakeup dependency from TIMER6 module (SWakeup signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1430. Register Call Summary for Register PM_IPU_TIMER6_WKDEP

PRCM Register Manual
- Not Supported Functionality (Registers and Bitfields): [0][1][2][3][4]
- IPU_PRM Register Summary: [5]

Table 3-1431. RM_IPU_TIMER6_CONTEXT

Address Offset	0x0000 0064		
Physical Address	0x4AE0 6564	**Instance**	IPU_PRM
Description	This register contains dedicated TIMER6 context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of ABE_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1432. Register Call Summary for Register RM_IPU_TIMER6_CONTEXT

PRCM Register Manual
- IPU_PRM Register Summary: [0]

Table 3-1433. PM_IPU_TIMER7_WKDEP

Address Offset	0x0000 0068		
Physical Address	0x4AE0 6568	**Instance**	IPU_PRM
Description	This register controls wakeup dependency based on TIMER7 service requests.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						WKUPDEP_TIMER7_EVE4	WKUPDEP_TIMER7_EVE3	WKUPDEP_TIMER7_EVE2	WKUPDEP_TIMER7_EVE1	WKUPDEP_TIMER7_DSP2	WKUPDEP_TIMER7_IPU1	RESERVED	WKUPDEP_TIMER7_DSP1	WKUPDEP_TIMER7_IPU2	WKUPDEP_TIMER7_MPU

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	WKUPDEP_TIMER7_EVE4	Wakeup dependency from TIMER7 module (SWakeup signal) towards EVE4 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
8	WKUPDEP_TIMER7_EVE3	Wakeup dependency from TIMER7 (SWakeup signal) towards EVE3 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
7	WKUPDEP_TIMER7_EVE2	Wakeup dependency from TIMER7 module (SWakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_TIMER7_EVE1	Wakeup dependency from TIMER7 module (SWakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
5	WKUPDEP_TIMER7_DSP2	Wakeup dependency from TIMER7 module (SWakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_TIMER7_IPU1	Wakeup dependency from TIMER7 module (SWakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	RESERVED		R	0x0
2	WKUPDEP_TIMER7_DSP1	Wakeup dependency from TIMER7 module (SWakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_TIMER7_IPU2	Wakeup dependency from TIMER7 module (SWakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_TIMER7_MPU	Wakeup dependency from TIMER7 module (SWakeup signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1434. Register Call Summary for Register PM_IPU_TIMER7_WKDEP

PRCM Register Manual
- Not Supported Functionality (Registers and Bitfields): [0][1][2][3][4]
- IPU_PRM Register Summary: [5]

Table 3-1435. RM_IPU_TIMER7_CONTEXT

Address Offset	0x0000 006C		
Physical Address	0x4AE0 656C	**Instance**	IPU_PRM
Description	This register contains dedicated TIMER7 context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of ABE_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1436. Register Call Summary for Register RM_IPU_TIMER7_CONTEXT

PRCM Register Manual
- IPU_PRM Register Summary: [0]

Table 3-1437. PM_IPU_TIMER8_WKDEP

Address Offset	0x0000 0070		
Physical Address	0x4AE0 6570	**Instance**	IPU_PRM
Description	This register controls wakeup dependency based on TIMER8 service requests.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						WKUPDEP_TIMER8_EVE4	WKUPDEP_TIMER8_EVE3	WKUPDEP_TIMER8_EVE2	WKUPDEP_TIMER8_EVE1	WKUPDEP_TIMER8_DSP2	WKUPDEP_TIMER8_IPU1	RESERVED	WKUPDEP_TIMER8_DSP1	WKUPDEP_TIMER8_IPU2	WKUPDEP_TIMER8_MPU

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	WKUPDEP_TIMER8_EVE4	Wakeup dependency from TIMER8 module (SWakeup signal) towards EVE4 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
8	WKUPDEP_TIMER8_EVE3	Wakeup dependency from TIMER8 (SWakeup signal) towards EVE3 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
7	WKUPDEP_TIMER8_EVE2	Wakeup dependency from TIMER8 module (SWakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_TIMER8_EVE1	Wakeup dependency from TIMER8 module (SWakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
5	WKUPDEP_TIMER8_DSP2	Wakeup dependency from TIMER8 module (SWakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_TIMER8_IPU1	Wakeup dependency from TIMER8 module (SWakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	RESERVED		R	0x0
2	WKUPDEP_TIMER8_DSP1	Wakeup dependency from TIMER8 module (SWakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_TIMER8_IPU2	Wakeup dependency from TIMER8 module (SWakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_TIMER8_MPU	Wakeup dependency from TIMER8 module (SWakeup signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1438. Register Call Summary for Register PM_IPU_TIMER8_WKDEP

PRCM Register Manual
- Not Supported Functionality (Registers and Bitfields): [0][1][2][3][4]
- IPU_PRM Register Summary: [5]

Table 3-1439. RM_IPU_TIMER8_CONTEXT

Address Offset	0x0000 0074		
Physical Address	0x4AE0 6574	**Instance**	IPU_PRM
Description	This register contains dedicated TIMER8 context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of ABE_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1440. Register Call Summary for Register RM_IPU_TIMER8_CONTEXT

PRCM Register Manual
- IPU_PRM Register Summary: [0]

Table 3-1441. PM_IPU_I2C5_WKDEP

Address Offset	0x0000 0078		
Physical Address	0x4AE0 6578	**Instance**	IPU_PRM
Description	This register controls wakeup dependency based on I2C5 service requests.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																WKUPDEP_I2C5_DMA_DSP2	RESERVED	WKUPDEP_I2C5_DMA_SDMA	WKUPDEP_I2C5_DMA_DSP1	RESERVED		WKUPDEP_I2C5_IRQ_EVE4	WKUPDEP_I2C5_IRQ_EVE3	WKUPDEP_I2C5_IRQ_EVE2	WKUPDEP_I2C5_IRQ_EVE1	WKUPDEP_I2C5_IRQ_DSP2	WKUPDEP_I2C5_IRQ_IPU1	RESERVED	WKUPDEP_I2C5_IRQ_DSP1	WKUPDEP_I2C5_IRQ_IPU2	WKUPDEP_I2C5_IRQ_MPU

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0
15	WKUPDEP_I2C5_DMA_DSP2	Wakeup dependency from I2C5 module (SWakeup_dma signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
14	RESERVED		R	0x0
13	WKUPDEP_I2C5_DMA_SDMA	Wakeup dependency from I2C5 module (SWakeup_dma signal) towards SDMA + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
12	WKUPDEP_I2C5_DMA_DSP1	Wakeup dependency from I2C5 module (SWakeup_dma signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
11:10	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
9	WKUPDEP_I2C5_IRQ_EVE4	Wakeup dependency from I2C5 module (SWakeup IRQ signal) towards EVE4 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
8	WKUPDEP_I2C5_IRQ_EVE3	Wakeup dependency from I2C5 module (SWakeup IRQ signal) towards EVE3 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
7	WKUPDEP_I2C5_IRQ_EVE2	Wakeup dependency from I2C5 module (SWakeup IRQ signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_I2C5_IRQ_EVE1	Wakeup dependency from I2C5 module (SWakeup IRQ signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_I2C5_IRQ_DSP2	Wakeup dependency from I2C5 module (SWakeup IRQ signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_I2C5_IRQ_IPU1	Wakeup dependency from I2C5 module (SWakeup IRQ signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	RESERVED		R	0x0
2	WKUPDEP_I2C5_IRQ_DSP1	Wakeup dependency from I2C5 module (SWakeup IRQ signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_I2C5_IRQ_IPU2	Wakeup dependency from I2C5 module (SWakeup IRQ signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_I2C5_IRQ_MPU	Wakeup dependency from I2C5 module (SWakeup IRQ signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1442. RM_IPU_I2C5_CONTEXT

Address Offset	0x0000 007C		
Physical Address	0x4AE0 657C	**Instance**	IPU_PRM
Description	This register contains dedicated I2C5 context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4PER_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1443. PM_IPU_UART6_WKDEP

Address Offset	0x0000 0080		
Physical Address	0x4AE0 6580	**Instance**	IPU_PRM
Description	This register controls wakeup dependency based on UART6 service requests.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						WKUPDEP_UART6_EVE4	WKUPDEP_UART6_EVE3	WKUPDEP_UART6_EVE2	WKUPDEP_UART6_EVE1	WKUPDEP_UART6_DSP2	WKUPDEP_UART6_IPU1	WKUPDEP_UART6_SDMA	WKUPDEP_UART6_DSP1	WKUPDEP_UART6_IPU2	WKUPDEP_UART6_MPU

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	WKUPDEP_UART6_EVE4	Wakeup dependency from UART6 module (SWakeup signal) towards EVE4 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
8	WKUPDEP_UART6_EVE3	Wakeup dependency from UART6 module (SWakeup signal) towards EVE3 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
7	WKUPDEP_UART6_EVE2	Wakeup dependency from UART6 module (SWakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
6	WKUPDEP_UART6_EVE1	Wakeup dependency from UART6 module (SWakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_UART6_DSP2	Wakeup dependency from UART6 module (SWakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_UART6_IPU1	Wakeup dependency from UART6 module (SWakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	WKUPDEP_UART6_SDMA	Wakeup dependency from UART6 module (SWakeup signal) towards SDMA + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
2	WKUPDEP_UART6_DSP1	Wakeup dependency from UART6 module (SWakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_UART6_IPU2	Wakeup dependency from UART6 module (SWakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_UART6_MPU	Wakeup dependency from UART6 module (SWakeup signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1444. RM_IPU_UART6_CONTEXT

Address Offset	0x0000 0084		
Physical Address	0x4AE0 6584	**Instance**	IPU_PRM
Description	This register contains dedicated UART6 context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							LOSTMEM_RETAINED_BANK	RESERVED						LOSTCONTEXT_RFF	RESERVED

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	LOSTMEM_RETAINED_BANK	Specify if memory-based context in RETAINED_BANK memory bank has been lost due to a previous power transition or other reset source. 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
7:2	RESERVED		R	0x0
1	LOSTCONTEXT_RFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4PER_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
0	RESERVED		R	0x0

3.12.48 IVA_PRM registers

3.12.48.1 IVA_PRM Register Summary

Table 3-1445. IVA_PRM Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	IVA_PRM Physical Address
PM_IVA_PWRSTCTRL	RW	32	0x0000 0000	0x4AE0 6F00
PM_IVA_PWRSTST	RW	32	0x0000 0004	0x4AE0 6F04
RM_IVA_RSTCTRL	RW	32	0x0000 0010	0x4AE0 6F10
RM_IVA_RSTST	RW	32	0x0000 0014	0x4AE0 6F14
RM_IVA_IVA_CONTEXT	RW	32	0x0000 0024	0x4AE0 6F24
RM_IVA_SL2_CONTEXT	RW	32	0x0000 002C	0x4AE0 6F2C

3.12.48.2 IVA_PRM Register Description

Table 3-1446. PM_IVA_PWRSTCTRL

Address Offset	0x0000 0000		
Physical Address	0x4AE0 6F00	**Instance**	IVA_PRM
Description	This register controls the IVA power state to reach upon a domain sleep transition		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								TCM2_MEM_ONSTATE		TCM1_MEM_ONSTATE		SL2_MEM_ONSTATE		HWA_MEM_ONSTATE		RESERVED				TCM2_MEM_RETSTATE		TCM1_MEM_RETSTATE		SL2_MEM_RETSTATE		HWA_MEM_RETSTATE		RESERVED		LOWPOWERSTATECHANGE		RESERVED		LOGICRETSTATE		POWERSTATE	

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:22	TCM2_MEM_ONSTATE	TCM_CORE memory state when domain is ON. 0x3: Memory bank is on when the domain is ON.	R	0x3
21:20	TCM1_MEM_ONSTATE	TCM1 memory state when domain is ON. 0x3: Memory bank is on when the domain is ON.	R	0x3
19:18	SL2_MEM_ONSTATE	SL2 memory state when domain is ON. 0x3: Memory bank is on when the domain is ON.	R	0x3
17:16	HWA_MEM_ONSTATE	HWA memory state when domain is ON. 0x3: Memory bank is on when the domain is ON.	R	0x3
15:12	RESERVED		R	0x0
11	TCM2_MEM_RETSTATE	TCM2 memory state when domain is RETENTION. 0x0: Memory bank is off when the domain is in the RETENTION state. 0x1: Memory bank is retained when domain is in RETENTION state.	RW	0x1
10	TCM1_MEM_RETSTATE	TCM1 memory state when domain is RETENTION. 0x0: Memory bank is off when the domain is in the RETENTION state. 0x1: Memory bank is retained when domain is in RETENTION state.	RW	0x1
9	SL2_MEM_RETSTATE	SL2 memory state when domain is RETENTION. 0x0: Memory bank is off when the domain is in the RETENTION state. 0x1: Memory bank is retained when domain is in RETENTION state.	RW	0x1
8	HWA_MEM_RETSTATE	HWA memory state when domain is RETENTION. 0x0: Memory bank is off when the domain is in the RETENTION state.	R	0x0
7:5	RESERVED		R	0x0
4	LOWPOWERSTATECHANGE	Power state change request when domain has already performed a sleep transition. Allows going into deeper low power state without waking up the power domain. 0x0: Do not request a low power state change. 0x1: Request a low power state change. This bit is automatically cleared when the power state is effectively changed or when power state is ON.	RW	0x0
3	RESERVED		R	0x0
2	LOGICRETSTATE	Logic state when power domain is RETENTION 0x0: Whole logic is off when the domain is in RETENTION state.	R	0x0
1:0	POWERSTATE	Power state control 0x0: OFF state 0x1: RETENTION state 0x2: INACTIVE state 0x3: ON State	RW	0x3

Table 3-1447. PM_IVA_PWRSTST

Address Offset	0x0000 0004		
Physical Address	0x4AE0 6F04	**Instance**	IVA_PRM
Description	This register provides a status on the current IVA power domain state. [warm reset insensitive]		
Type	RW		

31 30 29 28 27 26	25 24	23 22 21	20	19 18 17 16	15 14 13 12	11 10	9 8	7 6	5 4	3	2	1 0
RESERVED	LASTPOWERSTATEENTERED	RESERVED	INTRANSITION	RESERVED		TCM2_MEM_STATEST	TCM1_MEM_STATEST	SL2_MEM_STATEST	HWA_MEM_STATEST	RESERVED	LOGICSTATEST	POWERSTATEST

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:24	LASTPOWERSTATEENTERED	Last low power state entered. Set to 0x3 upon write of the same only. This register is intended for debug purpose only. 0x0: Power domain was previously OFF 0x1: Power domain was previously in RETENTION 0x2: Power domain was previously ON-INACTIVE 0x3: Power domain was previously ON-ACTIVE	RW	0x0
23:21	RESERVED		R	0x0
20	INTRANSITION	Domain transition status 0x0: No on-going transition on power domain 0x1: Power domain transition is in progress.	R	0x0
19:12	RESERVED		R	0x0
11:10	TCM2_MEM_STATEST	TCM2 memory state status 0x0: Memory is OFF 0x1: Memory is RETENTION 0x2: Reserved 0x3: Memory is ON	R	0x3
9:8	TCM1_MEM_STATEST	TCM1 memory state status 0x0: Memory is OFF 0x1: Memory is RETENTION 0x2: Reserved 0x3: Memory is ON	R	0x3
7:6	SL2_MEM_STATEST	SL2 memory state status 0x0: Memory is OFF 0x1: Memory is RETENTION 0x2: Reserved 0x3: Memory is ON	R	0x3
5:4	HWA_MEM_STATEST	HWA memory state status 0x0: Memory is OFF 0x1: Memory is RETENTION 0x2: Reserved 0x3: Memory is ON	R	0x3
3	RESERVED		R	0x0
2	LOGICSTATEST	Logic state status 0x0: Logic in domain is OFF 0x1: Logic in domain is ON	R	0x1

Bits	Field Name	Description	Type	Reset
1:0	POWERSTATEST	Current power state status 0x0: Power domain is OFF 0x1: Power domain is in RETENTION 0x2: Power domain is ON-INACTIVE 0x3: Power domain is ON-ACTIVE	R	0x3

Table 3-1448. RM_IVA_RSTCTRL

Address Offset	0x0000 0010		
Physical Address	0x4AE0 6F10	**Instance**	IVA_PRM
Description	This register controls the release of the IVA sub-system resets.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																													RST_LOGIC	RST_SEQ2	RST_SEQ1

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0
2	RST_LOGIC	IVA logic and SL2 reset control 0x0: Reset is cleared for the IVA logic and SL2 0x1: Reset is asserted for IVA logic and SL2	RW	0x1
1	RST_SEQ2	IVA Sequencer2 reset control 0x0: Reset is cleared for IVA Sequencer CPU2 0x1: Reset is asserted for IVA Sequencer CPU2	RW	0x1
0	RST_SEQ1	IVA sequencer1 reset control 0x0: Reset is cleared for the IVA Sequencer CPU1 0x1: Reset is asserted for the IVA sequencer CPU1	RW	0x1

Table 3-1449. RM_IVA_RSTST

Address Offset	0x0000 0014		
Physical Address	0x4AE0 6F14	**Instance**	IVA_PRM
Description	This register logs the different reset sources of the IVA domain. Each bit is set upon release of the domain reset signal. Must be cleared by software. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								RST_ICECRUSHER_SEQ2	RST_ICECRUSHER_SEQ1	RST_EMULATION_SEQ2	RST_EMULATION_SEQ1	RST_LOGIC	RST_SEQ2	RST_SEQ1	

Bits	Field Name	Description	Type	Reset
31:7	RESERVED		R	0x0
6	RST_ICECRUSHER_SEQ2	Sequencer2 CPU has been reset due to IVA ICECRUSHER2 reset event 0x0: No icecrusher reset 0x1: Sequencer2 has been reset upon icecrusher reset	RW	0x0
5	RST_ICECRUSHER_SEQ1	Sequencer1 CPU has been reset due to IVA ICECRUSHER1 reset event 0x0: No icecrusher reset 0x1: Sequencer1 has been reset upon icecrusher reset	RW	0x0
4	RST_EMULATION_SEQ2	Sequencer2 CPU has been reset due to emulation reset source e.g. assert reset command initiated by the icepick module 0x0: No emulation reset 0x1: Sequencer2 has been reset upon emulation reset	RW	0x0
3	RST_EMULATION_SEQ1	Sequencer1 CPU has been reset due to emulation reset source e.g. assert reset command initiated by the icepick module 0x0: No emulation reset 0x1: Sequencer1 has been reset upon emulation reset	RW	0x0
2	RST_LOGIC	IVA logic and SL2 SW reset 0x0: No SW reset occurred 0x1: IVA logic and SL2 has been reset upon SW reset	RW	0x0
1	RST_SEQ2	IVA Sequencer2 CPU SW reset 0x0: No SW reset occurred 0x1: Sequencer2 has been reset upon SW reset	RW	0x0
0	RST_SEQ1	IVA Sequencer1 CPU SW reset 0x0: No SW reset occurred 0x1: Sequencer1 has been reset upon SW reset	RW	0x0

Table 3-1450. RM_IVA_IVA_CONTEXT

Address Offset	0x0000 0024		
Physical Address	0x4AE0 6F24	**Instance**	IVA_PRM
Description	This register contains dedicated IVA context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																					LOSTMEM_HWA_MEM	LOSTMEM_TCM2_MEM	LOSTMEM_TCM1_MEM	RESERVED							LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:11	RESERVED		R	0x0
10	LOSTMEM_HWA_MEM	Specify if memory-based context in HWA_MEM memory bank has been lost due to a previous power transition or other reset source. 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Bits	Field Name	Description	Type	Reset
9	LOSTMEM_TCM2_MEM	Specify if memory-based context in TCM2_MEM memory bank has been lost due to a previous power transition or other reset source. 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
8	LOSTMEM_TCM1_MEM	Specify if memory-based context in TCM1_MEM memory bank has been lost due to a previous power transition or other reset source. 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
7:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of IVA_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1451. RM_IVA_SL2_CONTEXT

Address Offset	0x0000 002C		
Physical Address	0x4AE0 6F2C	**Instance**	IVA_PRM
Description	This register contains dedicated SL2 context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							LOSTMEM_SL2_MEM	RESERVED							LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	LOSTMEM_SL2_MEM	Specify if memory-based context in SL2_MEM memory bank has been lost due to a previous power transition or other reset source. 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
7:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of IVA_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

3.12.49 L3INIT_PRM registers

3.12.49.1 L3INIT_PRM Register Summary

Table 3-1452. L3INIT_PRM Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	L3INIT_PRM Physical Address
PM_L3INIT_PWRSTCTRL	RW	32	0x0000 0000	0x4AE0 7300
PM_L3INIT_PWRSTST	RW	32	0x0000 0004	0x4AE0 7304
RM_PCIESS_RSTCTRL	RW	32	0x0000 0010	0x4AE0 7310
RM_PCIESS_RSTST	RW	32	0x0000 0014	0x4AE0 7314
PM_L3INIT_MMC1_WKDEP	RW	32	0x0000 0028	0x4AE0 7328
RM_L3INIT_MMC1_CONTEXT	RW	32	0x0000 002C	0x4AE0 732C
PM_L3INIT_MMC2_WKDEP	RW	32	0x0000 0030	0x4AE0 7330
RM_L3INIT_MMC2_CONTEXT	RW	32	0x0000 0034	0x4AE0 7334
PM_L3INIT_USB_OTG_SS2_WKDEP	RW	32	0x0000 0040	0x4AE0 7340
RM_L3INIT_USB_OTG_SS2_CONTEXT	RW	32	0x0000 0044	0x4AE0 7344
PM_L3INIT_USB_OTG_SS3_WKDEP	RW	32	0x0000 0048	0x4AE0 7348
RM_L3INIT_USB_OTG_SS3_CONTEXT	RW	32	0x0000 004C	0x4AE0 734C
PM_L3INIT_USB_OTG_SS4_WKDEP	RW	32	0x0000 0050	0x4AE0 7350
RM_L3INIT_USB_OTG_SS4_CONTEXT	RW	32	0x0000 0054	0x4AE0 7354
RM_L3INIT_MLB_SS_CONTEXT	RW	32	0x0000 005C	0x4AE0 735C
RM_L3INIT_IEEE1500_2_OCP_CONTEXT	RW	32	0x0000 007C	0x4AE0 737C
PM_L3INIT_SATA_WKDEP	RW	32	0x0000 0088	0x4AE0 7388
RM_L3INIT_SATA_CONTEXT	RW	32	0x0000 008C	0x4AE0 738C
PM_PCIE_PCIESS1_WKDEP	RW	32	0x0000 00B0	0x4AE0 73B0
RM_PCIE_PCIESS1_CONTEXT	RW	32	0x0000 00B4	0x4AE0 73B4
PM_PCIE_PCIESS2_WKDEP	RW	32	0x0000 00B8	0x4AE0 73B8
RM_PCIE_PCIESS2_CONTEXT	RW	32	0x0000 00BC	0x4AE0 73BC
RM_GMAC_GMAC_CONTEXT	RW	32	0x0000 00D4	0x4AE0 73D4
RM_L3INIT_OCP2SCP1_CONTEXT	RW	32	0x0000 00E4	0x4AE0 73E4
RM_L3INIT_OCP2SCP3_CONTEXT	RW	32	0x0000 00EC	0x4AE0 73EC
PM_L3INIT_USB_OTG_SS1_WKDEP	RW	32	0x0000 00F0	0x4AE0 73F0
RM_L3INIT_USB_OTG_SS1_CONTEXT	RW	32	0x0000 00F4	0x4AE0 73F4

3.12.49.2 L3INIT_PRM Register Description

Table 3-1453. PM_L3INIT_PWRSTCTRL

Address Offset	0x0000 0000		
Physical Address	0x4AE0 7300	**Instance**	L3INIT_PRM
Description	This register controls the L3INIT power state to reach upon a domain sleep transition.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												GMAC_BANK_ONSTATE		L3INIT_BANK2_ONSTATE		L3INIT_BANK1_ONSTATE		RESERVED			GMAC_BANK_RETSTATE	L3INIT_BANK2_RETSTATE	L3INIT_BANK1_RETSTATE	RESERVED			LOWPOWERSTATECHANGE	RESERVED	LOGICRETSTATE	POWERSTATE	

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19:18	GMAC_BANK_ONSTATE	GMAC BANK state when domain is ON. 0x3: Memory bank is on when the domain is ON.	R	0x3
17:16	L3INIT_BANK2_ONSTATE	L3INIT BANK2 state when domain is ON. 0x3: Memory bank is on when the domain is ON.	R	0x3
15:14	L3INIT_BANK1_ONSTATE	L3INIT BANK1 state when domain is ON. 0x3: Memory bank is on when the domain is ON.	R	0x3
13:11	RESERVED		R	0x0
10	GMAC_BANK_RETSTATE	GMAC BANK state when domain is RETENTION. 0x1: Memory bank is retained when domain is in RETENTION state.	R	0x1
9	L3INIT_BANK2_RETSTATE	L3INIT BANK2 state when domain is RETENTION. 0x1: Memory bank is retained when domain is in RETENTION state.	R	0x1
8	L3INIT_BANK1_RETSTATE	L3INIT BANK1 state when domain is RETENTION. 0x0: Memory bank is off when the domain is in the RETENTION state.	R	0x0
7:5	RESERVED		R	0x0
4	LOWPOWERSTATECHANGE	Power state change request when domain has already performed a sleep transition. Allows going into deeper low power state without waking up the power domain. 0x0: Do not request a low power state change. 0x1: Request a low power state change. This bit is automatically cleared when the power state is effectively changed or when power state is ON.	RW	0x0
3	RESERVED		R	0x0
2	LOGICRETSTATE	Logic state when power domain is RETENTION 0x0: Only retention registers are retained and remaing logic is off when the domain is in RETENTION state. 0x1: Whole logic is retained when domain is in RETENTION state.	RW	0x1
1:0	POWERSTATE	Power state control 0x0: OFF state 0x1: RETENTION state 0x2: INACTIVE state 0x3: ON State	RW	0x0

Table 3-1454. Register Call Summary for Register PM_L3INIT_PWRSTCTRL

Power Management Functional Description
PRCM Register Manual • Not Supported Functionality (Registers and Bitfields): [9][10][11][12][13] • L3INIT_PRM Register Summary: [14]

Table 3-1455. PM_L3INIT_PWRSTST

Address Offset	0x0000 0004		
Physical Address	0x4AE0 7304	**Instance**	L3INIT_PRM
Description	This register provides a status on the current L3INIT power domain state. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															
						LASTPOWERSTATEENTERED																									
								RESERVED																							
											INTRANSITION																				
												RESERVED																			
																								L3INIT_GMAC_STATEST							
																										L3INIT_BANK2_STATEST					
																												L3INIT_BANK1_STATEST				
																													RESERVED		
																														LOGICSTATEST	
																															POWERSTATEST

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:24	LASTPOWERSTATEENTERED	Last low power state entered. Set to 0x3 upon write of the same only. This register is intended for debug purpose only. 0x0: Power domain was previously OFF 0x1: Power domain was previously in RETENTION 0x2: Power domain was previously ON-INACTIVE 0x3: Power domain was previously ON-ACTIVE	RW	0x0
23:21	RESERVED		R	0x0
20	INTRANSITION	Domain transition status 0x0: No on-going transition on power domain 0x1: Power domain transition is in progress.	R	0x0
19:10	RESERVED		R	0x0
9:8	L3INIT_GMAC_STATEST	L3INIT GMAC state status 0x0: Memory is OFF 0x1: Reserved 0x2: Reserved 0x3: Memory is ON	R	0x3
7:6	L3INIT_BANK2_STATEST	L3INIT BANK2 state status 0x0: Memory is OFF 0x1: Memory is RETENTION 0x2: Reserved 0x3: Memory is ON	R	0x3

Bits	Field Name	Description	Type	Reset
5:4	L3INIT_BANK1_STATEST	L3INIT BANK1 state status 0x0: Memory is OFF 0x1: Reserved 0x2: Reserved 0x3: Memory is ON	R	0x3
3	RESERVED		R	0x0
2	LOGICSTATEST	Logic state status 0x0: Logic in domain is OFF 0x1: Logic in domain is ON	R	0x1
1:0	POWERSTATEST	Current power state status 0x0: Power domain is OFF 0x1: Power domain is in RETENTION 0x2: Power domain is ON-INACTIVE 0x3: Power domain is ON-ACTIVE	R	0x3

Table 3-1456. Register Call Summary for Register PM_L3INIT_PWRSTST

Power Management Functional Description

PRCM Register Manual

- Not Supported Functionality (Registers and Bitfields): [7][8][9][10][11][12]
- L3INIT_PRM Register Summary: [13]

Table 3-1457. RM_PCIESS_RSTCTRL

Address Offset	0x0000 0010		
Physical Address	0x4AE0 7310	**Instance**	L3INIT_PRM
Description	This register controls the release of the PCIESS local reset.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														RST_LOCAL_PCIE2	RST_LOCAL_PCIE1

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1	RST_LOCAL_PCIE2	PCIESS2 local reset control 0x0: Reset is cleared for the IVA Sequencer CPU1 0x1: Reset is asserted for the IVA sequencer CPU1	RW	0x1
0	RST_LOCAL_PCIE1	PCIESS1 local reset control 0x0: Reset is cleared for the IVA Sequencer CPU1 0x1: Reset is asserted for the IVA sequencer CPU1	RW	0x1

Table 3-1458. RM_PCIESS_RSTST

Address Offset	0x0000 0014		
Physical Address	0x4AE0 7314	**Instance**	L3INIT_PRM
Description	This register logs the different reset sources of the PCIESS domain. Each bit is set upon release of the domain reset signal. Must be cleared by software. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														RST_LOCAL_PCIE2	RST_LOCAL_PCIE1

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1	RST_LOCAL_PCIE2	PCIESS2 local SW reset 0x0: No SW reset occurred 0x1: Sequencer1 has been reset upon SW reset	RW	0x0
0	RST_LOCAL_PCIE1	PCIESS1 local SW reset 0x0: No SW reset occurred 0x1: Sequencer1 has been reset upon SW reset	RW	0x0

Table 3-1459. PM_L3INIT_MMC1_WKDEP

Address Offset	0x0000 0028		
Physical Address	0x4AE0 7328	**Instance**	L3INIT_PRM
Description	This register controls wakeup dependency based on MMC1 service requests.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						WKUPDEP_MMC1_EVE4	WKUPDEP_MMC1_EVE3	WKUPDEP_MMC1_EVE2	WKUPDEP_MMC1_EVE1	WKUPDEP_MMC1_DSP2	WKUPDEP_MMC1_IPU1	WKUPDEP_MMC1_SDMA	WKUPDEP_MMC1_DSP1	WKUPDEP_MMC1_IPU2	WKUPDEP_MMC1_MPU

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	WKUPDEP_MMC1_EVE4	Wakeup dependency from MMC1 module (SWakeup signal) towards EVE4 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
8	WKUPDEP_MMC1_EVE3	Wakeup dependency from MMC1 module (SWakeup signal) towards EVE3 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
7	WKUPDEP_MMC1_EVE2	Wakeup dependency from MMC1 module (SWakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_MMC1_EVE1	Wakeup dependency from MMC1 module (SWakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_MMC1_DSP2	Wakeup dependency from MMC1 module (SWakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_MMC1_IPU1	Wakeup dependency from MMC1 module (SWakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	WKUPDEP_MMC1_SDMA	Wakeup dependency from MMC1 module (SWakeup signal) towards SDMA + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
2	WKUPDEP_MMC1_DSP1	Wakeup dependency from MMC1 module (SWakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_MMC1_IPU2	Wakeup dependency from MMC1 module (SWakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_MMC1_MPU	Wakeup dependency from MMC1 module (SWakeup signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1460. RM_L3INIT_MMC1_CONTEXT

Address Offset	0x0000 002C		
Physical Address	0x4AE0 732C	**Instance**	L3INIT_PRM
Description	This register contains dedicated MMC1 context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							LOSTMEM_L3INIT_BANK1	RESERVED						LOSTCONTEXT_RFF	RESERVED

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	LOSTMEM_L3INIT_BANK1	Specify if memory-based context in L3INIT_BANK1 memory bank has been lost due to a previous power transition or other reset source. 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
7:2	RESERVED		R	0x0
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L3INIT_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
0	RESERVED		R	0x0

Table 3-1461. PM_L3INIT_MMC2_WKDEP

Address Offset	0x0000 0030		
Physical Address	0x4AE0 7330	**Instance**	L3INIT_PRM
Description	This register controls wakeup dependency based on MMC2 service requests.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						WKUPDEP_MMC2_EVE4	WKUPDEP_MMC2_EVE3	WKUPDEP_MMC2_EVE2	WKUPDEP_MMC2_EVE1	WKUPDEP_MMC2_DSP2	WKUPDEP_MMC2_IPU1	WKUPDEP_MMC2_SDMA	WKUPDEP_MMC2_DSP1	WKUPDEP_MMC2_IPU2	WKUPDEP_MMC2_MPU

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	WKUPDEP_MMC2_EVE4	Wakeup dependency from MMC2 module (SWakeup signal) towards EVE4 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
8	WKUPDEP_MMC2_EVE3	Wakeup dependency from MMC2 module (SWakeup signal) towards EVE3 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
7	WKUPDEP_MMC2_EVE2	Wakeup dependency from MMC2 module (SWakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_MMC2_EVE1	Wakeup dependency from MMC2 module (SWakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_MMC2_DSP2	Wakeup dependency from MMC2 module (SWakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_MMC2_IPU1	Wakeup dependency from MMC2 module (SWakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	WKUPDEP_MMC2_SDMA	Wakeup dependency from MMC2 module (SWakeup signal) towards SDMA + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
2	WKUPDEP_MMC2_DSP1	Wakeup dependency from MMC2 module (SWakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_MMC2_IPU2	Wakeup dependency from MMC2 module (SWakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_MMC2_MPU	Wakeup dependency from MMC2 module (SWakeup signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1462. RM_L3INIT_MMC2_CONTEXT

Address Offset	0x0000 0034		
Physical Address	0x4AE0 7334	**Instance**	L3INIT_PRM
Description	This register contains dedicated MMC2 context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							LOSTMEM_L3INIT_BANK1	RESERVED						LOSTCONTEXT_RFF	RESERVED

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	LOSTMEM_L3INIT_BANK1	Specify if memory-based context in L3INIT_BANK1 memory bank has been lost due to a previous power transition or other reset source. 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
7:2	RESERVED		R	0x0
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L3INIT_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
0	RESERVED		R	0x0

Table 3-1463. PM_L3INIT_USB_OTG_SS2_WKDEP

Address Offset	0x0000 0040		
Physical Address	0x4AE0 7340	**Instance**	L3INIT_PRM
Description	This register controls wakeup dependency based on USB_OTG_SS2 service requests.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						WKUPDEP_USB_OTG_SS2_EVE4	WKUPDEP_USB_OTG_SS2_EVE3	WKUPDEP_USB_OTG_SS2_EVE2	WKUPDEP_USB_OTG_SS2_EVE1	WKUPDEP_USB_OTG_SS2_DSP2	WKUPDEP_USB_OTG_SS2_IPU1	RESERVED	WKUPDEP_USB_OTG_SS2_DSP1	WKUPDEP_USB_OTG_SS2_IPU2	WKUPDEP_USB_OTG_SS2_MPU

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	WKUPDEP_USB_OTG_SS2_EVE4	Wakeup dependency from USB2 module (SWakeup signal) towards EVE4 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
8	WKUPDEP_USB_OTG_SS2_EVE3	Wakeup dependency from USB2 module (SWakeup signal) towards EVE3 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
7	WKUPDEP_USB_OTG_SS2_EVE2	Wakeup dependency from USB2 module (SWakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_USB_OTG_SS2_EVE1	Wakeup dependency from USB2 module (SWakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_USB_OTG_SS2_DSP2	Wakeup dependency from USB2 module (SWakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_USB_OTG_SS2_IPU1	Wakeup dependency from USB2 module (SWakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	RESERVED		R	0x0
2	WKUPDEP_USB_OTG_SS2_DSP1	Wakeup dependency from USB2 module (SWakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_USB_OTG_SS2_IPU2	Wakeup dependency from USB2 module (SWakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_USB_OTG_SS2_MPU	Wakeup dependency from USB2 module (SWakeup signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1464. RM_L3INIT_USB_OTG_SS2_CONTEXT

Address Offset	0x0000 0044		
Physical Address	0x4AE0 7344	**Instance**	L3INIT_PRM
Description	This register contains dedicated USB_OTG_SS2 context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							LOSTMEM_L3INIT_BANK1	RESERVED						LOSTCONTEXT_RFF	RESERVED

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	LOSTMEM_L3INIT_BANK1	Specify if memory-based context in L3INIT_BANK1 memory bank has been lost due to a previous power transition or other reset source. 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
7:2	RESERVED		R	0x0
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L3INIT_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
0	RESERVED		R	0x0

Table 3-1465. PM_L3INIT_USB_OTG_SS3_WKDEP

Address Offset	0x0000 0048		
Physical Address	0x4AE0 7348	**Instance**	L3INIT_PRM
Description	This register controls wakeup dependency based on USB_OTG_SS3 service requests.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						WKUPDEP_USB_OTG_SS3_EVE4	WKUPDEP_USB_OTG_SS3_EVE3	WKUPDEP_USB_OTG_SS3_EVE2	WKUPDEP_USB_OTG_SS3_EVE1	WKUPDEP_USB_OTG_SS3_DSP2	WKUPDEP_USB_OTG_SS3_IPU1	RESERVED	WKUPDEP_USB_OTG_SS3_DSP1	WKUPDEP_USB_OTG_SS3_IPU2	WKUPDEP_USB_OTG_SS3_MPU

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	WKUPDEP_USB_OTG_SS3_EVE4	Wakeup dependency from USB3 module (SWakeup signal) towards EVE4 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
8	WKUPDEP_USB_OTG_SS3_EVE3	Wakeup dependency from USB3 module (SWakeup signal) towards EVE3 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
7	WKUPDEP_USB_OTG_SS3_EVE2	Wakeup dependency from USB3 module (SWakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_USB_OTG_SS3_EVE1	Wakeup dependency from USB3 module (SWakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_USB_OTG_SS3_DSP2	Wakeup dependency from USB3 module (SWakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_USB_OTG_SS3_IPU1	Wakeup dependency from USB3 module (SWakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	RESERVED		R	0x0
2	WKUPDEP_USB_OTG_SS3_DSP1	Wakeup dependency from USB3 module (SWakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_USB_OTG_SS3_IPU2	Wakeup dependency from USB3 module (SWakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_USB_OTG_SS3_MPU	Wakeup dependency from USB3 module (SWakeup signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1466. RM_L3INIT_USB_OTG_SS3_CONTEXT

Address Offset	0x0000 004C		
Physical Address	0x4AE0 734C	**Instance**	L3INIT_PRM
Description	This register contains dedicated USB_OTG_SS3 context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							LOSTMEM_L3INIT_BANK1	RESERVED						LOSTCONTEXT_RFF	RESERVED

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	LOSTMEM_L3INIT_BANK1	Specify if memory-based context in L3INIT_BANK1 memory bank has been lost due to a previous power transition or other reset source. 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
7:2	RESERVED		R	0x0
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L3INIT_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
0	RESERVED		R	0x0

Table 3-1467. PM_L3INIT_USB_OTG_SS4_WKDEP

Address Offset	0x0000 0050		
Physical Address	0x4AE0 7350	**Instance**	L3INIT_PRM
Description	This register controls wakeup dependency based on USB_OTG_SS4 service requests.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						WKUPDEP_USB_OTG_SS4_EVE4	WKUPDEP_USB_OTG_SS4_EVE3	WKUPDEP_USB_OTG_SS4_EVE2	WKUPDEP_USB_OTG_SS4_EVE1	WKUPDEP_USB_OTG_SS4_DSP2	WKUPDEP_USB_OTG_SS4_IPU1	RESERVED	WKUPDEP_USB_OTG_SS4_DSP1	WKUPDEP_USB_OTG_SS4_IPU2	WKUPDEP_USB_OTG_SS4_MPU

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	WKUPDEP_USB_OTG_SS4_EVE4	Wakeup dependency from USB4 module (SWakeup signal) towards EVE4 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
8	WKUPDEP_USB_OTG_SS4_EVE3	Wakeup dependency from USB4 module (SWakeup signal) towards EVE3 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
7	WKUPDEP_USB_OTG_SS4_EVE2	Wakeup dependency from USB4 module (SWakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_USB_OTG_SS4_EVE1	Wakeup dependency from USB4 module (SWakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_USB_OTG_SS4_DSP2	Wakeup dependency from USB4 module (SWakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_USB_OTG_SS4_IPU1	Wakeup dependency from USB4 module (SWakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	RESERVED		R	0x0
2	WKUPDEP_USB_OTG_SS4_DSP1	Wakeup dependency from USB4 module (SWakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_USB_OTG_SS4_IPU2	Wakeup dependency from USB4 module (SWakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_USB_OTG_SS4_MPU	Wakeup dependency from USB4 module (SWakeup signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1468. RM_L3INIT_USB_OTG_SS4_CONTEXT

Address Offset	0x0000 0054		
Physical Address	0x4AE0 7354	**Instance**	L3INIT_PRM
Description	This register contains dedicated USB_OTG_SS4 context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							LOSTMEM_L3INIT_BANK1	RESERVED						LOSTCONTEXT_RFF	RESERVED

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	LOSTMEM_L3INIT_BANK1	Specify if memory-based context in L3INIT_BANK1 memory bank has been lost due to a previous power transition or other reset source. 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
7:2	RESERVED		R	0x0
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L3INIT_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
0	RESERVED		R	0x0

Table 3-1469. RM_L3INIT_MLB_SS_CONTEXT

Address Offset	0x0000 005C		
Physical Address	0x4AE0 735C	**Instance**	L3INIT_PRM
Description	This register contains dedicated MLBSS context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							LOSTMEM_MLB_BANK	RESERVED							LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	LOSTMEM_MLB_BANK	Specify if memory-based context in MLB_MEM memory bank has been lost due to a previous power transition or other reset source. 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
7:1	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L3INIT_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1470. RM_L3INIT_IEEE1500_2_OCP_CONTEXT

Address Offset	0x0000 007C		
Physical Address	0x4AE0 737C	**Instance**	L3INIT_PRM
Description	This register contains dedicated IEEE1500_2_OCP context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved	R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L3INIT_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1471. Register Call Summary for Register RM_L3INIT_IEEE1500_2_OCP_CONTEXT

Power Management Functional Description
PRCM Register Manual • L3INIT_PRM Register Summary: [1]

Table 3-1472. PM_L3INIT_SATA_WKDEP

Address Offset	0x0000 0088		
Physical Address	0x4AE0 7388	**Instance**	L3INIT_PRM
Description	This register controls wakeup dependency based on SATA service requests.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						WKUPDEP_SATA_EVE4	WKUPDEP_SATA_EVE3	WKUPDEP_SATA_EVE2	WKUPDEP_SATA_EVE1	WKUPDEP_SATA_DSP2	WKUPDEP_SATA_IPU1	RESERVED	WKUPDEP_SATA_DSP1	WKUPDEP_SATA_IPU2	WKUPDEP_SATA_MPU

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	WKUPDEP_SATA_EVE4	Wakeup dependency from SATA module (SWakeup signal) towards EVE4 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
8	WKUPDEP_SATA_EVE3	Wakeup dependency from SATA module (SWakeup signal) towards EVE3 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
7	WKUPDEP_SATA_EVE2	Wakeup dependency from SATA module (SWakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_SATA_EVE1	Wakeup dependency from SATA module (SWakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_SATA_DSP2	Wakeup dependency from SATA module (SWakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_SATA_IPU1	Wakeup dependency from SATA module (SWakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	RESERVED		R	0x0
2	WKUPDEP_SATA_DSP1	Wakeup dependency from SATA module (SWakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_SATA_IPU2	Wakeup dependency from SATA module (SWakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_SATA_MPU	Wakeup dependency from SATA module (SWakeup signal) towards MPU + L3MAIN1 + L4CFG domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1473. RM_L3INIT_SATA_CONTEXT

Address Offset	0x0000 008C		
Physical Address	0x4AE0 738C	**Instance**	L3INIT_PRM
Description	This register contains dedicated SATA context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							LOSTMEM_L3INIT_BANK1	RESERVED							LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	LOSTMEM_L3INIT_BANK1	Specify if memory-based context in L3INIT_BANK1 memory bank has been lost due to a previous power transition or other reset source. 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
7:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L3INIT_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1474. PM_PCIE_PCIESS1_WKDEP

Address Offset	0x0000 00B0		
Physical Address	0x4AE0 73B0	**Instance**	L3INIT_PRM
Description	This register controls wakeup dependency based on PCIESS1 service requests.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						WKUPDEP_PCIESS1_EVE4	WKUPDEP_PCIESS1_EVE3	WKUPDEP_PCIESS1_EVE2	WKUPDEP_PCIESS1_EVE1	WKUPDEP_PCIESS1_DSP2	WKUPDEP_PCIESS1_IPU1	RESERVED	WKUPDEP_PCIESS1_DSP1	WKUPDEP_PCIESS1_IPU2	WKUPDEP_PCIESS1_MPU

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	WKUPDEP_PCIESS1_EVE4	Wakeup dependency from PCIESS1 module (SWakeup signal) towards EVE4 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
8	WKUPDEP_PCIESS1_EVE3	Wakeup dependency from PCIESS1 module (SWakeup signal) towards EVE3 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
7	WKUPDEP_PCIESS1_EVE2	Wakeup dependency from PCIESS1 module (SWakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_PCIESS1_EVE1	Wakeup dependency from PCIESS1 module (SWakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_PCIESS1_DSP2	Wakeup dependency from PCIESS1 module (SWakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_PCIESS1_IPU1	Wakeup dependency from PCIESS1 module (SWakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	RESERVED		R	0x0
2	WKUPDEP_PCIESS1_DSP1	Wakeup dependency from PCIESS1 module (SWakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_PCIESS1_IPU2	Wakeup dependency from PCIESS1 module (SWakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_PCIESS1_MPU	Wakeup dependency from PCIESS1 module (SWakeup signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1475. RM_PCIE_PCIESS1_CONTEXT

Address Offset	0x0000 00B4		
Physical Address	0x4AE0 73B4	**Instance**	L3INIT_PRM
Description	This register contains dedicated PCIESS1 context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							LOSTMEM_L3INIT_BANK1	RESERVED							LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	LOSTMEM_L3INIT_BANK1	Specify if memory-based context in PCIESS1_BANK memory bank has been lost due to a previous power transition or other reset source. 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
7:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L3INIT_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1476. PM_PCIE_PCIESS2_WKDEP

Address Offset	0x0000 00B8		
Physical Address	0x4AE0 73B8	**Instance**	L3INIT_PRM
Description	This register controls wakeup dependency based on PCIESS2 service requests.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						WKUPDEP_PCIESS2_EVE4	WKUPDEP_PCIESS2_EVE3	WKUPDEP_PCIESS2_EVE2	WKUPDEP_PCIESS2_EVE1	WKUPDEP_PCIESS2_DSP2	WKUPDEP_PCIESS2_IPU1	RESERVED	WKUPDEP_PCIESS2_DSP1	WKUPDEP_PCIESS2_IPU2	WKUPDEP_PCIESS2_MPU

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	WKUPDEP_PCIESS2_EVE4	Wakeup dependency from PCIESS2 module (SWakeup signal) towards EVE4 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
8	WKUPDEP_PCIESS2_EVE3	Wakeup dependency from PCIESS2 module (SWakeup signal) towards EVE3 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
7	WKUPDEP_PCIESS2_EVE2	Wakeup dependency from PCIESS2 module (SWakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_PCIESS2_EVE1	Wakeup dependency from PCIESS2 module (SWakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_PCIESS2_DSP2	Wakeup dependency from PCIESS2 module (SWakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_PCIESS2_IPU1	Wakeup dependency from PCIESS2 module (SWakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	RESERVED		R	0x0
2	WKUPDEP_PCIESS2_DSP1	Wakeup dependency from PCIESS2 module (SWakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_PCIESS2_IPU2	Wakeup dependency from PCIESS2 module (SWakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_PCIESS2_MPU	Wakeup dependency from PCIESS2 module (SWakeup signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1477. RM_PCIE_PCIESS2_CONTEXT

Address Offset	0x0000 00BC		
Physical Address	0x4AE0 73BC	**Instance**	L3INIT_PRM
Description	This register contains dedicated PCIESS2 context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							LOSTMEM_L3INIT_BANK1	RESERVED							LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	LOSTMEM_L3INIT_BANK1	Specify if memory-based context in PCIESS1_BANK memory bank has been lost due to a previous power transition or other reset source. 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
7:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L3INIT_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1478. RM_GMAC_GMAC_CONTEXT

Address Offset	0x0000 00D4		
Physical Address	0x4AE0 73D4	**Instance**	L3INIT_PRM
Description	This register contains dedicated GMAC context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							LOSTMEM_GMAC_BANK	RESERVED							LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	LOSTMEM_GMAC_BANK	Specify if memory-based context in GMAC_BANK memory bank has been lost due to a previous power transition or other reset source. 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
7:1	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L3INIT_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1479. Register Call Summary for Register RM_GMAC_GMAC_CONTEXT

Power Management Functional Description
PRCM Register Manual • L3INIT_PRM Register Summary: [1]

Table 3-1480. RM_L3INIT_OCP2SCP1_CONTEXT

Address Offset	0x0000 00E4		
Physical Address	0x4AE0 73E4	**Instance**	L3INIT_PRM
Description	This register contains dedicated OCP2SCP1 context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L3INIT_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1481. RM_L3INIT_OCP2SCP3_CONTEXT

Address Offset	0x0000 00EC		
Physical Address	0x4AE0 73EC	**Instance**	L3INIT_PRM
Description	This register contains dedicated OCP2SCP3 context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L3INIT_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1482. PM_L3INIT_USB_OTG_SS1_WKDEP

Address Offset	0x0000 00F0		
Physical Address	0x4AE0 73F0	**Instance**	L3INIT_PRM
Description	This register controls wakeup dependency based on USB_OTG_SS1 service requests.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						WKUPDEP_USB_OTG_SS1_EVE4	WKUPDEP_USB_OTG_SS1_EVE3	WKUPDEP_USB_OTG_SS1_EVE2	WKUPDEP_USB_OTG_SS1_EVE1	WKUPDEP_USB_OTG_SS1_DSP2	WKUPDEP_USB_OTG_SS1_IPU1	RESERVED	WKUPDEP_USB_OTG_SS1_DSP1	WKUPDEP_USB_OTG_SS1_IPU2	WKUPDEP_USB_OTG_SS1_MPU

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	WKUPDEP_USB_OTG_SS1_EVE4	Wakeup dependency from USB1 module (SWakeup signal) towards EVE4 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
8	WKUPDEP_USB_OTG_SS1_EVE3	Wakeup dependency from USB1 module (SWakeup signal) towards EVE3 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
7	WKUPDEP_USB_OTG_SS1_EVE2	Wakeup dependency from USB1 module (SWakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_USB_OTG_SS1_EVE1	Wakeup dependency from USB1 module (SWakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_USB_OTG_SS1_DSP2	Wakeup dependency from USB1 module (SWakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
4	WKUPDEP_USB_OTG_SS1_IPU1	Wakeup dependency from USB1 module (SWakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	RESERVED		R	0x0
2	WKUPDEP_USB_OTG_SS1_DSP1	Wakeup dependency from USB1 module (SWakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_USB_OTG_SS1_IPU2	Wakeup dependency from USB1 module (SWakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_USB_OTG_SS1_MPU	Wakeup dependency from USB1 module (SWakeup signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1483. RM_L3INIT_USB_OTG_SS1_CONTEXT

Address Offset	0x0000 00F4		
Physical Address	0x4AE0 73F4	**Instance**	L3INIT_PRM
Description	This register contains dedicated USB_OTG_SS1 context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							LOSTMEM_L3INIT_BANK1	RESERVED						LOSTCONTEXT_RFF	RESERVED

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	LOSTMEM_L3INIT_BANK1	Specify if memory-based context in L3INIT_BANK1 memory bank has been lost due to a previous power transition or other reset source. 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
7:2	RESERVED		R	0x0
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L3INIT_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
0	RESERVED		R	0x0

3.12.50 L4PER_PRM registers

3.12.50.1 L4PER_PRM Register Summary

Table 3-1484. L4PER_PRM Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	L4PER_PRM Physical Address
PM_L4PER_PWRSTCTRL	RW	32	0x0000 0000	0x4AE0 7400
PM_L4PER_PWRSTST	RW	32	0x0000 0004	0x4AE0 7404
RM_L4PER2_L4PER2_CONTEXT	RW	32	0x0000 000C	0x4AE0 740C
RM_L4PER3_L4PER3_CONTEXT	RW	32	0x0000 0014	0x4AE0 7414
RM_L4PER2_PRUSS1_CONTEXT	RW	32	0x0000 001C	0x4AE0 741C
RM_L4PER2_PRUSS2_CONTEXT	RW	32	0x0000 0024	0x4AE0 7424
PM_L4PER_DCC6_WKDEP	RW	32	0x0000 0028	0x4AE0 7428
RM_L4PER_DCC6_CONTEXT	RW	32	0x0000 002C	0x4AE0 742C
PM_L4PER_DCC7_WKDEP	RW	32	0x0000 0030	0x4AE0 7430
RM_L4PER_DCC7_CONTEXT	RW	32	0x0000 0034	0x4AE0 7434
PM_L4PER_TIMER2_WKDEP	RW	32	0x0000 0038	0x4AE0 7438
RM_L4PER_TIMER2_CONTEXT	RW	32	0x0000 003C	0x4AE0 743C
PM_L4PER_TIMER3_WKDEP	RW	32	0x0000 0040	0x4AE0 7440
RM_L4PER_TIMER3_CONTEXT	RW	32	0x0000 0044	0x4AE0 7444
PM_L4PER_TIMER4_WKDEP	RW	32	0x0000 0048	0x4AE0 7448
RM_L4PER_TIMER4_CONTEXT	RW	32	0x0000 004C	0x4AE0 744C
PM_L4PER_DCC5_WKDEP	RW	32	0x0000 0050	0x4AE0 7450
RM_L4PER_DCC5_CONTEXT	RW	32	0x0000 0054	0x4AE0 7454
RM_L4PER_ELM_CONTEXT	RW	32	0x0000 005C	0x4AE0 745C
PM_L4PER_GPIO2_WKDEP	RW	32	0x0000 0060	0x4AE0 7460
RM_L4PER_GPIO2_CONTEXT	RW	32	0x0000 0064	0x4AE0 7464
PM_L4PER_GPIO3_WKDEP	RW	32	0x0000 0068	0x4AE0 7468
RM_L4PER_GPIO3_CONTEXT	RW	32	0x0000 006C	0x4AE0 746C
PM_L4PER_GPIO4_WKDEP	RW	32	0x0000 0070	0x4AE0 7470
RM_L4PER_GPIO4_CONTEXT	RW	32	0x0000 0074	0x4AE0 7474
PM_L4PER_GPIO5_WKDEP	RW	32	0x0000 0078	0x4AE0 7478
RM_L4PER_GPIO5_CONTEXT	RW	32	0x0000 007C	0x4AE0 747C
PM_L4PER_GPIO6_WKDEP	RW	32	0x0000 0080	0x4AE0 7480
RM_L4PER_GPIO6_CONTEXT	RW	32	0x0000 0084	0x4AE0 7484
RM_L4PER_ESM_CONTEXT	RW	32	0x0000 008C	0x4AE0 748C
RM_L4PER2_PWMSS2_CONTEXT	RW	32	0x0000 0094	0x4AE0 7494
RM_L4PER2_PWMSS3_CONTEXT	RW	32	0x0000 009C	0x4AE0 749C
PM_L4PER_I2C1_WKDEP	RW	32	0x0000 00A0	0x4AE0 74A0
RM_L4PER_I2C1_CONTEXT	RW	32	0x0000 00A4	0x4AE0 74A4
PM_L4PER_I2C2_WKDEP	RW	32	0x0000 00A8	0x4AE0 74A8
RM_L4PER_I2C2_CONTEXT	RW	32	0x0000 00AC	0x4AE0 74AC
PM_L4PER_I2C3_WKDEP	RW	32	0x0000 00B0	0x4AE0 74B0
RM_L4PER_I2C3_CONTEXT	RW	32	0x0000 00B4	0x4AE0 74B4
PM_L4PER_I2C4_WKDEP	RW	32	0x0000 00B8	0x4AE0 74B8
RM_L4PER_I2C4_CONTEXT	RW	32	0x0000 00BC	0x4AE0 74BC
RM_L4PER_L4PER1_CONTEXT	RW	32	0x0000 00C0	0x4AE0 74C0
RM_L4PER2_PWMSS1_CONTEXT	RW	32	0x0000 00C4	0x4AE0 74C4
PM_L4PER_DCC1_WKDEP	RW	32	0x0000 00C8	0x4AE0 74C8

Table 3-1484. L4PER_PRM Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	L4PER_PRM Physical Address
RM_L4PER3_DCC1_CONTEXT	RW	32	0x0000 00CC	0x4AE0 74CC
PM_L4PER_DCC2_WKDEP	RW	32	0x0000 00D0	0x4AE0 74D0
RM_L4PER3_DCC2_CONTEXT	RW	32	0x0000 00D4	0x4AE0 74D4
PM_L4PER_DCC3_WKDEP	RW	32	0x0000 00D8	0x4AE0 74D8
RM_L4PER3_DCC3_CONTEXT	RW	32	0x0000 00DC	0x4AE0 74DC
PM_L4PER_MCSPI1_WKDEP	RW	32	0x0000 00F0	0x4AE0 74F0
RM_L4PER_MCSPI1_CONTEXT	RW	32	0x0000 00F4	0x4AE0 74F4
PM_L4PER_MCSPI2_WKDEP	RW	32	0x0000 00F8	0x4AE0 74F8
RM_L4PER_MCSPI2_CONTEXT	RW	32	0x0000 00FC	0x4AE0 74FC
PM_L4PER_MCSPI3_WKDEP	RW	32	0x0000 0100	0x4AE0 7500
RM_L4PER_MCSPI3_CONTEXT	RW	32	0x0000 0104	0x4AE0 7504
PM_L4PER_MCSPI4_WKDEP	RW	32	0x0000 0108	0x4AE0 7508
RM_L4PER_MCSPI4_CONTEXT	RW	32	0x0000 010C	0x4AE0 750C
PM_L4PER_GPIO7_WKDEP	RW	32	0x0000 0110	0x4AE0 7510
RM_L4PER_GPIO7_CONTEXT	RW	32	0x0000 0114	0x4AE0 7514
PM_L4PER_GPIO8_WKDEP	RW	32	0x0000 0118	0x4AE0 7518
RM_L4PER_GPIO8_CONTEXT	RW	32	0x0000 011C	0x4AE0 751C
PM_L4PER_MMC3_WKDEP	RW	32	0x0000 0120	0x4AE0 7520
RM_L4PER_MMC3_CONTEXT	RW	32	0x0000 0124	0x4AE0 7524
PM_L4PER_MMC4_WKDEP	RW	32	0x0000 0128	0x4AE0 7528
RM_L4PER_MMC4_CONTEXT	RW	32	0x0000 012C	0x4AE0 752C
PM_L4PER_DCC4_WKDEP	RW	32	0x0000 0130	0x4AE0 7530
RM_L4PER3_DCC4_CONTEXT	RW	32	0x0000 0134	0x4AE0 7534
PM_L4PER2_QSPI_WKDEP	RW	32	0x0000 0138	0x4AE0 7538
RM_L4PER2_QSPI_CONTEXT	RW	32	0x0000 013C	0x4AE0 753C
PM_L4PER_UART1_WKDEP	RW	32	0x0000 0140	0x4AE0 7540
RM_L4PER_UART1_CONTEXT	RW	32	0x0000 0144	0x4AE0 7544
PM_L4PER_UART2_WKDEP	RW	32	0x0000 0148	0x4AE0 7548
RM_L4PER_UART2_CONTEXT	RW	32	0x0000 014C	0x4AE0 754C
PM_L4PER_UART3_WKDEP	RW	32	0x0000 0150	0x4AE0 7550
RM_L4PER_UART3_CONTEXT	RW	32	0x0000 0154	0x4AE0 7554
PM_L4PER_UART4_WKDEP	RW	32	0x0000 0158	0x4AE0 7558
RM_L4PER_UART4_CONTEXT	RW	32	0x0000 015C	0x4AE0 755C
PM_L4PER2_ADC_WKDEP	RW	32	0x0000 0160	0x4AE0 7560
RM_L4PER2_ADC_CONTEXT	RW	32	0x0000 0164	0x4AE0 7564
PM_L4PER2_MCASP3_WKDEP	RW	32	0x0000 0168	0x4AE0 7568
RM_L4PER2_MCASP3_CONTEXT	RW	32	0x0000 016C	0x4AE0 756C
PM_L4PER_UART5_WKDEP	RW	32	0x0000 0170	0x4AE0 7570
RM_L4PER_UART5_CONTEXT	RW	32	0x0000 0174	0x4AE0 7574
PM_L4PER2_MCASP5_WKDEP	RW	32	0x0000 0178	0x4AE0 7578
RM_L4PER2_MCASP5_CONTEXT	RW	32	0x0000 017C	0x4AE0 757C
PM_L4PER2_MCASP6_WKDEP	RW	32	0x0000 0180	0x4AE0 7580
RM_L4PER2_MCASP6_CONTEXT	RW	32	0x0000 0184	0x4AE0 7584
PM_L4PER2_MCASP7_WKDEP	RW	32	0x0000 0188	0x4AE0 7588
RM_L4PER2_MCASP7_CONTEXT	RW	32	0x0000 018C	0x4AE0 758C
PM_L4PER2_MCASP8_WKDEP	RW	32	0x0000 0190	0x4AE0 7590
RM_L4PER2_MCASP8_CONTEXT	RW	32	0x0000 0194	0x4AE0 7594

Table 3-1484. L4PER_PRM Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	L4PER_PRM Physical Address
PM_L4PER2_MCASP4_WKDEP	RW	32	0x0000 0198	0x4AE0 7598
RM_L4PER2_MCASP4_CONTEXT	RW	32	0x0000 019C	0x4AE0 759C
RM_L4SEC_AES1_CONTEXT	RW	32	0x0000 01A4	0x4AE0 75A4
RM_L4SEC_AES2_CONTEXT	RW	32	0x0000 01AC	0x4AE0 75AC
RM_L4SEC_DES3DES_CONTEXT	RW	32	0x0000 01B4	0x4AE0 75B4
RM_L4SEC_FPKA_CONTEXT	RW	32	0x0000 01BC	0x4AE0 75BC
RM_L4SEC_RNG_CONTEXT	RW	32	0x0000 01C4	0x4AE0 75C4
RM_L4SEC_SHA2MD51_CONTEXT	RW	32	0x0000 01CC	0x4AE0 75CC
PM_L4PER2_UART7_WKDEP	RW	32	0x0000 01D0	0x4AE0 75D0
RM_L4PER2_UART7_CONTEXT	RW	32	0x0000 01D4	0x4AE0 75D4
RM_L4SEC_DMA_CRYPTO_CONTEXT	RW	32	0x0000 01DC	0x4AE0 75DC
PM_L4PER2_UART8_WKDEP	RW	32	0x0000 01E0	0x4AE0 75E0
RM_L4PER2_UART8_CONTEXT	RW	32	0x0000 01E4	0x4AE0 75E4
PM_L4PER2_UART9_WKDEP	RW	32	0x0000 01E8	0x4AE0 75E8
RM_L4PER2_UART9_CONTEXT	RW	32	0x0000 01EC	0x4AE0 75EC
PM_L4PER2_DCAN2_WKDEP	RW	32	0x0000 01F0	0x4AE0 75F0
RM_L4PER2_DCAN2_CONTEXT	RW	32	0x0000 01F4	0x4AE0 75F4
RM_L4SEC_SHA2MD52_CONTEXT	RW	32	0x0000 01FC	0x4AE0 75FC

3.12.50.2 L4PER_PRM Register Description

Table 3-1485. PM_L4PER_PWRSTCTRL

Address Offset	0x0000 0000		
Physical Address	0x4AE0 7400	**Instance**	L4PER_PRM
Description	This register controls the L4PER power state to reach upon a domain sleep transition		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												NONRETAINED_BANK_ONSTATE		RETAINED_BANK_ONSTATE		RESERVED							NONRETAINED_BANK_RETSTATE	RETAINED_BANK_RETSTATE	RESERVED			LOWPOWERSTATECHANGE	RESERVED	LOGICRETSTATE	POWERSTATE	

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19:18	NONRETAINED_BANK_ONSTATE	NONRETAINED_BANK state when domain is ON. 0x3: Memory bank is on when the domain is ON.	R	0x3
17:16	RETAINED_BANK_ONSTATE	RETAINED_BANK state when domain is ON. 0x3: Memory bank is on when the domain is ON.	R	0x3

Bits	Field Name	Description	Type	Reset
15:10	RESERVED		R	0x0
9	NONRETAINED_BANK_RETSTATE	NONRETAINED_BANK state when domain is RETENTION. 0x0: Memory bank is off when the domain is in the RETENTION state.	R	0x0
8	RETAINED_BANK_RETSTATE	RETAINED_BANK state when domain is RETENTION. 0x1: Memory bank is retained when domain is in RETENTION state.	R	0x1
7:5	RESERVED		R	0x0
4	LOWPOWERSTATECHANGE	Power state change request when domain has already performed a sleep transition. Allows going into deeper low power state without waking up the power domain. 0x0: Do not request a low power state change. 0x1: Request a low power state change. This bit is automatically cleared when the power state is effectively changed or when power state is ON.	RW	0x0
3	RESERVED		R	0x0
2	LOGICRETSTATE	Logic state when power domain is RETENTION 0x0: Only retention registers are retained and remaing logic is off when the domain is in RETENTION state. 0x1: Whole logic is retained when domain is in RETENTION state.	RW	0x1
1:0	POWERSTATE	Power state control 0x0: Reserved 0x1: RETENTION state 0x2: INACTIVE state 0x3: ON State	RW	0x3

Table 3-1486. Register Call Summary for Register PM_L4PER_PWRSTCTRL

Power Management Functional Description
PRCM Register Manual • Not Supported Functionality (Registers and Bitfields): [7][8][9][10][11] • L4PER_PRM Register Summary: [12]

Table 3-1487. PM_L4PER_PWRSTST

Address Offset	0x0000 0004		
Physical Address	0x4AE0 7404	**Instance**	L4PER_PRM
Description	This register provides a status on the current L4PER power domain state. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED							LASTPOWERSTATEENTERED	RESERVED			INTRANSITION	RESERVED												NONRETAINED_BANK_STATEST		RETAINED_BANK_STATEST		RESERVED	LOGICSTATEST		POWERSTATEST	

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:24	LASTPOWERSTATEENTERED	Last low power state entered. Set to 0x3 upon write of the same only. This register is intended for debug purpose only. 0x0: Power domain was previously OFF 0x1: Power domain was previously in RETENTION 0x2: Power domain was previously ON-INACTIVE 0x3: Power domain was previously ON-ACTIVE	RW	0x0
23:21	RESERVED		R	0x0
20	INTRANSITION	Domain transition status 0x0: No on-going transition on power domain 0x1: Power domain transition is in progress.	R	0x0
19:8	RESERVED		R	0x0
7:6	NONRETAINED_BANK_STATEST	NONRETAINED_BANK state status 0x0: Memory is OFF 0x1: Reserved 0x2: Reserved 0x3: Memory is ON	R	0x3
5:4	RETAINED_BANK_STATEST	RETAINED_BANK state status 0x0: Memory is OFF 0x1: Memory is RETENTION 0x2: Reserved 0x3: Memory is ON	R	0x3
3	RESERVED		R	0x0
2	LOGICSTATEST	Logic state status 0x0: Logic in domain is OFF 0x1: Logic in domain is ON	R	0x1
1:0	POWERSTATEST	Current power state status 0x0: Reserved 0x1: Power domain is in RETENTION 0x2: Power domain is ON-INACTIVE 0x3: Power domain is ON-ACTIVE	R	0x3

Table 3-1488. Register Call Summary for Register PM_L4PER_PWRSTST

Power Management Functional Description
PRCM Register Manual • Not Supported Functionality (Registers and Bitfields): [6][7][8][9][10][11] • L4PER_PRM Register Summary: [12]

Table 3-1489. RM_L4PER2_L4PER2_CONTEXT

Address Offset	0x0000 000C		
Physical Address	0x4AE0 740C	**Instance**	L4PER_PRM
Description	This register contains dedicated L4_PER2 context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														LOSTCONTEXT_RFF	LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4PER_PWRON_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4PER_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1490. Register Call Summary for Register RM_L4PER2_L4PER2_CONTEXT

Power Management Functional Description
PRCM Register Manual • L4PER_PRM Register Summary: [2]

Table 3-1491. RM_L4PER3_L4PER3_CONTEXT

Address Offset	0x0000 0014		
Physical Address	0x4AE0 7414	**Instance**	L4PER_PRM
Description	This register contains dedicated L4_PER3 context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														LOSTCONTEXT_RFF	LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4PER_PWRON_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Bits	Field Name	Description	Type	Reset
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4PER_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1492. Register Call Summary for Register RM_L4PER3_L4PER3_CONTEXT

Power Management Functional Description
PRCM Register Manual • L4PER_PRM Register Summary: [2]

Table 3-1493. RM_L4PER2_PRUSS1_CONTEXT

Address Offset	0x0000 001C		
Physical Address	0x4AE0 741C	**Instance**	L4PER_PRM
Description	This register contains dedicated PRUSS0 context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							LOSTMEM_PRUSS1_BANK	RESERVED							LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	LOSTMEM_PRUSS1_BANK	Specify if memory-based context in pruss memory bank has been lost due to a previous power transition or other reset source. 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
7:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L3INIT_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1494. RM_L4PER2_PRUSS2_CONTEXT

Address Offset	0x0000 0024		
Physical Address	0x4AE0 7424	**Instance**	L4PER_PRM
Description	This register contains dedicated PRUSS1 context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							LOSTMEM_PRUSS2_BANK	RESERVED							LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	LOSTMEM_PRUSS2_BANK	Specify if memory-based context in pruss memory bank has been lost due to a previous power transition or other reset source. 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
7:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L3INIT_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1495. PM_L4PER_DCC6_WKDEP

Address Offset	0x0000 0028		
Physical Address	0x4AE0 7428	**Instance**	L4PER_PRM
Description	This register controls wakeup dependency based on DCC6 service requests.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						WKUPDEP_DCC6_EVE4	WKUPDEP_DCC6_EVE3	WKUPDEP_DCC6_EVE2	WKUPDEP_DCC6_EVE1	WKUPDEP_DCC6_DSP2	WKUPDEP_DCC6_IPU1	RESERVED	WKUPDEP_DCC6_DSP1	WKUPDEP_DCC6_IPU2	WKUPDEP_DCC6_MPU

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	WKUPDEP_DCC6_EVE4	Wakeup dependency from DCC6 module (SWakeup IRQ signal) towards EVE4 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
8	WKUPDEP_DCC6_EVE3	Wakeup dependency from DCC6 module (SWakeup IRQ signal) towards EVE3 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
7	WKUPDEP_DCC6_EVE2	Wakeup dependency from DCC6 module (SWakeup IRQ signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_DCC6_EVE1	Wakeup dependency from DCC6 module (SWakeup IRQ signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_DCC6_DSP2	Wakeup dependency from DCC6 module (SWakeup IRQ signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_DCC6_IPU1	Wakeup dependency from DCC6 module (SWakeup IRQ signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	RESERVED		R	0x0
2	WKUPDEP_DCC6_DSP1	Wakeup dependency from DCC6 module (SWakeup IRQ signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_DCC6_IPU2	Wakeup dependency from DCC6 module (SWakeup IRQ signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_DCC6_MPU	Wakeup dependency from DCC6 module (SWakeup IRQ signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1496. RM_L4PER_DCC6_CONTEXT

Address Offset	0x0000 002C		
Physical Address	0x4AE0 742C	**Instance**	L4PER_PRM
Description	This register contains dedicated DCC6 context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4PER_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1497. Register Call Summary for Register RM_L4PER_DCC6_CONTEXT

Power Management Functional Description
PRCM Register Manual • L4PER_PRM Register Summary: [1]

Table 3-1498. PM_L4PER_DCC7_WKDEP

Address Offset	0x0000 0030		
Physical Address	0x4AE0 7430	**Instance**	L4PER_PRM
Description	This register controls wakeup dependency based on DCC7 service requests.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						WKUPDEP_DCC7_EVE4	WKUPDEP_DCC7_EVE3	WKUPDEP_DCC7_EVE2	WKUPDEP_DCC7_EVE1	WKUPDEP_DCC7_DSP2	WKUPDEP_DCC7_IPU1	RESERVED	WKUPDEP_DCC7_DSP1	WKUPDEP_DCC7_IPU2	WKUPDEP_DCC7_MPU

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	WKUPDEP_DCC7_EVE4	Wakeup dependency from DCC7 module (SWakeup IRQ signal) towards EVE4 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
8	WKUPDEP_DCC7_EVE3	Wakeup dependency from DCC7 module (SWakeup IRQ signal) towards EVE3 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
7	WKUPDEP_DCC7_EVE2	Wakeup dependency from DCC7 module (SWakeup IRQ signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_DCC7_EVE1	Wakeup dependency from DCC7 module (SWakeup IRQ signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
5	WKUPDEP_DCC7_DSP2	Wakeup dependency from DCC7 module (SWakeup IRQ signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_DCC7_IPU1	Wakeup dependency from DCC7 module (SWakeup IRQ signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	RESERVED		R	0x0
2	WKUPDEP_DCC7_DSP1	Wakeup dependency from DCC7 module (SWakeup IRQ signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_DCC7_IPU2	Wakeup dependency from DCC7 module (SWakeup IRQ signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_DCC7_MPU	Wakeup dependency from DCC7 module (SWakeup IRQ signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1499. RM_L4PER_DCC7_CONTEXT

Address Offset	0x0000 0034		
Physical Address	0x4AE0 7434	**Instance**	L4PER_PRM
Description	This register contains dedicated DCC7 context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4PER_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1500. Register Call Summary for Register RM_L4PER_DCC7_CONTEXT

Power Management Functional Description
PRCM Register Manual • L4PER_PRM Register Summary: [1]

Table 3-1501. PM_L4PER_TIMER2_WKDEP

Address Offset	0x0000 0038		
Physical Address	0x4AE0 7438	**Instance**	L4PER_PRM
Description	This register controls wakeup dependency based on TIMER2 service requests.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						WKUPDEP_TIMER2_EVE4	WKUPDEP_TIMER2_EVE3	WKUPDEP_TIMER2_EVE2	WKUPDEP_TIMER2_EVE1	WKUPDEP_TIMER2_DSP2	WKUPDEP_TIMER2_IPU1	RESERVED	WKUPDEP_TIMER2_DSP1	WKUPDEP_TIMER2_IPU2	WKUPDEP_TIMER2_MPU

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	WKUPDEP_TIMER2_EVE4	Wakeup dependency from TIMER2 module (SWakeup IRQ signal) towards EVE4 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
8	WKUPDEP_TIMER2_EVE3	Wakeup dependency from TIMER2 module (SWakeup IRQ signal) towards EVE3 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
7	WKUPDEP_TIMER2_EVE2	Wakeup dependency from TIMER2 module (SWakeup IRQ signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_TIMER2_EVE1	Wakeup dependency from TIMER2 module (SWakeup IRQ signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_TIMER2_DSP2	Wakeup dependency from TIMER2 module (SWakeup IRQ signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
4	WKUPDEP_TIMER2_IPU1	Wakeup dependency from TIMER2 module (SWakeup IRQ signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	RESERVED		R	0x0
2	WKUPDEP_TIMER2_DSP1	Wakeup dependency from TIMER2 module (SWakeup IRQ signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_TIMER2_IPU2	Wakeup dependency from TIMER2 module (SWakeup IRQ signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_TIMER2_MPU	Wakeup dependency from TIMER2 module (SWakeup IRQ signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1502. Register Call Summary for Register PM_L4PER_TIMER2_WKDEP

Clock Management Functional Description
- Clock Domain Dependency: [0][1][2][3]

PRCM Register Manual
- Not Supported Functionality (Registers and Bitfields): [4][5][6][7][8]
- L4PER_PRM Register Summary: [9]

Table 3-1503. RM_L4PER_TIMER2_CONTEXT

Address Offset	0x0000 003C		
Physical Address	0x4AE0 743C	**Instance**	L4PER_PRM
Description	This register contains dedicated TIMER2 context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4PER_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1504. Register Call Summary for Register RM_L4PER_TIMER2_CONTEXT

Power Management Functional Description
PRCM Register Manual • L4PER_PRM Register Summary: [1]

Table 3-1505. PM_L4PER_TIMER3_WKDEP

Address Offset	0x0000 0040		
Physical Address	0x4AE0 7440	**Instance**	L4PER_PRM
Description	This register controls wakeup dependency based on TIMER3 service requests.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						WKUPDEP_TIMER3_EVE4	WKUPDEP_TIMER3_EVE3	WKUPDEP_TIMER3_EVE2	WKUPDEP_TIMER3_EVE1	WKUPDEP_TIMER3_DSP2	WKUPDEP_TIMER3_IPU1	RESERVED	WKUPDEP_TIMER3_DSP1	WKUPDEP_TIMER3_IPU2	WKUPDEP_TIMER3_MPU

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	WKUPDEP_TIMER3_EVE4	Wakeup dependency from TIMER3 module (SWakeup IRQ signal) towards EVE4 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
8	WKUPDEP_TIMER3_EVE3	Wakeup dependency from TIMER3 module (SWakeup IRQ signal) towards EVE3 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
7	WKUPDEP_TIMER3_EVE2	Wakeup dependency from TIMER3 module (SWakeup IRQ signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_TIMER3_EVE1	Wakeup dependency from TIMER3 module (SWakeup IRQ signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_TIMER3_DSP2	Wakeup dependency from TIMER3 module (SWakeup IRQ signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
4	WKUPDEP_TIMER3_IPU1	Wakeup dependency from TIMER3 module (SWakeup IRQ signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	RESERVED		R	0x0
2	WKUPDEP_TIMER3_DSP1	Wakeup dependency from TIMER3 module (SWakeup IRQ signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_TIMER3_IPU2	Wakeup dependency from TIMER3 module (SWakeup IRQ signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_TIMER3_MPU	Wakeup dependency from TIMER3 module (SWakeup IRQ signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1506. Register Call Summary for Register PM_L4PER_TIMER3_WKDEP

Clock Management Functional Description
- Clock Domain Dependency: [0][1][2][3]

PRCM Register Manual
- Not Supported Functionality (Registers and Bitfields): [4][5][6][7][8]
- L4PER_PRM Register Summary: [9]

Table 3-1507. RM_L4PER_TIMER3_CONTEXT

Address Offset	0x0000 0044		
Physical Address	0x4AE0 7444	**Instance**	L4PER_PRM
Description	This register contains dedicated TIMER3 context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4PER_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1508. Register Call Summary for Register RM_L4PER_TIMER3_CONTEXT

Power Management Functional Description
PRCM Register Manual • L4PER_PRM Register Summary: [1]

Table 3-1509. PM_L4PER_TIMER4_WKDEP

Address Offset	0x0000 0048		
Physical Address	0x4AE0 7448	**Instance**	L4PER_PRM
Description	This register controls wakeup dependency based on TIMER4 service requests.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						WKUPDEP_TIMER4_EVE4	WKUPDEP_TIMER4_EVE3	WKUPDEP_TIMER4_EVE2	WKUPDEP_TIMER4_EVE1	WKUPDEP_TIMER4_DSP2	WKUPDEP_TIMER4_IPU1	RESERVED	WKUPDEP_TIMER4_DSP1	WKUPDEP_TIMER4_IPU2	WKUPDEP_TIMER4_MPU

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	WKUPDEP_TIMER4_EVE4	Wakeup dependency from TIMER4 module (SWakeup IRQ signal) towards EVE4 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
8	WKUPDEP_TIMER4_EVE3	Wakeup dependency from TIMER4 module (SWakeup IRQ signal) towards EVE3 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
7	WKUPDEP_TIMER4_EVE2	Wakeup dependency from TIMER4 module (SWakeup IRQ signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_TIMER4_EVE1	Wakeup dependency from TIMER4 module (SWakeup IRQ signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_TIMER4_DSP2	Wakeup dependency from TIMER4 module (SWakeup IRQ signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
4	WKUPDEP_TIMER4_IPU1	Wakeup dependency from TIMER4 module (SWakeup IRQ signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	RESERVED		R	0x0
2	WKUPDEP_TIMER4_DSP1	Wakeup dependency from TIMER4 module (SWakeup IRQ signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_TIMER4_IPU2	Wakeup dependency from TIMER4 module (SWakeup IRQ signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_TIMER4_MPU	Wakeup dependency from TIMER4 module (SWakeup IRQ signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1510. Register Call Summary for Register PM_L4PER_TIMER4_WKDEP

Clock Management Functional Description
- Clock Domain Dependency: [0][1][2][3]

PRCM Register Manual
- Not Supported Functionality (Registers and Bitfields): [4][5][6][7][8]
- L4PER_PRM Register Summary: [9]

Table 3-1511. RM_L4PER_TIMER4_CONTEXT

Address Offset	0x0000 004C		
Physical Address	0x4AE0 744C	**Instance**	L4PER_PRM
Description	This register contains dedicated TIMER4 context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4PER_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1512. Register Call Summary for Register RM_L4PER_TIMER4_CONTEXT

Power Management Functional Description
PRCM Register Manual • L4PER_PRM Register Summary: [1]

Table 3-1513. PM_L4PER_DCC5_WKDEP

Address Offset	0x0000 0050		
Physical Address	0x4AE0 7450	**Instance**	L4PER_PRM
Description	This register controls wakeup dependency based on DCC5 service requests.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						WKUPDEP_DCC5_EVE4	WKUPDEP_DCC5_EVE3	WKUPDEP_DCC5_EVE2	WKUPDEP_DCC5_EVE1	WKUPDEP_DCC5_DSP2	WKUPDEP_DCC5_IPU1	RESERVED	WKUPDEP_DCC5_DSP1	WKUPDEP_DCC5_IPU2	WKUPDEP_DCC5_MPU

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	WKUPDEP_DCC5_EVE4	Wakeup dependency from DCC5 module (SWakeup IRQ signal) towards EVE4 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
8	WKUPDEP_DCC5_EVE3	Wakeup dependency from TIMER4 module (SWakeup IRQ signal) towards EVE3 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
7	WKUPDEP_DCC5_EVE2	Wakeup dependency from DCC5 module (SWakeup IRQ signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_DCC5_EVE1	Wakeup dependency from DCC5 module (SWakeup IRQ signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_DCC5_DSP2	Wakeup dependency from TIMER4 module (SWakeup IRQ signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_DCC5_IPU1	Wakeup dependency from DCC5 module (SWakeup IRQ signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
3	RESERVED		R	0x0
2	WKUPDEP_DCC5_DSP1	Wakeup dependency from TIMER4 module (SWakeup IRQ signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_DCC5_IPU2	Wakeup dependency from TIMER4 module (SWakeup IRQ signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_DCC5_MPU	Wakeup dependency from TIMER4 module (SWakeup IRQ signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1514. RM_L4PER_DCC5_CONTEXT

Address Offset	0x0000 0054		
Physical Address	0x4AE0 7454	**Instance**	L4PER_PRM
Description	This register contains dedicated DCC5 context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4PER_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1515. Register Call Summary for Register RM_L4PER_DCC5_CONTEXT

Power Management Functional Description
PRCM Register Manual • L4PER_PRM Register Summary: [1]

Table 3-1516. RM_L4PER_ELM_CONTEXT

Address Offset	0x0000 005C		
Physical Address	0x4AE0 745C	**Instance**	L4PER_PRM
Description	This register contains dedicated ELM context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4PER_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1517. Register Call Summary for Register RM_L4PER_ELM_CONTEXT

Power Management Functional Description
PRCM Register Manual • L4PER_PRM Register Summary: [1]

Table 3-1518. PM_L4PER_GPIO2_WKDEP

Address Offset	0x0000 0060		
Physical Address	0x4AE0 7460	**Instance**	L4PER_PRM
Description	This register controls wakeup dependency based on GPIO2 service requests.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												WKUPDEP_GPIO2_IRQ2_EVE4	WKUPDEP_GPIO2_IRQ2_EVE3	WKUPDEP_GPIO2_IRQ2_EVE2	WKUPDEP_GPIO2_IRQ2_EVE1	WKUPDEP_GPIO2_IRQ2_DSP2	WKUPDEP_GPIO2_IRQ2_IPU1	RESERVED	WKUPDEP_GPIO2_IRQ2_DSP1	WKUPDEP_GPIO2_IRQ2_IPU2	WKUPDEP_GPIO2_IRQ2_MPU	WKUPDEP_GPIO2_IRQ1_EVE4	WKUPDEP_GPIO2_IRQ1_EVE3	WKUPDEP_GPIO2_IRQ1_EVE2	WKUPDEP_GPIO2_IRQ1_EVE1	WKUPDEP_GPIO2_IRQ1_DSP2	WKUPDEP_GPIO2_IRQ1_IPU1	RESERVED	WKUPDEP_GPIO2_IRQ1_DSP1	WKUPDEP_GPIO2_IRQ1_IPU2	WKUPDEP_GPIO2_IRQ1_MPU

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	WKUPDEP_GPIO2_IRQ2_EVE4	Wakeup dependency from GPIO2 module (SWakeup signal) towards EVE4 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
18	WKUPDEP_GPIO2_IRQ2_EVE3	Wakeup dependency from GPIO2 module (SWakeup signal) towards EVE3 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
17	WKUPDEP_GPIO2_IRQ2_EVE2	Wakeup dependency from GPIO2 module (SWakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
16	WKUPDEP_GPIO2_IRQ2_EVE1	Wakeup dependency from GPIO2 module (SWakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
15	WKUPDEP_GPIO2_IRQ2_DSP2	Wakeup dependency from GPIO2 module (SWakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
14	WKUPDEP_GPIO2_IRQ2_IPU1	Wakeup dependency from GPIO2 module (SWakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
13	RESERVED		R	0x0
12	WKUPDEP_GPIO2_IRQ2_DSP1	Wakeup dependency from GPIO2 module (SWakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
11	WKUPDEP_GPIO2_IRQ2_IPU2	Wakeup dependency from GPIO2 module (SWakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
10	WKUPDEP_GPIO2_IRQ2_MPU	Wakeup dependency from GPIO2 module (SWakeup signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
9	WKUPDEP_GPIO2_IRQ1_EVE4	Wakeup dependency from GPIO2 module (SWakeup signal) towards EVE4 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
8	WKUPDEP_GPIO2_IRQ1_EVE3	Wakeup dependency from GPIO2 module (SWakeup signal) towards EVE3 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
7	WKUPDEP_GPIO2_IRQ1_EVE2	Wakeup dependency from GPIO2 module (SWakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
6	WKUPDEP_GPIO2_IRQ1_EVE1	Wakeup dependency from GPIO2 module (SWakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_GPIO2_IRQ1_DSP2	Wakeup dependency from GPIO2 module (SWakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_GPIO2_IRQ1_IPU1	Wakeup dependency from GPIO2 module (SWakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	RESERVED		R	0x0
2	WKUPDEP_GPIO2_IRQ1_DSP1	Wakeup dependency from GPIO2 module (SWakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_GPIO2_IRQ1_IPU2	Wakeup dependency from GPIO2 module (SWakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_GPIO2_IRQ1_MPU	Wakeup dependency from GPIO2 module (SWakeup signal for POROCPSINTERRUPT1) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1519. Register Call Summary for Register PM_L4PER_GPIO2_WKDEP

Clock Management Functional Description • Clock Domain Dependency: [0][1][2][3][4][5][6][7]
PRCM Register Manual • Not Supported Functionality (Registers and Bitfields): [8][9][10][11][12][13][14][15][16][17] • L4PER_PRM Register Summary: [18]

Table 3-1520. RM_L4PER_GPIO2_CONTEXT

Address Offset	0x0000 0064		
Physical Address	0x4AE0 7464	**Instance**	L4PER_PRM
Description	This register contains dedicated GPIO2 context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														LOSTCONTEXT_RFF	RESERVED

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4PER_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
0	RESERVED		R	0x0

Table 3-1521. Register Call Summary for Register RM_L4PER_GPIO2_CONTEXT

Power Management Functional Description
PRCM Register Manual • L4PER_PRM Register Summary: [1]

Table 3-1522. PM_L4PER_GPIO3_WKDEP

Address Offset	0x0000 0068		
Physical Address	0x4AE0 7468	**Instance**	L4PER_PRM
Description	This register controls wakeup dependency based on GPIO3 service requests.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												WKUPDEP_GPIO3_IRQ2_EVE4	WKUPDEP_GPIO3_IRQ2_EVE3	WKUPDEP_GPIO3_IRQ2_EVE2	WKUPDEP_GPIO3_IRQ2_EVE1	WKUPDEP_GPIO3_IRQ2_DSP2	WKUPDEP_GPIO3_IRQ2_IPU1	RESERVED	WKUPDEP_GPIO3_IRQ2_DSP1	WKUPDEP_GPIO3_IRQ2_IPU2	WKUPDEP_GPIO3_IRQ2_MPU	WKUPDEP_GPIO3_IRQ1_EVE4	WKUPDEP_GPIO3_IRQ1_EVE3	WKUPDEP_GPIO3_IRQ1_EVE2	WKUPDEP_GPIO3_IRQ1_EVE1	WKUPDEP_GPIO3_IRQ1_DSP2	WKUPDEP_GPIO3_IRQ1_IPU1	RESERVED	WKUPDEP_GPIO3_IRQ1_DSP1	WKUPDEP_GPIO3_IRQ1_IPU2	WKUPDEP_GPIO3_IRQ1_MPU

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	WKUPDEP_GPIO3_IRQ2_EVE4	Wakeup dependency from GPIO3 module (SWakeup signal) towards EVE4 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
18	WKUPDEP_GPIO3_IRQ2_EVE3	Wakeup dependency from GPIO3 module (SWakeup signal) towards EVE3 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
17	WKUPDEP_GPIO3_IRQ2_EVE2	Wakeup dependency from GPIO3 module (SWakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
16	WKUPDEP_GPIO3_IRQ2_EVE1	Wakeup dependency from GPIO3 module (SWakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
15	WKUPDEP_GPIO3_IRQ2_DSP2	Wakeup dependency from GPIO3 module (SWakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
14	WKUPDEP_GPIO3_IRQ2_IPU1	Wakeup dependency from GPIO3 module (SWakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
13	RESERVED		R	0x0
12	WKUPDEP_GPIO3_IRQ2_DSP1	Wakeup dependency from GPIO3 module (SWakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
11	WKUPDEP_GPIO3_IRQ2_IPU2	Wakeup dependency from GPIO3 module (SWakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
10	WKUPDEP_GPIO3_IRQ2_MPU	Wakeup dependency from GPIO3 module (SWakeup signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
9	WKUPDEP_GPIO3_IRQ1_EVE4	Wakeup dependency from GPIO3 module (SWakeup signal) towards EVE4 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
8	WKUPDEP_GPIO3_IRQ1_EVE3	Wakeup dependency from GPIO3 module (SWakeup signal) towards EVE3 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
7	WKUPDEP_GPIO3_IRQ1_EVE2	Wakeup dependency from GPIO3 module (SWakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
6	WKUPDEP_GPIO3_IRQ1_EVE1	Wakeup dependency from GPIO3 module (SWakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_GPIO3_IRQ1_DSP2	Wakeup dependency from GPIO3 module (SWakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_GPIO3_IRQ1_IPU1	Wakeup dependency from GPIO3 module (SWakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	RESERVED		R	0x0
2	WKUPDEP_GPIO3_IRQ1_DSP1	Wakeup dependency from GPIO3 module (SWakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_GPIO3_IRQ1_IPU2	3Wakeup dependency from GPIO2 module (SWakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_GPIO3_IRQ1_MPU	Wakeup dependency from GPIO3 module (SWakeup signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1523. Register Call Summary for Register PM_L4PER_GPIO3_WKDEP

Clock Management Functional Description • Clock Domain Dependency: [0][1][2][3][4][5][6][7]
PRCM Register Manual • Not Supported Functionality (Registers and Bitfields): [8][9][10][11][12][13][14][15][16][17] • L4PER_PRM Register Summary: [18]

Table 3-1524. RM_L4PER_GPIO3_CONTEXT

Address Offset	0x0000 006C		
Physical Address	0x4AE0 746C	**Instance**	L4PER_PRM
Description	This register contains dedicated GPIO3 context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														LOSTCONTEXT_RFF	RESERVED

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4PER_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
0	RESERVED		R	0x0

Table 3-1525. Register Call Summary for Register RM_L4PER_GPIO3_CONTEXT

Power Management Functional Description
PRCM Register Manual • L4PER_PRM Register Summary: [1]

Table 3-1526. PM_L4PER_GPIO4_WKDEP

Address Offset	0x0000 0070		
Physical Address	0x4AE0 7470	**Instance**	L4PER_PRM
Description	This register controls wakeup dependency based on GPIO4 service requests.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												WKUPDEP_GPIO4_IRQ2_EVE4	WKUPDEP_GPIO4_IRQ2_EVE3	WKUPDEP_GPIO4_IRQ2_EVE2	WKUPDEP_GPIO4_IRQ2_EVE1	WKUPDEP_GPIO4_IRQ2_DSP2	WKUPDEP_GPIO4_IRQ2_IPU1	RESERVED	WKUPDEP_GPIO4_IRQ2_DSP1	WKUPDEP_GPIO4_IRQ2_IPU2	WKUPDEP_GPIO4_IRQ2_MPU	WKUPDEP_GPIO4_IRQ1_EVE4	WKUPDEP_GPIO4_IRQ1_EVE3	WKUPDEP_GPIO4_IRQ1_EVE2	WKUPDEP_GPIO4_IRQ1_EVE1	WKUPDEP_GPIO4_IRQ1_DSP2	WKUPDEP_GPIO4_IRQ1_IPU1	RESERVED	WKUPDEP_GPIO4_IRQ1_DSP1	WKUPDEP_GPIO4_IRQ1_IPU2	WKUPDEP_GPIO4_IRQ1_MPU

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	WKUPDEP_GPIO4_IRQ2_EVE4	Wakeup dependency from GPIO4 module (SWakeup signal) towards EVE4 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
18	WKUPDEP_GPIO4_IRQ2_EVE3	Wakeup dependency from GPIO4 module (SWakeup signal) towards EVE3 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
17	WKUPDEP_GPIO4_IRQ2_EVE2	Wakeup dependency from GPIO4 module (SWakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
16	WKUPDEP_GPIO4_IRQ2_EVE1	Wakeup dependency from GPIO4 module (SWakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
15	WKUPDEP_GPIO4_IRQ2_DSP2	Wakeup dependency from GPIO4 module (SWakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
14	WKUPDEP_GPIO4_IRQ2_IPU1	Wakeup dependency from GPIO4 module (SWakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
13	RESERVED		R	0x0
12	WKUPDEP_GPIO4_IRQ2_DSP1	Wakeup dependency from GPIO4 module (SWakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
11	WKUPDEP_GPIO4_IRQ2_IPU2	Wakeup dependency from GPIO4 module (SWakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
10	WKUPDEP_GPIO4_IRQ2_MPU	Wakeup dependency from GPIO4 module (SWakeup signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
9	WKUPDEP_GPIO4_IRQ1_EVE4	Wakeup dependency from GPIO4 module (SWakeup signal) towards EVE4 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
8	WKUPDEP_GPIO4_IRQ1_EVE3	Wakeup dependency from GPIO4 module (SWakeup signal) towards EVE3 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
7	WKUPDEP_GPIO4_IRQ1_EVE2	Wakeup dependency from GPIO4 module (SWakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
6	WKUPDEP_GPIO4_IRQ1_EVE1	Wakeup dependency from GPIO4 module (SWakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_GPIO4_IRQ1_DSP2	Wakeup dependency from GPIO4 module (SWakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_GPIO4_IRQ1_IPU1	Wakeup dependency from GPIO4 module (SWakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	RESERVED		R	0x0
2	WKUPDEP_GPIO4_IRQ1_DSP1	Wakeup dependency from GPIO4 module (SWakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_GPIO4_IRQ1_IPU2	Wakeup dependency from GPIO4 module (SWakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_GPIO4_IRQ1_MPU	Wakeup dependency from GPIO4 module (SWakeup signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1527. Register Call Summary for Register PM_L4PER_GPIO4_WKDEP

Clock Management Functional Description • Clock Domain Dependency: [0][1][2][3][4][5][6][7]
PRCM Register Manual • Not Supported Functionality (Registers and Bitfields): [8][9][10][11][12][13][14][15][16][17] • L4PER_PRM Register Summary: [18]

Table 3-1528. RM_L4PER_GPIO4_CONTEXT

Address Offset	0x0000 0074		
Physical Address	0x4AE0 7474	**Instance**	L4PER_PRM
Description	This register contains dedicated GPIO4 context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														LOSTCONTEXT_RFF	RESERVED

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4PER_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
0	RESERVED		R	0x0

Table 3-1529. Register Call Summary for Register RM_L4PER_GPIO4_CONTEXT

Power Management Functional Description
PRCM Register Manual • L4PER_PRM Register Summary: [1]

Table 3-1530. PM_L4PER_GPIO5_WKDEP

Address Offset	0x0000 0078		
Physical Address	0x4AE0 7478	**Instance**	L4PER_PRM
Description	This register controls wakeup dependency based on GPIO5 service requests.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												WKUPDEP_GPIO5_IRQ2_EVE4	WKUPDEP_GPIO5_IRQ2_EVE3	WKUPDEP_GPIO5_IRQ2_EVE2	WKUPDEP_GPIO5_IRQ2_EVE1	WKUPDEP_GPIO5_IRQ2_DSP2	WKUPDEP_GPIO5_IRQ2_IPU1	RESERVED	WKUPDEP_GPIO5_IRQ2_DSP1	WKUPDEP_GPIO5_IRQ2_IPU2	WKUPDEP_GPIO5_IRQ2_MPU	WKUPDEP_GPIO5_IRQ1_EVE4	WKUPDEP_GPIO5_IRQ1_EVE3	WKUPDEP_GPIO5_IRQ1_EVE2	WKUPDEP_GPIO5_IRQ1_EVE1	WKUPDEP_GPIO5_IRQ1_DSP2	WKUPDEP_GPIO5_IRQ1_IPU1	RESERVED	WKUPDEP_GPIO5_IRQ1_DSP1	WKUPDEP_GPIO5_IRQ1_IPU2	WKUPDEP_GPIO5_IRQ1_MPU

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	WKUPDEP_GPIO5_IRQ2_EVE4	Wakeup dependency from GPIO5 module (SWakeup signal) towards EVE4 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
18	WKUPDEP_GPIO5_IRQ2_EVE3	Wakeup dependency from GPIO5 module (SWakeup signal) towards EVE3 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
17	WKUPDEP_GPIO5_IRQ2_EVE2	Wakeup dependency from GPIO5 module (SWakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
16	WKUPDEP_GPIO5_IRQ2_EVE1	Wakeup dependency from GPIO5 module (SWakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
15	WKUPDEP_GPIO5_IRQ2_DSP2	Wakeup dependency from GPIO5 module (SWakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
14	WKUPDEP_GPIO5_IRQ2_IPU1	Wakeup dependency from GPIO5 module (SWakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
13	RESERVED		R	0x0
12	WKUPDEP_GPIO5_IRQ2_DSP1	Wakeup dependency from GPIO5 module (SWakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
11	WKUPDEP_GPIO5_IRQ2_IPU2	Wakeup dependency from GPIO5 module (SWakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
10	WKUPDEP_GPIO5_IRQ2_MPU	Wakeup dependency from GPIO5 module (SWakeup signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
9	WKUPDEP_GPIO5_IRQ1_EVE4	Wakeup dependency from GPIO5 module (SWakeup signal) towards EVE4 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
8	WKUPDEP_GPIO5_IRQ1_EVE3	Wakeup dependency from GPIO5 module (SWakeup signal) towards EVE3 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
7	WKUPDEP_GPIO5_IRQ1_EVE2	Wakeup dependency from GPIO5 module (SWakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
6	WKUPDEP_GPIO5_IRQ1_EVE1	Wakeup dependency from GPIO5 module (SWakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_GPIO5_IRQ1_DSP2	Wakeup dependency from GPIO5 module (SWakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_GPIO5_IRQ1_IPU1	Wakeup dependency from GPIO5 module (SWakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	RESERVED		R	0x0
2	WKUPDEP_GPIO5_IRQ1_DSP1	Wakeup dependency from GPIO5 module (SWakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_GPIO5_IRQ1_IPU2	5Wakeup dependency from GPIO4 module (SWakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_GPIO5_IRQ1_MPU	Wakeup dependency from GPIO5 module (SWakeup signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1531. RM_L4PER_GPIO5_CONTEXT

Address Offset	0x0000 007C		
Physical Address	0x4AE0 747C	**Instance**	L4PER_PRM
Description	This register contains dedicated GPIO5 context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														LOSTCONTEXT_RFF	RESERVED

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4PER_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
0	RESERVED		R	0x0

Table 3-1532. PM_L4PER_GPIO6_WKDEP

Address Offset	0x0000 0080		
Physical Address	0x4AE0 7480	**Instance**	L4PER_PRM
Description	This register controls wakeup dependency based on GPIO6 service requests.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												WKUPDEP_GPIO6_IRQ2_EVE4	WKUPDEP_GPIO6_IRQ2_EVE3	WKUPDEP_GPIO6_IRQ2_EVE2	WKUPDEP_GPIO6_IRQ2_EVE1	WKUPDEP_GPIO6_IRQ2_DSP2	WKUPDEP_GPIO6_IRQ2_IPU1	RESERVED	WKUPDEP_GPIO6_IRQ2_DSP1	WKUPDEP_GPIO6_IRQ2_IPU2	WKUPDEP_GPIO6_IRQ2_MPU	WKUPDEP_GPIO6_IRQ1_EVE4	WKUPDEP_GPIO6_IRQ1_EVE3	WKUPDEP_GPIO6_IRQ1_EVE2	WKUPDEP_GPIO6_IRQ1_EVE1	WKUPDEP_GPIO6_IRQ1_DSP2	WKUPDEP_GPIO6_IRQ1_IPU1	RESERVED	WKUPDEP_GPIO6_IRQ1_DSP1	WKUPDEP_GPIO6_IRQ1_IPU2	WKUPDEP_GPIO6_IRQ1_MPU

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	WKUPDEP_GPIO6_IRQ2_EVE4	Wakeup dependency from GPIO6 module (SWakeup signal) towards EVE4 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
18	WKUPDEP_GPIO6_IRQ2_EVE3	Wakeup dependency from GPIO6 module (SWakeup signal) towards EVE3 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
17	WKUPDEP_GPIO6_IRQ2_EVE2	Wakeup dependency from GPIO6 module (SWakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
16	WKUPDEP_GPIO6_IRQ2_EVE1	Wakeup dependency from GPIO6 module (SWakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
15	WKUPDEP_GPIO6_IRQ2_DSP2	Wakeup dependency from GPIO6 module (SWakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
14	WKUPDEP_GPIO6_IRQ2_IPU1	Wakeup dependency from GPIO6 module (SWakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
13	RESERVED		R	0x0
12	WKUPDEP_GPIO6_IRQ2_DSP1	Wakeup dependency from GPIO6 module (SWakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
11	WKUPDEP_GPIO6_IRQ2_IPU2	Wakeup dependency from GPIO6 module (SWakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
10	WKUPDEP_GPIO6_IRQ2_MPU	Wakeup dependency from GPIO6 module (SWakeup signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
9	WKUPDEP_GPIO6_IRQ1_EVE4	Wakeup dependency from GPIO6 module (SWakeup signal) towards EVE4 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
8	WKUPDEP_GPIO6_IRQ1_EVE3	Wakeup dependency from GPIO6 module (SWakeup signal) towards EVE3 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
7	WKUPDEP_GPIO6_IRQ1_EVE2	Wakeup dependency from GPIO6 module (SWakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_GPIO6_IRQ1_EVE1	Wakeup dependency from GPIO6 module (SWakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_GPIO6_IRQ1_DSP2	Wakeup dependency from GPIO6 module (SWakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_GPIO6_IRQ1_IPU1	Wakeup dependency from GPIO6 module (SWakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
2	WKUPDEP_GPIO6_IRQ1_DSP1	Wakeup dependency from GPIO6 module (SWakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_GPIO6_IRQ1_IPU2	5Wakeup dependency from GPIO6 module (SWakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_GPIO6_IRQ1_MPU	Wakeup dependency from GPIO6 module (SWakeup signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1533. RM_L4PER_GPIO6_CONTEXT

Address Offset	0x0000 0084		
Physical Address	0x4AE0 7484	**Instance**	L4PER_PRM
Description	This register contains dedicated GPIO6 context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														LOSTCONTEXT_RFF	RESERVED

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4PER_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
0	RESERVED		R	0x0

Table 3-1534. RM_L4PER_ESM_CONTEXT

Address Offset	0x0000 008C		
Physical Address	0x4AE0 748C	**Instance**	L4PER_PRM
Description	This register contains dedicated ESM context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4PER_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1535. Register Call Summary for Register RM_L4PER_ESM_CONTEXT

Power Management Functional Description
PRCM Register Manual • L4PER_PRM Register Summary: [1]

Table 3-1536. RM_L4PER2_PWMSS2_CONTEXT

Address Offset	0x0000 0094		
Physical Address	0x4AE0 7494	**Instance**	L4PER_PRM
Description	This register contains dedicated PWMSS2 context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1537. RM_L4PER2_PWMSS3_CONTEXT

Address Offset	0x0000 009C		
Physical Address	0x4AE0 749C	**Instance**	L4PER_PRM
Description	This register contains dedicated PWMSS3 context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1538. PM_L4PER_I2C1_WKDEP

Address Offset	0x0000 00A0		
Physical Address	0x4AE0 74A0	**Instance**	L4PER_PRM
Description	This register controls wakeup dependency based on I2C1 service requests.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																WKUPDEP_I2C1_DMA_DSP2	RESERVED	WKUPDEP_I2C1_DMA_SDMA	WKUPDEP_I2C1_DMA_DSP1	RESERVED		WKUPDEP_I2C1_IRQ_EVE4	WKUPDEP_I2C1_IRQ_EVE3	WKUPDEP_I2C1_IRQ_EVE2	WKUPDEP_I2C1_IRQ_EVE1	WKUPDEP_I2C1_IRQ_DSP2	WKUPDEP_I2C1_IRQ_IPU1	RESERVED	WKUPDEP_I2C1_IRQ_DSP1	WKUPDEP_I2C1_IRQ_IPU2	WKUPDEP_I2C1_IRQ_MPU

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0
15	WKUPDEP_I2C1_DMA_DSP2	Wakeup dependency from I2C1 module (SWakeup_dma signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
14	RESERVED		R	0x0
13	WKUPDEP_I2C1_DMA_SDMA	Wakeup dependency from I2C1 module (SWakeup_dma signal) towards SDMA + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
12	WKUPDEP_I2C1_DMA_DSP1	Wakeup dependency from I2C1 module (SWakeup_dma signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1

Bits	Field Name	Description	Type	Reset
11:10	RESERVED		R	0x0
9	WKUPDEP_I2C1_IRQ_EVE4	Wakeup dependency from I2C1 module (SWakeup IRQ signal) towards EVE4 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
8	WKUPDEP_I2C1_IRQ_EVE3	Wakeup dependency from I2C1 module (SWakeup IRQ signal) towards EVE3 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
7	WKUPDEP_I2C1_IRQ_EVE2	Wakeup dependency from I2C1 module (SWakeup IRQ signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_I2C1_IRQ_EVE1	Wakeup dependency from I2C1 module (SWakeup IRQ signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_I2C1_IRQ_DSP2	Wakeup dependency from I2C1 module (SWakeup IRQ signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_I2C1_IRQ_IPU1	Wakeup dependency from I2C1 module (SWakeup IRQ signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	RESERVED		R	0x0
2	WKUPDEP_I2C1_IRQ_DSP1	Wakeup dependency from I2C1 module (SWakeup IRQ signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_I2C1_IRQ_IPU2	Wakeup dependency from I2C1 module (SWakeup IRQ signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_I2C1_IRQ_MPU	Wakeup dependency from I2C1 module (SWakeup IRQ signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1539. Register Call Summary for Register PM_L4PER_I2C1_WKDEP

Clock Management Functional Description • Clock Domain Dependency: [0][1][2][3][4][5]
PRCM Register Manual • Not Supported Functionality (Registers and Bitfields): [6][7][8][9][10][11] • L4PER_PRM Register Summary: [12]

Table 3-1540. RM_L4PER_I2C1_CONTEXT

Address Offset	0x0000 00A4		
Physical Address	0x4AE0 74A4	**Instance**	L4PER_PRM
Description	This register contains dedicated I2C1 context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														LOSTCONTEXT_RFF	RESERVED

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4PER_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
0	RESERVED		R	0x0

Table 3-1541. Register Call Summary for Register RM_L4PER_I2C1_CONTEXT

Power Management Functional Description

PRCM Register Manual
- L4PER_PRM Register Summary: [1]

Table 3-1542. PM_L4PER_I2C2_WKDEP

Address Offset	0x0000 00A8		
Physical Address	0x4AE0 74A8	**Instance**	L4PER_PRM
Description	This register controls wakeup dependency based on I2C2 service requests.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																WKUPDEP_I2C2_DMA_DSP2	RESERVED	WKUPDEP_I2C2_DMA_SDMA	WKUPDEP_I2C2_DMA_DSP1	RESERVED		WKUPDEP_I2C2_IRQ_EVE4	WKUPDEP_I2C2_IRQ_EVE3	WKUPDEP_I2C2_IRQ_EVE2	WKUPDEP_I2C2_IRQ_EVE1	WKUPDEP_I2C2_IRQ_DSP2	WKUPDEP_I2C2_IRQ_IPU1	RESERVED	WKUPDEP_I2C2_IRQ_DSP1	WKUPDEP_I2C2_IRQ_IPU2	WKUPDEP_I2C2_IRQ_MPU

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0
15	WKUPDEP_I2C2_DMA_DSP2	Wakeup dependency from I2C2 module (SWakeup_dma signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
14	RESERVED		R	0x0
13	WKUPDEP_I2C2_DMA_SDMA	Wakeup dependency from I2C2 module (SWakeup_dma signal) towards SDMA + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
12	WKUPDEP_I2C2_DMA_DSP1	Wakeup dependency from I2C2 module (SWakeup_dma signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
11:10	RESERVED		R	0x0
9	WKUPDEP_I2C2_IRQ_EVE4	Wakeup dependency from I2C2 module (SWakeup IRQ signal) towards EVE4 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
8	WKUPDEP_I2C2_IRQ_EVE3	Wakeup dependency from I2C2 module (SWakeup IRQ signal) towards EVE3 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
7	WKUPDEP_I2C2_IRQ_EVE2	Wakeup dependency from I2C2 module (SWakeup IRQ signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_I2C2_IRQ_EVE1	Wakeup dependency from I2C2 module (SWakeup IRQ signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_I2C2_IRQ_DSP2	Wakeup dependency from I2C2 module (SWakeup IRQ signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_I2C2_IRQ_IPU1	Wakeup dependency from I2C2 module (SWakeup IRQ signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	RESERVED		R	0x0
2	WKUPDEP_I2C2_IRQ_DSP1	Wakeup dependency from I2C2 module (SWakeup IRQ signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
1	WKUPDEP_I2C2_IRQ_IPU2	Wakeup dependency from I2C2 module (SWakeup IRQ signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_I2C2_IRQ_MPU	Wakeup dependency from I2C2 module (SWakeup IRQ signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1543. Register Call Summary for Register PM_L4PER_I2C2_WKDEP

Clock Management Functional Description • Clock Domain Dependency: [0][1][2][3][4][5]
PRCM Register Manual • Not Supported Functionality (Registers and Bitfields): [6][7][8][9][10] • L4PER_PRM Register Summary: [11]

Table 3-1544. RM_L4PER_I2C2_CONTEXT

Address Offset	0x0000 00AC		
Physical Address	0x4AE0 74AC	**Instance**	L4PER_PRM
Description	This register contains dedicated I2C2 context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4PER_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1545. Register Call Summary for Register RM_L4PER_I2C2_CONTEXT

Power Management Functional Description
PRCM Register Manual • L4PER_PRM Register Summary: [1]

Table 3-1546. PM_L4PER_I2C3_WKDEP

Address Offset	0x0000 00B0		
Physical Address	0x4AE0 74B0	**Instance**	L4PER_PRM
Description	This register controls wakeup dependency based on I2C3 service requests.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																WKUPDEP_I2C3_DMA_DSP2	RESERVED	WKUPDEP_I2C3_DMA_SDMA	WKUPDEP_I2C3_DMA_DSP1	RESERVED		WKUPDEP_I2C3_IRQ_EVE4	WKUPDEP_I2C3_IRQ_EVE3	WKUPDEP_I2C3_IRQ_EVE2	WKUPDEP_I2C3_IRQ_EVE1	WKUPDEP_I2C3_IRQ_DSP2	WKUPDEP_I2C3_IRQ_IPU1	RESERVED	WKUPDEP_I2C3_IRQ_DSP1	WKUPDEP_I2C3_IRQ_IPU2	WKUPDEP_I2C3_IRQ_MPU

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0
15	WKUPDEP_I2C3_DMA_DSP2	Wakeup dependency from I2C3 module (SWakeup_dma signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
14	RESERVED		R	0x0
13	WKUPDEP_I2C3_DMA_SDMA	Wakeup dependency from I2C3 module (SWakeup_dma signal) towards SDMA + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
12	WKUPDEP_I2C3_DMA_DSP1	Wakeup dependency from I2C3 module (SWakeup_dma signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
11:10	RESERVED		R	0x0
9	WKUPDEP_I2C3_IRQ_EVE4	Wakeup dependency from I2C3 module (SWakeup IRQ signal) towards EVE4 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
8	WKUPDEP_I2C3_IRQ_EVE3	Wakeup dependency from I2C3 module (SWakeup IRQ signal) towards EVE3 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
7	WKUPDEP_I2C3_IRQ_EVE2	Wakeup dependency from I2C3 module (SWakeup IRQ signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
6	WKUPDEP_I2C3_IRQ_EVE1	Wakeup dependency from I2C3 module (SWakeup IRQ signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_I2C3_IRQ_DSP2	Wakeup dependency from I2C3 module (SWakeup IRQ signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_I2C3_IRQ_IPU1	Wakeup dependency from I2C3 module (SWakeup IRQ signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	RESERVED		R	0x0
2	WKUPDEP_I2C3_IRQ_DSP1	Wakeup dependency from I2C3 module (SWakeup IRQ signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_I2C3_IRQ_IPU2	Wakeup dependency from I2C3 module (SWakeup IRQ signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_I2C3_IRQ_MPU	Wakeup dependency from I2C3 module (SWakeup IRQ signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1547. RM_L4PER_I2C3_CONTEXT

Address Offset	0x0000 00B4		
Physical Address	0x4AE0 74B4	**Instance**	L4PER_PRM
Description	This register contains dedicated I2C3 context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4PER_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1548. PM_L4PER_I2C4_WKDEP

Address Offset	0x0000 00B8		
Physical Address	0x4AE0 74B8	**Instance**	L4PER_PRM
Description	This register controls wakeup dependency based on I2C4 service requests.		
Type	RW		

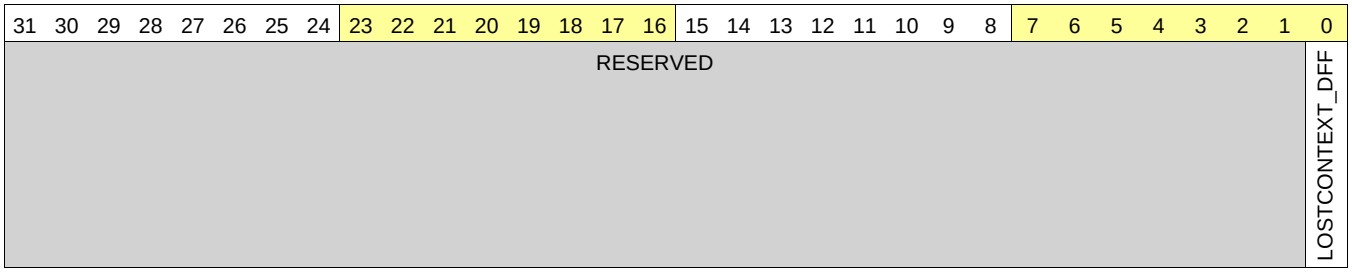
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																WKUPDEP_I2C4_DMA_DSP2	RESERVED	WKUPDEP_I2C4_DMA_SDMA	WKUPDEP_I2C4_DMA_DSP1	RESERVED		WKUPDEP_I2C4_IRQ_EVE4	WKUPDEP_I2C4_IRQ_EVE3	WKUPDEP_I2C4_IRQ_EVE2	WKUPDEP_I2C4_IRQ_EVE1	WKUPDEP_I2C4_IRQ_DSP2	WKUPDEP_I2C4_IRQ_IPU1	RESERVED	WKUPDEP_I2C4_IRQ_DSP1	WKUPDEP_I2C4_IRQ_IPU2	WKUPDEP_I2C4_IRQ_MPU

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0
15	WKUPDEP_I2C4_DMA_DSP2	Wakeup dependency from I2C4 module (SWakeup_dma signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
14	RESERVED		R	0x0
13	WKUPDEP_I2C4_DMA_SDMA	Wakeup dependency from I2C4 module (SWakeup_dma signal) towards SDMA + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
12	WKUPDEP_I2C4_DMA_DSP1	Wakeup dependency from I2C4 module (SWakeup_dma signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
11:10	RESERVED		R	0x0
9	WKUPDEP_I2C4_IRQ_EVE4	Wakeup dependency from I2C4 module (SWakeup IRQ signal) towards EVE4 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
8	WKUPDEP_I2C4_IRQ_EVE3	Wakeup dependency from I2C4 module (SWakeup IRQ signal) towards EVE3 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
7	WKUPDEP_I2C4_IRQ_EVE2	Wakeup dependency from I2C4 module (SWakeup IRQ signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_I2C4_IRQ_EVE1	Wakeup dependency from I2C4 module (SWakeup IRQ signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_I2C4_IRQ_DSP2	Wakeup dependency from I2C4 module (SWakeup IRQ signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_I2C4_IRQ_IPU1	Wakeup dependency from I2C4 module (SWakeup IRQ signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	RESERVED		R	0x0
2	WKUPDEP_I2C4_IRQ_DSP1	Wakeup dependency from I2C4 module (SWakeup IRQ signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_I2C4_IRQ_IPU2	Wakeup dependency from I2C4 module (SWakeup IRQ signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_I2C4_IRQ_MPU	Wakeup dependency from I2C4 module (SWakeup IRQ signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1549. RM_L4PER_I2C4_CONTEXT

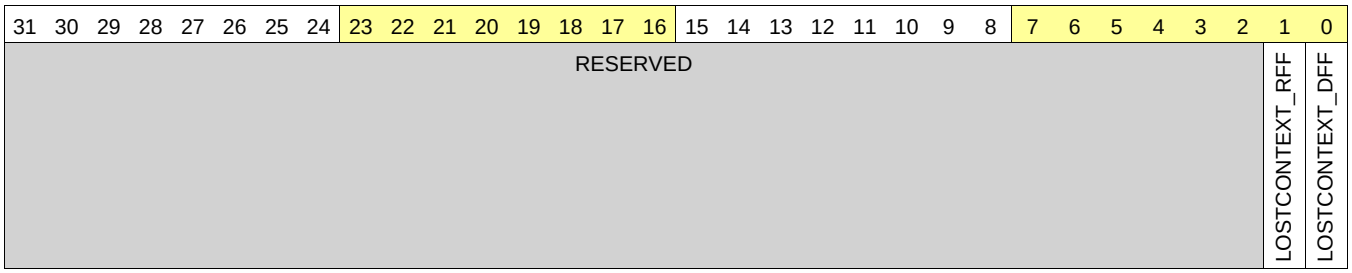
Address Offset	0x0000 00BC		
Physical Address	0x4AE0 74BC	**Instance**	L4PER_PRM
Description	This register contains dedicated I2C4 context statuses. [warm reset insensitive]		
Type	RW		



Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4PER_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1550. RM_L4PER_L4PER1_CONTEXT

Address Offset	0x0000 00C0		
Physical Address	0x4AE0 74C0	**Instance**	L4PER_PRM
Description	This register contains dedicated L4_PER1 context statuses. [warm reset insensitive]		
Type	RW		



Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4PER_PWRON_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4PER_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1551. Register Call Summary for Register RM_L4PER_L4PER1_CONTEXT

Power Management Functional Description
PRCM Register Manual • L4PER_PRM Register Summary: [2]

Table 3-1552. RM_L4PER2_PWMSS1_CONTEXT

Address Offset	0x0000 00C4		
Physical Address	0x4AE0 74C4	**Instance**	L4PER_PRM
Description	This register contains dedicated PWMSS1 context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1553. Register Call Summary for Register RM_L4PER2_PWMSS1_CONTEXT

Power Management Functional Description
PRCM Register Manual • L4PER_PRM Register Summary: [1]

Table 3-1554. PM_L4PER_DCC1_WKDEP

Address Offset	0x0000 00C8		
Physical Address	0x4AE0 74C8	**Instance**	L4PER_PRM
Description	This register controls wakeup dependency based on DCC1 service requests.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						WKUPDEP_DCC1_EVE4	WKUPDEP_DCC1_EVE3	WKUPDEP_DCC1_EVE2	WKUPDEP_DCC1_EVE1	WKUPDEP_DCC1_DSP2	WKUPDEP_DCC1_IPU1	RESERVED	WKUPDEP_DCC1_DSP1	WKUPDEP_DCC1_IPU2	WKUPDEP_DCC1_MPU

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	WKUPDEP_DCC1_EVE4	Wakeup dependency from DCC1 module (SWakeup IRQ signal) towards EVE4 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
8	WKUPDEP_DCC1_EVE3	Wakeup dependency from DCC1 module (SWakeup IRQ signal) towards EVE3 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
7	WKUPDEP_DCC1_EVE2	Wakeup dependency from DCC1 module (SWakeup IRQ signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_DCC1_EVE1	Wakeup dependency from DCC1 module (SWakeup IRQ signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_DCC1_DSP2	Wakeup dependency from DCC1 module (SWakeup IRQ signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_DCC1_IPU1	Wakeup dependency from DCC1 module (SWakeup IRQ signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	RESERVED		R	0x0
2	WKUPDEP_DCC1_DSP1	Wakeup dependency from DCC1 module (SWakeup IRQ signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_DCC1_IPU2	Wakeup dependency from DCC1 module (SWakeup IRQ signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_DCC1_MPU	Wakeup dependency from DCC1 module (SWakeup IRQ signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1555. RM_L4PER3_DCC1_CONTEXT

Address Offset	0x0000 00CC		
Physical Address	0x4AE0 74CC	**Instance**	L4PER_PRM
Description	This register contains dedicated DCC1 context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4PER_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1556. Register Call Summary for Register RM_L4PER3_DCC1_CONTEXT

Power Management Functional Description
PRCM Register Manual • L4PER_PRM Register Summary: [1]

Table 3-1557. PM_L4PER_DCC2_WKDEP

Address Offset	0x0000 00D0		
Physical Address	0x4AE0 74D0	**Instance**	L4PER_PRM
Description	This register controls wakeup dependency based on DCC2 service requests.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						WKUPDEP_DCC2_EVE4	WKUPDEP_DCC2_EVE3	WKUPDEP_DCC2_EVE2	WKUPDEP_DCC2_EVE1	WKUPDEP_DCC2_DSP2	WKUPDEP_DCC2_IPU1	RESERVED	WKUPDEP_DCC2_DSP1	WKUPDEP_DCC2_IPU2	WKUPDEP_DCC2_MPU

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	WKUPDEP_DCC2_EVE4	Wakeup dependency from DCC2 module (SWakeup IRQ signal) towards EVE4 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
8	WKUPDEP_DCC2_EVE3	Wakeup dependency from DCC2 module (SWakeup IRQ signal) towards EVE3 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
7	WKUPDEP_DCC2_EVE2	Wakeup dependency from DCC2 module (SWakeup IRQ signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_DCC2_EVE1	Wakeup dependency from DCC2 module (SWakeup IRQ signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_DCC2_DSP2	Wakeup dependency from DCC2 module (SWakeup IRQ signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_DCC2_IPU1	Wakeup dependency from DCC2 module (SWakeup IRQ signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	RESERVED		R	0x0
2	WKUPDEP_DCC2_DSP1	Wakeup dependency from DCC2 module (SWakeup IRQ signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_DCC2_IPU2	Wakeup dependency from DCC2 module (SWakeup IRQ signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_DCC2_MPU	Wakeup dependency from DCC2 module (SWakeup IRQ signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1558. RM_L4PER3_DCC2_CONTEXT

Address Offset	0x0000 00D4		
Physical Address	0x4AE0 74D4	**Instance**	L4PER_PRM
Description	This register contains dedicated DCC2 context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4PER_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1559. Register Call Summary for Register RM_L4PER3_DCC2_CONTEXT

Power Management Functional Description
PRCM Register Manual • L4PER_PRM Register Summary: [1]

Table 3-1560. PM_L4PER_DCC3_WKDEP

Address Offset	0x0000 00D8		
Physical Address	0x4AE0 74D8	**Instance**	L4PER_PRM
Description	This register controls wakeup dependency based on DCC3 service requests.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						WKUPDEP_DCC3_EVE4	WKUPDEP_DCC3_EVE3	WKUPDEP_DCC3_EVE2	WKUPDEP_DCC3_EVE1	WKUPDEP_DCC3_DSP2	WKUPDEP_DCC3_IPU1	RESERVED	WKUPDEP_DCC3_DSP1	WKUPDEP_DCC3_IPU2	WKUPDEP_DCC3_MPU

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	WKUPDEP_DCC3_EVE4	5Wakeup dependency from DCC3 module (SWakeup IRQ signal) towards EVE4 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
8	WKUPDEP_DCC3_EVE3	Wakeup dependency from DCC3 module (SWakeup IRQ signal) towards EVE3 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
7	WKUPDEP_DCC3_EVE2	Wakeup dependency from DCC3 module (SWakeup IRQ signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_DCC3_EVE1	Wakeup dependency from DCC3 module (SWakeup IRQ signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
5	WKUPDEP_DCC3_DSP2	Wakeup dependency from DCC3 module (SWakeup IRQ signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_DCC3_IPU1	Wakeup dependency from DCC3 module (SWakeup IRQ signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	RESERVED		R	0x0
2	WKUPDEP_DCC3_DSP1	Wakeup dependency from DCC3 module (SWakeup IRQ signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_DCC3_IPU2	Wakeup dependency from DCC3 module (SWakeup IRQ signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_DCC3_MPU	Wakeup dependency from DCC3 module (SWakeup IRQ signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1561. RM_L4PER3_DCC3_CONTEXT

Address Offset	0x0000 00DC		
Physical Address	0x4AE0 74DC	**Instance**	L4PER_PRM
Description	This register contains dedicated DCC3 context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4PER_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1562. Register Call Summary for Register RM_L4PER3_DCC3_CONTEXT

Power Management Functional Description
PRCM Register Manual • L4PER_PRM Register Summary: [1]

Table 3-1563. PM_L4PER_MCSPI1_WKDEP

Address Offset	0x0000 00F0		
Physical Address	0x4AE0 74F0	**Instance**	L4PER_PRM
Description	This register controls wakeup dependency based on McSPI1 service requests.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						WKUPDEP_MCSPI1_EVE4	WKUPDEP_MCSPI1_EVE3	WKUPDEP_MCSPI1_EVE2	WKUPDEP_MCSPI1_EVE1	WKUPDEP_MCSPI1_DSP2	WKUPDEP_MCSPI1_IPU1	WKUPDEP_MCSPI1_SDMA	WKUPDEP_MCSPI1_DSP1	WKUPDEP_MCSPI1_IPU2	WKUPDEP_MCSPI1_MPU								

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	WKUPDEP_MCSPI1_EVE4	Wakeup dependency from McSPI1 module (SWakeup signal) towards EVE4 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
8	WKUPDEP_MCSPI1_EVE3	Wakeup dependency from McSPI1 module (SWakeup signal) towards EVE3 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
7	WKUPDEP_MCSPI1_EVE2	Wakeup dependency from McSPI1 module (SWakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_MCSPI1_EVE1	Wakeup dependency from McSPI1 module (SWakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_MCSPI1_DSP2	Wakeup dependency from McSPI1 module (SWakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
4	WKUPDEP_MCSPI1_IPU1	Wakeup dependency from McSPI1 module (SWakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	WKUPDEP_MCSPI1_SDMA	Wakeup dependency from McSPI1 module (SWakeup signal) towards SDMA + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
2	WKUPDEP_MCSPI1_DSP1	Wakeup dependency from McSPI1 module (SWakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_MCSPI1_IPU2	Wakeup dependency from McSPI1 module (SWakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_MCSPI1_MPU	Wakeup dependency from McSPI1 module (SWakeup signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1564. Register Call Summary for Register PM_L4PER_MCSPI1_WKDEP

Clock Management Functional Description • Clock Domain Dependency: [0][1][2][3]
PRCM Register Manual • Not Supported Functionality (Registers and Bitfields): [4][5][6][7][8][9] • L4PER_PRM Register Summary: [10]

Table 3-1565. RM_L4PER_MCSPI1_CONTEXT

Address Offset	0x0000 00F4		
Physical Address	0x4AE0 74F4	**Instance**	L4PER_PRM
Description	This register contains dedicated McSPI1 context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4PER_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1566. Register Call Summary for Register RM_L4PER_MCSPI1_CONTEXT

Power Management Functional Description
PRCM Register Manual • L4PER_PRM Register Summary: [1]

Table 3-1567. PM_L4PER_MCSPI2_WKDEP

Address Offset	0x0000 00F8		
Physical Address	0x4AE0 74F8	**Instance**	L4PER_PRM
Description	This register controls wakeup dependency based on McSPI2 service requests.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						WKUPDEP_MCSPI2_EVE4	WKUPDEP_MCSPI2_EVE3	WKUPDEP_MCSPI2_EVE2	WKUPDEP_MCSPI2_EVE1	WKUPDEP_MCSPI2_DSP2	WKUPDEP_MCSPI2_IPU1	WKUPDEP_MCSPI2_SDMA	WKUPDEP_MCSPI2_DSP1	WKUPDEP_MCSPI2_IPU2	WKUPDEP_MCSPI2_MPU

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	WKUPDEP_MCSPI2_EVE4	Wakeup dependency from McSPI2 module (SWakeup signal) towards EVE4 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
8	WKUPDEP_MCSPI2_EVE3	Wakeup dependency from McSPI2 module (SWakeup signal) towards EVE3 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
7	WKUPDEP_MCSPI2_EVE2	Wakeup dependency from McSPI2 module (SWakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_MCSPI2_EVE1	Wakeup dependency from McSPI2 module (SWakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
5	WKUPDEP_MCSPI2_DSP2	Wakeup dependency from McSPI2 module (SWakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_MCSPI2_IPU1	Wakeup dependency from McSPI2 module (SWakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	WKUPDEP_MCSPI2_SDMA	Wakeup dependency from McSPI2 module (SWakeup signal) towards SDMA + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
2	WKUPDEP_MCSPI2_DSP1	Wakeup dependency from McSPI2 module (SWakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_MCSPI2_IPU2	Wakeup dependency from McSPI2 module (SWakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_MCSPI2_MPU	Wakeup dependency from McSPI2 module (SWakeup signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1568. Register Call Summary for Register PM_L4PER_MCSPI2_WKDEP

Clock Management Functional Description • Clock Domain Dependency: [0][1][2][3]
PRCM Register Manual • Not Supported Functionality (Registers and Bitfields): [4][5][6][7][8][9] • L4PER_PRM Register Summary: [10]

Table 3-1569. RM_L4PER_MCSPI2_CONTEXT

Address Offset	0x0000 00FC		
Physical Address	0x4AE0 74FC	**Instance**	L4PER_PRM
Description	This register contains dedicated McSPI2 context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4PER_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1570. Register Call Summary for Register RM_L4PER_MCSPI2_CONTEXT

Power Management Functional Description
PRCM Register Manual • L4PER_PRM Register Summary: [1]

Table 3-1571. PM_L4PER_MCSPI3_WKDEP

Address Offset	0x0000 0100		
Physical Address	0x4AE0 7500	**Instance**	L4PER_PRM
Description	This register controls wakeup dependency based on McSPI3 service requests.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						WKUPDEP_MCSPI3_EVE4	WKUPDEP_MCSPI3_EVE3	WKUPDEP_MCSPI3_EVE2	WKUPDEP_MCSPI3_EVE1	WKUPDEP_MCSPI3_DSP2	WKUPDEP_MCSPI3_IPU1	WKUPDEP_MCSPI3_SDMA	WKUPDEP_MCSPI3_DSP1	WKUPDEP_MCSPI3_IPU2	WKUPDEP_MCSPI3_MPU

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	WKUPDEP_MCSPI3_EVE4	Wakeup dependency from McSPI3 module (SWakeup signal) towards EVE4 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
8	WKUPDEP_MCSPI3_EVE3	Wakeup dependency from McSPI3 module (SWakeup signal) towards EVE3 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
7	WKUPDEP_MCSPI3_EVE2	Wakeup dependency from McSPI3 module (SWakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_MCSPI3_EVE1	Wakeup dependency from McSPI3 module (SWakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
5	WKUPDEP_MCSPI3_DSP2	Wakeup dependency from McSPI3 module (SWakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_MCSPI3_IPU1	Wakeup dependency from McSPI3 module (SWakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	WKUPDEP_MCSPI3_SDMA	Wakeup dependency from McSPI3 module (SWakeup signal) towards SDMA + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
2	WKUPDEP_MCSPI3_DSP1	Wakeup dependency from McSPI3 module (SWakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_MCSPI3_IPU2	Wakeup dependency from McSPI3 module (SWakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_MCSPI3_MPU	Wakeup dependency from McSPI3 module (SWakeup signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1572. Register Call Summary for Register PM_L4PER_MCSPI3_WKDEP

Clock Management Functional Description
- Clock Domain Dependency: [0][1][2][3]

PRCM Register Manual
- Not Supported Functionality (Registers and Bitfields): [4][5][6][7][8][9]
- L4PER_PRM Register Summary: [10]

Table 3-1573. RM_L4PER_MCSPI3_CONTEXT

Address Offset	0x0000 0104		
Physical Address	0x4AE0 7504	**Instance**	L4PER_PRM
Description	This register contains dedicated McSPI3 context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4PER_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1574. Register Call Summary for Register RM_L4PER_MCSPI3_CONTEXT

Power Management Functional Description
PRCM Register Manual • L4PER_PRM Register Summary: [1]

Table 3-1575. PM_L4PER_MCSPI4_WKDEP

Address Offset	0x0000 0108		
Physical Address	0x4AE0 7508	**Instance**	L4PER_PRM
Description	This register controls wakeup dependency based on McSPI4 service requests.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						WKUPDEP_MCSPI4_EVE4	WKUPDEP_MCSPI4_EVE3	WKUPDEP_MCSPI4_EVE2	WKUPDEP_MCSPI4_EVE1	WKUPDEP_MCSPI4_DSP2	WKUPDEP_MCSPI4_IPU1	WKUPDEP_MCSPI4_SDMA	WKUPDEP_MCSPI4_DSP1	WKUPDEP_MCSPI4_IPU2	WKUPDEP_MCSPI4_MPU

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	WKUPDEP_MCSPI4_EVE4	Wakeup dependency from McSPI4 module (SWakeup signal) towards EVE4 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
8	WKUPDEP_MCSPI4_EVE3	Wakeup dependency from McSPI4 module (SWakeup signal) towards EVE3 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
7	WKUPDEP_MCSPI4_EVE2	Wakeup dependency from McSPI4 module (SWakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_MCSPI4_EVE1	Wakeup dependency from McSPI4 module (SWakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
5	WKUPDEP_MCSPI4_DSP2	Wakeup dependency from McSPI4 module (SWakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_MCSPI4_IPU1	Wakeup dependency from McSPI4 module (SWakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	WKUPDEP_MCSPI4_SDMA	Wakeup dependency from McSPI4 module (SWakeup signal) towards SDMA + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
2	WKUPDEP_MCSPI4_DSP1	Wakeup dependency from McSPI4 module (SWakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_MCSPI4_IPU2	Wakeup dependency from McSPI4 module (SWakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_MCSPI4_MPU	Wakeup dependency from McSPI4 module (SWakeup signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1576. Register Call Summary for Register PM_L4PER_MCSPI4_WKDEP

Clock Management Functional Description • Clock Domain Dependency: [0][1][2][3]
PRCM Register Manual • Not Supported Functionality (Registers and Bitfields): [4][5][6][7][8][9] • L4PER_PRM Register Summary: [10]

Table 3-1577. RM_L4PER_MCSPI4_CONTEXT

Address Offset	0x0000 010C		
Physical Address	0x4AE0 750C	**Instance**	L4PER_PRM
Description	This register contains dedicated McSPI4 context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4PER_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1578. Register Call Summary for Register RM_L4PER_MCSPI4_CONTEXT

Power Management Functional Description
PRCM Register Manual • L4PER_PRM Register Summary: [1]

Table 3-1579. PM_L4PER_GPIO7_WKDEP

Address Offset	0x0000 0110		
Physical Address	0x4AE0 7510	**Instance**	L4PER_PRM
Description	This register controls wakeup dependency based on GPIO7 service requests.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												WKUPDEP_GPIO7_IRQ2_EVE4	WKUPDEP_GPIO7_IRQ2_EVE3	WKUPDEP_GPIO7_IRQ2_EVE2	WKUPDEP_GPIO7_IRQ2_EVE1	WKUPDEP_GPIO7_IRQ2_DSP2	WKUPDEP_GPIO7_IRQ2_IPU1	RESERVED	WKUPDEP_GPIO7_IRQ2_DSP1	WKUPDEP_GPIO7_IRQ2_IPU2	WKUPDEP_GPIO7_IRQ2_MPU	WKUPDEP_GPIO7_IRQ1_EVE4	WKUPDEP_GPIO7_IRQ1_EVE3	WKUPDEP_GPIO7_IRQ1_EVE2	WKUPDEP_GPIO7_IRQ1_EVE1	WKUPDEP_GPIO7_IRQ1_DSP2	WKUPDEP_GPIO7_IRQ1_IPU1	RESERVED	WKUPDEP_GPIO7_IRQ1_DSP1	WKUPDEP_GPIO7_IRQ1_IPU2	WKUPDEP_GPIO7_IRQ1_MPU

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	WKUPDEP_GPIO7_IRQ2_EVE4	Wakeup dependency from GPIO7 module (SWakeup signal) towards EVE4 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
18	WKUPDEP_GPIO7_IRQ2_EVE3	Wakeup dependency from GPIO7 module (SWakeup signal) towards EVE3 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
17	WKUPDEP_GPIO7_IRQ2_EVE2	Wakeup dependency from GPIO7 module (SWakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
16	WKUPDEP_GPIO7_IRQ2_EVE1	Wakeup dependency from GPIO7 module (SWakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
15	WKUPDEP_GPIO7_IRQ2_DSP2	Wakeup dependency from GPIO7 module (SWakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
14	WKUPDEP_GPIO7_IRQ2_IPU1	Wakeup dependency from GPIO7 module (SWakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
13	RESERVED		R	0x0
12	WKUPDEP_GPIO7_IRQ2_DSP1	Wakeup dependency from GPIO7 module (SWakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
11	WKUPDEP_GPIO7_IRQ2_IPU2	Wakeup dependency from GPIO7 module (SWakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
10	WKUPDEP_GPIO7_IRQ2_MPU	Wakeup dependency from GPIO7 module (SWakeup signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
9	WKUPDEP_GPIO7_IRQ1_EVE4	Wakeup dependency from GPIO7 module (SWakeup signal) towards EVE4 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
8	WKUPDEP_GPIO7_IRQ1_EVE3	Wakeup dependency from GPIO7 module (SWakeup signal) towards EVE3 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
7	WKUPDEP_GPIO7_IRQ1_EVE2	Wakeup dependency from GPIO7 module (SWakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_GPIO7_IRQ1_EVE1	Wakeup dependency from GPIO7 module (SWakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_GPIO7_IRQ1_DSP2	Wakeup dependency from GPIO7 module (SWakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
4	WKUPDEP_GPIO7_IRQ1_IPU1	Wakeup dependency from GPIO7 module (SWakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	RESERVED		R	0x0
2	WKUPDEP_GPIO7_IRQ1_DSP1	Wakeup dependency from GPIO7 module (SWakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_GPIO7_IRQ1_IPU2	5Wakeup dependency from GPIO7 module (SWakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_GPIO7_IRQ1_MPU	Wakeup dependency from GPIO7 module (SWakeup signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1580. RM_L4PER_GPIO7_CONTEXT

Address Offset	0x0000 0114		
Physical Address	0x4AE0 7514	**Instance**	L4PER_PRM
Description	This register contains dedicated GPIO7 context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														LOSTCONTEXT_RFF	RESERVED

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4PER_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
0	RESERVED		R	0x0

Table 3-1581. PM_L4PER_GPIO8_WKDEP

Address Offset	0x0000 0118		
Physical Address	0x4AE0 7518	**Instance**	L4PER_PRM
Description	This register controls wakeup dependency based on GPIO8 service requests.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												WKUPDEP_GPIO8_IRQ2_EVE4	WKUPDEP_GPIO8_IRQ2_EVE3	WKUPDEP_GPIO8_IRQ2_EVE2	WKUPDEP_GPIO8_IRQ2_EVE1	WKUPDEP_GPIO8_IRQ2_DSP2	WKUPDEP_GPIO8_IRQ2_IPU1	RESERVED	WKUPDEP_GPIO8_IRQ2_DSP1	WKUPDEP_GPIO8_IRQ2_IPU2	WKUPDEP_GPIO8_IRQ2_MPU	WKUPDEP_GPIO8_IRQ1_EVE4	WKUPDEP_GPIO8_IRQ1_EVE3	WKUPDEP_GPIO8_IRQ1_EVE2	WKUPDEP_GPIO8_IRQ1_EVE1	WKUPDEP_GPIO8_IRQ1_DSP2	WKUPDEP_GPIO8_IRQ1_IPU1	RESERVED	WKUPDEP_GPIO8_IRQ1_DSP1	WKUPDEP_GPIO8_IRQ1_IPU2	WKUPDEP_GPIO8_IRQ1_MPU

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	WKUPDEP_GPIO8_IRQ2_EVE4	Wakeup dependency from GPIO8 module (SWakeup signal) towards EVE4 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
18	WKUPDEP_GPIO8_IRQ2_EVE3	Wakeup dependency from GPIO8 module (SWakeup signal) towards EVE3 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
17	WKUPDEP_GPIO8_IRQ2_EVE2	Wakeup dependency from GPIO8 module (SWakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
16	WKUPDEP_GPIO8_IRQ2_EVE1	Wakeup dependency from GPIO8 module (SWakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
15	WKUPDEP_GPIO8_IRQ2_DSP2	Wakeup dependency from GPIO8 module (SWakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
14	WKUPDEP_GPIO8_IRQ2_IPU1	Wakeup dependency from GPIO8 module (SWakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
13	RESERVED		R	0x0
12	WKUPDEP_GPIO8_IRQ2_DSP1	Wakeup dependency from GPIO8 module (SWakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
11	WKUPDEP_GPIO8_IRQ2_IPU2	Wakeup dependency from GPIO8 module (SWakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
10	WKUPDEP_GPIO8_IRQ2_MPU	Wakeup dependency from GPIO8 module (SWakeup signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
9	WKUPDEP_GPIO8_IRQ1_EVE4	Wakeup dependency from GPIO8 module (SWakeup signal) towards EVE4 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
8	WKUPDEP_GPIO8_IRQ1_EVE3	Wakeup dependency from GPIO8 (SWakeup signal) towards EVE3 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
7	WKUPDEP_GPIO8_IRQ1_EVE2	Wakeup dependency from GPIO8 module (SWakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_GPIO8_IRQ1_EVE1	Wakeup dependency from GPIO8 module (SWakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_GPIO8_IRQ1_DSP2	Wakeup dependency from GPIO8 module (SWakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_GPIO8_IRQ1_IPU1	Wakeup dependency from GPIO8 module (SWakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	RESERVED		R	0x0
2	WKUPDEP_GPIO8_IRQ1_DSP1	Wakeup dependency from GPIO8 module (SWakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_GPIO8_IRQ1_IPU2	5Wakeup dependency from GPIO8 module (SWakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_GPIO8_IRQ1_MPU	Wakeup dependency from GPIO8 module (SWakeup signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1582. RM_L4PER_GPIO8_CONTEXT

Address Offset	0x0000 011C		
Physical Address	0x4AE0 751C	**Instance**	L4PER_PRM
Description	This register contains dedicated GPIO8 context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														LOSTCONTEXT_RFF	RESERVED

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4PER_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
0	RESERVED		R	0x0

Table 3-1583. PM_L4PER_MMC3_WKDEP

Address Offset	0x0000 0120		
Physical Address	0x4AE0 7520	**Instance**	L4PER_PRM
Description	This register controls wakeup dependency based on MMC3 service requests.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						WKUPDEP_MMC3_EVE4	WKUPDEP_MMC3_EVE3	WKUPDEP_MMC3_EVE2	WKUPDEP_MMC3_EVE1	WKUPDEP_MMC3_DSP2	WKUPDEP_MMC3_IPU1	WKUPDEP_MMC3_SDMA	WKUPDEP_MMC3_DSP1	WKUPDEP_MMC3_IPU2	WKUPDEP_MMC3_MPU

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	WKUPDEP_MMC3_EVE4	Wakeup dependency from MMC3 module (SWakeup signal) towards EVE4 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
8	WKUPDEP_MMC3_EVE3	Wakeup dependency from MMC3 module (SWakeup signal) towards EVE3 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
7	WKUPDEP_MMC3_EVE2	Wakeup dependency from MMC3 module (SWakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_MMC3_EVE1	Wakeup dependency from MMC3 module (SWakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_MMC3_DSP2	Wakeup dependency from MMC3 module (SWakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_MMC3_IPU1	Wakeup dependency from MMC3 module (SWakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	WKUPDEP_MMC3_SDMA	Wakeup dependency from MMC3 module (SWakeup signal) towards SDMA + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
2	WKUPDEP_MMC3_DSP1	Wakeup dependency from MMC3 module (SWakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_MMC3_IPU2	Wakeup dependency from MMC3 module (SWakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_MMC3_MPU	Wakeup dependency from MMC3 module (SWakeup signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1584. RM_L4PER_MMC3_CONTEXT

Address Offset	0x0000 0124		
Physical Address	0x4AE0 7524	**Instance**	L4PER_PRM
Description	This register contains dedicated MMC3 context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							LOSTMEM_NONRETAINED_BANK	RESERVED							LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	LOSTMEM_NONRETAINED_BANK	Specify if memory-based context in NONRETAINED_BANK memory bank has been lost due to a previous power transition or other reset source. 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
7:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4PER_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1585. PM_L4PER_MMC4_WKDEP

Address Offset	0x0000 0128		
Physical Address	0x4AE0 7528	**Instance**	L4PER_PRM
Description	This register controls wakeup dependency based on MMC4 service requests.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						WKUPDEP_MMC4_EVE4	WKUPDEP_MMC4_EVE3	WKUPDEP_MMC4_EVE2	WKUPDEP_MMC4_EVE1	WKUPDEP_MMC4_DSP2	WKUPDEP_MMC4_IPU1	WKUPDEP_MMC4_SDMA	WKUPDEP_MMC4_DSP1	WKUPDEP_MMC4_IPU2	WKUPDEP_MMC4_MPU

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	WKUPDEP_MMC4_EVE4	Wakeup dependency from MMC4 module (SWakeup signal) towards EVE4 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
8	WKUPDEP_MMC4_EVE3	Wakeup dependency from MMC4 module (SWakeup signal) towards EVE3 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
7	WKUPDEP_MMC4_EVE2	Wakeup dependency from MMC4 module (SWakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_MMC4_EVE1	Wakeup dependency from MMC4 module (SWakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_MMC4_DSP2	Wakeup dependency from MMC4 module (SWakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_MMC4_IPU1	Wakeup dependency from MMC4 module (SWakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	WKUPDEP_MMC4_SDMA	Wakeup dependency from MMC4 module (SWakeup signal) towards SDMA + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
2	WKUPDEP_MMC4_DSP1	Wakeup dependency from MMC4 module (SWakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_MMC4_IPU2	Wakeup dependency from MMC4 module (SWakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_MMC4_MPU	Wakeup dependency from MMC4 module (SWakeup signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1586. Register Call Summary for Register PM_L4PER_MMC4_WKDEP

Clock Management Functional Description • Clock Domain Dependency: [0][1][2][3]
PRCM Register Manual • Not Supported Functionality (Registers and Bitfields): [4][5][6][7][8][9] • L4PER_PRM Register Summary: [10]

Table 3-1587. RM_L4PER_MMC4_CONTEXT

Address Offset	0x0000 012C		
Physical Address	0x4AE0 752C	**Instance**	L4PER_PRM
Description	This register contains dedicated MMC4 context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							LOSTMEM_NONRETAINED_BANK	RESERVED							LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	LOSTMEM_NONRETAINED_BANK	Specify if memory-based context in NONRETAINED_BANK memory bank has been lost due to a previous power transition or other reset source. 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
7:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4PER_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1588. Register Call Summary for Register RM_L4PER_MMC4_CONTEXT

Power Management Functional Description
PRCM Register Manual • L4PER_PRM Register Summary: [1]

Table 3-1589. PM_L4PER_DCC4_WKDEP

Address Offset	0x0000 0130		
Physical Address	0x4AE0 7530	**Instance**	L4PER_PRM
Description	This register controls wakeup dependency based on DCC4 service requests.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						WKUPDEP_DCC4_EVE4	WKUPDEP_DCC4_EVE3	WKUPDEP_DCC4_EVE2	WKUPDEP_DCC4_EVE1	WKUPDEP_DCC4_DSP2	WKUPDEP_DCC4_IPU1	RESERVED	WKUPDEP_DCC4_DSP1	WKUPDEP_DCC4_IPU2	WKUPDEP_DCC4_MPU

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	WKUPDEP_DCC4_EVE4	Wakeup dependency from DCC4 module (SWakeup IRQ signal) towards EVE4 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
8	WKUPDEP_DCC4_EVE3	Wakeup dependency from DCC4 module (SWakeup IRQ signal) towards EVE3 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
7	WKUPDEP_DCC4_EVE2	Wakeup dependency from DCC4 module (SWakeup IRQ signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_DCC4_EVE1	Wakeup dependency from DCC4 module (SWakeup IRQ signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_DCC4_DSP2	Wakeup dependency from DCC4 module (SWakeup IRQ signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_DCC4_IPU1	Wakeup dependency from DCC4 module (SWakeup IRQ signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	RESERVED		R	0x0
2	WKUPDEP_DCC4_DSP1	Wakeup dependency from DCC4 module (SWakeup IRQ signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_DCC4_IPU2	Wakeup dependency from DCC4 module (SWakeup IRQ signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
0	WKUPDEP_DCC4_MPU	6Wakeup dependency from DCC1 module (SWakeup IRQ signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1590. RM_L4PER3_DCC4_CONTEXT

Address Offset	0x0000 0134		
Physical Address	0x4AE0 7534	**Instance**	L4PER_PRM
Description	This register contains dedicated DCC4 context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4PER_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1591. Register Call Summary for Register RM_L4PER3_DCC4_CONTEXT

Power Management Functional Description

PRCM Register Manual

- L4PER_PRM Register Summary: [1]

Table 3-1592. PM_L4PER2_QSPI_WKDEP

Address Offset	0x0000 0138		
Physical Address	0x4AE0 7538	**Instance**	L4PER_PRM
Description	This register controls wakeup dependency based on QSPI service requests.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						WKUPDEP_QSPI_EVE4	WKUPDEP_QSPI_EVE3	WKUPDEP_QSPI_EVE2	WKUPDEP_QSPI_EVE1	WKUPDEP_QSPI_DSP2	WKUPDEP_QSPI_IPU1	RESERVED	WKUPDEP_QSPI_DSP1	WKUPDEP_QSPI_IPU2	WKUPDEP_QSPI_MPU

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	WKUPDEP_QSPI_EVE4	Wakeup dependency from QSPI module (SWakeup IRQ signal) towards EVE4 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
8	WKUPDEP_QSPI_EVE3	Wakeup dependency from QSPI module (SWakeup IRQ signal) towards EVE3 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
7	WKUPDEP_QSPI_EVE2	Wakeup dependency from QSPI module (SWakeup IRQ signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_QSPI_EVE1	Wakeup dependency from QSPI module (SWakeup IRQ signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_QSPI_DSP2	Wakeup dependency from QSPI module (SWakeup IRQ signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_QSPI_IPU1	Wakeup dependency from QSPI module (SWakeup IRQ signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	RESERVED		R	0x0
2	WKUPDEP_QSPI_DSP1	Wakeup dependency from QSPI module (SWakeup IRQ signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_QSPI_IPU2	Wakeup dependency from QSPI module (SWakeup IRQ signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_QSPI_MPU	Wakeup dependency from QSPI module (SWakeup IRQ signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1593. Register Call Summary for Register PM_L4PER2_QSPI_WKDEP

Clock Management Functional Description
- Clock Domain Dependency: [0][1][2][3]

PRCM Register Manual
- Not Supported Functionality (Registers and Bitfields): [4][5][6][7][8]
- L4PER_PRM Register Summary: [9]

Table 3-1594. RM_L4PER2_QSPI_CONTEXT

Address Offset	0x0000 013C		
Physical Address	0x4AE0 753C	**Instance**	L4PER_PRM
Description	This register contains dedicated QSPI context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CORE_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1595. Register Call Summary for Register RM_L4PER2_QSPI_CONTEXT

Power Management Functional Description

PRCM Register Manual

- L4PER_PRM Register Summary: [1]

Table 3-1596. PM_L4PER_UART1_WKDEP

Address Offset	0x0000 0140		
Physical Address	0x4AE0 7540	**Instance**	L4PER_PRM
Description	This register controls wakeup dependency based on UART1 service requests.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						WKUPDEP_UART1_EVE4	WKUPDEP_UART1_EVE3	WKUPDEP_UART1_EVE2	WKUPDEP_UART1_EVE1	WKUPDEP_UART1_DSP2	WKUPDEP_UART1_IPU1	WKUPDEP_UART1_SDMA	WKUPDEP_UART1_DSP1	WKUPDEP_UART1_IPU2	WKUPDEP_UART1_MPU								

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	WKUPDEP_UART1_EVE4	Wakeup dependency from UART1 module (SWakeup signal) towards EVE4 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
8	WKUPDEP_UART1_EVE3	Wakeup dependency from UART1 module (SWakeup signal) towards EVE3 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
7	WKUPDEP_UART1_EVE2	Wakeup dependency from UART1 module (SWakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_UART1_EVE1	Wakeup dependency from UART1 module (SWakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_UART1_DSP2	Wakeup dependency from UART1 module (SWakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_UART1_IPU1	Wakeup dependency from UART1 module (SWakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	WKUPDEP_UART1_SDMA	Wakeup dependency from UART1 module (SWakeup signal) towards SDMA + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
2	WKUPDEP_UART1_DSP1	Wakeup dependency from UART1 module (SWakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_UART1_IPU2	Wakeup dependency from UART1 module (SWakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_UART1_MPU	Wakeup dependency from UART1 module (SWakeup signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1597. Register Call Summary for Register PM_L4PER_UART1_WKDEP

Clock Management Functional Description
• Clock Domain Dependency: [0][1][2][3]
PRCM Register Manual
• Not Supported Functionality (Registers and Bitfields): [4][5][6][7][8][9]
• L4PER_PRM Register Summary: [10]

Table 3-1598. RM_L4PER_UART1_CONTEXT

Address Offset	0x0000 0144		
Physical Address	0x4AE0 7544	**Instance**	L4PER_PRM
Description	This register contains dedicated UART1 context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							LOSTMEM_RETAINED_BANK	RESERVED						LOSTCONTEXT_RFF	RESERVED

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	LOSTMEM_RETAINED_BANK	Specify if memory-based context in RETAINED_BANK memory bank has been lost due to a previous power transition or other reset source. 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
7:2	RESERVED		R	0x0
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4PER_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
0	RESERVED		R	0x0

Table 3-1599. Register Call Summary for Register RM_L4PER_UART1_CONTEXT

Power Management Functional Description
PRCM Register Manual
• L4PER_PRM Register Summary: [1]

Table 3-1600. PM_L4PER_UART2_WKDEP

Address Offset	0x0000 0148		
Physical Address	0x4AE0 7548	**Instance**	L4PER_PRM
Description	This register controls wakeup dependency based on UART2 service requests.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						WKUPDEP_UART2_EVE4	WKUPDEP_UART2_EVE3	WKUPDEP_UART2_EVE2	WKUPDEP_UART2_EVE1	WKUPDEP_UART2_DSP2	WKUPDEP_UART2_IPU1	WKUPDEP_UART2_SDMA	WKUPDEP_UART2_DSP1	WKUPDEP_UART2_IPU2	WKUPDEP_UART2_MPU

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	WKUPDEP_UART2_EVE4	Wakeup dependency from UART2 module (SWakeup signal) towards EVE4 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
8	WKUPDEP_UART2_EVE3	Wakeup dependency from UART2 module (SWakeup signal) towards EVE3 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
7	WKUPDEP_UART2_EVE2	Wakeup dependency from UART2 module (SWakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_UART2_EVE1	Wakeup dependency from UART2 module (SWakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_UART2_DSP2	Wakeup dependency from UART2 module (SWakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_UART2_IPU1	Wakeup dependency from UART2 module (SWakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	WKUPDEP_UART2_SDMA	2Wakeup dependency from UART1 module (SWakeup signal) towards SDMA + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
2	WKUPDEP_UART2_DSP1	Wakeup dependency from UART2 module (SWakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
1	WKUPDEP_UART2_IPU2	Wakeup dependency from UART2 module (SWakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_UART2_MPU	Wakeup dependency from UART2 module (SWakeup signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1601. Register Call Summary for Register PM_L4PER_UART2_WKDEP

Clock Management Functional Description
- Clock Domain Dependency: [0][1][2][3]

PRCM Register Manual
- Not Supported Functionality (Registers and Bitfields): [4][5][6][7][8][9]
- L4PER_PRM Register Summary: [10]

Table 3-1602. RM_L4PER_UART2_CONTEXT

Address Offset	0x0000 014C		
Physical Address	0x4AE0 754C	**Instance**	L4PER_PRM
Description	This register contains dedicated UART2 context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							LOSTMEM_RETAINED_BANK	RESERVED						LOSTCONTEXT_RFF	RESERVED

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	LOSTMEM_RETAINED_BANK	Specify if memory-based context in RETAINED_BANK memory bank has been lost due to a previous power transition or other reset source. 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
7:2	RESERVED		R	0x0
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4PER_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
0	RESERVED		R	0x0

Table 3-1603. Register Call Summary for Register RM_L4PER_UART2_CONTEXT

Power Management Functional Description
PRCM Register Manual • L4PER_PRM Register Summary: [1]

Table 3-1604. PM_L4PER_UART3_WKDEP

Address Offset	0x0000 0150		
Physical Address	0x4AE0 7550	**Instance**	L4PER_PRM
Description	This register controls wakeup dependency based on UART3 service requests.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						WKUPDEP_UART3_EVE4	WKUPDEP_UART3_EVE3	WKUPDEP_UART3_EVE2	WKUPDEP_UART3_EVE1	WKUPDEP_UART3_DSP2	WKUPDEP_UART3_IPU1	WKUPDEP_UART3_SDMA	WKUPDEP_UART3_DSP1	WKUPDEP_UART3_IPU2	WKUPDEP_UART3_MPU

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	WKUPDEP_UART3_EVE4	Wakeup dependency from UART3 module (SWakeup signal) towards EVE4 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
8	WKUPDEP_UART3_EVE3	Wakeup dependency from UART3 module (SWakeup signal) towards EVE3 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
7	WKUPDEP_UART3_EVE2	Wakeup dependency from UART3 module (SWakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_UART3_EVE1	Wakeup dependency from UART3 module (SWakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_UART3_DSP2	Wakeup dependency from UART3 module (SWakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
4	WKUPDEP_UART3_IPU1	Wakeup dependency from UART3 module (SWakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	WKUPDEP_UART3_SDMA	Wakeup dependency from UART3 module (SWakeup signal) towards SDMA + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
2	WKUPDEP_UART3_DSP1	Wakeup dependency from UART3 module (SWakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_UART3_IPU2	Wakeup dependency from UART3 module (SWakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_UART3_MPU	Wakeup dependency from UART3 module (SWakeup signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1605. Register Call Summary for Register PM_L4PER_UART3_WKDEP

Clock Management Functional Description
- Clock Domain Dependency: [0][1][2][3]

PRCM Register Manual
- Not Supported Functionality (Registers and Bitfields): [4][5][6][7][8][9]
- L4PER_PRM Register Summary: [10]

Table 3-1606. RM_L4PER_UART3_CONTEXT

Address Offset	0x0000 0154		
Physical Address	0x4AE0 7554	**Instance**	L4PER_PRM
Description	This register contains dedicated UART3 context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							LOSTMEM_RETAINED_BANK	RESERVED						LOSTCONTEXT_RFF	RESERVED

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	LOSTMEM_RETAINED_BANK	Specify if memory-based context in RETAINED_BANK memory bank has been lost due to a previous power transition or other reset source. 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
7:2	RESERVED		R	0x0
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4PER_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
0	RESERVED		R	0x0

Table 3-1607. Register Call Summary for Register RM_L4PER_UART3_CONTEXT

Power Management Functional Description
PRCM Register Manual • L4PER_PRM Register Summary: [1]

Table 3-1608. PM_L4PER_UART4_WKDEP

Address Offset	0x0000 0158		
Physical Address	0x4AE0 7558	**Instance**	L4PER_PRM
Description	This register controls wakeup dependency based on UART4 service requests.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						WKUPDEP_UART4_EVE4	WKUPDEP_UART4_EVE3	WKUPDEP_UART4_EVE2	WKUPDEP_UART4_EVE1	WKUPDEP_UART4_DSP2	WKUPDEP_UART4_IPU1	WKUPDEP_UART4_SDMA	WKUPDEP_UART4_DSP1	WKUPDEP_UART4_IPU2	WKUPDEP_UART4_MPU								

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	WKUPDEP_UART4_EVE4	Wakeup dependency from UART4 module (SWakeup signal) towards EVE4 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
8	WKUPDEP_UART4_EVE3	Wakeup dependency from UART4 module (SWakeup signal) towards EVE3 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
7	WKUPDEP_UART4_EVE2	Wakeup dependency from UART4 module (SWakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_UART4_EVE1	Wakeup dependency from UART4 module (SWakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_UART4_DSP2	Wakeup dependency from UART4 module (SWakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_UART4_IPU1	Wakeup dependency from UART4 module (SWakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	WKUPDEP_UART4_SDMA	Wakeup dependency from UART4 module (SWakeup signal) towards SDMA + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
2	WKUPDEP_UART4_DSP1	Wakeup dependency from UART4 module (SWakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_UART4_IPU2	Wakeup dependency from UART4 module (SWakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_UART4_MPU	Wakeup dependency from UART4 module (SWakeup signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1609. Register Call Summary for Register PM_L4PER_UART4_WKDEP

PRCM Register Manual
- L4PER_PRM Register Summary: [0]

Table 3-1610. RM_L4PER_UART4_CONTEXT

Address Offset	0x0000 015C		
Physical Address	0x4AE0 755C	**Instance**	L4PER_PRM
Description	This register contains dedicated UART4 context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							LOSTMEM_RETAINED_BANK	RESERVED						LOSTCONTEXT_RFF	RESERVED

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	LOSTMEM_RETAINED_BANK	Specify if memory-based context in RETAINED_BANK memory bank has been lost due to a previous power transition or other reset source. 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
7:2	RESERVED		R	0x0
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4PER_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
0	RESERVED		R	0x0

Table 3-1611. Register Call Summary for Register RM_L4PER_UART4_CONTEXT

PRCM Register Manual
- L4PER_PRM Register Summary: [0]

Table 3-1612. PM_L4PER2_ADC_WKDEP

Address Offset	0x0000 0160		
Physical Address	0x4AE0 7560	**Instance**	L4PER_PRM
Description	This register controls wakeup dependency based on ADC service requests.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																WKUPDEP_ADC_DMA_DSP2	RESERVED	WKUPDEP_ADC_DMA_SDMA	WKUPDEP_ADC_DMA_DSP1	RESERVED		WKUPDEP_ADC_IRQ_EVE4	WKUPDEP_ADC_IRQ_EVE3	WKUPDEP_ADC_IRQ_EVE2	WKUPDEP_ADC_IRQ_EVE1	WKUPDEP_ADC_IRQ_DSP2	WKUPDEP_ADC_IRQ_IPU1	RESERVED	WKUPDEP_ADC_IRQ_DSP1	WKUPDEP_ADC_IRQ_IPU2	WKUPDEP_ADC_IRQ_MPU

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0
15	WKUPDEP_ADC_DMA_DSP2	Wakeup dependency from ADC module (SWakeup_dma signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
14	RESERVED		R	0x0
13	WKUPDEP_ADC_DMA_SDMA	Wakeup dependency from ADC module (SWakeup_dma signal) towards SDMA + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
12	WKUPDEP_ADC_DMA_DSP1	Wakeup dependency from ADC module (SWakeup_dma signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
11:10	RESERVED		R	0x0
9	WKUPDEP_ADC_IRQ_EVE4	Wakeup dependency from ADC module (SWakeup IRQ signal) towards EVE4 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
8	WKUPDEP_ADC_IRQ_EVE3	Wakeup dependency from ADC module (SWakeup IRQ signal) towards EVE3 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
7	WKUPDEP_ADC_IRQ_EVE2	Wakeup dependency from ADC module (SWakeup IRQ signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_ADC_IRQ_EVE1	Wakeup dependency from ADC module (SWakeup IRQ signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_ADC_IRQ_DSP2	Wakeup dependency from ADC module (SWakeup IRQ signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_ADC_IRQ_IPU1	Wakeup dependency from ADC module (SWakeup IRQ signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	RESERVED		R	0x0
2	WKUPDEP_ADC_IRQ_DSP1	Wakeup dependency from ADC module (SWakeup IRQ signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
1	WKUPDEP_ADC_IRQ_IPU2	Wakeup dependency from ADC module (SWakeup IRQ signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_ADC_IRQ_MPU	Wakeup dependency from ADC module (SWakeup IRQ signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1613. Register Call Summary for Register PM_L4PER2_ADC_WKDEP

Clock Management Functional Description • Clock Domain Dependency: [0][1][2][3][4][5]
PRCM Register Manual • Not Supported Functionality (Registers and Bitfields): [6][7][8][9][10][11] • L4PER_PRM Register Summary: [12]

Table 3-1614. RM_L4PER2_ADC_CONTEXT

Address Offset	0x0000 0164		
Physical Address	0x4AE0 7564	**Instance**	L4PER_PRM
Description	This register contains dedicated ADC context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4PER_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1615. Register Call Summary for Register RM_L4PER2_ADC_CONTEXT

Power Management Functional Description
PRCM Register Manual • L4PER_PRM Register Summary: [2]

Table 3-1616. PM_L4PER2_MCASP3_WKDEP

Address Offset	0x0000 0168		
Physical Address	0x4AE0 7568	**Instance**	L4PER_PRM
Description	This register controls wakeup dependency based on McASP3 service requests.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																WKUPDEP_MCASP3_DMA_DSP2	RESERVED	WKUPDEP_MCASP3_DMA_SDMA	WKUPDEP_MCASP3_DMA_DSP1	RESERVED		WKUPDEP_MCASP3_IRQ_EVE4	WKUPDEP_MCASP3_IRQ_EVE3	WKUPDEP_MCASP3_IRQ_EVE2	WKUPDEP_MCASP3_IRQ_EVE1	WKUPDEP_MCASP3_IRQ_DSP2	WKUPDEP_MCASP3_IRQ_IPU1	RESERVED	WKUPDEP_MCASP3_IRQ_DSP1	WKUPDEP_MCASP3_IRQ_IPU2	WKUPDEP_MCASP3_IRQ_MPU

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0
15	WKUPDEP_MCASP3_DMA_DSP2	Wakeup dependency from McASP3 module (SWakeup_dma signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
14	RESERVED		R	0x0
13	WKUPDEP_MCASP3_DMA_SDMA	Wakeup dependency from McASP3 module (SWakeup_dma signal) towards SDMA + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
12	WKUPDEP_MCASP3_DMA_DSP1	3Wakeup dependency from ADC module (SWakeup_dma signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
11:10	RESERVED		R	0x0
9	WKUPDEP_MCASP3_IRQ_EVE4	Wakeup dependency from McASP3 module (SWakeup IRQ signal) towards EVE4 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
8	WKUPDEP_MCASP3_IRQ_EVE3	Wakeup dependency from McASP3 module (SWakeup IRQ signal) towards EVE3 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
7	WKUPDEP_MCASP3_IRQ_EVE2	Wakeup dependency from McASP3 module (SWakeup IRQ signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
6	WKUPDEP_MCASP3_IRQ_EVE1	Wakeup dependency from McASP3 module (SWakeup IRQ signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_MCASP3_IRQ_DSP2	Wakeup dependency from McASP3 module (SWakeup IRQ signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_MCASP3_IRQ_IPU1	Wakeup dependency from McASP3 module (SWakeup IRQ signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	RESERVED		R	0x0
2	WKUPDEP_MCASP3_IRQ_DSP1	Wakeup dependency from McASP3 module (SWakeup IRQ signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_MCASP3_IRQ_IPU2	Wakeup dependency from McASP3 module (SWakeup IRQ signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_MCASP3_IRQ_MPU	Wakeup dependency from McASP3 module (SWakeup IRQ signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1617. RM_L4PER2_MCASP3_CONTEXT

Address Offset	0x0000 016C		
Physical Address	0x4AE0 756C	**Instance**	L4PER_PRM
Description	This register contains dedicated McASP3 context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4PER_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1618. PM_L4PER_UART5_WKDEP

Address Offset	0x0000 0170		
Physical Address	0x4AE0 7570	**Instance**	L4PER_PRM
Description	This register controls wakeup dependency based on UART5 service requests.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						WKUPDEP_UART5_EVE4	WKUPDEP_UART5_EVE3	WKUPDEP_UART5_EVE2	WKUPDEP_UART5_EVE1	WKUPDEP_UART5_DSP2	WKUPDEP_UART5_IPU1	WKUPDEP_UART5_SDMA	WKUPDEP_UART5_DSP1	WKUPDEP_UART5_IPU2	WKUPDEP_UART5_MPU

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	WKUPDEP_UART5_EVE4	Wakeup dependency from UART5 module (SWakeup signal) towards EVE4 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
8	WKUPDEP_UART5_EVE3	Wakeup dependency from UART5 module (SWakeup signal) towards EVE3 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
7	WKUPDEP_UART5_EVE2	Wakeup dependency from UART5 module (SWakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_UART5_EVE1	Wakeup dependency from UART5 module (SWakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_UART5_DSP2	Wakeup dependency from UART5 module (SWakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
4	WKUPDEP_UART5_IPU1	Wakeup dependency from UART5 module (SWakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	WKUPDEP_UART5_SDMA	Wakeup dependency from UART5 module (SWakeup signal) towards SDMA + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
2	WKUPDEP_UART5_DSP1	Wakeup dependency from UART5 module (SWakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_UART5_IPU2	Wakeup dependency from UART5 module (SWakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_UART5_MPU	Wakeup dependency from UART5 module (SWakeup signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1619. RM_L4PER_UART5_CONTEXT

Address Offset	0x0000 0174		
Physical Address	0x4AE0 7574	**Instance**	L4PER_PRM
Description	This register contains dedicated UART5 context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							LOSTMEM_RETAINED_BANK	RESERVED						LOSTCONTEXT_RFF	RESERVED

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	LOSTMEM_RETAINED_BANK	Specify if memory-based context in RETAINED_BANK memory bank has been lost due to a previous power transition or other reset source. 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
7:2	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4PER_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
0	RESERVED		R	0x0

Table 3-1620. PM_L4PER2_MCASP5_WKDEP

Address Offset	0x0000 0178		
Physical Address	0x4AE0 7578	**Instance**	L4PER_PRM
Description	This register controls wakeup dependency based on McASP5 service requests.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																WKUPDEP_MCASP5_DMA_DSP2	RESERVED	WKUPDEP_MCASP5_DMA_SDMA	WKUPDEP_MCASP5_DMA_DSP1	RESERVED		WKUPDEP_MCASP5_IRQ_EVE4	WKUPDEP_MCASP5_IRQ_EVE3	WKUPDEP_MCASP5_IRQ_EVE2	WKUPDEP_MCASP5_IRQ_EVE1	WKUPDEP_MCASP5_IRQ_DSP2	WKUPDEP_MCASP5_IRQ_IPU1	RESERVED	WKUPDEP_MCASP5_IRQ_DSP1	WKUPDEP_MCASP5_IRQ_IPU2	WKUPDEP_MCASP5_IRQ_MPU

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0
15	WKUPDEP_MCASP5_DMA_DSP2	Wakeup dependency from McASP5 module (SWakeup_dma signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
14	RESERVED		R	0x0
13	WKUPDEP_MCASP5_DMA_SDMA	Wakeup dependency from McASP5 module (SWakeup_dma signal) towards SDMA + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
12	WKUPDEP_MCASP5_DMA_DSP1	Wakeup dependency from McASP5 module (SWakeup_dma signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
11:10	RESERVED		R	0x0
9	WKUPDEP_MCASP5_IRQ_EVE4	Wakeup dependency from McASP5 module (SWakeup IRQ signal) towards EVE4 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
8	WKUPDEP_MCASP5_IRQ_EVE3	Wakeup dependency from McASP5 module (SWakeup IRQ signal) towards EVE3 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
7	WKUPDEP_MCASP5_IRQ_EVE2	Wakeup dependency from McASP5 module (SWakeup IRQ signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_MCASP5_IRQ_EVE1	Wakeup dependency from McASP5 module (SWakeup IRQ signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_MCASP5_IRQ_DSP2	Wakeup dependency from McASP5 module (SWakeup IRQ signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_MCASP5_IRQ_IPU1	Wakeup dependency from McASP5 module (SWakeup IRQ signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	RESERVED		R	0x0
2	WKUPDEP_MCASP5_IRQ_DSP1	Wakeup dependency from McASP5 module (SWakeup IRQ signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_MCASP5_IRQ_IPU2	Wakeup dependency from ADC module (SWakeup IRQ signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_MCASP5_IRQ_MPU	Wakeup dependency from McASP5 module (SWakeup IRQ signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1621. RM_L4PER2_MCASP5_CONTEXT

Address Offset	0x0000 017C		
Physical Address	0x4AE0 757C	**Instance**	L4PER_PRM
Description	This register contains dedicated McASP5 context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4PER_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1622. PM_L4PER2_MCASP6_WKDEP

Address Offset	0x0000 0180		
Physical Address	0x4AE0 7580	**Instance**	L4PER_PRM
Description	This register controls wakeup dependency based on McASP6 service requests.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																WKUPDEP_MCASP6_DMA_DSP2	RESERVED	WKUPDEP_MCASP6_DMA_SDMA	WKUPDEP_MCASP6_DMA_DSP1	RESERVED		WKUPDEP_MCASP6_IRQ_EVE4	WKUPDEP_MCASP6_IRQ_EVE3	WKUPDEP_MCASP6_IRQ_EVE2	WKUPDEP_MCASP6_IRQ_EVE1	WKUPDEP_MCASP6_IRQ_DSP2	WKUPDEP_MCASP6_IRQ_IPU1	RESERVED	WKUPDEP_MCASP6_IRQ_DSP1	WKUPDEP_MCASP6_IRQ_IPU2	WKUPDEP_MCASP6_IRQ_MPU

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0
15	WKUPDEP_MCASP6_DMA_DSP2	Wakeup dependency from McASP6 module (SWakeup_dma signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
14	RESERVED		R	0x0
13	WKUPDEP_MCASP6_DMA_SDMA	Wakeup dependency from McASP6 module (SWakeup_dma signal) towards SDMA + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1

Bits	Field Name	Description	Type	Reset
12	WKUPDEP_MCASP6_DMA_DSP1	Wakeup dependency from McASP6 module (SWakeup_dma signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
11:10	RESERVED		R	0x0
9	WKUPDEP_MCASP6_IRQ_EVE4	Wakeup dependency from McASP6 module (SWakeup IRQ signal) towards EVE4 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
8	WKUPDEP_MCASP6_IRQ_EVE3	Wakeup dependency from McASP6 module (SWakeup IRQ signal) towards EVE3 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
7	WKUPDEP_MCASP6_IRQ_EVE2	Wakeup dependency from McASP6 module (SWakeup IRQ signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_MCASP6_IRQ_EVE1	Wakeup dependency from McASP6 module (SWakeup IRQ signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_MCASP6_IRQ_DSP2	Wakeup dependency from McASP6 module (SWakeup IRQ signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_MCASP6_IRQ_IPU1	Wakeup dependency from McASP6 module (SWakeup IRQ signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	RESERVED		R	0x0
2	WKUPDEP_MCASP6_IRQ_DSP1	Wakeup dependency from McASP6 module (SWakeup IRQ signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_MCASP6_IRQ_IPU2	Wakeup dependency from McASP6 (SWakeup IRQ signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_MCASP6_IRQ_MPU	Wakeup dependency from McASP6 module (SWakeup IRQ signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1623. RM_L4PER2_MCASP6_CONTEXT

Address Offset	0x0000 0184		
Physical Address	0x4AE0 7584	**Instance**	L4PER_PRM
Description	This register contains dedicated McASP6 context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4PER_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1624. PM_L4PER2_MCASP7_WKDEP

Address Offset	0x0000 0188		
Physical Address	0x4AE0 7588	**Instance**	L4PER_PRM
Description	This register controls wakeup dependency based on McASP7 service requests.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																WKUPDEP_MCASP7_DMA_DSP2	RESERVED	WKUPDEP_MCASP7_DMA_SDMA	WKUPDEP_MCASP7_DMA_DSP1	RESERVED		WKUPDEP_MCASP7_IRQ_EVE4	WKUPDEP_MCASP7_IRQ_EVE3	WKUPDEP_MCASP7_IRQ_EVE2	WKUPDEP_MCASP7_IRQ_EVE1	WKUPDEP_MCASP7_IRQ_DSP2	WKUPDEP_MCASP7_IRQ_IPU1	RESERVED	WKUPDEP_MCASP7_IRQ_DSP1	WKUPDEP_MCASP7_IRQ_IPU2	WKUPDEP_MCASP7_IRQ_MPU

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0
15	WKUPDEP_MCASP7_DMA_DSP2	Wakeup dependency from McASP7 module (SWakeup_dma signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
14	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
13	WKUPDEP_MCASP7_DMA_SDMA	Wakeup dependency from McASP7 module (SWakeup_dma signal) towards SDMA + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
12	WKUPDEP_MCASP7_DMA_DSP1	Wakeup dependency from McASP7 module (SWakeup_dma signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
11:10	RESERVED		R	0x0
9	WKUPDEP_MCASP7_IRQ_EVE4	Wakeup dependency from McASP7 module (SWakeup IRQ signal) towards EVE4 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
8	WKUPDEP_MCASP7_IRQ_EVE3	Wakeup dependency from McASP7 module (SWakeup IRQ signal) towards EVE3 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
7	WKUPDEP_MCASP7_IRQ_EVE2	Wakeup dependency from McASP7 module (SWakeup IRQ signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_MCASP7_IRQ_EVE1	Wakeup dependency from McASP7 module (SWakeup IRQ signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_MCASP7_IRQ_DSP2	Wakeup dependency from McASP7 module (SWakeup IRQ signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_MCASP7_IRQ_IPU1	Wakeup dependency from McASP7 module (SWakeup IRQ signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	RESERVED		R	0x0
2	WKUPDEP_MCASP7_IRQ_DSP1	Wakeup dependency from McASP7 module (SWakeup IRQ signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_MCASP7_IRQ_IPU2	Wakeup dependency from McASP7 module (SWakeup IRQ signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
0	WKUPDEP_MCASP7_IRQ_MPU	Wakeup dependency from McASP7 module (SWakeup IRQ signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1625. RM_L4PER2_MCASP7_CONTEXT

Address Offset	0x0000 018C		
Physical Address	0x4AE0 758C	**Instance**	L4PER_PRM
Description	This register contains dedicated McASP7 context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4PER_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1626. PM_L4PER2_MCASP8_WKDEP

Address Offset	0x0000 0190		
Physical Address	0x4AE0 7590	**Instance**	L4PER_PRM
Description	This register controls wakeup dependency based on McASP8 service requests.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																WKUPDEP_MCASP8_DMA_DSP2	RESERVED	WKUPDEP_MCASP8_DMA_SDMA	WKUPDEP_MCASP8_DMA_DSP1	RESERVED		WKUPDEP_MCASP8_IRQ_EVE4	WKUPDEP_MCASP8_IRQ_EVE3	WKUPDEP_MCASP8_IRQ_EVE2	WKUPDEP_MCASP8_IRQ_EVE1	WKUPDEP_MCASP8_IRQ_DSP2	WKUPDEP_MCASP8_IRQ_IPU1	RESERVED	WKUPDEP_MCASP8_IRQ_DSP1	WKUPDEP_MCASP8_IRQ_IPU2	WKUPDEP_MCASP8_IRQ_MPU

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0
15	WKUPDEP_MCASP8_DMA_DSP2	Wakeup dependency from McASP8 module (SWakeup_dma signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
14	RESERVED		R	0x0
13	WKUPDEP_MCASP8_DMA_SDMA	Wakeup dependency from McASP8 module (SWakeup_dma signal) towards SDMA + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
12	WKUPDEP_MCASP8_DMA_DSP1	Wakeup dependency from McASP8 module (SWakeup_dma signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
11:10	RESERVED		R	0x0
9	WKUPDEP_MCASP8_IRQ_EVE4	Wakeup dependency from McASP8 module (SWakeup IRQ signal) towards EVE4 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
8	WKUPDEP_MCASP8_IRQ_EVE3	Wakeup dependency from McASP8 module (SWakeup IRQ signal) towards EVE3 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
7	WKUPDEP_MCASP8_IRQ_EVE2	Wakeup dependency from McASP8 module (SWakeup IRQ signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_MCASP8_IRQ_EVE1	Wakeup dependency from McASP8 module (SWakeup IRQ signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_MCASP8_IRQ_DSP2	Wakeup dependency from McASP8 module (SWakeup IRQ signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_MCASP8_IRQ_IPU1	Wakeup dependency from McASP8 module (SWakeup IRQ signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	RESERVED		R	0x0
2	WKUPDEP_MCASP8_IRQ_DSP1	Wakeup dependency from McASP8 module (SWakeup IRQ signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
1	WKUPDEP_MCASP8_IRQ_IPU2	Wakeup dependency from McASP8 module (SWakeup IRQ signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_MCASP8_IRQ_MPU	Wakeup dependency from McASP8 module (SWakeup IRQ signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1627. RM_L4PER2_MCASP8_CONTEXT

Address Offset	0x0000 0194		
Physical Address	0x4AE0 7594	**Instance**	L4PER_PRM
Description	This register contains dedicated McASP8 context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of ABE_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1628. PM_L4PER2_MCASP4_WKDEP

Address Offset	0x0000 0198		
Physical Address	0x4AE0 7598	**Instance**	L4PER_PRM
Description	This register controls wakeup dependency based on McASP4 service requests.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																WKUPDEP_MCASP4_DMA_DSP2	RESERVED	WKUPDEP_MCASP4_DMA_SDMA	WKUPDEP_MCASP4_DMA_DSP1	RESERVED		WKUPDEP_MCASP4_IRQ_EVE4	WKUPDEP_MCASP4_IRQ_EVE3	WKUPDEP_MCASP4_IRQ_EVE2	WKUPDEP_MCASP4_IRQ_EVE1	WKUPDEP_MCASP4_IRQ_DSP2	WKUPDEP_MCASP4_IRQ_IPU1	RESERVED	WKUPDEP_MCASP4_IRQ_DSP1	WKUPDEP_MCASP4_IRQ_IPU2	WKUPDEP_MCASP4_IRQ_MPU

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0
15	WKUPDEP_MCASP4_DMA_DSP2	Wakeup dependency from McASP4 module (SWakeup_dma signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
14	RESERVED		R	0x0
13	WKUPDEP_MCASP4_DMA_SDMA	Wakeup dependency from McASP4 module (SWakeup_dma signal) towards SDMA + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
12	WKUPDEP_MCASP4_DMA_DSP1	Wakeup dependency from McASP4 module (SWakeup_dma signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x1
11:10	RESERVED		R	0x0
9	WKUPDEP_MCASP4_IRQ_EVE4	Wakeup dependency from McASP4 module (SWakeup IRQ signal) towards EVE4 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
8	WKUPDEP_MCASP4_IRQ_EVE3	Wakeup dependency from McASP4 module (SWakeup IRQ signal) towards EVE3 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
7	WKUPDEP_MCASP4_IRQ_EVE2	Wakeup dependency from McASP4 module (SWakeup IRQ signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_MCASP4_IRQ_EVE1	Wakeup dependency from McASP4 module (SWakeup IRQ signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
5	WKUPDEP_MCASP4_IRQ_DSP2	Wakeup dependency from McASP4 module (SWakeup IRQ signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_MCASP4_IRQ_IPU1	Wakeup dependency from McASP4 module (SWakeup IRQ signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	RESERVED		R	0x0
2	WKUPDEP_MCASP4_IRQ_DSP1	Wakeup dependency from McASP4 module (SWakeup IRQ signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_MCASP4_IRQ_IPU2	Wakeup dependency from McASP4 module (SWakeup IRQ signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_MCASP4_IRQ_MPU	Wakeup dependency from McASP4 module (SWakeup IRQ signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1629. RM_L4PER2_MCASP4_CONTEXT

Address Offset	0x0000 019C		
Physical Address	0x4AE0 759C	**Instance**	L4PER_PRM
Description	This register contains dedicated McASP4 context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4PER_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1630. RM_L4SEC_AES1_CONTEXT

Address Offset	0x0000 01A4		
Physical Address	0x4AE0 75A4	**Instance**	L4PER_PRM
Description	This register contains dedicated AES1 context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														LOSTCONTEXT_RFF	RESERVED

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4PER_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
0	RESERVED		R	0x0

Table 3-1631. RM_L4SEC_AES2_CONTEXT

Address Offset	0x0000 01AC		
Physical Address	0x4AE0 75AC	**Instance**	L4PER_PRM
Description	This register contains dedicated AES2 context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														LOSTCONTEXT_RFF	RESERVED

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4PER_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
0	RESERVED		R	0x0

Table 3-1632. RM_L4SEC_DES3DES_CONTEXT

Address Offset	0x0000 01B4		
Physical Address	0x4AE0 75B4	**Instance**	L4PER_PRM
Description	This register contains dedicated DES3DES context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														LOSTCONTEXT_RFF	RESERVED

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4PER_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
0	RESERVED		R	0x0

Table 3-1633. RM_L4SEC_FPKA_CONTEXT

Address Offset	0x0000 01BC		
Physical Address	0x4AE0 75BC	**Instance**	L4PER_PRM
Description	This register contains dedicated FPKA context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							LOSTMEM_NONRETAINED_BANK	RESERVED							LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	LOSTMEM_NONRETAINED_BANK	Specify if memory-based context in NONRETAINED_BANK memory bank has been lost due to a previous power transition or other reset source. 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
7:1	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4PER_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1634. RM_L4SEC_RNG_CONTEXT

Address Offset	0x0000 01C4		
Physical Address	0x4AE0 75C4	**Instance**	L4PER_PRM
Description	This register contains dedicated RNG context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														LOSTCONTEXT_RFF	RESERVED

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4PER_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
0	RESERVED		R	0x0

Table 3-1635. RM_L4SEC_SHA2MD51_CONTEXT

Address Offset	0x0000 01CC		
Physical Address	0x4AE0 75CC	**Instance**	L4PER_PRM
Description	This register contains dedicated SHA2MD51 context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														LOSTCONTEXT_RFF	RESERVED

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4PER_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
0	RESERVED		R	0x0

Table 3-1636. PM_L4PER2_UART7_WKDEP

Address Offset	0x0000 01D0		
Physical Address	0x4AE0 75D0	**Instance**	L4PER_PRM
Description	This register controls wakeup dependency based on UART7 service requests.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						WKUPDEP_UART7_EVE4	WKUPDEP_UART7_EVE3	WKUPDEP_UART7_EVE2	WKUPDEP_UART7_EVE1	WKUPDEP_UART7_DSP2	WKUPDEP_UART7_IPU1	WKUPDEP_UART7_SDMA	WKUPDEP_UART7_DSP1	WKUPDEP_UART7_IPU2	WKUPDEP_UART7_MPU

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	WKUPDEP_UART7_EVE4	Wakeup dependency from UART7 module (SWakeup signal) towards EVE4 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
8	WKUPDEP_UART7_EVE3	Wakeup dependency from UART7 module (SWakeup signal) towards EVE3 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
7	WKUPDEP_UART7_EVE2	Wakeup dependency from UART7 module (SWakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_UART7_EVE1	Wakeup dependency from UART7 module (SWakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_UART7_DSP2	Wakeup dependency from UART7 module (SWakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
4	WKUPDEP_UART7_IPU1	Wakeup dependency from UART7 module (SWakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	WKUPDEP_UART7_SDMA	Wakeup dependency from UART7 module (SWakeup signal) towards SDMA + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
2	WKUPDEP_UART7_DSP1	Wakeup dependency from UART7 module (SWakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_UART7_IPU2	Wakeup dependency from UART7 module (SWakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_UART7_MPU	Wakeup dependency from UART7 module (SWakeup signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1637. RM_L4PER2_UART7_CONTEXT

Address Offset	0x0000 01D4		
Physical Address	0x4AE0 75D4	**Instance**	L4PER_PRM
Description	This register contains dedicated UART7 context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							LOSTMEM_RETAINED_BANK	RESERVED						LOSTCONTEXT_RFF	RESERVED

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	LOSTMEM_RETAINED_BANK	Specify if memory-based context in RETAINED_BANK memory bank has been lost due to a previous power transition or other reset source. 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
7:2	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4PER_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
0	RESERVED		R	0x0

Table 3-1638. RM_L4SEC_DMA_CRYPTO_CONTEXT

Address Offset	0x0000 01DC		
Physical Address	0x4AE0 75DC	**Instance**	L4PER_PRM
Description	This register contains dedicated DMA_CRYPTO context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							LOSTMEM_RETAINED_BANK	RESERVED						LOSTCONTEXT_RFF	RESERVED

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	LOSTMEM_RETAINED_BANK	Specify if memory-based context in RETAINED_BANK memory bank has been lost due to a previous power transition or other reset source. 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
7:2	RESERVED		R	0x0
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4PER_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
0	RESERVED		R	0x0

Table 3-1639. PM_L4PER2_UART8_WKDEP

Address Offset	0x0000 01E0		
Physical Address	0x4AE0 75E0	**Instance**	L4PER_PRM
Description	This register controls wakeup dependency based on UART8 service requests.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						WKUPDEP_UART8_EVE4	WKUPDEP_UART8_EVE3	WKUPDEP_UART8_EVE2	WKUPDEP_UART8_EVE1	WKUPDEP_UART8_DSP2	WKUPDEP_UART8_IPU1	WKUPDEP_UART8_SDMA	WKUPDEP_UART8_DSP1	WKUPDEP_UART8_IPU2	WKUPDEP_UART8_MPU

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	WKUPDEP_UART8_EVE4	Wakeup dependency from UART8 module (SWakeup signal) towards EVE4 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
8	WKUPDEP_UART8_EVE3	Wakeup dependency from UART8 module (SWakeup signal) towards EVE3 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
7	WKUPDEP_UART8_EVE2	Wakeup dependency from UART8 module (SWakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_UART8_EVE1	Wakeup dependency from UART8 module (SWakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_UART8_DSP2	Wakeup dependency from UART8 module (SWakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_UART8_IPU1	Wakeup dependency from UART8 module (SWakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	WKUPDEP_UART8_SDMA	Wakeup dependency from UART8 module (SWakeup signal) towards SDMA + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
2	WKUPDEP_UART8_DSP1	Wakeup dependency from UART8 module (SWakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
1	WKUPDEP_UART8_IPU2	Wakeup dependency from UART8 module (SWakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_UART8_MPU	Wakeup dependency from UART8 module (SWakeup signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1640. RM_L4PER2_UART8_CONTEXT

Address Offset	0x0000 01E4		
Physical Address	0x4AE0 75E4	**Instance**	L4PER_PRM
Description	This register contains dedicated UART8 context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							LOSTMEM_RETAINED_BANK	RESERVED						LOSTCONTEXT_RFF	RESERVED

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	LOSTMEM_RETAINED_BANK	Specify if memory-based context in RETAINED_BANK memory bank has been lost due to a previous power transition or other reset source. 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
7:2	RESERVED		R	0x0
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4PER_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
0	RESERVED		R	0x0

Table 3-1641. PM_L4PER2_UART9_WKDEP

Address Offset	0x0000 01E8		
Physical Address	0x4AE0 75E8	**Instance**	L4PER_PRM
Description	This register controls wakeup dependency based on UART9 service requests.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						WKUPDEP_UART9_EVE4	WKUPDEP_UART9_EVE3	WKUPDEP_UART9_EVE2	WKUPDEP_UART9_EVE1	WKUPDEP_UART9_DSP2	WKUPDEP_UART9_IPU1	WKUPDEP_UART9_SDMA	WKUPDEP_UART9_DSP1	WKUPDEP_UART9_IPU2	WKUPDEP_UART9_MPU

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	WKUPDEP_UART9_EVE4	Wakeup dependency from UART9 module (SWakeup signal) towards EVE4 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
8	WKUPDEP_UART9_EVE3	Wakeup dependency from UART9 module (SWakeup signal) towards EVE3 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
7	WKUPDEP_UART9_EVE2	Wakeup dependency from UART9 module (SWakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_UART9_EVE1	Wakeup dependency from UART9 module (SWakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_UART9_DSP2	Wakeup dependency from UART9 module (SWakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_UART9_IPU1	Wakeup dependency from UART9 module (SWakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	WKUPDEP_UART9_SDMA	Wakeup dependency from UART9 module (SWakeup signal) towards SDMA + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
2	WKUPDEP_UART9_DSP1	Wakeup dependency from UART4 module (SWakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
1	WKUPDEP_UART9_IPU2	Wakeup dependency from UART4 module (SWakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_UART9_MPU	Wakeup dependency from UART9 module (SWakeup signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1642. RM_L4PER2_UART9_CONTEXT

Address Offset	0x0000 01EC		
Physical Address	0x4AE0 75EC	**Instance**	L4PER_PRM
Description	This register contains dedicated UART9 context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							LOSTMEM_RETAINED_BANK	RESERVED						LOSTCONTEXT_RFF	RESERVED

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	LOSTMEM_RETAINED_BANK	Specify if memory-based context in RETAINED_BANK memory bank has been lost due to a previous power transition or other reset source. 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
7:2	RESERVED		R	0x0
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4PER_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
0	RESERVED		R	0x0

Table 3-1643. PM_L4PER2_DCAN2_WKDEP

Address Offset	0x0000 01F0		
Physical Address	0x4AE0 75F0	**Instance**	L4PER_PRM
Description	This register controls wakeup dependency based on MCAN service requests.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						WKUPDEP_DCAN2_EVE4	WKUPDEP_DCAN2_EVE3	WKUPDEP_DCAN2_EVE2	WKUPDEP_DCAN2_EVE1	WKUPDEP_DCAN2_DSP2	WKUPDEP_DCAN2_IPU1	WKUPDEP_DCAN2_SDMA	WKUPDEP_DCAN2_DSP1	WKUPDEP_DCAN2_IPU2	WKUPDEP_DCAN2_MPU

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	WKUPDEP_DCAN2_EVE4	Wakeup dependency from MCAN module (SWakeup signal) towards EVE4 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
8	WKUPDEP_DCAN2_EVE3	Wakeup dependency from MCAN module (SWakeup signal) towards EVE3 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
7	WKUPDEP_DCAN2_EVE2	Wakeup dependency from MCAN module (SWakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_DCAN2_EVE1	Wakeup dependency from MCAN module (SWakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_DCAN2_DSP2	Wakeup dependency from MCAN module (SWakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_DCAN2_IPU1	Wakeup dependency from MCAN module (SWakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	WKUPDEP_DCAN2_SDMA	Wakeup dependency from MCAN module (SWakeup signal) towards SDMA + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
2	WKUPDEP_DCAN2_DSP1	Wakeup dependency from MCAN module (SWakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
1	WKUPDEP_DCAN2_IPU2	Wakeup dependency from MCAN module (SWakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_DCAN2_MPU	Wakeup dependency from MCAN module (SWakeup signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1644. Register Call Summary for Register PM_L4PER2_DCAN2_WKDEP

Clock Management Functional Description
- Clock Domain Dependency: [0][1][2][3]

PRCM Register Manual
- Not Supported Functionality (Registers and Bitfields): [4][5][6][7][8][9]
- L4PER_PRM Register Summary: [10]

Table 3-1645. RM_L4PER2_DCAN2_CONTEXT

Address Offset	0x0000 01F4		
Physical Address	0x4AE0 75F4	**Instance**	L4PER_PRM
Description	This register contains dedicated MCAN context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							LOSTMEM_DCAN_BANK	RESERVED							LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	LOSTMEM_DCAN_BANK	Specify if memory-based context in DCAN memory bank has been lost due to a previous power transition or other reset source. 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
7:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L3INIT_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1646. Register Call Summary for Register RM_L4PER2_DCAN2_CONTEXT

Power Management Functional Description
PRCM Register Manual • L4PER_PRM Register Summary: [1]

Table 3-1647. RM_L4SEC_SHA2MD52_CONTEXT

Address Offset	0x0000 01FC		
Physical Address	0x4AE0 75FC	**Instance**	L4PER_PRM
Description	This register contains dedicated SHA2MD52 context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														LOSTCONTEXT_RFF	RESERVED

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4PER_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
0	RESERVED		R	0x0

3.12.51 MPU_PRM registers

3.12.51.1 MPU_PRM Register Summary

Table 3-1648. MPU_PRM Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	MPU_PRM Physical Address
PM_MPU_PWRSTCTRL	RW	32	0x0000 0000	0x4AE0 6300
PM_MPU_PWRSTST	RW	32	0x0000 0004	0x4AE0 6304
RM_MPU_MPU_CONTEXT	RW	32	0x0000 0024	0x4AE0 6324

3.12.51.2 MPU_PRM Register Description

Table 3-1649. PM_MPU_PWRSTCTRL

Address Offset	0x0000 0000		
Physical Address	0x4AE0 6300	**Instance**	MPU_PRM

Table 3-1649. PM_MPU_PWRSTCTRL (continued)

Description	This register controls the MPU domain power state to reach upon a domain sleep transition. If the value programmed in this register correspond to a lower power state than the one programmed in MPU-SS for CPU0 and/or CPU1, then value of this register is overwritten in PRCM logic to limit the power state to enter. Notes: - Even if value of this register is overwritten in PRCM logic, value of this register remains unchanged. - If user programs MPU power domain to go to CSWRET, then he can not program L2$ to OFF mode.
Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED										MPU_RAM_ONSTATE		MPU_L2_ONSTATE				RESERVED					MPU_RAM_RETSTATE	MPU_L2_RETSTATE		RESERVED			LOWPOWERSTATECHANGE	RESERVED	LOGICRETSTATE	POWERSTATE	

Bits	Field Name	Description	Type	Reset
31:22	RESERVED		R	0x0
21:20	MPU_RAM_ONSTATE	MPU_RAM memory state when domain is ON. 0x3: Memory bank is on when the domain is ON.	R	0x3
19:18	MPU_L2_ONSTATE	MPU_L2 memory state when domain is ON. 0x3: Memory bank is on when the domain is ON.	R	0x3
17:11	RESERVED		R	0x0
10	MPU_RAM_RETSTATE	MPU_RAM memory state when domain is RETENTION. 0x1: Memory bank is retained when domain is in RETENTION state.	R	0x1
9	MPU_L2_RETSTATE	MPU_L2 memory state when domain is RETENTION. Should always be same as or higher than LogicRETState bit-field. 0x0: Memory bank is off when the domain is in the RETENTION state. 0x1: Memory bank is retained when domain is in RETENTION state.	RW	0x1
8:5	RESERVED		R	0x0
4	LOWPOWERSTATECHANGE	Power state change request when domain has already performed a sleep transition. Allows going into deeper low power state without waking up the power domain. 0x0: Do not request a low power state change.	R	0x0
3	RESERVED		R	0x0
2	LOGICRETSTATE	Logic state when power domain is RETENTION 0x0: Only retention registers are retained and remaing logic is off when the domain is in RETENTION state. 0x1: Whole logic is retained when domain is in RETENTION state.	RW	0x1
1:0	POWERSTATE	Power state control 0x0: Reserved 0x1: RETENTION state 0x2: INACTIVE state 0x3: ON State	RW	0x3

Table 3-1650. PM_MPU_PWRSTST

Address Offset	0x0000 0004		
Physical Address	0x4AE0 6304	**Instance**	MPU_PRM
Description	This register provides a status on the MPU domain current power state. [warm reset insensitive]		
Type	RW		

31 30 29 28 27 26	25 24	23 22 21	20	19 18 17 16	15 14 13 12 11 10	9 8	7 6	5 4 3	2	1 0
RESERVED	LASTPOWERSTATEENTERED	RESERVED	INTRANSITION	RESERVED		MPU_RAM_STATEST	MPU_L2_STATEST	RESERVED	LOGICSTATEST	POWERSTATEST

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:24	LASTPOWERSTATEENTERED	Last low power state entered. Set to 0x3 upon write of the same only. This register is intended for debug purpose only. 0x0: Power domain was previously OFF 0x1: Power domain was previously in RETENTION 0x2: Power domain was previously ON-INACTIVE 0x3: Power domain was previously ON-ACTIVE	RW	0x0
23:21	RESERVED		R	0x0
20	INTRANSITION	Domain transition status 0x0: No on-going transition on power domain 0x1: Power domain transition is in progress.	R	0x0
19:10	RESERVED		R	0x0
9:8	MPU_RAM_STATEST	MPU_RAM memory state status 0x0: Memory is OFF 0x1: Memory is RETENTION 0x2: Reserved 0x3: Memory is ON	R	0x3
7:6	MPU_L2_STATEST	MPU_L2 memory state status 0x0: Memory is OFF 0x1: Memory is RETENTION 0x2: Reserved 0x3: Memory is ON	R	0x3
5:3	RESERVED		R	0x0
2	LOGICSTATEST	Logic state status 0x0: Logic in domain is OFF 0x1: Logic in domain is ON	R	0x1
1:0	POWERSTATEST	Current power state status 0x0: Power domain is OFF 0x1: Power domain is in RETENTION 0x2: Power domain is ON-INACTIVE 0x3: Power domain is ON-ACTIVE	R	0x3

Table 3-1651. RM_MPU_MPU_CONTEXT

Address Offset	0x0000 0024		
Physical Address	0x4AE0 6324	**Instance**	MPU_PRM
Description	This register contains dedicated MPU context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																					LOSTMEM_MPU_RAM	LOSTMEM_MPU_L2	RESERVED							LOSTCONTEXT_RFF	LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:11	RESERVED		R	0x0
10	LOSTMEM_MPU_RAM	Specify if memory-based context in MPU_RAM memory bank has been lost due to a previous power transition or other reset source (not affected by a global warm reset). 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
9	LOSTMEM_MPU_L2	Specify if memory-based context in MPU_L2 memory bank has been lost due to a previous power transition or other reset source. 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
8:2	RESERVED		R	0x0
1	LOSTCONTEXT_RFF	Specify if RFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of MPU_MA_PWRON_RET_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of MPU_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

3.12.52 OCP_SOCKET_PRM registers

3.12.52.1 OCP_SOCKET_PRM Register Summary

Table 3-1652. OCP_SOCKET_PRM Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	OCP_SOCKET_PRM Physical Address
REVISION_PRM	R	32	0x0000 0000	0x4AE0 6000
PRM_IRQSTATUS_MPU	RW	32	0x0000 0010	0x4AE0 6010
PRM_IRQSTATUS_MPU_2	RW	32	0x0000 0014	0x4AE0 6014
PRM_IRQENABLE_MPU	RW	32	0x0000 0018	0x4AE0 6018
PRM_IRQENABLE_MPU_2	RW	32	0x0000 001C	0x4AE0 601C
PRM_IRQSTATUS_IPU2	RW	32	0x0000 0020	0x4AE0 6020

Table 3-1652. OCP_SOCKET_PRM Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	OCP_SOCKET_PRM Physical Address
PRM_IRQENABLE_IPU2	RW	32	0x0000 0028	0x4AE0 6028
PRM_IRQSTATUS_DSP1	RW	32	0x0000 0030	0x4AE0 6030
PRM_IRQENABLE_DSP1	RW	32	0x0000 0038	0x4AE0 6038
CM_PRM_PROFILING_CLKCTRL	RW	32	0x0000 0040	0x4AE0 6040
PRM_IRQENABLE_DSP2	RW	32	0x0000 0044	0x4AE0 6044
PRM_IRQENABLE_EVE1	RW	32	0x0000 0048	0x4AE0 6048
PRM_IRQENABLE_EVE2	RW	32	0x0000 004C	0x4AE0 604C
PRM_IRQENABLE_EVE3	RW	32	0x0000 0050	0x4AE0 6050
PRM_IRQENABLE_EVE4	RW	32	0x0000 0054	0x4AE0 6054
PRM_IRQENABLE_IPU1	RW	32	0x0000 0058	0x4AE0 6058
PRM_IRQSTATUS_DSP2	RW	32	0x0000 005C	0x4AE0 605C
PRM_IRQSTATUS_EVE1	RW	32	0x0000 0060	0x4AE0 6060
PRM_IRQSTATUS_EVE2	RW	32	0x0000 0064	0x4AE0 6064
PRM_IRQSTATUS_EVE3	RW	32	0x0000 0068	0x4AE0 6068
PRM_IRQSTATUS_EVE4	RW	32	0x0000 006C	0x4AE0 606C
PRM_IRQSTATUS_IPU1	RW	32	0x0000 0070	0x4AE0 6070
PRM_DEBUG_CFG1	RW	32	0x0000 00E4	0x4AE0 60E4
PRM_DEBUG_CFG2	RW	32	0x0000 00E8	0x4AE0 60E8
PRM_DEBUG_CFG3	RW	32	0x0000 00EC	0x4AE0 60EC
PRM_DEBUG_CFG	RW	32	0x0000 00F0	0x4AE0 60F0
PRM_DEBUG_OUT	R	32	0x0000 00F4	0x4AE0 60F4

3.12.52.2 OCP_SOCKET_PRM Register Description

Table 3-1653. REVISION_PRM

Address Offset	0x0000 0000		
Physical Address	0x4AE0 6000	**Instance**	OCP_SOCKET_PRM
Description	This register contains the IP revision code for the PRM part of the PRCM		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCHEME		RESERVED		FUNC												R_RTL					X_MAJOR			CUSTOM		Y_MINOR					

Bits	Field Name	Description	Type	Reset
31:30	SCHEME		R	0x1
29:28	RESERVED		R	0x0
27:16	FUNC	Function indicates a software compatible module family.	R	0x0
15:11	R_RTL	RTL Version (R)	R	0x0
10:8	X_MAJOR	Major Revision (X)	R	0x3
7:6	CUSTOM	Indicates a special version for a particular device. 0x0: Non custom (standard) revision	R	0x0
5:0	Y_MINOR	Minor Revision (Y)	R	0x1

Table 3-1654. Register Call Summary for Register REVISION_PRM

PRCM Register Manual

- OCP_SOCKET_PRM Register Summary: [0]

Table 3-1655. PRM_IRQSTATUS_MPU

Address Offset	0x0000 0010		
Physical Address	0x4AE0 6010	**Instance**	OCP_SOCKET_PRM
Description	This register provides status on MPU interrupt events. Any event is logged independently of the corresponding IRQENABLE value. SW is required to clear a set bit by writing a '1' into the bit-position to be cleared.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	ABB_IVA_DONE_ST	ABB_DSPEVE_DONE_ST	ABB_GPU_DONE_ST	RESERVED															DPLL_EVE_RECAL_ST	DPLL_DSP_RECAL_ST	RESERVED	IO_ST	TRANSITION_ST	DPLL_DDR_RECAL_ST	DPLL_GPU_RECAL_ST	DPLL_GMAC_RECAL_ST	DPLL_ABE_RECAL_ST	DPLL_PER_RECAL_ST	DPLL_IVA_RECAL_ST	DPLL_MPU_RECAL_ST	DPLL_CORE_RECAL_ST

Bits	Field Name	Description	Type	Reset
31	RESERVED		R	0x0
30	ABB_IVA_DONE_ST	IVA ABB mode change completion status. This status is set for both automatic transition upon a voltage transition, and OPP change (when OPP_CHANGE bit is cleared by hardware in PRM_ABBLDO_MM_CTRL register). It is cleared by SW. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
29	ABB_DSPEVE_DONE_ST	DSPEVE ABB mode change completion status. This status is set for both automatic transition upon a voltage transition, and OPP change (when OPP_CHANGE bit is cleared by hardware in PRM_ABBLDO_MM_CTRL register). It is cleared by SW. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
28	ABB_GPU_DONE_ST	GPU ABB mode change completion status. This status is set for both automatic transition upon a voltage transition, and OPP change (when OPP_CHANGE bit is cleared by hardware in PRM_ABBLDO_MM_CTRL register). It is cleared by SW. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
27:13	RESERVED		R	0x0
12	DPLL_EVE_RECAL_ST	EVE DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
11	DPLL_DSP_RECAL_ST	DSP DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
10	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
9	IO_ST	IO pad event interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
8	TRANSITION_ST	Software supervised transition completed event interrupt status (any domain). Asserted upon completion of any clock domain force wakeup transition or upon completion of any power domain sleep transition with at least one enclosed clock domain configured in forced-sleep. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
7	DPLL_DDR_RECAL_ST	DDR DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
6	DPLL_GPU_RECAL_ST	GPU DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
5	DPLL_GMAC_RECAL_ST	GMAC DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
4	DPLL_ABE_RECAL_ST	ABE DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
3	DPLL_PER_RECAL_ST	PER DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
2	DPLL_IVA_RECAL_ST	IVA DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
1	DPLL_MPU_RECAL_ST	MPU DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
0	DPLL_CORE_RECAL_ST	CORE DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0

Table 3-1656. Register Call Summary for Register PRM_IRQSTATUS_MPU

Clock Management Functional Description • DPLL Recalibration: [0]
PRCM Register Manual • Not Supported Functionality (Registers and Bitfields): [1][2][3][4][5][6][7][8][9][10][11][12][13][14] • OCP_SOCKET_PRM Register Summary: [15]

Table 3-1657. PRM_IRQSTATUS_MPU_2

Address Offset	0x0000 0014		
Physical Address	0x4AE0 6014	**Instance**	OCP_SOCKET_PRM
Description	This register provides status on MPU interrupt events. Any event is logged independently of the corresponding IRQENABLE value. SW is required to clear a set bit by writing a '1' into the bit-position to be cleared.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								ABB_MPU_DONE_ST	RESERVED						

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x0
7	ABB_MPU_DONE_ST	MPU ABB mode change completion status. This status is set for both automatic transition upon a voltage transition, and OPP change (when OPP_CHANGE bit is cleared by hardware in PRM_ABBLDO_MPU_CTRL register). It is cleared by SW. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
6:0	RESERVED		R	0x0

Table 3-1658. PRM_IRQENABLE_MPU

Address Offset	0x0000 0018		
Physical Address	0x4AE0 6018	**Instance**	OCP_SOCKET_PRM
Description	This register is used to enable or disable MPU interrupt activation upon presence of corresponding IRQSTATUS bit.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ABB_IVA_DONE_EN	ABB_DSPEVE_DONE_EN	ABB_GPU_DONE_EN	RESERVED																	DPLL_EVE_RECAL_EN	DPLL_DSP_RECAL_EN	IO_EN	TRANSITION_EN	DPLL_DDR_RECAL_EN	DPLL_GPU_RECAL_EN	DPLL_GMAC_RECAL_EN	DPLL_ABE_RECAL_EN	DPLL_PER_RECAL_EN	DPLL_IVA_RECAL_EN	DPLL_MPU_RECAL_EN	DPLL_CORE_RECAL_EN

Bits	Field Name	Description	Type	Reset
31	ABB_IVA_DONE_EN	IVA ABB mode change done enable 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
30	ABB_DSPEVE_DONE_EN	DSPEVE ABB mode change done enable 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
29	ABB_GPU_DONE_EN	GPU ABB mode change done enable 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
28:12	RESERVED		R	0x0
11	DPLL_EVE_RECAL_EN	EVE DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
10	DPLL_DSP_RECAL_EN	DSP DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
9	IO_EN	IO pad event interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
8	TRANSITION_EN	Software supervised transition completed event interrupt enable (any domain) 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
7	DPLL_DDR_RECAL_EN	DDR DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
6	DPLL_GPU_RECAL_EN	GPU DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
5	DPLL_GMAC_RECAL_EN	GMAC DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
4	DPLL_ABE_RECAL_EN	ABEDPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
3	DPLL_PER_RECAL_EN	PER DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
2	DPLL_IVA_RECAL_EN	IVA DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
1	DPLL_MPU_RECAL_EN	MPU DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
0	DPLL_CORE_RECAL_EN	CORE DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0

Table 3-1659. Register Call Summary for Register PRM_IRQENABLE_MPU

Clock Management Functional Description
- DPLL Recalibration: [0]

PRCM Register Manual
- Not Supported Functionality (Registers and Bitfields): [1][2][3][4][5][6][7][8][9][10][11][12][13][14]
- OCP_SOCKET_PRM Register Summary: [15]

Table 3-1660. PRM_IRQENABLE_MPU_2

Address Offset	0x0000 001C		
Physical Address	0x4AE0 601C	**Instance**	OCP_SOCKET_PRM
Description	This register is used to enable or disable MPU interrupt activation upon presence of corresponding IRQSTATUS bit.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								ABB_MPU_DONE_EN	RESERVED						

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x0
7	ABB_MPU_DONE_EN	MPU ABB mode change done enable 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
6:0	RESERVED		R	0x0

Table 3-1661. PRM_IRQSTATUS_IPU2

Address Offset	0x0000 0020		
Physical Address	0x4AE0 6020	**Instance**	OCP_SOCKET_PRM
Description	This register provides status on IPU2 interrupt events. Any event is logged independently of the corresponding IRQENABLE value. SW is required to clear a set bit by writing a '1' into the bit-position to be cleared.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ABB_MPU_DONE_ST	ABB_IVA_DONE_ST	ABB_DSPEVE_DONE_ST	ABB_GPU_DONE_ST	RESERVED															DPLL_EVE_RECAL_ST	DPLL_DSP_RECAL_ST	FORCEWKUP_ST	IO_ST	TRANSITION_ST	DPLL_DDR_RECAL_ST	DPLL_GPU_RECAL_ST	DPLL_GMAC_RECAL_ST	DPLL_ABE_RECAL_ST	DPLL_PER_RECAL_ST	DPLL_IVA_RECAL_ST	DPLL_MPU_RECAL_ST	DPLL_CORE_RECAL_ST

Bits	Field Name	Description	Type	Reset
31	ABB_MPU_DONE_ST	MPU ABB mode change completion status. This status is set for both automatic transition upon a voltage transition, and OPP change (when OPP_CHANGE bit is cleared by hardware in PRM_ABBLDO_MM_CTRL register). It is cleared by SW. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
30	ABB_IVA_DONE_ST	IVA ABB mode change completion status. This status is set for both automatic transition upon a voltage transition, and OPP change (when OPP_CHANGE bit is cleared by hardware in PRM_ABBLDO_MM_CTRL register). It is cleared by SW. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0

Bits	Field Name	Description	Type	Reset
29	ABB_DSPEVE_DONE_ST	DSPEVE ABB mode change completion status. This status is set for both automatic transition upon a voltage transition, and OPP change (when OPP_CHANGE bit is cleared by hardware in PRM_ABBLDO_MM_CTRL register). It is cleared by SW. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
28	ABB_GPU_DONE_ST	GPU ABB mode change completion status. This status is set for both automatic transition upon a voltage transition, and OPP change (when OPP_CHANGE bit is cleared by hardware in PRM_ABBLDO_MM_CTRL register). It is cleared by SW. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
27:13	RESERVED		R	0x0
12	DPLL_EVE_RECAL_ST	EVE DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
11	DPLL_DSP_RECAL_ST	DSP DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
10	FORCEWKUP_ST	IPU domain software supervised wakeup transition completed event interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
9	IO_ST	IO pad event interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
8	TRANSITION_ST	Software supervised transition completed event interrupt status (any domain). Asserted upon completion of any clock domain force wakeup transition or upon completion of any power domain sleep transition with at least one enclosed clock domain configured in forced-sleep. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
7	DPLL_DDR_RECAL_ST	DDR DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
6	DPLL_GPU_RECAL_ST	GPU DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
5	DPLL_GMAC_RECAL_ST	GMAC DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
4	DPLL_ABE_RECAL_ST	ABE DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
3	DPLL_PER_RECAL_ST	PER DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
2	DPLL_IVA_RECAL_ST	IVA DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0

Bits	Field Name	Description	Type	Reset
1	DPLL_MPU_RECAL_ST	MPU DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
0	DPLL_CORE_RECAL_ST	CORE DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0

Table 3-1662. PRM_IRQENABLE_IPU2

Address Offset	0x0000 0028		
Physical Address	0x4AE0 6028	**Instance**	OCP_SOCKET_PRM
Description	This register is used to enable or disable IPU2 interrupt activation upon presence of corresponding IRQSTATUS bit.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ABB_MPU_DONE_EN	ABB_IVA_DONE_EN	ABB_DSPEVE_DONE_EN	ABB_GPU_DONE_EN	RESERVED															DPLL_EVE_RECAL_EN	DPLL_DSP_RECAL_EN	FORCEWKUP_EN	IO_EN	TRANSITION_EN	DPLL_DDR_RECAL_EN	DPLL_GPU_RECAL_EN	DPLL_GMAC_RECAL_EN	DPLL_ABE_RECAL_EN	DPLL_PER_RECAL_EN	DPLL_IVA_RECAL_EN	DPLL_MPU_RECAL_EN	DPLL_CORE_RECAL_EN

Bits	Field Name	Description	Type	Reset
31	ABB_MPU_DONE_EN	MPU ABB mode change done enable 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
30	ABB_IVA_DONE_EN	IVA ABB mode change done enable 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
29	ABB_DSPEVE_DONE_EN	DSPEVE ABB mode change done enable 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
28	ABB_GPU_DONE_EN	GPU ABB mode change done enable 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
27:13	RESERVED		R	0x0
12	DPLL_EVE_RECAL_EN	EVE DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
11	DPLL_DSP_RECAL_EN	DSP DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
10	FORCEWKUP_EN	IPU domain software supervised wakeup transition completed event interrupt enable. 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
9	IO_EN	IO pad event interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
8	TRANSITION_EN	Software supervised transition completed event interrupt enable (any domain) 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
7	DPLL_DDR_RECAL_EN	DDR DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
6	DPLL_GPU_RECAL_EN	GPU DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
5	DPLL_GMAC_RECAL_EN	GMAC DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
4	DPLL_ABE_RECAL_EN	ABEDPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
3	DPLL_PER_RECAL_EN	PER DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
2	DPLL_IVA_RECAL_EN	IVA DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
1	DPLL_MPU_RECAL_EN	MPU DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
0	DPLL_CORE_RECAL_EN	CORE DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0

Table 3-1663. PRM_IRQSTATUS_DSP1

Address Offset	0x0000 0030		
Physical Address	0x4AE0 6030	**Instance**	OCP_SOCKET_PRM
Description	This register provides status on DSP1 interrupt events. Any event is logged independently of the corresponding IRQENABLE value. SW is required to clear a set bit by writing a '1' into the bit-position to be cleared.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ABB_MPU_DONE_ST	ABB_IVA_DONE_ST	ABB_DSPEVE_DONE_ST	ABB_GPU_DONE_ST	RESERVED															DPLL_EVE_RECAL_ST	DPLL_DSP_RECAL_ST	FORCEWKUP_ST	IO_ST	RESERVED	DPLL_DDR_RECAL_ST	DPLL_GPU_RECAL_ST	DPLL_GMAC_RECAL_ST	DPLL_ABE_RECAL_ST	DPLL_PER_RECAL_ST	DPLL_IVA_RECAL_ST	DPLL_MPU_RECAL_ST	DPLL_CORE_RECAL_ST

Bits	Field Name	Description	Type	Reset
31	ABB_MPU_DONE_ST	MPU ABB mode change completion status. This status is set for both automatic transition upon a voltage transition, and OPP change (when OPP_CHANGE bit is cleared by hardware in PRM_ABBLDO_MM_CTRL register). It is cleared by SW. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
30	ABB_IVA_DONE_ST	IVA ABB mode change completion status. This status is set for both automatic transition upon a voltage transition, and OPP change (when OPP_CHANGE bit is cleared by hardware in PRM_ABBLDO_MM_CTRL register). It is cleared by SW. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
29	ABB_DSPEVE_DONE_ST	DSPEVE ABB mode change completion status. This status is set for both automatic transition upon a voltage transition, and OPP change (when OPP_CHANGE bit is cleared by hardware in PRM_ABBLDO_MM_CTRL register). It is cleared by SW. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
28	ABB_GPU_DONE_ST	GPU ABB mode change completion status. This status is set for both automatic transition upon a voltage transition, and OPP change (when OPP_CHANGE bit is cleared by hardware in PRM_ABBLDO_MM_CTRL register). It is cleared by SW. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
27:13	RESERVED		R	0x0
12	DPLL_EVE_RECAL_ST	EVE DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
11	DPLL_DSP_RECAL_ST	DSP DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
10	FORCEWKUP_ST	IPU domain software supervised wakeup transition completed event interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
9	IO_ST	IO pad event interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
8	RESERVED		R	0x0
7	DPLL_DDR_RECAL_ST	DDR DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
6	DPLL_GPU_RECAL_ST	GPU DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
5	DPLL_GMAC_RECAL_ST	GMAC DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0

Bits	Field Name	Description	Type	Reset
4	DPLL_ABE_RECAL_ST	ABE DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
3	DPLL_PER_RECAL_ST	PER DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
2	DPLL_IVA_RECAL_ST	IVA DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
1	DPLL_MPU_RECAL_ST	MPU DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
0	DPLL_CORE_RECAL_ST	CORE DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0

Table 3-1664. Register Call Summary for Register PRM_IRQSTATUS_DSP1

Clock Management Functional Description
- DPLL Recalibration: [0]
- DPLL_PER Recalibration: [1]
- DPLL_CORE Recalibration: [2]
- DPLL_EVE_VID_DSP Recalibration: [3]
- DPLL_GMAC_DSP Recalibration: [4]
- DPLL_DDR Recalibration: [5]

PRCM Register Manual
- Not Supported Functionality (Registers and Bitfields): [6][7][8][9][10][11][12][13]
- OCP_SOCKET_PRM Register Summary: [14]

Table 3-1665. PRM_IRQENABLE_DSP1

Address Offset	0x0000 0038		
Physical Address	0x4AE0 6038	**Instance**	OCP_SOCKET_PRM
Description	This register is used to enable or disable DSP1 interrupt activation upon presence of corresponding IRQSTATUS bit.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ABB_MPU_DONE_EN	ABB_IVA_DONE_EN	ABB_DSPEVE_DONE_EN	ABB_GPU_DONE_EN	RESERVED														DPLL_USB_RECAL_EN	DPLL_EVE_RECAL_EN	DPLL_DSP_RECAL_EN	FORCEWKUP_EN	IO_EN	RESERVED	DPLL_DDR_RECAL_EN	DPLL_GPU_RECAL_EN	DPLL_GMAC_RECAL_EN	DPLL_ABE_RECAL_EN	DPLL_PER_RECAL_EN	DPLL_IVA_RECAL_EN	DPLL_MPU_RECAL_EN	DPLL_CORE_RECAL_EN

Bits	Field Name	Description	Type	Reset
31	ABB_MPU_DONE_EN	MPU ABB mode change done enable 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0

Bits	Field Name	Description	Type	Reset
30	ABB_IVA_DONE_EN	IVA ABB mode change done enable 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
29	ABB_DSPEVE_DONE_EN	DSPEVE ABB mode change done enable 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
28	ABB_GPU_DONE_EN	GPU ABB mode change done enable 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
27:14	RESERVED		R	0x0
13	DPLL_USB_RECAL_EN	USB DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
12	DPLL_EVE_RECAL_EN	EVE DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
11	DPLL_DSP_RECAL_EN	DSP DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
10	FORCEWKUP_EN	IPU domain software supervised wakeup transition completed event interrupt enable. 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
9	IO_EN	IO pad event interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
8	RESERVED		R	0x0
7	DPLL_DDR_RECAL_EN	DDR DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
6	DPLL_GPU_RECAL_EN	GPU DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
5	DPLL_GMAC_RECAL_EN	GMAC DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
4	DPLL_ABE_RECAL_EN	ABEDPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
3	DPLL_PER_RECAL_EN	PER DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
2	DPLL_IVA_RECAL_EN	IVA DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
1	DPLL_MPU_RECAL_EN	MPU DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
0	DPLL_CORE_RECAL_EN	CORE DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0

Table 3-1666. Register Call Summary for Register PRM_IRQENABLE_DSP1

Clock Management Functional Description • DPLL Recalibration: [0] • DPLL_PER Recalibration: [1] • DPLL_CORE Recalibration: [2] • DPLL_EVE_VID_DSP Recalibration: [3] • DPLL_GMAC_DSP Recalibration: [4] • DPLL_DDR Recalibration: [5]
PRCM Register Manual • Not Supported Functionality (Registers and Bitfields): [6][7][8][9][10][11][12][13][14] • OCP_SOCKET_PRM Register Summary: [15]

Table 3-1667. CM_PRM_PROFILING_CLKCTRL

Address Offset	0x0000 0040		
Physical Address	0x4AE0 6040	**Instance**	OCP_SOCKET_PRM
Description	This register manages the PRM_PROFILING clock. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														IDLEST		RESERVED								MODULEMODE							

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status 0x0: Module is fully functional 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle 0x3: Module is disabled	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. OCP configuration port is not accessible. 0x1: Module is managed automatically by HW along with EMU domain. OCP configuration port is accessible only when EMU domain is on. 0x2: Reserved 0x3: Reserved	RW	0x0

Table 3-1668. Register Call Summary for Register CM_PRM_PROFILING_CLKCTRL

PRCM Register Manual • OCP_SOCKET_PRM Register Summary: [0]

Table 3-1669. PRM_IRQENABLE_DSP2

Address Offset	0x0000 0044		
Physical Address	0x4AE0 6044	**Instance**	OCP_SOCKET_PRM
Description	This register is used to enable or disable DSP2 interrupt activation upon presence of corresponding IRQSTATUS bit.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ABB_MPU_DONE_EN	ABB_IVA_DONE_EN	ABB_DSPEVE_DONE_EN	ABB_GPU_DONE_EN	RESERVED															DPLL_EVE_RECAL_EN	DPLL_DSP_RECAL_EN	FORCEWKUP_EN	IO_EN	RESERVED	DPLL_DDR_RECAL_EN	DPLL_GPU_RECAL_EN	DPLL_GMAC_RECAL_EN	DPLL_ABE_RECAL_EN	DPLL_PER_RECAL_EN	DPLL_IVA_RECAL_EN	DPLL_MPU_RECAL_EN	DPLL_CORE_RECAL_EN

Bits	Field Name	Description	Type	Reset
31	ABB_MPU_DONE_EN	MPU ABB mode change done enable 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
30	ABB_IVA_DONE_EN	IVA ABB mode change done enable 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
29	ABB_DSPEVE_DONE_EN	DSPEVE ABB mode change done enable 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
28	ABB_GPU_DONE_EN	GPU ABB mode change done enable 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
27:13	RESERVED		R	0x0
12	DPLL_EVE_RECAL_EN	EVE DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
11	DPLL_DSP_RECAL_EN	DSP DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
10	FORCEWKUP_EN	IPU domain software supervised wakeup transition completed event interrupt enable. 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
9	IO_EN	IO pad event interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
8	RESERVED		R	0x0
7	DPLL_DDR_RECAL_EN	DDR DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
6	DPLL_GPU_RECAL_EN	GPU DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
5	DPLL_GMAC_RECAL_EN	GMAC DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
4	DPLL_ABE_RECAL_EN	ABEDPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
3	DPLL_PER_RECAL_EN	PER DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
2	DPLL_IVA_RECAL_EN	IVA DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
1	DPLL_MPU_RECAL_EN	MPU DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
0	DPLL_CORE_RECAL_EN	CORE DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0

Table 3-1670. Register Call Summary for Register PRM_IRQENABLE_DSP2

Clock Management Functional Description
- DPLL Recalibration: [0]
- DPLL_PER Recalibration: [1]
- DPLL_CORE Recalibration: [2]
- DPLL_EVE_VID_DSP Recalibration: [3]
- DPLL_GMAC_DSP Recalibration: [4]
- DPLL_DDR Recalibration: [5]

PRCM Register Manual
- Not Supported Functionality (Registers and Bitfields): [6][7][8][9][10][11][12][13]
- OCP_SOCKET_PRM Register Summary: [14]

Table 3-1671. PRM_IRQENABLE_EVE1

Address Offset	0x0000 0048		
Physical Address	0x4AE0 6048	**Instance**	OCP_SOCKET_PRM
Description	This register is used to enable or disable EVE1 interrupt activation upon presence of corresponding IRQSTATUS bit.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ABB_MPU_DONE_EN	ABB_IVA_DONE_EN	ABB_DSPEVE_DONE_EN	ABB_GPU_DONE_EN	RESERVED															DPLL_EVE_RECAL_EN	DPLL_DSP_RECAL_EN	FORCEWKUP_EN	IO_EN	RESERVED	DPLL_DDR_RECAL_EN	DPLL_GPU_RECAL_EN	DPLL_GMAC_RECAL_EN	DPLL_ABE_RECAL_EN	DPLL_PER_RECAL_EN	DPLL_IVA_RECAL_EN	DPLL_MPU_RECAL_EN	DPLL_CORE_RECAL_EN

Bits	Field Name	Description	Type	Reset
31	ABB_MPU_DONE_EN	MPU ABB mode change done enable 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
30	ABB_IVA_DONE_EN	IVA ABB mode change done enable 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
29	ABB_DSPEVE_DONE_EN	DSPEVE ABB mode change done enable 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
28	ABB_GPU_DONE_EN	GPU ABB mode change done enable 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
27:13	RESERVED		R	0x0
12	DPLL_EVE_RECAL_EN	EVE DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
11	DPLL_DSP_RECAL_EN	DSP DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
10	FORCEWKUP_EN	IPU domain software supervised wakeup transition completed event interrupt enable. 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
9	IO_EN	IO pad event interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
8	RESERVED		R	0x0
7	DPLL_DDR_RECAL_EN	DDR DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
6	DPLL_GPU_RECAL_EN	GPU DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
5	DPLL_GMAC_RECAL_EN	GMAC DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
4	DPLL_ABE_RECAL_EN	ABEDPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
3	DPLL_PER_RECAL_EN	PER DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
2	DPLL_IVA_RECAL_EN	IVA DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
1	DPLL_MPU_RECAL_EN	MPU DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
0	DPLL_CORE_RECAL_EN	CORE DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0

Table 3-1672. Register Call Summary for Register PRM_IRQENABLE_EVE1

Clock Management Functional Description
- DPLL_PER Recalibration: [0]
- DPLL_CORE Recalibration: [1]
- DPLL_EVE_VID_DSP Recalibration: [2]
- DPLL_GMAC_DSP Recalibration: [3]
- DPLL_DDR Recalibration: [4]

PRCM Register Manual
- OCP_SOCKET_PRM Register Summary: [5]

Table 3-1673. PRM_IRQENABLE_EVE2

Address Offset	0x0000 004C		
Physical Address	0x4AE0 604C	**Instance**	OCP_SOCKET_PRM
Description	This register is used to enable or disable EVE2 interrupt activation upon presence of corresponding IRQSTATUS bit.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ABB_MPU_DONE_EN	ABB_IVA_DONE_EN	ABB_DSPEVE_DONE_EN	ABB_GPU_DONE_EN	RESERVED															DPLL_EVE_RECAL_EN	DPLL_DSP_RECAL_EN	FORCEWKUP_EN	IO_EN	RESERVED	DPLL_DDR_RECAL_EN	DPLL_GPU_RECAL_EN	DPLL_GMAC_RECAL_EN	DPLL_ABE_RECAL_EN	DPLL_PER_RECAL_EN	DPLL_IVA_RECAL_EN	DPLL_MPU_RECAL_EN	DPLL_CORE_RECAL_EN

Bits	Field Name	Description	Type	Reset
31	ABB_MPU_DONE_EN	MPU ABB mode change done enable 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
30	ABB_IVA_DONE_EN	IVA ABB mode change done enable 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0

Bits	Field Name	Description	Type	Reset
29	ABB_DSPEVE_DONE_EN	DSPEVE ABB mode change done enable 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
28	ABB_GPU_DONE_EN	GPU ABB mode change done enable 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
27:13	RESERVED		R	0x0
12	DPLL_EVE_RECAL_EN	EVE DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
11	DPLL_DSP_RECAL_EN	DSP DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
10	FORCEWKUP_EN	IPU domain software supervised wakeup transition completed event interrupt enable. 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
9	IO_EN	IO pad event interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
8	RESERVED		R	0x0
7	DPLL_DDR_RECAL_EN	DDR DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
6	DPLL_GPU_RECAL_EN	GPU DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
5	DPLL_GMAC_RECAL_EN	GMAC DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
4	DPLL_ABE_RECAL_EN	ABEDPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
3	DPLL_PER_RECAL_EN	PER DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
2	DPLL_IVA_RECAL_EN	IVA DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
1	DPLL_MPU_RECAL_EN	MPU DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
0	DPLL_CORE_RECAL_EN	CORE DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0

Table 3-1674. PRM_IRQENABLE_EVE3

Address Offset	0x0000 0050		
Physical Address	0x4AE0 6050	**Instance**	OCP_SOCKET_PRM
Description	This register is used to enable or disable EVE3 interrupt activation upon presence of corresponding IRQSTATUS bit.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ABB_MPU_DONE_EN	ABB_IVA_DONE_EN	ABB_DSPEVE_DONE_EN	ABB_GPU_DONE_EN	RESERVED															DPLL_EVE_RECAL_EN	DPLL_DSP_RECAL_EN	FORCEWKUP_EN	IO_EN	RESERVED	DPLL_DDR_RECAL_EN	DPLL_GPU_RECAL_EN	DPLL_GMAC_RECAL_EN	DPLL_ABE_RECAL_EN	DPLL_PER_RECAL_EN	DPLL_IVA_RECAL_EN	DPLL_MPU_RECAL_EN	DPLL_CORE_RECAL_EN

Bits	Field Name	Description	Type	Reset
31	ABB_MPU_DONE_EN	MPU ABB mode change done enable 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
30	ABB_IVA_DONE_EN	IVA ABB mode change done enable 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
29	ABB_DSPEVE_DONE_EN	DSPEVE ABB mode change done enable 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
28	ABB_GPU_DONE_EN	GPU ABB mode change done enable 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
27:13	RESERVED		R	0x0
12	DPLL_EVE_RECAL_EN	EVE DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
11	DPLL_DSP_RECAL_EN	DSP DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
10	FORCEWKUP_EN	IPU domain software supervised wakeup transition completed event interrupt enable. 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
9	IO_EN	IO pad event interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
8	RESERVED		R	0x0
7	DPLL_DDR_RECAL_EN	DDR DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
6	DPLL_GPU_RECAL_EN	GPU DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
5	DPLL_GMAC_RECAL_EN	GMAC DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
4	DPLL_ABE_RECAL_EN	ABEDPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
3	DPLL_PER_RECAL_EN	PER DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
2	DPLL_IVA_RECAL_EN	IVA DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
1	DPLL_MPU_RECAL_EN	MPU DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
0	DPLL_CORE_RECAL_EN	CORE DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0

Table 3-1675. PRM_IRQENABLE_EVE4

Address Offset	0x0000 0054		
Physical Address	0x4AE0 6054	**Instance**	OCP_SOCKET_PRM
Description	This register is used to enable or disable EVE4 interrupt activation upon presence of corresponding IRQSTATUS bit.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ABB_MPU_DONE_EN	ABB_IVA_DONE_EN	ABB_DSPEVE_DONE_EN	ABB_GPU_DONE_EN	RESERVED															DPLL_EVE_RECAL_EN	DPLL_DSP_RECAL_EN	FORCEWKUP_EN	IO_EN	RESERVED	DPLL_DDR_RECAL_EN	DPLL_GPU_RECAL_EN	DPLL_GMAC_RECAL_EN	DPLL_ABE_RECAL_EN	DPLL_PER_RECAL_EN	DPLL_IVA_RECAL_EN	DPLL_MPU_RECAL_EN	DPLL_CORE_RECAL_EN

Bits	Field Name	Description	Type	Reset
31	ABB_MPU_DONE_EN	MPU ABB mode change done enable 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
30	ABB_IVA_DONE_EN	IVA ABB mode change done enable 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
29	ABB_DSPEVE_DONE_EN	DSPEVE ABB mode change done enable 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0

Bits	Field Name	Description	Type	Reset
28	ABB_GPU_DONE_EN	GPU ABB mode change done enable 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
27:13	RESERVED		R	0x0
12	DPLL_EVE_RECAL_EN	EVE DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
11	DPLL_DSP_RECAL_EN	DSP DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
10	FORCEWKUP_EN	IPU domain software supervised wakeup transition completed event interrupt enable. 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
9	IO_EN	IO pad event interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
8	RESERVED		R	0x0
7	DPLL_DDR_RECAL_EN	DDR DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
6	DPLL_GPU_RECAL_EN	GPU DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
5	DPLL_GMAC_RECAL_EN	GMAC DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
4	DPLL_ABE_RECAL_EN	ABEDPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
3	DPLL_PER_RECAL_EN	PER DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
2	DPLL_IVA_RECAL_EN	IVA DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
1	DPLL_MPU_RECAL_EN	MPU DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
0	DPLL_CORE_RECAL_EN	CORE DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0

Table 3-1676. PRM_IRQENABLE_IPU1

Address Offset	0x0000 0058		
Physical Address	0x4AE0 6058	**Instance**	OCP_SOCKET_PRM
Description	This register is used to enable or disable IPU1 interrupt activation upon presence of corresponding IRQSTATUS bit.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ABB_MPU_DONE_EN	ABB_IVA_DONE_EN	ABB_DSPEVE_DONE_EN	ABB_GPU_DONE_EN	RESERVED															DPLL_EVE_RECAL_EN	DPLL_DSP_RECAL_EN	FORCEWKUP_EN	IO_EN	TRANSITION_EN	DPLL_DDR_RECAL_EN	DPLL_GPU_RECAL_EN	DPLL_GMAC_RECAL_EN	DPLL_ABE_RECAL_EN	DPLL_PER_RECAL_EN	DPLL_IVA_RECAL_EN	DPLL_MPU_RECAL_EN	DPLL_CORE_RECAL_EN

Bits	Field Name	Description	Type	Reset
31	ABB_MPU_DONE_EN	MPU ABB mode change done enable 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
30	ABB_IVA_DONE_EN	IVA ABB mode change done enable 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
29	ABB_DSPEVE_DONE_EN	DSPEVE ABB mode change done enable 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
28	ABB_GPU_DONE_EN	GPU ABB mode change done enable 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
27:13	RESERVED		R	0x0
12	DPLL_EVE_RECAL_EN	EVE DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
11	DPLL_DSP_RECAL_EN	DSP DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
10	FORCEWKUP_EN	IPU domain software supervised wakeup transition completed event interrupt enable. 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
9	IO_EN	IO pad event interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
8	TRANSITION_EN	Software supervised transition completed event interrupt enable (any domain) 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
7	DPLL_DDR_RECAL_EN	DDR DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
6	DPLL_GPU_RECAL_EN	GPU DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
5	DPLL_GMAC_RECAL_EN	GMAC DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
4	DPLL_ABE_RECAL_EN	ABEDPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
3	DPLL_PER_RECAL_EN	PER DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
2	DPLL_IVA_RECAL_EN	IVA DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
1	DPLL_MPU_RECAL_EN	MPU DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
0	DPLL_CORE_RECAL_EN	CORE DPLL recalibration interrupt enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0

Table 3-1677. Register Call Summary for Register PRM_IRQENABLE_IPU1

Clock Management Functional Description
- DPLL Recalibration: [0]
- DPLL_PER Recalibration: [1]
- DPLL_CORE Recalibration: [2]
- DPLL_EVE_VID_DSP Recalibration: [3]
- DPLL_GMAC_DSP Recalibration: [4]
- DPLL_DDR Recalibration: [5]

PRCM Register Manual
- OCP_SOCKET_PRM Register Summary: [6]

Table 3-1678. PRM_IRQSTATUS_DSP2

Address Offset	0x0000 005C		
Physical Address	0x4AE0 605C	**Instance**	OCP_SOCKET_PRM
Description	This register provides status on DSP interrupt events. Any event is logged independently of the corresponding IRQENABLE value. SW is required to clear a set bit by writing a '1' into the bit-position to be cleared.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ABB_MPU_DONE_ST	ABB_IVA_DONE_ST	ABB_DSPEVE_DONE_ST	ABB_GPU_DONE_ST	RESERVED															DPLL_EVE_RECAL_ST	DPLL_DSP_RECAL_ST	FORCEWKUP_ST	IO_ST	RESERVED	DPLL_DDR_RECAL_ST	DPLL_GPU_RECAL_ST	DPLL_GMAC_RECAL_ST	DPLL_ABE_RECAL_ST	DPLL_PER_RECAL_ST	DPLL_IVA_RECAL_ST	DPLL_MPU_RECAL_ST	DPLL_CORE_RECAL_ST

Bits	Field Name	Description	Type	Reset
31	ABB_MPU_DONE_ST	MPU ABB mode change completion status. This status is set for both automatic transition upon a voltage transition, and OPP change (when OPP_CHANGE bit is cleared by hardware in PRM_ABBLDO_MM_CTRL register). It is cleared by SW. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
30	ABB_IVA_DONE_ST	IVA ABB mode change completion status. This status is set for both automatic transition upon a voltage transition, and OPP change (when OPP_CHANGE bit is cleared by hardware in PRM_ABBLDO_MM_CTRL register). It is cleared by SW. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
29	ABB_DSPEVE_DONE_ST	DSPEVE ABB mode change completion status. This status is set for both automatic transition upon a voltage transition, and OPP change (when OPP_CHANGE bit is cleared by hardware in PRM_ABBLDO_MM_CTRL register). It is cleared by SW. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
28	ABB_GPU_DONE_ST	GPU ABB mode change completion status. This status is set for both automatic transition upon a voltage transition, and OPP change (when OPP_CHANGE bit is cleared by hardware in PRM_ABBLDO_MM_CTRL register). It is cleared by SW. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
27:13	RESERVED		R	0x0
12	DPLL_EVE_RECAL_ST	EVE DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
11	DPLL_DSP_RECAL_ST	DSP DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
10	FORCEWKUP_ST	IPU domain software supervised wakeup transition completed event interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
9	IO_ST	IO pad event interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
8	RESERVED		R	0x0
7	DPLL_DDR_RECAL_ST	DDR DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
6	DPLL_GPU_RECAL_ST	GPU DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
5	DPLL_GMAC_RECAL_ST	GMAC DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0

Bits	Field Name	Description	Type	Reset
4	DPLL_ABE_RECAL_ST	ABE DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
3	DPLL_PER_RECAL_ST	PER DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
2	DPLL_IVA_RECAL_ST	IVA DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
1	DPLL_MPU_RECAL_ST	MPU DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
0	DPLL_CORE_RECAL_ST	CORE DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0

Table 3-1679. Register Call Summary for Register PRM_IRQSTATUS_DSP2

Clock Management Functional Description
- DPLL Recalibration: [0]
- DPLL_PER Recalibration: [1]
- DPLL_CORE Recalibration: [2]
- DPLL_EVE_VID_DSP Recalibration: [3]
- DPLL_GMAC_DSP Recalibration: [4]
- DPLL_DDR Recalibration: [5]

PRCM Register Manual
- Not Supported Functionality (Registers and Bitfields): [6][7][8][9][10][11][12][13]
- OCP_SOCKET_PRM Register Summary: [14]

Table 3-1680. PRM_IRQSTATUS_EVE1

Address Offset	0x0000 0060		
Physical Address	0x4AE0 6060	**Instance**	OCP_SOCKET_PRM
Description	This register provides status on EVE interrupt events. Any event is logged independently of the corresponding IRQENABLE value. SW is required to clear a set bit by writing a '1' into the bit-position to be cleared.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ABB_MPU_DONE_ST	ABB_IVA_DONE_ST	ABB_DSPEVE_DONE_ST	ABB_GPU_DONE_ST	RESERVED															DPLL_EVE_RECAL_ST	DPLL_DSP_RECAL_ST	FORCEWKUP_ST	IO_ST	RESERVED	DPLL_DDR_RECAL_ST	DPLL_GPU_RECAL_ST	DPLL_GMAC_RECAL_ST	DPLL_ABE_RECAL_ST	DPLL_PER_RECAL_ST	DPLL_IVA_RECAL_ST	DPLL_MPU_RECAL_ST	DPLL_CORE_RECAL_ST

Bits	Field Name	Description	Type	Reset
31	ABB_MPU_DONE_ST	MPU ABB mode change completion status. This status is set for both automatic transition upon a voltage transition, and OPP change (when OPP_CHANGE bit is cleared by hardware in PRM_ABBLDO_MM_CTRL register). It is cleared by SW. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
30	ABB_IVA_DONE_ST	IVA ABB mode change completion status. This status is set for both automatic transition upon a voltage transition, and OPP change (when OPP_CHANGE bit is cleared by hardware in PRM_ABBLDO_MM_CTRL register). It is cleared by SW. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
29	ABB_DSPEVE_DONE_ST	DSPEVE ABB mode change completion status. This status is set for both automatic transition upon a voltage transition, and OPP change (when OPP_CHANGE bit is cleared by hardware in PRM_ABBLDO_MM_CTRL register). It is cleared by SW. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
28	ABB_GPU_DONE_ST	GPU ABB mode change completion status. This status is set for both automatic transition upon a voltage transition, and OPP change (when OPP_CHANGE bit is cleared by hardware in PRM_ABBLDO_MM_CTRL register). It is cleared by SW. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
27:13	RESERVED		R	0x0
12	DPLL_EVE_RECAL_ST	EVE DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
11	DPLL_DSP_RECAL_ST	DSP DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
10	FORCEWKUP_ST	IPU domain software supervised wakeup transition completed event interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
9	IO_ST	IO pad event interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
8	RESERVED		R	0x0
7	DPLL_DDR_RECAL_ST	DDR DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
6	DPLL_GPU_RECAL_ST	GPU DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
5	DPLL_GMAC_RECAL_ST	GMAC DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0

Bits	Field Name	Description	Type	Reset
4	DPLL_ABE_RECAL_ST	ABE DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
3	DPLL_PER_RECAL_ST	PER DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
2	DPLL_IVA_RECAL_ST	IVA DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
1	DPLL_MPU_RECAL_ST	MPU DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
0	DPLL_CORE_RECAL_ST	CORE DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0

Table 3-1681. Register Call Summary for Register PRM_IRQSTATUS_EVE1

Clock Management Functional Description
- DPLL_PER Recalibration: [0]
- DPLL_CORE Recalibration: [1]
- DPLL_EVE_VID_DSP Recalibration: [2]
- DPLL_GMAC_DSP Recalibration: [3]
- DPLL_DDR Recalibration: [4]

PRCM Register Manual
- OCP_SOCKET_PRM Register Summary: [5]

Table 3-1682. PRM_IRQSTATUS_EVE2

Address Offset	0x0000 0064		
Physical Address	0x4AE0 6064	**Instance**	OCP_SOCKET_PRM
Description	This register provides status on EVE interrupt events. Any event is logged independently of the corresponding IRQENABLE value. SW is required to clear a set bit by writing a '1' into the bit-position to be cleared.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ABB_MPU_DONE_ST	ABB_IVA_DONE_ST	ABB_DSPEVE_DONE_ST	ABB_GPU_DONE_ST	RESERVED															DPLL_EVE_RECAL_ST	DPLL_DSP_RECAL_ST	FORCEWKUP_ST	IO_ST	RESERVED	DPLL_DDR_RECAL_ST	DPLL_GPU_RECAL_ST	DPLL_GMAC_RECAL_ST	DPLL_ABE_RECAL_ST	DPLL_PER_RECAL_ST	DPLL_IVA_RECAL_ST	DPLL_MPU_RECAL_ST	DPLL_CORE_RECAL_ST

Bits	Field Name	Description	Type	Reset
31	ABB_MPU_DONE_ST	MPU ABB mode change completion status. This status is set for both automatic transition upon a voltage transition, and OPP change (when OPP_CHANGE bit is cleared by hardware in PRM_ABBLDO_MM_CTRL register). It is cleared by SW. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
30	ABB_IVA_DONE_ST	IVA ABB mode change completion status. This status is set for both automatic transition upon a voltage transition, and OPP change (when OPP_CHANGE bit is cleared by hardware in PRM_ABBLDO_MM_CTRL register). It is cleared by SW. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
29	ABB_DSPEVE_DONE_ST	DSPEVE ABB mode change completion status. This status is set for both automatic transition upon a voltage transition, and OPP change (when OPP_CHANGE bit is cleared by hardware in PRM_ABBLDO_MM_CTRL register). It is cleared by SW. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
28	ABB_GPU_DONE_ST	GPU ABB mode change completion status. This status is set for both automatic transition upon a voltage transition, and OPP change (when OPP_CHANGE bit is cleared by hardware in PRM_ABBLDO_MM_CTRL register). It is cleared by SW. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
27:13	RESERVED		R	0x0
12	DPLL_EVE_RECAL_ST	EVE DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
11	DPLL_DSP_RECAL_ST	DSP DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
10	FORCEWKUP_ST	IPU domain software supervised wakeup transition completed event interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
9	IO_ST	IO pad event interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
8	RESERVED		R	0x0
7	DPLL_DDR_RECAL_ST	DDR DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
6	DPLL_GPU_RECAL_ST	GPU DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
5	DPLL_GMAC_RECAL_ST	GMAC DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0

Bits	Field Name	Description	Type	Reset
4	DPLL_ABE_RECAL_ST	ABE DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
3	DPLL_PER_RECAL_ST	PER DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
2	DPLL_IVA_RECAL_ST	IVA DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
1	DPLL_MPU_RECAL_ST	MPU DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
0	DPLL_CORE_RECAL_ST	CORE DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0

Table 3-1683. PRM_IRQSTATUS_EVE3

Address Offset	0x0000 0068		
Physical Address	0x4AE0 6068	**Instance**	OCP_SOCKET_PRM
Description	This register provides status on EVE interrupt events. Any event is logged independently of the corresponding IRQENABLE value. SW is required to clear a set bit by writing a '1' into the bit-position to be cleared.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ABB_MPU_DONE_ST	ABB_IVA_DONE_ST	ABB_DSPEVE_DONE_ST	ABB_GPU_DONE_ST	RESERVED															DPLL_EVE_RECAL_ST	DPLL_DSP_RECAL_ST	FORCEWKUP_ST	IO_ST	RESERVED	DPLL_DDR_RECAL_ST	DPLL_GPU_RECAL_ST	DPLL_GMAC_RECAL_ST	DPLL_ABE_RECAL_ST	DPLL_PER_RECAL_ST	DPLL_IVA_RECAL_ST	DPLL_MPU_RECAL_ST	DPLL_CORE_RECAL_ST

Bits	Field Name	Description	Type	Reset
31	ABB_MPU_DONE_ST	MPU ABB mode change completion status. This status is set for both automatic transition upon a voltage transition, and OPP change (when OPP_CHANGE bit is cleared by hardware in PRM_ABBLDO_MM_CTRL register). It is cleared by SW. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
30	ABB_IVA_DONE_ST	IVA ABB mode change completion status. This status is set for both automatic transition upon a voltage transition, and OPP change (when OPP_CHANGE bit is cleared by hardware in PRM_ABBLDO_MM_CTRL register). It is cleared by SW. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0

Bits	Field Name	Description	Type	Reset
29	ABB_DSPEVE_DONE_ST	DSPEVE ABB mode change completion status. This status is set for both automatic transition upon a voltage transition, and OPP change (when OPP_CHANGE bit is cleared by hardware in PRM_ABBLDO_MM_CTRL register). It is cleared by SW. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
28	ABB_GPU_DONE_ST	GPU ABB mode change completion status. This status is set for both automatic transition upon a voltage transition, and OPP change (when OPP_CHANGE bit is cleared by hardware in PRM_ABBLDO_MM_CTRL register). It is cleared by SW. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
27:13	RESERVED		R	0x0
12	DPLL_EVE_RECAL_ST	EVE DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
11	DPLL_DSP_RECAL_ST	DSP DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
10	FORCEWKUP_ST	IPU domain software supervised wakeup transition completed event interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
9	IO_ST	IO pad event interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
8	RESERVED		R	0x0
7	DPLL_DDR_RECAL_ST	DDR DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
6	DPLL_GPU_RECAL_ST	GPU DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
5	DPLL_GMAC_RECAL_ST	GMAC DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
4	DPLL_ABE_RECAL_ST	ABE DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
3	DPLL_PER_RECAL_ST	PER DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
2	DPLL_IVA_RECAL_ST	IVA DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
1	DPLL_MPU_RECAL_ST	MPU DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0

Bits	Field Name	Description	Type	Reset
0	DPLL_CORE_RECAL_ST	CORE DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0

Table 3-1684. PRM_IRQSTATUS_EVE4

Address Offset	0x0000 006C		
Physical Address	0x4AE0 606C	**Instance**	OCP_SOCKET_PRM
Description	This register provides status on EVE interrupt events. Any event is logged independently of the corresponding IRQENABLE value. SW is required to clear a set bit by writing a '1' into the bit-position to be cleared.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ABB_MPU_DONE_ST	ABB_IVA_DONE_ST	ABB_DSPEVE_DONE_ST	ABB_GPU_DONE_ST	RESERVED															DPLL_EVE_RECAL_ST	DPLL_DSP_RECAL_ST	FORCEWKUP_ST	IO_ST	RESERVED	DPLL_DDR_RECAL_ST	DPLL_GPU_RECAL_ST	DPLL_GMAC_RECAL_ST	DPLL_ABE_RECAL_ST	DPLL_PER_RECAL_ST	DPLL_IVA_RECAL_ST	DPLL_MPU_RECAL_ST	DPLL_CORE_RECAL_ST

Bits	Field Name	Description	Type	Reset
31	ABB_MPU_DONE_ST	MPU ABB mode change completion status. This status is set for both automatic transition upon a voltage transition, and OPP change (when OPP_CHANGE bit is cleared by hardware in PRM_ABBLDO_MM_CTRL register). It is cleared by SW. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
30	ABB_IVA_DONE_ST	IVA ABB mode change completion status. This status is set for both automatic transition upon a voltage transition, and OPP change (when OPP_CHANGE bit is cleared by hardware in PRM_ABBLDO_MM_CTRL register). It is cleared by SW. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
29	ABB_DSPEVE_DONE_ST	DSPEVE ABB mode change completion status. This status is set for both automatic transition upon a voltage transition, and OPP change (when OPP_CHANGE bit is cleared by hardware in PRM_ABBLDO_MM_CTRL register). It is cleared by SW. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
28	ABB_GPU_DONE_ST	GPU ABB mode change completion status. This status is set for both automatic transition upon a voltage transition, and OPP change (when OPP_CHANGE bit is cleared by hardware in PRM_ABBLDO_MM_CTRL register). It is cleared by SW. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
27:13	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
12	DPLL_EVE_RECAL_ST	EVE DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
11	DPLL_DSP_RECAL_ST	DSP DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
10	FORCEWKUP_ST	IPU domain software supervised wakeup transition completed event interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
9	IO_ST	IO pad event interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
8	RESERVED		R	0x0
7	DPLL_DDR_RECAL_ST	DDR DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
6	DPLL_GPU_RECAL_ST	GPU DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
5	DPLL_GMAC_RECAL_ST	GMAC DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
4	DPLL_ABE_RECAL_ST	ABE DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
3	DPLL_PER_RECAL_ST	PER DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
2	DPLL_IVA_RECAL_ST	IVA DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
1	DPLL_MPU_RECAL_ST	MPU DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
0	DPLL_CORE_RECAL_ST	CORE DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0

Table 3-1685. PRM_IRQSTATUS_IPU1

Address Offset	0x0000 0070		
Physical Address	0x4AE0 6070	**Instance**	OCP_SOCKET_PRM
Description	This register provides status on IPU1 interrupt events. Any event is logged independently of the corresponding IRQENABLE value. SW is required to clear a set bit by writing a '1' into the bit-position to be cleared.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ABB_MPU_DONE_ST	ABB_IVA_DONE_ST	ABB_DSPEVE_DONE_ST	ABB_GPU_DONE_ST	RESERVED															DPLL_EVE_RECAL_ST	DPLL_DSP_RECAL_ST	FORCEWKUP_ST	IO_ST	TRANSITION_ST	DPLL_DDR_RECAL_ST	DPLL_GPU_RECAL_ST	DPLL_GMAC_RECAL_ST	DPLL_ABE_RECAL_ST	DPLL_PER_RECAL_ST	DPLL_IVA_RECAL_ST	DPLL_MPU_RECAL_ST	DPLL_CORE_RECAL_ST

Bits	Field Name	Description	Type	Reset
31	ABB_MPU_DONE_ST	MPU ABB mode change completion status. This status is set for both automatic transition upon a voltage transition, and OPP change (when OPP_CHANGE bit is cleared by hardware in PRM_ABBLDO_MM_CTRL register). It is cleared by SW. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
30	ABB_IVA_DONE_ST	IVA ABB mode change completion status. This status is set for both automatic transition upon a voltage transition, and OPP change (when OPP_CHANGE bit is cleared by hardware in PRM_ABBLDO_MM_CTRL register). It is cleared by SW. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
29	ABB_DSPEVE_DONE_ST	DSPEVE ABB mode change completion status. This status is set for both automatic transition upon a voltage transition, and OPP change (when OPP_CHANGE bit is cleared by hardware in PRM_ABBLDO_MM_CTRL register). It is cleared by SW. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
28	ABB_GPU_DONE_ST	GPU ABB mode change completion status. This status is set for both automatic transition upon a voltage transition, and OPP change (when OPP_CHANGE bit is cleared by hardware in PRM_ABBLDO_MM_CTRL register). It is cleared by SW. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
27:13	RESERVED		R	0x0
12	DPLL_EVE_RECAL_ST	EVE DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
11	DPLL_DSP_RECAL_ST	DSP DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
10	FORCEWKUP_ST	IPU domain software supervised wakeup transition completed event interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
9	IO_ST	IO pad event interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0

Bits	Field Name	Description	Type	Reset
8	TRANSITION_ST	Software supervised transition completed event interrupt status (any domain). Asserted upon completion of any clock domain force wakeup transition or upon completion of any power domain sleep transition with at least one enclosed clock domain configured in forced-sleep. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
7	DPLL_DDR_RECAL_ST	DDR DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
6	DPLL_GPU_RECAL_ST	GPU DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
5	DPLL_GMAC_RECAL_ST	GMAC DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
4	DPLL_ABE_RECAL_ST	ABE DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
3	DPLL_PER_RECAL_ST	PER DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
2	DPLL_IVA_RECAL_ST	IVA DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
1	DPLL_MPU_RECAL_ST	MPU DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0
0	DPLL_CORE_RECAL_ST	CORE DPLL recalibration interrupt status. 0x0: No interrupt 0x1: Interrupt is pending	RW	0x0

Table 3-1686. Register Call Summary for Register PRM_IRQSTATUS_IPU1

Clock Management Functional Description
- DPLL Recalibration: [0]
- DPLL_PER Recalibration: [1]
- DPLL_CORE Recalibration: [2]
- DPLL_EVE_VID_DSP Recalibration: [3]
- DPLL_GMAC_DSP Recalibration: [4]
- DPLL_DDR Recalibration: [5]

PRCM Register Manual
- Not Supported Functionality (Registers and Bitfields): [6][7][8][9][10][11][12][13]
- OCP_SOCKET_PRM Register Summary: [14]

Table 3-1687. PRM_DEBUG_CFG1

Address Offset	0x0000 00E4		
Physical Address	0x4AE0 60E4	**Instance**	OCP_SOCKET_PRM
Description	This register is used to configure the PRM's 32-bit debug output. There is one 7-bit source select field for selecting from a shared set of 8-bit internal signal blocks per byte. The signals included in each block are specified in the PRCM integration specification. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								SEL1							

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8:0	SEL1	Internal signal block select for debug word byte-1	RW	0x0

Table 3-1688. Register Call Summary for Register PRM_DEBUG_CFG1

PRCM Register Manual
- OCP_SOCKET_PRM Register Summary: [0]

Table 3-1689. PRM_DEBUG_CFG2

Address Offset	0x0000 00E8		
Physical Address	0x4AE0 60E8	**Instance**	OCP_SOCKET_PRM
Description	This register is used to configure the PRM's 32-bit debug output. There is one 7-bit source select field for selecting from a shared set of 8-bit internal signal blocks per byte. The signals included in each block are specified in the PRCM integration specification. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								SEL2							

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8:0	SEL2	Internal signal block select for debug word byte-2	RW	0x0

Table 3-1690. Register Call Summary for Register PRM_DEBUG_CFG2

PRCM Register Manual
- OCP_SOCKET_PRM Register Summary: [0]

Table 3-1691. PRM_DEBUG_CFG3

Address Offset	0x0000 00EC		
Physical Address	0x4AE0 60EC	**Instance**	OCP_SOCKET_PRM
Description	This register is used to configure the PRM's 32-bit debug output. There is one 7-bit source select field for selecting from a shared set of 8-bit internal signal blocks per byte. The signals included in each block are specified in the PRCM integration specification. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								SEL3							

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8:0	SEL3	Internal signal block select for debug word byte-3	RW	0x0

Table 3-1692. Register Call Summary for Register PRM_DEBUG_CFG3

PRCM Register Manual
- OCP_SOCKET_PRM Register Summary: [0]

Table 3-1693. PRM_DEBUG_CFG

Address Offset	0x0000 00F0		
Physical Address	0x4AE0 60F0	**Instance**	OCP_SOCKET_PRM
Description	This register is used to configure the PRM's 32-bit debug output. There is one 7-bit source select field for selecting from a shared set of 8-bit internal signal blocks per byte. The signals included in each block are specified in the PRCM integration specification. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								SEL0							

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8:0	SEL0	Internal signal block select for debug word byte-0	RW	0x0

Table 3-1694. Register Call Summary for Register PRM_DEBUG_CFG

PRCM Register Manual
- OCP_SOCKET_PRM Register Summary: [0]

Table 3-1695. PRM_DEBUG_OUT

Address Offset	0x0000 00F4		
Physical Address	0x4AE0 60F4	**Instance**	OCP_SOCKET_PRM
Description	This register is used to monitor the PRM's 32 bit HEDEBUG BUS [warm reset insensitive]		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OUTPUT																															

Bits	Field Name	Description	Type	Reset
31:0	OUTPUT	HW DEBUG OUTPUT	R	0x0

Table 3-1696. Register Call Summary for Register PRM_DEBUG_OUT

PRCM Register Manual
- OCP_SOCKET_PRM Register Summary: [0]

3.12.53 RTC_PRM registers

3.12.53.1 RTC_PRM Register Summary

Table 3-1697. RTC_PRM Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	RTC_PRM Physical Address
PM_RTC_RTCSS_WKDEP	RW	32	0x0000 0000	0x4AE0 7C60
RM_RTC_RTCSS_CONTEXT	RW	32	0x0000 0004	0x4AE0 7C64

3.12.53.2 RTC_PRM Register Description

Table 3-1698. PM_RTC_RTCSS_WKDEP

Address Offset	0x0000 0000		
Physical Address	0x4AE0 7C60	**Instance**	RTC_PRM
Description	This register controls wakeup dependency based on RTCSS service requests.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												WKUPDEP_RTC_IRQ2_EVE4	WKUPDEP_RTC_IRQ2_EVE3	WKUPDEP_RTC_IRQ2_EVE2	WKUPDEP_RTC_IRQ2_EVE1	WKUPDEP_RTC_IRQ2_DSP2	WKUPDEP_RTC_IRQ2_IPU1	RESERVED	WKUPDEP_RTC_IRQ2_DSP1	WKUPDEP_RTC_IRQ2_IPU2	WKUPDEP_RTC_IRQ2_MPU	WKUPDEP_RTC_IRQ1_EVE4	WKUPDEP_RTC_IRQ1_EVE3	WKUPDEP_RTC_IRQ1_EVE2	WKUPDEP_RTC_IRQ1_EVE1	WKUPDEP_RTC_IRQ1_DSP2	WKUPDEP_RTC_IRQ1_IPU1	RESERVED	WKUPDEP_RTC_IRQ1_DSP1	WKUPDEP_RTC_IRQ1_IPU2	WKUPDEP_RTC_IRQ1_MPU

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	WKUPDEP_RTC_IRQ2_EVE4	Wakeup dependency from RTCSS module (timer_swakeup signal) towards EVE4 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
18	WKUPDEP_RTC_IRQ2_EVE3	Wakeup dependency from RTCSS module (timer_swakeup signal) towards EVE3 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
17	WKUPDEP_RTC_IRQ2_EVE2	Wakeup dependency from RTCSS module (timer_swakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
16	WKUPDEP_RTC_IRQ2_EVE1	Wakeup dependency from RTCSS module (timer_swakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
15	WKUPDEP_RTC_IRQ2_DSP2	Wakeup dependency from RTCSS module (timer_swakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
14	WKUPDEP_RTC_IRQ2_IPU1	Wakeup dependency from RTCSS module (timer_swakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
13	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
12	WKUPDEP_RTC_IRQ2_DSP1	Wakeup dependency from RTCSS module (timer_swakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
11	WKUPDEP_RTC_IRQ2_IPU2	Wakeup dependency from RTCSS module (timer_swakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
10	WKUPDEP_RTC_IRQ2_MPU	Wakeup dependency from RTCSS module (timer_swakeup signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
9	WKUPDEP_RTC_IRQ1_EVE4	Wakeup dependency from RTCSS module (alarm_swakeup signal) towards EVE4 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
8	WKUPDEP_RTC_IRQ1_EVE3	Wakeup dependency from RTCSS module (alarm_swakeup signal) towards EVE3 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
7	WKUPDEP_RTC_IRQ1_EVE2	Wakeup dependency from RTCSS module (alarm_swakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_RTC_IRQ1_EVE1	Wakeup dependency from RTCSS module (alarm_swakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_RTC_IRQ1_DSP2	Wakeup dependency from RTCSS module (alarm_swakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_RTC_IRQ1_IPU1	Wakeup dependency from RTCSS module (alarm_swakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	RESERVED		R	0x0
2	WKUPDEP_RTC_IRQ1_DSP1	Wakeup dependency from RTCSS module (alarm_swakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_RTC_IRQ1_IPU2	Wakeup dependency from RTCSS module (alarm_swakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
0	WKUPDEP_RTC_IRQ1_MPU	Wakeup dependency from RTCSS module (alarm_swakeup signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1699. RM_RTC_RTCSS_CONTEXT

Address Offset	0x0000 0004		
Physical Address	0x4AE0 7C64	**Instance**	RTC_PRM
Description	This register contains dedicated RTCSS context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of L4PER_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

3.12.54 ISS_PRM registers

3.12.54.1 ISS_PRM Register Summary

Table 3-1700. ISS_PRM Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	ISS_PRM Physical Address
PM_ISS_PWRSTCTRL	RW	32	0x0000 0000	0x4AE0 7C80
PM_ISS_PWRSTST	RW	32	0x0000 0004	0x4AE0 7C84
PM_ISS_ISS_WKDEP	RW	32	0x0000 0020	0x4AE0 7CA0
RM_ISS_ISS_CONTEXT	RW	32	0x0000 0024	0x4AE0 7CA4

3.12.54.2 ISS_PRM Register Description

Table 3-1701. PM_ISS_PWRSTCTRL

Address Offset	0x0000 0000		
Physical Address	0x4AE0 7C80	**Instance**	ISS_PRM
Description	This register controls the ISS power state to reach upon a domain sleep transition		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														ISS_BANK_ONSTATE		RESERVED							ISS_BANK_RETSTATE	RESERVED			LOWPOWERSTATECHANGE	RESERVED	LOGICRETSTATE	POWERSTATE	

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	ISS_BANK_ONSTATE	ISS_BANK state when domain is ON. 0x3: Memory bank is on when the domain is ON.	R	0x3
15:9	RESERVED		R	0x0
8	ISS_BANK_RETSTATE	ISS_BANK state when domain is RETENTION. 0x0: Memory bank is off when the domain is in the RETENTION state. 0x1: Memory bank is retained when domain is in RETENTION state.	RW	0x1
7:5	RESERVED		R	0x0
4	LOWPOWERSTATECHANGE	Power state change request when domain has already performed a sleep transition. Allows going into deeper low power state without waking up the power domain. 0x0: Do not request a low power state change. 0x1: Request a low power state change. This bit is automatically cleared when the power state is effectively changed or when power state is ON.	RW	0x0
3	RESERVED		R	0x0
2	LOGICRETSTATE	Logic state when power domain is RETENTION 0x0: Only retention registers are retained and remaing logic is off when the domain is in RETENTION state. 0x1: Whole logic is retained when domain is in RETENTION state.	RW	0x1
1:0	POWERSTATE	Power state control 0x0: OFF state 0x1: RETENTION state 0x2: INACTIVE state 0x3: ON State	RW	0x3

Table 3-1702. Register Call Summary for Register PM_ISS_PWRSTCTRL

Power Management Functional Description
- Logic and Memory Area Power Modes Control and Status: [0][1][2][3][4]

PRCM Register Manual
- Not Supported Functionality (Registers and Bitfields): [5][6][7][8]
- ISS_PRM Register Summary: [9]

Table 3-1703. PM_ISS_PWRSTST

Address Offset	0x0000 0004		
Physical Address	0x4AE0 7C84	**Instance**	ISS_PRM
Description	This register provides a status on the ISS domain current power state. [warm reset insensitive]		

Table 3-1703. PM_ISS_PWRSTST (continued)

Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED						LASTPOWERSTATEENTERED		RESERVED			INTRANSITION	RESERVED														ISS_BANK_STATEST		RESERVED		LOGICSTATEST		POWERSTATEST	

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:24	LASTPOWERSTATEENTERED	Last low power state entered. Set to 0x3 upon write of the same only. This register is intended for debug purpose only. 0x0: Power domain was previously OFF 0x1: Power domain was previously in RETENTION 0x2: Power domain was previously ON-INACTIVE 0x3: Power domain was previously ON-ACTIVE	RW	0x0
23:21	RESERVED		R	0x0
20	INTRANSITION	Domain transition status 0x0: No on-going transition on power domain 0x1: Power domain transition is in progress.	R	0x0
19:6	RESERVED		R	0x0
5:4	ISS_BANK_STATEST	ISS_BANK memory state status 0x0: Memory is OFF 0x1: Memory is RETENTION 0x2: Reserved 0x3: Memory is ON	R	0x3
3	RESERVED		R	0x0
2	LOGICSTATEST	Logic state status 0x0: Logic in domain is OFF 0x1: Logic in domain is ON	R	0x1
1:0	POWERSTATEST	Current power state status 0x0: Power domain is OFF 0x1: Power domain is in RETENTION 0x2: Power domain is ON-INACTIVE 0x3: Power domain is ON-ACTIVE	R	0x3

Table 3-1704. Register Call Summary for Register PM_ISS_PWRSTST

Power Management Functional Description

- Logic and Memory Area Power Modes Control and Status: [0][1][2][3][4]

PRCM Register Manual

- Not Supported Functionality (Registers and Bitfields): [5][6][7][8][9][10]
- ISS_PRM Register Summary: [11]

Table 3-1705. PM_ISS_ISS_WKDEP

Address Offset	0x0000 0020		
Physical Address	0x4AE0 7CA0	**Instance**	ISS_PRM
Description	This register controls wakeup dependency based on ISS service requests.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						WKUPDEP_ISS_EVE4	WKUPDEP_ISS_EVE3	WKUPDEP_ISS_EVE2	WKUPDEP_ISS_EVE1	WKUPDEP_ISS_DSP2	WKUPDEP_ISS_IPU1	RESERVED	WKUPDEP_ISS_DSP1	WKUPDEP_ISS_IPU2	WKUPDEP_ISS_MPU

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	WKUPDEP_ISS_EVE4	Wakeup dependency from ISS module (Swakeup signal) towards EVE4 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
8	WKUPDEP_ISS_EVE3	Wakeup dependency from ISS module (Swakeup signal) towards EVE3 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
7	WKUPDEP_ISS_EVE2	Wakeup dependency from ISS module (Swakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_ISS_EVE1	Wakeup dependency from ISS module (Swakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_ISS_DSP2	Wakeup dependency from ISS module (Swakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_ISS_IPU1	Wakeup dependency from ISS module (Swakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	RESERVED		R	0x0
2	WKUPDEP_ISS_DSP1	Wakeup dependency from ISS module (Swakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
1	WKUPDEP_ISS_IPU2	Wakeup dependency from ISS module (Swakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_ISS_MPU	Wakeup dependency from ISS module (Swakeup signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1706. Register Call Summary for Register PM_ISS_ISS_WKDEP

Clock Management Functional Description
- Clock Domain Dependency: [0][1][2][3]

PRCM Register Manual
- Not Supported Functionality (Registers and Bitfields): [4][5][6][7][8]
- ISS_PRM Register Summary: [9]

Table 3-1707. RM_ISS_ISS_CONTEXT

Address Offset	0x0000 0024		
Physical Address	0x4AE0 7CA4	**Instance**	ISS_PRM
Description	This register contains dedicated ISS context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							LOSTMEM_ISS_BANK	RESERVED							LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	LOSTMEM_ISS_BANK	Specify if memory-based context in ISS memory bank has been lost due to a previous power transition or other reset source. 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
7:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of ISS_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1708. Register Call Summary for Register RM_ISS_ISS_CONTEXT

Power Management Functional Description • PD_ISS Description: [0]
PRCM Register Manual • ISS_PRM Register Summary: [1]

3.12.55 WKUPAON_CM registers

3.12.55.1 WKUPAON_CM Register Summary

Table 3-1709. WKUPAON_CM Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	WKUPAON_CM Physical Address
CM_WKUPAON_CLKSTCTRL	RW	32	0x0000 0000	0x4AE0 7800
CM_WKUPAON_L4_WKUP_CLKCTRL	R	32	0x0000 0020	0x4AE0 7820
CM_WKUPAON_WD_TIMER1_CLKCTRL	R	32	0x0000 0028	0x4AE0 7828
CM_WKUPAON_WD_TIMER2_CLKCTRL	RW	32	0x0000 0030	0x4AE0 7830
CM_WKUPAON_GPIO1_CLKCTRL	RW	32	0x0000 0038	0x4AE0 7838
CM_WKUPAON_TIMER1_CLKCTRL	RW	32	0x0000 0040	0x4AE0 7840
CM_WKUPAON_TIMER12_CLKCTRL	R	32	0x0000 0048	0x4AE0 7848
CM_WKUPAON_COUNTER_32K_CLKCTRL	R	32	0x0000 0050	0x4AE0 7850
CM_WKUPAON_SAR_RAM_CLKCTRL	R	32	0x0000 0060	0x4AE0 7860
CM_WKUPAON_KBD_CLKCTRL	RW	32	0x0000 0078	0x4AE0 7878
CM_WKUPAON_UART10_CLKCTRL	RW	32	0x0000 0080	0x4AE0 7880
CM_WKUPAON_DCAN1_CLKCTRL	RW	32	0x0000 0088	0x4AE0 7888
CM_WKUPAON_SCRM_CLKCTRL	RW	32	0x0000 0090	0x4AE0 7890
CM_WKUPAON_IO_SRCOMP_CLKCTRL	RW	32	0x0000 0098	0x4AE0 7898
CM_WKUPAON_ADC_CLKCTRL	RW	32	0x0000 00A0	0x4AE0 78A0
CM_WKUPAON_SPARE_SAFETY1_CLKCTRL	R	32	0x0000 00B0	0x4AE0 78B0
CM_WKUPAON_RTI1_CLKCTRL	R	32	0x0000 00B8	0x4AE0 78B8
CM_WKUPAON_RTI2_CLKCTRL	R	32	0x0000 00C0	0x4AE0 78C0
CM_WKUPAON_RTI3_CLKCTRL	R	32	0x0000 00C8	0x4AE0 78C8
CM_WKUPAON_RTI4_CLKCTRL	R	32	0x0000 00D0	0x4AE0 78D0
CM_WKUPAON_RTI5_CLKCTRL	R	32	0x0000 00D8	0x4AE0 78D8

3.12.55.2 WKUPAON_CM Register Description

Table 3-1710. CM_WKUPAON_CLKSTCTRL

Address Offset	0x0000 0000		
Physical Address	0x4AE0 7800	**Instance**	WKUPAON_CM
Description	This register enables the domain power state transition. It controls the HW supervised domain power state transition between ON-ACTIVE and ON-INACTIVE states. It also hold one status bit per clock input of the domain.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												CLKACTIVITY_ADC_L3_GICLK	CLKACTIVITY_UART10_GFCLK	CLKACTIVITY_TIMER1_GFCLK	CLKACTIVITY_DCAN1_SYS_CLK	CLKACTIVITY_SYS_CLK_ALL	CLKACTIVITY_SYS_CLK_FUNC	CLKACTIVITY_WKUPAON_IO_SRCOMP_GFCLK	CLKACTIVITY_WKUPAON_GICLK	CLKACTIVITY_WKUPAON_SYS_GFCLK	CLKACTIVITY_ADC_GFCLK	CLKACTIVITY_ABE_LP_CLK	CLKACTIVITY_SYS_CLK	RESERVED						CLKTRCTRL	

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	CLKACTIVITY_ADC_L3_GICLK	This field indicates the state of the ADC_L3_GICLK clock in the domain(it includes profiling, EMU_SYS_GCLK and all functional SYS_CLK. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
18	CLKACTIVITY_UART10_GFCLK	This field indicates the state of the UART10_GFCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
17	CLKACTIVITY_TIMER1_GFCLK	This field indicates the state of the TIMER1_GFCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
16	CLKACTIVITY_DCAN1_SYS_CLK	This field indicates the state of the DCAN1_SYS_CLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
15	CLKACTIVITY_SYS_CLK_ALL	This field indicates the state of the SYS_CLK runing at SCRM level because of any SCRM clock request. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
14	CLKACTIVITY_SYS_CLK_FUNC	This field indicates the state of the functional SYS_CLK clocks in the domain (this exclude activity of EMU_GCLK clock). [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0

Bits	Field Name	Description	Type	Reset
13	CLKACTIVITY_WKUPAON_IO_SRCOMP_GFCLK	This field indicates the state of the WKUPAON_IO_SRCOMP_GFCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
12	CLKACTIVITY_WKUPAON_GICLK	This field indicates the state of the WKUPAON_GICLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
11	CLKACTIVITY_WKUPAON_SYS_GFCLK	This field indicates the state of the WKUPAON_SYS_GFCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
10	CLKACTIVITY_ADC_GFCLK	This field indicates the state of the ADC_GFCLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
9	CLKACTIVITY_ABE_LP_CLK	This field indicates the state of the ABE_LP_CLK clock in the domain. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
8	CLKACTIVITY_SYS_CLK	This field indicates the state of the SYS_CLK clock in the domain(it includes profiling, EMU_SYS_GCLK and all functional SYS_CLK. [warm reset insensitive] 0x0: Corresponding clock is definitely gated 0x1: Corresponding clock is running or gating/ungating transition is on-going	R	0x0
7:2	RESERVED		R	0x0
1:0	CLKTRCTRL	Controls the clock state transition of the WKUPAON clock domain. 0x0: NO_SLEEP: Sleep transition cannot be initiated. Wakeup transition may however occur. 0x1: Reserved 0x2: SW_WKUP: Start a software forced wake-up transition on the domain. 0x3: HW_AUTO: Automatic transition is enabled. Sleep and wakeup transition are based upon hardware conditions.	RW	0x0

Table 3-1711. Register Call Summary for Register CM_WKUPAON_CLKSTCTRL

Clock Management Functional Description • Clock Domain Modes: [0][1][2][3][4][7][8][9]
PRCM Register Manual • Not Supported Functionality (Registers and Bitfields): [10][11] • WKUPAON_CM Register Summary: [12]

Table 3-1712. CM_WKUPAON_L4_WKUP_CLKCTRL

Address Offset	0x0000 0020		
Physical Address	0x4AE0 7820	**Instance**	WKUPAON_CM
Description	This register manages the WKUPAON clocks.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														IDLEST		RESERVED								MODULEMODE							

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state.	R	0x1

Table 3-1713. Register Call Summary for Register CM_WKUPAON_L4_WKUP_CLKCTRL

Clock Management Functional Description
- Clock Domain Module Attributes: [0][1]

PRCM Register Manual
- WKUPAON_CM Register Summary: [2]

Table 3-1714. CM_WKUPAON_WD_TIMER1_CLKCTRL

Address Offset	0x0000 0028		
Physical Address	0x4AE0 7828	**Instance**	WKUPAON_CM
Description	This register manages the WD_TIMER1 clocks.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														IDLEST		RESERVED								MODULEMODE							

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state.	R	0x1

Table 3-1715. CM_WKUPAON_WD_TIMER2_CLKCTRL

Address Offset	0x0000 0030		
Physical Address	0x4AE0 7830	**Instance**	WKUPAON_CM
Description	This register manages the WD_TIMER2 clocks.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														IDLEST		RESERVED														MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guaranteed to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-1716. CM_WKUPAON_GPIO1_CLKCTRL

Address Offset	0x0000 0038		
Physical Address	0x4AE0 7838	**Instance**	WKUPAON_CM
Description	This register manages the GPIO1 clocks.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														IDLEST		RESERVED							OPTFCLKEN_DBCLK	RESERVED						MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:9	RESERVED		R	0x0
8	OPTFCLKEN_DBCLK	Optional functional clock control. 0x0: Optional functional clock is disabled 0x1: Optional functional clock is enabled	RW	0x0
7:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state. 0x2: Reserved 0x3: Reserved	RW	0x0

Table 3-1717. Register Call Summary for Register CM_WKUPAON_GPIO1_CLKCTRL

Clock Management Functional Description
- Clock Domain Modes: [0]
- Clock Domain Module Attributes: [1][2]

PRCM Register Manual
- WKUPAON_CM Register Summary: [3]

Table 3-1718. CM_WKUPAON_TIMER1_CLKCTRL

Address Offset	0x0000 0040		
Physical Address	0x4AE0 7840	**Instance**	WKUPAON_CM
Description	This register manages the TIMER1 clocks.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				CLKSEL				RESERVED						IDLEST		RESERVED														MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:24	CLKSEL	Select the source of the functional clock 0x0: Selects SYS_CLK1 0x1: Selects FUNC_32K_CLK 0x2: Selects SYS_CLK2 0x3: Selects XREF_CLK0 0x4: Selects XREF_CLK1 0x5: Selects XREF_CLK0 0x6: Selects XREF_CLK1 0x7: Selects ABE_GICLK 0x8: Selects divided version of SYS_CLK1. See CM_CLKSEL_VIDEO1_TIMER 0x9: Selects divided version of SYS_CLK1. See CM_CLKSEL_VIDEO2_TIMER 0xA: Selects divided version of SYS_CLK1. See CM_CLKSEL_HDMI_TIMER Others: RESERVED	RW	0x0
23:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guaranteed to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-1719. Register Call Summary for Register CM_WKUPAON_TIMER1_CLKCTRL

Clock Management Functional Description

- CM_CORE_AON_TIMER Overview: [0]
- Clock Domain Module Attributes: [1][2]

PRCM Register Manual

- Not Supported Functionality (Registers and Bitfields): [3][4][5][6][7]
- WKUPAON_CM Register Summary: [8]

Table 3-1720. CM_WKUPAON_TIMER12_CLKCTRL

Address Offset	0x0000 0048		
Physical Address	0x4AE0 7848	**Instance**	WKUPAON_CM
Description	This register manages the TIMER12 clocks.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														IDLEST		RESERVED								MODULEMODE							

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state.	R	0x1

Table 3-1721. CM_WKUPAON_COUNTER_32K_CLKCTRL

Address Offset	0x0000 0050		
Physical Address	0x4AE0 7850	**Instance**	WKUPAON_CM
Description	This register manages the COUNTER_32K clocks.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														IDLEST		RESERVED								MODULEMODE							

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state.	R	0x1

Table 3-1722. Register Call Summary for Register CM_WKUPAON_COUNTER_32K_CLKCTRL

Clock Management Functional Description • Clock Domain Module Attributes: [0][1]
PRCM Register Manual • WKUPAON_CM Register Summary: [2]

Table 3-1723. CM_WKUPAON_SAR_RAM_CLKCTRL

Address Offset	0x0000 0060		
Physical Address	0x4AE0 7860	**Instance**	WKUPAON_CM
Description	This register manages the SAR_RAM clocks.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														IDLEST		RESERVED														MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state.	R	0x1

Table 3-1724. CM_WKUPAON_KBD_CLKCTRL

Address Offset	0x0000 0078		
Physical Address	0x4AE0 7878	**Instance**	WKUPAON_CM
Description	This register manages the KBD clocks.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														IDLEST		RESERVED								MODULEMODE							

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guaranteed to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-1725. CM_WKUPAON_UART10_CLKCTRL

Address Offset	0x0000 0080		
Physical Address	0x4AE0 7880	**Instance**	WKUPAON_CM
Description	This register manages the UART10 clocks.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED							CLKSEL	RESERVED						IDLEST		RESERVED														MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24	CLKSEL	Selects functional clock for UART between FUNCTION_48M_CLK and FUNC_192M_CLK 0x0: Selects FUNC_48M_CLK 0x1: Selects FUNC_192M_CLK	RW	0x0
23:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guaranteed to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-1726. CM_WKUPAON_DCAN1_CLKCTRL

Address Offset	0x0000 0088		
Physical Address	0x4AE0 7888	**Instance**	WKUPAON_CM
Description	This register manages the DCAN1 clocks.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED							CLKSEL	RESERVED						IDLEST		RESERVED														MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24	CLKSEL	Selects functional clock for DCAN1 0x0: Selects SYS_CLK1 0x1: Selects SYS_CLK2	RW	0x0

Bits	Field Name	Description	Type	Reset
23:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guaranteed to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-1727. Register Call Summary for Register CM_WKUPAON_DCAN1_CLKCTRL

Clock Management Functional Description
- PRM Clock Source: [0]
- Clock Domain Module Attributes: [1][2]

PRCM Register Manual
- WKUPAON_CM Register Summary: [3]

Table 3-1728. CM_WKUPAON_SCRM_CLKCTRL

Address Offset	0x0000 0090		
Physical Address	0x4AE0 7890	**Instance**	WKUPAON_CM
Description	This register manages the SCRM clocks.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						OPTFCLKEN_SCRM_PER	OPTFCLKEN_SCRM_CORE	RESERVED							

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	OPTFCLKEN_SCRM_PER	Optional functional clock control. 0x0: Optional functional clock is disabled 0x1: Optional functional clock is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
8	OPTFCLKEN_SCRM_CORE	Optional functional clock control. 0x0: Optional functional clock is disabled 0x1: Optional functional clock is enabled	RW	0x0
7:0	RESERVED		R	0x0

Table 3-1729. CM_WKUPAON_IO_SRCOMP_CLKCTRL

Address Offset	0x0000 0098		
Physical Address	0x4AE0 7898	**Instance**	WKUPAON_CM
Description	This register manages the clock delivered to the IO Slew rate compensation cells. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							CLKEN_SRCOMP_FCLK	RESERVED							

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	CLKEN_SRCOMP_FCLK	Functional clock control. 0x0: Functional clock is disabled 0x1: Functional clock is enabled.	RW	0x0
7:0	RESERVED		R	0x0

Table 3-1730. CM_WKUPAON_ADC_CLKCTRL

Address Offset	0x0000 00A0		
Physical Address	0x4AE0 78A0	**Instance**	WKUPAON_CM
Description	This register manages the ADC clocks.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														IDLEST		RESERVED														MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x0: Module is disabled by SW. Any OCP access to module results in an error, except if resulting from a module wakeup (asynchronous wakeup). 0x1: Reserved 0x2: Module is explicitly enabled. Interface clock (if not used for functions) may be gated according to the clock domain state. Functional clocks are guaranteed to stay present. As long as in this configuration, power domain sleep transition cannot happen. 0x3: Reserved	RW	0x0

Table 3-1731. CM_WKUPAON_SPARE_SAFETY1_CLKCTRL

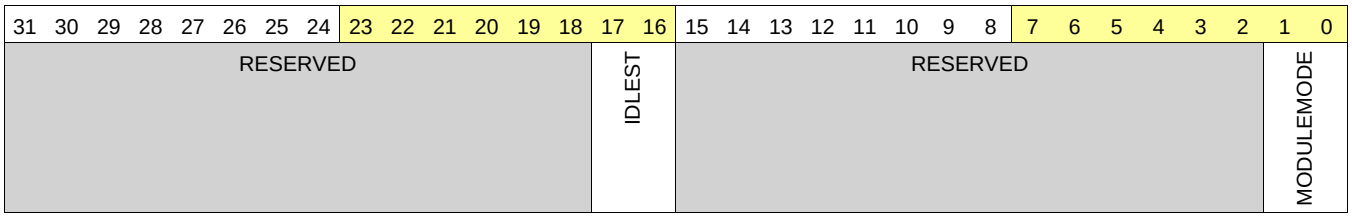
Address Offset	0x0000 00B0		
Physical Address	0x4AE0 78B0	**Instance**	WKUPAON_CM
Description	This register manages the SPARE_SAFETY1 clocks.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														IDLEST		RESERVED														MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state.	R	0x1

Table 3-1732. CM_WKUPAON_RTI1_CLKCTRL

Address Offset	0x0000 00B8		
Physical Address	0x4AE0 78B8	**Instance**	WKUPAON_CM
Description	This register manages the RTI1 clocks.		
Type	R		



Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state.	R	0x1

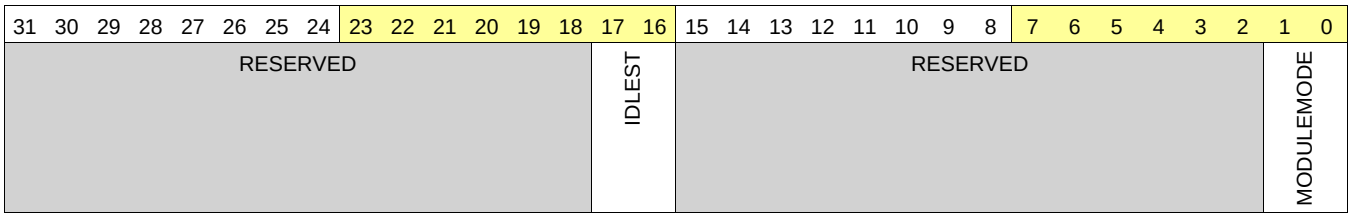
Table 3-1733. Register Call Summary for Register CM_WKUPAON_RTI1_CLKCTRL

Clock Management Functional Description
- Clock Domain Module Attributes: [0][1]

PRCM Register Manual
- WKUPAON_CM Register Summary: [2]

Table 3-1734. CM_WKUPAON_RTI2_CLKCTRL

Address Offset	0x0000 00C0		
Physical Address	0x4AE0 78C0	**Instance**	WKUPAON_CM
Description	This register manages the RTI2 clocks.		
Type	R		



Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state.	R	0x1

Table 3-1735. Register Call Summary for Register CM_WKUPAON_RTI2_CLKCTRL

Clock Management Functional Description • Clock Domain Module Attributes: [0][1]
PRCM Register Manual • WKUPAON_CM Register Summary: [2]

Table 3-1736. CM_WKUPAON_RTI3_CLKCTRL

Address Offset	0x0000 00C8		
Physical Address	0x4AE0 78C8	**Instance**	WKUPAON_CM
Description	This register manages the RTI3 clocks.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														IDLEST		RESERVED														MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state.	R	0x1

Table 3-1737. Register Call Summary for Register CM_WKUPAON_RTI3_CLKCTRL

Clock Management Functional Description • Clock Domain Module Attributes: [0][1]
PRCM Register Manual • WKUPAON_CM Register Summary: [2]

Table 3-1738. CM_WKUPAON_RTI4_CLKCTRL

Address Offset	0x0000 00D0		
Physical Address	0x4AE0 78D0	**Instance**	WKUPAON_CM
Description	This register manages the RTI4 clocks.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														IDLEST		RESERVED														MODULEMODE	

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state.	R	0x1

Table 3-1739. Register Call Summary for Register CM_WKUPAON_RTI4_CLKCTRL

Clock Management Functional Description • Clock Domain Module Attributes: [0][1]
PRCM Register Manual • WKUPAON_CM Register Summary: [2]

Table 3-1740. CM_WKUPAON_RTI5_CLKCTRL

Address Offset	0x0000 00D8		
Physical Address	0x4AE0 78D8	**Instance**	WKUPAON_CM
Description	This register manages the RTI5 clocks.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														IDLEST		RESERVED								MODULEMODE							

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17:16	IDLEST	Module idle status. [warm reset insensitive] 0x0: Module is fully functional, including OCP 0x1: Module is performing transition: wakeup, or sleep, or sleep abortion 0x2: Module is in Idle mode (only OCP part). It is functional if using separate functional clock 0x3: Module is disabled and cannot be accessed	R	0x3
15:2	RESERVED		R	0x0
1:0	MODULEMODE	Control the way mandatory clocks are managed. 0x1: Module is managed automatically by HW according to clock domain transition. A clock domain sleep transition put module into idle. A wakeup domain transition put it back into function. If CLKTRCTRL=3, any OCP access to module is always granted. Module clocks may be gated according to the clock domain state.	R	0x1

Table 3-1741. Register Call Summary for Register CM_WKUPAON_RTI5_CLKCTRL

Clock Management Functional Description
- Clock Domain Module Attributes: [0][1]

PRCM Register Manual
- WKUPAON_CM Register Summary: [2]

3.12.56 WKUPAON_PRM registers

3.12.56.1 WKUPAON_PRM Register Summary

Table 3-1742. WKUPAON_PRM Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	WKUPAON_PRM Physical Address
RM_WKUPAON_L4_WKUP_CONTEXT	RW	32	0x0000 0000	0x4AE0 7724
PM_WKUPAON_WD_TIMER1_WKDEP	RW	32	0x0000 0004	0x4AE0 7728
RM_WKUPAON_WD_TIMER1_CONTEXT	RW	32	0x0000 0008	0x4AE0 772C
PM_WKUPAON_WD_TIMER2_WKDEP	RW	32	0x0000 000C	0x4AE0 7730
RM_WKUPAON_WD_TIMER2_CONTEXT	RW	32	0x0000 0010	0x4AE0 7734
PM_WKUPAON_GPIO1_WKDEP	RW	32	0x0000 0014	0x4AE0 7738
RM_WKUPAON_GPIO1_CONTEXT	RW	32	0x0000 0018	0x4AE0 773C

Table 3-1742. WKUPAON_PRM Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	WKUPAON_PRM Physical Address
PM_WKUPAON_TIMER1_WKDEP	RW	32	0x0000 001C	0x4AE0 7740
RM_WKUPAON_TIMER1_CONTEXT	RW	32	0x0000 0020	0x4AE0 7744
PM_WKUPAON_TIMER12_WKDEP	RW	32	0x0000 0024	0x4AE0 7748
RM_WKUPAON_TIMER12_CONTEXT	RW	32	0x0000 0028	0x4AE0 774C
RM_WKUPAON_COUNTER_32K_CONTEXT	RW	32	0x0000 0030	0x4AE0 7754
RM_WKUPAON_SAR_RAM_CONTEXT	RW	32	0x0000 0040	0x4AE0 7764
PM_WKUPAON_KBD_WKDEP	RW	32	0x0000 0054	0x4AE0 7778
RM_WKUPAON_KBD_CONTEXT	RW	32	0x0000 0058	0x4AE0 777C
PM_WKUPAON_UART10_WKDEP	RW	32	0x0000 005C	0x4AE0 7780
RM_WKUPAON_UART10_CONTEXT	RW	32	0x0000 0060	0x4AE0 7784
PM_WKUPAON_DCAN1_WKDEP	RW	32	0x0000 0064	0x4AE0 7788
RM_WKUPAON_DCAN1_CONTEXT	RW	32	0x0000 0068	0x4AE0 778C
PM_WKUPAON_ADC_WKDEP	RW	32	0x0000 007C	0x4AE0 77A0
RM_WKUPAON_ADC_CONTEXT	RW	32	0x0000 0080	0x4AE0 77A4
RM_WKUPAON_SPARE_SAFETY1_CONTEXT	RW	32	0x0000 0090	0x4AE0 77B4
RM_WKUPAON_RTI1_CONTEXT	RW	32	0x0000 0098	0x4AE0 77BC
RM_WKUPAON_RTI2_CONTEXT	RW	32	0x0000 00A0	0x4AE0 77C4
RM_WKUPAON_RTI3_CONTEXT	RW	32	0x0000 00A8	0x4AE0 77CC
RM_WKUPAON_RTI4_CONTEXT	RW	32	0x0000 00B0	0x4AE0 77D4
RM_WKUPAON_RTI5_CONTEXT	RW	32	0x0000 00B8	0x4AE0 77DC

3.12.56.2 WKUPAON_PRM Register Description

Table 3-1743. RM_WKUPAON_L4_WKUP_CONTEXT

Address Offset	0x0000 0000		
Physical Address	0x4AE0 7724	**Instance**	WKUPAON_PRM
Description	This register contains dedicated L4_WKUP context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of WKUPAON_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1744. Register Call Summary for Register RM_WKUPAON_L4_WKUP_CONTEXT

Power Management Functional Description • PD_WKUPAON Description: [0]
PRCM Register Manual • WKUPAON_PRM Register Summary: [1]

Table 3-1745. PM_WKUPAON_WD_TIMER1_WKDEP

Address Offset	0x0000 0004		
Physical Address	0x4AE0 7728	**Instance**	WKUPAON_PRM
Description	This register controls wakeup dependency based on WD_TIMER1 service requests.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						WKUPDEP_WD_TIMER1_EVE4	WKUPDEP_WD_TIMER1_EVE3	WKUPDEP_WD_TIMER1_EVE2	WKUPDEP_WD_TIMER1_EVE1	WKUPDEP_WD_TIMER1_DSP2	WKUPDEP_WD_TIMER1_IPU1	RESERVED	WKUPDEP_WD_TIMER1_DSP1	WKUPDEP_WD_TIMER1_IPU2	WKUPDEP_WD_TIMER1_MPU

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	WKUPDEP_WD_TIMER1_EVE4	Wakeup dependency from TIMER module (SWakeup IRQ signal) towards EVE4 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
8	WKUPDEP_WD_TIMER1_EVE3	Wakeup dependency from TIMER module (SWakeup IRQ signal) towards EVE3 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
7	WKUPDEP_WD_TIMER1_EVE2	Wakeup dependency from TIMER module (SWakeup IRQ signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_WD_TIMER1_EVE1	Wakeup dependency from TIMER module (SWakeup IRQ signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_WD_TIMER1_DSP2	Wakeup dependency from TIMER module (SWakeup IRQ signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
4	WKUPDEP_WD_TIMER1_IPU1	Wakeup dependency from TIMER module (SWakeup IRQ signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	RESERVED		R	0x0
2	WKUPDEP_WD_TIMER1_DSP1	Wakeup dependency from TIMER module (SWakeup IRQ signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_WD_TIMER1_IPU2	Wakeup dependency from TIMER module (SWakeup IRQ signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_WD_TIMER1_MPU	Wakeup dependency from WDT1 module (SWakeup signal) towards MPU + L3MAIN1 + L4CFG domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1746. RM_WKUPAON_WD_TIMER1_CONTEXT

Address Offset	0x0000 0008		
Physical Address	0x4AE0 772C	**Instance**	WKUPAON_PRM
Description	This register contains dedicated WD_TIMER1 context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of WKUPAON_PWRON_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1747. PM_WKUPAON_WD_TIMER2_WKDEP

Address Offset	0x0000 000C		
Physical Address	0x4AE0 7730	**Instance**	WKUPAON_PRM
Description	This register controls wakeup dependency based on WD_TIMER2 service requests.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						WKUPDEP_WD_TIMER2_EVE4	WKUPDEP_WD_TIMER2_EVE3	WKUPDEP_WD_TIMER2_EVE2	WKUPDEP_WD_TIMER2_EVE1	WKUPDEP_WD_TIMER2_DSP2	WKUPDEP_WD_TIMER2_IPU1	RESERVED	WKUPDEP_WD_TIMER2_DSP1	WKUPDEP_WD_TIMER2_IPU2	WKUPDEP_WD_TIMER2_MPU

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	WKUPDEP_WD_TIMER2_EVE4	Wakeup dependency from TIMER module (SWakeup IRQ signal) towards EVE4 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
8	WKUPDEP_WD_TIMER2_EVE3	Wakeup dependency from TIMER module (SWakeup IRQ signal) towards EVE3 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
7	WKUPDEP_WD_TIMER2_EVE2	Wakeup dependency from TIMER module (SWakeup IRQ signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_WD_TIMER2_EVE1	Wakeup dependency from TIMER module (SWakeup IRQ signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_WD_TIMER2_DSP2	Wakeup dependency from TIMER module (SWakeup IRQ signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_WD_TIMER2_IPU1	Wakeup dependency from TIMER module (SWakeup IRQ signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	RESERVED		R	0x0
2	WKUPDEP_WD_TIMER2_DSP1	Wakeup dependency from TIMER module (SWakeup IRQ signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_WD_TIMER2_IPU2	Wakeup dependency from TIMER module (SWakeup IRQ signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
0	WKUPDEP_WD_TIMER2_MPU	Wakeup dependency from WDT2 module (SWakeup signal) towards MPU + L3MAIN1 + L4CFG domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1748. RM_WKUPAON_WD_TIMER2_CONTEXT

Address Offset	0x0000 0010		
Physical Address	0x4AE0 7734	**Instance**	WKUPAON_PRM
Description	This register contains dedicated WD_TIMER2 context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of WKUPAON_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1749. PM_WKUPAON_GPIO1_WKDEP

Address Offset	0x0000 0014		
Physical Address	0x4AE0 7738	**Instance**	WKUPAON_PRM
Description	This register controls wakeup dependency based on GPIO1 service requests.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												WKUPDEP_GPIO1_IRQ2_EVE4	WKUPDEP_GPIO1_IRQ2_EVE3	WKUPDEP_GPIO1_IRQ2_EVE2	WKUPDEP_GPIO1_IRQ2_EVE1	WKUPDEP_GPIO1_IRQ2_DSP2	WKUPDEP_GPIO1_IRQ2_IPU1	RESERVED	WKUPDEP_GPIO1_IRQ2_DSP1	WKUPDEP_GPIO1_IRQ2_IPU2	WKUPDEP_GPIO1_IRQ2_MPU	WKUPDEP_GPIO1_IRQ1_EVE4	WKUPDEP_GPIO1_IRQ1_EVE3	WKUPDEP_GPIO1_IRQ1_EVE2	WKUPDEP_GPIO1_IRQ1_EVE1	WKUPDEP_GPIO1_IRQ1_DSP2	WKUPDEP_GPIO1_IRQ1_IPU1	RESERVED	WKUPDEP_GPIO1_IRQ1_DSP1	WKUPDEP_GPIO1_IRQ1_IPU2	WKUPDEP_GPIO1_IRQ1_MPU

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	WKUPDEP_GPIO1_IRQ2_EVE4	Wakeup dependency from GPIO1 module (SWakeup signal) towards EVE4 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
18	WKUPDEP_GPIO1_IRQ2_EVE3	Wakeup dependency from GPIO1 module (SWakeup signal) towards EVE3 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
17	WKUPDEP_GPIO1_IRQ2_EVE2	Wakeup dependency from GPIO1 module (SWakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
16	WKUPDEP_GPIO1_IRQ2_EVE1	Wakeup dependency from GPIO1 module (SWakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
15	WKUPDEP_GPIO1_IRQ2_DSP2	Wakeup dependency from GPIO1 module (SWakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
14	WKUPDEP_GPIO1_IRQ2_IPU1	Wakeup dependency from GPIO1 module (SWakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
13	RESERVED		R	0x0
12	WKUPDEP_GPIO1_IRQ2_DSP1	Wakeup dependency from GPIO1 module (SWakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
11	WKUPDEP_GPIO1_IRQ2_IPU2	Wakeup dependency from GPIO1 module (SWakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
10	WKUPDEP_GPIO1_IRQ2_MPU	Wakeup dependency from GPIO1 module (SWakeup signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
9	WKUPDEP_GPIO1_IRQ1_EVE4	Wakeup dependency from GPIO1 module (SWakeup signal) towards EVE4 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
8	WKUPDEP_GPIO1_IRQ1_EVE3	Wakeup dependency from GPIO1 module (SWakeup signal) towards EVE3 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
7	WKUPDEP_GPIO1_IRQ1_EVE2	Wakeup dependency from GPIO1 module (SWakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_GPIO1_IRQ1_EVE1	Wakeup dependency from GPIO1 module (SWakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_GPIO1_IRQ1_DSP2	Wakeup dependency from GPIO1 module (SWakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_GPIO1_IRQ1_IPU1	Wakeup dependency from GPIO1 module (SWakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	RESERVED		R	0x0
2	WKUPDEP_GPIO1_IRQ1_DSP1	Wakeup dependency from GPIO1 module (SWakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_GPIO1_IRQ1_IPU2	Wakeup dependency from GPIO1 module (SWakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_GPIO1_IRQ1_MPU	Wakeup dependency from GPIO1 module (SWakeup signal for POROCPSINTERRUPT1) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1750. Register Call Summary for Register PM_WKUPAON_GPIO1_WKDEP

Clock Management Functional Description
- Clock Domain Dependency: [0][1][2][3][4][5][6][7]

PRCM Register Manual
- Not Supported Functionality (Registers and Bitfields): [8][9][10][11][12][13][14][15][16][17]
- WKUPAON_PRM Register Summary: [18]

Table 3-1751. RM_WKUPAON_GPIO1_CONTEXT

Address Offset	0x0000 0018		
Physical Address	0x4AE0 773C	**Instance**	WKUPAON_PRM
Description	This register contains dedicated GPIO1 context statuses. [warm reset insensitive]		

Table 3-1751. RM_WKUPAON_GPIO1_CONTEXT (continued)

Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of WKUPAON_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1752. Register Call Summary for Register RM_WKUPAON_GPIO1_CONTEXT

Power Management Functional Description
- PD_WKUPAON Description: [0]

PRCM Register Manual
- WKUPAON_PRM Register Summary: [1]

Table 3-1753. PM_WKUPAON_TIMER1_WKDEP

Address Offset	0x0000 001C		
Physical Address	0x4AE0 7740	**Instance**	WKUPAON_PRM
Description	This register controls wakeup dependency based on TIMER1 service requests.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						WKUPDEP_TIMER1_EVE4	WKUPDEP_TIMER1_EVE3	WKUPDEP_TIMER1_EVE2	WKUPDEP_TIMER1_EVE1	WKUPDEP_TIMER1_DSP2	WKUPDEP_TIMER1_IPU1	RESERVED	WKUPDEP_TIMER1_DSP1	WKUPDEP_TIMER1_IPU2	WKUPDEP_TIMER1_MPU

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	WKUPDEP_TIMER1_EVE4	Wakeup dependency from TIMER1 module (SWakeup IRQ signal) towards EVE4 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
8	WKUPDEP_TIMER1_EVE3	Wakeup dependency from TIMER1 module (SWakeup IRQ signal) towards EVE3 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
7	WKUPDEP_TIMER1_EVE2	Wakeup dependency from TIMER1 module (SWakeup IRQ signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_TIMER1_EVE1	Wakeup dependency from TIMER1 module (SWakeup IRQ signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_TIMER1_DSP2	Wakeup dependency from TIMER1 module (SWakeup IRQ signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_TIMER1_IPU1	Wakeup dependency from TIMER1 module (SWakeup IRQ signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	RESERVED		R	0x0
2	WKUPDEP_TIMER1_DSP1	Wakeup dependency from TIMER1 module (SWakeup IRQ signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_TIMER1_IPU2	Wakeup dependency from TIMER4 module (SWakeup IRQ signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_TIMER1_MPU	Wakeup dependency from TIMER1 module (SWakeup IRQ signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1754. Register Call Summary for Register PM_WKUPAON_TIMER1_WKDEP

Clock Management Functional Description • Clock Domain Dependency: [0][1][2][3]
PRCM Register Manual • Not Supported Functionality (Registers and Bitfields): [4][5][6][7][8] • WKUPAON_PRM Register Summary: [9]

Table 3-1755. RM_WKUPAON_TIMER1_CONTEXT

Address Offset	0x0000 0020		
Physical Address	0x4AE0 7744	**Instance**	WKUPAON_PRM
Description	This register contains dedicated TIMER1 context statuses. [warm reset insensitive]		

Table 3-1755. RM_WKUPAON_TIMER1_CONTEXT (continued)

Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of WKUPAON_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1756. Register Call Summary for Register RM_WKUPAON_TIMER1_CONTEXT

Power Management Functional Description
- PD_WKUPAON Description: [0]

PRCM Register Manual
- WKUPAON_PRM Register Summary: [1]

Table 3-1757. PM_WKUPAON_TIMER12_WKDEP

Address Offset	0x0000 0024		
Physical Address	0x4AE0 7748	Instance	WKUPAON_PRM
Description	This register controls wakeup dependency based on TIMER12 service requests.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						WKUPDEP_TIMER12_EVE4	WKUPDEP_TIMER12_EVE3	WKUPDEP_TIMER12_EVE2	WKUPDEP_TIMER12_EVE1	WKUPDEP_TIMER12_DSP2	WKUPDEP_TIMER12_IPU1	RESERVED	WKUPDEP_TIMER12_DSP1	WKUPDEP_TIMER12_IPU2	WKUPDEP_TIMER12_MPU

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	WKUPDEP_TIMER12_EVE4	Wakeup dependency from TIMER12 module (SWakeup IRQ signal) towards EVE4 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
8	WKUPDEP_TIMER12_EVE3	Wakeup dependency from TIMER12 module (SWakeup IRQ signal) towards EVE3 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
7	WKUPDEP_TIMER12_EVE2	Wakeup dependency from TIMER12 module (SWakeup IRQ signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_TIMER12_EVE1	Wakeup dependency from TIMER12 module (SWakeup IRQ signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_TIMER12_DSP2	Wakeup dependency from TIMER12 module (SWakeup IRQ signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_TIMER12_IPU1	Wakeup dependency from TIMER12 module (SWakeup IRQ signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	RESERVED		R	0x0
2	WKUPDEP_TIMER12_DSP1	Wakeup dependency from TIMER12 module (SWakeup IRQ signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_TIMER12_IPU2	Wakeup dependency from TIMER12 module (SWakeup IRQ signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_TIMER12_MPU	Wakeup dependency from TIMER12 module (SWakeup IRQ signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1758. RM_WKUPAON_TIMER12_CONTEXT

Address Offset	0x0000 0028		
Physical Address	0x4AE0 774C	**Instance**	WKUPAON_PRM
Description	This register contains dedicated TIMER12 context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of WKUPAON_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1759. RM_WKUPAON_COUNTER_32K_CONTEXT

Address Offset	0x0000 0030		
Physical Address	0x4AE0 7754	**Instance**	WKUPAON_PRM
Description	This register contains dedicated COUNTER_32K context statuses. This bit-field is only sensitive to the external power-on reset (SYS_PWRON_RST reset line)		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of WKUPAON_SYS_PWRON_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1760. Register Call Summary for Register RM_WKUPAON_COUNTER_32K_CONTEXT

Power Management Functional Description
- PD_WKUPAON Description: [0]

PRCM Register Manual
- WKUPAON_PRM Register Summary: [1]

Table 3-1761. RM_WKUPAON_SAR_RAM_CONTEXT

Address Offset	0x0000 0040		
Physical Address	0x4AE0 7764	**Instance**	WKUPAON_PRM
Description	This register contains dedicated SAR_RAM context statuses. [warm reset insensitive]		

Table 3-1761. RM_WKUPAON_SAR_RAM_CONTEXT (continued)

Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							LOSTMEM_WKUP_BANK	RESERVED							LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	LOSTMEM_WKUP_BANK	Specify if memory-based context in WKUP_BANK memory bank has been lost due to a previous global cold reset. 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
7:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of WKUPAON_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1762. PM_WKUPAON_KBD_WKDEP

Address Offset	0x0000 0054		
Physical Address	0x4AE0 7778	Instance	WKUPAON_PRM
Description	This register controls wakeup dependency based on KBD service requests.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						WKUPDEP_KBD_EVE4	WKUPDEP_KBD_EVE3	WKUPDEP_KBD_EVE2	WKUPDEP_KBD_EVE1	WKUPDEP_KBD_DSP2	WKUPDEP_KBD_IPU1	RESERVED	WKUPDEP_KBD_DSP1	WKUPDEP_KBD_IPU2	WKUPDEP_KBD_MPU

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	WKUPDEP_KBD_EVE4	Wakeup dependency from KBD module (SWakeup IRQ signal) towards EVE4 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
8	WKUPDEP_KBD_EVE3	Wakeup dependency from KBD module (SWakeup IRQ signal) towards EVE3 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
7	WKUPDEP_KBD_EVE2	Wakeup dependency from KBD module (SWakeup IRQ signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_KBD_EVE1	Wakeup dependency from KBD module (SWakeup IRQ signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_KBD_DSP2	Wakeup dependency from KBD module (SWakeup IRQ signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_KBD_IPU1	Wakeup dependency from KBD module (SWakeup IRQ signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	RESERVED		R	0x0
2	WKUPDEP_KBD_DSP1	Wakeup dependency from KBD module (SWakeup IRQ signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_KBD_IPU2	Wakeup dependency from KBD module (SWakeup IRQ signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_KBD_MPU	Wakeup dependency from KBD module (SWakeup IRQ signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1763. RM_WKUPAON_KBD_CONTEXT

Address Offset	0x0000 0058		
Physical Address	0x4AE0 777C	**Instance**	WKUPAON_PRM
Description	This register contains dedicated KBD context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of WKUPAON_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1764. PM_WKUPAON_UART10_WKDEP

Address Offset	0x0000 005C		
Physical Address	0x4AE0 7780	**Instance**	WKUPAON_PRM
Description	This register controls wakeup dependency based on UART10 service requests.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						WKUPDEP_UART10_EVE4	WKUPDEP_UART10_EVE3	WKUPDEP_UART10_EVE2	WKUPDEP_UART10_EVE1	WKUPDEP_UART10_DSP2	WKUPDEP_UART10_IPU1	WKUPDEP_UART10_SDMA	WKUPDEP_UART10_DSP1	WKUPDEP_UART10_IPU2	WKUPDEP_UART10_MPU

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	WKUPDEP_UART10_EVE4	Wakeup dependency from UART10 module (SWakeup signal) towards EVE4 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
8	WKUPDEP_UART10_EVE3	Wakeup dependency from UART10 module (SWakeup signal) towards EVE3 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
7	WKUPDEP_UART10_EVE2	Wakeup dependency from UART10 module (SWakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
6	WKUPDEP_UART10_EVE1	Wakeup dependency from UART10 module (SWakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_UART10_DSP2	Wakeup dependency from UART10 module (SWakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_UART10_IPU1	Wakeup dependency from UART10 module (SWakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	WKUPDEP_UART10_SDMA	Wakeup dependency from UART10 module (SWakeup signal) towards SDMA + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
2	WKUPDEP_UART10_DSP1	Wakeup dependency from UART10 module (SWakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_UART10_IPU2	Wakeup dependency from UART10 module (SWakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_UART10_MPU	Wakeup dependency from UART10 module (SWakeup signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1765. RM_WKUPAON_UART10_CONTEXT

Address Offset	0x0000 0060		
Physical Address	0x4AE0 7784	**Instance**	WKUPAON_PRM
Description	This register contains dedicated UART10 context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							LOSTMEM_RETAINED_BANK	RESERVED							LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	LOSTMEM_RETAINED_BANK	Specify if memory-based context in UART memory bank has been lost due to a previous power transition or other reset source. 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
7:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of WKUPAON_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1766. PM_WKUPAON_DCAN1_WKDEP

Address Offset	0x0000 0064		
Physical Address	0x4AE0 7788	**Instance**	WKUPAON_PRM
Description	This register controls wakeup dependency based on DCAN1 service requests.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						WKUPDEP_DCAN1_EVE4	WKUPDEP_DCAN1_EVE3	WKUPDEP_DCAN1_EVE2	WKUPDEP_DCAN1_EVE1	WKUPDEP_DCAN1_DSP2	WKUPDEP_DCAN1_IPU1	WKUPDEP_DCAN1_SDMA	WKUPDEP_DCAN1_DSP1	WKUPDEP_DCAN1_IPU2	WKUPDEP_DCAN1_MPU

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	WKUPDEP_DCAN1_EVE4	Wakeup dependency from DCAN1 module (SWakeup signal) towards EVE4 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
8	WKUPDEP_DCAN1_EVE3	Wakeup dependency from DCAN1 module (SWakeup signal) towards EVE3 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
7	WKUPDEP_DCAN1_EVE2	Wakeup dependency from DCAN1 module (SWakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
6	WKUPDEP_DCAN1_EVE1	Wakeup dependency from DCAN1 module (SWakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
5	WKUPDEP_DCAN1_DSP2	Wakeup dependency from DCAN1 module (SWakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_DCAN1_IPU1	Wakeup dependency from DCAN1 module (SWakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	WKUPDEP_DCAN1_SDMA	Wakeup dependency from DCAN1 module (SWakeup signal) towards SDMA + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
2	WKUPDEP_DCAN1_DSP1	Wakeup dependency from DCAN1 module (SWakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_DCAN1_IPU2	Wakeup dependency from DCAN1 module (SWakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_DCAN1_MPU	Wakeup dependency from DCAN1 module (SWakeup signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1767. Register Call Summary for Register PM_WKUPAON_DCAN1_WKDEP

PRCM Register Manual
- Not Supported Functionality (Registers and Bitfields): [0][1][2][3][4][5]
- WKUPAON_PRM Register Summary: [6]

Table 3-1768. RM_WKUPAON_DCAN1_CONTEXT

Address Offset	0x0000 0068		
Physical Address	0x4AE0 778C	**Instance**	WKUPAON_PRM
Description	This register contains dedicated DCAN1 context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							LOSTMEM_DCAN_MEM	RESERVED							LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	LOSTMEM_DCAN_MEM	Specify if memory-based context in DCAN memory bank has been lost due to a previous power transition or other reset source. 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1
7:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of WKUPAON_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1769. Register Call Summary for Register RM_WKUPAON_DCAN1_CONTEXT

Power Management Functional Description
- PD_WKUPAON Description: [0]

PRCM Register Manual
- WKUPAON_PRM Register Summary: [1]

Table 3-1770. PM_WKUPAON_ADC_WKDEP

Address Offset	0x0000 007C		
Physical Address	0x4AE0 77A0	**Instance**	WKUPAON_PRM
Description	This register controls wakeup dependency based on ADC service requests.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						WKUPDEP_ADC_EVE4	WKUPDEP_ADC_EVE3	WKUPDEP_ADC_EVE2	WKUPDEP_ADC_EVE1	WKUPDEP_ADC_DSP2	WKUPDEP_ADC_IPU1	RESERVED	WKUPDEP_ADC_DSP1	WKUPDEP_ADC_IPU2	WKUPDEP_ADC_MPU

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	WKUPDEP_ADC_EVE4	Wakeup dependency from ADC module (SWakeup signal) towards EVE4 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
8	WKUPDEP_ADC_EVE3	Wakeup dependency from ADC module (SWakeup signal) towards EVE3 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
7	WKUPDEP_ADC_EVE2	Wakeup dependency from ADC module (SWakeup signal) towards EVE2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
6	WKUPDEP_ADC_EVE1	Wakeup dependency from ADC module (SWakeup signal) towards EVE1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
5	WKUPDEP_ADC_DSP2	Wakeup dependency from ADC module (SWakeup signal) towards DSP2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
4	WKUPDEP_ADC_IPU1	Wakeup dependency from ADC module (SWakeup signal) towards IPU1 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
3	RESERVED		R	0x0
2	WKUPDEP_ADC_DSP1	Wakeup dependency from ADC module (SWakeup signal) towards DSP + L3MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
1	WKUPDEP_ADC_IPU2	Wakeup dependency from ADC module (SWakeup signal) towards IPU2 + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0
0	WKUPDEP_ADC_MPU	Wakeup dependency from ADC module (SWakeup signal) towards MPU + L3_MAIN1 + L4PER1 + L4PER2 + L4PER3 domains 0x0: Dependency is disabled 0x1: Dependency is enabled	RW	0x0

Table 3-1771. RM_WKUPAON_ADC_CONTEXT

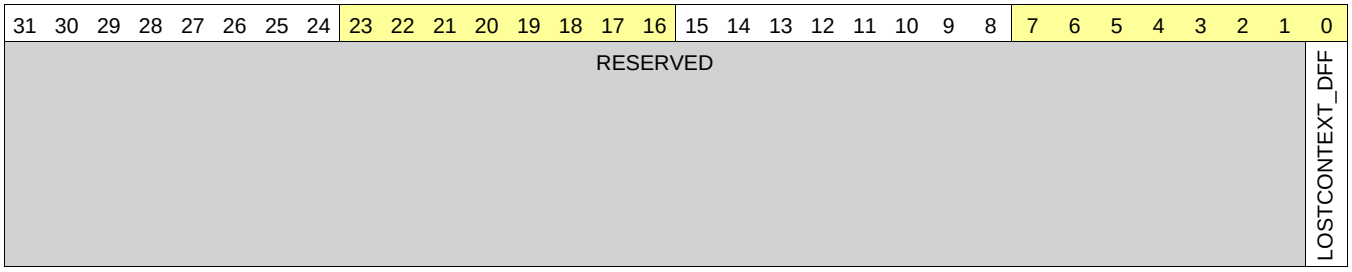
Address Offset	0x0000 0080		
Physical Address	0x4AE0 77A4	**Instance**	WKUPAON_PRM
Description	This register contains dedicated ADC context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of CAM_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1772. RM_WKUPAON_SPARE_SAFETY1_CONTEXT

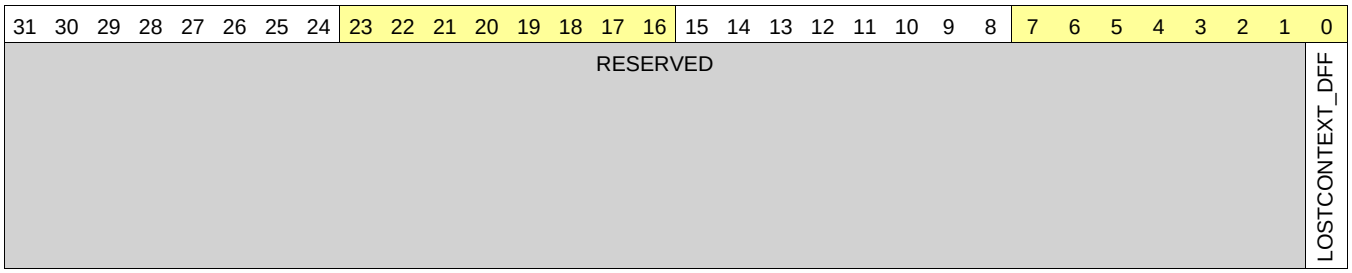
Address Offset	0x0000 0090		
Physical Address	0x4AE0 77B4	**Instance**	WKUPAON_PRM
Description	This register contains dedicated SPARE_SAFETY1 context statuses. [warm reset insensitive]		
Type	RW		



Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of WKUPAON_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1773. RM_WKUPAON_RTI1_CONTEXT

Address Offset	0x0000 0098		
Physical Address	0x4AE0 77BC	**Instance**	WKUPAON_PRM
Description	This register contains dedicated RTI1 context statuses. [warm reset insensitive]		
Type	RW		



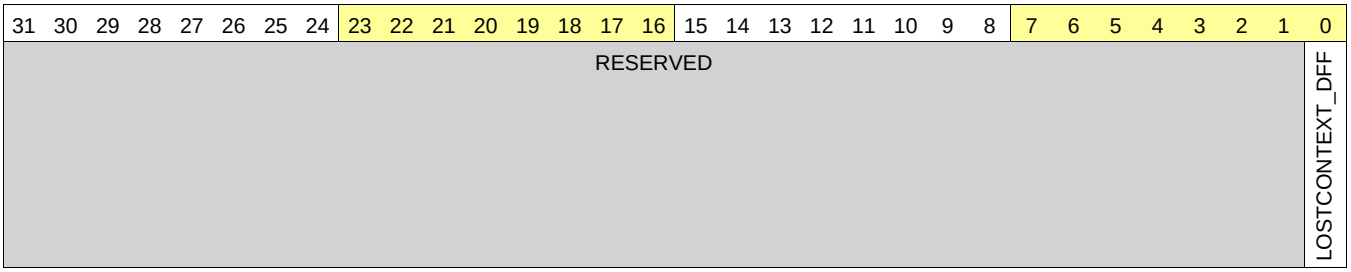
Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of WKUPAON_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1774. Register Call Summary for Register RM_WKUPAON_RTI1_CONTEXT

PRCM Register Manual
- WKUPAON_PRM Register Summary: [0]

Table 3-1775. RM_WKUPAON_RTI2_CONTEXT

Address Offset	0x0000 00A0		
Physical Address	0x4AE0 77C4	**Instance**	WKUPAON_PRM
Description	This register contains dedicated RTI2 context statuses. [warm reset insensitive]		
Type	RW		



Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of WKUPAON_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1776. Register Call Summary for Register RM_WKUPAON_RTI2_CONTEXT

PRCM Register Manual
- WKUPAON_PRM Register Summary: [0]

Table 3-1777. RM_WKUPAON_RTI3_CONTEXT

Address Offset	0x0000 00A8		
Physical Address	0x4AE0 77CC	**Instance**	WKUPAON_PRM
Description	This register contains dedicated RTI3 context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of WKUPAON_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1778. Register Call Summary for Register RM_WKUPAON_RTI3_CONTEXT

PRCM Register Manual
- WKUPAON_PRM Register Summary: [0]

Table 3-1779. RM_WKUPAON_RTI4_CONTEXT

Address Offset	0x0000 00B0		
Physical Address	0x4AE0 77D4	**Instance**	WKUPAON_PRM
Description	This register contains dedicated RTI4 context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of WKUPAON_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1780. Register Call Summary for Register RM_WKUPAON_RTI4_CONTEXT

PRCM Register Manual
- WKUPAON_PRM Register Summary: [0]

Table 3-1781. RM_WKUPAON_RTI5_CONTEXT

Address Offset	0x0000 00B8		
Physical Address	0x4AE0 77DC	**Instance**	WKUPAON_PRM
Description	This register contains dedicated RTI5 context statuses. [warm reset insensitive]		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															LOSTCONTEXT_DFF

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	LOSTCONTEXT_DFF	Specify if DFF-based context has been lost due to a previous power transition or other reset source. (set upon assertion of WKUPAON_RST signal) 0x0: Context has been maintained 0x1: Context has been lost	RW	0x1

Table 3-1782. Register Call Summary for Register RM_WKUPAON_RTI5_CONTEXT

PRCM Register Manual
- WKUPAON_PRM Register Summary: [0]

DSP Subsystems

This chapter describes the features and functions of the device integrated digital processing subsystems.

NOTE: Devices may include up to two identical instances of DSP subsystem (DSP1 and DSP2). Refer to the device *Data Manual* for number of DSP instances on a specific device. This chapter describes the two DSP device superset. For devices with a single DSP, the information on DSP2 can be ignored.

Topic **Page**

4.1 DSP Subsystems Overview

The device includes two identical instances (DSP1 and DSP2) of a digital signal processor (DSP) subsystem, based on the TI's standard TMS320C66x DSP CorePac core.

The TMS320C66x DSP core enhances the TMS320C674x core, which merges the C674x floating point and the C64x+ fixed-point instruction set architectures. The C66x DSP is object-code compatible with the C64x+/C674x DSPs.

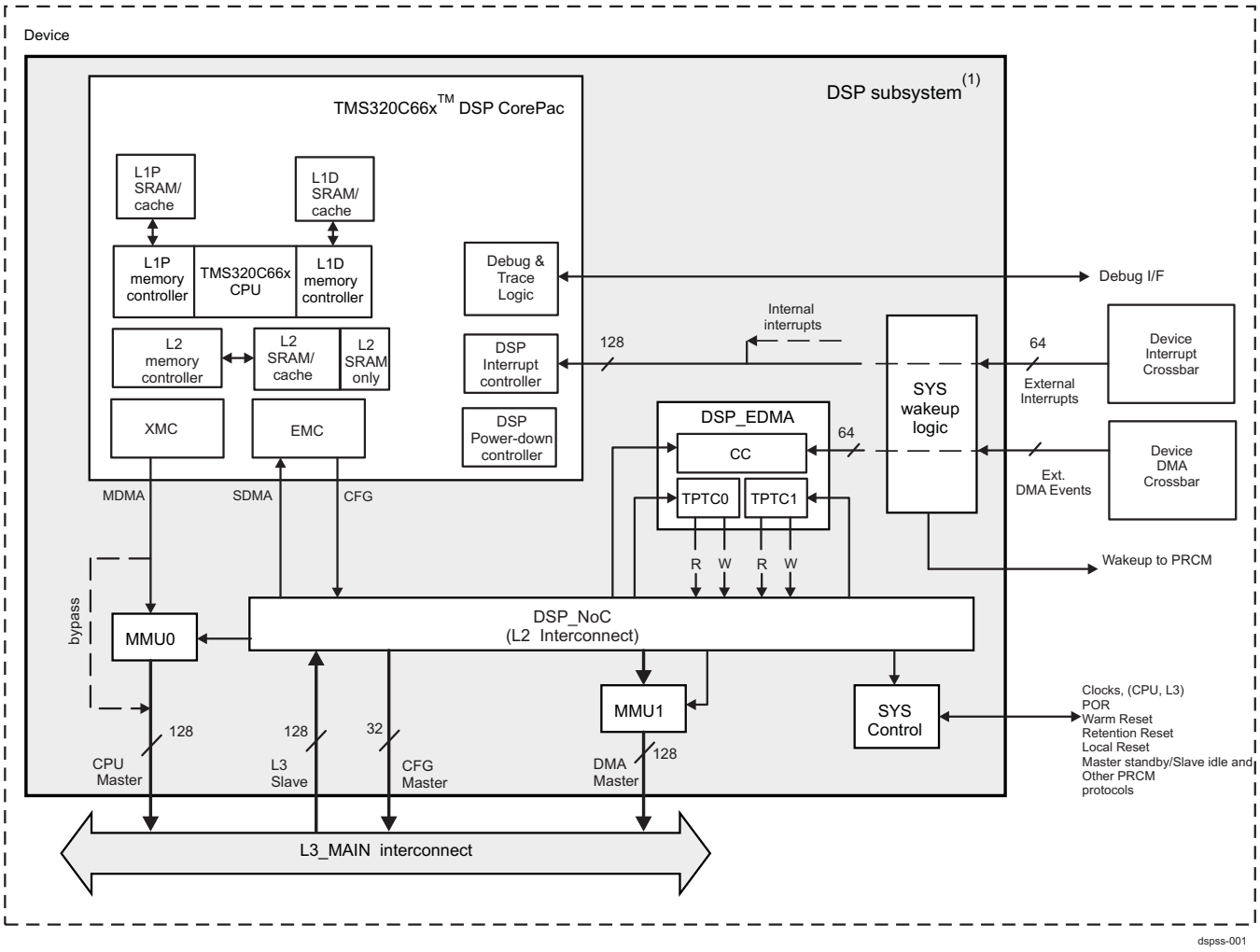
For more information on the TMS320C66x core CPU, see the *TMS320C66x DSP CPU and Instruction Set Reference Guide*, (SPRUGH7).

Each of the two DSP subsystems integrated in the device includes the following components:

- A TMS320C66x CorePac DSP core that encompasses :
  - L1 program-dedicated (L1P) cacheable memory
  - L1 data-dedicated (L1D) cacheable memory
  - L2 (program and data) cacheable memory
  - Extended Memory Controller (XMC)
  - External Memory Controller (EMC)
  - DSP CorePac located interrupt controller (INTC)
  - DSP CorePac located power-down controller (PDC)
- Dedicated enhanced data memory access engine - EDMA, to transfer data from/to memories and peripherals external to the DSP subsystems and to local DSP memory (most commonly L2 SRAM). The external DMA requests are passed through DSP system level (SYS) wakeup logic, and collected from the DSP1 / DSP2 dedicated outputs of the device DMA Events Crossbar for each of the two subsystems.
- A level 2 (L2) interconnect network (DSP NoC) to allow connectivity between different modules of the subsystem or the remainder of the device via the device L3_MAIN interconnect.
- Two memory management units (on EDMA L2 interconnect and DSP MDMA paths) for accessing the device L3_MAIN interconnect address space
- Dedicated system control logic (DSP_SYSTEM) responsible for power management, clock generation, and connection to the device power, reset, and clock management (PRCM) module

Figure 4-1 is the DSP subsystem top-level architecture.

Figure 4-1. DSP Subsystem Highlight



NOTE: (1) : This diagram shows a single DSP instance. Each device may have **one or two identical DSP instances**, refer to corresponding *device Data Manual* for the specific device support.

4.1.1 DSP Subsystems Key Features

The TMS320C66x Instruction Set Architecture (ISA) is the latest for the C6000 family. As with its predecessors (C64x, C64x+ and C674x), the C66x is an advanced VLIW architecture with 8 functional units (two multiplier units and six arithmetic logic units) that operate in parallel. The C66x CPU has a total of 64 general-purpose 32-bit registers.

Some features of the DSP C6000 family devices are :

- Advanced VLIW CPU with eight functional units (two multipliers and six ALUs) which:
  - Executes up to eight instructions per cycle for up to ten times the performance of typical DSPs
  - Allows designers to develop highly effective RISC-like code for fast development time
- Instruction packing
  - Gives code size equivalence for eight instructions executed serially or in parallel
  - Reduces code size, program fetches, and power consumption
- Conditional execution of most instructions

  - Reduces costly branching
  - Increases parallelism for higher sustained performance
- Efficient code execution on independent functional units
  - Industry's most efficient C compiler on DSP benchmark suite
  - Industry's first assembly optimizer for fast development and improved parallelization
- 8-/16-/32-bit/64-bit data support, providing efficient memory support for a variety of applications
- 40-bit arithmetic options which add extra precision for vocoders and other computationally intensive applications
- Saturation and normalization to provide support for key arithmetic operations
- Field manipulation and instruction extract, set, clear, and bit counting support common operation found in control and data manipulation applications.

The C66x CPU has the following additional features :

- Each multiplier can perform two 16 × 16-bit or four 8 × 8 bit multiplies every clock cycle.
- Quad 8-bit and dual 16-bit instruction set extensions with data flow support
- Support for non-aligned 32-bit (word) and 64-bit (double word) memory accesses
- Special communication-specific instructions have been added to address common operations in error-correcting codes.
- Bit count and rotate hardware extends support for bit-level algorithms.
- Compact instructions: Common instructions (AND, ADD, LD, MPY) have 16-bit versions to reduce code size.
- Protected mode operation: A two-level system of privileged program execution to support higher-capability operating systems and system features such as memory protection.
- Exceptions support for error detection and program redirection to provide robust code execution
- Hardware support for modulo loop operation to reduce code size and allow interrupts during fully-pipelined code
- Each multiplier can perform 32 × 32 bit multiplies
- Additional instructions to support complex multiplies allowing up to eight 16-bit multiply/add/subtracts per clock cycle

The TMS320C66x has the following key improvements to the ISA :

- 4x Multiply Accumulate improvement for both fixed and floating point
- Improvement of the floating point arithmetic
- Enhancement of the vector processing capability for fixed and floating point
- Addition of domain-specific instructions for complex arithmetic and matrix operations

On the C66x ISA, the vector processing capability is improved by extending the width of the SIMD instructions. The C674x DSP supports 2-way SIMD operations for 16-bit data and 4-way SIMD operations for 8-bit data. C66x enhances this capabilities with the addition of SIMD instructions for 32-bit data allowing operation on 128-bit vectors. For example the QMPY32 instruction is able to perform the element to element multiplication between two vectors of four 32-bit data each.

C66x ISA includes a set of specific instructions to handle complex arithmetic and matrix operations.

- **TMS320C66x DSP CorePac memory components :**
  - A 32-KiB L1 program memory (L1P) configurable as cache and / or SRAM:
    - When configured as a cache, the L1P is a 1-way set-associative cache with a 32-byte cache line
    - The DSP CorePac L1P memory controller provides bandwidth management, memory protection, and power-down functions
    - The L1P is capable of cache block and global coherence operations
    - The L1P controller has an Error Detection (ED) mechanism, including necessary SRAM
    - The L1P memory can be fully configured as a cache or SRAM

        - Page size for L1P memory is 2KB
    - A 32-KiB L1 data memory (L1D) configurable as cache and / or SRAM:
        - When configured as a cache, the L1D is a 2-way set-associative cache with a 64-byte cache line
        - The DSP CorePac L1D memory controller provides bandwidth management, memory protection, and power-down functions
        - The L1D memory can be fully configured as a cache or SRAM
        - No support for error correction or detection
        - Page size for L1D memory is 2KB
    - A 288-KiB (program and data) L2 memory, only part of which is cacheable:
        - When configured as a cache, the L2 memory is a 4-way set associative cache with a 128-byte cache line
        - Only 256 KiB of L2 memory can be configured as cache or SRAM
        - 32 KiB of the L2 memory is always mapped as SRAM
        - The L2 memory controller has an Error Correction Code (ECC) and ED mechanism, including necessary SRAM
        - The L2 memory controller supports hardware prefetching and also provides bandwidth management, memory protection, and power-down functions.
        - Page size for L2 memory is 16KB
- The **External Memory Controller (EMC)** is a bridge from the C66x CorePac to the rest of the DSP subsystem and device. It has :
    - a 32-bit configuration port (CFG) providing access to local subsystem resources (like DSP_EDMA, DSP_SYSTEM, etc) or to L3_MAIN resources accessible via the CFG address range.
    - a 128-bit slave-DMA port (SDMA) which provides accesses of system masters outside the DSP subsystem to resources inside the DSP subsystem or C66x DSP CorePac memories, i.e. when the DSP subsystem is the slave in a transaction.
- The Extended Memory Controller (XMC) processes requests from the L2 Cache Controller (which are a result of CPU instruction fetches, load/store commands, cache operations) to device resources via the C66x DSP CorePac 128-bit master DMA (MDMA) port :
    - Memory protection for addresses outside C66x DSP CorePac generated over device L3_MAIN on the MDMA port
    - Prefetch, multi-in-flight requests
- A DSP local **Interrupt Controller (INTC)** in the DSP C66x CorePac, interfaces the system events to the DSP C66x core CPU interrupt and exceptions inputs. Each DSP subsystem C66x CorePac interrupt controller supports up to 128 system events of which 64 interrupts are external to DSP subsystems, collected from the DSP1 /DSP2 dedicated outputs of the device Interrupt Crossbar.
- **Local Enhanced Direct Memory Access (EDMA) controller feaures:**
    - Channel controller (CC) : 64-channel, 128 PaRAM, 2 Queues
    - 2 x Third-party Transfer Controllers (TPTC0 and TPTC1):
        - Each TC has a 128-bit read port and a 128-bit write port
        - 2KiB FIFOs on each TPTC
    - 1-dimensional/2-dimensional (1D/2D) addressing
    - Chaining capability
- **DSP subsystems integrated MMUs**:
    - Two MMUs are integrated:
        - The MMU0 is located between DSP MDMA master port and the device L3_MAIN interconnect and can be optionally bypassed
        - The MMU1 is located between the EDMA master port and the device L3_MAIN interconnect
- A DSP local **Power-Down Controller (PDC)** is responsible to power-down various parts of the DSP C66x CorePac, or the entire DSP C66x CorePac.

- The DSP subsystems **System Control logic** provides:
  - Slave idle and master standby protocols with device PRCM for powerdown
  - OCP Disconnect handshake for init and target busses
  - Asynchronous reset
  - Power-down modes :
    - "Clockstop" mode featuring wake-up on interrupt event. The DMA event wake-up is managed in software.
- The device DSP subsystems are supplied by a PRCM DPLL, but each DSP1/2 **has integrated its own PLL module** outside the C66x CorePac for clock gating and division.
- **Each of the two device DSP subsystems has following port instances** to connect to remaining part of the device. See also Figure 4-1:
  - A 128-bit initiator (DSP MDMA master) port for MDMA/Cache requests
  - A 128-bit initiator (DSP EDMA master) port for EDMA requests
  - A 32-bit initiator (DSP CFG master) port for configuration requests
  - A 128-bit target (DSP slave) port for requests to DSP memories and various peripherals
- **C66x DSP subsystems (DSPSS) safety aspects** :
  - Above mentioned memory ECC/ED mechanisms
  - MMUs enable mapping of only the necessary application space to the processor
  - Memory Protection Units internal to the DSPSS (in L1P,L1D and L2 memory controllers) and external to DSPSS (firewalls) to help define legal accesses and raise exceptions on illegal accesses
  - Exceptions: Memory errors, various DSP errors, MMU errors and some system errors are detected and cause exceptions. The exceptions could be handled by the DSP or by a designated safety processor at the chip level. Note that it may not be possible for the safety processor to completely handle some exceptions

Unsupported features on the C66x DSP core for the device are:

- The Extended Memory Controller MPAX (memory protection and address extension) 36-bit addressing is NOT supported

Known DSP subsystem powermode restrictions for the device are:

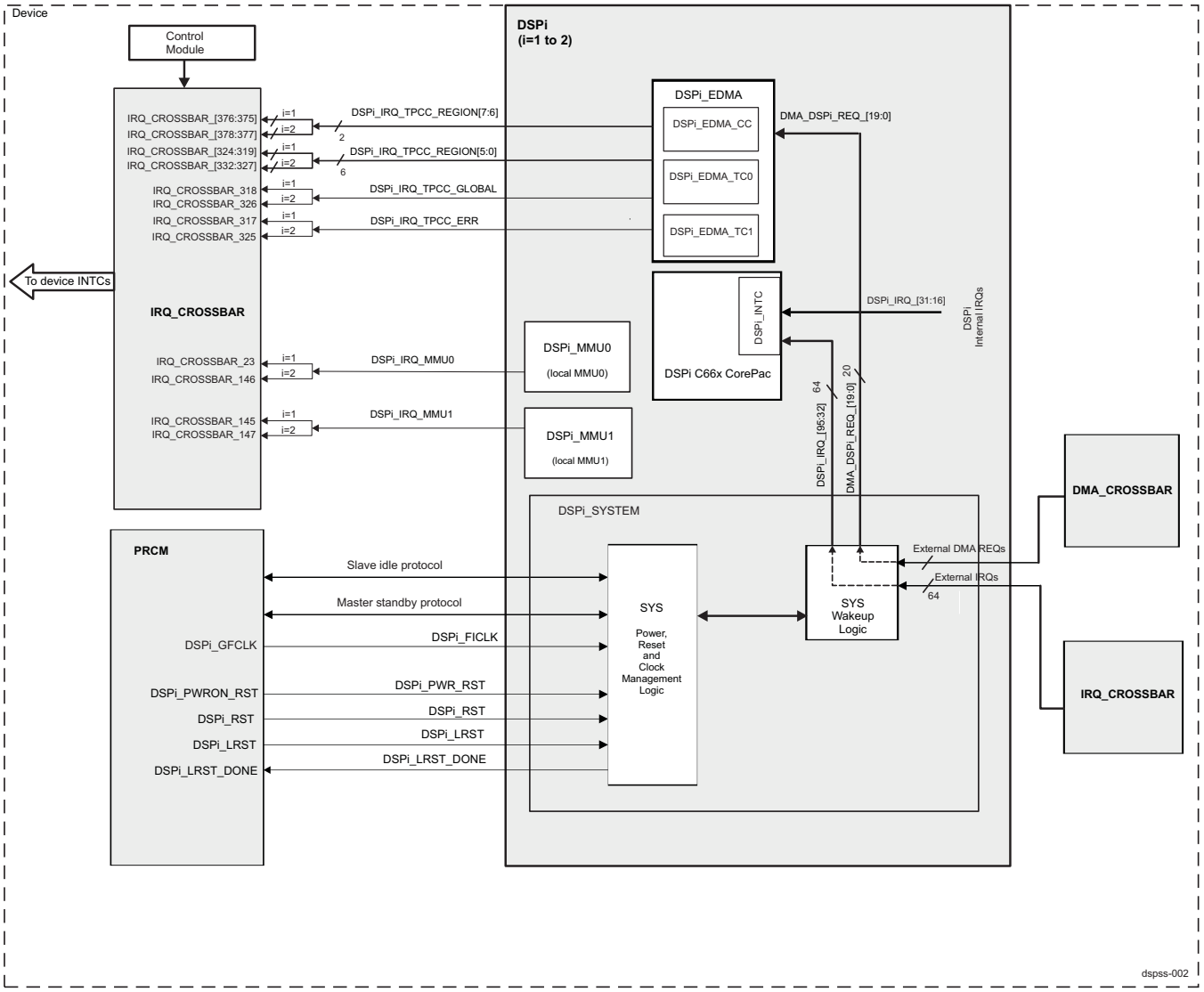
- "Full logic / RAM retention" mode featuring wake-up on both interrupt or DMA event (logic in “always on” domain). Only OFF mode is supported by DSP subsystem, **requiring full boot.**

4.2 DSP Subsystem Integration

This section describes the integration of the module in the device, including information about clocks, resets, and hardware requests.

Figure 4-2 shows the integration of the DSP subsystem.

Figure 4-2. DSP Subsystem Integration



NOTE: For more information about the slave idle protocol and the wake-up request, see Section 3.1.1.1.2, *Module-Level Clock Management*, in Chapter 3, *Power, Reset, and Clock Management*.

Table 4-1 through Table 4-3 summarize the integration of the module in the device.

Table 4-1. DSP Integration Attributes

Module Instance	Attributes	
	Power Domain	Interconnect
DSP1	PD_DSP1	L3_MAIN
DSP2	PD_DSP2	L3_MAIN

Table 4-2. DSP Clocks and Resets

Clocks				
Module Instance	**Destination Signal Name**	**Source Signal Name**	**Source**	**Description**
DSP1	DSP1_FICLK	DSP1_GFCLK	PRCM module	DSP1 subsystem gateable interface and functional clock.
DSP2	DSP2_FICLK	DSP2_GFCLK	PRCM module	DSP2 subsystem gateable interface and functional clock.
Resets				
Module Instance	**Destination Signal Name**	**Source Signal Name**	**Source**	**Description**
DSP1	DSP1_PWR_RST	DSP1_PWRON_RST	PRCM module	For information about PRCM reset sources and distribution, see Section 3.7.2, *PD_DSP1 Description* in Chapter 3, *Power, Reset, and Clock Management*. For DSP1 local reset details see also the Section 4.3.3.2.
	DSP1_RST	DSP1_RST	PRCM module	
	DSP1_LRST	DSP1_LRST	PRCM module	
	DSP1_LRST_DONE[(1)]	DSP1_LRST_DONE	DSP1	
DSP2	DSP2_PWR_RST	DSP2_PWRON_RST	PRCM module	For information about PRCM reset sources and distribution, see Section 3.7.3, *PD_DSP2 Description* in Chapter 3, *Power, Reset, and Clock Management*. For DSP2 local reset details see also the Section 4.3.3.2.
	DSP2_RST	DSP2_RST	PRCM module	
	DSP2_LRST	DSP2_LRST	PRCM module	
	DSP2_LRST_DONE[(1)]	DSP2_LRST_DONE	DSP2	

(1) Destination of this local reset monitoring signal is the PRCM

NOTE: For information about PRCM clock gating and management, see Section 3.7.2, *PD_DSP1 Description* and Section 3.7.3, *PD_DSP2 Description* in Chapter 3, *Power, Reset, and Clock Management*.

The DSP1 / DSP2 generates a number of interrupt requests (IRQs) mapped via the device IRQ_CROSBBAR to other device interrupt controllers (outside DSP subsystem). They are described in Table 4-3.

Table 4-3. DSP Hardware Requests

Interrupt Requests				
Module Instance	**Source Signal Name**	**Destination IRQ_CROSSBAR Input**	**Default Mapping**	**Description**
DSP1	DSP1_IRQ_MMU0	IRQ_CROSSBAR_23	DSP1_IRQ_54 DSP2_IRQ_54	Interrupt from the DSP1 subsystem local MMU0 (DSP1_MMU0CFG).
	DSP1_IRQ_MMU1	IRQ_CROSSBAR_145	-	DSP1 subsystem local MMU1 (DSP1_MMUCFG1) interrupt. This IRQ source signal is not mapped by default to any device INTC.
	DSP1_IRQ_TPCC_ERR	IRQ_CROSSBAR_317	-	DSP1 subsystem aggregated ("OR-ed") error interrupt. This IRQ source signal is not mapped by default to any device INTC.
	DSP1_IRQ_TPCC_GLOBAL	IRQ_CROSSBAR_318	-	DSP1 subsystem EDMA channel controller global interrupt. This IRQ source signal is not mapped by default to any device INTC.
	DSP1_IRQ_TPCC_REGION0	IRQ_CROSSBAR_319	-	DSP1 subsystem EDMA channel controller REGION0 interrupt. This IRQ source signal is not mapped by default to any device INTC.
	DSP1_IRQ_TPCC_REGION1	IRQ_CROSSBAR_320	-	DSP1 subsystem EDMA channel controller REGION1 interrupt. This IRQ source signal is not mapped by default to any device INTC.
	DSP1_IRQ_TPCC_REGION2	IRQ_CROSSBAR_321	-	DSP1 subsystem EDMA channel controller REGION2 interrupt. This IRQ source signal is not mapped by default to any device INTC.

Table 4-3. DSP Hardware Requests (continued)

	DSP1_IRQ_TPCC_REGION3	IRQ_CROSSBAR_322	-	DSP1 subsystem EDMA channel controller REGION3 interrupt. This IRQ source signal is not mapped by default to any device INTC.
	DSP1_IRQ_TPCC_REGION4	IRQ_CROSSBAR_323	-	DSP1 subsystem EDMA channel controller REGION4 interrupt. This IRQ source signal is not mapped by default to any device INTC.
	DSP1_IRQ_TPCC_REGION5	IRQ_CROSSBAR_324	-	DSP1 subsystem EDMA channel controller REGION5 interrupt. This IRQ source signal is not mapped by default to any device INTC.
	DSP1_IRQ_TPCC_REGION6	IRQ_CROSSBAR_375	-	DSP1 subsystem EDMA channel controller REGION6 interrupt. This IRQ source signal is not mapped by default to any device INTC.
	DSP1_IRQ_TPCC_REGION7	IRQ_CROSSBAR_376	-	DSP1 subsystem EDMA channel controller REGION7 interrupt. This IRQ source signal is not mapped by default to any device INTC.
DSP2	DSP2_IRQ_MMU0	IRQ_CROSSBAR_146	-	DSP2 subsystem local MMU0 (DSP2_MMU0CFG) interrupt. This IRQ source signal is not mapped by default to any device INTC.
	DSP2_IRQ_MMU1	IRQ_CROSSBAR_147	-	DSP2 subsystem local MMU1 (DSP2_MMUCFG1) interrupt. This IRQ source signal is not mapped by default to any device INTC.
	DSP2_IRQ_TPCC_ERR	IRQ_CROSSBAR_325	-	DSP2 subsystem aggregated ("OR-ed") error interrupt. This IRQ source signal is not mapped by default to any device INTC.
	DSP2_IRQ_TPCC_GLOBAL	IRQ_CROSSBAR_326	-	DSP2 subsystem EDMA channel controller global interrupt. This IRQ source signal is not mapped by default to any device INTC.
	DSP2_IRQ_TPCC_REGION0	IRQ_CROSSBAR_327	-	DSP2 subsystem EDMA channel controller REGION0 interrupt. This IRQ source signal is not mapped by default to any device INTC.

Table 4-3. DSP Hardware Requests (continued)

DSP2_IRQ_TPCC_REGION1	IRQ_CROSSBAR_328	-	DSP2 subsystem EDMA channel controller REGION1 interrupt. This IRQ source signal is not mapped by default to any device INTC.
DSP2_IRQ_TPCC_REGION2	IRQ_CROSSBAR_329	-	DSP2 subsystem EDMA channel controller REGION2 interrupt. This IRQ source signal is not mapped by default to any device INTC.
DSP2_IRQ_TPCC_REGION3	IRQ_CROSSBAR_330	-	DSP2 subsystem EDMA channel controller REGION3 interrupt. This IRQ source signal is not mapped by default to any device INTC.
DSP2_IRQ_TPCC_REGION4	IRQ_CROSSBAR_331	-	DSP2 subsystem EDMA channel controller REGION4 interrupt. This IRQ source signal is not mapped by default to any device INTC.
DSP2_IRQ_TPCC_REGION5	IRQ_CROSSBAR_332	-	DSP2 subsystem EDMA channel controller REGION5 interrupt. This IRQ source signal is not mapped by default to any device INTC.
DSP2_IRQ_TPCC_REGION6	IRQ_CROSSBAR_377	-	DSP2 subsystem EDMA channel controller REGION6 interrupt. This IRQ source signal is not mapped by default to any device INTC.
DSP2_IRQ_TPCC_REGION7	IRQ_CROSSBAR_378	-	DSP2 subsystem EDMA channel controller REGION7 interrupt. This IRQ source signal is not mapped by default to any device INTC.

NOTE: The DSP1 / DSP2 does NOT generate any DMA requests towards other device DMA controllers outside DSP1 / DSP2 (device-level EDMA, etc.).

NOTE: The "**Default Mapping**" column in Table 4-3, *DSP Hardware Requests* shows the default mapping of module IRQ source signals. These IRQ source signals can also be mapped to other lines of each device Interrupt controller through the IRQ_CROSSBAR module.
For more information about the IRQ_CROSSBAR module, see Section 13.4.6.4, *IRQ_CROSSBAR Module Functional Description*, in Chapter 13, *Control Module*.
For more information about the device interrupt controllers, see Chapter 12, *Interrupt Controllers*.

NOTE:

- For a description of the interrupt source controls at DSP subsystem level, see Section 4.3.4, *DSP Interrupt Requests*.

DSP1/DSP2 subsystem external interrupt sources: The default interrupt sources mapped by the device IRQ_CROSSBAR to the DSP1 / DSP2 interrupt controller lines are described in the Chapter 12, *Interrupt Controllers*. The programmable muxing of various external interrupt sources to the DSP1_INTC.DSP1_IRQ_x and DSP2_INTC.DSP2_IRQ_x input lines (where x=32 to 95) is covered in the Section 13.4.6.4, *IRQ_CROSSBAR Module Functional Description*, of the Chapter 13, *Control Module*.

DSP1/DSP2 subsystem internal interrupt sources: The mapping of DSP subsystem internal IRQ sources to DSP1_IRQ_x / DSP2_IRQ_x lines (where x=0 to 31 and x=96 to 127 for DSP subsystem internal event sources) is described in the Section 4.3.4, *DSP Interrupt Requests*.

DSP1/DSP2 subsystem external DMA request sources: The Table 4-6 and Table 4-7 lists the default DSP1 and DSP2 external DMA request sources, respectively, routed via the device DMA_CROSSBAR to the DSP1_EDMA / DSP2_EDMA channel controller inputs (DMA_DSP1_DREQ_i / DMA_DSP2_DREQ_i).

4.3 DSP Subsystems Functional Description

4.3.1 DSP Subsystems Block Diagram

Each of the device two DSP subsystems - DSP1 and DSP2 is composed of a DSP C66x CorePac coupled with several other submodules that enable its integration in the device architecture. Device DSP subsystem provides :

- a 128-bit master data port (MDMA) on device L3_MAIN with a dedicated DSP subsystem local MMU (MMU0) on the path.
- a 32-bit master configuration port (CFG) on device L3_MAIN through which DSP host configures various device located peripherals (external to the DSP subsystem).
- a 128-bit slave DMA port (SDMA) on device L3_MAIN which allows external initiators (masters) to DSP to manipulate some portion of its config / status registers (those which are mapped in the L3_MAIN space) in the device
- a 128-bit master EDMA port - which allows the DSP_EDMA traffic controllers to initiate transfers on L3_MAIN.

The C66x DSP subsystem is illustrated in the Figure 4-1.

4.3.2 DSP Subsystem Components

4.3.2.1 C66x DSP Subsystem Introduction

The key component of the C66x DSP subsystem is built on the TI's high performance TMS320C66x DSP CorePac which consists of a single TMS320C66x CPU (DSP_C0) processor along with a level 1 (L1P and L1D) cacheable SRAM and a level 2 (L2) cacheable SRAM memories interfaced via associated local L1P, L1D and L2 memory controllers, respectively. A DSP C66x CorePac includes also some other internal peripheral components, see Section 4.3.2.2.3 for details.

This chapter provides an overview of the DSP subsystem and the following considerations associated with it :

- DSP C66x CorePac Core and L1 / L2 Memories
- DSP System control and configuration :
  - clock management
  - wake-up event generation
  - interrupt masking
- DSP Booting
- DSP subsystem internal memory and external memory (L3_MAIN) space views
- DSP INTC interrupts mapping, event combining and exception generation
- DMA requests mapping to EDMA channels and EDMA traffic routing
- Others

For more information on the TMS320C66x DSP CorePac, refer to the *TMS320C66x DSP CorePac User Guide*, (SPRUGW0C), the *TMS320C66x DSP Cache User Guide*, (SPRUGY8) and the *TMS320C66x DSP CPU and Instruction Set Reference Guide*, (SPRUGH7).

4.3.2.2 DSP TMS320C66x CorePac

The TMS320C66x DSP CorePac is illustrated on Figure 4-1. It consists of a single DSP C66x CPU (DSP_C0) processor tightly coupled with level 1 - L1P (program), L1D (data) cacheable SRAM memories and level 2 (L2) cacheable SRAM memories. The C66x CorePac integrated memories are interfaced via associated local L1P, L1D and L2 memory controllers, respectively.

Additionally, the DSP C66x CorePac contains the following internal peripherals:

- an interrupt controller (DSP_INTC) to service DSP C66x CorePac internal and external interrupt events
- a power-down controller (DSP_PDC)
- an external memory controller - DSP_EMC
- an extended memory controller - DSP_XMC_CTRL
- a bandwidth manager - BWM with local controls to the L1P-, L1D- and L2-memories
- an internal direct memory access controller - IDMA

The C66x CorePac DSP also instantiates Debug and Trace logic, part of which is implemented in the DSP core C66x CPU. For more details, refer to the Chapter 26, *On-Chip Debug Support* .

4.3.2.2.1 DSP TMS320C66x CorePac CPU

The DSP C66x CorePac CPU includes following key components :

- A program fetch unit
- 16-/32-bit instruction dispatch unit, advanced instruction packing
- Instruction decode unit
- Two data paths, each with four functional units
- 64 x 32-bit general purpose registers
- Control logic and associated registers
- An internal interrupt and exception controller
- Test, emulation logic
- Internal DMA (IDMA) for transfers between internal memories

For more information on the TMS320C66x central processing unit, see the *TMS320C66x DSP CPU and Instruction Set Reference Guide*, (SPRUGH7).

4.3.2.2.2 DSP TMS320C66x CorePac Internal Memory Controllers and Memories

The TMS320C66x DSP CorePac implements a two-level internal cache-based memory architecture.

4.3.2.2.2.1 Level 1 Memories

Level 1 memory (L1) is split into separate program memory (L1P memory) and data memory (L1D memory). Each of the memories can be split into static RAM (normal addressable on-chip memory) and cache.

L1P memory is dedicated to TMS320C66x CPU program words storage and is interfaced via a dedicated L1P memory controller in the DSP C66x CorePac. User can choose to initialize one part of the L1P memory as cache and the other as SRAM. The entire 32 KiB memory is cacheable. The L1P features a dynamically configurable cache size (4 KiB, 8 KiB, 16 KiB and 32 KiB) defined via L1P controller configuration register - L1PCFG[2:0] L1PMODE bitfield. **Note that, the L1P controller maps the cache space by starting at the top of the L1P memory map (i.e. from most significant address) and working downwards. The L1P mapped SRAM size is 32 KiB minus the configured cache size.**

NOTE: The L1P cache / SRAM is ONLY read-accessible by the C66x CPU processor. The DSP C66x CorePac external DMAs (SDMA and EDMA) and internal DMA (IDMA) are the only initiators which can write to the L1P memory. The CPU may however write access and modify certain L1P cache/SRAM controller registers if such access is allowed for the register.

NOTE: In the device integrated DSP, at reset, the entire 32 KiB- L1P memory is initialized as a cache (reset value of L1PMODE=0x7).

For more information on the L1P cache/SRAM memories, refer to the *TMS320C66x DSP Cache User Guide*, (SPRUGY8).

L1D memory is used for level 1 CPU data storage and is interfaced via a dedicated L1D memory controller in the DSP C66x CorePac. User can choose to initialize one part of the L1D memory as cache and the other as SRAM. The entire 32 KiB memory is cacheable. The L1D features a dynamically configurable cache size (4 KiB, 8 KiB, 16 KiB and 32 KiB) defined via L1D configuration register - L1DCFG[2:0] L1DMODE bitfield. **Note that, the L1D controller maps the cache space by starting at the top of the L1D memory map (i.e. from most significant address) and working downwards. The L1D mapped SRAM size is 32 KiB minus the configured cache size.**

NOTE: In the device integrated DSP, at reset, the entire 32 KiB- L1D memory is initialized as a cache (i.e. reset value of L1DMODE = 0x7).

For more information on the L1D cache/SRAM memories, refer to the *TMS320C66x DSP Cache User Guide*, (SPRUGY8).

4.3.2.2.2 Level 2 Memory

The L2 memory can also be split into L2 RAM (normal addressable on-chip memory) and L2-cache for caching external to the DSP meagmodule memory locations. The on-chip integrated L2 memory total size is 288 KiB. The L2 memory is shared between data and program word sources within and outside the DSP C66x CorePac. The L2 memory is divided into two physical 128 bit-wide banks, accesses to which are interleaved on address LSB. Each of the two L2 banks is further split into 4 subbanks.

NOTE: Only 256 KiB of the L2 memory are cacheable in the device DSP. The remaining 32 KiB are always mapped as static RAM.

The L2 memory features a dynamically configurable cache size (32 KiB, 64 KiB, 128 KiB and 256 KiB) defined via L2 configuration register - L2CFG[2:0] L2MODE bitfield. The additional (to the 32KiB fixed SRAM L2) SRAM available is 256-KiB minus the cache size.

L2 memory controller is responsible for MDMA bus error events reporting. The DSP C66x CorePac MDMA bus error event is exported outside the C66x DSP CorePac in the subsystem, and can be enabled to trigger the ERRINT_IRQ aggregated interrupt output. See also corresponding **"MDMAERREVT"** event in the Table 4-5.

NOTE: The MDMA bus error event is not exported outside DSP subsystem. However it is merged (OR-ed) along with other error event sources within the DSP subsystem to produce a single ERRINT_IRQ interrupt exported outside the DSP subsystem.

For more details on ERRINT_IRQ generation and asscoiated event registers at DSP_SYSTEM level, refer to the Section 4.3.4.2.2.

4.3.2.2.3 DSP C66x CorePac Internal Peripherals

The DSP C66x CorePac includes the following internal peripherals:

- DSP interrupt controller (DSP_INTC)
- DSP power-down controller (DSP_PDC)
- Bandwidth manager (DSP_BWM)
- Memory Protection Hardware
- Internal DMA (DSP_IDMA) controller

- External Memory Controller (DSP_EMC)
- Extended Memory Controller (DSP_XMC_CTRL) including prefetch buffer logic
- Error Detection logic for the L1P memory
- Error Detection and Correction (ECC) logic for the L2 memory

This section briefly describes the DSP_INTC, DSP_PDC, DSP_BWM, DSP_IDMA, DSP_EMC, DSP_XMC_CTRL controller, L1P Error detection and L2 ECC logic. For more information on the TMS320C66x DSP CorePac, see the *TMS320C66x DSP CorePac User Guide*, (SPRUGW0C).

4.3.2.2.3.1 DSP C66x CorePac Interrupt Controller (DSP INTC)

The DSP C66x CorePac includes an interrupt controller (DSP_INTC) and can receive a total of 128 system events as inputs. They include DSP-generated events and chip-level events.

In addition to these 128 events, a non-maskable (NMI) event (see the Section 4.3.4.1.1) and reset events are mapped to the DSP_INTC as well, and are routed straight through to the DSP CPU core.

For more details on the DSP_INTC functionalities and its corresponding control / status registers (part of the DSP_ICFG local configuration space), refer to the section *Interrupt Controller* within the *TMS320C66x DSP CorePac User Guide*, (SPRUGW0C).

For more details on input interrupt mappings and associated IRQ wake-up events, refer to the Section 4.3.4.1.

For more information about the device DSP_INTC, see Chapter 12, *Interrupt Controllers*. For more information on chip level IRQ mapping via the device IRQ_CROSSBAR module, see Section 13.4.6.4, *IRQ_CROSSBAR Module Functional Description*, in Chapter 13, *Control Module*.

4.3.2.2.3.2 DSP C66x CorePac Power-Down Controller (DSP PDC)

The DSP C66x CorePac includes a power-down controller (PDC). The PDC can power-down all of the following components of the DSP C66x CorePac and internal memories of the DSP subsystem:

- C66x CPU
- L1P Memory
- L2 Memory
- Cache controllers
- Entire TMS320C66x DSP CorePac

Refer to the *Power-Down Controller* in the *TMS320C66x DSP CorePac User Guide*, (SPRUGW0C) for detailed descriptions of the DSP C66x CorePac components power-down control.

4.3.2.2.3.3 DSP C66x CorePac Bandwidth Manager (BWM)

The DSP C66x CorePac implements a bandwidth manager (BWM) to assure that some requestors do NOT block resources in the C66x CorePac DSP for extended periods of time.

Refer to the *Bandiwdth Management Architecture* in the*TMS320C66x DSP CorePac User Guide*, (SPRUGW0C) for detailed descriptions of the DSP C66x CorePac components power-down control.

4.3.2.2.3.4 DSP C66x CorePac Memory Protection Hardware

The C66x Core Pac memory protection architecture introduces in the DSP a combination of DSP privilege levels and a memory system permission structure. This provides several benefits to the system, as follows:

The DSP C66x CorePac MP events are exported outside the DSP C66x CorePac in DSP subsystem, and can be enabled to trigger the ERRINT_IRQ aggregated interrupt output. See also corresponding **memory protection fault** events listed in the Table 4-5.

NOTE: The memory protection exception events are not exported outside DSP subsystem. However they are merged (OR-ed) along with other error event sources within the DSP subsystem to produce a single ERRINT_IRQ interrupt exported outside the DSP subsystem.

For more details on ERRINT_IRQ generation and associated event registers at DSP_SYSTEM level, refer to the Section 4.3.4.2.2.

NOTE: IDMA, DMA or System initiators should not issue read/write requests to regions of DSP L1P, L1D, or L2 memory configured as cache. In such cases the corresponding MPPA register should be set to 0x0 to disallow external read/write accesses.

Refer to the section *Memory Protection* in the *TMS320C66x DSP CorePac User Guide*, (SPRUGW0C) for detailed descriptions of the DSP C66x CorePac components power-down control.

4.3.2.2.3.5 DSP C66x CorePac Internal DMA (IDMA) Controller

The IDMA controller performs fast block transfers between any two memory locations local to the DSP C66x CorePac. Local memory locations are defined as those in Level 1 program (L1P), Level 1 data (L1D), and Level 2 (L2) memories, or in the external peripheral configuration (CFG) port.

The IDMA configuration / status registers themselves are part of the DSP_ICFG and are visible only to C66x CPU.

NOTE: **The IDMA cannot transfer data to or from the internal DSP memory-mapped register space (DSP_ICFG).**

The IDMA exception is mapped to the system level ERRINT_IRQ interrupt event. For more details, refer to the Section 4.3.4.2.2 and the Table 4-5.

The DSP C66x CorePac IDMA error event is exported outside the DSP C66x CorePac in the subsystem, and can be enabled to trigger the ERRINT_IRQ aggregated interrupt output. See also corresponding **"EMC_IDMAERR"** event in the Table 4-5.

NOTE: The IDMA exception error event is not exported outside DSP subsystem. However it is merged (OR-ed) along with other error event sources within the DSP subsystem to produce a single ERRINT_IRQ interrupt exported outside the DSP subsystem.

For more details on ERRINT_IRQ generation and asscoiated event registers at DSP_SYSTEM level, refer to the Section 4.3.4.2.2.

The IDMA is fully described in the section *Internal Direct Memory Access (IDMA) Controller* of the *TMS320C66x DSP CorePac User Guide*, (SPRUGW0C).

4.3.2.2.3.6 DSP C66x CorePac External Memory Controller

The DSP C66x CorePac has an embedded External Memory Controller which acts as a bridge between the DSP C66x CorePac CPU and the remaining part of DSP subsystem. It implements two ports interfacing the DSP C66x CorePac environment:

- An external peripheral 32-bit CFG port which acts as an (DSP CPU cfg) initiator on the DSP_NoC L2 interconnect. It is the root source of all C66x CPU external configuration traffic **(excluding the DSP_ICFG traffic which takes place only within the DSP C66x CorePac)** towards the subsystem.
- An SDMA slave port which is a target on the L2 DSP_NoC interconnect. It generally accepts traffic initiated on DSP_NoC by the:
  - DSP_EDMA_TC0 and DSP_EDMA_TC1
  - DSP external slave port on the L3_MAIN

In summary:

- The CPU CFG port provides access to the memory-mapped registers which control various peripherals

and resources within the DSP subsystem, such as the MMUs, DSP_EDMA controllers, DSP system control and wakeup logic, DSP NoC itself, DSP external peripherals, etc.

- The DSP system masters found outside the DSP C66x CorePac such as the DSP_EDMA controllers (TC0 and TC1) or L3_MAIN masters (IPU1, etc.) access the SDMA slave port to reach resources inside the DSP C66x CorePac. In respect to the SDMA port, the DSP C66x CorePac is the slave in the transaction.

The DSP_EMC controller adds following functionalities to the DSP C66x CorePac:

- Reporting errors related to the C66x CPU external peripheral configuration bus (associated registers)

NOTE: Regarding PrivID versus AID mapping functionality of the EMC, the C66x DSP CorePac is able to distinguish ONLY "local" vs "external" requests. The device integrated C66x DSP CorePac has the SDMA PrivID input tied-off to a value of 0x0. On DSP C66x CorePac SDMA port, this means that no distinguishing can be made between external requests which come over DSP_NoC interconnect from local DSP_EDMA and those coming from other initiators accesing DSP via the device L3_MAIN interconnect.

This limits the funcitionality of the internal memory protection registers.

The DSP C66x CorePac EMC error event is exported outside the DSP C66x CorePac in the subsystem, and is capable to trigger the ERRINT_IRQ aggregated interrupt output. See also corresponding **EMC_BUSERR** event in the Table 4-5.

NOTE: The EMC configuration bus error event is not exported outside DSP subsystem. However it is merged (OR-ed) along with other error event sources within the DSP subsystem to produce a single ERRINT_IRQ interrupt exported outside the DSP subsystem.

For more details on ERRINT_IRQ generation and asscoiated event registers at DSP_SYSTEM level, refer to the Section 4.3.4.2.2.

For various DSP_NoC initiator vs target mappings, refer to the Table 4-8.

Note that, there are DSP_NoC pressure (Mflag bus) controls in DSP_SYSTEM logic related to the C66x CPU CFG and DSP_NoC SDMA init traffic. They are described in the Section 4.3.8.3.

The EMC functionalities / registers are fully described in the section *External Memory Controller (EMC)* of the *TMS320C66x DSP CorePac User Guide*, (SPRUGW0C).

4.3.2.2.3.7 DSP C66x CorePac Extended Memory Controller

The DSP C66x CorePac located extended memory controller (DSP_XMC_CTRL) implements a local DMA master port (MDMA) which provides the primary path for C66x CPU and cache requests to the device level memories such as (DDR or L3 SRAM) and peripheral / memory mapped register space. Via some additional logic, including DSP_SYSTEM controls and a local DSP MMU - DSP_MMU0 on the path (with option to bypass), C66x local MDMA port is mapped to the DSP subsystem CPU master port (i.e. MDMA master port of the DSP CPU on L3_MAIN). The DSP C66x Corepac MDMA port is mapped to the DSP Subsystem CPU Master Port (with DSP_MMU0 involved or not involved) to allow fast accesses (DSP_NoC not involved) to the external SDRAM (via the L3_MAIN) or to L3 SRAM (via the L3_MAIN).

The memory protection settings in MPAX defines types of the memory accesses permitted on various address ranges within DSP C66x CorePac 32-bit address map.

The DSP_XMC_CTRL also instantiates program and data prefetch buffer logic to reduce time during servicing read requests from the L1D, L1P and L2 memory controllers. The aim is to buffer program and data fetches from external L3_MAIN memory locations. While the program prefetch buffer is organized as 4 entry x 32 byte, the data prefetch buffer is organized in 8 slots, with 128 bytes per slot. The DSP_XMC_CTRL prefetch reduces the penalty associated with accesses to the L3_MAIN SDRAM upon L1P, L1D and L2-cache read-misses.

NOTE: the DSP_XMC_CTRL registers are part of the DSP_ICFG space, hence they are not accessible outside the DSP C66x CorePac (visible only to the C66x CPU).

In summary the DSP_XMC_CTRL provides :

- a master path from the DSP C66x CorePac level 2 cache / SRAM memory to device memory such as SDRAM or L3 SRAM or peripheral / MMR address space.
- memory protection on external address ranges related to L3_MAIN RAMs (via MPAX unit) :
  - 16 user-defined address ranges (MPAX segments) can be used to divide external memory space
- Address translation
- Data and Program prefetch buffer logic
- 12- address candidate buffer that acts as a "stream detection filter"

NOTE: The device DSP subsystem does NOT use the DSP C66x CorePac multicore shared memory controller (MSMC) port to add more static RAM within subsystem boundaries, i.e. no additional SRAM is available in the DSP except for the L1P, L1D and L2 memories. Only the DSP_XMC_CTRL controller MDMA port on the L3_MAIN interconnect is used to extend DSP available RAM memory via a direct or DSP_MMU0 translated access to the device EMIF DDR memories and OCMC RAMs.

NOTE: **Only the memory protection function of the DSP_XMC_CTRL MPAX unit is intergated** and used in the device DSP subsystem. The MDMA port 32-bit to 36-bit address extension function is NOT used in the device DSP because L3_MAIN address bus width is 32-bit. The DSP_MMU0 does NOT perform an address size (32b -> 36b) extension as well.

The XMC functionalities / registers are fully described in the section *Extended Memory Controller (XMC)* of the *TMS320C66x DSP CorePac User Guide*, (SPRUGW0C).

4.3.2.2.3.7.1 XMC MDMA Accesses at DSP System Level

4.3.2.2.3.7.1.1 DSP System MPAX Logic

The default configuration of MPAX registers provides a 32-bit view of system memory on L3_MAIN.

In summary, each MPAX segment (mentioned above) is programmed with a starting virtual base address, segment sizes from 4 GiB down to 4 KiB, replacement address (i.e., physical address); and permission attributes. Provided that DSP_MMU0 can be used to perform address translation, in most cases the replacement address will equal the base address (i.e., virtual == physical from DSP C66x CorePac perespective).

The system level implementation of MPAX logic allows the C66x CPU to change permission without being required to flush the cache.

The C66x CPU subsystem relies on the MPAXn.PERM field to properly configure the permissions for remote address ranges. The MDMA.rperm[6:0] signals are tie-off to a fixed value of 0x7F on the DSP C66x CorePac boundary.

4.3.2.2.3.7.1.2 MDMA Non-Post Override Control

The C66x corepac submits writes denoted as either “cacheable” or non-cacheable. Write accesses that are non-cacheable will be submitted as interconnect (L3_MAIN) non-posted writes; whereas write accesses that are cacheable are submitted as interconnect posted writes. An exception for the cache writes to L3_MAIN is that in the case of a cache block write-back operation (when actual cache evict busrts are actually issued towards L3_MAIN connected memory), a non-posted write is submitted.

NOTE: In order to provide a safety net for interconnects that may do aggressive reordering, a memory-mapped register SW control is provided - DSP_SYS_BUS_CONFIG[24] NOPOSTOVERRIDE. When set, this results in all write commands being issued as non-posted. This bit defaults to set, and thus the default behavior is for non-posted writes to be used exclusively.

4.3.2.2.3.8 L1P Memory Error Detection Logic

The L1P memory detection logic (no correction is implemented) uses a 4-bit parity per 256-bit location (1-bit parity per 64-bit line quadrant).

The L1P error detection logic features:

- L1P error detection command, status and address controls (registers)
- L1P_ED error detection exception / interrupt to the DSP_INTC upon DMA / IDMA access
- a direct exception event to C66x CPU (DSP_INTC not involved) upon parity error during a program fetch from L1P-cache
- L1P-cache error recovery

The L1P parity error detection event is exported outside the DSP C66x CorePac in the subsystem, and can be enabled to trigger the ERRINT_IRQ aggregated interrupt output. See also corresponding **"PMC_ED"** event in the Table 4-5.

NOTE: The L1P error detection event is not exported outside DSP subsystem. However it is merged (OR-ed) along with other error event sources within the DSP subsystem to produce a single ERRINT_IRQ interrupt exported outside the DSP subsystem.

For more details on ERRINT_IRQ generation and asscoiated event registers at DSP_SYSTEM level, refer to the Section 4.3.4.2.2.

For more details on L1P error detection logic, refer to the section *L1P Error Detection*, of the *TMS320C66x DSP CorePac User Guide*, (SPRUGW0C).

4.3.2.2.3.9 L2 Memory Error Detection and Correction Logic

The L2 Memory error detection and correction logic (ECC) implements a distance-3 “detect 2, correct 1” Hamming code based error correction / detection algorithm. A 12-bit hamming code per 256-bit is used.

The L2 error detection and correction logic features:

- L2 error detection command, status and address controls (registers)
- L2 EDC enable
- L2 error detection event counter
- 2x L2 EDC exception / interrupts mapped to the DSP_INTC :
  - L2_ED1 = "error corrected" event
  - L2_ED2 = "error-not-corrected" event

The two L2 memory error correction events are exported outside the DSP C66x CorePac in the subsystem, and can be enabled to trigger the ERRINT_IRQ aggregated interrupt output. See also corresponding **"UMC_ED1" and "UMC_ED2"** events in the Table 4-5.

NOTE: The L2 error detection events are not exported outside DSP subsystem. However they are merged (OR-ed) along with other error event sources within the DSP subsystem to produce a single ERRINT_IRQ interrupt exported outside the DSP subsystem.

For more details on ERRINT_IRQ generation and asscoiated event registers at DSP_SYSTEM level, refer to the Section 4.3.4.2.2.

For more details on L2 error detection and correction logic, refer to the section *L2 Error Detection and Correction* of the*TMS320C66x DSP CorePac User Guide*, (SPRUGW0C).

4.3.2.3 DSP Debug and Trace Support

The DSP subsystem offers full support for the native DSP C66x CorePac debug features. This includes Advanced Event Triggering (AET) and Trace.

4.3.2.3.1 DSP Advanced Event Triggering (AET)

AET capability can be used to debug complex problems as well as understand performance characteristics of user applications. AET provides the following capabilities:

- Hardware Program Breakpoints: specify addresses or address ranges that can generate events such as halting the processor or triggering the trace capture.
- Data Watchpoints: specify data variable addresses, address ranges, or data values that can generate events such as halting the processor or triggering the trace capture.
- Counters: count the occurrence of an event or cycles for performance monitoring.
- State Sequencing: allows combinations of hardware program breakpoints and data watchpoints to precisely generate events for complex sequences.

4.3.2.3.2 DSP Trace Support

Trace is a debug technology that provides a detailed, historical account of application code execution, timing, and data accesses. Trace collects, compresses, and exports debug information for analysis. Trace works in real-time and does not impact the execution of the system. Trace is supported via Code Composer Studio.

See also the Chapter 26, *On-Chip Debug Support* .

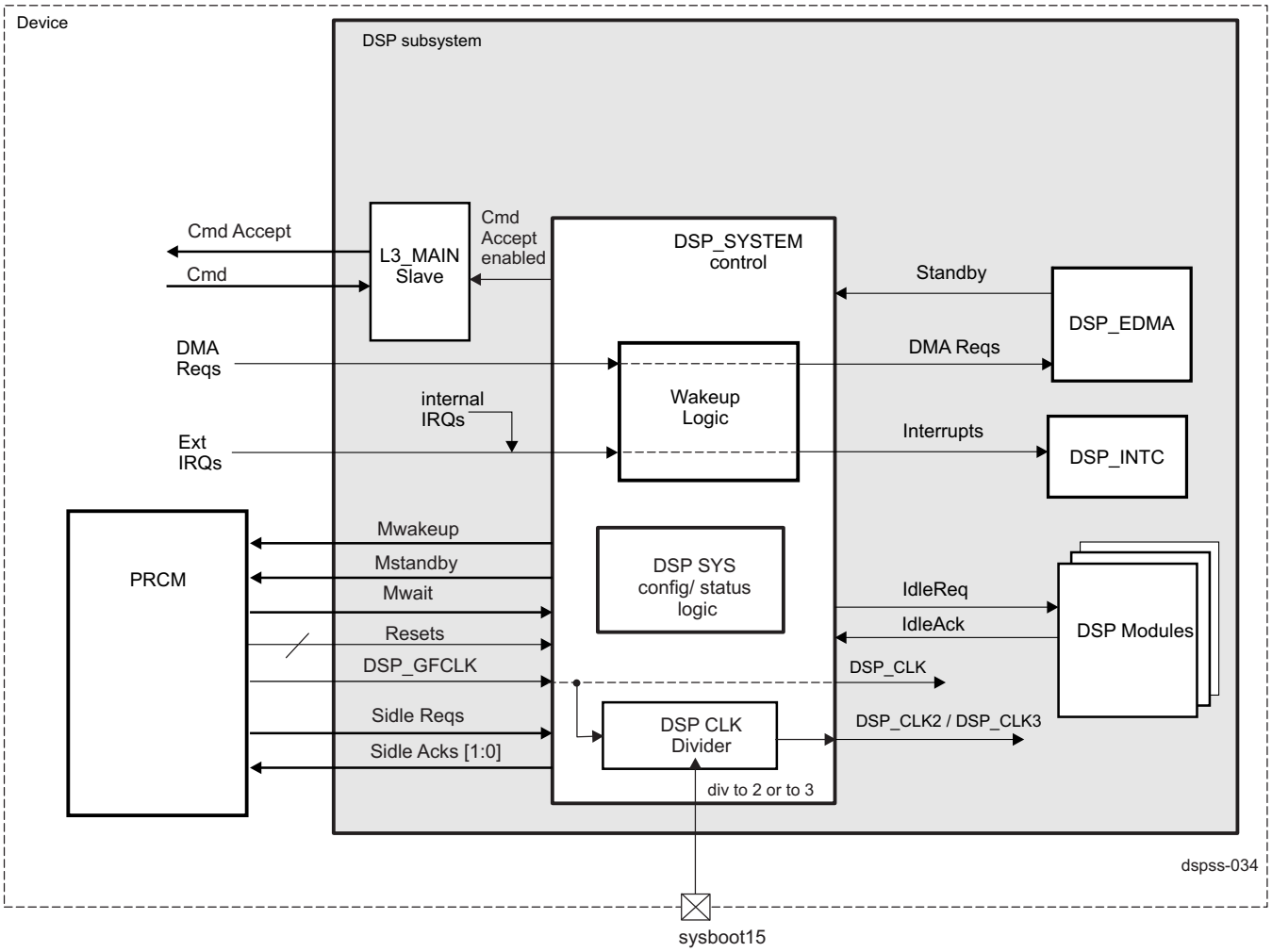
4.3.3 DSP System Control Logic

The DSP_SYSTEM module controls the following functions:

- Generation of the divided clocks (DSP_CLK2 or DSP_CLK3) to all components of the DSP subsystem
- Synchronization of the DSP divided clocks
- PRCM module power handshaking
- Reset input resynchronization of the active-to-inactive transition to the CD1_CLK clock
- DSP subsystem top level configuraiton registers and its access from the DSP core.

Figure 4-3 highlights the DSP_SYSTEM and its connectivities to the surrounding blocks within the subsystem and in the device.

Figure 4-3. DSP_SYSTEM Block Diagram



4.3.3.1 DSP System Clocks

The DSP1 and DSP2 subsystems inputs a primary non-divided clock (DSP1_FICLK / DSP2_FICLK) and based on it (DSP_CLK1), internally generates either **a divided by 2 clock** (DSP_CLK2) version or **a divided by 3 clock** (DSP_CLK3). The divided clock determines the operation rate of the DSP subsystem logic and bus interfaces. The division is defined upon device boot time through signal level externally applied on the device **sysboot15** input. The actual bit configuration is latched upon power-on reset in Control Module register CTRL_CORE_BOOTSTRAP[15] SYS_BOOT_15_CLOCK_DIVIDER boot status bit. For more details, refer to the Section 13.4.6.13.1, *System Boot Status Settings* of the chapter, *Control Module*.

NOTE: Only DSP_CLK3 clock is supported on this SoC. Upon boot time, sysboot15 set at '1' selects a DSP_CLK3 divided clock version for the DSP subsystem logic and bus interfaces. For proper device operation, sysboot15 must be tied to vdd.

The clock operating mode setting (DSP_CLK2 or DSP_CLK3) must be static just before and continually after reset deassertion. This signal will also drive the configuration to the DSP C66x CorePac for the XMC_MDMA_CLK, EMC_SDMA_CLK, and EMC_CFG_CLK configurations.

The DSPSS1 / DSPSS2 subsystem input clock frequency (DSP_CLK1) corresponds to the PRCM DSP1_GFCLK / DSP2_GFCLK frequency that is configured in the device PRCM registers.

NOTE: For valid DSP_CLK1 (and hence for DSP_CLK3 = DSP_CLK1 / 3) frequency range, see the Operating Performance Points section of the device Data Manual.

The Section 4.3.2 also shows the distribution of the different DSP subsystems blocks within the two DSP local clock domains CD0_CLK (running on DSP_CLK frequency) and CD1_CLK (running on DSP_CLK2 or DSP_CLK3 frequency).

4.3.3.2 DSP Hardware Resets

The DSP uses the same reset sources than those mapped to the DSP C66x CorePac; i.e. DSP C66x CorePac reset inputs will be pinned out as DSP system reset inputs.

The Table 4-4 summarizes the DSP hardware reset inputs and their functional descriptions.

Table 4-4. Summary of the DSP1 and DSP2 Hardware Resets

DSP1 reset input	DSP1 reset "done" output to PRCM	Description
DSP1_PWR_RST	-	This is power-on reset signal used inside DSP1 to reset mainly the emulation logic. It resets the entire DSP1 logic.
DSP1_RST	-	Reset signal used to reset all logic inside DSP1 except Emulation logic.
DSP1_LRST	-	Reset applied ONLY to the C66x CPU inside DSP1
-	DSP1_LRST_DONE	Indicates completion of the DSP1 local C66x CPU reset to device PRCM
DSP2 reset input	**DSP2 reset "done" output to PRCM**	**Description**
DSP2_PWR_RST	-	This is power-on reset signal used inside DSP2 to reset mainly the emulation logic. It resets the entire DSP2 logic.
DSP2_RST	-	Reset signal used to reset all logic inside DSP2 except Emulation logic.
DSP2_LRST	-	Reset applied ONLY to the C66x CPU inside DSP2
-	DSP2_LRST_DONE	Indicates completion of the DSP2 local C66x CPU reset to device PRCM

See also the Section 4.2 for more information on the PRCM reset sources to DSP reset inputs connectivity.

Refer to the Section 3.5.6.2, *DSP1 Subsystem Power-on Reset Sequence* and the Section 3.5.6.4, *DSP2 Subsystem Power-on Reset Sequence* in the chapter, *Power, Reset and Clock Manamgement* for more details on the DSP1 and DSP2 power-on reset sequence, respectively.

NOTE: In the case of DSP1 / DSP2 recovery from the “Powerdown-grid off”, a full power-on-reset sequence is required before re-booting and resuming functional operation.

The DSP host (IPU) software must ensure that the PRCM functional clock DSP1_GFCLK / DSP2_GFCLK is enabled to the DSP1 / DSP2, respectively, prior to starting the DSP1 / DSP2 power-on reset sequence.

4.3.3.3 DSP Software Resets

During a software reset on the DSP, all resets described in Table 4-4 are asserted, except for the power-on DSP_PWRON_RST signal which remains de-asserted in this case.

The DSP subsystem does NOT implement any local software reset controls. The software reset assertion and DSP_LRST completion monitoring is done in PRCM located registers (part of the DSP1_PRM / DSP2_PRM address space).

Refer to the Section 3.5.6.3, *DSP1 Subsystem Software Warm Reset Sequence* and the Section 3.5.6.5, *DSP2 Subsystem Software Warm Reset Sequence* in the chapter, *Power, Reset and Clock Manamgement* for more details on the DSP1 and DSP2 software reset sequence and related software controls, respectively.

4.3.3.4 DSP Power Management

The supported power-down modes are:

- Slave idle and master standby protocols for powerdown
- "Disconnect from interconnect" handshake for init and target busses
- Clock Stop mode - wakeup on interrupt or DMA event
- Grid OFF mode : No power supply is switched-on

NOTE: Powerdown-retention mode is NOT supported by DSP subsystem. **The DSP recovery from the Powerdown-grid OFF mode requires full boot.**

The DSP C66x CorePac natively supports “CLKSTOP/Static Powerdown” and “POWERDOWN (Grid off)” modes of operation. Once the device PRCM restores clocks and power supply, then the DSP C66x CorePac can exit static-powerdown.

The DSP_SYSTEM wakeup logic is implemented in the "always-on" clock / power supply domain. This logic monitors new interrupts / events and will drive the IDLE wakeup request when a new interrupt / event occurs.

4.3.3.4.1 DSP System Powerdown Protocols

For each of the powerdown modes – Static or Powerdown-Grid Off, the device PRCM will control whether clocks are gated, or whether supplies are reduced or removed.

NOTE: In the case of “Powerdown-grid off”, a full power-on-reset sequence is required before re-booting and resuming functional operation.

In the static powerdown modes - the DSP recognizes new level interrupts while in a clock-gated state (and drive wakeup request). During this powerdown mode all internal state will be retained, including DSP C66x CorePac, interconnect, EDMA, memories, etc.

The following protocols are implemented with PRCM:

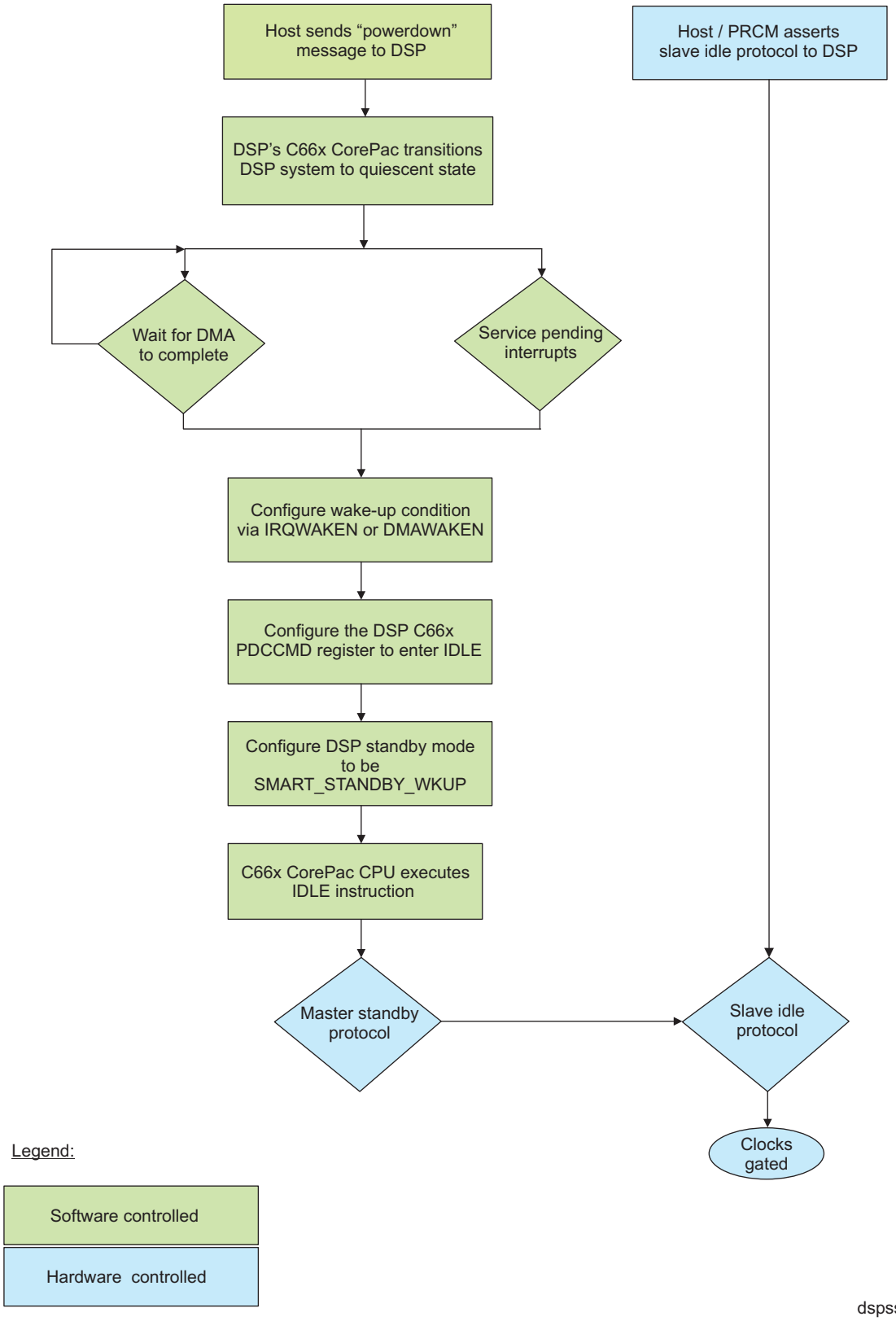
- Slave idle protocol with device PRCM for powerdown (wake-up capable)
- Master standby protocol with device PRCM for powerdown
- Interconnect disconnect for master and slave ports

The Master standby and slave idle protocols behaviour is controlled in the DSP_SYS_SYSCONFIG register.

4.3.3.4.2 DSP Software and Hardware Power Down Sequence Overview

Figure 4-4 highlights the high level flow-chart for entry into any of the DSP powerdown modes. The system host (typically) first informs the DSP that it should enter a powerdown mode. The host sends a software message (normally via system level mailbox+interrupt). In parallel, the PRCM (via host or DSP programmation) will hardware assert an SIdleReq request to the DSP via the IDLE Protocol connection. At the next stage, the C66x CPU, in general, performs any software bookkeeping necessary to transition the DSP subsystem to a quiescent state. This may include : waiting for outstanding DMA transfers to complete, waiting for outstanding DMA transfers to complete, etc. The C66x processor should finally execute the IDLE instruction when it is ready to be powered-down. Assuming the DSP_SYS_SYSCONFIG[5:4] STANDBYMODE is enabled, then the hardware will transition to an idle state and notify to the system the intention to enter powerdown state to the system via the master standby and slave idle protocols. After IDLE and MSTANDBY handshake is completed, the DSP clocks are optionally gated; and supply rails are optionally reduced or turned off.

Figure 4-4. Extended Duration Sleep Software and Hardware Sequence



NOTE: The PM_DSPx_PWRSTCTRL[1:0] POWERSTATE bit field in device PRCM must be set to 0x3 (ON state) prior to performing the sequence shown in Figure 4-4 for the transition to be successful.

4.3.3.4.3 DSP IDLE Wakeup

In order to facilitate auto-wakeup of DSP C66x, the IDLE protocol's wakeup capability is used. Wakeup operation is enabled if DSP_SYS_SYSCONFIG[3:2] IDLEMODE is set to 0x3.

In this mode, while in IDLE state, if an external input interrupt source is asserted (if enabled via the DSP_SYS_IRQWAKEEN0 / DSP_SYS_IRQWAKEEN1 mask) or if an external DMA event source is asserted (if enabled via the DSP_SYS_DMAWAKEEN0 / DSP_SYS_DMAWAKEEN1 mask) or if the DSP subsystem NMI input is asserted **(note that there is no wake enable mask for the non-maskable interrupt)** then the Mwakeup signal is asserted to the PRCM which is expected to observe the Mwakeup. Upon such assertion the PRCM enables the clocks, exiting the "Standby" and "Idle" states. at this point the C66x CPU is able to branch to the pending interrupt service routine. The MWakeup is deasserted when all IRQ or DMA requests enabled in the DSP_SYS_IRQWAKEEN0/1 and DSP_SYS_DMAWAKEEN0/1 are deasserted.

The Wakeup logic controlling assertion of the MWakeup request is completely asynchronous because in IDLE mode the clock may not be present. It relies on level sensitive interrupts.

NOTE: The DSP_EDMA must be manually removed from IDLE / Standby state. During that time, it is possible that the EDMA input event is no longer pending and may not have been recognized/latched as an EVENT to the EDMA. In that case, the user SW can enable the DSP EDMA_WAKE_INT (in associated DSP_SYS_EDMAWAKE0_IRQENABLE_SET register) to recognize in the (DSP_SYS_EDMAWAKE0_IRQSTATUS_RAW /DSP_SYS_EDMAWAKE0_IRQSTATUS) which specific EDMA event was asserted and caused the wakeup condition. The DSP software can then trigger the corresponding DSP_EDMA channel manually (by setting the ESR) or by servicing the interrupt/event manually via reads and writes. For more details, refer to the Section 4.3.5.1.

4.3.3.4.4 DSP SYSTEM IRQWAKEEN registers

The DSP_SYS_IRQWAKEEN0 / DSP_SYS_IRQWAKEEN1 masking bits must be appropriately set for any valid interrupt mapped from device IRQ_CROSSBAR to the DSP subsystem boundary, to enable its path (passing through the DSP_SYSTEM wakeup logic) to the DSP local interrupt controller - DSP_INTC.

CAUTION

In order for a given interrupt to be serviced by the DSP (even when the Idle Instruction is NOT being executed), the Interrupt must be enabled in the coresponding DSP_SYS_IRQWAKEEN0 or DSP_SYS_IRQWAKEEN1 register.

4.3.3.4.5 DSP Automatic Power Transition

This section provides register details for configuring the DSP1 subsystem in automatic power transition mode. The same should be considered for DSP2 in corresponding PRCM and DSP2 related registers.

The DSP1 module is supposed to be configured to automatic management in PRCM.DSP1_CM_CORE_AON via setting the register CM_DSP1_DSP1_CLKCTRL[1:0] MODULEMODE bitfield to 0x1. The DSP1 clock domain is supposed to be configured in automatic "HW_AUTO" transition (setting bitfield CM_DSP1_CLKSTCTRL[1:0] CLKTRCTRL=0x3).
The power state (controls are in the PRCM.DSP1_PRM instance) to reach upon a sleep transition is configured in the PM_DSP1_PWRSTCTRL[1:0]POWERSTATE bitfield.

4.3.4 DSP Interrupt Requests

The DSP subsystem relies on the DSP C66x CorePac local interrupt controller - DSP_INTC for mapping the various input interrupts to the C66x CPU, that are :

- generated outside the DSP, by the device intergated modules and subsystems
- generated within the DSP subsystems but outside the DSP C66x CorePac
- generated by different components within the DSP C66x CorePac

In addition, a non-maskable input interrupt, direct mapped on a C66x processor NMI input is implemented. It is mapped via a register that resides within the device Control Module. Both the maskable interrupts and the non-maskable interrupts are synchronized internally.

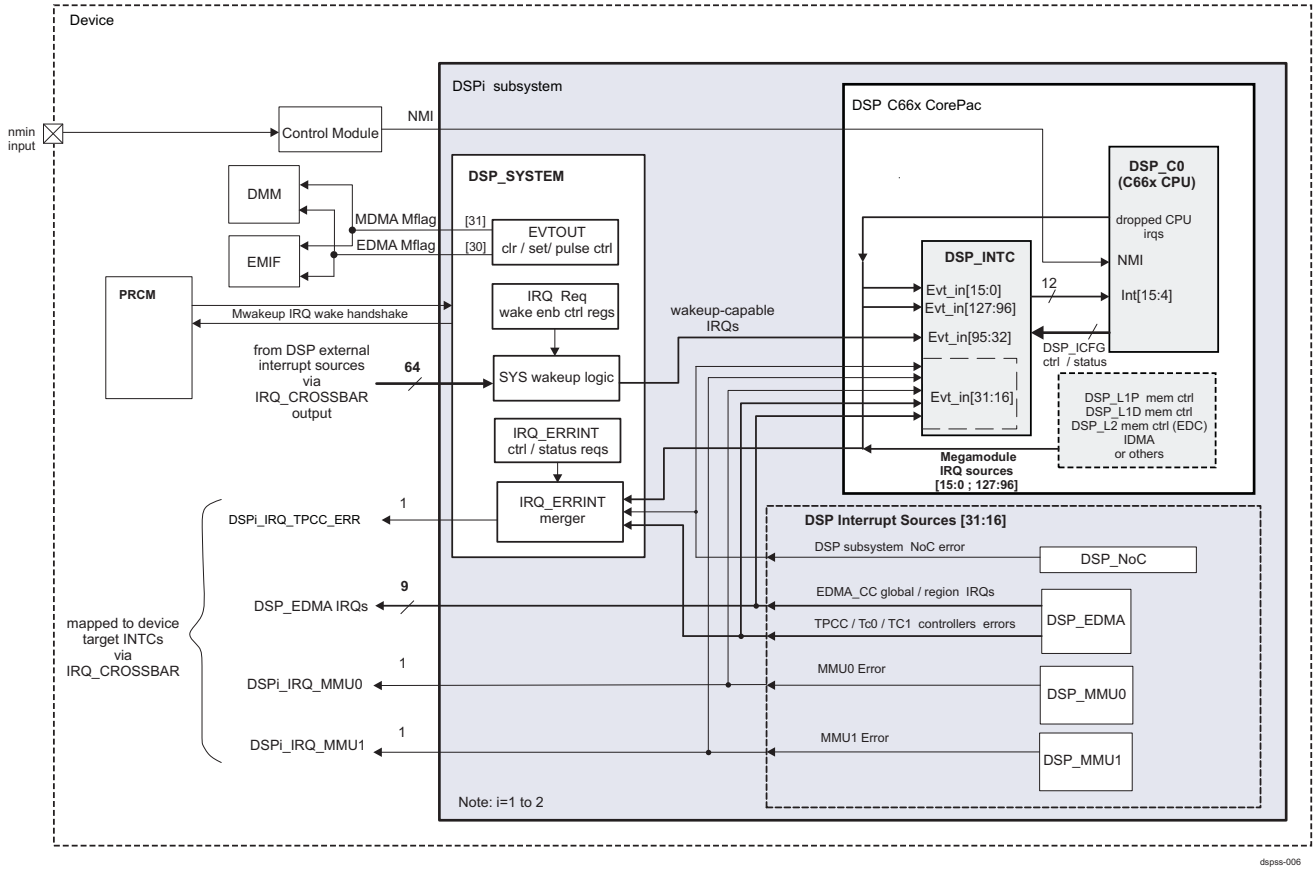
Part of the DSP subsystem module generated interrupts which are output as follows:

- DSP_EDMA interrupts
- DSP_MMU0 and DSP_MMU1 interrupts
- Error interrupts

Figure 4-5 shows how are the interrupt sources organized. To manage and expand the interrupt capabilities of the DSP C66x CorePac (internal and external interrupt requests), the DSP subsystem includes two levels of interrupt control :

- The DSP C66x CorePac local Interrupt controller - DSP_INTC
- The System control logic - DSP_SYSTEM

Figure 4-5. DSP Subsystem Interrupt Management



4.3.4.1 DSP Input Interrupts

In summary, the DSP_INTC accepts up to 124 event inputs, and flexibly maps those down to 12 interrupt inputs to the DSP. The mapping can be 1:1 (input:output), or can use the event combiner to map multiple interrupts (within a 32-bit group) to one of the DSP interrupt inputs. In general, many of the 124 interrupt controller inputs are collected within the DSP C66x CorePac, and are NOT available at the DSP C66x CorePac boundaries.

The C66x CPU dropped event is exported outside the C66x CorePac in DSP subsystem, and can be enabled to trigger the ERRINT_IRQ aggregated interrupt output. See also corresponding **"INTERR" event** listed in the Table 4-5.

NOTE: The dropped CPU event is not exported outside DSP subsystem. However it is merged (OR-ed) along with other error event sources within the DSP subsystem to produce a single ERRINT_IRQ interrupt exported outside the DSP subsystem.

For more details on ERRINT_IRQ generation and asscoiated event registers at DSP_SYSTEM level, refer to the Section 4.3.4.2.2.

Part of the input interrupts, generated by DSP peripherals that are located outside the DSP C66x CorePac - DSP_EDMA (DSP_EDMA_CC, DSP_EDMA_TC0, DSP_EDMA_TC1) , DSP_MMU0, DSP_MMU1 and DSP_NoC, are also mapped to outputs at the DSP subsystem boundary, such that they can be exported to system hosts via the device IRQ_CROSSBAR.

Of particular interest, MMUs (DSP_MMU0 and DSP_MMU1) interrupts will typically be serviced by the device host instead of by the local DSP core.

Any interrupt input at DSP subsystem boundaries (i.e. excluding the DSP subsystem internal IRQ sources that reside in and outside the DSP C66x CorePac) can be used to wake-up the DSP subsystem from an IDLE state. This is described in the Section 4.3.3.4.3 and is controlled by the DSP_SYSTEM logic register DSP_SYS_SYSCONFIG [3:2] IDLEMODE bitfield along with the DSP_SYS_IRQWAKEEN0 / DSP_SYS_IRQWAKEEN1 registers.

CAUTION

The DSP_SYS_IRQWAKEEN0 / DSP_SYS_IRQWAKEEN1 bits MUST be enabled for externally mapped interrupts (DSP_INTC[95:32]) to be serviced by the DSP regardless of the DSP power state (IDLE or non-IDLE).

The DSP C66x CorePac DSP_INTC registers are NOT readable by any entity other than the C66x CPU, because they are part of the DSP_ICFG C66x CorePac internal configuration space (see also the Section 4.3.10). Hereby, only the C66x itself is able to service these interrupt events. The only way for these interrupts to be cleared is for the DSP CPU to clear the state in the EVTFLAGi (where i=0 to 3) register, or via reset assertion.

NOTE: For cases where the DSP maps an interrupt directly, the DSP is not strictly required to clear the EVTFLAGi register. User software must take the extra step of clearing the EVTFLAGi to cause the corresponding output interrupt to be cleared and re-asserted upon a new input event assertion.

4.3.4.1.1 DSP Non-maskable Interrupt Input

The device DSP also supports a non-maskable interrupt (NMI) directly mapped to the NMI input of the C66x CPU. This line is also mapped to the NMEVT input of the DSP local INTC, and can be used as an exception signal, too. At system level, the NMI interrupt mapping to the DSP_INTC is controlled via the **device core Control Module** register as follows:

- CTRL_CORE_NMI_DESTINATION_2 [15:8] DSP1 = 0x1 enables the DSP1 to receive the NMI coming from the device **nmin** input.
- CTRL_CORE_NMI_DESTINATION_2 [23:26] DSP2 = 0x1 enables the DSP2 to receive the NMI

coming from the device **nmin** input.

For more details on the NMI receive enable bit mapping, refer to the Section 13.5, *Control Module Register Manual* in the chapter, *Control Module*.

4.3.4.2 DSP Event and Interrupt Generation Outputs

4.3.4.2.1 DSP MDMA and DSP EDMA Mflag Event Outputs

The Mflag events generated by DSP subsystem EVTOUT bus are represented in the Figure 4-5.

A couple of the DSP EVTOUT bus outputs - EVTOUT[31] and EVTOUT[30] are used for generation of MFLAGs dedicated to the DSP MDMA and EDMA ports, respectively. DSP MFLAGs are connected directly to DMM and EMIF. The DSP MFLAGs participate in the DMM Emergency and EMIF MFLAG prioritization schemes. At the L3 Level Bandwidth regulators connected to the DSP MDMA and EDMA ports can be used to control DSP traffic versus other device traffic.

The device DSP subsystem is able to generate the 2 output Mflag events via the following DSP_SYSTEM module located registers :

- DSP_SYS_EVTOUT_SET[31:30]
- DSP_SYS_EVTOUT_CLR[31:30]

The current state of the outputs can be detected by reading any of these "pseudo" register bits.

NOTE: Only EVTOUT[31:30] outputs are implemented in the device, hence only bits [31:30] of the mentioned DSP_SYS_EVTOUT_x registers are used.

The DSP_SYS_EVTOUT_SET register unconditionally drives the corresponding output event to '1'. The DSP_SYS_EVTOUT_CLR register unconditionally drives the corresponding output event to a '0'.

4.3.4.2.2 DSP Aggregated Error Interrupt Output

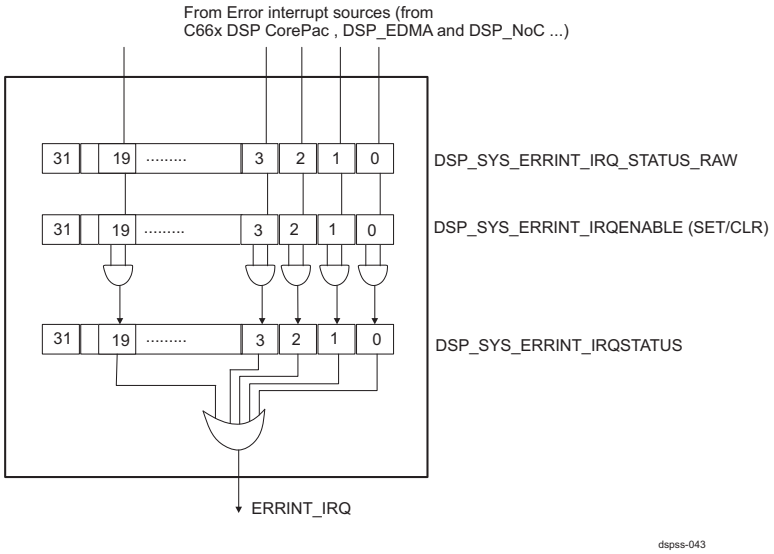
The aggregated error interrupt of the DSP subsystem is shown in the Figure 4-5.

The subset of those events that correspond to: **DSP C66x CorePac generated error events**, **DSP_EDMA error interrupts** and **L2 DSP_NoC interconnect error interrupt**, is reduced by an OR-schematic to a single ERRINT_IRQ output interrupt which is made available on DSP subsystem boundary. It is expected that one of the DSP system hosts monitors the interrupts/error conditions in safety conscious systems.

Figure 4-6 shows a functional representation of the DSP error interrupt "OR"-reduction logic. In summary, there exists **an unmasked status** (DSP_SYS_ERRINT_IRQSTATUS_RAW) register, two complementary enable bit-vector registers (DSP_SYS_ERRINT_IRQENABLE_SET / DSP_SYS_ERRINT_IRQENABLE_CLR), and a masked status register (DSP_SYS_ERRINT_IRQSTATUS). The ERRINT event is asserted when any enabled error interrupt input is asserted.

NOTE: The ERRINT_IRQ output can be programmatically mapped as the DSPi_IRQ_TPCC_ERR (where i=1 to 2) interrupt to all device (dsp hosts) interrupt controllers via the device IRQ_CROSSBAR. For more information on the IRQ_CROSSBAR module, see Section 13.4.6.4, *IRQ_CROSSBAR Module Functional Description*, in Chapter 13, *Control Module*.
For more information about the device interrupt controllers, see Chapter 12, *Interrupt Controllers*.

Figure 4-6. ERRINT Diagram



The Table 4-5 details the mapping of error event output sources to the bit positions within the following DSP error event related registers :

- DSP_SYS_ERRINT_IRQSTATUS_RAW
- DSP_SYS_ERRINT_IRQSTATUS
- DSP_SYS_ERRINT_IRQENABLE_SET
- DSP_SYS_ERRINT_IRQENABLE_CLR

Following functional descriptions are valid for the above registers :

- **IRQ status raw register -** This register provides a per-event raw interrupt status vector. The Raw status is set even if the corresponding event is not enabled. Software can write '1' to set the (raw) status for debug purposes.
- **IRQ status register -** This register provides a per-event enabled interrupt status vector. The Enabled status is set if the corresponding event is enabled and the raw status is set. Software can write 1 to clear the (raw) status after the interrupt has been serviced. The clear takes effect even if the interrupt is not enabled.
- **IRQ enable register -** This register provides a per-event interrupt enable bit vector. Software can write 1 to set (i.e., enable the corresponding interrupt). Reads of this register return the actual state of the enable register (and is the same as reading the corresponding "IRQ Clear" - register)
- **IRQ clear register -** This register provides a per-event interrupt enable bit vector. Software can write 1 to clear (i.e., disable the corresponding interrupt). Reads of this register return the actual state of the enable register (and is the same as reading the corresponding "IRQ Set" register)

NOTE: A DSP_SYS_ERRINT_IRQSTATUS_RAW bit is set even if the corresponding event is NOT enabled in the DSP_SYS_ERRINT_IRQENABLE_SET.

Table 4-5. DSP ERRINT Interrupt Mapping

Interrupt Number	Name	Description
0	tpcc_errint_level	DSP EDMA CC error interrupt
1	tptc_errint0_level	DSP EDMA TC0 error interrupt
2	tptc_errint1_level	DSP EDMA TC1 error interrupt
3	noc_errint_level	DSP L2 Interconnect (DSP_NoC) error interrupt

Table 4-5. DSP ERRINT Interrupt Mapping (continued)

Interrupt Number	Name	Description
4	INTERR	DSP C66x CorePac Dropped CPU Interrupt event
5	EMC_IDMAERR	DSP C66x CorePac Invalid IDMA Parameters
6	MDMAERREVT	DSP C66x CorePac VbusM Error Event
7	PMC_ED	DSP C66x CorePac Single bit error detected during DMA read
8	UMC_ED1	DSP C66x CorePac Corrected bit error detected
9	UMC_ED2	DSP C66x CorePac Uncorrected bit error detected
10	SYS_CMPA	DSP C66x CorePac CPU memory protection fault
11	PMC_CMPA	DSP C66x CorePac CPU memory protection fault
12	PMC_DMPA	DSP C66x CorePac DMA memory protection fault
13	DMC_CMPA	DSP C66x CorePac CPU memory protection fault
14	DMC_DMPA	DSP C66x CorePac DMA memory protection fault
15	UMC_CMPA	DSP C66x CorePac CPU memory protection fault
16	UMC_DMPA	DSP C66x CorePac DMA memory protection fault
17	EMC_CMPA	DSP C66x CorePac CPU memory protection fault
18	EMC_BUSERR	DSP C66x CorePac Bus Error Interrupt
19	Reserved	-
20	Reserved	-
21	Reserved	-
22	Reserved	-

Note that neither of the events, listed in Table 4-5 , is exported as a separate hardware interrupt off the DSP boundary.

4.3.4.2.3 Non-DSP C66x CorePac Generated Peripheral Interrupt Outputs

The non-DSP C66x CorePac interrupts generated by peripherals within the DSP subsystem are also summarized in the Figure 4-5.

Besides the aggregated error event ERRINT_IRQ interrupts – DSPi_IRQ_TPCC_ERR (where i=1 to 2), interrupts (see also Figure 4-5) **generated individually by DSPSS peripherals located outside the DSP C66x CorePac**, are mapped as separate IRQ outputs at DSP boundaries. They are sourced by the DSP_EDMA_CC, DSP_EDMA_TC0, DSP_EDMA_TC1, DSP_MMU0, DSP_MMU1 and DSP_NoC and exported to other host INTCs via the device IRQ_CROSSBAR. Refer to the Section 4.2, for more information on these DSP interrupt outputs mapping.

4.3.5 DSP DMA Requests

The DSP_EDMA_CC (channel controller) supports 64 hardware event inputs, that can be used to synchronize the 64 DMA channels. These event inputs are provided at the DSP subsystem boundary via the device DMA_CROSSBAR and can be mapped to sources within the device.

The DSP subsystem receives DMA requests from certain peripherals, such as the McASP module. The DMA requests path through the DSP logic is shown in Figure 4-7.

Similar to the interrupts received at DSP subsystem boundary, the DSP EDMA requests are first routed through the wakeup generation logic of the DSP_SYSTEM module, hence, each DMA request received by the DSP subsystem can wakeup the system from DSP low power modes (including wakeup from DSP OFF mode). To enable the DMA requests mapped via the DMA_CROSSBAR to **DSP_EDMA_CC [19:0]** inputs, corresponding bits in range [19:0] of the register DSP_SYS_DMAWAKEEN0 must be enabled in software.

CAUTION

The DMA request corresponding DSP_SYS_DMAWAKEEN0 / DSP_SYS_DMAWAKEEN1 MUST be enabled, for the DMA requests to be serviced by the DSP regardless of the DSP being in IDLE or active state.

Figure 4-7. DSP DMA Requests

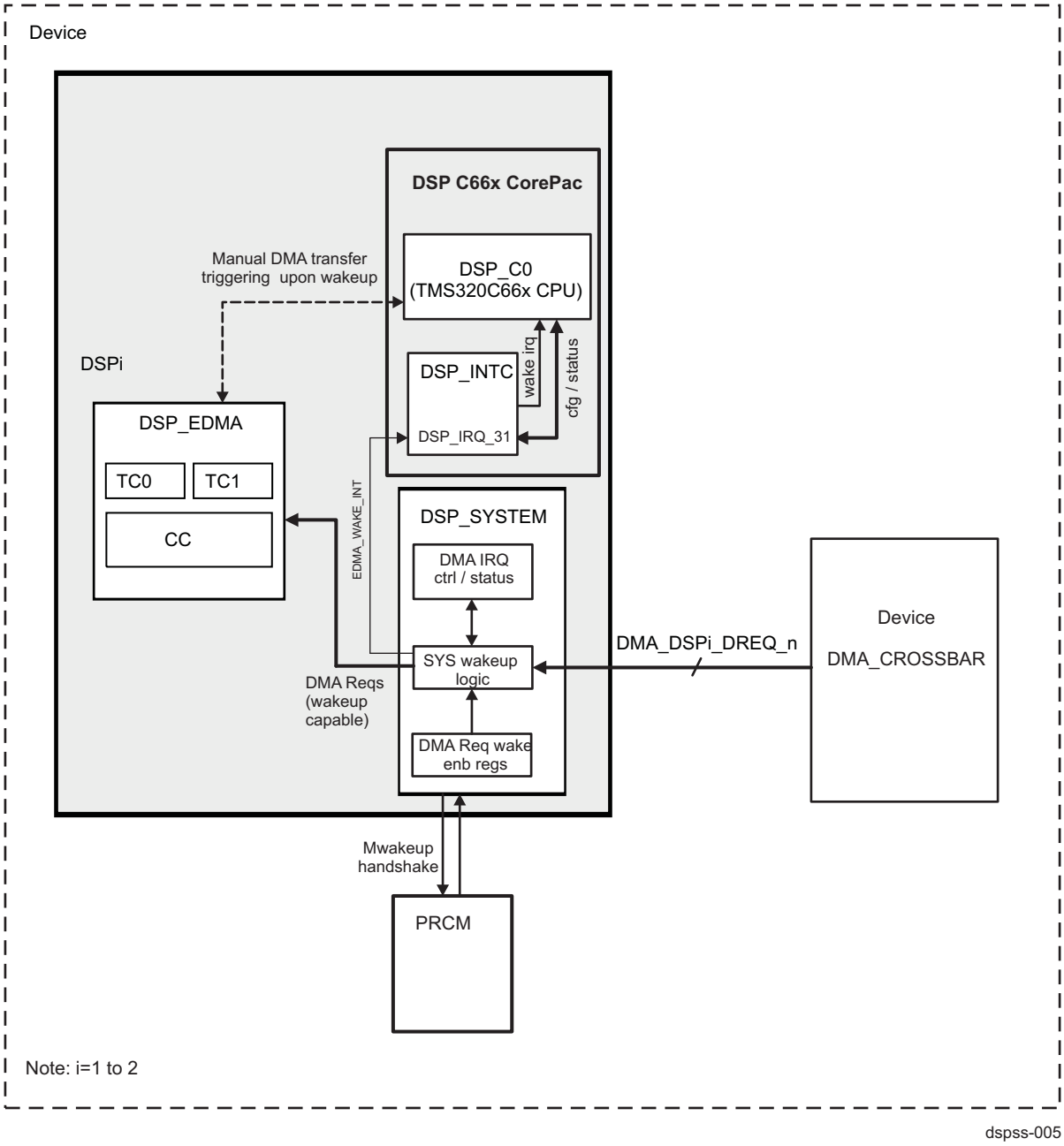


Table 4-6 and Table 4-7 list the default DMA sources for the DSP1_EDMA and DSP2_EDMA controllers. In addition, DSP1_EDMA / DSP2_EDMA inputs (DMA_DSP1_DREQ_[19:0] / DMA_DSP2_DREQ_[19:0]) can alternatively be sourced through the associated DMA_CROSSBAR from one of the 256 multiplexed device DMA sources listed in . The CTRL_CORE_DMA_DSP1_DREQ_y_z / CTRL_CORE_DMA_DSP2_DREQ_y_z registers (where *y* and *z* are indexes of DSP1_EDMA / DSP2_EDMA input lines) in the Control Module are used to select between the default DMA sources and the multiplexed DMA sources.

For more details on the device DMA_CROSSBAR multiplexing registers structure, refer to Section 13.4.6.5, *DMA_CROSSBAR Module Functional Description* of chapter, *Control Module*.

Table 4-6. DSP1_EDMA Default Request Mapping

DMA Request Line	DMA_CROSSBAR Instance Number	DMA_CROSSBAR Configuration Register	DMA_CROSSBAR Default Input Index	Default DMA Source Name	Default DMA Source Description
DMA_DSP1_DREQ_0	1	CTRL_CORE_DMA_DSP1_DREQ_0_1[7:0]	128	McASP1_DREQ_RX	McASP1 receive event
DMA_DSP1_DREQ_1	2	CTRL_CORE_DMA_DSP1_DREQ_0_1[23:16]	129	McASP1_DREQ_TX	McASP1 transmit event
DMA_DSP1_DREQ_2	3	CTRL_CORE_DMA_DSP1_DREQ_2_3[7:0]	130	Reserved	Reserved
DMA_DSP1_DREQ_3	4	CTRL_CORE_DMA_DSP1_DREQ_2_3[23:16]	131	Reserved	Reserved
DMA_DSP1_DREQ_4	5	CTRL_CORE_DMA_DSP1_DREQ_4_5[7:0]	132	Reserved	Reserved
DMA_DSP1_DREQ_5	6	CTRL_CORE_DMA_DSP1_DREQ_4_5[23:16]	133	Reserved	Reserved
DMA_DSP1_DREQ_6	7	CTRL_CORE_DMA_DSP1_DREQ_6_7[7:0]	134	Reserved	Reserved
DMA_DSP1_DREQ_7	8	CTRL_CORE_DMA_DSP1_DREQ_6_7[23:16]	135	Reserved	Reserved
DMA_DSP1_DREQ_8	9	CTRL_CORE_DMA_DSP1_DREQ_8_9[7:0]	136	Reserved	Reserved
DMA_DSP1_DREQ_9	10	CTRL_CORE_DMA_DSP1_DREQ_8_9[23:16]	137	Reserved	Reserved
DMA_DSP1_DREQ_10	11	CTRL_CORE_DMA_DSP1_DREQ_10_11[7:0]	138	Reserved	Reserved
DMA_DSP1_DREQ_11	12	CTRL_CORE_DMA_DSP1_DREQ_10_11[23:16]	139	Reserved	Reserved
DMA_DSP1_DREQ_12	13	CTRL_CORE_DMA_DSP1_DREQ_12_13[7:0]	140	Reserved	Reserved
DMA_DSP1_DREQ_13	14	CTRL_CORE_DMA_DSP1_DREQ_12_13[23:16]	141	Reserved	Reserved
DMA_DSP1_DREQ_14	15	CTRL_CORE_DMA_DSP1_DREQ_14_15[7:0]	142	Reserved	Reserved
DMA_DSP1_DREQ_15	16	CTRL_CORE_DMA_DSP1_DREQ_14_15[23:16]	143	Reserved	Reserved
DMA_DSP1_DREQ_16	17	CTRL_CORE_DMA_DSP1_DREQ_16_17[7:0]	154	Reserved	Reserved
DMA_DSP1_DREQ_17	18	CTRL_CORE_DMA_DSP1_DREQ_16_17[23:16]	155	Reserved	Reserved
DMA_DSP1_DREQ_18	19	CTRL_CORE_DMA_DSP1_DREQ_18_19[7:0]	156	Reserved	Reserved
DMA_DSP1_DREQ_19	20	CTRL_CORE_DMA_DSP1_DREQ_18_19[23:16]	157	Reserved	Reserved
DMA_DSP1_DREQ_20 - DMA_DSP1_DREQ_63	N/A	N/A	N/A	Reserved	Reserved

Table 4-7. DSP2_EDMA Default Request Mapping

DMA Request Line	DMA_CROSSBAR Instance Number	DMA_CROSSBAR Configuration Register	DMA_CROSSBAR Default Input Index	Default DMA Source Name	Default DMA Source Description
DMA_DSP2_DREQ_0	1	CTRL_CORE_DMA_DSP2_DREQ_0_1[7:0]	128	McASP1_DREQ_RX	McASP1 receive event
DMA_DSP2_DREQ_1	2	CTRL_CORE_DMA_DSP2_DREQ_0_1[23:16]	129	McASP1_DREQ_TX	McASP1 transmit event
DMA_DSP2_DREQ_2	3	CTRL_CORE_DMA_DSP2_DREQ_2_3[7:0]	130	Reserved	Reserved
DMA_DSP2_DREQ_3	4	CTRL_CORE_DMA_DSP2_DREQ_2_3[23:16]	131	Reserved	Reserved
DMA_DSP2_DREQ_4	5	CTRL_CORE_DMA_DSP2_DREQ_4_5[7:0]	132	Reserved	Reserved

Table 4-7. DSP2_EDMA Default Request Mapping (continued)

DMA Request Line	DMA_ CROSSBAR Instance Number	DMA_CROSSBAR Configuration Register	DMA_ CROSSBAR Default Input Index	Default DMA Source Name	Default DMA Source Description
DMA_DSP2_DREQ_5	6	CTRL_CORE_DMA_DSP2_DREQ_4_5[23:16]	133	Reserved	Reserved
DMA_DSP2_DREQ_6	7	CTRL_CORE_DMA_DSP2_DREQ_6_7[7:0]	134	Reserved	Reserved
DMA_DSP2_DREQ_7	8	CTRL_CORE_DMA_DSP2_DREQ_6_7[23:16]	135	Reserved	Reserved
DMA_DSP2_DREQ_8	9	CTRL_CORE_DMA_DSP2_DREQ_8_9[7:0]	136	Reserved	Reserved
DMA_DSP2_DREQ_9	10	CTRL_CORE_DMA_DSP2_DREQ_8_9[23:16]	137	Reserved	Reserved
DMA_DSP2_DREQ_10	11	CTRL_CORE_DMA_DSP2_DREQ_10_11[7:0]	138	Reserved	Reserved
DMA_DSP2_DREQ_11	12	CTRL_CORE_DMA_DSP2_DREQ_10_11[23:16]	139	Reserved	Reserved
DMA_DSP2_DREQ_12	13	CTRL_CORE_DMA_DSP2_DREQ_12_13[7:0]	140	Reserved	Reserved
DMA_DSP2_DREQ_13	14	CTRL_CORE_DMA_DSP2_DREQ_12_13[23:16]	141	Reserved	Reserved
DMA_DSP2_DREQ_14	15	CTRL_CORE_DMA_DSP2_DREQ_14_15[7:0]	142	Reserved	Reserved
DMA_DSP2_DREQ_15	16	CTRL_CORE_DMA_DSP2_DREQ_14_15[23:16]	143	Reserved	Reserved
DMA_DSP2_DREQ_16	17	CTRL_CORE_DMA_DSP2_DREQ_16_17[7:0]	154	Reserved	Reserved
DMA_DSP2_DREQ_17	18	CTRL_CORE_DMA_DSP2_DREQ_16_17[23:16]	155	Reserved	Reserved
DMA_DSP2_DREQ_18	19	CTRL_CORE_DMA_DSP2_DREQ_18_19[7:0]	156	Reserved	Reserved
DMA_DSP2_DREQ_19	20	CTRL_CORE_DMA_DSP2_DREQ_18_19[23:16]	157	Reserved	Reserved
DMA_DSP2_DREQ_20 - DMA_DSP2_DREQ_63	N/A	N/A	N/A	Reserved	Reserved

4.3.5.1 DSP EDMA Wakeup Interrupt

This section provides description of the registers used for the **EDMA wakeup interrupt** functionality, including the EDMA_WAKE_INT IRQ status and enable fields. The EDMA Wakeup Interrupt allows incoming EDMA events to be latched and an interrupt sent to the DSP (if enabled). This interrupt is generated in the DSP_SYSTEM as an single "OR-ed" output of all external DMA requests latched in DSP subsystem. This output is further synchronized to DSP_FCLK and mapped as the EDMA_WAKE_INT event to the DSP_IRQ_31 input of the C66x DSP CorePac DSP_INTC. The C66x CPU is expected to service the interrupt by triggering the corresponding EDMA channel manually, or by servicing the request via normal reads and writes (instead of using the EDMA). This functionality is required since the EDMA is not capable of following the smart wakeup protocol.

NOTE: The DSP_SYS_DMAWAKEEN0 / DSP_SYS_DMAWAKEEN1 registers are used for enabling the assertion of the 'Mwakeup' asynchronous wakeup request to the device PRCM upon DMA requests reception. The interrupt functionality of the registers: DSP_SYS_EDMAWAKE0_x covered in this subsection is specifically for generating **an wake interrupt** to the DSP. In most cases, the enable mask for the two sets of registers should be set to the same value.

The EDMAWAKE0 registers corresponding to the EDMA Events 19 thru 0 (msbit to lsbit) are as follows:

- DSP_SYS_EDMAWAKE0_IRQSTATUS_RAW[19:0]
- DSP_SYS_EDMAWAKE0_IRQSTATUS[19:0]
- DSP_SYS_EDMAWAKE0_IRQENABLE_SET[19:0]
- DSP_SYS_EDMAWAKE0_IRQENABLE_CLR[19:0]

Following functional descriptions are valid for the above registers :

- **IRQ status raw register -** This register provides a per-event raw interrupt status vector. The Raw status is set even if the corresponding event is not enabled. Software can write '1' to set the (raw) status for debug purposes.
- **IRQ status register -** This register provides a per-event enabled interrupt status vector. The Enabled status is set if the corresponding event is enabled and the raw status is set. Software can write 1 to clear the (raw) status after the interrupt has been serviced. The clear takes effect even if the interrupt is not enabled.
- **IRQ enable register -** This register provides a per-event interrupt enable bit vector. Software can write 1 to set (i.e., enable the corresponding interrupt). Reads of this register return the actual state of the enable register (and is the same as reading the corresponding "IRQ Clear" - register)
- **IRQ clear register -** This register provides a per-event interrupt enable bit vector. Software can write 1 to clear (i.e., disable the corresponding interrupt). Reads of this register return the actual state of the enable register (and is the same as reading the corresponding "IRQ Set" register)

NOTE: A DSP_SYS_EDMAWAKE0_IRQSTATUS_RAW bit is set even if the corresponding event is NOT enabled in the DSP_SYS_EDMAWAKE0_IRQENABLE_SET[19:0].

4.3.6 DSP Intergated Memory Management Units

4.3.6.1 DSP MMUs Overview

A standalone memory management unit (DSP_MMU0) is included within the DSP1 (DSP1_MMU0) and DSP2 (DSP2_MMU0) subsystems boundaries. The DSP_MMU0 is integrated on the C66x CPU MDMA path to the device L3_MAIN interconnect. This provides several benefits including protection of the system memories from corruption by DSP1 and DSP2 accidental accesses.

A standalone memory management unit (DSP_MMU1) is included within the DSP1 (DSP1_MMU1) and DSP2 (DSP2_MMU1) subsystems boundaries. The DSP_MMU1 is integrated on the EDMA data path which starts from the L2 DSP_NoC interconnect and leaves the DSP subsystem on the DSP EDMA master port. This provides several benefits including protection of the device L3_MAIN memory space from corruption by DSP1 and DSP2 DMA (DSP1_EDMA and DSP2_EDMA, respectively) accidental accesses.

Both DSP MMUs generate interrupts which are internally mapped to the DSP C66x CorePac DSP_INTC and output to the device IRQ_CROSSBAR. See also the Section 4.2 and Section 4.3.4.

CAUTION

In the case of a page fault, a DSP C66x CorePac CPU is unable to service it's own DSP_MMU0 and DSP_MMU1 interrupts . The device host processor is expected to manage any TLB patches as necessary.

Both DSP MMUs (on MDMA and EDMA paths respectively) have identical functionalities.

- 32-bit input and ouput address width (to match L3_MAIN address width)
- 32 TLB cache entries
- 32 + 1 tags
- 128-bit data bus for MDMA and EDMA

4.3.6.2 Routing MDMA Traffic through DSP MMU0

DSP C66x CPU traffic initiated on the DSP MDMA port can be optionally routed through the DSP_MMU0 on the 32-bit MDMA address path. This is controlled in two levels :

- **global** by the DSP_SYSTEM register DSP_SYS_MMU_CONFIG [0] MMU0_EN bit. This bit acts as a mux-select : setting it to 0b1 enables requests to use the DSP_MMU0; clearing this bit to 0b0 disables MMU table lookup and causes accesses to use the non-translated address (MMU bypass). By default the DSP_MMU0 is disabled in DSP_SYSTEM and MDMA port traffic bypasses the DSP_MMU0.
- **local** by MMU enable control in a dedicated DSP_MMU0 memory mapped register. It is used to enable the MMU functionality after the page tables are programmed for MMU operation. For details, refer to the Chapter 15, *Memory Management Units*.

NOTE: For the DSP_MMU0 to operate, SW should enable it both at the DSP_SYSTEM global level and DSP_MMU0 local register level.

When enabling the DSP_MMU0, software must take care that no transactions are in flight through that MMU. This is typically handled by issueing a DSP "MFENCE" instruction operation. Note that the local enable bit inside the DSP_MMU0 must be configured as normal (refer to the Chapter 15, *Memory Management Units*.)

For more information on the MFENCE operation, refer to the section, *C66x CPU Instruction Set* of the *TMS320C66x DSP CPU and Instruction Set*).

In addition, the DSP_MMU0 traffic can be aborted in case of a lockup via the DSP_SYS_MMU_CONFIG [8] MMU0_ABORT bit. In other words, this bit can be used to clear a hang condition that may occur if the DSP_MMU0 encounters a page fault that cannot be serviced.

For more information on device DSP_MMU0 functionality and register settings, refer to the Section 15.3, *MMU Functional Description* and Section 15.5, *MMU Register Manual*, in the chapter, *Memory Management Units*, respectively.

4.3.6.3 Routing EDMA Traffic thorugh DSP MMU1

The DSP_EDMA traffics initiated on the DSP EDMA master port can be optionally routed through the DSP_MMU1 on the EDMA address path. This is controlled in two levels :

- **global** by the DSP_SYSTEM register DSP_SYS_MMU_CONFIG [4] MMU1_EN bit. This bit acts as a

mux-select : setting it to 0b1 enables requests to use the DSP_MMU1; clearing this bit to 0b0 disables MMU table lookup and causes accesses to use the non-translated address (MMU bypass). By default the DSP_MMU1 is disabled in DSP_SYSTEM and EDMA traffic bypasses the DSP_MMU1.

- **local** by MMU enable control in a dedicated DSP_MMU1 memory mapped register. It is used to enable the MMU functionality after the page tables are programmed for MMU operation. For details, refer to the Section 15.5, *MMU Register Manual*, in the Chapter 15,*Memory Management Units*.

NOTE: For the DSP_MMU1 to operate, SW should enable it both at the DSP_SYSTEM global level and DSP_MMU1 local register level.

When enabling the DSP_MMU1, software must take care that no transactions are in flight through that MMU. This is typically handled by disabling any EDMA transactions prior to enabling the MMU. Note that the local enable bit inside the DSP_MMU1 must be configured as normal (refer to the Chapter 15,*Memory Management Units*.)

For more information on the MFENCE operation, refer to the section, *C66x CPU Instruction Set* of the *TMS320C66x DSP CPU and Instruction Set*).

In addition, the DSP_MMU1 traffic can be aborted in case of a lockup via the DSP_SYS_MMU_CONFIG [12] MMU1_ABORT bit. In other words, this bit can be used to clear a hang condition that may occur if the DSP_MMU1 encounters a page fault that cannot be serviced.

For more information on device DSP_MMU1 functionality and settings, refer to the Section 15.3, *MMU Functional Description* and the Section 15.5, *MMU Register Manual*, in the chapter, *Memory Management Units*, respectively.

4.3.7 DSP Integrated EDMA Subsystem

This section represents an overview of the DSP integrated EDMA functionalities, as well as the subsystem level and device related register controls. For more details on the EDMA functionalities and programming registers, refer to the Chapter 11, *Enhanced DMA*.

4.3.7.1 DSP EDMA Overview

The enhanced-DMA subsystem which is part of the DSP1 (DSP1_EDMA) and the DSP2 (DSP2_EDMA) subsystems is the primary DMA engine for transfers between system memory (DDR and/or L3_MAIN SRAM) and DSP internal memories (L1s and L2).

The Channel Controller - DSP_EDMA_CC serves as the “user interface” of the DSP_EDMA. The two Transfer Controllers - DSP_EDMA_TC0 and DSP_EDMA_TC1 serve as the data transfer engines of the DSP_EDMA. The C66x CPU tipically programs the Channel Controller, which in turn submits Transfer Requests (TR) to the appropriate Transfer Controller. Interrupts are posted in the DSP_EDMA_CC upon transfer completion (if requested), and signaled to the C66x. The EDMA TC completion interrupt is not supported/connected.

The DSP_EDMA is primarily used to perform block transfers between DSP C66x CorePac memories (mostly L2 memory) and system memory (mostly DDR or L3 SRAM).

The DSP_EDMA is configured with 2 Queues (in the CC). Two DSP_EDMA traffic controllers (TC) offer high performance and preemptability of transfers. For typical use cases, it is expected that low latency/small payload transfers) use Queue0/TC0 and high bandwidth/large payload transfers (e.g., DDR on L3_MAIN or DSP local L2 SRAM) will use Queue1/TC1.

DSP_EDMA_CC configuration in the device features :

- 64x EDMA channels
- 8x QDMA channels
- 64x interrupt channels
- 128x PaRAM entries
- 2x Event Queues
- 2x Traffic controllers

- memory protection support
- channel mapping capability
- 8x memory protected and Shadow Regions

DSP_EDMA_TC0/TC1 configuration in the device features :

- 2048 Byte FIFO support
- multitag support
- 16-bit data bus
- 4-level destination register depth
- 7-bit address for internal FIFOs
- 16 IDs for Read commands
- 16 IDs for Write commands

NOTE: The device DSP integrated EDMA controller instances (DSP_EDMA_CC, DSP_EDMA_TC0 and DSP_EDMA_TC1) are functionally identical with the device EDMA controller instances (EDMA_TPCC, EDMA_TPTC0 and EDMA_TPTC1). The only difference is that the DSP_EDMA instances are located at different physical addresses.

For more details on the DSP_EDMA_CC, DSP_EDMA_TC0 and DSP_EDMA_TC1 controllers functionalities, refer to Section 11.4, *EDMA Controller Functional Description*, in Chapter 11, *Enhanced DMA*.

The DSP_EDMA instances, their corresponding registers summary and descriptions are covered in the Section 11.7, *EDMA Register Manual* of the Chapter 11, *Enhanced DMA*.

4.3.7.2 DSP System and Device Level Settings of DSP EDMA

DSP_EDMA traffic TC0 and TC1 controllers "Active" or "Idle" status: can be monitored in DSP_SYSTEM located :

- DSP_SYS_STAT[1] TC0_STAT bit
- DSP_SYS_STAT[2] TC1_STAT bit

The **default** burst size for both the DSP_EDMA_TC0 and DSP_EDMA_TC1 can be defined in DSP_SYSTEM register. This is achieved via programming :

- DSP_SYS_BUS_CONFIG [1:0] TC0_DBS
- DSP_SYS_BUS_CONFIG [5:4] TC1_DBS

There are also other DSP_EDMA controls associated with DSP_NoC interconnect pressure settings. For more details, refer to the Section 4.3.8.

The 3 error events associated with the DSP_EDMA_CC, DSP_EDMA_TC0 and DSP_EDMA_TC1 are exported outside the DSP C66x CorePac in the subsystem, and are able to trigger the ERRINT_IRQ aggregated interrupt output. See also corresponding **"tpcc_errint_level", "tptc_errint0_level" and "tptc_errint1_level"** events, respectively in the Table 4-5.

NOTE: The DSP_EDMA_CC, DSP_EDMA_TC0 and DSP_EDMA_TC1 events are NOT exported outside DSP subsystem. However they are merged (OR-ed) along with other error event sources within the DSP subsystem to produce a single ERRINT_IRQ interrupt exported outside the DSP subsystem.

For more details on ERRINT_IRQ generation and asscoiated event registers at DSP_SYSTEM level, refer to the Section 4.3.4.2.2.

The DSP_SYSTEM logic is assigned to route the external DMA requests to the EDMA hardware request inputs. Additionally, EDMA events can conditionally wake-up the DSP system from a low power mode, via software enabling DSP_SYSTEM MWakeup handshake with the device PRCM. This mechanism is described in the Section 4.3.5.

The programmable muxing of various external DMA request sources to the DSP EDMA. DMA_DSP1_DREQ_x and DMA_DSP2_DREQ_x input lines (where x=0 to 19) is covered in the Section 13.4.6.5, *DMA_CROSSBAR Module Functional Description*, of the Chapter 13, *Control Module*.

DSP1/DSP2 subsystem external DMA request sources : For the default DSP1 / DSP2 external DMA request sources, routed via the device DMA_CROSSBAR to the DSP1_EDMA / DSP2_EDMA channel controller inputs (DMA_DSP1_DREQ_i / DMA_DSP2_DREQ_i), respectively, refer to the Section 4.3.5.

4.3.8 DSP L2 interconnect Network

A 128-bit level 2 (L2) Interconnect from Arteris - FlexNoC® is instantiated in the DSP subsystem, outside the DSP C66x CorePac. It is signified as "DSP_NoC" throughout this chapter.

NOTE: The C66x master MDMA data does NOT flow through the DSP_NoC.

The system and local initiators on DSP_NoC are as follows :

- local C66x CPU 32-bit CFG master port which traffic is split via DSP_NoC fabric into several configuration target traffics inside and outside the DSP subsystem.
- SDMA initiator port on DSP_NoC which conveys accesses towards DSP Memories and memory-mapped registers initiated outside the DSP subsystem via the L3_MAIN interconnect.
- EDMA traffic controllers - TC0 read / write initiator ports
- EDMA traffic controllers - TC1 read / write initiator ports

NOTE: The DSP_ICFG space is not visible to SDMA initiators (DSP_EDMA or DSP hosts on L3_MAIN) with CFG traffic.

The targets on the DSP_NoC are as follows :

- DSP C66x CorePac SDMA port
- Internal CFG targets on the DSP_NoC :
  - DSP_MMU0 Cfg
  - DSP_MMU1 Cfg
  - DSP_SYSTEM Cfg
  - DSP_EDMA_CC Cfg
  - DSP_EDMA_TC0 Cfg
  - DSP_EDMA_TC1 Cfg
- 32-bit CFG port on L3_MAIN (it acts as master on the L3_MAIN)
- EDMA Target port which conveys EDMA bidi transfers outside the DSP (through or bypassing DSP_MMU1).

The Table 4-8 summarizes the interconnections which can be established between DSP initiators and targets over the L2 DSP_NoC in the device. In this table HW implemented interconnections are marked with an asterics.

Table 4-8. DSP_NoC Defined Connectivities

		DSP_NoC Initiators			
		DSP C66x CorePac CFG init	EDMA_TC0 init	EDMA_TC1 init	SDMA init (mapped to SDMA port on L3_MAIN)
DSP_NoC Targets	DSP C66x CorePac SDMA (slave) port	n.a.	*	*	*
	DSP_MMU0 Cfg	*	n.a.	n.a.	*
	DSP_MMU1 Cfg	*	n.a.	n.a.	*
	DSP_SYSTEM Cfg	*	n.a.	n.a.	*
	DSP_EDMA_CC Cfg	*	n.a.	n.a.	*
	DSP_EDMA_TC0 Cfg	*	n.a.	n.a.	*
	DSP_EDMA_TC1 Cfg	*	n.a.	n.a.	*
	DSP_NoC Cfg	*	n.a.	n.a.	*
	Cfg port (Cfg Init on L3_MAIN)	*	n.a.	n.a.	n.a.
	Master DMA port (DSP DMA init on L3_MAIN)	n.a.	*	*	n.a.

A DSP_NoC error event (combination of several local to the interconnect events) is exported outside the DSP C66x CorePac in the subsystem, and can be enabled to trigger the ERRINT_IRQ aggregated interrupt output. See also corresponding **"noc_errint_level"** event in the Table 4-5.

NOTE: The DSP_NoC event is NOT exported outside DSP subsystem. However it is merged (OR-ed) along with other error event sources within the DSP subsystem to produce a single ERRINT_IRQ interrupt exported outside the DSP subsystem.

For more details on ERRINT_IRQ generation and asscoiated event registers at DSP_SYSTEM level, refer to the Section 4.3.4.2.2.

4.3.8.1 DSP Public Firewall Settings

The DSP1 and DSP2 L2 Interconnect (DSP1_NoC and DSP2_NoC, respectively) implements two firewalls – dsp firewall0 (DSP_FW0) is used to protect DSP_MMU0's configuration space (which includes the TLB) and dsp firewall1 is used to protect DSP_MMU1's configuration space (which includes the TLB). Access permission is based on the privilege level, domain, ConnID, and access types of a request.

The default value of 0xFFFF_FFFF in the MRM region 0 permission registers:

- L3_DSPSS_INIT_OCP_MMU0_CTRL_TARG_OCP_FW_503000_MRM_PERMISSION_REGION_LOW_0
- L3_DSPSS_INIT_OCP_MMU0_CTRL_TARG_OCP_FW_503000_MRM_PERMISSION_REGION_HIGH_0
- L3_DSPSS_INIT_OCP_MMU1_CTRL_TARG_OCP_FW_504000_MRM_PERMISSION_REGION_LOW_0
- L3_DSPSS_INIT_OCP_MMU1_CTRL_TARG_OCP_FW_504000_MRM_PERMISSION_REGION_HIGH_0

permits any requestor to access the DSP_MMU0 and DSP_MMU1 configuration space.

For more information on the access region definitions, public privilege access, public user access and initiator permission settings, which are identical between DSP_NoC firewalls and L3_MAIN interconnect firewalls, refer to the Section 9.2.3.7.3, *L3_MAIN Firewall Functionality*, in the, Section 9.2, *L3_MAIN Interconnect*.

There are also several other DSP_NoC registers - L3_DSPSS_INIT_OCP_MMU0_CTRL_TARG_OCP_FW_503000_LOGICAL_ADDR_ERRLOG_0, L3_DSPSS_INIT_OCP_MMU0_CTRL_TARG_OCP_FW_503000_REGUPDATE_CONTROL used for error handling, firewall reset and other purposes. These are functionally identical with the corresponding L3_MAIN interconnect registers, described in the Section 9.2.3.8, *L3_MAIN Interconnect Error Handling*, in the Section 9.2, *L3_MAIN Interconnect*.

The various firewall access control registers are part of the C66x CPU local accessible - DSP_FW_L2_NOC_CFG address space, and L3_MAIN initiators accessible DSP1/2_FW_L2_NOC_CFG configuration space. The corresponding MMU0 and MMU1 configuration space firewall registers (DSP_FW0 starting at offset **0x0000_0000** , and DSP_FW1 starting at offset **0x0000_1000**) are summarized and described in the Section 4.4.4.

4.3.8.2 DSP NoC Flag Mux and Error Log Registers

The DSP_NoC registers (starting at offset >= 0x0000_4000) are used for error logging and flag muxing purposes. The status information stored in there can be used for example to resolve issues related to DSP_NoC access conflicts, for debug purposes, etc.

For more information, refer to the Section 9.2.3.5, *Flag Muxing*, in the Section 9.2, *L3_MAIN Interconnect*.

4.3.8.3 DSP NoC Arbitration

A pressure based arbitration is implemented for the DSP_NoC interconnect.

A DSP_NoC local MFlag mechanism is used but it is SW controlled in the DSP_SYSTEM configuration space.

This is done via the register DSP_SYS_BUS_CONFIG bitfields :

- TC0_L2PRES - for DSP_EDMA_TC0 pressure control
- TC1_L2PRES - for DSP_EDMA_TC1 pressure control
- CFG_L2PRES - for the DSP C66x CorePac 32-bit CFG pressure control
- SDMA_L2PRES - for pressure control of the DSP system and L3_MAIN accesses targetting the DSP C66x CorePac SDMA slave port

The pressure for each port is signaled on a MFlag[1:0] bus and conveys a value of 0 (lowest), 1 (medium), or 3 (highest). A value of 0x2 is reserved/undefined.

NOTE: The default pressure level for all ports is 0x0 and is recommended for most systems. This results in round-robin arbitration across active requests.

4.3.9 DSP Boot Configuration

DSP1 subsystem boot vector input which defines the 22-bit DSP1 Boot Address is mapped to the device core control module register CTRL_CORE_CONTROL_DSP1_RST_VECT[21:0] DSP1_RST_VECT bitfield. DSP2 subsystem boot vector input which defines the 22-bit DSP2 Boot Address is mapped to the device core control module register CTRL_CORE_CONTROL_DSP2_RST_VECT[21:0] DSP2_RST_VECT bitfield.In general, the device host loads code to a given address location in the device system memory, sets the DSP1_RST_VECT / DSP2_RST_VECT bitfield to the address value, and then release the DSP1 / DSP2 from reset. At that point, the DSP1 / DSP2 will begin fetching code from that location.

NOTE: If the values of the control core module CTRL_CORE_CONTROL_DSP1_RST_VECT[21:0] / CTRL_CORE_CONTROL_DSP2_RST_VECT [21:0] register change, the values will be taken into account by DSP upon the next reset.

NOTE: Upon device boot time (a power-on reset applied), the device "sysboot15" input latched in the Control Module bootstrap register defines the value of DSP functional clock divider (2 or 3). For more details, refer to the Section 4.3.3.1

4.3.10 DSP Internal and External Memory Views

4.3.10.1 C66x CPU View of the Address Space

The **C66x CPU View** represents the view from the DSP, which result from program fetches, or load / store instructions. Accesses to DSP memories (L1P, L1D, L2) and to DSP Internal configuration space (DSP_ICFG) are intercepted within the DSP C66x CorePac (whether using local or global addresses).

The DSP C66x CorePac CFG (C66x CPU 32-bit master port) interface is strictly for non-cacheable loads and stores, and is intended to be used for I/O space or memory mapped registers (MMR) space. DSP C66x CorePac CFG accesses are routed / arbitrated by the DSP_NoC L2 inteconnect. DSP C66x CorePac CFG accesses are mapped between 0x01C0_0000 and 0x0FFF_FFFF.

NOTE: The DSP C66x CorePac CFG initiator interface is strictly for non-cacheable loads and stores within MMR and I/O spaces.

DSP configuration accesses to **external to DSP subsystem** peripherals can be issued on either the DSP subsystem CFG Master port which is mapped to the device L3_MAIN or DSP 32-bit CFG "system" interface is connected to the chip level L3_MAIN interconnect and used to access L3_MAIN addressses that are mapped between 0x01C0_0000 and 0x0FFF_FFFF.

DSP **accesses (to non-DSP memories like SDRAM on L3_MAIN) for addresses above 0x1000_0000 are handled via the DSP (XMC) MDMA 128-bit master interface and are routed to the DSP subsystem MDMA Initiator port either through DSP_ MMU0 or bypassing MMU.**

NOTE: In some cases, L3_MAIN peripherals may be mapped to both the MDMA bus and the CFG bus. In that case, there may be a latency advantage of using the CFG address for those peripherals

Table 4-9 shows the DSP C66x CPU memory view of the various DSP C66x CorePac internal and external resources.

Table 4-9. C66x CPU View Map

C66x CPU View (DSP C66x CorePac Internal only, MDMA or CFG init ports)[1]		Size	DSP Memory Region	Function
Start Address	End Address			
0x0080_0000	0x0084_7FFF	288 KiB	DSP_L2	DSP L2 SRAM (local)
0x00E0_0000	0x00E0_7FFF	32 KiB	DSP_L1P	DSP L1P SRAM (local)
0x00F0_0000	0x00F0_7FFF	32KiB	DSP_L1D	DSP L1D SRAM (local)
0x0180_0000	0x01BF_FFFF	4096 KiB	DSP_ICFG	DSP Internal CFG [2]
0x01D0_0000	0x01D0_0FFF	4 KiB	DSP_SYSTEM	DSP_SYSTEM Memory Mapped Registers block
0x01D0_1000	0x01D0_1FFF	4 KiB	DSP_MMU0CFG	DSP MMU0 configuration / registers
0x01D0_2000	0x01D0_2FFF	4 KiB	DSP_MMU1CFG	DSP MMU1 configuration / registers
0x01D0_5000	0x01D0_5FFF	4 KiB	DSP_EDMA_TC0	DSP_EDMA Transfer Controller 0
0x01D0_6000	0x01D0_6FFF	4 KiB	DSP_EDMA_TC1	DSP_EDMA Transfer Controller 1
0x01D0_7000	0x01D0_7FFF	4 KiB	DSP_NoC	DSP L2 interconnect registers
0x01D1_0000	0x01D1_7FFF	32 KiB	DSP_EDMA_CC	DSP_EDMA Channel Controller
0x0200_0000	0x020F_FFFF	1 MiB	EVE	DSP configuration traffic to the EVE (mapped on the DSP CFG interface)
0x0210_0000	0x023F_FFFF	3 MiB	Reserved	Reserved
0x0330_0000	0x033F_FFFF	1 MiB	EDMA_TPCC	DSP configuration traffic to the EDMA_TPCC (mapped on the DSP CFG interface)
0x0340_0000	0x034F_FFFF	1 MiB	EDMA_TC0	DSP configuration traffic to the EDMA_TC0 (mapped on the DSP CFG interface)
0x0350_0000	0x035F_FFFF	1 MiB	EDMA_TC1	DSP configuration traffic to the EDMA_TC1 (mapped on the DSP CFG interface)
0x0800_0000	0x0800_FFFF	64 KiB	DSP_XMC_CTRL MMRs	DSP internal MMRs for XMC controller (non-cache)
0x0802_0000	0x080F_FFFF	896 KiB	MDMA non-cache	MDMA initiator (non-cache) to L3_MAIN (DSP_MMU0)
0x0810_0000	0x0BBF_FFFF	59 MiB		
0x1000_0000	0x10FF_FFFF	16 MiB	DSP1 L1P, L1D and L2 memories	An image of DSP1 C66x CorePac internal space - only L1P, L1D and L2 memories (global) [3]
0x1000_0000	0x10FF_FFFF	16 MiB	DSP2 MDMA (cached)	DSP2 MDMA initiator (cached) to L3_MAIN (through DSP2_MMU0)

(1) Only the C66x CPU view of device implemented functional memory regions are shown. The remaining regions are reserved.

(2) The internal configuration space registers are visible ONLY to the C66x CPU (DSP_C0) within the DSP C66x CorePac, i.e. they are NOT visible to initiators outside the DSP C66x CorePac.

(3) The DSP1 CPU sees an image of its own memories in the 0x1000_0000 - 0x10FF_FFFF address range (the same mapped also at lower addresses 0x0080_0000 - 0x00F0_7FFF). On the other side, DSP2 CPU addresses, generated in the same range (with DSP2_MMU0 involved), are mapped to the DSP2 MDMA port (cached transfers to/from L3_MAIN connected system memories).

Table 4-9. C66x CPU View Map (continued)

C66x CPU View (DSP C66x CorePac Internal only, MDMA or CFG init ports)[1]		Size	DSP Memory Region	Function
Start Address	End Address			
0x1100_0000	0x11FF_FFFF	16 MiB	DSP2 L1P, L1D and L2 memories	An image of DSP2 C66x CorePac internal space - only L1P, L1D and L2 memories (global) [4]
			DSP1 MDMA (cached)	MDMA initiator (cached) to L3_MAIN (through DSP1_MMU0)
0x1200_0000	0x1FFF_FFFF	224 MiB	MDMA (cached)	DSP1 and DSP2 MDMA initiator (cached) to L3_MAIN (through DSP1_MMU0 / DSP2_MMU0, respectively)
0x2000_0000	0xFFFF_FFFF	3584 MiB	MDMA (cached)	DSP1 and DSP2 MDMA initiator (cached) to L3_MAIN (through DSP1_MMU0 / DSP2_MMU0, respectively)

(4) The DSP2 CPU sees an image of its own memories in the 0x1100_0000 - 0x11FF_FFFF address range (the same mapped also at lower addresses 0x0080_0000 - 0x00F0_7FFF). On the other side, DSP1 CPU addresses, generated in the same range (with DSP1_MMU0 involved), are mapped to the DSP1 MDMA port (cached transfers to / from L3_MAIN connected system memories).

Refer to the Section 2.2, *L3_MAIN Memory Space Mapping*, in the chapter, *Memory Mapping*, for the addresses of the L3_MAIN space memory-mapped registers. Refer to the Section 2.6, *DSP Subsystem Memory Space Mapping* in the same chapter for a description of the DSP1 and DSP2 internal memory, additional memory, and peripherals that the DSP1 and DSP2 have access to.

4.3.10.2 DSP_EDMA View of the Address Space

EDMA is able to initiate internal accesses directly to the DSP memories via the DSP C66x CorePac SDMA bus. The access is conducted to the DSP C66x CorePac internal memories over the L2 DSP_NoC interconnect.

Table 4-10 shows the DSP integrated EDMA controller memory view of the various DSP C66x CorePac internal and external resources.

Table 4-10. DSP EDMA Controller View Map

DSP_EDMA Controller View (EDMA master internal / external port)		Size	DSP Memory Region	Function
Start Address	End Address			
0x0080_0000	0x0084_7FFF	288 KiB	DSP_L2	DSP L2 SRAM (local)
0x00E0_0000	0x00E0_7FFF	32 KiB	DSP_L1P	DSP L1P SRAM (local)
0x00F0_0000	0x00F0_7FFF	32KiB	DSP_L1D	DSP L1D SRAM (local)
0x0802_0000	0x0BBF_FFFF	59 MiB	EDMA to L3_MAIN	EDMA initiator (DSP_MMU1)
0x1000_0000	0x10FF_FFFF	16 MiB	DSP L1/L2	An image of DSP C66x CorePac internal space - only L1P, L1D and L2 memories (global) [1]
0x2000_0000	0xFFFF_FFFF	3584 MiB	DMA OCP	L3_MAIN interconnect memory via MMU1 / DMA OCP Initiator

(1) The internal configuration space registers DSP_ICFG are visible ONLY to the C66x CPU (DSP_C0) within the DSP C66x CorePac , i.e. they are NOT visible to the DSP_EDMA.

Access from EDMA to external resources on L3_MAIN are routed via DSP subsystem **EDMA initiator port**. **Note that these accesses are transferred through the DSP_MMU1 memory management unit.**

NOTE: The DSP_EDMA can NOT access the DSP_ICFG (DSP C66x CorePac internal) addresses.

With the DSP_MMU1 disabled, the subset of the memory map used for DSP_EDMA internal accesses will NOT be visible. Thus only addresse which equal 0x2000_0000 and above will be considered as valid 32-bit addresses (i.e. L3_MAIN space accesses only).

Refer to the Section 2.2, *L3_MAIN Memory Space Mapping*, in the chapter, *Memory Mapping*, for the addresses of the L3_MAIN space memory-mapped registers. Refer to the Section 2.6, *DSP Subsystem Memory Space Mapping* in the same chapter for a description of the DSP1 and DSP2 internal memory, additional memory, and peripherals that the DSP1 and DSP2 have access to.

4.3.10.3 L3_MAIN View of the DSP Address Space

System initiated accesses (i.e. external-to-DSP accesses over device L3_MAIN) to DSP are issued over the DSP SDMA target port.

NOTE: The MSB-bits of the address are truncated to only provide an 8 MiB view of the memory map within the DSP subsystem. Notice that the relative offsets of the DSP CFG space is different for the OCP SDMA target port relative to the DSP internal initiators.

The SDMA target bus is able to access internal subsystem address space (such as DSP_EDMA, DSP_MMU0, DPS_MMU1, etc.), or the DSP local memory address space. The SDMA target bus **is NOT able to access the DSP ICFG space (such as DSP_INTC, DSP_BWM, etc)** or the other initiator ports on the DSP subsystem boundary (i.e., accesses cannot go through DSPSS to get to the DSP_EDMA initiator port, L3_MAIN CFG initiator port).

The DSP slave DMA port memory map - Table 4-11 shows an 8 MiB window (23-bit address) both from the SDMA Target bus (0x0000_0000 through 0x007F_FFFF), as well as the EDMA (0x0080_0000 through 0x00FF_FFFF). The DSP C66x CorePac internally views itself as a 16 MiB window where 0x0000_0000 through 0x007F_FFFF is reserved, L2 SRAM starts at 0x0080_0000, L1P SRAM starts at 0x00E0_0000, and L1D SRAM starts at 0x00F0_0000).

Table 4-11. SDMA Target Port Memory Map

System L3_MAIN View[1]		Size	DSP Memory Region	Function
Start Address	End Address			
0x0000_0000	0x0004_7FFF	288 KiB	DSP_L2	DSP L2 SRAM (local)
0x0050_0000	0x0050_0FFF	4 KiB	DSP_SYSTEM	DSP SYSTEM MMR Block
0x0050_1000	0x0050_1FFF	4 KiB	DSP_MMU0CFG	DSP MMU0 configuration / regs
0x0050_2000	0x0050_2FFF	4 KiB	DSP_MMU1CFG	DSP MMU1 configuration / regs
0x0050_5000	0x0050_5FFF	4 KiB	DSP_EDMA_TC0	DSP EDMA Transfer Controller 0
0x0050_6000	0x0050_6FFF	4 KiB	DSP_EDMA_TC1	DSP EDMA Transfer Controller 1
0x0050_7000	0x0050_7FFF	4 KiB	DSP_NoC	DSP L2 Interconnect registers
0x0051_0000	0x0051_7FFF	32 KiB	DSP_EDMA_CC	DSP EDMA Channel Controller
0x0060_0000	0x0060_7FFF	32 KiB	DSP_L1P	DSP L1P SRAM (local)
0x0070_0000	0x0070_7FFF	32 KiB	DSP_L1D	DSP L1D SRAM (local)

(1) Only system (L3_MAIN) view over functionally used regions are shown. The remaining regions are reserved.

Refer to the Section 2.2, *L3_MAIN Memory Space Mapping*, in the chapter, *Memory Mapping*, for the addresses of the L3_MAIN space memory-mapped registers. Refer to the Section 2.6, *DSP Subsystem Memory Space Mapping* in the same chapter for a description of the DSP1 and DSP2 internal memory, additional memory, and peripherals that the DSP1 and DSP2 have access to.

4.4 DSP Subsystem Register Manual

This section describes the DSP Subsystem instances registers.

4.4.1 DSP Subsystem Instance Summary

Table 4-12. DSP Subsystem Instance Summary

Module Name	Module Base Address	Size
DSP_ICFG	0x0180 0000[(1)]	4 KiB
DSP_SYSTEM	0x01D0 0000[(1)]	256 Bytes
DSP_FW_L2_NOC_CFG	0x01D0 3000[(1)]	8576 Bytes
DSP1_SYSTEM	0x40D0 0000	256 Bytes
DSP1_FW_L2_NOC_CFG	0x40D0 3000	8576 Bytes
DSP2_SYSTEM	0x4150 0000	256 Bytes
DSP2_FW_L2_NOC_CFG	0x4150 3000	8576 Bytes

(1) **The registers of DSP subsystem instances prefixed only with DSP in the name, and NOT DSP1 or DSP2, are NOT visible on the device L3_MAIN. They are visible only within the DSP_ICFG internal configuration space hence accessible only by the DSP C66x CPU.**

NOTE: For more details on the DSP_MMU0 and DSP_MMU1 registers, as well as their :

- DSP_MMU0CFG and DSP_MMU1CFG physical addresses accesible only by DSP_C0 CPU core in the DSP subsystem
- DSP1_MMU0CFG and DSP1_MMU1CFG physical addresses visible on L3_MAIN
- DSP2_MMU0CFG and DSP2_MMU1CFG physical addresses visible on L3_MAIN

refer to Section 15.5, *MMU Register Manual*, in Chapter 15, *Memory Management Units*.

NOTE: For more details on the DSP_EDMA_CC, DSP_EDMA_TC0 and DSP_EDMA_TC1 registers, as well as their :

- DSP1_EDMA_CC, DSP1_EDMA_TC0 and DSP1_EDMA_TC1 physical addresses visible on L3_MAIN
- DSP2_EDMA_CC, DSP2_EDMA_TC0 and DSP2_EDMA_TC1 physical addresses visible on L3_MAIN

refer to Section 11.7, *EDMA Register Manual*, in Chapter 11, *Enhanced DMA*.

CAUTION

The L1P, L1D and L2 memory controller registers mapped in the L3_MAIN are limited to 32-bit data access; 16- and 8-bit access are not allowed and can corrupt register content.

4.4.2 DSP_ICFG Registers

4.4.2.1 DSP_ICFG Register Summary

NOTE: The DSP_ICFG addresses are visible only within the DSP core internal configuration space.

Table 4-13. DSP_ICFG Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	DSP_ICFG Physical Address	Register Reset Value[(1)]
EVTFLAG0	R	32	0x0000 0000	0x0180 0000	0x0000 0000
EVTFLAG1	R	32	0x0000 0004	0x0180 0004	0x0000 0000
EVTFLAG2	R	32	0x0000 0008	0x0180 0008	0x0000 0000
EVTFLAG3	R	32	0x0000 000C	0x0180 000C	0x0000 0000
EVTSET0	W	32	0x0000 0020	0x0180 0020	0x0000 0000
EVTSET1	W	32	0x0000 0024	0x0180 0024	0x0000 0000
EVTSET2	W	32	0x0000 0028	0x0180 0028	0x0000 0000
EVTSET3	W	32	0x0000 002C	0x0180 002C	0x0000 0000
EVTCLR0	W	32	0x0000 0040	0x0180 0040	0x0000 0000
EVTCLR1	W	32	0x0000 0044	0x0180 0044	0x0000 0000
EVTCLR2	W	32	0x0000 0048	0x0180 0048	0x0000 0000
EVTCLR3	W	32	0x0000 004C	0x0180 004C	0x0000 0000
EVTMASK0	RW	32	0x0000 0080	0x0180 0080	0x0000 0000
EVTMASK1	RW	32	0x0000 0084	0x0180 0084	0x0000 0000
EVTMASK2	RW	32	0x0000 0088	0x0180 0088	0x0000 0000
EVTMASK3	RW	32	0x0000 008C	0x0180 008C	0x0000 0000
MEVTFLAG0	R	32	0x0000 00A0	0x0180 00A0	0x0000 0000
MEVTFLAG1	R	32	0x0000 00A4	0x0180 00A4	0x0000 0000
MEVTFLAG2	R	32	0x0000 00A8	0x0180 00A8	0x0000 0000
MEVTFLAG3	R	32	0x0000 00AC	0x0180 00AC	0x0000 0000
EXPMASK0	RW	32	0x0000 00C0	0x0180 00C0	0xFFFF FFFF
EXPMASK1	RW	32	0x0000 00C4	0x0180 00C4	0xFFFF FFFF
EXPMASK2	RW	32	0x0000 00C8	0x0180 00C8	0xFFFF FFFF
EXPMASK3	RW	32	0x0000 00CC	0x0180 00CC	0xFFFF FFFF
MEXPFLAG0	R	32	0x0000 00E0	0x0180 00E0	0x0000 0000
MEXPFLAG1	R	32	0x0000 00E4	0x0180 00E4	0x0000 0000
MEXPFLAG2	R	32	0x0000 00E8	0x0180 00E8	0x0000 0000
MEXPFLAG3	R	32	0x0000 00EC	0x0180 00EC	0x0000 0000
INTMUX1	RW	32	0x0000 0104	0x0180 0104	0x0706 0504
INTMUX2	RW	32	0x0000 0108	0x0180 0108	0x0B0A 0908
INTMUX3	RW	32	0x0000 010C	0x0180 010C	0x0F0E 0D0C
AEGMUX0	RW	32	0x0000 0140	0x0180 0140	0x0302 0100
AEGMUX1	RW	32	0x0000 0144	0x0180 0144	0x0706 0504
INTXSTAT	RW	32	0x0000 0180	0x0180 0180	0x0000 0000
INTXCLR	RW	32	0x0000 0184	0x0180 0184	0x0000 0000
INTDMASK	RW	32	0x0000 0188	0x0180 0188	0x0000 0000
EVTASRT	RW	32	0x0000 01C0	0x0180 01C0	0x0302 0100
PDCCMD	RW	32	0x0001 0000	0x0181 0000	0x0000 0000
MM_REVID	RW	32	0x0001 2000	0x0181 2000	0x0000 0000
IDMA0_STAT	RW	32	0x0002 0000	0x0182 0000	0x0000 0000

(1) This column considers not ONLY the DSP C66x CorePac specific reset values, but also the device level specific reset values.

Table 4-13. DSP_ICFG Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	DSP_ICFG Physical Address	Register Reset Value[1]
IDMA0_MASK	RW	32	0x0002 0004	0x0182 0004	0x0000 0000
IDMA0_SOURCE	RW	32	0x0002 0008	0x0182 0008	0x0000 0000
IDMA0_DEST	RW	32	0x0002 000C	0x0182 000C	0x0000 0000
IDMA0_COUNT	RW	32	0x0002 0010	0x0182 0010	0x0000 0000
IDMA1_STAT	RW	32	0x0002 0100	0x0182 0100	0x0000 0000
IDMA1_SOURCE	RW	32	0x0002 0108	0x0182 0108	0x0000 0000
IDMA1_DEST	RW	32	0x0002 010C	0x0182 010C	0x0000 0000
IDMA1_COUNT	RW	32	0x0002 0110	0x0182 0110	0x0000 0000
CPUARBE	RW	32	0x0002 0200	0x0182 0200	0x0001 0010
IDMAARBE	RW	32	0x0002 0204	0x0182 0204	0x0000 0010
SDMAARBE	RW	32	0x0002 0208	0x0182 0208	0x0000 0001
ECFGARBE	RW	32	0x0002 0210	0x0182 0210	0x0007 0000
ICFGMPFAR	R	32	0x0002 0300	0x0182 0300	0x0000 0000
ICFGMPFSR	RW	32	0x0002 0304	0x0182 0304	0x0000 0000
ICFGMPFCR	RW	32	0x0002 0308	0x0182 0308	0x0000 0000
ECFGERR	RW	32	0x0002 0408	0x0182 0408	0x0000 0000
ECFGERRCLR	RW	32	0x0002 040C	0x0182 040C	0x0000 0000
PAMAP0	RW	32	0x0002 0500	0x0182 0500	0x0000 0000
PAMAP1	RW	32	0x0002 0504	0x0182 0504	0x0000 0001
PAMAP2	RW	32	0x0002 0508	0x0182 0508	0x0000 0002
PAMAP3	RW	32	0x0002 050C	0x0182 050C	0x0000 0003
PAMAP4	RW	32	0x0002 0510	0x0182 0510	0x0000 0004
PAMAP5	RW	32	0x0002 0514	0x0182 0514	0x0000 0005
PAMAP6	RW	32	0x0002 0518	0x0182 0518	0x0000 0006
PAMAP7	RW	32	0x0002 051C	0x0182 051C	0x0000 0007
PAMAP8	RW	32	0x0002 0520	0x0182 0520	0x0000 0007
PAMAP9	RW	32	0x0002 0524	0x0182 0524	0x0000 0007
PAMAP10	RW	32	0x0002 0528	0x0182 0528	0x0000 0007
PAMAP11	RW	32	0x0002 052C	0x0182 052C	0x0000 0007
PAMAP12	RW	32	0x0002 0530	0x0182 0530	0x0000 0007
PAMAP13	RW	32	0x0002 0534	0x0182 0534	0x0000 0007
PAMAP14	RW	32	0x0002 0538	0x0182 0538	0x0000 0007
PAMAP15	RW	32	0x0002 053C	0x0182 053C	0x0000 0007
L2CFG	RW	32	0x0004 0000	0x0184 0000	0x0100 0000
L1PCFG	RW	32	0x0004 0020	0x0184 0020	0x0000 0007
L1PCC	RW	32	0x0004 0024	0x0184 0024	0x0000 0000
L1DCFG	RW	32	0x0004 0040	0x0184 0040	0x0000 0007
L1DCC	RW	32	0x0004 0044	0x0184 0044	0x0000 0000
CPUARBU	RW	32	0x0004 1000	0x0184 1000	0x0001 0010
IDMAARBU	RW	32	0x0004 1004	0x0184 1004	0x0000 0010
SDMAARBU	RW	32	0x0004 1008	0x0184 1008	0x0000 0001
UCARBU	RW	32	0x0004 100C	0x0184 100C	0x0000 0020
MDMAARBU	RW	32	0x0004 1010	0x0184 1010	0x0607 0000
CPUARBD	RW	32	0x0004 1040	0x0184 1040	0x0001 0010
IDMAARBD	RW	32	0x0004 1044	0x0184 1044	0x0000 0010
SDMAARBD	RW	32	0x0004 1048	0x0184 1048	0x0000 0001
UCARBD	RW	32	0x0004 104C	0x0184 104C	0x0000 0020

Table 4-13. DSP_ICFG Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	DSP_ICFG Physical Address	Register Reset Value[1]
L2WBAR	W	32	0x0004 4000	0x0184 4000	0x0000 0000
L2WWC	RW	32	0x0004 4004	0x0184 4004	0x0000 0000
L2WIBAR	W	32	0x0004 4010	0x0184 4010	0x0000 0000
L2WIWC	RW	32	0x0004 4014	0x0184 4014	0x0000 0000
L2IBAR	W	32	0x0004 4018	0x0184 4018	0x0000 0000
L2IWC	RW	32	0x0004 401C	0x0184 401C	0x0000 0000
L1PIBAR	W	32	0x0004 4020	0x0184 4020	0x0000 0000
L1PIWC	RW	32	0x0004 4024	0x0184 4024	0x0000 0000
L1DWIBAR	W	32	0x0004 4030	0x0184 4030	0x0000 0000
L1DWIWC	RW	32	0x0004 4034	0x0184 4034	0x0000 0000
L1DWBAR	W	32	0x0004 4040	0x0184 4040	0x0000 0000
L1DWWC	RW	32	0x0004 4044	0x0184 4044	0x0000 0000
L1DIBAR	W	32	0x0004 4048	0x0184 4048	0x0000 0000
L1DIWC	RW	32	0x0004 404C	0x0184 404C	0x0000 0000
L2WB	RW	32	0x0004 5000	0x0184 5000	0x0000 0000
L2WBINV	RW	32	0x0004 5004	0x0184 5004	0x0000 0000
L2INV	RW	32	0x0004 5008	0x0184 5008	0x0000 0000
L1PINV	RW	32	0x0004 5028	0x0184 5028	0x0000 0000
L1DWB	RW	32	0x0004 5040	0x0184 5040	0x0000 0000
L1DWBINV	RW	32	0x0004 5044	0x0184 5044	0x0000 0000
L1DINV	RW	32	0x0004 5048	0x0184 5048	0x0000 0000
L2EDSTAT	RW	32	0x0004 6004	0x0184 6004	0x0000 0001
L2EDCMD	RW	32	0x0004 6008	0x0184 6008	0x0000 0001
L2EDADDR	RW	32	0x0004 600C	0x0184 600C	0x0000 0000
L2EDCPEC	RW	32	0x0004 6018	0x0184 6018	0x0000 0000
L2EDCNEC	RW	32	0x0004 601C	0x0184 601C	0x0000 0000
MDMAERR	RW	32	0x0004 6020	0x0184 6020	0x0000 0000
MDMAERRCLR	RW	32	0x0004 6024	0x0184 6024	0x0000 0000
L2EDCEN	RW	32	0x0004 6030	0x0184 6030	0x0000 001F
L1PEDSTAT	RW	32	0x0004 6404	0x0184 6404	0x0000 0000
L1PEDCMD	RW	32	0x0004 6408	0x0184 6408	0x0000 0000
L1PEDADDR	RW	32	0x0004 640C	0x0184 640C	0x0000 0000
MARk[2]	RW	32	0x0004 8000 + (0x4*k)	0x0184 8000 + (0x4*k)	See [3]
L2MPFAR	R	32	0x0004 A000	0x0184 A000	0x0000 0000
L2MPFSR	RW	32	0x0004 A004	0x0184 A004	0x0000 0000
L2MPFCR	RW	32	0x0004 A008	0x0184 A008	0x0000 0000
L2MPPAm[4]	RW	32	0x0004 A200 + (0x4*m)	0x0184 A200 + (0x4*m)	0x0000 FFFF
L1PMPFAR	R	32	0x0004 A400	0x0184 A400	0x0000 0000
L1PMPFSR	RW	32	0x0004 A404	0x0184 A404	0x0000 0000
L1PMPFCR	RW	32	0x0004 A408	0x0184 A408	0x0000 0000
L1PMPPA16	RW	32	0x0004 A640	0x0184 A640	0x0000 FFFF

(2) k = 0 to 255

(3) MAR0 = 0x0000 0001;
MAR1…MAR11 = 0x0000 0000;
MAR12…MAR15 = 0x0000 000D;
MAR16…MAR255 = 0x0000 000C.

(4) m = 0 to 31

Table 4-13. DSP_ICFG Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	DSP_ICFG Physical Address	Register Reset Value[1]
L1PMPPA17	RW	32	0x0004 A644	0x0184 A644	0x0000 FFFF
L1PMPPA18	RW	32	0x0004 A648	0x0184 A648	0x0000 FFFF
L1PMPPA19	RW	32	0x0004 A64C	0x0184 A64C	0x0000 FFFF
L1PMPPA20	RW	32	0x0004 A650	0x0184 A650	0x0000 FFFF
L1PMPPA21	RW	32	0x0004 A654	0x0184 A654	0x0000 FFFF
L1PMPPA22	RW	32	0x0004 A658	0x0184 A658	0x0000 FFFF
L1PMPPA23	RW	32	0x0004 A65C	0x0184 A65C	0x0000 FFFF
L1PMPPA24	RW	32	0x0004 A660	0x0184 A660	0x0000 FFFF
L1PMPPA25	RW	32	0x0004 A664	0x0184 A664	0x0000 FFFF
L1PMPPA26	RW	32	0x0004 A668	0x0184 A668	0x0000 FFFF
L1PMPPA27	RW	32	0x0004 A66C	0x0184 A66C	0x0000 FFFF
L1PMPPA28	RW	32	0x0004 A670	0x0184 A670	0x0000 FFFF
L1PMPPA29	RW	32	0x0004 A674	0x0184 A674	0x0000 FFFF
L1PMPPA30	RW	32	0x0004 A678	0x0184 A678	0x0000 FFFF
L1PMPPA31	RW	32	0x0004 A67C	0x0184 A67C	0x0000 FFFF
L1DMPFAR	R	32	0x0004 AC00	0x0184 AC00	0x0000 0000
L1DMPFSR	RW	32	0x0004 AC04	0x0184 AC04	0x0000 0000
L1DMPFCR	RW	32	0x0004 AC08	0x0184 AC08	0x0000 0000
MPLK0	W	32	0x0004 AD00	0x0184 AD00	0x0000 0000
MPLK1	W	32	0x0004 AD04	0x0184 AD04	0x0000 0000
MPLK2	W	32	0x0004 AD08	0x0184 AD08	0x0000 0000
MPLK3	W	32	0x0004 AD0C	0x0184 AD0C	0x0000 0000
MPLKCMD	RW	32	0x0004 AD10	0x0184 AD10	0x0000 0000
MPLKSTAT	RW	32	0x0004 AD14	0x0184 AD14	0x0000 0002
L1DMPPA16	RW	32	0x0004 AE40	0x0184 AE40	0x0000 FFF6
L1DMPPA17	RW	32	0x0004 AE44	0x0184 AE44	0x0000 FFF6
L1DMPPA18	RW	32	0x0004 AE48	0x0184 AE48	0x0000 FFF6
L1DMPPA19	RW	32	0x0004 AE4C	0x0184 AE4C	0x0000 FFF6
L1DMPPA20	RW	32	0x0004 AE50	0x0184 AE50	0x0000 FFF6
L1DMPPA21	RW	32	0x0004 AE54	0x0184 AE54	0x0000 FFF6
L1DMPPA22	RW	32	0x0004 AE58	0x0184 AE58	0x0000 FFF6
L1DMPPA23	RW	32	0x0004 AE5C	0x0184 AE5C	0x0000 FFF6
L1DMPPA24	RW	32	0x0004 AE60	0x0184 AE60	0x0000 FFF6
L1DMPPA25	RW	32	0x0004 AE64	0x0184 AE64	0x0000 FFF6
L1DMPPA26	RW	32	0x0004 AE68	0x0184 AE68	0x0000 FFF6
L1DMPPA27	RW	32	0x0004 AE6C	0x0184 AE6C	0x0000 FFF6
L1DMPPA28	RW	32	0x0004 AE70	0x0184 AE70	0x0000 FFF6
L1DMPPA29	RW	32	0x0004 AE74	0x0184 AE74	0x0000 FFF6
L1DMPPA30	RW	32	0x0004 AE78	0x0184 AE78	0x0000 FFF6
L1DMPPA31	RW	32	0x0004 AE7C	0x0184 AE7C	0x0000 FFF6

4.4.2.2 DSP_ICFG Register Description

For bitfield descriptions of all registers which reside within the DSP_ICFG configuration address space, refer to the *TMS320C66x DSP CorePac User Guide*, (SPRUGW0C).

4.4.3 DSP_SYSTEM Registers

4.4.3.1 DSP_SYSTEM Register Summary

NOTE: While the DSP1_SYSTEM and DSP2_SYSTEM addresses are part of the device L3_MAIN memory space, **the DSP_SYSTEM addresses are visible only within the DSP_ICFG internal configuration space (visible only to C66x CPU and debug logic).**

Table 4-14. DSP_SYSTEM and DSP1_SYSTEM Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	DSP_SYSTEM Physical Address DSP1 and DSP2 private	DSP1_SYSTEM Physical Address L3_MAIN Interconnect
DSP_SYS_REVISION	R	32	0x0000 0000	0x01D0 0000	0x40D0 0000
DSP_SYS_HWINFO	R	32	0x0000 0004	0x01D0 0004	0x40D0 0004
DSP_SYS_SYSCONFIG	RW	32	0x0000 0008	0x01D0 0008	0x40D0 0008
DSP_SYS_STAT	R	32	0x0000 000C	0x01D0 000C	0x40D0 000C
DSP_SYS_DISC_CONFIG	RW	32	0x0000 0010	0x01D0 0010	0x40D0 0010
DSP_SYS_BUS_CONFIG	RW	32	0x0000 0014	0x01D0 0014	0x40D0 0014
DSP_SYS_MMU_CONFIG	RW	32	0x0000 0018	0x01D0 0018	0x40D0 0018
DSP_SYS_IRQWAKEEN0	RW	32	0x0000 0020	0x01D0 0020	0x40D0 0020
DSP_SYS_IRQWAKEEN1	RW	32	0x0000 0024	0x01D0 0024	0x40D0 0024
DSP_SYS_DMAWAKEEN0	RW	32	0x0000 0030	0x01D0 0030	0x40D0 0030
DSP_SYS_DMAWAKEEN1	RW	32	0x0000 0034	0x01D0 0034	0x40D0 0034
DSP_SYS_EVTOUT_SET	RW	32	0x0000 0040	0x01D0 0040	0x40D0 0040
DSP_SYS_EVTOUT_CLR	RW	32	0x0000 0044	0x01D0 0044	0x40D0 0044
RESERVED	R	32	0x0000 0048	0x01D0 0048	0x40D0 0048
DSP_SYS_ERRINT_IRQSTATUS_RAW	RW	32	0x0000 0050	0x01D0 0050	0x40D0 0050
DSP_SYS_ERRINT_IRQSTATUS	RW	32	0x0000 0054	0x01D0 0054	0x40D0 0054
DSP_SYS_ERRINT_IRQENABLE_SET	RW	32	0x0000 0058	0x01D0 0058	0x40D0 0058
DSP_SYS_ERRINT_IRQENABLE_CLR	RW	32	0x0000 005C	0x01D0 005C	0x40D0 005C
DSP_SYS_EDMAWAKE0_IRQSTATUS_RAW	RW	32	0x0000 0060	0x01D0 0060	0x40D0 0060
DSP_SYS_EDMAWAKE0_IRQSTATUS	RW	32	0x0000 0064	0x01D0 0064	0x40D0 0064
DSP_SYS_EDMAWAKE0_IRQENABLE_SET	RW	32	0x0000 0068	0x01D0 0068	0x40D0 0068
DSP_SYS_EDMAWAKE0_IRQENABLE_CLR	RW	32	0x0000 006C	0x01D0 006C	0x40D0 006C
DSP_SYS_EDMAWAKE1_IRQSTATUS_RAW	RW	32	0x0000 0070	0x01D0 0070	0x40D0 0070
DSP_SYS_EDMAWAKE1_IRQSTATUS	RW	32	0x0000 0074	0x01D0 0074	0x40D0 0074
DSP_SYS_EDMAWAKE1_IRQENABLE_SET	RW	32	0x0000 0078	0x01D0 0078	0x40D0 0078
DSP_SYS_EDMAWAKE1_IRQENABLE_CLR	RW	32	0x0000 007C	0x01D0 007C	0x40D0 007C
RESERVED	R	32	0x0000 00E0	0x01D0 00E0	0x40D0 00E0
RESERVED	R	32	0x0000 00E4	0x01D0 00E4	0x40D0 00E4
DSP_SYS_HW_DBGOUT_SEL	RW	32	0x0000 00F8	0x01D0 00F8	0x40D0 00F8
DSP_SYS_HW_DBGOUT_VAL	R	32	0x0000 00FC	0x01D0 00FC	0x40D0 00FC

Table 4-15. DSP2_SYSTEM Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	DSP2_SYSTEM Physical Address L3_MAIN Interconnect
DSP_SYS_REVISION	R	32	0x0000 0000	0x4150 0000
DSP_SYS_HWINFO	R	32	0x0000 0004	0x4150 0004
DSP_SYS_SYSCONFIG	RW	32	0x0000 0008	0x4150 0008
DSP_SYS_STAT	R	32	0x0000 000C	0x4150 000C
DSP_SYS_DISC_CONFIG	RW	32	0x0000 0010	0x4150 0010
DSP_SYS_BUS_CONFIG	RW	32	0x0000 0014	0x4150 0014
DSP_SYS_MMU_CONFIG	RW	32	0x0000 0018	0x4150 0018
DSP_SYS_IRQWAKEEN0	RW	32	0x0000 0020	0x4150 0020
DSP_SYS_IRQWAKEEN1	RW	32	0x0000 0024	0x4150 0024
DSP_SYS_DMAWAKEEN0	RW	32	0x0000 0030	0x4150 0030
DSP_SYS_DMAWAKEEN1	RW	32	0x0000 0034	0x4150 0034
DSP_SYS_EVTOUT_SET	RW	32	0x0000 0040	0x4150 0040
DSP_SYS_EVTOUT_CLR	RW	32	0x0000 0044	0x4150 0044
RESERVED	R	32	0x0000 0048	0x4150 0048
DSP_SYS_ERRINT_IRQSTATUS_RAW	RW	32	0x0000 0050	0x4150 0050
DSP_SYS_ERRINT_IRQSTATUS	RW	32	0x0000 0054	0x4150 0054
DSP_SYS_ERRINT_IRQENABLE_SET	RW	32	0x0000 0058	0x4150 0058
DSP_SYS_ERRINT_IRQENABLE_CLR	RW	32	0x0000 005C	0x4150 005C
DSP_SYS_EDMAWAKE0_IRQSTATUS_RAW	RW	32	0x0000 0060	0x4150 0060
DSP_SYS_EDMAWAKE0_IRQSTATUS	RW	32	0x0000 0064	0x4150 0064
DSP_SYS_EDMAWAKE0_IRQENABLE_SET	RW	32	0x0000 0068	0x4150 0068
DSP_SYS_EDMAWAKE0_IRQENABLE_CLR	RW	32	0x0000 006C	0x4150 006C
DSP_SYS_EDMAWAKE1_IRQSTATUS_RAW	RW	32	0x0000 0070	0x4150 0070
DSP_SYS_EDMAWAKE1_IRQSTATUS	RW	32	0x0000 0074	0x4150 0074
DSP_SYS_EDMAWAKE1_IRQENABLE_SET	RW	32	0x0000 0078	0x4150 0078
DSP_SYS_EDMAWAKE1_IRQENABLE_CLR	RW	32	0x0000 007C	0x4150 007C
RESERVED	R	32	0x0000 00E0	0x4150 00E0
RESERVED	R	32	0x0000 00E4	0x4150 00E4
DSP_SYS_HW_DBGOUT_SEL	RW	32	0x0000 00F8	0x4150 00F8
DSP_SYS_HW_DBGOUT_VAL	R	32	0x0000 00FC	0x4150 00FC

4.4.3.2 DSP_SYSTEM Register Description

Table 4-16. DSP_SYS_REVISION

Address Offset	0x0000 0000		
Physical Address	0x01D0 0000 0x40D0 0000 0x4150 0000	**Instance**	DSP_SYSTEM DSP1_SYSTEM DSP2_SYSTEM
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REVISION																															

Bits	Field Name	Description	Type	Reset
31:0	REVISION	IP Revision	R	0x-(1)

(1) TI Internal Data

Table 4-17. Register Call Summary for Register DSP_SYS_REVISION

DSP Subsystem Register Manual
- DSP_SYSTEM Register Summary: [0][1]

Table 4-18. DSP_SYS_HWINFO

Address Offset	0x0000 0004		
Physical Address	0x01D0 0004 0x40D0 0004 0x4150 0004	**Instance**	DSP_SYSTEM DSP1_SYSTEM DSP2_SYSTEM
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INFO																												NUM			

Bits	Field Name	Description	Type	Reset
31:4	INFO	0x0: No configurable options in subsystem.	R	0x0
3:0	NUM	Instance Number Set by subsystem input. In a multi-DSP system, provides a unique/incrementing values for each DSP.	R	0x0

Table 4-19. Register Call Summary for Register DSP_SYS_HWINFO

DSP Subsystem Register Manual
- DSP_SYSTEM Register Summary: [0][1]

Table 4-20. DSP_SYS_SYSCONFIG

Address Offset	0x0000 0008		
Physical Address	0x01D0 0008 0x40D0 0008 0x4150 0008	**Instance**	DSP_SYSTEM DSP1_SYSTEM DSP2_SYSTEM
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							RESERVED	RESERVED		STANDBYMODE		IDLEMODE		RESERVED	

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reserved. Read returns 0.	R	0x00 0000
8	RESERVED	Reserved. User must write 0.	RW	0x0
7:6	RESERVED	Reserved. Read returns 0.	R	0x0

Bits	Field Name	Description	Type	Reset
5:4	STANDBYMODE	0x0: FORCE_STANDBY This mode is a backup mode intended to be used only if the smart-idle mode is bugged. When in this mode, the SAF asserts with minimal hardware condition the "STANDBY" status. It is the responsibility of the software to ensure that the SAF is in a correct quiet state before programming this mode. Additionally when in this mode, the SAF is not allowed to generate wakeup request. 0x1: NO_STANDBY This mode is a backup mode intended to be used only if the smart-idle mode is bugged. When in this mode, the SAF in C66xOSS asserts the "STANDBY" status. 0x2: SMART_STANDBY default. C66xOSS generates the standby status based upon all hardware internal status, namely after having performed all hardware operations necessary to be in a correct quiet state. Additionally when in this mode, the SAF is not allowed to generate wakeup request. 0x3: SMART_STANDBY_WKUP Same as Smart-Standby. (C66xOSS generates the standby status based upon all hardware internal status, namely after having performed all hardware operations necessary to be in a correct quiet state). . Additionally when in this mode, the SAF is allowed to generate wakeup request	RW	0x2
3:2	IDLEMODE	0x0: FORCE_IDLE This mode is a backup mode intended to be used only if the smart-idle mode is bugged. When in this mode, the IAF acknowledges a request to go idle from the power manager with minimal hardware condition. It is the responsibility of the software to ensure that the IAF are in a correct quiet state before requesting a force-idle transition. Additionally when in this mode, the IAF is not allowed to generate any wakeup request. 0x1: NO_IDLE When in this mode, the IAF disregards any request to go idle from the power manager. 0x2: SMART_IDLE default. When in this mode, the IAF acknowledges a request to go idle from the power manager after having performed all hardware operations necessary to be in a correct quiet state. Additionally when in this mode, the IAF is not allowed to generate any wakeup request 0x3: SMARTIDLEWKUP When in this mode, the IAF acknowledges a request to go idle from the power manager after having performed all hardware operations necessary for the IAF to be in a correct quiet state. Additionally when in this mode, the IAF is allowed to generate wakeup request.	RW	0x2
1:0	RESERVED	Reserved	R	0x0

Table 4-21. Register Call Summary for Register DSP_SYS_SYSCONFIG

DSP Subsystems Functional Description
- DSP Power Management: [0][1][2]
- DSP Input Interrupts: [3]

DSP Subsystem Register Manual
- DSP_SYSTEM Register Summary: [4][5]

Table 4-22. DSP_SYS_STAT

Address Offset	0x0000 000C		
Physical Address	0x01D0 000C 0x40D0 000C 0x4150 000C	**Instance**	DSP_SYSTEM DSP1_SYSTEM DSP2_SYSTEM
Description	This register is intended to provide indication to software (including a remote host) as to whether the DSP is able to enter a low power mode.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																										OCPI_DISC_STAT		RESERVED	TC1_STAT	TC0_STAT	C66X_STAT

Bits	Field Name	Description	Type	Reset
31:6	RESERVED		R	0x000 0000
5:4	OCPI_DISC_STAT	L3_MAIN (OCP) Initiator(s) Disconnect Status Read 0x0 : OCP inititiator ports are disconnected Read 0x1 : OCP initiator ports are attempting to disconnect. Read 0x2 : OCP initiator ports are active, no request to disconnect is pending.	R	0x2
3	RESERVED		R	0
2	TC1_STAT	EDMA TC1 Status 0x0: IDLE 0x1: ACTIVE - Active, based on inverse of tptc1_mstandby	R	1
1	TC0_STAT	EDMA TC0 Status 0x0: IDLE 0x1: ACTIVE - Active, based on inverse of tptc0_mstandby	R	1
0	C66X_STAT	C66x Status 0x0: IDLE C66x core is idle 0x1: ACTIVE C66x core is active.	R	1

Table 4-23. Register Call Summary for Register DSP_SYS_STAT

DSP Subsystems Functional Description

- DSP System and Device Level Settings of DSP EDMA: [0][1]

DSP Subsystem Register Manual

- DSP_SYSTEM Register Summary: [2][3]

Table 4-24. DSP_SYS_DISC_CONFIG

Address Offset	0x0000 0010		
Physical Address	0x01D0 0010 0x40D0 0010 0x4150 0010	**Instance**	DSP_SYSTEM DSP1_SYSTEM DSP2_SYSTEM
Description	This register is used to manually disconnect the OCP busses.		

Table 4-24. DSP_SYS_DISC_CONFIG (continued)

Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															OCPI_DISC

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	OCPI_DISC	OCP Initiator (on L3_MAIN) Disconnect request Read 0: Disconnect not in progress, or has completed. Write 0: No effect. Read 1: Disconnect request is in progress. Write 1: Request for OCP Initiator to disconnect and mask write byte enable signals.	RW	0

Table 4-25. Register Call Summary for Register DSP_SYS_DISC_CONFIG

DSP Subsystem Register Manual
- DSP_SYSTEM Register Summary: [0][1]

Table 4-26. DSP_SYS_BUS_CONFIG

Address Offset	0x0000 0014		
Physical Address	0x01D0 0014 0x40D0 0014 0x4150 0014	**Instance**	DSP_SYSTEM DSP1_SYSTEM DSP2_SYSTEM
Description	This register controls the burst and priority settings for the internal initiators.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	SDMA_PRI			RESERVED			NOPOSTOVERRIDE	RESERVED		SDMA_L2PRES		RESERVED		CFG_L2PRES		RESERVED		TC1_L2PRES		RESERVED		TC0_L2PRES		RESERVED		TC1_DBS		RESERVED		TC0_DBS	

Bits	Field Name	Description	Type	Reset
31	RESERVED		R	0
30:28	SDMA_PRI	Sets the CBA/VBusM Priority for the DSP C66x CorePac SDMA port. Can typically be left at default value. 0x0 is highest, ..., 0x7 is lowest priority.	RW	0x4
27:25	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
24	NOPOSTOVERRIDE	OCP Posted Write vs Non-Posted Write override 0x0: MIX Posted writes are used for cacheable write transactions. Non-posted writes are used for non-cacheable write transactions. 0x1: NOPOST Non-posted writes are used exclusively.	RW	1
23:22	RESERVED		R	0x0
21:20	SDMA_L2PRES	OCP Target port L2 Interconnect Pressure. Driven on ocp mflag to control arbitration within the L2 interconnect. 0x0: LOW - Lowest pressure 0x1: MED - Medium pressure 0x2: Reserved 0x3: HIGH - High pressure	RW	0x0
19:18	RESERVED		R	0x0
17:16	CFG_L2PRES	DSP CFG L2 Interconnect Pressure. Driven on ocp mflag to control arbitration within the L2 interconnect 0x0: LOW - Lowest pressure 0x1: MED - Medium pressure 0x2: Reserved 0x3: HIGH - High pressure	RW	0x0
15:14	RESERVED		R	0x0
13:12	TC1_L2PRES	TC1 L2 Interconnect Pressure. Driven on ocp mflag to control arbitration within the L2 interconnect 0x0: LOW - Lowest pressure 0x1: MED - Medium pressure 0x2: Reserved 0x3: HIGH - High pressure	RW	0x0
11:10	RESERVED		R	0x0
9:8	TC0_L2PRES	TC0 L2 Interconnect Pressure. Driven on ocp mflag to control arbitration within the L2 interconnect. 0x0: LOW - Lowest pressure 0x1: MED - Medium pressure 0x2: Reserved 0x3: HIGH - High pressure	RW	0x0
7:6	RESERVED		R	0x0
5:4	TC1_DBS	TC1 Default Burst size. 0x0: BYTE_16 - "16-Byte" bursts 0x1: BYTE_32 - "32-Byte" bursts 0x2: BYTE_64 - "64-Byte" bursts 0x3: BYTE_128 - "128-Byte" bursts	RW	0x3
3:2	RESERVED		R	0x0
1:0	TC0_DBS	TC0 Default Burst size 0x0: BYTE_16 - "16-Byte" bursts 0x1: BYTE_32 - "32-Byte" bursts 0x2: BYTE_64 - "64-Byte" bursts 0x3: BYTE_128 - "128-Byte" bursts	RW	0x3

Table 4-27. Register Call Summary for Register DSP_SYS_BUS_CONFIG

DSP Subsystems Functional Description • DSP TMS320C66x CorePac: [0] • DSP System and Device Level Settings of DSP EDMA: [1][2] • DSP NoC Arbitration: [3]
DSP Subsystem Register Manual • DSP_SYSTEM Register Summary: [4][5]

Table 4-28. DSP_SYS_MMU_CONFIG

Address Offset	0x0000 0018		
Physical Address	0x01D0 0018 0x40D0 0018 0x4150 0018	**Instance**	DSP_SYSTEM DSP1_SYSTEM DSP2_SYSTEM
Description	This register is used to enable the subsystem MMUs.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																			MMU1_ABORT	RESERVED			MMU0_ABORT	RESERVED			MMU1_EN	RESERVED			MMU0_EN

Bits	Field Name	Description	Type	Reset
31:13	RESERVED		R	0x0
12	MMU1_ABORT	MMU1 Abort 0x0: NOABORT = Abort not requested. 0x1: ABORT = MMU abort requested. Can be used in case of page translation failure or lockup.	RW	0
11:9	RESERVED		R	0x0
8	MMU0_ABORT	MMU0 Abort 0x0: NOABORT = Abort not requested. 0x1: ABORT = MMU abort requested. Can be used in case of page translation failure or lockup.	RW	0
7:5	RESERVED		R	0x0
4	MMU1_EN	MMU1 Enable 0x0: DISABLED = MMU is disabled and the MMU IP is bypassed. 0x1: ENABLED = MMU is enabled. (The MMU mmrs (including Enable bit) need to be set in addition to this bit.)	RW	0
3:1	RESERVED		R	0x0
0	MMU0_EN	MMU0 Enable 0x0: DISABLED = MMU is disabled and the MMU IP is bypassed. 0x1: ENABLED = MMU is enabled. (The MMU mmrs (including Enable bit) need to be set in addition to this bit.)	RW	0

Table 4-29. Register Call Summary for Register DSP_SYS_MMU_CONFIG

DSP Subsystems Functional Description
- Routing MDMA Traffic through DSP MMU0: [0][1]
- Routing EDMA Traffic thorugh DSP MMU1: [2][3]

DSP Subsystem Register Manual
- DSP_SYSTEM Register Summary: [4][5]

Table 4-30. DSP_SYS_IRQWAKEEN0

Address Offset	0x0000 0020		
Physical Address	0x01D0 0020 0x40D0 0020 0x4150 0020	**Instance**	DSP_SYSTEM DSP1_SYSTEM DSP2_SYSTEM
Description	The register provides a global interrupt wakeup enable bit vector that defines which input interrupts are used to cause a wake from powerdown state (via the slave idle protocol). IRQWAKEEN0 is for interrupt inputs 63 thru 32, and IRQWAKEEN1 is for interrupt inputs 95 thru 64. Internal interrupts cannot cause a wake condition since in this state there is no guaranteed clock and all sub-modules should be in idle state. Software can write 1 to set and 0 to clear the corresponding bit (i.e., enable the corresponding interrupt for wakeup). Reads of this register return the actual state of the enable register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ENABLE																															

Bits	Field Name	Description	Type	Reset
31:0	ENABLE	Wakeup Enable bit vector for interrupt #n+32 0x0: DISABLE = Interrupt #n+32 disabled for wakeup 0x1: ENABLE = Interrupt #n+32 enabled for wakeup	RW	0x0

Table 4-31. Register Call Summary for Register DSP_SYS_IRQWAKEEN0

DSP Subsystems Functional Description
- DSP Power Management: [0][1][2][3]
- DSP Input Interrupts: [4][5]

DSP Subsystem Register Manual
- DSP_SYSTEM Register Summary: [6][7]

Table 4-32. DSP_SYS_IRQWAKEEN1

Address Offset	0x0000 0024		
Physical Address	0x01D0 0024 0x40D0 0024 0x4150 0024	**Instance**	DSP_SYSTEM DSP1_SYSTEM DSP2_SYSTEM
Description	The register provides a global interrupt wakeup enable bit vector that defines which input interrupts are used to cause a wake from powerdown state (via the slave idle protocol). IRQWAKEEN0 is for interrupt inputs 63 thru 32, and IRQWAKEEN1 is for interrupt inputs 95 thru 64. Internal interrupts cannot cause a wake condition since in this state there is no guaranteed clock and all sub-modules should be in idle state. Software can write 1 to set and 0 to clear the corresponding bit (i.e., enable the corresponding interrupt for wakeup). Reads of this register return the actual state of the enable register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ENABLE																															

Bits	Field Name	Description	Type	Reset
31:0	ENABLE	Wakeup Enable bit vector for interrupt #n+64 0x0: DISABLE = Interrupt #n+64 disabled for wakeup 0x1: ENABLE = Interrupt #n+64 enabled for wakeup	RW	0x0

Table 4-33. Register Call Summary for Register DSP_SYS_IRQWAKEEN1

DSP Subsystems Functional Description
- DSP Power Management: [0][1][2]
- DSP Input Interrupts: [3][4]

DSP Subsystem Register Manual
- DSP_SYSTEM Register Summary: [5][6]

Table 4-34. DSP_SYS_DMAWAKEEN0

Address Offset	0x0000 0030		
Physical Address	0x01D0 0030 0x40D0 0030 0x4150 0030	**Instance**	DSP_SYSTEM DSP1_SYSTEM DSP2_SYSTEM
Description	The register provides a global dma event wakeup enable bit vector that defines which input dma events are used to cause a wake from powerdown state (via the slave idle protocol). DMAWAKEEN0 is for dma event inputs 31 thru 0, and DMAWAKEEN1 is for dma event inputs 63 thru 32. Software can write 1 to set and 0 to clear the corresponding bit (i.e., enable the corresponding dma event for wakeup). Reads of this register return the actual state of the enable register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ENABLE																															

Bits	Field Name	Description	Type	Reset
31:0	ENABLE	Wakeup Enable for event #n 0x0: DISABLE = Interrupt #n disabled for wakeup 0x1: ENABLE = Interrupt #n enabled for wakeup	RW	0x0

Table 4-35. Register Call Summary for Register DSP_SYS_DMAWAKEEN0

DSP Subsystems Functional Description
- DSP Power Management: [0][1]
- DSP DMA Requests: [2][3]
- DSP EDMA Wakeup Interrupt: [4]

DSP Subsystem Register Manual
- DSP_SYSTEM Register Summary: [5][6]

Table 4-36. DSP_SYS_DMAWAKEEN1

Address Offset	0x0000 0034		
Physical Address	0x01D0 0034 0x40D0 0034 0x4150 0034	**Instance**	DSP_SYSTEM DSP1_SYSTEM DSP2_SYSTEM
Description	The register provides a global dma event wakeup enable bit vector that defines which input dma events are used to cause a wake from powerdown state (via the slave idle protocol). DMAWAKEEN0 is for dma event inputs 31 thru 0, and DMAWAKEEN1 is for dma event inputs 63 thru 32. Software can write 1 to set and 0 to clear the corresponding bit (i.e., enable the corresponding dma event for wakeup). Reads of this register return the actual state of the enable register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ENABLE																															

Bits	Field Name	Description	Type	Reset
31:0	ENABLE	Wakeup Enable for event #n+32 0x0: DISABLE = Interrupt #n+32 disabled for wakeup 0x1: ENABLE = Interrupt #n+32 enabled for wakeup	RW	0x0

Table 4-37. Register Call Summary for Register DSP_SYS_DMAWAKEEN1

DSP Subsystems Functional Description
- DSP Power Management: [0]
- DSP DMA Requests: [1]
- DSP EDMA Wakeup Interrupt: [2]

DSP Subsystem Register Manual
- DSP_SYSTEM Register Summary: [3][4]

Table 4-38. DSP_SYS_EVTOUT_SET

Address Offset	0x0000 0040		
Physical Address	0x01D0 0040 0x40D0 0040 0x4150 0040	**Instance**	DSP_SYSTEM DSP1_SYSTEM DSP2_SYSTEM
Description	These registers can be used to drive event outputs from the DSP subsystem to a desired state.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EVENT																															

Bits	Field Name	Description	Type	Reset
31:0	EVENT	Output Event for event #n Write 0x00 0001: Drive output event #n high/1. Read 0x00 0000: Event #n is low/0. Read 0x00 0001 : Event #n is high/1. Write 0x00 0000 : No action.	RW	0x0

Table 4-39. Register Call Summary for Register DSP_SYS_EVTOUT_SET

DSP Subsystems Functional Description
- DSP Event and Interrupt Generation Outputs: [0][1]

DSP Subsystem Register Manual
- DSP_SYSTEM Register Summary: [2][3]

Table 4-40. DSP_SYS_EVTOUT_CLR

Address Offset	0x0000 0044		
Physical Address	0x01D0 0044 0x40D0 0044 0x4150 0044	**Instance**	DSP_SYSTEM DSP1_SYSTEM DSP2_SYSTEM
Description	These registers can be used to drive event outputs from the DSP subsystem to a desired state.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EVENT																															

Bits	Field Name	Description	Type	Reset
31:0	EVENT	Output Event for event #n Read 0x00 0000: Event #n is low/0. Write 0x00 0001: Drive output event #n low/0. Read 0x00 0001 : Event #n is high/1. Write 0x00 0000 : No action.	RW	0x0

Table 4-41. Register Call Summary for Register DSP_SYS_EVTOUT_CLR

DSP Subsystems Functional Description • DSP Event and Interrupt Generation Outputs: [0][1]
DSP Subsystem Register Manual • DSP_SYSTEM Register Summary: [2][3]

Table 4-42. DSP_SYS_ERRINT_IRQSTATUS_RAW

Address Offset	0x0000 0050		
Physical Address	0x01D0 0050 0x40D0 0050 0x4150 0050	**Instance**	DSP_SYSTEM DSP1_SYSTEM DSP2_SYSTEM
Description	This register provides a per-event raw interrupt status vector		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED													EVENT																		

Bits	Field Name	Description	Type	Reset
31:19	RESERVED		R	0x0
18:0	EVENT	Settable raw status for event #n Read 0x00 0000 : No event pending Write 0x00 0000 : No action Read 0x00 0001: Event pending Write 0x00 0001: Set event (for debug)	RW	0x0

Table 4-43. Register Call Summary for Register DSP_SYS_ERRINT_IRQSTATUS_RAW

DSP Subsystems Functional Description • DSP Event and Interrupt Generation Outputs: [0][1][2]
DSP Subsystem Register Manual • DSP_SYSTEM Register Summary: [3][4]

Table 4-44. DSP_SYS_ERRINT_IRQSTATUS

Address Offset	0x0000 0054		
Physical Address	0x01D0 0054 0x40D0 0054 0x4150 0054	**Instance**	DSP_SYSTEM DSP1_SYSTEM DSP2_SYSTEM
Description	This register provides a per-event enabled interrupt status vector.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																EVENT															

Bits	Field Name	Description	Type	Reset
31:19	RESERVED		R	0x0
18:0	EVENT	Clearable, enabled status for event #n Read 0x00 0000 : No enabled event pending Write 0x00 0000 : No action Read 0x00 0001 : Enabled Event pending Write 0x00 0001 : Clear raw event	RW	0x0

Table 4-45. Register Call Summary for Register DSP_SYS_ERRINT_IRQSTATUS

DSP Subsystems Functional Description • DSP Event and Interrupt Generation Outputs: [0][1]
DSP Subsystem Register Manual • DSP_SYSTEM Register Summary: [2][3]

Table 4-46. DSP_SYS_ERRINT_IRQENABLE_SET

Address Offset	0x0000 0058		
Physical Address	0x01D0 0058 0x40D0 0058 0x4150 0058	**Instance**	DSP_SYSTEM DSP1_SYSTEM DSP2_SYSTEM
Description	This register provides a per-event interrupt enable bit vector.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																ENABLE															

Bits	Field Name	Description	Type	Reset
31:19	RESERVED		R	0x0
18:0	ENABLE	Enable for event #n Read 0x00 0000 : Interrupt disabled Write 0x00 0000 : No action Read 0x00 0001 : Interrupt enabled Write 0x00 0001 : Enable interrupt	RW	0x0

Table 4-47. Register Call Summary for Register DSP_SYS_ERRINT_IRQENABLE_SET

DSP Subsystems Functional Description • DSP Event and Interrupt Generation Outputs: [0][1][2]
DSP Subsystem Register Manual • DSP_SYSTEM Register Summary: [3][4]

Table 4-48. DSP_SYS_ERRINT_IRQENABLE_CLR

Address Offset	0x0000 005C		
Physical Address	0x01D0 005C 0x40D0 005C 0x4150 005C	**Instance**	DSP_SYSTEM DSP1_SYSTEM DSP2_SYSTEM
Description	This register provides a per-event interrupt enable bit vector.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								ENABLE																							

Bits	Field Name	Description	Type	Reset
31:19	RESERVED		R	0x0
18:0	ENABLE	Enable for event #n Read 0x00 0000 : Interrupt disabled Write 0x00 0000 : No action Read 0x00 0001 : Interrupt enabled Write 0x00 0001 : Disable interrupt (i.e., clear ENABLEn bit)	RW	0x0

Table 4-49. Register Call Summary for Register DSP_SYS_ERRINT_IRQENABLE_CLR

DSP Subsystems Functional Description • DSP Event and Interrupt Generation Outputs: [0][1]
DSP Subsystem Register Manual • DSP_SYSTEM Register Summary: [2][3]

Table 4-50. DSP_SYS_EDMAWAKE0_IRQSTATUS_RAW

Address Offset	0x0000 0060		
Physical Address	0x01D0 0060 0x40D0 0060 0x4150 0060	**Instance**	DSP_SYSTEM DSP1_SYSTEM DSP2_SYSTEM
Description	This register provides a per-event raw interrupt status vector		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EVENT																															

Bits	Field Name	Description	Type	Reset
31:0	EVENT	Settable raw status for event #n Read 0x0000 0000 : No event pending Write 0x0000 0001 : Set event (for debug) Read 0x0000 0001 : Event pending Write 0x0000 0000 : No action	RW	0x0

Table 4-51. Register Call Summary for Register DSP_SYS_EDMAWAKE0_IRQSTATUS_RAW

DSP Subsystems Functional Description • DSP Power Management: [0] • DSP EDMA Wakeup Interrupt: [1][2]
DSP Subsystem Register Manual • DSP_SYSTEM Register Summary: [3][4]

Table 4-52. DSP_SYS_EDMAWAKE0_IRQSTATUS

Address Offset	0x0000 0064		
Physical Address	0x01D0 0064 0x40D0 0064 0x4150 0064	**Instance**	DSP_SYSTEM DSP1_SYSTEM DSP2_SYSTEM
Description	This register provides a per-event enabled interrupt status vector.		

Table 4-52. DSP_SYS_EDMAWAKE0_IRQSTATUS (continued)

Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EVENT																															

Bits	Field Name	Description	Type	Reset
31:0	EVENT	Clearable, enabled status for event #n Read 0x0000 0000 : No enabled event pending Write 0x0000 0001 : Clear raw event Read 0x0000 0001 : Enabled Event pending Write 0x0000 0000 : No action	RW	0x0

Table 4-53. Register Call Summary for Register DSP_SYS_EDMAWAKE0_IRQSTATUS

DSP Subsystems Functional Description
- DSP Power Management: [0]
- DSP EDMA Wakeup Interrupt: [1]

DSP Subsystem Register Manual
- DSP_SYSTEM Register Summary: [2][3]

Table 4-54. DSP_SYS_EDMAWAKE0_IRQENABLE_SET

Address Offset	0x0000 0068		
Physical Address	0x01D0 0068 0x40D0 0068 0x4150 0068	**Instance**	DSP_SYSTEM DSP1_SYSTEM DSP2_SYSTEM
Description	This register provides a per-event interrupt enable bit vector.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ENABLE																															

Bits	Field Name	Description	Type	Reset
31:0	ENABLE	Enable for event #n Read 0x0000 0000: Interrupt disabled Write 0x0000 0001: Enable interrupt Read 0x0000 0001: Interrupt enabled Write 0x0000 0000: No action	RW	0x0000 0000

Table 4-55. Register Call Summary for Register DSP_SYS_EDMAWAKE0_IRQENABLE_SET

DSP Subsystems Functional Description
- DSP Power Management: [0]
- DSP EDMA Wakeup Interrupt: [1][2]

DSP Subsystem Register Manual
- DSP_SYSTEM Register Summary: [3][4]

Table 4-56. DSP_SYS_EDMAWAKE0_IRQENABLE_CLR

Address Offset	0x0000 006C		
Physical Address	0x01D0 006C 0x40D0 006C 0x4150 006C	**Instance**	DSP_SYSTEM DSP1_SYSTEM DSP2_SYSTEM
Description	This register provides a per-event interrupt enable bit vector.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ENABLE																															

Bits	Field Name	Description	Type	Reset
31:0	ENABLE	Enable for event #n Read 0x0000 0000: Interrupt disabled Write 0x0000 0001: Disable interrupt (i.e., clear ENABLEn bit) Read 0x0000 0001: Interrupt enabled Write 0x0000 0000: No action	RW	0x0

Table 4-57. Register Call Summary for Register DSP_SYS_EDMAWAKE0_IRQENABLE_CLR

DSP Subsystems Functional Description
- DSP EDMA Wakeup Interrupt: [0]

DSP Subsystem Register Manual
- DSP_SYSTEM Register Summary: [1][2]

Table 4-58. DSP_SYS_EDMAWAKE1_IRQSTATUS_RAW

Address Offset	0x0000 0070		
Physical Address	0x01D0 0070 0x40D0 0070 0x4150 0070	**Instance**	DSP_SYSTEM DSP1_SYSTEM DSP2_SYSTEM
Description	This register provides a per-event raw interrupt status vector		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EVENT																															

Bits	Field Name	Description	Type	Reset
31:0	EVENT	Settable raw status for event #n+32 Read 0x0000 0000: No event pending Write 0x0000 0001: Set event (for debug) Read 0x0000 0001: Event pending Write 0x0000 0000 : No action	RW	0x0

Table 4-59. Register Call Summary for Register DSP_SYS_EDMAWAKE1_IRQSTATUS_RAW

DSP Subsystem Register Manual
- DSP_SYSTEM Register Summary: [0][1]

Table 4-60. DSP_SYS_EDMAWAKE1_IRQSTATUS

Address Offset	0x0000 0074		
Physical Address	0x01D0 0074 0x40D0 0074 0x4150 0074	**Instance**	DSP_SYSTEM DSP1_SYSTEM DSP2_SYSTEM
Description	This register provides a per-event enabled interrupt status vector.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EVENT																															

Bits	Field Name	Description	Type	Reset
31:0	EVENT	Clearable, enabled status for event #n+32 Read 0x0000 0000: No enabled event pending Write 0x0000 0001: Clear raw event Read 0x0000 0001: Enabled Event pending Write 0x0000 0000: No action	RW	0x0

Table 4-61. Register Call Summary for Register DSP_SYS_EDMAWAKE1_IRQSTATUS

DSP Subsystem Register Manual

- DSP_SYSTEM Register Summary: [0][1]

Table 4-62. DSP_SYS_EDMAWAKE1_IRQENABLE_SET

Address Offset	0x0000 0078		
Physical Address	0x01D0 0078 0x40D0 0078 0x4150 0078	**Instance**	DSP_SYSTEM DSP1_SYSTEM DSP2_SYSTEM
Description	This register provides a per-event interrupt enable bit vector.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ENABLE																															

Bits	Field Name	Description	Type	Reset
31:0	ENABLE	Enable for event #n+32 Read 0x0000 0000: Interrupt disabled Write 0x0000 0001: Enable interrupt Read 0x0000 0001: Interrupt enabled Write 0x0000 0000: No action	RW	0x0

Table 4-63. Register Call Summary for Register DSP_SYS_EDMAWAKE1_IRQENABLE_SET

DSP Subsystem Register Manual

- DSP_SYSTEM Register Summary: [0][1]

Table 4-64. DSP_SYS_EDMAWAKE1_IRQENABLE_CLR

Address Offset	0x0000 007C		
Physical Address	0x01D0 007C 0x40D0 007C 0x4150 007C	**Instance**	DSP_SYSTEM DSP1_SYSTEM DSP2_SYSTEM
Description	This register provides a per-event interrupt enable bit vector.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ENABLE																															

Bits	Field Name	Description	Type	Reset
31:0	ENABLE	Enable for event #n+32 Read 0x0000 0000: Interrupt disabled Write 0x0000 0001: Disable interrupt (i.e., clear ENABLEn bit) Read 0x0000 0001: Interrupt enabled Write 0x0000 0000 : No action	RW	0x0

Table 4-65. Register Call Summary for Register DSP_SYS_EDMAWAKE1_IRQENABLE_CLR

DSP Subsystem Register Manual
- DSP_SYSTEM Register Summary: [0][1]

Table 4-66. DSP_SYS_HW_DBGOUT_SEL

Address Offset	0x0000 00F8		
Physical Address	0x01D0 00F8 0x40D0 00F8 0x4150 00F8	**Instance**	DSP_SYSTEM DSP1_SYSTEM DSP2_SYSTEM
Description	THis register is used to select which group of internal signals are mapped to the hw_dbgout output bus.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												GROUP			

Bits	Field Name	Description	Type	Reset
31:4	RESERVED		R	0x0000000
3:0	GROUP	Debug Group output control mux select 0x0 : Disabled, debug outputs driven to 0x0. 0x1 : G1 = select output group 1 0x2 : G2 = select output group 2 N: GN = select output group N	RW	0x0

Table 4-67. Register Call Summary for Register DSP_SYS_HW_DBGOUT_SEL

DSP Subsystem Register Manual
- DSP_SYSTEM Register Summary: [0][1]

Table 4-68. DSP_SYS_HW_DBGOUT_VAL

Address Offset	0x0000 00FC		
Physical Address	0x01D0 00FC 0x40D0 00FC 0x4150 00FC	**Instance**	DSP_SYSTEM DSP1_SYSTEM DSP2_SYSTEM
Description	This register is used to read the value of the currently selected debug output group.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VALUE																															

Bits	Field Name	Description	Type	Reset
31:0	VALUE	Read returns state of hw_dbgout bus	R	0x0

Table 4-69. Register Call Summary for Register DSP_SYS_HW_DBGOUT_VAL

DSP Subsystem Register Manual

- DSP_SYSTEM Register Summary: [0][1]

4.4.4 DSP_FW_L2_NOC_CFG Registers

This section covers the DSPSS level defined public firewall (DSP_MMU0, DSP_MMU1) and L2 interconnect (DSP_NoC) functional registers.

4.4.4.1 DSP_FW_L2_NOC_CFG Register Summary

NOTE: While the DSP1_FW_L2_NOC_CFG and DSP2_FW_L2_NOC_CFG addresses are part of the device L3_MAIN memory space, **the DSP_FW_L2_NOC_CFG addresses are visible only within the DSP_ICFG internal configuration space (visible only to C66x CPU and debug logic).**

Table 4-70. DSP_FW_L2_NOC_CFG and DSP1_FW_L2_NOC_CFG Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	DSP_FW_L2_NOC_CFG DSP1 and DSP2 private	DSP1_FW_L2_NOC_CFG Physical Address L3_MAIN Interconnect
L3_DSPSS_INIT_OCP_MMU0_CTRL_TARG_OCP_FW_503000_ERROR_LOG_0	RW	32	0x0000 0000	0x01D0 3000	0x40D0 3000
L3_DSPSS_INIT_OCP_MMU0_CTRL_TARG_OCP_FW_503000_LOGICAL_ADDR_ERRLOG_0	R	32	0x0000 0004	0x01D0 3004	0x40D0 3004
L3_DSPSS_INIT_OCP_MMU0_CTRL_TARG_OCP_FW_503000_REGUPDATE_CONTROL	RW	32	0x0000 0040	0x01D0 3040	0x40D0 3040
L3_DSPSS_INIT_OCP_MMU0_CTRL_TARG_OCP_FW_503000_MRM_PERMISSION_REGION_LOW_0	RW	32	0x0000 0088	0x01D0 3088	0x40D0 3088
L3_DSPSS_INIT_OCP_MMU0_CTRL_TARG_OCP_FW_503000_MRM_PERMISSION_REGION_HIGH_0	RW	32	0x0000 008C	0x01D0 308C	0x40D0 308C
L3_DSPSS_INIT_OCP_MMU0_CTRL_TARG_OCP_FW_503000_START_REGION_1	RW	32	0x0000 0090	0x01D0 3090	0x40D0 3090
L3_DSPSS_INIT_OCP_MMU0_CTRL_TARG_OCP_FW_503000_END_REGION_1	RW	32	0x0000 0094	0x01D0 3094	0x40D0 3094
L3_DSPSS_INIT_OCP_MMU0_CTRL_TARG_OCP_FW_503000_MRM_PERMISSION_REGION_LOW_1	RW	32	0x0000 0098	0x01D0 3098	0x40D0 3098
L3_DSPSS_INIT_OCP_MMU0_CTRL_TARG_OCP_FW_503000_MRM_PERMISSION_REGION_HIGH_1	RW	32	0x0000 009C	0x01D0 309C	0x40D0 309C
L3_DSPSS_INIT_OCP_MMU1_CTRL_TARG_OCP_FW_504000_ERROR_LOG_0	RW	32	0x0000 1000	0x01D0 4000	0x40D0 4000

Table 4-70. DSP_FW_L2_NOC_CFG and DSP1_FW_L2_NOC_CFG Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	DSP_FW_L2_NOC_CFG DSP1 and DSP2 private	DSP1_FW_L2_NOC_CFG Physical Address L3_MAIN Interconnect
L3_DSPSS_INIT_OCP_MMU1_CTRL_TARG_OCP_FW_504000_LOGICAL_ADDR_ERRLOG_0	R	32	0x0000 1004	0x01D0 4004	0x40D0 4004
L3_DSPSS_INIT_OCP_MMU1_CTRL_TARG_OCP_FW_504000_REGUPDATE_CONTROL	RW	32	0x0000 1040	0x01D0 4040	0x40D0 4040
L3_DSPSS_INIT_OCP_MMU1_CTRL_TARG_OCP_FW_504000_MRM_PERMISSION_REGION_LOW_0	RW	32	0x0000 1088	0x01D0 4088	0x40D0 4088
L3_DSPSS_INIT_OCP_MMU1_CTRL_TARG_OCP_FW_504000_MRM_PERMISSION_REGION_HIGH_0	RW	32	0x0000 108C	0x01D0 408C	0x40D0 408C
DSPNOC_FLAGMUX_ID_COREID	R	32	0x0000 4000	0x01D0 7000	0x40D0 7000
DSPNOC_FLAGMUX_ID_REVISIONID	R	32	0x0000 4004	0x01D0 7004	0x40D0 7004
DSPNOC_FLAGMUX_FAULTEN	RW	32	0x0000 4008	0x01D0 7008	0x40D0 7008
DSPNOC_FLAGMUX_FAULTSTATUS	R	32	0x0000 400C	0x01D0 700C	0x40D0 700C
DSPNOC_FLAGMUX_FLAGINEN0	RW	32	0x0000 4010	0x01D0 7010	0x40D0 7010
DSPNOC_FLAGMUX_FLAGINSTATUS0	R	32	0x0000 4014	0x01D0 7014	0x40D0 7014
DSPNOC_ERRORLOG_ID_COREID	R	32	0x0000 4200	0x01D0 7200	0x40D0 7200
DSPNOC_ERRORLOG_ID_REVISIONID	R	32	0x0000 4204	0x01D0 7204	0x40D0 7204
DSPNOC_ERRORLOG_FAULTEN	RW	32	0x0000 4208	0x01D0 7208	0x40D0 7208
DSPNOC_ERRORLOG_ERRVLD	R	32	0x0000 420C	0x01D0 720C	0x40D0 720C
DSPNOC_ERRORLOG_ERRCLR	RW	32	0x0000 4210	0x01D0 7210	0x40D0 7210
DSPNOC_ERRORLOG_ERRLOG0	R	32	0x0000 4214	0x01D0 7214	0x40D0 7214
DSPNOC_ERRORLOG_ERRLOG1	R	32	0x0000 4218	0x01D0 7218	0x40D0 7218
DSPNOC_ERRORLOG_ERRLOG3	R	32	0x0000 4220	0x01D0 7220	0x40D0 7220
DSPNOC_ERRORLOG_ERRLOG5	R	32	0x0000 4228	0x01D0 7228	0x40D0 7228

Table 4-71. DSP2_FW_L2_NOC_CFG Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	DSP2_FW_L2_NOC_CFG Physical Address L3_MAIN Interconnect
L3_DSPSS_INIT_OCP_MMU0_CTRL_TARG_OCP_FW_503000_ERROR_LOG_0	RW	32	0x0000 0000	0x4150 3000
L3_DSPSS_INIT_OCP_MMU0_CTRL_TARG_OCP_FW_503000_LOGICAL_ADDR_ERRLOG_0	R	32	0x0000 0004	0x4150 3004
L3_DSPSS_INIT_OCP_MMU0_CTRL_TARG_OCP_FW_503000_REGUPDATE_CONTROL	RW	32	0x0000 0040	0x4150 3040
L3_DSPSS_INIT_OCP_MMU0_CTRL_TARG_OCP_FW_503000_MRM_PERMISSION_REGION_LOW_0	RW	32	0x0000 0088	0x4150 3088
L3_DSPSS_INIT_OCP_MMU0_CTRL_TARG_OCP_FW_503000_MRM_PERMISSION_REGION_HIGH_0	RW	32	0x0000 008C	0x4150 308C
L3_DSPSS_INIT_OCP_MMU0_CTRL_TARG_OCP_FW_503000_START_REGION_1	RW	32	0x0000 0090	0x4150 3090
L3_DSPSS_INIT_OCP_MMU0_CTRL_TARG_OCP_FW_503000_END_REGION_1	RW	32	0x0000 0094	0x4150 3094
L3_DSPSS_INIT_OCP_MMU0_CTRL_TARG_OCP_FW_503000_MRM_PERMISSION_REGION_LOW_1	RW	32	0x0000 0098	0x4150 3098
L3_DSPSS_INIT_OCP_MMU0_CTRL_TARG_OCP_FW_503000_MRM_PERMISSION_REGION_HIGH_1	RW	32	0x0000 009C	0x4150 309C
L3_DSPSS_INIT_OCP_MMU1_CTRL_TARG_OCP_FW_504000_ERROR_LOG_0	RW	32	0x0000 1000	0x4150 4000
L3_DSPSS_INIT_OCP_MMU1_CTRL_TARG_OCP_FW_504000_LOGICAL_ADDR_ERRLOG_0	R	32	0x0000 1004	0x4150 4004
L3_DSPSS_INIT_OCP_MMU1_CTRL_TARG_OCP_FW_504000_REGUPDATE_CONTROL	RW	32	0x0000 1040	0x4150 4040

Table 4-71. DSP2_FW_L2_NOC_CFG Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	DSP2_FW_L2_NOC_CFG Physical Address L3_MAIN Interconnect
L3_DSPSS_INIT_OCP_MMU1_CTRL_TARG_OCP_FW_504000_MRM_PERMISSION_REGION_LOW_0	RW	32	0x0000 1088	0x4150 4088
L3_DSPSS_INIT_OCP_MMU1_CTRL_TARG_OCP_FW_504000_MRM_PERMISSION_REGION_HIGH_0	RW	32	0x0000 108C	0x4150 408C
DSPNOC_FLAGMUX_ID_COREID	R	32	0x0000 4000	0x4150 7000
DSPNOC_FLAGMUX_ID_REVISIONID	R	32	0x0000 4004	0x4150 7004
DSPNOC_FLAGMUX_FAULTEN	RW	32	0x0000 4008	0x4150 7008
DSPNOC_FLAGMUX_FAULTSTATUS	R	32	0x0000 400C	0x4150 700C
DSPNOC_FLAGMUX_FLAGINEN0	RW	32	0x0000 4010	0x4150 7010
DSPNOC_FLAGMUX_FLAGINSTATUS0	R	32	0x0000 4014	0x4150 7014
DSPNOC_ERRORLOG_ID_COREID	R	32	0x0000 4200	0x4150 7200
DSPNOC_ERRORLOG_ID_REVISIONID	R	32	0x0000 4204	0x4150 7204
DSPNOC_ERRORLOG_FAULTEN	RW	32	0x0000 4208	0x4150 7208
DSPNOC_ERRORLOG_ERRVLD	R	32	0x0000 420C	0x4150 720C
DSPNOC_ERRORLOG_ERRCLR	RW	32	0x0000 4210	0x4150 7210
DSPNOC_ERRORLOG_ERRLOG0	R	32	0x0000 4214	0x4150 7214
DSPNOC_ERRORLOG_ERRLOG1	R	32	0x0000 4218	0x4150 7218
DSPNOC_ERRORLOG_ERRLOG3	R	32	0x0000 4220	0x4150 7220
DSPNOC_ERRORLOG_ERRLOG5	R	32	0x0000 4228	0x4150 7228

4.4.4.2 DSP_FW_L2_NOC_CFG Register Description

Table 4-72. L3_DSPSS_INIT_OCP_MMU0_CTRL_TARG_OCP_FW_503000_ERROR_LOG_0

Address Offset	0x0000 0000		
Physical Address	0x01D0 3000 0x40D0 3000 0x4150 3000	**Instance**	DSP_FW_L2_NOC_CFG DSP1_FW_L2_NOC_CFG DSP2_FW_L2_NOC_CFG
Description	Core 0 Error log register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				BLK_BURST_VIOLATION	RESERVED			REGION_START_ERRLOG			REGION_END_ERRLOG					REQINFO_ERRLOG																

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27	BLK_BURST_VIOLATION	2D burst not allowed or exceeding allowed size	RW	0x0
26	RESERVED		R	0x0
25:21	REGION_START_ERRLOG	Wrong access hit this region number	RW	0x0
20:16	REGION_END_ERRLOG	Wrong access hit this region number	RW	0x0
15:0	REQINFO_ERRLOG	Error in reqinfo vector	RW	0x0

Table 4-73. Register Call Summary for Register L3_DSPSS_INIT_OCP_MMU0_CTRL_TARG_OCP_FW_503000_ERROR_LOG_0

DSP Subsystem Register Manual
- DSP_FW_L2_NOC_CFG Register Summary: [0][1]

Table 4-74. L3_DSPSS_INIT_OCP_MMU0_CTRL_TARG_OCP_FW_503000_LOGICAL_ADDR_ERRLOG_0

Address Offset	0x0000 0004		
Physical Address	0x01D0 3004 0x40D0 3004 0x4150 3004	**Instance**	DSP_FW_L2_NOC_CFG DSP1_FW_L2_NOC_CFG DSP2_FW_L2_NOC_CFG
Description	Core 0 Logical Physical Address Error log register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								SLVOFS_LOGICAL																							

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:0	SLVOFS_LOGICAL	Address generated by the Arm before being translated	R	0x0

Table 4-75. Register Call Summary for Register L3_DSPSS_INIT_OCP_MMU0_CTRL_TARG_OCP_FW_503000_LOGICAL_ADDR_ERRLOG_0

DSP Subsystems Functional Description
- DSP Public Firewall Settings: [0]

DSP Subsystem Register Manual
- DSP_FW_L2_NOC_CFG Register Summary: [1][2]

Table 4-76. L3_DSPSS_INIT_OCP_MMU0_CTRL_TARG_OCP_FW_503000_REGUPDATE_CONTROL

Address Offset	0x0000 0040		
Physical Address	0x01D0 3040 0x40D0 3040 0x4150 3040	**Instance**	DSP_FW_L2_NOC_CFG DSP1_FW_L2_NOC_CFG DSP2_FW_L2_NOC_CFG
Description	Register update control register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								RESERVED																						FW_LOAD_REQ	FW_UPDATE_REQ

Bits	Field Name	Description	Type	Reset
31	RESERVED		R	0x0
30:2	RESERVED	Reserved	R	0x0
1	FW_LOAD_REQ	HW set/SW clear	RW	0x1
0	FW_UPDATE_REQ	HW set/SW clear	RW	0x1

Table 4-77. Register Call Summary for Register L3_DSPSS_INIT_OCP_MMU0_CTRL_TARG_OCP_FW_503000_REGUPDATE_CONTROL

DSP Subsystems Functional Description • DSP Public Firewall Settings: [0]
DSP Subsystem Register Manual • DSP_FW_L2_NOC_CFG Register Summary: [1][2]

Table 4-78. L3_DSPSS_INIT_OCP_MMU0_CTRL_TARG_OCP_FW_503000_MRM_PERMISSION_REGION_LOW_0

Address Offset	0x0000 0088		
Physical Address	0x01D0 3088 0x40D0 3088 0x4150 3088	**Instance**	DSP_FW_L2_NOC_CFG DSP1_FW_L2_NOC_CFG DSP2_FW_L2_NOC_CFG
Description	MRM_PERMISSION_REGION_0_LOW register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																PUB_PRV_DEBUG	PUB_USR_DEBUG	RESERVED		PUB_PRV_READ	PUB_PRV_WRITE	PUB_PRV_EXE	PUB_USR_READ	PUB_USR_WRITE	PUB_USR_EXE	RESERVED					

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		RW	0xFFFF
15	PUB_PRV_DEBUG	Public Privilege Domain Debug Allowed	RW	0x1
14	PUB_USR_DEBUG	Public User Domain Debug Allowed	RW	0x1
13:12	RESERVED		R	0x3
11	PUB_PRV_READ	Public Privilege Read Allowed	RW	0x1
10	PUB_PRV_WRITE	Public Privilege Write Allowed	RW	0x1
9	PUB_PRV_EXE	Public Privilege Exe Allowed	RW	0x1
8	PUB_USR_READ	Public User Read Access Allowed	RW	0x1
7	PUB_USR_WRITE	Public User Write Access Allowed	RW	0x1
6	PUB_USR_EXE	Public User Exe Access Allowed	RW	0x1
5:0	RESERVED		R	0x3F

Table 4-79. Register Call Summary for Register L3_DSPSS_INIT_OCP_MMU0_CTRL_TARG_OCP_FW_503000_MRM_PERMISSION_REGION_LOW_0

DSP Subsystems Functional Description • DSP Public Firewall Settings: [0]
DSP Subsystem Register Manual • DSP_FW_L2_NOC_CFG Register Summary: [1][2]

Table 4-80. L3_DSPSS_INIT_OCP_MMU0_CTRL_TARG_OCP_FW_503000_MRM_PERMISSION_REGION_HIGH_0

Address Offset	0x0000 008C

Table 4-80. L3_DSPSS_INIT_OCP_MMU0_CTRL_TARG_OCP_FW_503000_MRM_PERMISSION_REGION_HIGH_0 (continued)

Physical Address	0x01D0 308C 0x40D0 308C 0x4150 308C	**Instance**	DSP_FW_L2_NOC_CFG DSP1_FW_L2_NOC_CFG DSP2_FW_L2_NOC_CFG
Description	RM_PERMISSION_REGION_0_HIGH register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
W15	R15	W14	R14	W13	R13	W12	R12	W11	R11	W10	R10	W9	R9	W8	R8	W7	R7	W6	R6	W5	R5	W4	R4	W3	R3	W2	R2	W1	R1	W0	R0

Bits	Field Name	Description	Type	Reset
31	W15	Initiator ID15 permission	RW	0x1
30	R15	Initiator ID15 permission	RW	0x1
29	W14	Initiator ID14 permission	RW	0x1
28	R14	Initiator ID14 permission	RW	0x1
27	W13	Initiator ID13 permission	RW	0x1
26	R13	Initiator ID13 permission	RW	0x1
25	W12	Initiator ID12 permission	RW	0x1
24	R12	Initiator ID12 permission	RW	0x1
23	W11	Initiator ID11 permission	RW	0x1
22	R11	Initiator ID11 permission	RW	0x1
21	W10	Initiator ID10 permission	RW	0x1
20	R10	Initiator ID10 permission	RW	0x1
19	W9	Initiator ID9 permission	RW	0x1
18	R9	Initiator ID9 permission	RW	0x1
17	W8	Initiator ID8 permission	RW	0x1
16	R8	Initiator ID8 permission	RW	0x1
15	W7	Initiator ID7 permission	RW	0x1
14	R7	Initiator ID7 permission	RW	0x1
13	W6	Initiator ID6 permission	RW	0x1
12	R6	Initiator ID6 permission	RW	0x1
11	W5	Initiator ID5 permission	RW	0x1
10	R5	Initiator ID5 permission	RW	0x1
9	W4	Initiator ID4 permission	RW	0x1
8	R4	Initiator ID4 permission	RW	0x1
7	W3	Initiator ID3 permission	RW	0x1
6	R3	Initiator ID3 permission	RW	0x1
5	W2	Initiator ID2 permission	RW	0x1
4	R2	Initiator ID2 permission	RW	0x1
3	W1	Initiator ID1 permission	RW	0x1
2	R1	Initiator ID1 permission	RW	0x1
1	W0	Initiator ID0 permission	RW	0x1
0	R0	Initiator ID0 permission	RW	0x1

Table 4-81. Register Call Summary for Register L3_DSPSS_INIT_OCP_MMU0_CTRL_TARG_OCP_FW_503000_MRM_PERMISSION_REGION_HIGH_0

DSP Subsystems Functional Description • DSP Public Firewall Settings: [0]
DSP Subsystem Register Manual • DSP_FW_L2_NOC_CFG Register Summary: [1][2]

Table 4-82. L3_DSPSS_INIT_OCP_MMU0_CTRL_TARG_OCP_FW_503000_START_REGION_1

Address Offset	0x0000 0090		
Physical Address	0x01D0 3090 0x40D0 3090 0x4150 3090	**Instance**	DSP_FW_L2_NOC_CFG DSP1_FW_L2_NOC_CFG DSP2_FW_L2_NOC_CFG
Description	Start physical address of region 1		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												START_REGION_1			

Bits	Field Name	Description	Type	Reset
31:4	RESERVED		R	0x0
3:0	START_REGION_1	Physical target start address of firewall region 1	RW	0x0

Table 4-83. Register Call Summary for Register L3_DSPSS_INIT_OCP_MMU0_CTRL_TARG_OCP_FW_503000_START_REGION_1

DSP Subsystem Register Manual • DSP_FW_L2_NOC_CFG Register Summary: [0][1]

Table 4-84. L3_DSPSS_INIT_OCP_MMU0_CTRL_TARG_OCP_FW_503000_END_REGION_1

Address Offset	0x0000 0094		
Physical Address	0x01D0 3094 0x40D0 3094 0x4150 3094	**Instance**	DSP_FW_L2_NOC_CFG DSP1_FW_L2_NOC_CFG DSP2_FW_L2_NOC_CFG
Description	End physical address of region 1		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
END_REGION_1_ENABLE	RESERVED																											END_REGION_1			

Bits	Field Name	Description	Type	Reset
31	END_REGION_1_ENABLE	End Region 1 enable	RW	0x0
30:4	RESERVED		R	0x0
3:0	END_REGION_1	Physical target end address of firewall region 1	RW	0x0

Table 4-85. Register Call Summary for Register L3_DSPSS_INIT_OCP_MMU0_CTRL_TARG_OCP_FW_503000_END_REGION_1

DSP Subsystem Register Manual
- DSP_FW_L2_NOC_CFG Register Summary: [0][1]

Table 4-86. L3_DSPSS_INIT_OCP_MMU0_CTRL_TARG_OCP_FW_503000_MRM_PERMISSION_REGION_LOW_1

Address Offset	0x0000 0098		
Physical Address	0x01D0 3098 0x40D0 3098 0x4150 3098	**Instance**	DSP_FW_L2_NOC_CFG DSP1_FW_L2_NOC_CFG DSP2_FW_L2_NOC_CFG
Description	RM_PERMISSION_REGION_1_LOW register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																PUB_PRV_DEBUG	PUB_USR_DEBUG	RESERVED		PUB_PRV_READ	PUB_PRV_WRITE	PUB_PRV_EXE	PUB_USR_READ	PUB_USR_WRITE	PUB_USR_EXE	RESERVED					

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		RW	0xFFFF
15	PUB_PRV_DEBUG	Public Privilege Domain Debug Allowed	RW	0x1
14	PUB_USR_DEBUG	Public User Domain Debug Allowed	RW	0x1
13:12	RESERVED		R	0x3
11	PUB_PRV_READ	Public Privilege Read Allowed	RW	0x1
10	PUB_PRV_WRITE	Public Privilege Write Allowed	RW	0x1
9	PUB_PRV_EXE	Public Privilege Exe Allowed	RW	0x1
8	PUB_USR_READ	Public User Read Access Allowed	RW	0x1
7	PUB_USR_WRITE	Public User Write Access Allowed	RW	0x1
6	PUB_USR_EXE	Public User Exe Access Allowed	RW	0x1
5:0	RESERVED		R	0x3F

Table 4-87. Register Call Summary for Register L3_DSPSS_INIT_OCP_MMU0_CTRL_TARG_OCP_FW_503000_MRM_PERMISSION_REGION_LOW_1

DSP Subsystem Register Manual
- DSP_FW_L2_NOC_CFG Register Summary: [0][1]

Table 4-88. L3_DSPSS_INIT_OCP_MMU0_CTRL_TARG_OCP_FW_503000_MRM_PERMISSION_REGION_HIGH_1

Address Offset	0x0000 009C		
Physical Address	0x01D0 309C 0x40D0 309C 0x4150 309C	**Instance**	DSP_FW_L2_NOC_CFG DSP1_FW_L2_NOC_CFG DSP2_FW_L2_NOC_CFG
Description	RM_PERMISSION_REGION_1_HIGH register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
W15	R15	W14	R14	W13	R13	W12	R12	W11	R11	W10	R10	W9	R9	W8	R8	W7	R7	W6	R6	W5	R5	W4	R4	W3	R3	W2	R2	W1	R1	W0	R0

Bits	Field Name	Description	Type	Reset
31	W15	Initiator ID15 permission	RW	0x1
30	R15	Initiator ID15 permission	RW	0x1
29	W14	Initiator ID14 permission	RW	0x1
28	R14	Initiator ID14 permission	RW	0x1
27	W13	Initiator ID13 permission	RW	0x1
26	R13	Initiator ID13 permission	RW	0x1
25	W12	Initiator ID12 permission	RW	0x1
24	R12	Initiator ID12 permission	RW	0x1
23	W11	Initiator ID11 permission	RW	0x1
22	R11	Initiator ID11 permission	RW	0x1
21	W10	Initiator ID10 permission	RW	0x1
20	R10	Initiator ID10 permission	RW	0x1
19	W9	Initiator ID9 permission	RW	0x1
18	R9	Initiator ID9 permission	RW	0x1
17	W8	Initiator ID8 permission	RW	0x1
16	R8	Initiator ID8 permission	RW	0x1
15	W7	Initiator ID7 permission	RW	0x1
14	R7	Initiator ID7 permission	RW	0x1
13	W6	Initiator ID6 permission	RW	0x1
12	R6	Initiator ID6 permission	RW	0x1
11	W5	Initiator ID5 permission	RW	0x1
10	R5	Initiator ID5 permission	RW	0x1
9	W4	Initiator ID4 permission	RW	0x1
8	R4	Initiator ID4 permission	RW	0x1
7	W3	Initiator ID3 permission	RW	0x1
6	R3	Initiator ID3 permission	RW	0x1
5	W2	Initiator ID2 permission	RW	0x1
4	R2	Initiator ID2 permission	RW	0x1
3	W1	Initiator ID1 permission	RW	0x1
2	R1	Initiator ID1 permission	RW	0x1
1	W0	Initiator ID0 permission	RW	0x1
0	R0	Initiator ID0 permission	RW	0x1

Table 4-89. Register Call Summary for Register L3_DSPSS_INIT_OCP_MMU0_CTRL_TARG_OCP_FW_503000_MRM_PERMISSION_REGION_HIGH_1

DSP Subsystem Register Manual
- DSP_FW_L2_NOC_CFG Register Summary: [0][1]

Table 4-90. L3_DSPSS_INIT_OCP_MMU1_CTRL_TARG_OCP_FW_504000_ERROR_LOG_0

Address Offset	0x0000 1000		
Physical Address	0x01D0 4000 0x40D0 4000 0x4150 4000	**Instance**	DSP_FW_L2_NOC_CFG DSP1_FW_L2_NOC_CFG DSP2_FW_L2_NOC_CFG
Description	Core 0 Error log register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				BLK_BURST_VIOLATION	RESERVED			REGION_START_ERRLOG				REGION_END_ERRLOG				REQINFO_ERRLOG																	

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27	BLK_BURST_VIOLATION	2D burst not allowed or exceeding allowed size	RW	0x0
26	RESERVED		R	0x0
25:21	REGION_START_ERRLOG	Wrong access hit this region number	RW	0x0
20:16	REGION_END_ERRLOG	Wrong access hit this region number	RW	0x0
15:0	REQINFO_ERRLOG	Error in reqinfo vector	RW	0x0

Table 4-91. Register Call Summary for Register L3_DSPSS_INIT_OCP_MMU1_CTRL_TARG_OCP_FW_504000_ERROR_LOG_0

DSP Subsystem Register Manual
- DSP_FW_L2_NOC_CFG Register Summary: [0][1]

Table 4-92. L3_DSPSS_INIT_OCP_MMU1_CTRL_TARG_OCP_FW_504000_LOGICAL_ADDR_ERRLOG_0

Address Offset	0x0000 1004		
Physical Address	0x01D0 4004 0x40D0 4004 0x4150 4004	**Instance**	DSP_FW_L2_NOC_CFG DSP1_FW_L2_NOC_CFG DSP2_FW_L2_NOC_CFG
Description	Core 0 Logical Physical Address Error log register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				SLVOFS_LOGICAL																											

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:0	SLVOFS_LOGICAL	Address generated by the Arm before being translated	R	0x0

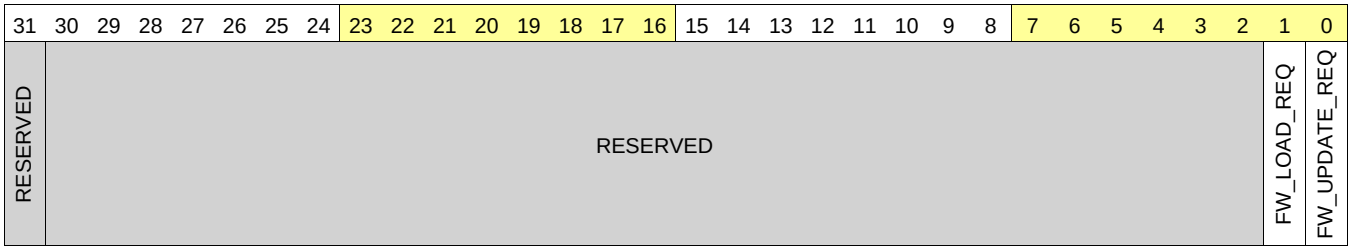
Table 4-93. Register Call Summary for Register L3_DSPSS_INIT_OCP_MMU1_CTRL_TARG_OCP_FW_504000_LOGICAL_ADDR_ERRLOG_0

DSP Subsystem Register Manual

- DSP_FW_L2_NOC_CFG Register Summary: [0][1]

Table 4-94. L3_DSPSS_INIT_OCP_MMU1_CTRL_TARG_OCP_FW_504000_REGUPDATE_CONTROL

Address Offset	0x0000 1040		
Physical Address	0x01D0 4040 0x40D0 4040 0x4150 4040	**Instance**	DSP_FW_L2_NOC_CFG DSP1_FW_L2_NOC_CFG DSP2_FW_L2_NOC_CFG
Description	Register update control register		
Type	RW		



Bits	Field Name	Description	Type	Reset
31	RESERVED		R	0x0
30:2	RESERVED	Reserved	R	0x0
1	FW_LOAD_REQ	HW set/SW clear	RW	0x1
0	FW_UPDATE_REQ	HW set/SW clear	RW	0x1

Table 4-95. Register Call Summary for Register L3_DSPSS_INIT_OCP_MMU1_CTRL_TARG_OCP_FW_504000_REGUPDATE_CONTROL

DSP Subsystem Register Manual

- DSP_FW_L2_NOC_CFG Register Summary: [0][1]

Table 4-96. L3_DSPSS_INIT_OCP_MMU1_CTRL_TARG_OCP_FW_504000_MRM_PERMISSION_REGION_LOW_0

Address Offset	0x0000 1088		
Physical Address	0x01D0 4088 0x40D0 4088 0x4150 4088	**Instance**	DSP_FW_L2_NOC_CFG DSP1_FW_L2_NOC_CFG DSP2_FW_L2_NOC_CFG
Description	RM_PERMISSION_REGION_0_LOW register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																PUB_PRV_DEBUG	PUB_USR_DEBUG	RESERVED		PUB_PRV_READ	PUB_PRV_WRITE	PUB_PRV_EXE	PUB_USR_READ	PUB_USR_WRITE	PUB_USR_EXE	RESERVED					

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		RW	0xFFFF
15	PUB_PRV_DEBUG	Public Privilege Domain Debug Allowed	RW	0x1
14	PUB_USR_DEBUG	Public User Domain Debug Allowed	RW	0x1
13:12	RESERVED		R	0x3
11	PUB_PRV_READ	Public Privilege Read Allowed	RW	0x1
10	PUB_PRV_WRITE	Public Privilege Write Allowed	RW	0x1
9	PUB_PRV_EXE	Public Privilege Exe Allowed	RW	0x1
8	PUB_USR_READ	Public User Read Access Allowed	RW	0x1
7	PUB_USR_WRITE	Public User Write Access Allowed	RW	0x1
6	PUB_USR_EXE	Public User Exe Access Allowed	RW	0x1
5:0	RESERVED		R	0x3F

Table 4-97. Register Call Summary for Register L3_DSPSS_INIT_OCP_MMU1_CTRL_TARG_OCP_FW_504000_MRM_PERMISSION_REGION_LOW_0

DSP Subsystems Functional Description
- DSP Public Firewall Settings: [0]

DSP Subsystem Register Manual
- DSP_FW_L2_NOC_CFG Register Summary: [1][2]

Table 4-98. L3_DSPSS_INIT_OCP_MMU1_CTRL_TARG_OCP_FW_504000_MRM_PERMISSION_REGION_HIGH_0

Address Offset	0x0000 108C		
Physical Address	0x01D0 408C 0x40D0 408C 0x4150 408C	**Instance**	DSP_FW_L2_NOC_CFG DSP1_FW_L2_NOC_CFG DSP2_FW_L2_NOC_CFG
Description	RM_PERMISSION_REGION_0_HIGH register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
W15	R15	W14	R14	W13	R13	W12	R12	W11	R11	W10	R10	W9	R9	W8	R8	W7	R7	W6	R6	W5	R5	W4	R4	W3	R3	W2	R2	W1	R1	W0	R0

Bits	Field Name	Description	Type	Reset
31	W15	Initiator ID15 permission	RW	0x1
30	R15	Initiator ID15 permission	RW	0x1
29	W14	Initiator ID14 permission	RW	0x1
28	R14	Initiator ID14 permission	RW	0x1
27	W13	Initiator ID13 permission	RW	0x1
26	R13	Initiator ID13 permission	RW	0x1
25	W12	Initiator ID12 permission	RW	0x1

Bits	Field Name	Description	Type	Reset
24	R12	Initiator ID12 permission	RW	0x1
23	W11	Initiator ID11 permission	RW	0x1
22	R11	Initiator ID11 permission	RW	0x1
21	W10	Initiator ID10 permission	RW	0x1
20	R10	Initiator ID10 permission	RW	0x1
19	W9	Initiator ID9 permission	RW	0x1
18	R9	Initiator ID9 permission	RW	0x1
17	W8	Initiator ID8 permission	RW	0x1
16	R8	Initiator ID8 permission	RW	0x1
15	W7	Initiator ID7 permission	RW	0x1
14	R7	Initiator ID7 permission	RW	0x1
13	W6	Initiator ID6 permission	RW	0x1
12	R6	Initiator ID6 permission	RW	0x1
11	W5	Initiator ID5 permission	RW	0x1
10	R5	Initiator ID5 permission	RW	0x1
9	W4	Initiator ID4 permission	RW	0x1
8	R4	Initiator ID4 permission	RW	0x1
7	W3	Initiator ID3 permission	RW	0x1
6	R3	Initiator ID3 permission	RW	0x1
5	W2	Initiator ID2 permission	RW	0x1
4	R2	Initiator ID2 permission	RW	0x1
3	W1	Initiator ID1 permission	RW	0x1
2	R1	Initiator ID1 permission	RW	0x1
1	W0	Initiator ID0 permission	RW	0x1
0	R0	Initiator ID0 permission	RW	0x1

Table 4-99. Register Call Summary for Register L3_DSPSS_INIT_OCP_MMU1_CTRL_TARG_OCP_FW_504000_MRM_PERMISSION_REGION_HIGH_0

DSP Subsystems Functional Description
- DSP Public Firewall Settings: [0]

DSP Subsystem Register Manual
- DSP_FW_L2_NOC_CFG Register Summary: [1][2]

Table 4-100. DSPNOC_FLAGMUX_ID_COREID

Address Offset	0x0000 4000		
Physical Address	0x01D0 7000 0x40D0 7000 0x4150 7000	**Instance**	DSP_FW_L2_NOC_CFG DSP1_FW_L2_NOC_CFG DSP2_FW_L2_NOC_CFG
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CORECHECKSUM																								CORETYPEID							

Bits	Field Name	Description	Type	Reset
31:8	CORECHECKSUM	Field containing a checksum of the parameters of the IP.	R	0xff71d7
7:0	CORETYPEID	Field identifying the type of IP.	R	0xb

Table 4-101. Register Call Summary for Register DSPNOC_FLAGMUX_ID_COREID

DSP Subsystem Register Manual
- DSP_FW_L2_NOC_CFG Register Summary: [0][1]

Table 4-102. DSPNOC_FLAGMUX_ID_REVISIONID

Address Offset	0x0000 4004		
Physical Address	0x01D0 7004 0x40D0 7004 0x4150 7004	**Instance**	DSP_FW_L2_NOC_CFG DSP1_FW_L2_NOC_CFG DSP2_FW_L2_NOC_CFG
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REVISION																															

Bits	Field Name	Description	Type	Reset
31:0	REVISION	IP Revision.	R	0x-(1)

(1) TI internal data

Table 4-103. Register Call Summary for Register DSPNOC_FLAGMUX_ID_REVISIONID

DSP Subsystem Register Manual
- DSP_FW_L2_NOC_CFG Register Summary: [0][1]

Table 4-104. DSPNOC_FLAGMUX_FAULTEN

Address Offset	0x0000 4008		
Physical Address	0x01D0 7008 0x40D0 7008 0x4150 7008	**Instance**	DSP_FW_L2_NOC_CFG DSP1_FW_L2_NOC_CFG DSP2_FW_L2_NOC_CFG
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															FAULTEN

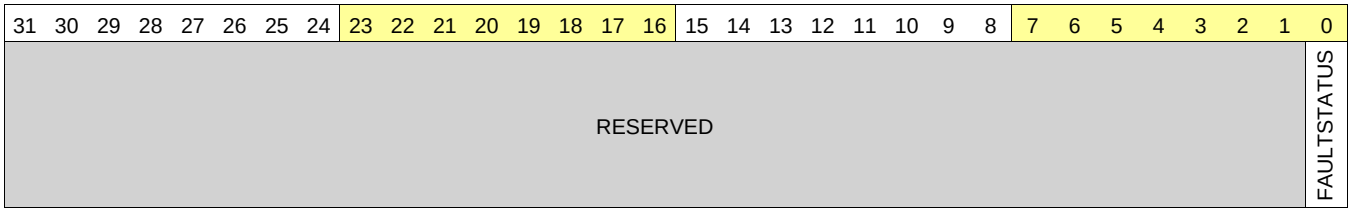
Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	FAULTEN	Global Fault Enable register	RW	0x1

Table 4-105. Register Call Summary for Register DSPNOC_FLAGMUX_FAULTEN

DSP Subsystem Register Manual
- DSP_FW_L2_NOC_CFG Register Summary: [0][1]

Table 4-106. DSPNOC_FLAGMUX_FAULTSTATUS

Address Offset	0x0000 400C		
Physical Address	0x01D0 700C 0x40D0 700C 0x4150 700C	**Instance**	DSP_FW_L2_NOC_CFG DSP1_FW_L2_NOC_CFG DSP2_FW_L2_NOC_CFG
Description			
Type	R		



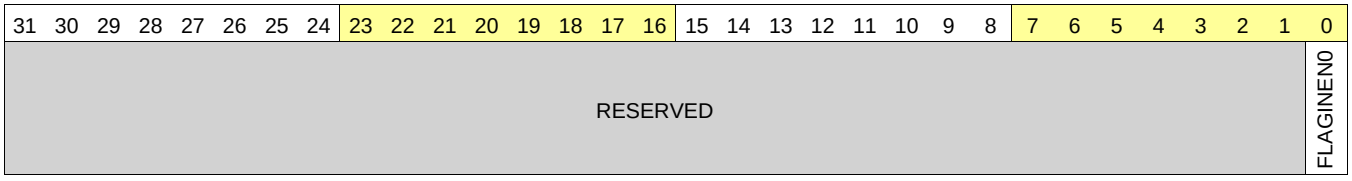
Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	FAULTSTATUS	Global Fault Status register	R	0x0

Table 4-107. Register Call Summary for Register DSPNOC_FLAGMUX_FAULTSTATUS

DSP Subsystem Register Manual
- DSP_FW_L2_NOC_CFG Register Summary: [0][1]

Table 4-108. DSPNOC_FLAGMUX_FLAGINEN0

Address Offset	0x0000 4010		
Physical Address	0x01D0 7010 0x40D0 7010 0x4150 7010	**Instance**	DSP_FW_L2_NOC_CFG DSP1_FW_L2_NOC_CFG DSP2_FW_L2_NOC_CFG
Description			
Type	RW		



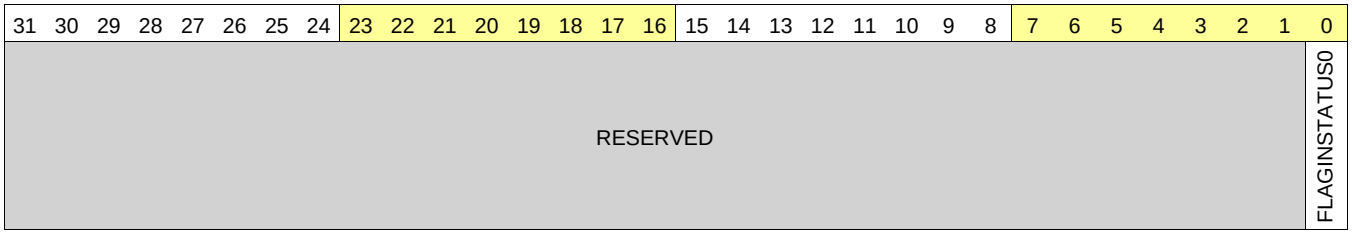
Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	FLAGINEN0	FlagIn Enable register #0	RW	0x1

Table 4-109. Register Call Summary for Register DSPNOC_FLAGMUX_FLAGINEN0

DSP Subsystem Register Manual
- DSP_FW_L2_NOC_CFG Register Summary: [0][1]

Table 4-110. DSPNOC_FLAGMUX_FLAGINSTATUS0

Address Offset	0x0000 4014		
Physical Address	0x01D0 7014 0x40D0 7014 0x4150 7014	**Instance**	DSP_FW_L2_NOC_CFG DSP1_FW_L2_NOC_CFG DSP2_FW_L2_NOC_CFG
Description			
Type	R		



Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	FLAGINSTATUS0	FlagIn Status register #0	R	0x0

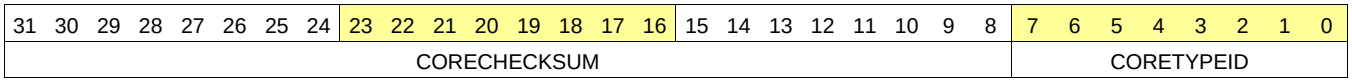
Table 4-111. Register Call Summary for Register DSPNOC_FLAGMUX_FLAGINSTATUS0

DSP Subsystem Register Manual

- DSP_FW_L2_NOC_CFG Register Summary: [0][1]

Table 4-112. DSPNOC_ERRORLOG_ID_COREID

Address Offset	0x0000 4200		
Physical Address	0x01D0 7200 0x40D0 7200 0x4150 7200	**Instance**	DSP_FW_L2_NOC_CFG DSP1_FW_L2_NOC_CFG DSP2_FW_L2_NOC_CFG
Description			
Type	R		



Bits	Field Name	Description	Type	Reset
31:8	CORECHECKSUM	Field containing a checksum of the parameters of the IP.	R	0xaf434
7:0	CORETYPEID	Field identifying the type of IP.	R	0xd

Table 4-113. Register Call Summary for Register DSPNOC_ERRORLOG_ID_COREID

DSP Subsystem Register Manual

- DSP_FW_L2_NOC_CFG Register Summary: [0][1]

Table 4-114. DSPNOC_ERRORLOG_ID_REVISIONID

Address Offset	0x0000 4204		
Physical Address	0x01D0 7204 0x40D0 7204 0x4150 7204	**Instance**	DSP_FW_L2_NOC_CFG DSP1_FW_L2_NOC_CFG DSP2_FW_L2_NOC_CFG
Description			

Table 4-114. DSPNOC_ERRORLOG_ID_REVISIONID (continued)

Type	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REVISION																															

Bits	Field Name	Description	Type	Reset
31:0	REVISION	IP Revision.	R	0x-(1)

(1) TI Internal data

Table 4-115. Register Call Summary for Register DSPNOC_ERRORLOG_ID_REVISIONID

DSP Subsystem Register Manual
- DSP_FW_L2_NOC_CFG Register Summary: [0][1]

Table 4-116. DSPNOC_ERRORLOG_FAULTEN

Address Offset	0x0000 4208		
Physical Address	0x01D0 7208 0x40D0 7208 0x4150 7208	**Instance**	DSP_FW_L2_NOC_CFG DSP1_FW_L2_NOC_CFG DSP2_FW_L2_NOC_CFG
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															FAULTEN

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	FAULTEN	Enable Fault output	RW	0x1

Table 4-117. Register Call Summary for Register DSPNOC_ERRORLOG_FAULTEN

DSP Subsystem Register Manual
- DSP_FW_L2_NOC_CFG Register Summary: [0][1]

Table 4-118. DSPNOC_ERRORLOG_ERRVLD

Address Offset	0x0000 420C		
Physical Address	0x01D0 720C 0x40D0 720C 0x4150 720C	**Instance**	DSP_FW_L2_NOC_CFG DSP1_FW_L2_NOC_CFG DSP2_FW_L2_NOC_CFG
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															ERRVLD

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	ERRVLD	Error logged Valid	R	0x0

Table 4-119. Register Call Summary for Register DSPNOC_ERRORLOG_ERRVLD

DSP Subsystem Register Manual
- DSP_FW_L2_NOC_CFG Register Summary: [0][1]

Table 4-120. DSPNOC_ERRORLOG_ERRCLR

Address Offset	0x0000 4210		
Physical Address	0x01D0 7210 0x40D0 7210 0x4150 7210	**Instance**	DSP_FW_L2_NOC_CFG DSP1_FW_L2_NOC_CFG DSP2_FW_L2_NOC_CFG
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															ERRCLR

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	ERRCLR	Clr ErrVld status	RW	0x0

Table 4-121. Register Call Summary for Register DSPNOC_ERRORLOG_ERRCLR

DSP Subsystem Register Manual
- DSP_FW_L2_NOC_CFG Register Summary: [0][1]

Table 4-122. DSPNOC_ERRORLOG_ERRLOG0

Address Offset	0x0000 4214		
Physical Address	0x01D0 7214 0x40D0 7214 0x4150 7214	**Instance**	DSP_FW_L2_NOC_CFG DSP1_FW_L2_NOC_CFG DSP2_FW_L2_NOC_CFG
Description	Header: Lock, Opcode, Len1, ErrCode values		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FORMAT	RESERVED							LEN1								RESERVED					ERRCODE			RESERVED			OPC				LOCK

Bits	Field Name	Description	Type	Reset
31	FORMAT	Format of ErrLog0 register	R	0x1
30:28	RESERVED		R	0x0
27:16	LEN1	Header: Len1 value	R	0x0
15:11	RESERVED		R	0x0
10:8	ERRCODE	Header: Error Code value	R	0x0

Bits	Field Name	Description	Type	Reset
7:5	RESERVED		R	0x0
4:1	OPC	Header: Opcode value	R	0x0
0	LOCK	Header: Lock bit value	R	0x0

Table 4-123. Register Call Summary for Register DSPNOC_ERRORLOG_ERRLOG0

DSP Subsystem Register Manual
- DSP_FW_L2_NOC_CFG Register Summary: [0][1]

Table 4-124. DSPNOC_ERRORLOG_ERRLOG1

Address Offset	0x0000 4218		
Physical Address	0x01D0 7218 0x40D0 7218 0x4150 7218	**Instance**	DSP_FW_L2_NOC_CFG DSP1_FW_L2_NOC_CFG DSP2_FW_L2_NOC_CFG
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																ERRLOG1															

Bits	Field Name	Description	Type	Reset
31:15	RESERVED		R	0x0
14:0	ERRLOG1	Header: RouteId lsb value	R	0x0

Table 4-125. Register Call Summary for Register DSPNOC_ERRORLOG_ERRLOG1

DSP Subsystem Register Manual
- DSP_FW_L2_NOC_CFG Register Summary: [0][1]

Table 4-126. DSPNOC_ERRORLOG_ERRLOG3

Address Offset	0x0000 4220		
Physical Address	0x01D0 7220 0x40D0 7220 0x4150 7220	**Instance**	DSP_FW_L2_NOC_CFG DSP1_FW_L2_NOC_CFG DSP2_FW_L2_NOC_CFG
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	ERRLOG3																														

Bits	Field Name	Description	Type	Reset
31	RESERVED		R	0x0
30:0	ERRLOG3	Header: Addr lsb value	R	0x0

Table 4-127. Register Call Summary for Register DSPNOC_ERRORLOG_ERRLOG3

DSP Subsystem Register Manual

- DSP_FW_L2_NOC_CFG Register Summary: [0][1]

Table 4-128. DSPNOC_ERRORLOG_ERRLOG5

Address Offset	0x0000 4228		
Physical Address	0x01D0 7228 0x40D0 7228 0x4150 7228	**Instance**	DSP_FW_L2_NOC_CFG DSP1_FW_L2_NOC_CFG DSP2_FW_L2_NOC_CFG
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								ERRLOG5																							

Bits	Field Name	Description	Type	Reset
31:22	RESERVED		R	0x0
21:0	ERRLOG5	Header: User lsb value	R	0x0

Table 4-129. Register Call Summary for Register DSPNOC_ERRORLOG_ERRLOG5

DSP Subsystem Register Manual

- DSP_FW_L2_NOC_CFG Register Summary: [0][1]

Dual Cortex-M4 IPU Subsystem

This chapter describes the dual Cortex-M4 image processor unit (IPU) subsystem.

Topic **Page**

5.1 Dual Cortex-M4 IPU Subsystem Overview

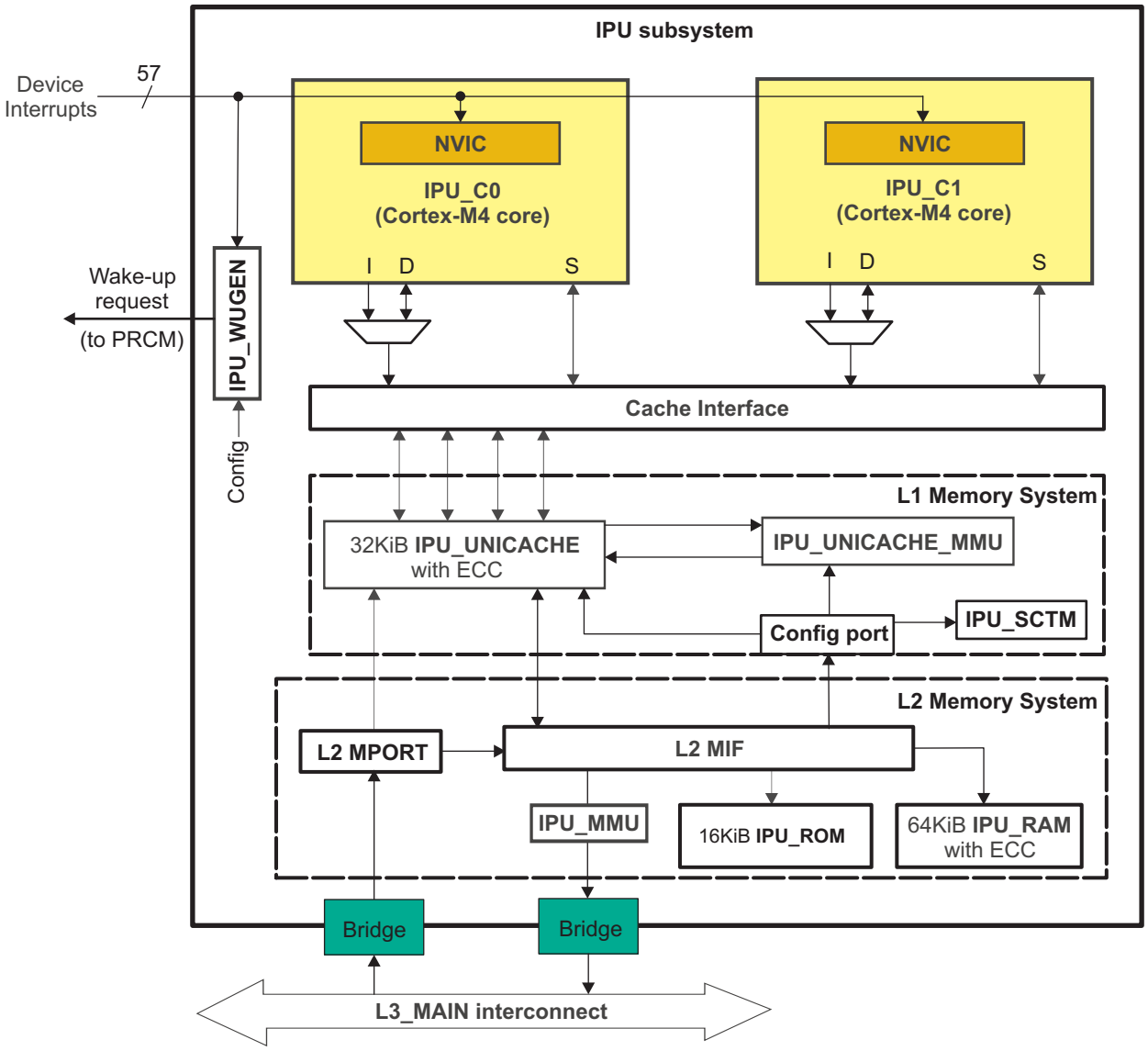
5.1.1 Introduction

The Imaging Processor Unit (IPU) subsystem contains two Arm® Cortex™-M4 cores (IPU_C0 and IPU_C1) that share a common level 1 (L1) cache (called unicache). The two Cortex-M4 cores are completely homogeneous to one another. Any task possible using one Cortex-M4 core is also possible using the other Cortex-M4 core. Both Cortex-M4 cores could be used for tasks such as running RTOS, controlling ISP, SIMCOP, DSS, and other functions. It is software responsibility to distribute the various tasks between the Cortex-M4 cores for optimal performance. The integrated interrupt handling of the IPU subsystem allows it to function as an efficient control unit.

IPU is the boot master of this device with its own boot ROM.

Figure 5-1 is a high-level block diagram of the IPU subsystem.

Figure 5-1. IPU Subsystem Overview



5.1.2 Key Features

The key features of the IPU subsystem are:

- Two Arm Cortex-M4 microprocessors (IPU_C0 and IPU_C1):
  - Armv7-M and Thumb®-2 instruction set architecture (ISA)
  - Armv6 SIMD and digital signal processor (DSP) extensions
  - Single-cycle MAC
  - Integrated nested vector interrupt controller (NVIC) (also called IPU_Cx_INTC, where x = 0, 1)
  - Integrated bus matrix
    - Bus arbiter
    - Bit-banding – atomic bit manipulation
    - Write buffer
    - Memory interface (I and D) plus system interface (S) and private peripheral bus (PPB)
  - Registers:
    - Thirteen general-purpose 32-bit registers
    - Link register (LR)
    - Program counter (PC)
    - Program status register, xPSR
    - Two banked SP registers
  - Integrated power management
  - Extensive debug capabilities
- Unicache interface:
  - AHBLite to unicache interface
  - Instruction and data interface
  - Supports interleaved Cortex-M4 requests
- L1 cache (IPU_UNICACHE):
  - 32KiB divided into 16 banks
  - 4-way
  - Runs at twice the Cortex-M4 CPU frequency
  - Cache configuration lock/freeze/preload
  - Internal MMU:
    - 16-entry region-based address translation
    - Read/write control and access type control
    - Execute Never (XN) MMU protection policy
    - Little-endian format
  - OCP port for configuration and cache maintenance
- Subsystem counter timer module (SCTM) connected to unicache
- L2 master interface (MIF):
  - Splitter for access to memory or OCP ports
  - Interleaved bank request for fast memory access
- L2 internal memories:
  - 16KiB ROM (IPU_ROM); used for device boot/initialization
  - 64KiB banked RAM – IPU_RAM
- L2 MMU (IPU_MMU): 32 entries with Table Walking Logic (TWL)
- Wake-up generator (IPU_WUGEN): Generates wake-up request from external interrupts
- Two OCP ports at IPU boundary (connected to the L3_MAIN interconnect):

  - Master port – allows the IPU to access system resources (memories and peripherals)
  - Slave port – allows other requestors to access a part of the IPU internal memory space
- Power management:
  - Local power-management control: Configurable through the IPU_WUGEN registers.
  - Two sleep modes supported by Cortex-M4, controlled by its integrated interrupt controller (NVIC).
  - Cortex-M4 system is clock-gated in both sleep modes.
  - NVIC interrupt interface stays awake.
  - Supports L1 cache and L2 memories retention
- Error-Correcting Code (ECC) supported for both L1 unicache and L2 RAM
- Debug/emulation features supported

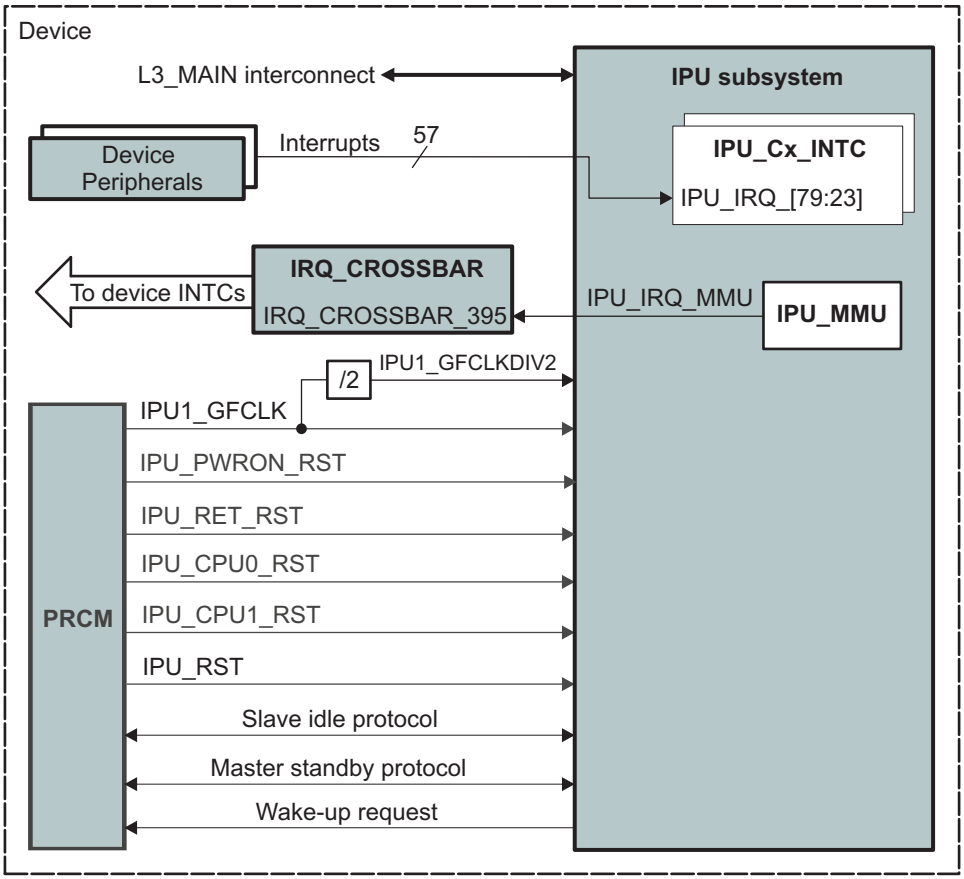
Section 5.3 provides more details about the IPU features.

5.2 Dual Cortex-M4 IPU Subsystem Integration

This section describes the integration of the module in the device, including information about clocks, resets, and hardware requests.

Figure 5-2 shows the IPU integration in the device.

Figure 5-2. IPU Subsystem Integration



NOTE: In Figure 5-2, only interrupts external to IPU (that is, outside IPU boundary) are shown. The complete IPU interrupt mapping is presented in Chapter 12, *Interrupt Controllers*.

NOTE: For more information about the master standby and slave idle protocols, and the wake-up request, see Section 3.1.1.1.2, *Module-Level Clock Management*, in Chapter 3, *Power, Reset, and Clock Management*.

Table 5-1 through Table 5-3 summarize the integration of the module in the device.

Table 5-1. IPU Integration Attributes

Module Instance	Attributes	
	Power Domain	Interconnect
IPU	PD_IPU, PD_COREAON	L3_MAIN

Table 5-2. IPU Hardware Requests

Interrupt Requests				
Module Instance	**Interrupt Name (Source)**	**IRQ_CROSSBAR Input (Destination)**	**Default Mapping**	**Description**
IPU	IPU_IRQ_MMU	IRQ_CROSSBAR_395	N/A	IPU_MMU fault interrupt. This IRQ is not mapped to any device interrupt controller by default, other than the IPU interrupt controller itself (IPU_IRQ_16 interrupt line). However, this IRQ can be mapped to other device interrupt controllers by properly configuring the corresponding IRQ_CROSSBAR related registers in the CTRL_MODULE.

Table 5-3. IPU Clocks and Resets

Clocks				
Module Instance	**Destination Signal Name**	**Source Signal Name**	**Source**	**Description**
IPU	IPU1_GFCLK	IPU1_GFCLK	PRCM	IPU interface and functional clock(s). See Section 5.2.1 for details.
	IPU1_GFCLKDIV2	IPU1_GFCLK		
Resets				
Module Instance	**Destination Signal Name**	**Source Signal Name**	**Source**	**Description**
IPU	IPU_PWRON_RST	IPU_PWRON_RST	PRCM	Power-on reset. Resets the entire IPU subsystem (except SWJ-DP)
	IPU_RET_RST	IPU_RET_RST	PRCM	Retention reset to few retention logic inside the IPU_UNICACHE
	IPU_CPU0_RST	IPU_CPU0_RST	PRCM	IPU_C0 reset. Resets the entire processor (except all the debug logic in the processor and SWJ-DP)
	IPU_CPU1_RST	IPU_CPU1_RST	PRCM	IPU_C1 reset. Resets the entire processor (except all the debug logic in the processor and SWJ-DP)
	IPU_RST	IPU_RST	PRCM	Reset signal to the IPU_UNICACHE and the IPU_MMU

For more information about clocks, resets, and power domains, see Chapter 3, *Power, Reset, and Clock Management*.

5.2.1 IPU Subsystem Clock Distribution

The IPU subsystem has two clock inputs:

- **IPU1_GFCLK**: Main source clock for IPU; feeds most of IPU internal modules:
  - IPU Unicache & MMU, L2 MIF & MPORT (all running on (1x) IPU_GFCLK)
  - Cortex-M4 cores, IPU_RAM, IPU_ROM (all running on (1/2x) IPU_GFCLK via internal divider)
- **IPU1_GFCLKDIV2**: This is IPU1_GFCLK externally divided by two; feeds IPU L2 MMU

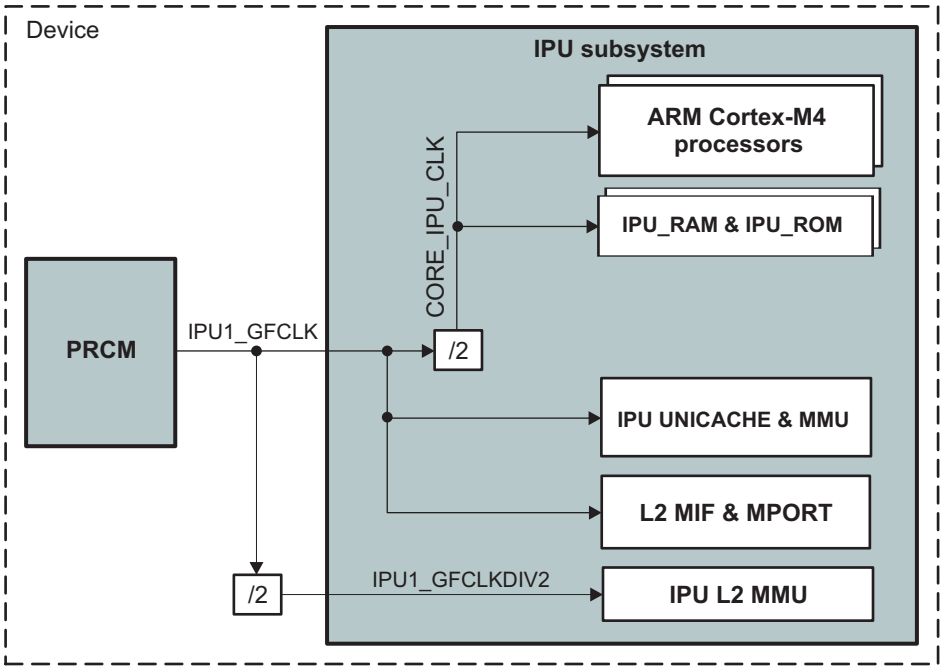
The IPU1_GFCLK itself can be derived from either:

- CORE_IPU_ISS_BOOST_CLK – main clock for IPU from DPLL_CORE, or
- DPLL_ABE_X2_CLK – alternative clock from DPLL_DDR

For more information, see Chapter 3, *Power, Reset, and Clock Management*.

Figure 5-3 shows the clocking scheme of the IPU subsystem.

Figure 5-3. IPU Subsystem Clocking Scheme



5.2.2 IPU Subsystem Reset Distribution

Three reset signals controlled by the PRCM module let the two Cortex-M4 processors and the rest of the IPU subsystem be reset independently. These three reset signals are: IPU_CPU0_RST, IPU_CPU1_RST, and IPU_RST. They can be controlled (assert/clear) by programming the following PRCM register bits:

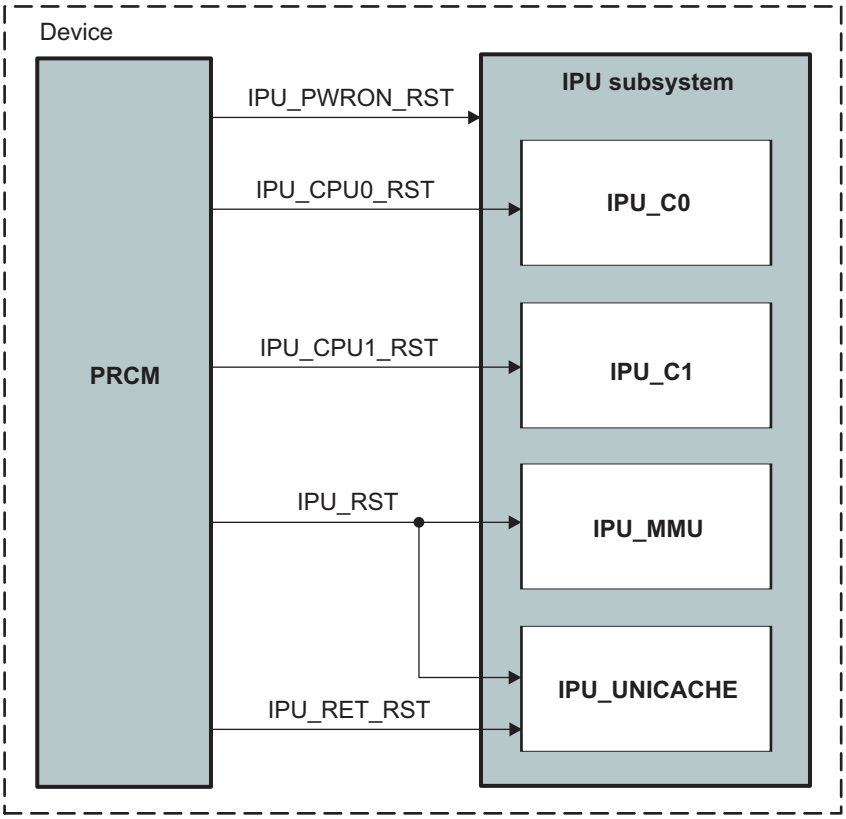
- RM_IPU_RSTCTRL[0] RST_CPU0 – reset control for IPU_C0.
- RM_IPU_RSTCTRL[1] RST_CPU1 – reset control for IPU_C1.
- RM_IPU_RSTCTRL[2] RST_IPU – reset control for IPU_UNICACHE and IPU_MMU.

At bootup, both IPU_CPU0_RST and IPU_CPU1_RST will be released (without any programming of the RST_CPU0 and RST_CPU1 bits) and thus, both Cortex-M4 cores will start executing their code. Subsequently, the RST_CPU0 and RST_CPU1 bits can be written to control the two CPU resets independently.

For details about IPU power-on reset sequence, see Chapter 3, *Power, Reset, and Clock Management*.

Figure 5-4 shows the reset scheme of the IPU subsystem.

Figure 5-4. IPU Subsystem Reset Scheme



5.3 Dual Cortex-M4 IPU Subsystem Functional Description

5.3.1 IPU Block Diagram

The IPU subsystem integrates the following blocks:

- Two fully identical Arm Cortex-M4 cores:
  - Revision r0p1
  - Integrated interrupt controller (NVIC)
- Memory system:
  - L1 memory system
    - Cache interface
    - 32KiB L1 unicache – IPU_UNICACHE
    - L1 MMU, serving the role of an attribute MMU (AMMU) for the unicache (and thus, labelled also as IPU_UNICACHE_MMU)
    - Profiling counters for the unicache – SCTM
  - L2 memory system
    - L2 MMU – IPU_MMU
    - 16KiB L2 ROM (IPU_ROM)
    - 64KiB L2 banked RAM – IPU_RAM
    - L2 master port – MPORT
    - L2 master interface – MIF
- Wake-up generator – IPU_WUGEN
- Two OCP ports at IPU boundary (connected to the L3_MAIN interconnect):
  - Master port – allows the IPU to access system resources (memories and peripherals)
  - Slave port – allows other requestors to access a part of the IPU internal memory space. For more information about which IPU modules are accessible via L3_MAIN, refer to Chapter 2, *Memory Mapping*.

The IPU subsystem is illustrated in Figure 5-1.

5.3.2 Cortex-M4 Core

5.3.2.1 Cortex-M4 Microprocessor

Refer to Arm *Cortex-M4 Technical Reference Manual* for a detailed description of the Cortex-M4 microprocessor. The Cortex-M4 version used in this device is the r0p1 revision.

5.3.2.2 Nested Vectored Interrupt Controller (NVIC)

Each Cortex-M4 core includes an Arm Nested Vectored Interrupt Controller (NVIC) which facilitates low-latency exception and interrupt handling. All exceptions are handled and prioritized within the NVIC. In order to facilitate parallel processing, the interrupt mapping is the same for the two cores. Each Cortex-M4 core receives the same interrupts, except for a few IPU internal interrupts. The interrupts are dynamically prioritized with 16 levels of priority defined for each core. In addition, Cortex-M4 also has a Non-Maskable Interrupt (NMI) input. When it is asserted, the NMI interrupt service routine is executed unconditionally.

The IPU interrupt mapping and an overview of the NVIC are presented in Chapter 12, *Interrupt Controllers*. For detailed description of NVIC functionality, refer to Arm *Cortex-M4 Technical Reference Manual*.

NOTE: The NVIC module is also referenced as IPU_Cx_INTC (x = 0, 1) throughout this device TRM.

5.3.2.3 Cortex-M4 Configuration in this Device

Different configurations are possible for the Cortex-M4 core. Table 5-4 lists the configuration for the Cortex-M4 in this device (same configuration for both cores).

Table 5-4. Cortex-M4 Configuration

Parameter	Implementation
Number of interrupts	96 external interrupts (in addition to the 16 Cortex-M4 internal interrupts)
Number of bits of interrupt priority	4 bits (defining 16 levels of priority)
Floating Point Unit (FPU)	Not included
Memory Protection Unit (MPU)	Not included
Trace Port Interface Unit (TPIU)	Not included
Istrumentation Trace Macrocell (ITM)	Not included
Embedded Trace Macrocell (ETM)	Not included
SW-DP or SWJ-DP	SW JTAG-DP interface (supported outside core level)
Data Watchpoint and Trace (DWT)	Included; 4 data watchpoints and event monitors
Flash-Patch and Breakpoint (FPB)	Included; 8 hardware breakpoints with program patching
Architectural clock-gating	Enabled

In adition, the following Cortex-M4 features are not supported in this device:

- The Cortex-M4 calibration register (SYST_CALIB) which is used to reload the SYSTICK timer to generate a fixed interval is not supported in the Cortex-M4 core within the IPU subsystem. The TENMS field is not available due to the different frequencies during voltage scaling. The SYSTICK timer stops counting when the processor is halted during debugging.
- The Cortex-M4 self-reset feature controllable through the SYSRESETREQ bit in the Cortex-M4 AIRCR register is not supported at the processor level. Writing to this bit will have no effect in the IPU subsystem. This feature is supported at the PRCM level. The global PRCM module can reset the complete IPU subsystem writing to the corresponding register.

For more information about the Cortex-M4 SYST_CALIB and AIRCR registers, refer to the Arm *Cortex-M4 Devices Generic User Guide*.

5.3.3 IPU Memory System

The IPU subsystem includes a 32KiB unified L1 cache (called IPU_UNICACHE) and a memory system built around the cache. The IPU memory system can be seen as a two-level hierarchical system:

- L1 memory system
- L2 memory system

The L1 memory system is the direct connection to the Cortex-M4 cores. It has the following features:

- Always runs on (1x) IPU_GFCLK frequency for both logic and memories
- Integrates a unified cache (IPU_UNICACHE) with a dedicated OCP port (for configuration and cache maintanance purposes). See Section 5.3.3.2 for more information about IPU_UNICACHE.
- Integrates L1 MMU which serves the role of an attribute MMU for the IPU_UNICACHE
  - L1 cache lookups provide L1 and L2 cache attributes
  - CPU lookup provides write policy attributes only
  - L1 lookup for address translation, and prefetch (if used)
  - L2 lookup for address translation, and prefetch (if used)
  - Integrated maintenance between L1 and L2
  - See Section 5.3.3.3 for more information about IPU_UNICACHE_MMU.
- Integrates a Subsystem Counter Timer Module (SCTM), with the ability to perform dynamic export of benchmarking data. See Section 5.3.3.4 for more information about the SCTM.

The L2 memory system has the following features:

- Always runs on (1x) IPU_GFCLK frequency for logic and (1/2x) for memories
- Master port (MPORT) for slave or DMA accesses
  - Built in coherency query to L1 cache to ensure transfers to L2 RAM are coherent
  - MPORT accesses must always have: logical = physical addresses
- Memory Interface (MIF) for internal memories
  - Decodes memory space
  - Decodes configuration space
- 16KiB ROM
- 64KiB banked RAM
- L2 MMU with lookup to L1
  - Address translation, and prefetch attributes

5.3.3.1 Cache Interface

The cache interface converts the data from the AHBLite protocol (supported by Cortex-M4) to the unicache. The AHBLite protocol is part of the AMBA® specification. The AHBLite protocol has been defined to support single master with several slaves. Four slave ports are required on the unicache to support the four buses from the Cortex-M4 cores (two per core).The I and D connections from each Cortex-M4 are multiplexed but the Cortex-M4 core internally prevents any conflict on this connection.

Because the Memory Protection Unit is not present within the Cortex-M4 in this device, default cache policies which are provided through the sideband signals are not used to access the cache. The cacheability is provided throught the AMMU present in the cache. AHBLite exclusive accesses are not supported at the unicache interface and an error is generated back to the interface when such access happens.

The ICode and DCode interface on each Cortex-M4 are a 32-bit AHBLite bus interface. Instruction fetches and vector fetches from code memory space (0x00000000 – 0x1FFFFFFF) are performed over the ICode bus. Data and debug accesses to code memory space (0x00000000 – 0x1FFFFFFF) are performed over the DCode bus. Control logic in the Dcode interface converts unaligned data and debug accesses into two or three (depending on the size and alignment of the unaligned access) aligned accesses.

The cache interface converts the AHBLite protocol to cache requests, converting burst access to sequential access to the cache. Although the Cortex-M4 processor supports unaligned transfers, there is no unaligned transfer on this bus, because the bus interface on the processor core converts the unaligned transfers into aligned transfers (adding extra cycles if required).

Since a bus multiplexer is used between I and D bus, the transfers cannot take place at the same time on these buses and the Cortex-M4 handles the arbitration between the two buses. The data bus (D bus) accesses are given higher priority.

5.3.3.2 L1 Unified Cache (IPU_UNICACHE)

The unicache provides a multi-access cache which is scalable for number of slaves and size. Integrated coherency management is provided by sideband signals of slave interfaces, as well as a dedicated maintenance interface. The cache architecture is two pipeline stages, where the first stage is a tag memory access, and the second stage is a banked data array. Each slave has a dedicated tag to enable high-frequency access with no dynamic arbitration between slaves, and the banked data array reduces the probability of conflicts between slaves.

5.3.3.2.1 IPU_UNICACHE Configuration in the Device

Table 5-5 describes the IPU_UNICACHE configuration in the device.

Table 5-5. IPU_UNICACHE Configuration

Parameter	Value
Way	4
Size	32 KiB
Bank elements	32 bits
Bank number	16
Slave interface data size	32 bits
Master interface data size	64 bits
Line size	256 bits
MMU lookup	Included
Number of slaves	4
Number of masters	1
Number of fill/prefetch buffers	Four prefetch buffers
Slave types	Cache interface

5.3.3.2.2 IPU_UNICACHE Maintenance

The IPU_UNICACHE allows basic maintenance operations, which are performed through its OCP configuration port:

1. Preload (CACHE_MAINT[0] PRELOAD)
2. Lock/Unlock (CACHE_MAINT[1] LOCK; CACHE_MAINT[2] UNLOCK)
3. Clean (CACHE_MAINT[3] CLEAN)
4. Invalidate (CACHE_MAINT[4] INVALIDATE)

Maintenance of the cache is performed between the start and end addresses (configurable via the CACHE_MTSTART[31:0] START_ADDR and CACHE_MTEND[31:0] END_ADDR bit fields, respectively). This allows for direct control of memory regions. All maintenance operations occur in the background and can generate an interrupt (if enabled via the CACHE_MAINT[5] INTERRUPT bit) when they complete. Such operations are protected by software semaphore, because only one operation at a time can be performed. The maintenance operations can also be performed using MMU small entries.

5.3.3.3 L1 MMU (IPU_UNICACHE_MMU)

The IPU_UNICACHE_MMU serves the role of an attribute MMU (AMMU) for the unicache. It provides the multi-access cache with region-based address translation, read/write control, access type control, and multi-level cache maintenance. The IPU_UNICACHE_MMU regions are register entry based to enable low-latency lookup and it is not a table-walking MMU.

The IPU_UNICACHE_MMU is connected directly to the unicache. There is a dedicated interface for a CPU to query cache policies for writes in supporting a pipeline write buffer that follows the write policies of the IPU_UNICACHE_MMU.

5.3.3.3.1 IPU_UNICACHE_MMU Configuration in the Device

Table 5-6 describes the IPU_UNICACHE_MMU configuration in the device.

Table 5-6. IPU_UNICACHE_MMU Configuration

Parameter	Values
Number of large pages	Four entries
Size of large pages	512 MiB or 32 MiB (configurable)
Number of medium pages	Two entries
Size of medium pages	256 KiB or 128 KiB (configurable)
Number of small pages	10 entries
Size of small pages	16 KiB or 4 KiB (configurable)
Number of patch pages	Not included
Size of line pages	256-bit
Number of comparison interfaces	4
Number of comparator sets	1
Write pipeline data comparison	Disabled
Number of IPU_UNICACHE maintenance interfaces	3
Size of entry address	32-bit

5.3.3.3.2 Page Attributes

As can be seen in Table 5-6, IPU_UNICACHE_MMU supports different page sizes: large, medium, and small. The number of large pages, number of medium pages, etc., is defined at design time. The size of the pages is configurable in the following IPU_UNICACHE_MMU registers:

- CACHE_MMU_LARGE_POLICY_i[1] SIZE
- CACHE_MMU_MED_POLICY_j[1] SIZE
- CACHE_MMU_SMALL_POLICY_k[1] SIZE

The different MMU page sizes can be used to create smaller policies within a larger region.

The logical source address is configured in:

- CACHE_MMU_LARGE_ADDR_i[31:25] ADDRESS
- CACHE_MMU_MED_ADDR_j[31:17] ADDRESS
- CACHE_MMU_SMALL_ADDR_k[31:12] ADDRESS

The logical source translated address is configured in:

- CACHE_MMU_LARGE_XLTE_i[31:25] ADDRESS
- CACHE_MMU_MED_XLTE_j[31:17] ADDRESS
- CACHE_MMU_SMALL_XLTE_k[31:12] ADDRESS

When the SIZE bit is set to 0, all the bits in the ADDRESS bit field of the corresponding logical source address and logical source translated address can be used.

When the SIZE bit is set to 1, the ADDRESS bit field of the corresponding logical source address and logical source translated address must be programmed only with addresses that can address the size of a second possible page.

5.3.3.3.3 Policy Support

The following policies are supported by the IPU_UNICACHE_MMU:

- L1 global execute only policy
- L1 global read only policy
- L1 global volatile policy
- Unique L1 and L2 cacheable / non-cacheable policies
- Unique L1 and L2 posted / non-posted policies
- Unique L1 and L2 allocate / non-allocate policies
- Unique L1 and L2 write through / write back policies
- L2 exclusion policy
- L1 or L2 address translation policy
- L1 or L2 prefetch policy

The various policies can be configured in the following registers:

- CACHE_MMU_LARGE_POLICY_i – for large pages
- CACHE_MMU_MED_POLICY_j – for medium pages
- CACHE_MMU_SMALL_POLICY_k – for small pages

5.3.3.4 Subsystem Counter Timer Module (SCTM)

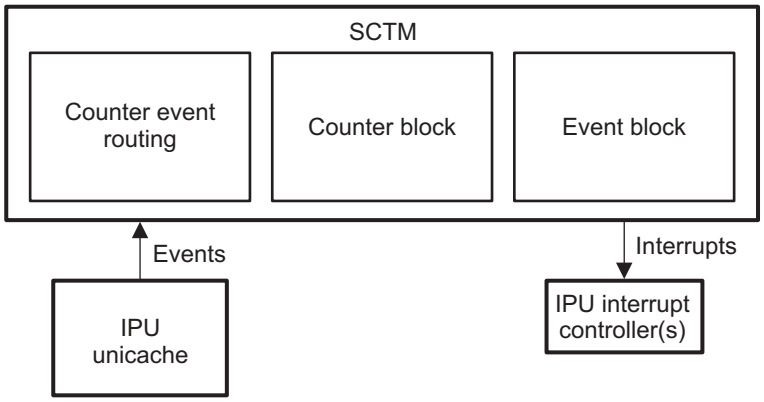
The Subsystem Counter Timer Module (SCTM) is a generic profile counter and timer module that provides the following functions:

- Counter functions:
  - Input events counting
  - Two counter modes:
    - Event counting
    - Duration counting
  - Counter chaining
- Timer functions:
  - Periodic intervals generation
  - Two timer modes:
    - Run-once mode
    - Restart mode
  - Events and/or interrupt generation

The SCTM has eight counters (two of which have timer functions). There are input events going into the SCTM and interrupt events going to the IPU INTC. For more information about the counter configuration and the SCTM input events, see Section 26.7.1, *IPU Subsystem Performance Monitoring*, in Chapter 26, *On-Chip Debug Support*. For more information about the mapping of the SCTM interrupt event signals to IPU interrupt controller inputs, see Section 12.3.3, *Interrupt Requests to IPU_Cx_INTC*, in Chapter 12, *Interrupt Controllers*.

Figure 5-5 shows the SCTM block diagram.

Figure 5-5. SCTM Block Diagram



5.3.3.4.1 Counter Functions

5.3.3.4.1.1 Input Events

Signals from within the subsystem are routed to the SCTM and used to control the counters and timers in the module. The routing of the input events from the module boundary to an individual counter is accomplished through an input event multiplexer and is controlled by the CACHE_SCTM_CTCR_WT_i[20:16] INPSEL and CACHE_SCTM_CTCR_WOT_j[20:16] INPSEL bit fields.

NOTE: For more information about input events to the module boundary, see Section 26.7.1.2, *Cache Events*, in Chapter 26, *On-Chip Debug Support*.

5.3.3.4.1.2 Counters

There are individual 32-bit counters in the SCTM. The counters count when the input event signals are asserted.

5.3.3.4.1.2.1 Counting Modes

The counters in the SCTM support two mutually exclusive counting modes:

- Event mode: The counter increments each time a rising edge is detected on the designated input event signal.
- Duration mode: The counter continually increments when the event input is asserted.

5.3.3.4.1.2.2 Counter Overflow

When the counter reaches the terminal value (0xFFFFFFF) it wraps and continues to increment. This is considered a timer overflow condition. The CACHE_SCTM_CTCR_WT_i[6] OVRFLW and CACHE_SCTM_CTCR_WOT_j[6] OVRFLW bits indicate that overflow has occurred. The overflow bit can be cleared by reading it. When chained, only the high-order counter overflows.

5.3.3.4.1.2.3 Counters and Processor State

The counters can be configured to alter their behavior based on the state of the CPU. The CACHE_SCTM_CTCR_WT_i[4] FREE and CACHE_SCTM_CTCR_WOT_j[4] FREE bits determine whether the counter will continue to operate when the processor enters debug halt state. When the FREE bit is set to 0, the counter stops incrementing while the debug halt input from the CPU is asserted. Normal operation resumes when the processor exits the debug halt state and the debug halt input is deasserted. When the FREE bit is set to 1, the state of the debug halt input is not used to control counter operation.

The CACHE_SCTM_CTCR_WT_i[5] IDLE and CACHE_SCTM_CTCR_WOT_j[5] IDLE bits determine whether the counter will continue to operate when the processor enters idle state (the processor is no longer executing instructions and is waiting for a wake-up event). When the IDLE bit is set to 0, the counter stops incrementing while the idle input from the CPU is asserted. Normal operation resumes when the processor exits the idle state and the idle input is deasserted. When the IDLE bit is set to 1, the state of the idle input is not used to control counter operation.

5.3.3.4.1.2.4 Chaining Counters

The individual 32-bit counters in the SCTM can be chained with an adjacent counter to form a 64-bit counter. Counters chained to a counter across an even-odd index boundary with the even counter contain the least-significant 32 bits of the 64-bit pairing. For example, counters 1 and 0 can be paired and counter 1 will contain bits 63:32 and counter 0 will contain bits 31:0. The high-order counter increments by 1 each time the low-order counter wraps.

Counters are chained by setting the CACHE_SCTM_CTCR_WT_i[2] CHAIN or CACHE_SCTM_CTCR_WOT_j[2] CHAIN bit for both counters. When chained, the counter control for both counters is taken from the CACHE_SCTM_CTCR_WT_i or CACHE_SCTM_CTCR_WOT_j register of the low-order counter. Other than the CHAIN bit, all other bits in the high-order CACHE_SCTM_CTCR_WT_i or CACHE_SCTM_CTCR_WOT_j register are ignored.

Chained counters can function only in counter mode. Timer mode is not supported.

The CACHE_SCTM_CTCR_WT_i[7] CHNSDW and CACHE_SCTM_CTCR_WOT_j[7] CHNSDW bits are used to indicate that a counter can provide atomics accessed when chained. These bits are valid only for counters with even indexes (the lower half of a 64-bit counter pair). When these bits are set, the counter can shadow the value of the lower half of the chained counter value at the same time the upper half of the counter is read. The shadowed value (not the current value) is returned when the value of the low-order counter is read. Therefore, when a chained counter has atomic read capability, an atomic counter value can be obtained simply by reading the high-order counter first, followed by the low-order counter. This order must always be observed to prevent reading stale counter values from the low-order counter. When counters are functioning independently, the shadow feature is deactivated and a read of the counter always returns the current value.

5.3.3.4.1.2.5 Enabling and Disabling Counters

After the counter is correctly configured, it can be started by setting the CACHE_SCTM_CTCR_WT_i[0] ENBL or CACHE_SCTM_CTCR_WOT_j[0] ENBL bit. At this point, the counter begins incrementing under the control of the configured event input. The counter can be disabled (counting stops) at any time by clearing the ENBL bit. Counters can be enabled and disabled dynamically during application flow.

Counters can also be enabled and disabled as groups through the CACHE_SCTM_CTGNBL register. This register provides control of the individual counter-enable in groups. This allows an application to enable or disable groups of counters in lockstep by setting corresponding bits to 1 (bit number corresponds to counter number).

5.3.3.4.1.2.6 Resetting Counters

The counters can be reset to their initial value (0x00000000) by writing 1 to the CACHE_SCTM_CTCR_WT_i[1] RESET or CACHE_SCTM_CTCR_WOT_j[1] RESET bit. If the counter is chained, the high-order and low-order counters are reset when the RESET bit is written for the low-order counter.

Counters can also be reset as groups through the CACHE_SCTM_CTGRST registers. These registers provide control of the individual counter reset in groups. This allows an application to reset groups of counters in lockstep.

5.3.3.4.2 Timer Functions

Counters 0 and 1 in the SCTM can function as timers. When operating as timers, interrupts and/or debug input events are generated when the value of the counter reaches a designated interval.

5.3.3.4.2.1 Periodic Intervals

The interval for a timer is contained in the CACHE_SCTM_TINTVLR_i register. There is a CACHE_SCTM_TINTVLR_i register for every timer-capable counter in the SCTM. Timers are initialized to 0. When the corresponding CACHE_SCTM_CTCNTR_k increments and matches the values designated in CACHE_SCTM_TINTVLR_i, the timer is considered to be triggered and events configured in CACHE_SCTM_CTCR_WT_i are generated.

Timers can function in one of two mutually exclusive modes:

- Run-once mode: The timer stops after the first interval match and is not re-enabled until the timer is reset to the initial value (0) by setting the CACHE_SCTM_CTCR_WT_i[1] RESET bit to 1.
- Restart mode: The timer automatically resets to the initial value (0) each time the designated interval is reached.

The CACHE_SCTM_CTCR_WT_i[10] RESTART bit is used to configure the timer mode.

5.3.3.4.2.2 Event Generation

Timers can generate interrupts and debug events. Interrupts are routed from the module boundary to the interrupt controller(s) in the subsystem. Debug events are routed as triggers to debug logic within the subsystem.

The generation of the interrupts is controlled by the CACHE_SCTM_CTCR_WT_i[8] INT bit. The generation of debug events is controlled by the CACHE_SCTM_CTCR_WT_i[9] DBG bit. The INT and DBG bits can be set simultaneously and both signals are generated on interval match.

If neither INT nor DBG is set, the timer function is disabled and the counter functions as a regular counter.

5.3.3.5 L2 MMU (IPU_MMU)

An additional MMU provides address translation for the accesses done from the IPU subsystem to the L3_MAIN interconnect. The main characteristics of this MMU are:

- 32 entries
- Compatible with Armv6 architecture MMU translation tables (protection bits not used)
- Page-based or access-based endianness conversion
- Two-level descriptor hierarchy
- One intermediate page table
- Four page sizes (16 MiB, 1 MiB, 64 KiB, 4 KiB)
- Page table alignment on 128-byte boundary for Arm11 compatibility

The configuration of the IPU_MMU can be done from one of the Cortex-M4 cores or from the L3_MAIN interconnect slave port. The accesses done to configure the IPU_MMU cannot be part of a burst access.

For more information about the IPU_MMU, see Chapter 15, *Memory Management Units*.

5.3.3.5.1 IPU_MMU Behavior on Page-Fault

Table 5-7. IPU_MMU Behavior on Page-Fault

Table-Walker Enabled	Table-Walker Disabled
Application	
Use Table-Walker to find translation. Update TLB cache if successful, set TRANSLATIONFAULT bit and interrupt if not. The following bits are used for the purpose: MMU_IRQENABLE[1] TRANSLATIONFAULT and MMU_IRQSTATUS[1] TRANSLATIONFAULT.	Set TLBMISS bit and interrupt and stall. The following registers are used for the purpose: MMU_IRQENABLE[0] TLBMISS and MMU_IRQSTATUS[0] TLBMISS.
Debug	

Table 5-7. IPU_MMU Behavior on Page-Fault (continued)

Table-Walker Enabled	Table-Walker Disabled
Application	
Use Table-Walker to find translation. Update TLB cache if successful (only if the MMU_CNTL[3] EMUTLBUPDATE bit is set), generate in-band bus error if not.	Set EMUMISS bit and interrupt and stall. The following bits are used for the purpose: MMU_IRQENABLE[2] EMUMISS and MMU_IRQSTATUS[2] EMUMISS.

The MMU fault interrupt line is connected to both Cortex-M4 cores (at IPU_IRQ_16 interrupt line) and is also propagated outside of the IPU subsystem and connected to an IRQ_CROSSBAR input (IRQ_CROSSBAR_395). The user can route this interrupt to other device processors by properly programming the corresponding Control Module registers.

The default behavior of the IPU_MMU previously described can be overridden by setting the MMU_GP_REG[0] BUS_ERR_BACK_EN bit to 1. Once this bit is set, all MMU faults (including TLB miss) return a bus error to the IPU subsystem (interrupt event XLATE_MMU_FAULT). This allows the end user to quickly establish the cause of the MMU fault by having appropriate code in the ISR.

5.3.3.6 L2 MPORT

The MPORT provides three functions:

- Configuration of the L2 MMU – only allowed during functional reset or IDLE
- Data transfer to L2 banked RAM
- Coherency control between L1 cache and L2 memory system:
  - a. Query on L1 cache to determine if line exists
  - b. Query interface cleans/invalidates line
  - c. Once query is complete, MPORT can complete transfer to L2 RAM

5.3.3.7 ECC Implementation

In IPU subsystem, ECC support is implemented for L1 cache and L2 RAM. This enables Single Error Correction (SEC) and Double Error Detection (DED) in the memories. Since read-modify-write support is not implemented for partial accesses, individual ECC code is generated for each granular memory block. This is how each row is organized into data and ECC blocks:

- **L1 Data**: Access granularity to L1 data cache memory is 8 bits. Hence, a 32-bit row is divided into 4 blocks and 5-bit ECC code is generated for each block (see Figure 5-6)
- **L1 Tag**: As shown in Figure 5-7, ECC code is generated individually for flag and tag bits. Lock and LRA flags can be accesses exclusively and hence ECC support for this bit is not possible.
- **L2 RAM**: For L2 RAM, 64-bit data can be accessed with a granularity of 8 bits and hence, 8 ECC blocks each of width 5-bits are implemented (see Figure 5-8).

Figure 5-6. L1 Data ECC

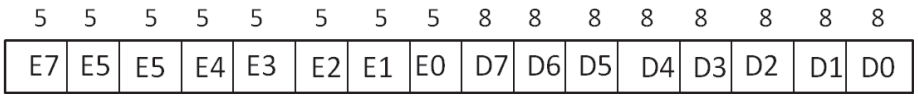


Figure 5-7. L1 Tag ECC

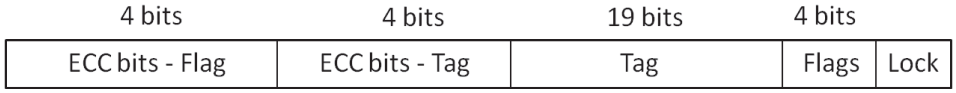
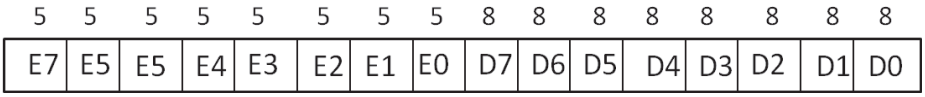


Figure 5-8. L2 RAM ECC



For all the ECC supported memories, error is detected/corrected for individual blocks and status is updated in the ECC registers in L1 unicache. As the errors can occur back-to-back, the registers are updated with the latest information. On SEC or DED event, interrupt is generated and propagated to both the Cortex-M4 cores. Also, in case of DED event in L1 data and L2 RAM, the error response is propagated along with the data so that it does not corrupt the system. If granular accesses are made to these memories, DED and error will still be reported for error occurring at other locations in the memory row. This is because entire row is read from the memory even for granular accesses. Moreover, DED event in L1 tag memory results in a tag miss and data is fetched from an external memory. The tag will still result in the hit if DED occurs in a way other than the one where hit condition occurred. However, DED interrupt will still be propagated to the register. In case of DED event in data or tag memory, it is recommended that the cache is invalidated by M4 in the interrupt service routine. For silicon validation, ECC blockimplementation also supports the individual masking of ecc or data block. On enabling the code/ data mask register, the data or ECC block can be corrupted and SEC / DED event can be validated accordingly.

5.3.4 IPU Power Management

5.3.4.1 Wake-Up Generator (IPU_WUGEN)

The IPU_WUGEN provides efficient power management by generating a wake-up request to the PRCM module when the IPU subsystem is in standby or full-idle mode and at least one enabled external interrupt becomes active. Upon assertion of the IPU wake-up request, the PRCM module will recover IPU functional clock (IPU_GFCLK) and thus, IPU will make a transition to active mode. In general, the IPU_WUGEN allows:

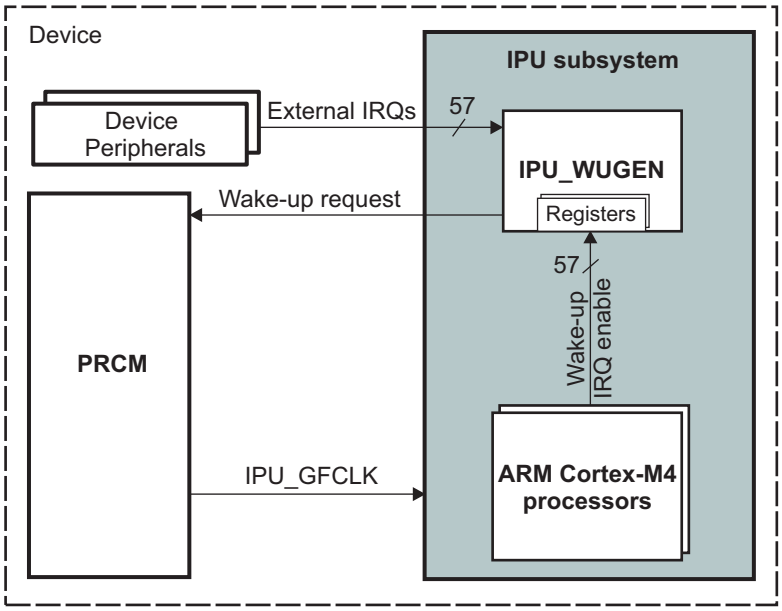
- Gating of the IPU subsystem clock dynamically, thus reducing power consumption
- Simplifying of dependencies in the PRCM module

Two retention registers are used to control which interrupts will cause a wake-up request generation:

- WUGEN_MEVT0 – wake-up interrupt enable for interrupts mapped to the IPU_IRQ_[47:23] interrupt lines.
- WUGEN_MEVT1 – wake-up interrupt enable for interrupts mapped to the IPU_IRQ_[79:48] interrupt lines.

Figure 5-9 is an overview of the IPU_WUGEN.

Figure 5-9. IPU_WUGEN Overview



5.3.4.2 Cortex-M4 Local Power Management

The Cortex-M4 processor provides two sleep modes to reduce power consumption:

- Sleep mode
- Deep-sleep mode

During sleep mode, the system clock can be stopped but the free running clock input should still be running to allow the processor to be woken by an interrupt or an event. The support of the sleep mode is done within the IPU subsytem. Deep-sleep mode also stops the processor clock but this can also be supported by the PRCM module. A combined signal is generated from the two Cortex-M4 processors in deep-sleep mode to initiate another power state and let the PRCM module handle the next power states. At this time, software must ensure that all IPU_UNICACHE background operations (for example, maintenance) are complete before PRCM asserts an idle request.

The sleep modes are invoked by wait for interrupt (WFI) or wait for event (WFE) instructions. For more information about entering/exiting sleep modes, see the Arm *Cortex-M4 Devices Generic user Guide*.

5.3.4.3 STANDBY and IDLE Protocols

The IPU subsystem supports the STANDBY and IDLE power-managment protocols. The following registers (part of the IPU_WUGEN register space) can be used to configure the different STANDBY and IDLE modes:

- STANDBY_CORE_SYSCONFIG[1:0] STANDBYMODE bit field. The default is smart wake-up STANDBY mode.
- IDLE_CORE_SYSCONFIG[1:0] IDLEMODE bit field. The default is smart wake-up IDLE mode.

NOTE: It is not recommended to change the default values of the STANDBY and IDLE modes in software.

5.3.4.4 Power Domains

The IPU subsystem is divided into two power domains (PD_IPU and PD_COREAON), which are controlled by the PRCM module.

The PD_IPU power domain is the main power domain and includes all the IPU subsystem components (two Arm Cortex-M4 processors, IPU_UNICACHE, IPU_ROM and IPU_RAM memories, and emulation\debug modules) except the IPU_WUGEN.

The PD_COREAON power domain is an always-on power domain. The PD_COREAON power domain contains the IPU_WUGEN, which generates a wake-up request from external interrupts. By this separate PD_COREAON power domain, the wake-up request can be generated even when the PD_IPU power domain is in OFF or RET state.

For information about the PD_IPU and PD_COREAON power domains, see Chapter 3, *Power, Reset, and Clock Management*.

5.3.4.5 Voltage Domain

The IPU subsystem is located within the CORE voltage domain (VD_CORE). All IPU logic (Cortex-M4 cores, IPU_UNICACHE/MMU, IPU_WUGEN, etc...) is fed by VD_CORE. All IPU memory arrays are fed by on-chip memory (SRAM) LDO dedicated to the CORE domain – SLDO_CORE.

For information about VD_CORE and SLDO_CORE, see Chapter 3, *Power, Reset, and Clock Management*.

5.3.4.6 Power States and Modes

Table 5-8 lists the different power modes and the expected states for each power domain.

Table 5-8. IPU Subsystem Power Modes

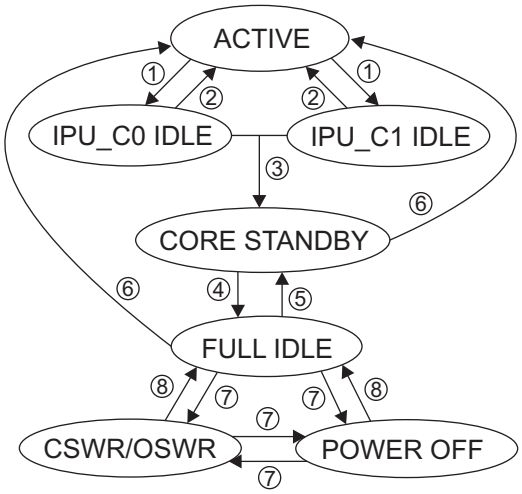
		Activity			Power Status	
	Functional Domain	Core		IPU subsystem	PD_IPU Power Domain	PD_COREAON Power Domain
	Modules included	IPU_C0	IPU_C1	IPU_WUGEN		
Power modes	Active	Active	Active	Active	ON	ON
	IPU_C0 idle	Idle	Active	Active	ON	ON
	IPU_C1 idle	Active	Idle	Active	ON	ON
	Core standby	Idle	Idle	Active	ON	ON
	Full idle	Idle	Idle	Idle	ON	ON
	CSWR	Idle	Idle	Idle	ON/LOWV	ON/LOWV
	OSWR	Idle	Idle	Idle	RET	ON
	Power off	Idle	Idle	Idle	OFF	OFF

The different power modes and their features are:

- Active mode: All functional domains are operative.
- IPU_C0 and IPU_C1 idle mode:
  - Only the CPU core is idled (when running WFE/WFI instructions).
  - Only one Cortex-M4 core can be in this mode. Interrupts or events can wake-up the core.
  - Can go into sleep or deep-sleep mode. Potentially, both cores can be in sleep mode.
  - When both cores are in deep-sleep mode, a standby request is sent to the PRCM module.
  - Software must ensure that all IPU_UNICACHE background operations (for example, maintenance) are complete before the PRCM module asserts an idle request.
- Core standby mode
  - In this mode, both cores in the CORE functional domain are in idle mode (an interrupt cannot wake up either of the cores).
  - The PRCM module must have acknowledged its acceptance by an MWait signal.
  - After this handshake, all power management is under the control of the PRCM module.
- Full-idle mode:
  - In this mode, the IPU subsystem functional domain is also idled.
  - This mode is only reached by PRCM's initiative. When PRCM finds all conditions (by hardware and/or software configuration) are met to idle, it sends idle requests to the local PRCM, and it affects each functional domain.
  - After coming to this mode, power states can be moved deeper.
- CSWR mode:
  - In this mode, the logic supply voltage is lowered to reduce static power consumption by leakage current. Logic power switch in the PD_IPU power domain is still closed (ON); thus, all logic states are retained.
  - IPU_WUGEN is still active and can generate wake-up request to PRCM.
  - L1 and/or L2 memories can go independently in retention depending upon the settings done at PRCM level.
- OSWR mode:
  - In this mode, logic power switch is open (OFF); thus, most logic states are lost except the retention logics. Voltage level is typically lowered, but not necessarily.
  - Cache/L2 memory contents are retained
  - WUGEN is still active and can generate wakeup request to PRCM.
- Power off mode:
  - The voltage source is shut down. The logic states, including the retention logic, are lost.
  - The IPU_WUGEN is not operating and only the PRCM module can trigger the IPU subsystem wakeup.
  - Reset must be applied to the IPU subsystem to restart the two Cortex-M4 processors and the memory subsystem.

Figure 5-10 shows the power mode transitions in the IPU subsystem.

Figure 5-10. IPU Power Mode Transitions



Legend for Figure 5-10:

- 1 = WFE/WFI instruction
- 2 = Events/interrupts
- 3 = "Deep-sleep" generated
- 4 = L1/L2/IPU_WUGEN functional domain idled
- 5 = Wakeup (OCP)
- 6 = Wakeup (IRQ)
- 7 = PRCM
- 8 = PRCM or wakeup

5.3.5 IPU Interprocessor Communication (IPC)

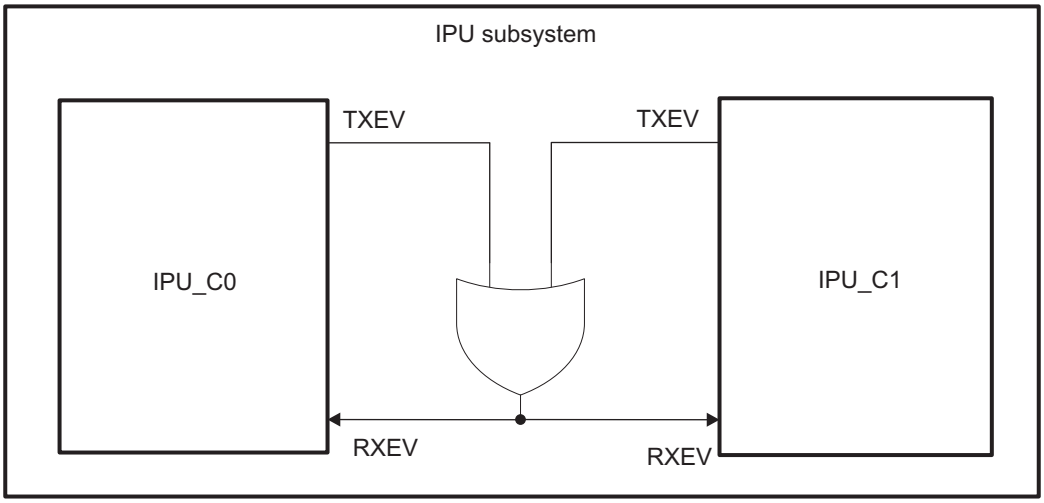
5.3.5.1 Use of WFE and SEV

The IPU subsystem provides a multiprocessor communication interface for synchronizing tasks. The Arm processors have one output signal, TXEV (transmit event), for sending events and one input signal, RXEV (receive event), for receiving events. Figure 5-11 shows how TXEV and RXEV are connected in the IPU subsystem.

When a WFE instruction is executed, the processor enters into sleep mode waiting for an event and continues instruction execution when an external event is received. With an SEV (send event) instruction, one processor can wake up the other processor, which is in sleep mode.

The WFE and SEV instructions can help reduce the number of iterations around a lock acquire loop (a spinlock), and thereby reduce power consumption. The basic mechanism involves an observer that is in a spinlock executing a WFE instruction, which suspends execution on that observer until an asynchronous exception or an explicit event (sent by an observer using the SEV instruction) is seen by that observer. The observer that holds the lock uses the SEV instruction to send an event after a lock is released.

Figure 5-11. Event Communication Connection in IPU Subsystem



5.3.5.2 Use of Interrupt for IPC

Each Cortex-M4 core can interrupt the other Cortex-M4 core by setting up an interrupt register (CORTEXM4_CTRL_REG). This register is used to trigger the corresponding 'per core' HWSEM_M4_IRQ interrupt (interrupt number 19). Because the priority level for that interrupt can be defined, it is possible to choose the task level at which the interrupt will run. For example, if IPU_C0 was active and IPU_C1 was idle (WFI state), when IPU_C0 completes its task it sets the bit for IPU_C1 in the control register (CORTEXM4_CTRL_REG[16] INT_CORTEX_2) and goes into sleep mode. IPU_C1 wakes up seeing this interrupt, and starts running its task. After the completion of its task, IPU_C1 sets the interrupt for IPU_C0 (CORTEXM4_CTRL_REG[0] INT_CORTEX_1), and then goes into WFI state. This kind of handshake ensures that if IPU_C0 and IPU_C1 are accessing the same resources (memory, registers etc.), only one of the CPUs at a time is active.

5.3.5.3 Use of the Bit-Band Feature for Semaphore Operations

The two Cortex-M4 cores share the same memory system, and it is possible to use the bit-band feature to carry semaphore operations. Because the bit-band alias writes are locked read-modify-write transfers, provided that all tasks changed only the lock bit representing themselves, the lock bits of other tasks are not lost, even if two tasks try to write to the same memory location at the same time.

Each Cortex-M4 core supports two bit-band regions.

Bit-band 1 applies to the virtual address space 0x2000 0000–0x200F FFFF (1 MiB). This virtual address space can be mapped to any physical address and bit-banding will apply to that region. It is recommended that the user map the L2 IPU_RAM (64 KiB) to this virtual space and use it only for bit-banding operations. If required, the user can define other available small and medium pages over and above the L2 IPU_RAM virtual space and further extend the use of the bit-band feature.

Bit-band 2 applies to the virtual address space 0x4000 0000–0x400F FFFF (1 MiB). The first 16 KiB of this space (0x4000 0000–0x4000 3FFF) are already reserved for small (one page) pages and cannot be remapped by software. The bit-band alias that corresponds to this 16-KiB region (0x4200 0000–0x4207 FFFF) must also be treated as reserved and no access should be made. The rest of bit-band 2 can be used by appropriately defining the available small and medium pages. In this device, because it is likely that during normal AMMU programming all of L3_MAIN is mapped to this region, it is highly recommended that the user use only bit-band 1 for all purposes and bit-band 2 only if it is necessary.

5.3.5.4 Cortex-M4 Private Memory Space

Each Cortex-M4 processor has its own memory space (address range 0xE000_1000 – 0xE00F_FFFF), inaccessible by the other processor (and also by the L3_MAIN interconnect). Because the registers in this memory space are not shared, they do not require the bit-band feature (semaphore) to read from and write to them.

5.3.6 IPU Memory Mapping

The Cortex-M4 processor has a predefined memory map that specifies which bus interface is to be used when a memory location is accessed. The Cortex-M4 also features a bit-band support from which a special region is defined. For more details about the Cortex-M4 memory model, refer to Arm *Cortex-M4 Technical Reference Manual*.

IPU memory map is presented in Chapter 2, *Memory Mapping*.

5.3.7 IPU Boot Configuration

IPU is the boot master of this device. The device initialization is described in Chapter 25, *Initialization*.

The IPU boot location is controlled via two Control Module registers:

- CTRL_CORE_CORTEX_M4_MMUADDRTRANSLTR[19:0] CORTEX_M4_MMUADDRTRANSLTR: Used to set the physical translated address for IPU AMMU
- CTRL_CORE_CORTEX_M4_MMUADDRLOGICTR[19:0] CORTEX_M4_MMUADDRLOGICTR: Used to set the logical source address for IPU AMMU

By default, two AMMU pages are enabled:

- Small page-0: Translates the 16KB address range from CORTEX_M4_MMUADDRLOGICTR to (CORTEX_M4_MMUADDRLOGICTR + 0x3FFF). If CORTEX_M4_MMUADDRLOGICTR is set to 0x00000, page-0 will control the boot location. If the boot location needs to be mapped to the L2 RAM (0x5502_0000), then CORTEX_M4_MMUADDRTRANSLTR needs to be set to 0x55020. This page is set as non-cacheable at reset.
- Small page-1: Loaded with the physical address of the IPU AMMU configuration registers (0x5508_0000 - 0x5508_0FFF), which is mapped to the virtual address range from 0x4000_0000 to 0x4000_0FFF. This page is also set as non-cacheable at reset.

NOTE: Small page-1 is by default 4KB. Software has to modify it to 16KB to cover L2MMU/WUGEN masks.

For IPU to boot from any location in L3:

1. Provide boot address through CORTEX_M4_MMUADDRTRANSLTR to AMMU page-0 (CORTEX_M4_MMUADDRLOGICTR set to 0x00000). Keep L2 MMU disabled (or enable L2 MMU but keep the same translation for 0x0; otherwise there will be L2-MMU page-walks / page-faults).
2. Set CORTEX_M4_MMUADDRTRANSLTR to 0x00000 (or any value). Host CPU re-programs AMMU page to map 0x0 virtual address to a physical L2 RAM / L3 location. Only after the programming is complete, Cortex-M4 reset is released. L2 MMU as described above.
3. Set CORTEX_M4_MMUADDRTRANSLTR to 0x00000 (no translation). Host CPU programs L2 MMU to do the address translation for 0x0.

For IPU to boot from L2 RAM:

1. This must be done through AMMU page-0. Use either (1) or (2) as described above.

5.3.8 IPU Debug and Emulation Features

The IPU subsystem (and the Cortex-M4 core, in particular) provides some debug and emulation capabilities. They are described in Chapter 26, *On-Chip Debug Support*.

5.4 Dual Cortex-M4 IPU Subsystem Register Manual

5.4.1 IPU Subsystem Instance Summary

Table 5-9 summarizes the IPU subsystem instances.

Table 5-9. IPU Subsystem Instance Summary

Module Name	Base Address (IPU Private Access)	Base Address (L3_MAIN Interconnect)	Size
IPU_UNICACHE_CFG	0x5508 0000	N/A	256 B
IPU_UNICACHE_SCTM	0x5508 0400	N/A	1 KiB
IPU_UNICACHE_MMU (AMMU)	0x5508 0800	0x5888 0800	2 KiB
IPU_WUGEN	0x5508 1000	N/A	4 KiB
IPU_MMU	0x5508 2000	0x5888 2000	4 KiB
IPU_C0_INTC(1)	0xE000 E000	N/A	4 KiB
IPU_C1_INTC(1)	0xE000 E000	N/A	4 KiB
IPU_C0_RW_TABLE(1)	0xE00F E000	N/A	4 KiB
IPU_C1_RW_TABLE(1)	0xE00F E000	N/A	4 KiB

(1) Different view from each Cortex-M4

5.4.2 IPU_UNICACHE_CFG Registers

5.4.2.1 IPU_UNICACHE_CFG Register Summary

Table 5-10. IPU_UNICACHE_CFG Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	Physical Address (IPU Private Access)
CACHE_CONFIG	RW	32	0x0000 0004	0x5508 0004
CACHE_INT	RW	32	0x0000 0008	0x5508 0008
CACHE_OCP	RW	32	0x0000 000C	0x5508 000C
CACHE_MAINT	RW	32	0x0000 0010	0x5508 0010
CACHE_MTSTART	RW	32	0x0000 0014	0x5508 0014
CACHE_MTEND	RW	32	0x0000 0018	0x5508 0018
CACHE_CTADDR	RW	32	0x0000 001C	0x5508 001C
CACHE_CTDATA	RW	32	0x0000 0020	0x5508 0020
ECC_CFG	RW	32	0x0000 0024	0x5508 0024
L1DATA_ERR_INFO	RW	32	0x0000 0028	0x5508 0028
RESERVED	R	32	0x0000 002C	0x5508 002C
L1DATA_ERR_ADDR_LOC	R	32	0x0000 0030	0x5508 0030
L1TAG_ERR_INFO	RW	32	0x0000 0034	0x5508 0034
RESERVED	R	32	0x0000 0038	0x5508 0038
L1TAG_ERR_ADDR_LOC	R	32	0x0000 003C	0x5508 003C
L2RAM_ERR_INFO	RW	32	0x0000 0040	0x5508 0040
RESERVED	R	32	0x0000 0044	0x5508 0044
L2RAM_ERR_ADDR_LOC	R	32	0x0000 0048	0x5508 0048

5.4.2.2 IPU_UNICACHE_CFG Register Description

Table 5-11. CACHE_CONFIG

Address Offset	0x0000 0004		
Physical Address	0x5508 0004	**Instance**	IPU_UNICACHE_CFG_IPU
Description	Configuration Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																											LOCK_MAIN	LOCK_PORT	LOCK_INT	BYPASS	CACHE_LOCK

Bits	Field Name	Description	Type	Reset
31:5	RESERVED	Reserved	R	0x0
4	LOCK_MAIN	Lock access to maintenance registers 0x0: Locked 0x1: Not locked	RW	1
3	LOCK_PORT	Lock access to interface registers 0x0: Locked 0x1: Not locked	RW	1
2	LOCK_INT	Lock access to interrupt registers 0x0: Locked 0x1: Not locked	RW	1
1	BYPASS	Bypass cache 0x0: Everything is non-cacheable. 0x1: Everything is cacheable.	RW	0
0	CACHE_LOCK	Unicache lock. Once this bit is set only debugger or hardware reset can clear. 0x0: No effect 0x1: Only debug accesses allowed	RW	0

Table 5-12. Register Call Summary for Register CACHE_CONFIG

Dual Cortex-M4 IPU Subsystem Register Manual
- IPU_UNICACHE_CFG Register Summary: [0]

Table 5-13. CACHE_INT

Address Offset	0x0000 0008		
Physical Address	0x5508 0008	**Instance**	IPU_UNICACHE_CFG_IPU
Description	Interrupt Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								PORT			READ	WRITE	MAINT	PAGEFAULT	CONFIG

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reserved.	R	0x000000
8:5	PORT	Slave interface number that has recorded an error	RW W1toClr	0x0
4	READ	Interface read response error	RW W1toClr	0
3	WRITE	Interface write response error	RW W1toClr	0
2	MAINT	Maintenance is completed	RW W1toClr	0
1	PAGEFAULT	Unicache MMU page fault	RW W1toClr	0
0	CONFIG	Configuration error	RW W1toClr	0

Table 5-14. Register Call Summary for Register CACHE_INT

Dual Cortex-M4 IPU Subsystem Register Manual
- IPU_UNICACHE_CFG Register Summary: [0]

Table 5-15. CACHE_OCP

Address Offset	0x0000 000C		
Physical Address	0x5508 000C	**Instance**	IPU_UNICACHE_CFG_IPU
Description	Interface Configuration Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																										CLEANBUF	PREFETCH	CACHED	WRALLOCATE	WRBUFFER	WRAP

Bits	Field Name	Description	Type	Reset
31:6	RESERVED	Reserved.	R	0x0000000
5	CLEANBUF	Clean write and prefetch buffers in cache 0x0: Do not clean 0x1: Clean	RW	0
4	PREFETCH	Always prefetch data 0x0: Follow MMU policies 0x1: Always prefetch	RW	0
3	CACHED	Follow cacheable sideband signals 0x0: Reads always not allocated, writes write through if cached 0x1: Slave sideband signals determine policy	RW	1
2	WRALLOCATE	Follow write allocate sideband signals 0x0: No writes are allocated independent to sideband 0x1: Follow sideband	RW	0
1	WRBUFFER	Write throughs and write back no allocate are buffered 0x0: Write throughs and write back no allocated are not buffered 0x1: Write throughs and write back no allocated are buffered	RW	0

Bits	Field Name	Description	Type	Reset
0	WRAP	OCP wrap mode (critical word first) 0x0: Disabled 0x1: Enabled	RW	0

Table 5-16. Register Call Summary for Register CACHE_OCP

Dual Cortex-M4 IPU Subsystem Register Manual
- IPU_UNICACHE_CFG Register Summary: [0]

Table 5-17. CACHE_MAINT

Address Offset	0x0000 0010		
Physical Address	0x5508 0010	**Instance**	IPU_UNICACHE_CFG_IPU
Description	Maintenance Configuration Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																										INTERRUPT	INVALIDATE	CLEAN	UNLOCK	LOCK	PRELOAD

Bits	Field Name	Description	Type	Reset
31:6	RESERVED	Reserved.	R	0x0000000
5	INTERRUPT	Generate interrupt when maintenance operation is complete 0x0: Do not generate interrupt 0x1: Generate interrupt Note: This bit is cleared by HW when maintenance is complete.	RW	0
4	INVALIDATE	Invalidate lines in region defined by maintenance start/end addresses 0x0: Do nothing 0x1: Invalidate Note: This bit is cleared by HW when maintenance is complete.	RW	0
3	CLEAN	Evict dirty lines in region defined by maintenance start/end addresses 0x0: Do nothing 0x1: Clean Note: This bit is cleared by HW when maintenance is complete.	RW	0
2	UNLOCK	Unlock region defined by maintenance start/end addresses 0x0: Do nothing 0x1: Unlock Note: This bit is cleared by HW when maintenance is complete.	RW	0
1	LOCK	Lock region defined by maintenance start/end addresses 0x0: Do nothing 0x1: Lock Note: This bit is cleared by HW when maintenance is complete.	RW	0

Bits	Field Name	Description	Type	Reset
0	PRELOAD	Preload region defined by maintenance start/end addresses 0x0: Do nothing 0x1: Preload Note: This bit is cleared by HW when maintenance is complete.	RW	0

Table 5-18. Register Call Summary for Register CACHE_MAINT

Dual Cortex-M4 IPU Subsystem Functional Description • IPU_UNICACHE Maintenance: [0][1][2][3][4][5]
Dual Cortex-M4 IPU Subsystem Register Manual • IPU_UNICACHE_CFG Register Summary: [6]

Table 5-19. CACHE_MTSTART

Address Offset	0x0000 0014		
Physical Address	0x5508 0014	**Instance**	IPU_UNICACHE_CFG_IPU
Description	Maintenance Start Configuration Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
START_ADDR																															

Bits	Field Name	Description	Type	Reset
31:0	START_ADDR	Start address of maintenance operations, reset to 0x0000 0000 when finished	RW	0x0000 0000

Table 5-20. Register Call Summary for Register CACHE_MTSTART

Dual Cortex-M4 IPU Subsystem Functional Description • IPU_UNICACHE Maintenance: [0]
Dual Cortex-M4 IPU Subsystem Register Manual • IPU_UNICACHE_CFG Register Summary: [1]

Table 5-21. CACHE_MTEND

Address Offset	0x0000 0018		
Physical Address	0x5508 0018	**Instance**	IPU_UNICACHE_CFG_IPU
Description	Maintenance End Configuration Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
END_ADDR																															

Bits	Field Name	Description	Type	Reset
31:0	END_ADDR	End address of maintenance operations, reset to 0x0000 0000 when finished	RW	0x0000 0000

Table 5-22. Register Call Summary for Register CACHE_MTEND

Dual Cortex-M4 IPU Subsystem Functional Description • IPU_UNICACHE Maintenance: [0]
Dual Cortex-M4 IPU Subsystem Register Manual • IPU_UNICACHE_CFG Register Summary: [1]

Table 5-23. CACHE_CTADDR

Address Offset	0x0000 001C		
Physical Address	0x5508 001C	**Instance**	IPU_UNICACHE_CFG_IPU
Description	Cache Test Address Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDRESS																															

Bits	Field Name	Description	Type	Reset
31:0	ADDRESS	Address of cache visibility when read CACHE_CTDATA register, autoincrements	RW	0x0000 0000

Table 5-24. Register Call Summary for Register CACHE_CTADDR

Dual Cortex-M4 IPU Subsystem Register Manual • IPU_UNICACHE_CFG Register Summary: [0] • IPU_UNICACHE_CFG Register Description: [1][2]

Table 5-25. CACHE_CTDATA

Address Offset	0x0000 0020		
Physical Address	0x5508 0020	**Instance**	IPU_UNICACHE_CFG_IPU
Description	Cache Test Data Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA																															

Bits	Field Name	Description	Type	Reset
31:0	DATA	Cache data at address of CACHE_CTADDR register, CACHE_CTADDR autoincrements each time CACHE_CTDATA is read	RW	0x0000 0000

Table 5-26. Register Call Summary for Register CACHE_CTDATA

Dual Cortex-M4 IPU Subsystem Register Manual • IPU_UNICACHE_CFG Register Summary: [0] • IPU_UNICACHE_CFG Register Description: [1][2]

Table 5-27. ECC_CFG

Address Offset	0x0000 0024		
Physical Address	0x5508 0024	**Instance**	IPU_UNICACHE_CFG_IPU
Description	ECC configuration register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																L2RAM_SEC_AUTO_EN	L2RAM_SRESP_EN	L2RAM_DATA_MASK	L2RAM_CODE_MASK	L2RAM_ECC_EN	L1TAG_SEC_AUTO_EN	RESERVED	L1TAG_DATA_MASK	L1TAG_CODE_MASK	L1TAG_ECC_EN	L1DATA_SEC_AUTO_EN	L1DATA_SRESP_EN	L1DATA_DATA_MASK	L1DATA_CODE_MASK	L1DATA_ECC_EN

Bits	Field Name	Description	Type	Reset
31:15	RESERVED	Reserved. Read returns 0s.	R	0x00 0000
14	L2RAM_SEC_AUTO_EN	Enables auto-correction of data in case of SEC error in L2 RAM	RW	0
13	L2RAM_SRESP_EN	Enables error response to master in case of DED error in L2 RAM	RW	0
12	L2RAM_DATA_MASK	L2 RAM ECC Code Mask register. Enabling this will mask any write to data block while ECC will be updated on any writes to the memory. This is for test purpose.	RW	0
11	L2RAM_CODE_MASK	L2 RAM ECC Code Mask register. Enabling this will mask any write to ECC code block of memory. This is for test purpose.	RW	0
10	L2RAM_ECC_EN	L2 RAM ECC enable	RW	0
9	L1TAG_SEC_AUTO_EN	Enables auto-correction of data in case of SEC error in L1 Tag	RW	0
8	RESERVED	Reserved.	RW	0
7	L1TAG_DATA_MASK	L1 Tag ECC Code Mask register. Enabling this will mask any write to data block while ECC will be updated on any writes to the memory. This is for test purpose.	RW	0
6	L1TAG_CODE_MASK	L1 Tag ECC Code Mask register. Enabling this will mask any write to ECC code block of memory. This is for test purpose.	RW	0
5	L1TAG_ECC_EN	L1 Tag ECC enable	RW	0
4	L1DATA_SEC_AUTO_EN	Enables auto-correction of data in case of SEC error in L1 Data	RW	0
3	L1DATA_SRESP_EN	Enables error response to master in case of DED error in L1 Data	RW	0
2	L1DATA_DATA_MASK	L1 Data ECC Code Mask register. Enabling this will mask any write to data block while ECC will be updated on any writes to the memory. This is for test purpose.	RW	0
1	L1DATA_CODE_MASK	L1 Data ECC Code Mask register. Enabling this will mask any write to ECC code block of memory. This is for test purpose.	RW	0
0	L1DATA_ECC_EN	L1 Data ECC enable	RW	0

Table 5-28. Register Call Summary for Register ECC_CFG

Dual Cortex-M4 IPU Subsystem Register Manual

- IPU_UNICACHE_CFG Register Summary: [0]

Table 5-29. L1DATA_ERR_INFO

Address Offset	0x0000 0028		
Physical Address	0x5508 0028	**Instance**	IPU_UNICACHE_CFG_IPU
Description	L1 Data ECC information register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												ACCESS_TYPE	DED	CODE_ERR	SEC

Bits	Field Name	Description	Type	Reset
31:4	RESERVED	Reserved. Read returns 0s.	R	0x000 0000
3	ACCESS_TYPE	Indicates what access type resulted in ECC error in L1 Data 0 - indicates access other than eviction 1 - eviction	RW W1toClr	0
2	DED	Indicates DED error in L1 Data	RW W1toClr	0
1	CODE_ERR	Indicates SEC error in ECC code area of memory in L1 Data	RW W1toClr	0
0	SEC	Indicates SEC error in L1 Data	RW W1toClr	0

Table 5-30. Register Call Summary for Register L1DATA_ERR_INFO

Dual Cortex-M4 IPU Subsystem Register Manual

- IPU_UNICACHE_CFG Register Summary: [0]

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														DATA_LOCATION	

Bits	Field Name	Description	Type	Reset
31:6	RESERVED	Reserved. Read returns 0s.	R	0x000 0000
5:0	DATA_LOCATION	Indicates bit of data in case of SEC error in L1 Data	R	0x00

Table 5-31. L1DATA_ERR_ADDR_LOC

Address Offset	0x0000 0030		
Physical Address	0x5508 0030	**Instance**	IPU_UNICACHE_CFG_IPU
Description	Indicates address location of error occurence in L1 Data		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR_LOCATION																															

Bits	Field Name	Description	Type	Reset
31:0	ADDR_LOCATION	Indicates virtual address where ECC error has occured in L1 Data	R	0x0000 0000

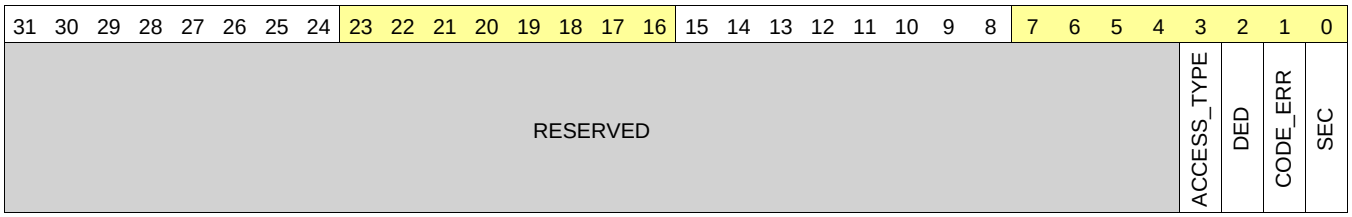
Table 5-32. Register Call Summary for Register L1DATA_ERR_ADDR_LOC

Dual Cortex-M4 IPU Subsystem Register Manual

- IPU_UNICACHE_CFG Register Summary: [0]

Table 5-33. L1TAG_ERR_INFO

Address Offset	0x0000 0034		
Physical Address	0x5508 0034	**Instance**	IPU_UNICACHE_CFG_IPU
Description	L1 Tag ECC information register		
Type	RW		

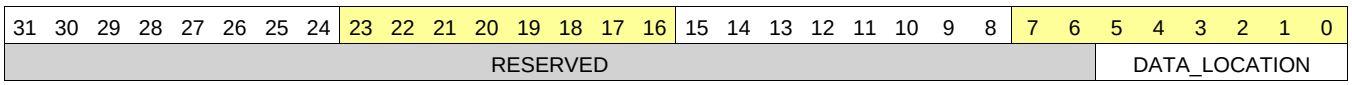


Bits	Field Name	Description	Type	Reset
31:4	RESERVED	Reserved. Read returns 0s.	R	0x000 0000
3	ACCESS_TYPE	Indicates what access type resulted in ECC error in L1 Tag 0 - indicates access other than eviction 1 - eviction	RW W1toClr	0
2	DED	Indicates DED error in L1 Tag	RW W1toClr	0
1	CODE_ERR	Indicates SEC error in ECC code area of memory in L1 Tag	RW W1toClr	0
0	SEC	Indicates SEC error in L1 Tag	RW W1toClr	0

Table 5-34. Register Call Summary for Register L1TAG_ERR_INFO

Dual Cortex-M4 IPU Subsystem Register Manual

- IPU_UNICACHE_CFG Register Summary: [0]



Bits	Field Name	Description	Type	Reset
31:6	RESERVED	Reserved. Read returns 0s.	R	0x000 0000
5:0	DATA_LOCATION	Indicates bit of data in case of SEC error in L1 Tag	R	0x00

Table 5-35. L1TAG_ERR_ADDR_LOC

Address Offset	0x0000 003C		
Physical Address	0x5508 003C	**Instance**	IPU_UNICACHE_CFG_IPU
Description	Indicates address location of error occurence in L1 Data		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR_LOCATION																															

Bits	Field Name	Description	Type	Reset
31:0	ADDR_LOCATION	Indicates virtual address for the access which resulted in an ECC error in L1 Tag memory.	R	0x0000 0000

Table 5-36. Register Call Summary for Register L1TAG_ERR_ADDR_LOC

Dual Cortex-M4 IPU Subsystem Register Manual

- IPU_UNICACHE_CFG Register Summary: [0]

Table 5-37. L2RAM_ERR_INFO

Address Offset	0x0000 0040		
Physical Address	0x5508 0040	**Instance**	IPU_UNICACHE_CFG_IPU
Description	L2 RAM ECC information register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																										ACCESS_TYPE_MPORT	ACCESS_TYPE_CACHE1	ACCESS_TYPE_CACHE2	DED	CODE_ERR	SEC

Bits	Field Name	Description	Type	Reset
31:6	RESERVED	Reserved. Read returns 0s.	R	0x000 0000
5	ACCESS_TYPE_MPORT	This bit is set whenever an access is made to L2RAM via the slave configuration interface.	RW W1toClr	0
4	ACCESS_TYPE_CACHE1	This bit is set whenever an access is made to L2RAM via the Unicache interface. This will be set even if Cache is disabled.	RW W1toClr	0
3	ACCESS_TYPE_CACHE2	This bit is reserved since L2 cache is not implemented in IPU subsystem.	RW W1toClr	0
2	DED	Indicates DED error in L2 RAM	RW W1toClr	0
1	CODE_ERR	Indicates SEC error in ECC code area of memory in L2 RAM	RW W1toClr	0
0	SEC	Indicates SEC error in L2 RAM	RW W1toClr	0

Table 5-38. Register Call Summary for Register L2RAM_ERR_INFO

Dual Cortex-M4 IPU Subsystem Register Manual
- IPU_UNICACHE_CFG Register Summary: [0]

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								DATA_LOCATION							

Bits	Field Name	Description	Type	Reset
31:6	RESERVED	Reserved. Read returns 0s.	R	0x000 0000
5:0	DATA_LOCATION	Indicates bit of data in case of SEC error in L2RAM	R	0x00

Table 5-39. L2RAM_ERR_ADDR_LOC

Address Offset	0x0000 0048		
Physical Address	0x5508 0048	**Instance**	IPU_UNICACHE_CFG_IPU
Description	Indicates address location of error occurence in L2 RAM		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR_LOCATION																															

Bits	Field Name	Description	Type	Reset
31:0	ADDR_LOCATION	Indicates physical address where ECC error has occured in L2RAM. The address reported is aligned to a 8 byte boundary	R	0x0000 0000

Table 5-40. Register Call Summary for Register L2RAM_ERR_ADDR_LOC

Dual Cortex-M4 IPU Subsystem Register Manual
- IPU_UNICACHE_CFG Register Summary: [0]

5.4.3 IPU_UNICACHE_SCTM Registers

5.4.3.1 IPU_UNICACHE_SCTM Register Summary

Table 5-41. IPU_UNICACHE_SCTM Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	Physical Address (IPU Private Access)
CACHE_SCTM_CTCNTL	RW	32	0x0000 0000	0x5508 0400
RESERVED	R	32	0x0000 0020	0x5508 0420
RESERVED	R	32	0x0000 0024	0x5508 0424
RESERVED	R	32	0x0000 0028	0x5508 0428
RESERVED	R	32	0x0000 002C	0x5508 042C
CACHE_SCTM_TINTVLR_i (1)	RW	32	0x0000 0040 + (0x4 * i)	0x5508 0440 + (0x4 * i)
CACHE_SCTM_CTDBGNUM	R	32	0x0000 007C	0x5508 047C
CACHE_SCTM_CTGNBL	RW	32	0x0000 00F0	0x5508 04F0
CACHE_SCTM_CTGRST	RW	32	0x0000 00F8	0x5508 04F8
CACHE_SCTM_CTCR_WT_i (1)	RW	32	0x0000 0100 + (0x4 * i)	0x5508 0500 + (0x4 * i)

Table 5-41. IPU_UNICACHE_SCTM Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	Physical Address (IPU Private Access)
CACHE_SCTM_CTCR_WOT_j [3]	RW	32	0x0000 0108 + (0x4 * j)	0x5508 0508 + (0x4 * j)
CACHE_SCTM_CTCNTR_k [2]	R	32	0x0000 0180 + (0x4 * k)	0x5508 0580 + (0x4 * k)

(1) i = 0 to 1
(2) k = 0 to 7
(3) j = 0 to 5

(1) i = 0 to 1
(2) k = 0 to 7
(3) j = 0 to 5

5.4.3.2 IPU_UNICACHE_SCTM Register Description

Table 5-42. CACHE_SCTM_CTCNTL

Address Offset	0x0000 0000		
Physical Address	0x5508 0400	**Instance**	IPU_UNICACHE_SCTM_IPU
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NUMSTM								NUMINPT								NUMTIMR				NUMCNTR				REVISION					IDLEMODE		ENBL

Bits	Field Name	Description	Type	Reset
31:26	NUMSTM	Number of timers that can export via STM	R	0x00
25:18	NUMINPT	Number of event input signals	R	0x1F
17:13	NUMTIMR	Number of timers in the module	R	0x02
12:7	NUMCNTR	Number of counters in the module	R	0x08
6:3	REVISION	Revision ID of SCTM	R	0x- TI internal data
2:1	IDLEMODE	Idle mode control 0x0: Force Idle mode 0x1: Ths SCTM will acknoledge the idle request, but never transition to the idle state 0x2: Ths SCTM uses the smart idle protocol. This is the default mode 0x3: Since the SCTM does not support internal wakeup, this mode is identical to smart_idle	RW	0x2
0	ENBL	SCTM global enable 0x0: This module is disabled. Only the configuration interface is functional. All other logic is reset 0x1: The module is enabled and individual counter/timers can be configured	RW	0

Table 5-43. Register Call Summary for Register CACHE_SCTM_CTCNTL

Dual Cortex-M4 IPU Subsystem Register Manual

- IPU_UNICACHE_SCTM Register Summary: [0]

Table 5-44. CACHE_SCTM_TINTVLR_i

Address Offset	0x0000 0040 + (0x4 * i)		
Physical Address	0x5508 0440 + (0x4 * i)	**Instance**	IPU_UNICACHE_SCTM_IPU
Description	These registers contain the interval match value for the corresponding timers in the SCTM		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTERVAL																															

Bits	Field Name	Description	Type	Reset
31:0	INTERVAL	Interval match value for the timers in the SCTM	RW	0x0000 0000

Table 5-45. Register Call Summary for Register CACHE_SCTM_TINTVLR_i

Dual Cortex-M4 IPU Subsystem Functional Description

- Timer Functions: [0][1][2]

Dual Cortex-M4 IPU Subsystem Register Manual

- IPU_UNICACHE_SCTM Register Summary: [3]

Table 5-46. CACHE_SCTM_CTDBGNUM

Address Offset	0x0000 007C		
Physical Address	0x5508 047C	**Instance**	IPU_UNICACHE_SCTM_IPU
Description	Counter Timer Number Debug Event Register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														NUMEVT	

Bits	Field Name	Description	Type	Reset
31:3	RESERVED	Reserved.	R	0x0000 0000
2:0	NUMEVT	Number of input selectors for debug events	R	0x0

Table 5-47. Register Call Summary for Register CACHE_SCTM_CTDBGNUM

Dual Cortex-M4 IPU Subsystem Register Manual

- IPU_UNICACHE_SCTM Register Summary: [0]

Table 5-48. CACHE_SCTM_CTGNBL

Address Offset	0x0000 00F0		
Physical Address	0x5508 04F0	**Instance**	IPU_UNICACHE_SCTM_IPU
Description	These registers provide for simultaneous enable/disable of 32 counters		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								ENABLE							

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Reserved.	R	0x000000
7:0	ENABLE	The counter enable bit field	RW	0x00

Table 5-49. Register Call Summary for Register CACHE_SCTM_CTGNBL

Dual Cortex-M4 IPU Subsystem Functional Description
- Counter Functions: [0]

Dual Cortex-M4 IPU Subsystem Register Manual
- IPU_UNICACHE_SCTM Register Summary: [1]

Table 5-50. CACHE_SCTM_CTGRST

Address Offset	0x0000 00F8		
Physical Address	0x5508 04F8	**Instance**	IPU_UNICACHE_SCTM_IPU
Description	These registers provide for simultaneous reset of 32 counters		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								RESET							

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Reserved.	R	0x000000
7:0	RESET	The counter reset bit field	RW	0x00

Table 5-51. Register Call Summary for Register CACHE_SCTM_CTGRST

Dual Cortex-M4 IPU Subsystem Functional Description
- Counter Functions: [0]

Dual Cortex-M4 IPU Subsystem Register Manual
- IPU_UNICACHE_SCTM Register Summary: [1]

Table 5-52. CACHE_SCTM_CTCR_WT_i

Address Offset	0x0000 0100 + (0x4 * i)		
Physical Address	0x5508 0500 + (0x4 * i)	**Instance**	IPU_UNICACHE_SCTM_IPU
Description	These registers contain the control and status settings for a single counter in the module. There will be a CTCR for every counter in the module (WT: with timer)		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								INPSEL								RESERVED					RESTART	DBG	INT	CHNSDW	OVRFLW	IDLE	FREE	DURMODE	CHAIN	RESET	ENBL

Bits	Field Name	Description	Type	Reset
31:21	RESERVED	Reserved.	R	0x000
20:16	INPSEL	Counter Timer input selection 0: Constantly asserted input that results in a free-running counter/timer 1-31: Index of event input signal selected	RW	0x00
15:11	RESERVED	Reserved.	R	0x00
10	RESTART	Restart the timer after an interval match 0: The timer stops after the first interval match. It must be manually reset by software before it starts counting again (run-once timer mode). 1: The timer immediately resets to 0 and begins incrementing again based on the current input configuration (restart timer mode).	RW	0
9	DBG	Signal debug logic on interval match 0: No debug event is generated. 1: Upon interval match, generates a debug event on the corresponding debug output event signal	RW	0
8	INT	Generate interrupt on interval match 0: No interrupt is generated. 1: Upon interval match, generates an interrupt on the corresponding interrupt output event signal	RW	0
7	CHNSDW	Counter has a shadow register for chain reads. 0: The read of the corresponding counter register returns the current value. 1: Read of the high-order counter register, simultaneously loads the current value of the 32 LSBs into a shadow register. The read of the counter register that corresponds to this counter returns the value of the shadow register. This is applicable only when the counter is chained.	R	0
6	OVRFLW	Counter has wrapped since it was last read 0: The counter has not wrapped since the last read. 1: The counter has wrapped since the last read.	R	0
5	IDLE	Counter ignores processor IDLE state 0: The counter does not increment during IDLE state. 1: The counter continues to function during IDLE state; applicable if IDLEMODE = 1.	RW	0
4	FREE	Counter ignores processor debug halt state 0: The counter does not increment (decrement) during the debug halt state.	RW	0

Bits	Field Name	Description	Type	Reset
		1: The counter continues to function during debug halt state.		
3	DURMODE	Counter is in duration or occurrence mode 0: The counter operates in event mode. The counter increments by 1 each time a rising edge is seen on the designated input event signal. 1: The counter operates in duration mode. The counter increments every time a clock cycle is seen and the corresponding input event signal is asserted.	RW	0
2	CHAIN	Counter is chained to an adjacent counter 0: The counter is not chained. 1: Reserved	RW	0
1	RESET	Counter reset control 0: No effect 1: The corresponding counter is reset to the initial value and the OVERFLW bit is cleared. It continues to function if it is still enabled.	RW	0
0	ENBL	Counter enable control 0: The counter does not increment. 1: The counter increments as configured.	RW	0

Table 5-53. Register Call Summary for Register CACHE_SCTM_CTCR_WT_i

Dual Cortex-M4 IPU Subsystem Functional Description
- Counter Functions: [0][1][2][3][4][5][6][7][8][9]
- Timer Functions: [10][11][12][13][14]

Dual Cortex-M4 IPU Subsystem Register Manual
- IPU_UNICACHE_SCTM Register Summary: [15]

Table 5-54. CACHE_SCTM_CTCR_WOT_j

Address Offset	0x0000 0108 + (0x4 * j)		
Physical Address	0x5508 0508 + (0x4 * j)	**Instance**	IPU_UNICACHE_SCTM_IPU
Description	These registers contain the control and status settings for a single counter in the module. There will be a CTCR for every counter in the module (WOT: without timer)		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								INPSEL								RESERVED								CHNSDW	OVRFLW	IDLE	FREE	DURMODE	CHAIN	RESET	ENBL

Bits	Field Name	Description	Type	Reset
31:21	RESERVED	Reserved.	R	0x000
20:16	INPSEL	Counter input selection 0: Constant low signal on the output interface 1–31: Index of event input signal selected	RW	0x000
15:8	RESERVED	Reserved.	R	0x00
7	CHNSDW	Counter has a shadow register for chain reads. 0: The read of the corresponding counter register returns the current value.	R	1

Bits	Field Name	Description	Type	Reset
		1: Read of the high-order counter register, simultaneously loads the current value of the 32 LSBs into a shadow register. The read of the counter register that corresponds to this counter returns the value of the shadow register. This is applicable only when the counter is chained.		
6	OVRFLW	Counter has wrapped since it was last read 0: The counter has not wrapped since the last read. 1: The counter has wrapped since the last read.	R	0
5	IDLE	Counter ignores processor IDLE state 0: The counter does not increment during IDLE state. 1: The counter continues to function during IDLE state; applicable if IDLEMODE = 1.	RW	0
4	FREE	Counter ignores processor debug halt state 0: The counter does not increment (decrement) during the debug halt state. 1: The counter continues to function during debug halt state.	RW	0
3	DURMODE	Counter is in duration or occurrence mode 0: The counter operates in event mode. The counter increments by 1 each time a rising edge is seen on the designated input event signal. 1: The counter operates in duration mode. The counter increments every time a clock cycle is seen and the corresponding input event signal is asserted.	RW	0
2	CHAIN	Counter is chained to an adjacent counter 0: The counter is not chained. 1: The counter is chained to its partner.	RW	0
1	RESET	Counter reset control 0: No effect 1: The corresponding counter is reset to the initial value and the OVERFLW bit is cleared. It continues to function if it is still enabled.	RW	0
0	ENBL	Counter enable control 0: The counter does not increment. 1: The counter increments as configured.	RW	0

Table 5-55. Register Call Summary for Register CACHE_SCTM_CTCR_WOT_j

Dual Cortex-M4 IPU Subsystem Functional Description • Counter Functions: [0][1][2][3][4][5][6][7][8][9]
Dual Cortex-M4 IPU Subsystem Register Manual • IPU_UNICACHE_SCTM Register Summary: [10]

Table 5-56. CACHE_SCTM_CTCNTR_k

Address Offset	0x0000 0180 + (0x4 * k)		
Physical Address	0x5508 0580 + (0x4 * k)	**Instance**	IPU_UNICACHE_SCTM_IPU
Description	These registers contain the value of an individual counter in the module. There will be a CTCNTR for every counter in the module		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
COUNT																															

Bits	Field Name	Description	Type	Reset
31:0	COUNT	Counter value	R	0x0000 0000

Table 5-57. Register Call Summary for Register CACHE_SCTM_CTCNTR_k

Dual Cortex-M4 IPU Subsystem Functional Description • Timer Functions: [0]
Dual Cortex-M4 IPU Subsystem Register Manual • IPU_UNICACHE_SCTM Register Summary: [1]

5.4.4 IPU_UNICACHE_MMU (AMMU) Registers

5.4.4.1 IPU_UNICACHE_MMU (AMMU) Register Summary

Table 5-58. IPU_UNICACHE_MMU (AMMU) Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	Physical Address (IPU Private Access)	Physical Address (L3_MAIN Access)
CACHE_MMU_LARGE_ADDR_i [1]	RW	32	0x0000 0000 + (0x4 * i)	0x5508 0800 + (0x4 * i)	0x5888 0800 + (0x4 * i)
CACHE_MMU_LARGE_XLTE_i [1]	RW	32	0x0000 0020 + (0x4 * i)	0x5508 0820 + (0x4 * i)	0x5888 0820 + (0x4 * i)
CACHE_MMU_LARGE_POLICY_i [1]	RW	32	0x0000 0040 + (0x4 * i)	0x5508 0840 + (0x4 * i)	0x5888 0840 + (0x4 * i)
CACHE_MMU_MED_ADDR_j [2]	RW	32	0x0000 0060 + (0x4 * j)	0x5508 0860 + (0x4 * j)	0x5888 0860 + (0x4 * j)
CACHE_MMU_MED_XLTE_j [2]	RW	32	0x0000 00A0 + (0x4 * j)	0x5508 08A0 + (0x4 * j)	0x5888 08A0 + (0x4 * j)
CACHE_MMU_MED_POLICY_j [2]	RW	32	0x0000 00E0 + (0x4 *j)	0x5508 08E0 + (0x4 * j)	0x5888 08E0 + (0x4 * j)
CACHE_MMU_SMALL_ADDR_k [3]	RW	32	0x0000 0120 + (0x4 * k)	0x5508 0920 + (0x4 * k)	0x5888 0920 + (0x4 * k)
CACHE_MMU_SMALL_XLTE_k [3]	RW	32	0x0000 01A0 + (0x4 * k)	0x5508 09A0 + (0x4 * k)	0x5888 09A0 + (0x4 * k)
CACHE_MMU_SMALL_POLICY_k [3]	RW	32	0x0000 0220 + (0x4 * k)	0x5508 0A20 + (0x4 * k)	0x5888 0A20 + (0x4 * k)
CACHE_MMU_SMALL_MAINT_k [3]	RW	32	0x0000 02A0 + (0x4 * k)	0x5508 0AA0 + (0x4 * k)	0x5888 0AA0 + (0x4 * k)
Reserved	RW	32	0x0000 04A8	0x5508 0CA8	0x5888 0CA8
Reserved	RW	32	0x0000 04AC	0x5508 0CAC	0x5888 0CAC
Reserved	RW	32	0x0000 04B0	0x5508 0CB0	0x5888 0CB0

(1) i = 0 to 3
(2) j = 0 to 1
(3) k = 0 to 9

Table 5-58. IPU_UNICACHE_MMU (AMMU) Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	Physical Address (IPU Private Access)	Physical Address (L3_MAIN Access)
Reserved	R	32	0x0000 04B4	0x5508 0CB4	0x5888 0CB4
CACHE_MMU_MMUCONFIG	RW	32	0x0000 04B8	0x5508 0CB8	0x5888 0CB8

5.4.4.2 IPU_UNICACHE_MMU (AMMU) Register Description

Table 5-59. CACHE_MMU_LARGE_ADDR_i

Address Offset	0x0000 0000 + (0x4 * i)		
Physical Address	0x5508 0800 + (0x4 * i)	**Instance**	IPU_UNICACHE_MMU_IPU
Description	Large page address		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDRESS								RESERVED																							

Bits	Field Name	Description	Type	Reset
31:25	ADDRESS	Logical source address	RW	0x00
24:0	RESERVED	Reserved.	R	0x0000000

Table 5-60. Register Call Summary for Register CACHE_MMU_LARGE_ADDR_i

Dual Cortex-M4 IPU Subsystem Functional Description
- Page Attributes: [0]

Dual Cortex-M4 IPU Subsystem Register Manual
- IPU_UNICACHE_MMU Register Summary: [1]

Table 5-61. CACHE_MMU_LARGE_XLTE_i

Address Offset	0x0000 0020 + (0x4 * i)		
Physical Address	0x5508 0820 + (0x4 * i)	**Instance**	IPU_UNICACHE_MMU_IPU
Description	Large page translated address		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDRESS								RESERVED																							IGNORE

Bits	Field Name	Description	Type	Reset
31:25	ADDRESS	Logical source translated address	RW	0x00
24:1	RESERVED	Reserved	R	0x000000
0	IGNORE	Do not use translated address.	RW	0

Table 5-62. Register Call Summary for Register CACHE_MMU_LARGE_XLTE_i

Dual Cortex-M4 IPU Subsystem Functional Description • Page Attributes: [0]
Dual Cortex-M4 IPU Subsystem Register Manual • IPU_UNICACHE_MMU Register Summary: [1]

Table 5-63. CACHE_MMU_LARGE_POLICY_i

Address Offset	0x0000 0040 + (0x4 * i)		
Physical Address	0x5508 0840 + (0x4 * i)	**Instance**	IPU_UNICACHE_MMU_IPU
Description	Large page policy		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												L1_WR_POLICY	L1_ALLOCATE	L1_POSTED	L1_CACHEABLE	RESERVED								PRELOAD	READ	EXECUTE	RESERVED			SIZE	ENABLE

Bits	Field Name	Description	Type	Reset
31:20	RESERVED	Reserved.	R	0x000
19	L1_WR_POLICY	L1 write policy 0x0: Write through 0x1: Write back	RW	0
18	L1_ALLOCATE	L1 allocate policy 0x0: No writes are allocated 0x1: Follow sideband	RW	0
17	L1_POSTED	L1 posted policy 0x0: Not posted 0x1: Posted	RW	0
16	L1_CACHEABLE	L1 cache policy 0x0: Non-cacheable 0x1: Cacheable	RW	0
15:7	RESERVED	Reserved	R	0x000
6	PRELOAD	Preload region 0x0: Do not preload 0x1: Preload	RW	0
5	READ	Read only	RW	0
4	EXECUTE	Execute only	RW	0
3:2	RESERVED	Reserved	R	0x0
1	SIZE	Size of page 0x0: 32 MiB 0x1: 512 MiB	RW	0
0	ENABLE	Enable page 0x0: Page not enabled 0x1: Page enabled	RW	0

Table 5-64. Register Call Summary for Register CACHE_MMU_LARGE_POLICY_i

Dual Cortex-M4 IPU Subsystem Functional Description
- Page Attributes: [0]
- Policy Support: [1]

Dual Cortex-M4 IPU Subsystem Register Manual
- IPU_UNICACHE_MMU Register Summary: [2]

Table 5-65. CACHE_MMU_MED_ADDR_j

Address Offset	0x0000 0060 + (0x4 * j)		
Physical Address	0x5508 0860 + (0x4 * j)	**Instance**	IPU_UNICACHE_MMU_IPU
Description	Medium page address		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDRESS																RESERVED															

Bits	Field Name	Description	Type	Reset
31:17	ADDRESS	Logical source address	RW	0x0000
16:0	RESERVED	Reserved	R	0x00000

Table 5-66. Register Call Summary for Register CACHE_MMU_MED_ADDR_j

Dual Cortex-M4 IPU Subsystem Functional Description
- Page Attributes: [0]

Dual Cortex-M4 IPU Subsystem Register Manual
- IPU_UNICACHE_MMU Register Summary: [1]

Table 5-67. CACHE_MMU_MED_XLTE_j

Address Offset	0x0000 00A0 + (0x4 * j)		
Physical Address	0x5508 08A0 + (0x4 * j)	**Instance**	IPU_UNICACHE_MMU_IPU
Description	Medium page translated address		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDRESS																RESERVED															IGNORE

Bits	Field Name	Description	Type	Reset
31:17	ADDRESS	Logical source translated address	RW	0x0000
16:1	RESERVED	Reserved.	R	0x0000
0	IGNORE	Do not use translated address.	RW	0

Table 5-68. Register Call Summary for Register CACHE_MMU_MED_XLTE_j

Dual Cortex-M4 IPU Subsystem Functional Description
- Page Attributes: [0]

Dual Cortex-M4 IPU Subsystem Register Manual
- IPU_UNICACHE_MMU Register Summary: [1]

Table 5-69. CACHE_MMU_MED_POLICY_j

Address Offset	0x0000 00E0 + (0x4 * j)		
Physical Address	0x5508 08E0 + (0x4 * j)	**Instance**	IPU_UNICACHE_MMU_IPU
Description	Medium page policy		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												L1_WR_POLICY	L1_ALLOCATE	L1_POSTED	L1_CACHEABLE	RESERVED								PRELOAD	READ	EXECUTE	RESERVED			SIZE	ENABLE

Bits	Field Name	Description	Type	Reset
31:20	RESERVED	Reserved	R	0x000
19	L1_WR_POLICY	L1 write policy 0x0: Write through 0x1: Write back	RW	0
18	L1_ALLOCATE	L1 allocate policy 0x0: No writes are allocated 0x1: Follow sideband	RW	0
17	L1_POSTED	L1 posted policy 0x0: Non-posted 0x1: Posted	RW	0
16	L1_CACHEABLE	L1 cache policy 0x0: Non-cacheable 0x1: Cacheable	RW	0
15:7	RESERVED	Reserved	R	0x000
6	PRELOAD	Preload region 0x0: Do not preload 0x1: Preload	RW	0
5	READ	Read only	RW	0
4	EXECUTE	Execute only	RW	0
3:2	RESERVED	Reserved	R	0x0
1	SIZE	Size of page 0x0: 128 KiB 0x1: 256 KiB	RW	0
0	ENABLE	Enable page 0x0: Page not enabled 0x1: Page enabled	RW	0

Table 5-70. Register Call Summary for Register CACHE_MMU_MED_POLICY_j

Dual Cortex-M4 IPU Subsystem Functional Description
- Page Attributes: [0]
- Policy Support: [1]

Dual Cortex-M4 IPU Subsystem Register Manual
- IPU_UNICACHE_MMU Register Summary: [2]

Table 5-71. CACHE_MMU_SMALL_ADDR_k

Address Offset	0x0000 0120 + (0x4 * k)		
Physical Address	0x5508 0920 + (0x4 * k)	**Instance**	IPU_UNICACHE_MMU_IPU
Description	Small page address		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDRESS																				RESERVED											

Bits	Field Name	Description	Type	Reset
31:12	ADDRESS	Logical source address	RW	See Table 5-73.
11:0	RESERVED	Reserved.	R	0x000

Table 5-72. Register Call Summary for Register CACHE_MMU_SMALL_ADDR_k

Dual Cortex-M4 IPU Subsystem Functional Description
- Page Attributes: [0]

Dual Cortex-M4 IPU Subsystem Register Manual
- IPU_UNICACHE_MMU Register Summary: [1]

Table 5-73. Reset Value for CACHE_MMU_SMALL_ADDR_k[31:12] ADDRESS

Instance	Reset Value
CACHE_MMU_SMALL_ADDR_0	Takes the value of CTRL_CORE_CORTEX_M4_MMUADDRLOGICTR[19:0] shifted 12-bit left
CACHE_MMU_SMALL_ADDR_1	0x40000
CACHE_MMU_SMALL_ADDR_[2..9]	0x00000

Table 5-74. CACHE_MMU_SMALL_XLTE_k

Address Offset	0x0000 01A0 + (0x4 * k)		
Physical Address	0x5508 09A0 + (0x4 * k)	**Instance**	IPU_UNICACHE_MMU_IPU
Description	Small page translated address		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDRESS																				RESERVED											IGNORE

Bits	Field Name	Description	Type	Reset
31:12	ADDRESS	Physical translated address	RW	See Table 5-76.
11:1	RESERVED	Reserved	R	0x000
0	IGNORE	Do not use translated address.	RW	0

Table 5-75. Register Call Summary for Register CACHE_MMU_SMALL_XLTE_k

Dual Cortex-M4 IPU Subsystem Functional Description
- Page Attributes: [0]

Dual Cortex-M4 IPU Subsystem Register Manual
- IPU_UNICACHE_MMU Register Summary: [1]

Table 5-76. Reset Value for CACHE_MMU_SMALL_XLTE_k[31:12] ADDRESS

Instance	Reset Value
CACHE_MMU_SMALL_XLTE_0	Takes the value of CTRL_CORE_CORTEX_M4_MMUADDRTRANSLTR[19:0] shifted 12-bit left
CACHE_MMU_SMALL_XLTE_1	0x55080
CACHE_MMU_SMALL_XLTE_[2..9]	0x00000

Table 5-77. CACHE_MMU_SMALL_POLICY_k

Address Offset	0x0000 0220 + (0x4 * k)		
Physical Address	0x5508 0A20 + (0x4 * k)	**Instance**	IPU_UNICACHE_MMU_IPU
Description	Small page policy		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												L1_WR_POLICY	L1_ALLOCATE	L1_POSTED	L1_CACHEABLE	RESERVED							COHERENCY	RESERVED	PRELOAD	READ	EXECUTE	RESERVED		SIZE	ENABLE

Bits	Field Name	Description	Type	Reset
31:20	RESERVED	Reserved.	R	0x000
19	L1_WR_POLICY	L1 write policy 0x0: Write through 0x1: Write back	RW	0
18	L1_ALLOCATE	L1 allocate policy 0x0: No writes are allocated 0x1: Follow sideband	RW	0
17	L1_POSTED	L1 posted policy 0x0: Non-posted 0x1: Posted	RW	0
16	L1_CACHEABLE	L1 cache policy 0x0: Non-cacheable 0x1: Cacheable	RW	0
15:9	RESERVED	Reserved	R	0x00
8	COHERENCY	Coherency	R	0
7	RESERVED	Reserved	R	0
6	PRELOAD	Preload region 0x0: Do not preload 0x1: Preload	RW	0
5	READ	Read only	RW	0
4	EXECUTE	Execute only	RW	0
3:2	RESERVED	Reserved	R	0x0
1	SIZE	Size of page 0x0: 4 KiB 0x1: 16 KiB	RW	0

Bits	Field Name	Description	Type	Reset
0	ENABLE	Enable page 0x0: Page not enabled 0x1: Page enabled	RW	0

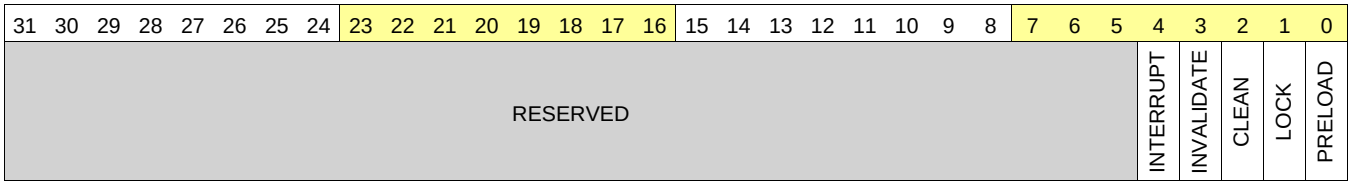
Table 5-78. Register Call Summary for Register CACHE_MMU_SMALL_POLICY_k

Dual Cortex-M4 IPU Subsystem Functional Description
- Page Attributes: [0]
- Policy Support: [1]

Dual Cortex-M4 IPU Subsystem Register Manual
- IPU_UNICACHE_MMU Register Summary: [2]

Table 5-79. CACHE_MMU_SMALL_MAINT_k

Address Offset	0x0000 02A0 + (0x4 * k)		
Physical Address	0x5508 0AA0 + (0x4 * k)	**Instance**	IPU_UNICACHE_MMU_IPU
Description	Small page maintenance configuration		
Type	RW		



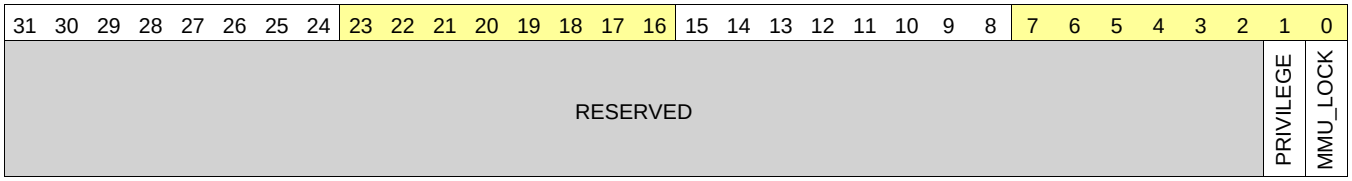
Bits	Field Name	Description	Type	Reset
31:5	RESERVED	Reserved.	R	0x0000000
4	INTERRUPT	Generate interrupt when maintenance operation is complete	RW	0
3	INVALIDATE	Invalidate page	RW	0
2	CLEAN	Evict page	RW	0
1	LOCK	Lock page	RW	0
0	PRELOAD	Preload page	RW	0

Table 5-80. Register Call Summary for Register CACHE_MMU_SMALL_MAINT_k

Dual Cortex-M4 IPU Subsystem Register Manual
- IPU_UNICACHE_MMU Register Summary: [0]

Table 5-81. CACHE_MMU_MMUCONFIG

Address Offset	0x0000 04B8		
Physical Address	0x5508 0CB8	**Instance**	IPU_UNICACHE_MMU_IPU
Description	MMU configuration register		
Type	RW		



Bits	Field Name	Description	Type	Reset
31:2	RESERVED	Reserved.	R	0x0000 0000
1	PRIVILEGE	Privilege bit. Once this bit is set, only global flush, debugger, or hardware reset can clear. 0x0: CPU can access everything. 0x1: CPU can only access maintenance, and DMA cannot access MMU at all.	RW	0
0	MMU_LOCK	MMU lock. Once this bit is set only a global flush, debugger, or hardware reset can clear. 0x0: CPU can access everything. 0x1: CPU can only access maintenance, and DMA cannot access the MMU.	RW	0

Table 5-82. Register Call Summary for Register CACHE_MMU_MMUCONFIG

Dual Cortex-M4 IPU Subsystem Register Manual
- IPU_UNICACHE_MMU Register Summary: [0]

5.4.5 IPU_MMU Registers

For information about the IPU_MMU registers and their description, see Chapter 15, *Memory Management Units*.

5.4.6 IPU_Cx_INTC Registers

For information about the IPU_Cx_INTC (NVICs) registers and their description, see the *Arm Cortex-M4 Technical Reference Manual* (available at infocenter.arm.com/help/index.jsp).

5.4.7 IPU_WUGEN Registers

5.4.7.1 IPU_WUGEN Register Summary

Table 5-83 summarizes the IPU_WUGENregister mapping.

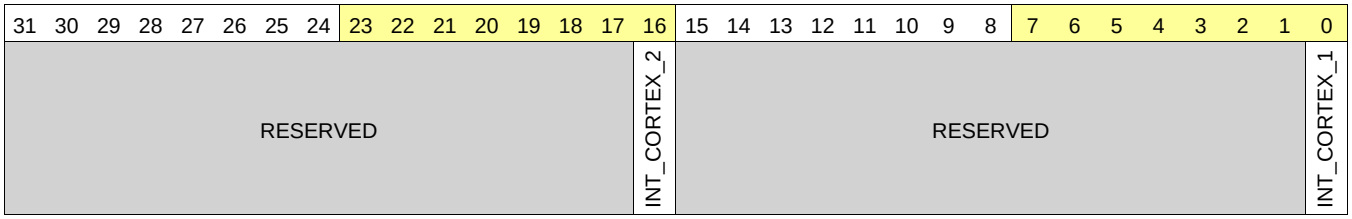
Table 5-83. IPU_WUGEN Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	Physical Address (IPU Private Access)
CORTEXM4_CTRL_REG	RW	32	0x0000 0000	0x5508 1000
STANDBY_CORE_SYSCONFIG	RW	32	0x0000 0004	0x5508 1004
IDLE_CORE_SYSCONFIG	RW	32	0x0000 0008	0x5508 1008
WUGEN_MEVT0	RW	32	0x0000 000C	0x5508 100C
WUGEN_MEVT1	RW	32	0x0000 0010	0x5508 1010
RESERVED	R	32	0x0000 0014	0x5508 1014

5.4.7.2 IPU_WUGEN Register Description

Table 5-84. CORTEXM4_CTRL_REG

Address Offset	0x0000 0000		
Physical Address	0x5508 1000	**Instance**	IPU_WUGEN_IPU
Description	The register is used by one CPU to interrupt the other, thus used as a handshake between the two CPUs 0x0: Interrupt is cleared; 0x1: Interrupt is set.		
Type	RW		



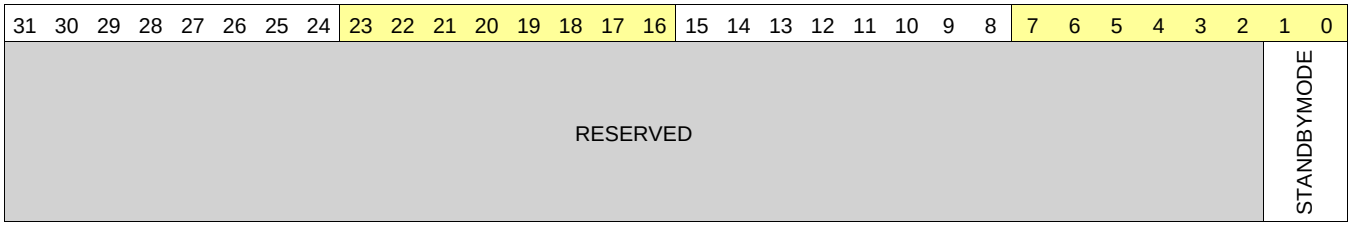
Bits	Field Name	Description	Type	Reset
31:17	RESERVED	Reserved	RW	0x0000 0000
16	INT_CORTEX_2	Interrupt to IPU_C1	RW	0
15:1	RESERVED	Reserved	RW	0x0000 0000
0	INT_CORTEX_1	Interrupt to IPU_C0	RW	0

Table 5-85. Register Call Summary for Register CORTEXM4_CTRL_REG

Dual Cortex-M4 IPU Subsystem Functional Description • Use of Interrupt for IPC: [0][1][2]
Dual Cortex-M4 IPU Subsystem Register Manual • IPU_WUGEN Register Summary: [3]

Table 5-86. STANDBY_CORE_SYSCONFIG

Address Offset	0x0000 0004		
Physical Address	0x5508 1004	**Instance**	IPU_WUGEN_IPU
Description	Standby protocol		
Type	RW		



Bits	Field Name	Description	Type	Reset
31:2	RESERVED	Reserved	RW	0x0000 0000
1:0	STANDBYMODE	0x0: Force-standby mode 0x1: No-standby mode 0x2: Smart-standby mode 0x3: Smart-standby wake-up mode – normal mode to be used	RW	0x3

Table 5-87. Register Call Summary for Register STANDBY_CORE_SYSCONFIG

Dual Cortex-M4 IPU Subsystem Functional Description • STANDBY and IDLE Protocols: [0]
Dual Cortex-M4 IPU Subsystem Register Manual • IPU_WUGEN Register Summary: [1]

Table 5-88. IDLE_CORE_SYSCONFIG

Address Offset	0x0000 0008		
Physical Address	0x5508 1008	**Instance**	IPU_WUGEN_IPU
Description	Idle protocol		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														IDLEMODE	

Bits	Field Name	Description	Type	Reset
31:2	RESERVED	Reserved	RW	0x0000 0000
1:0	IDLEMODE	0x0: Force-idle mode 0x1: No-idle mode 0x2: Smart-idle mode 0x3: Smart-idle wake-up mode – normal mode to be used	RW	0x3

Table 5-89. Register Call Summary for Register IDLE_CORE_SYSCONFIG

Dual Cortex-M4 IPU Subsystem Functional Description • STANDBY and IDLE Protocols: [0]
Dual Cortex-M4 IPU Subsystem Register Manual • IPU_WUGEN Register Summary: [1]

Table 5-90. WUGEN_MEVT0

Address Offset	0x0000 000C		
Physical Address	0x5508 100C	**Instance**	IPU_WUGEN_IPU
Description	This register contains the interrupt mask (LSB) wake-up enable bit per interrupt request: 0x0: Interrupt is disabled; 0x1: Interrupt is enabled.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MIRQ31	MIRQ30	MIRQ29	MIRQ28	MIRQ27	MIRQ26	MIRQ25	MIRQ24	MIRQ23	MIRQ22	MIRQ21	MIRQ20	MIRQ19	MIRQ18	MIRQ17	MIRQ16	MIRQ15	MIRQ14	MIRQ13	MIRQ12	MIRQ11	MIRQ10	MIRQ9	MIRQ8	MIRQ7	MIRQ6	MIRQ5	MIRQ4	MIRQ3	MIRQ2	MIRQ1	MIRQ0

Bits	Field Name	Description	Type	Reset
31	MIRQ31	Interrupt Mask bit 31	RW	0
30	MIRQ30	Interrupt Mask bit 30	RW	0
29	MIRQ29	Interrupt Mask bit 29	RW	0
28	MIRQ28	Interrupt Mask bit 28	RW	0
27	MIRQ27	Interrupt Mask bit 27	RW	0

Bits	Field Name	Description	Type	Reset
26	MIRQ26	Interrupt Mask bit 26	RW	0
25	MIRQ25	Interrupt Mask bit 25	RW	0
24	MIRQ24	Interrupt Mask bit 24	RW	0
23	MIRQ23	Interrupt Mask bit 23	RW	0
22	MIRQ22	Interrupt Mask bit 22	RW	0
21	MIRQ21	Interrupt Mask bit 21	RW	0
20	MIRQ20	Interrupt Mask bit 20	RW	0
19	MIRQ19	Interrupt Mask bit 19	RW	0
18	MIRQ18	Interrupt Mask bit 18	RW	0
17	MIRQ17	Interrupt Mask bit 17	RW	0
16	MIRQ16	Interrupt Mask bit 16	RW	0
15	MIRQ15	Interrupt Mask bit 15	RW	0
14	MIRQ14	Interrupt Mask bit 14	RW	0
13	MIRQ13	Interrupt Mask bit 13	RW	0
12	MIRQ12	Interrupt Mask bit 12	RW	0
11	MIRQ11	Interrupt Mask bit 11	RW	0
10	MIRQ10	Interrupt Mask bit 10	RW	0
9	MIRQ9	Interrupt Mask bit 9	RW	0
8	MIRQ8	Interrupt Mask bit 8	RW	0
7	MIRQ7	Interrupt Mask bit 7	RW	0
6	MIRQ6	Interrupt Mask bit 6	RW	0
5	MIRQ5	Interrupt Mask bit 5	RW	0
4	MIRQ4	Interrupt Mask bit 4	RW	0
3	MIRQ3	Interrupt Mask bit 3	RW	0
2	MIRQ2	Interrupt Mask bit 2	RW	0
1	MIRQ1	Interrupt Mask bit 1	RW	0
0	MIRQ0	Interrupt Mask bit 0	RW	0

Table 5-91. Register Call Summary for Register WUGEN_MEVT0

Dual Cortex-M4 IPU Subsystem Functional Description • Wake-Up Generator (IPU_WUGEN): [0]
Dual Cortex-M4 IPU Subsystem Register Manual • IPU_WUGEN Register Summary: [1]

Table 5-92. WUGEN_MEVT1

Address Offset	0x0000 0010		
Physical Address	0x5508 1010	**Instance**	IPU_WUGEN_IPU
Description	This register contains the interrupt mask (MSB) wake-up enable bit per interrupt request: 0x0: Interrupt is disabled; 0x1: Interrupt is enabled.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MIRQ63	MIRQ62	MIRQ61	MIRQ60	MIRQ59	MIRQ58	MIRQ57	MIRQ56	MIRQ55	MIRQ54	MIRQ53	MIRQ52	MIRQ51	MIRQ50	MIRQ49	MIRQ48	MIRQ47	MIRQ46	MIRQ45	MIRQ44	MIRQ43	MIRQ42	MIRQ41	MIRQ40	MIRQ39	MIRQ38	MIRQ37	MIRQ36	MIRQ35	MIRQ34	MIRQ33	MIRQ32

Bits	Field Name	Description	Type	Reset
31	MIRQ63	Interrupt Mask bit 63	RW	0
30	MIRQ62	Interrupt Mask bit 62	RW	0
29	MIRQ61	Interrupt Mask bit 61	RW	0
28	MIRQ60	Interrupt Mask bit 60	RW	0
27	MIRQ59	Interrupt Mask bit 59	RW	0
26	MIRQ58	Interrupt Mask bit 58	RW	0
25	MIRQ57	Interrupt Mask bit 57	RW	0
24	MIRQ56	Interrupt Mask bit 56	RW	0
23	MIRQ55	Interrupt Mask bit 55	RW	0
22	MIRQ54	Interrupt Mask bit 54	RW	0
21	MIRQ53	Interrupt Mask bit 53	RW	0
20	MIRQ52	Interrupt Mask bit 52	RW	0
19	MIRQ51	Interrupt Mask bit 51	RW	0
18	MIRQ50	Interrupt Mask bit 50	RW	0
17	MIRQ49	Interrupt Mask bit 49	RW	0
16	MIRQ48	Interrupt Mask bit 48	RW	0
15	MIRQ47	Interrupt Mask bit 47	RW	0
14	MIRQ46	Interrupt Mask bit 46	RW	0
13	MIRQ45	Interrupt Mask bit 45	RW	0
12	MIRQ44	Interrupt Mask bit 44	RW	0
11	MIRQ43	Interrupt Mask bit 43	RW	0
10	MIRQ42	Interrupt Mask bit 42	RW	0
9	MIRQ41	Interrupt Mask bit 41	RW	0
8	MIRQ40	Interrupt Mask bit 40	RW	0
7	MIRQ39	Interrupt Mask bit 39	RW	0
6	MIRQ38	Interrupt Mask bit 38	RW	0
5	MIRQ37	Interrupt Mask bit 37	RW	0
4	MIRQ36	Interrupt Mask bit 36	RW	0
3	MIRQ35	Interrupt Mask bit 35	RW	0
2	MIRQ34	Interrupt Mask bit 34	RW	0
1	MIRQ33	Interrupt Mask bit 33	RW	0
0	MIRQ32	Interrupt Mask bit 32	RW	0

Table 5-93. Register Call Summary for Register WUGEN_MEVT1

Dual Cortex-M4 IPU Subsystem Functional Description • Wake-Up Generator (IPU_WUGEN): [0]
Dual Cortex-M4 IPU Subsystem Register Manual • IPU_WUGEN Register Summary: [1]

5.4.8 IPU_Cx_RW_TABLE Registers

5.4.8.1 IPU_Cx_RW_TABLE Register Summary

Table 5-94. IPU_Cx_RW_TABLE Register Summary

Register Name	Type	Register Width (Bits)	Address Offset	Physical Address (IPU Private Access)	Physical Address (L3_MAIN Access)
CORTEXM4_RW_PID1	RW	32	0x0000 0000	0xE00F E000	N/A
CORTEXM4_RW_PID2	RW	32	0x0000 0004	0xE00F E004	N/A

Table 5-94. IPU_Cx_RW_TABLE Register Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	Physical Address (IPU Private Access)	Physical Address (L3_MAIN Access)
RESERVED	R	32	0x0000 0008	0xE00F E008	N/A

5.4.8.2 IPU_Cx_RW_TABLE Register Description

Table 5-95. CORTEXM4_RW_PID1

Address Offset	0x0000 0000		
Physical Address	0xE00F E000 0xE00F E000	**Instance**	IPU_C0_RW_TABLE_IPU IPU_C1_RW_TABLE_IPU
Description	Peripheral Identification register– allows the user software to differentiate between the two Arm Cortex-M4 processors (two CPUs). The same piece of code running on the two CPUs can result in different execution (for example, branch to different location) depending on the address stored in the register. The address is stored by the BIOS code. The register cannot be accessed when the BIOS code is running (used).		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BASEADD1																															

Bits	Field Name	Description	Type	Reset
31:0	BASEADD1	IPU_ROM memory address	RW	0x0000 0000

Table 5-96. Register Call Summary for Register CORTEXM4_RW_PID1

Dual Cortex-M4 IPU Subsystem Register Manual
- IPU_Cx_RW_TABLE Register Summary: [0]

Table 5-97. CORTEXM4_RW_PID2

Address Offset	0x0000 0004		
Physical Address	0xE00F E004 0xE00F E004	**Instance**	IPU_C0_RW_TABLE_IPU IPU_C1_RW_TABLE_IPU
Description	Peripheral Identification register – allows the user software to differentiate between the two Arm Cortex-M4 processors (two CPUs). The same piece of code running on the two CPUs can result in different execution (for example, branch to different location) depending on the address stored in the register. The address is stored by the BIOS code. The register cannot be accessed when the BIOS code is running (used).		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BASEADD2																															

Bits	Field Name	Description	Type	Reset
31:0	BASEADD2	IPU_ROM memory address	RW	0x0000 0000

Table 5-98. Register Call Summary for Register CORTEXM4_RW_PID2

Dual Cortex-M4 IPU Subsystem Register Manual
- IPU_Cx_RW_TABLE Register Summary: [0]

Embedded Vision Engine

This chapter describes the embedded vision engine (EVE) for the device.

Topic **Page**

6.1 Embedded Vision Engine (EVE) Subsystem

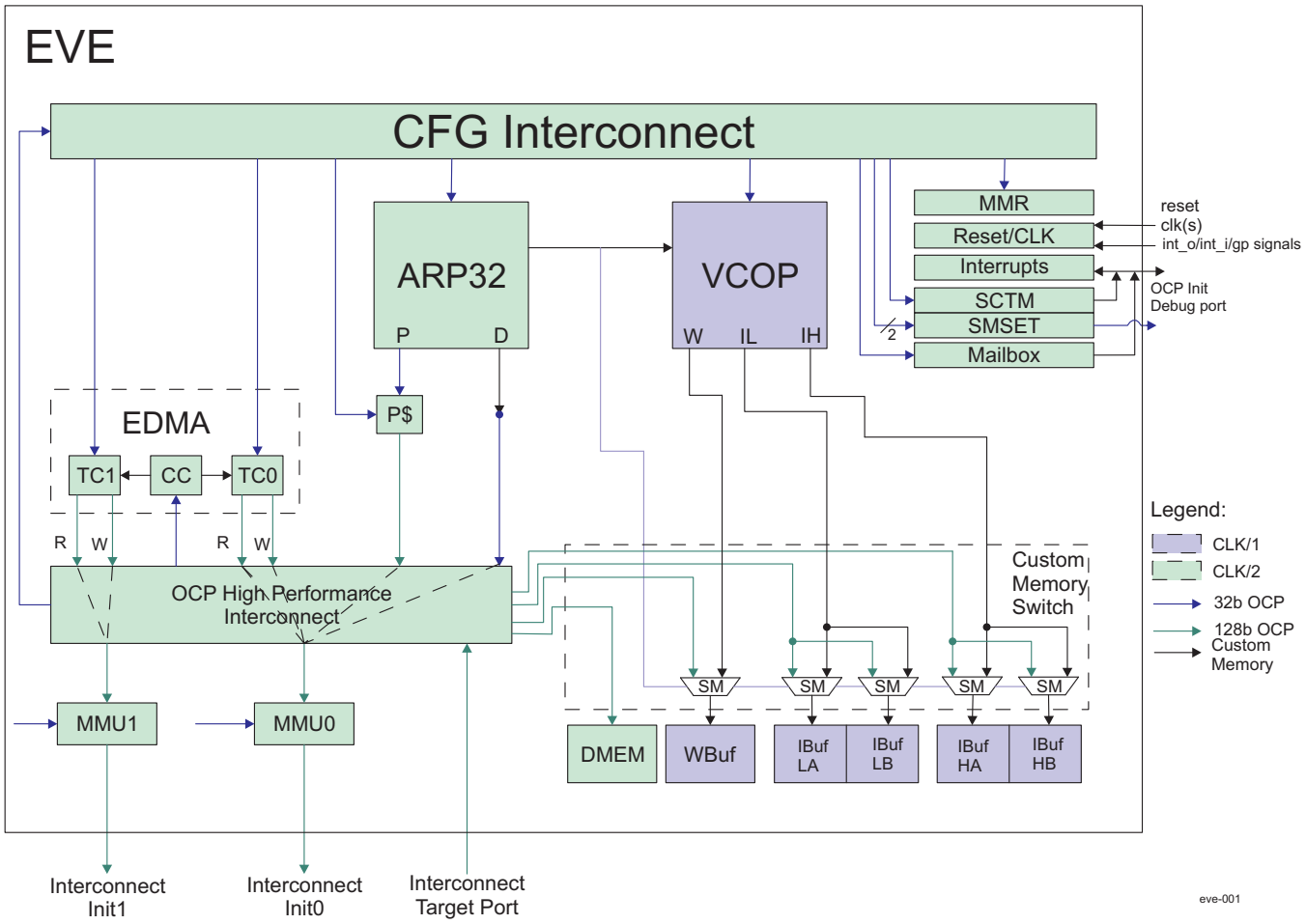
6.1.1 EVE Overview

The embedded vision engine (EVE) module is a programmable imaging and vision processing engine, intended for use in devices that serve customer electronics imaging and vision applications. Its programmability meets late-in-development or post-silicon processing requirements, and lets third parties or customers add differentiating features in imaging and vision products.

The device includes one instantiation of the EVE engine. A single EVE module consists of an ARP32 scalar core, a vector coprocessor (VCOP) vector core, and an Enhanced DMA (EDMA3) controller.

Figure 6-1 is a high-level block diagram of the EVE module.

Figure 6-1. EVE Overview



The ARP32 scalar core is the subsystem controller. It coordinates internal EVE interaction, as well as interaction with other components in the system on chip (SoC), like host processor and the DSP. The VCOP is a SIMD engine with built-in loop control and address generation.

The EDMA block is the local DMA. It is used for transferring data between system memories (typically SDRAM and/or L3 SRAM) and internal EVE memories.

Module interconnect is conceptually broken into two parts: A high-performance interconnect and a configuration interconnect. The high-performance interconnect serves as primary high bandwidth (partial) crossbar connection between the EDMA, ARP32, DMA, OCP initiator/target buses and EVE memories. The configuration interconnect provides connectivity to the various memory-mapped registers (MMR) within the EVE module.

The custom memory switch provides a statically muxed low-latency and high-bandwidth connection between the VCOP and the internal EVE memories. The custom memory switch also provides any accesses from the high-performance interconnect and EVE memories.

The interrupt controller (INTC) handles incoming interrupts, merging them with internal interrupt sources to drive the interrupt inputs of the ARP32 .

The MMR block includes control and status bits for general EVE functions that do not reside in other blocks, including reset and clock controller registers and memory static mux ownership.

The SCTM block includes debug logic that supports the counting and timing of various EVE level events, namely VCOP stall counts, VCOP access counts, ARP32 cache miss counts, and so on. The SMSET block provides software message tracing, as well as event tracking.

The reset-clock controller handles subsystem reset and power-management functions.

The EVE engine includes the following main features:

- Two 128-bit interconnect initiator ports used for:
  - Paging between system-level memory (L3 SRAM/DDR) and EVE memory (primarily IBUF, WBUF)
  - ARP32 program fetches to system memory (through program cache)
  - ARP32 load or store requests to system memory
  - ARP32 program cache-related read requests, including prefetch/preload requests
- 128-bit interconnect target port used for system-level host or DMA access to EVE memory or MMR space
- Scalar core (ARP32) with the following features:
  - 32KB program cache (direct mapped and prefetch)
  - 32KB data memory (DMEM)
- Vector core (VCOP):
  - 32KB working buffer (WBUF)
  - 16KB image buffer low copy A (IBUFLA)
  - 16KB image buffer low copy B (IBUFLB)
  - 16KB image buffer high copy A (IBUFHA)
  - 16KB image buffer high copy B (IBUFHB)
- EDMA channel controller (EDMACC): 128 PaRAM entries, 2 Queues
- EDMA transfer controllers: two instances, 2k FIFO each
- Memory Management Units (MMUs):
  - 32-entry TLB per MMU
  - Page walking with hardware
  - EDMA accesses and ARP32 program or data accesses to system memory space
  - Can limit EVE accesses to desired subset of system addresses
- Configuration interconnect for MMR and debug accesses
- High-performance interconnect for high throughput and high concurrency data transfers between connected endpoints
- Multiple interrupts for interrupt mapping, DMA event mapping, and interprocessor handshaking
- Support for slave idle and master standby protocols for clock gating
- No support for retention and memory array off modes
- Error detection on all memories:
  - Single bit error detect on DMEM, WBUF, IBUFLA, IBUFLB, IBUFHA, and IBUFHB
  - Double bit error detect on program cache
- Invalid instruction detection in the two processor units (ARP32 and VCOP)
- Debug support
  - Subsystem Counter Timer Module (SCTM) for counting and measuring of VCOP, EVE program

cache, and EDMA performance-related state
  - Software Messaging System Event Trace (SMSET) for trace of software messages and hardware events
  - ARP32 debug support: State visibility, breakpoint, run control, cross-triggering
  - VCOP debug support: State visibility and run control
- Interprocessor communication: Internal Mailbox for DSP/EVE communication

6.1.1.1 EVE Memories

Table 6-1 lists the memories inside the EVE subsystem. See Section 6.1.3.3, *Internal Memory Overview*, for an overview of each memory and its organization, see Section 6.1.3.15, *Memory Map*, for the subsystem memory mapping.

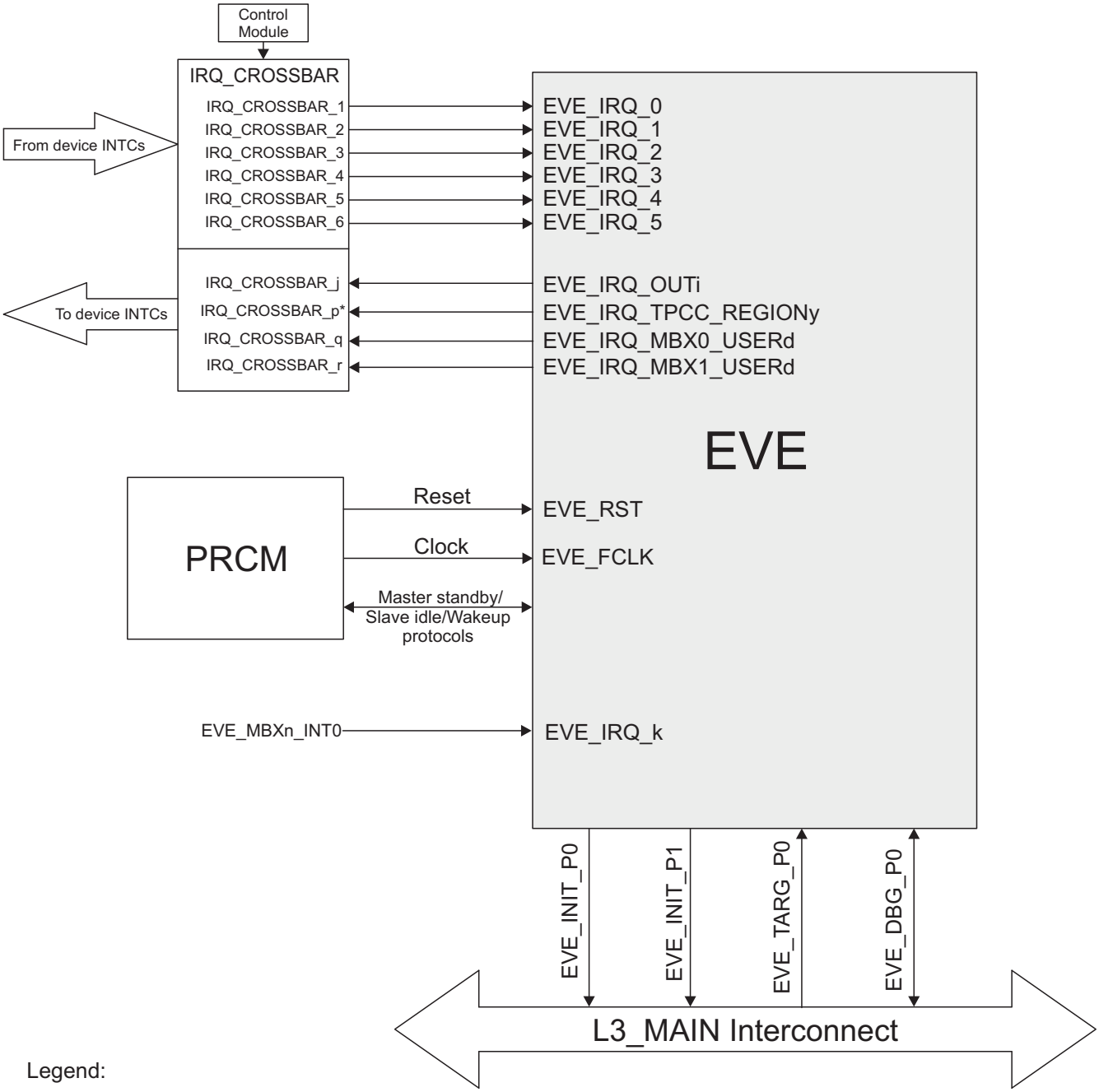
Table 6-1. EVE Subsystem Memory Blocks

Memory	Organization	Size	Function
PMEM	1024 × 256	32KB	ARP32 program cache
DMEM	2048 × 128	32KB	ARP32 data memory
WBUF	8 × 1024 × 32	32KB	VCOP working buffer
IBUFLA	8 × 512 × 32	16KB	Image buffer low copy A
IBUFLB	8 × 512 × 32	16KB	Image buffer low copy B
IBUFHA	8 × 512 × 32	16KB	Image buffer high copy A
IBUFHB	8 × 512 × 32	16KB	Image buffer high copy B

6.1.2 EVE Integration

Figure 6-2 shows the EVE integration inside the device. EVE includes a number of mailboxes, interrupts, DMA events, and general-purpose I/Os (GPIOs) to facilitate handshaking between the EVE modules and also between other initiators or targets such as DSP, IPU, VIP and the PRCM module.

Figure 6-2. EVE Integration



Legend:

j = 168 to 171
p = 281 to 283 (p = 398, for y = 4)*
q = 284 to 286
r = 287 to 289
n = 2, 3, 4

i = 0, 1, 2, 3
y = 1 to 4
d = 1 to 3

k = 28, 29, 30

eve-002

NOTE: Figure 6-2 is an overview of the EVE integration. For detailed explanation of the interrupt, mailbox, and general-purpose signals coming out or going in EVE subsystem, see Figure 6-3.

Table 6-2 through Table 6-4 summarize the integration of the EVE subsystem in the device.

Table 6-2. Integration Attributes

Module Instance	Attributes	
	Power Domain	Interconnect
EVE	PD_EVE	L3_MAIN

Table 6-3. Clocks and Resets

Module Instance	Destination Signal Name	Source Signal Name	Source	Description
Clocks				
EVE	EVE_FCLK	EVE_GFCLK	PRCM	EVE functional clock, gated. For more information about clock gating and management, refer to Chapter 3, *Power, Reset, and Clock Management (PRCM).*
Resets				
EVE	EVE_RST	EVE_RST	PRCM	For information about PRCM reset sources and distribution, see Section 3.1.1.1, *Reset Management Functional Description*, in Chapter 3, *Power, Reset, and Clock Management (PRCM).*
		EVE_CPU_RST		
		EVE_PWRON_RST		

Table 6-4. Hardware Requests

Interrupt Requests				
Module Instance	**Source Signal Name**	**Destination IRQ_CROSSBAR Input**	**Default Mapping**	**Description**
EVE	ELM_IRQ	IRQ_CROSSBAR_1	EVE_IRQ_0	Error location process completion
	EXT_SYS_IRQ_1	IRQ_CROSSBAR_2	EVE_IRQ_1	External interrupt (active low)
	CTRL_MODULE_CORE_IRQ_SEC_EVTS	IRQ_CROSSBAR_3	EVE_IRQ_2	Firewall out-band error
	L3_MAIN_IRQ_DBG_ERR	IRQ_CROSSBAR_4	EVE_IRQ_3	Interconnect debug error
	L3_MAIN_IRQ_APP_ERR	IRQ_CROSSBAR_5	EVE_IRQ_4	Interconnect application or non-attributes errors on L3
	PRM_IRQ_MPU	IRQ_CROSSBAR_6	EVE_IRQ_5	PRCM module interrupt
	MAILBOX2_INTERRUPT0	N/A	EVE_IRQ_28	Mailbox 2 interrupt 0
	MAILBOX3_INTERRUPT0	N/A	EVE_IRQ_29	Mailbox 3 interrupt 0
	MAILBOX4_INTERRUPT0	N/A	EVE_IRQ_30	Mailbox 4 interrupt 0
	EVE_IRQ_OUT0	IRQ_CROSSBAR_168	N/A	EVE interrupt output
	EVE_IRQ_OUT1	IRQ_CROSSBAR_169	N/A	EVE interrupt output
	EVE_IRQ_OUT2	IRQ_CROSSBAR_170	N/A	EVE interrupt output
	EVE_IRQ_OUT3	IRQ_CROSSBAR_171	N/A	EVE interrupt output
	EVE_IRQ_TPCC_REGION1	IRQ_CROSSBAR_281	N/A	EVE TPCC region interrupt1

Table 6-4. Hardware Requests (continued)

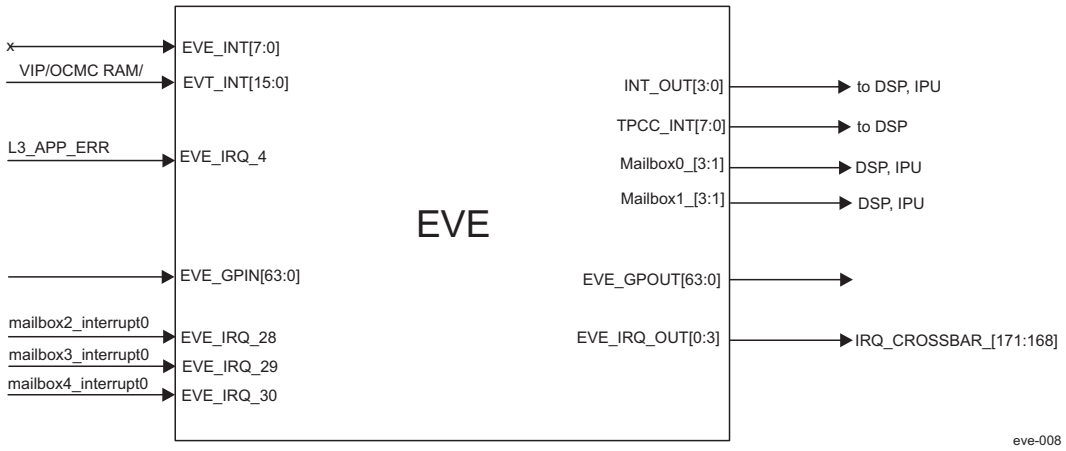
Interrupt Requests				
	EVE_IRQ_TPCC_REGION2	IRQ_CROSSBAR_282	N/A	EVE TPCC region interrupt2
	EVE_IRQ_TPCC_REGION3	IRQ_CROSSBAR_283	N/A	EVE TPCC region interrupt3
	EVE_IRQ_MBX0_USER1	IRQ_CROSSBAR_284	N/A	EVE Mailbox 0 user 1
	EVE_IRQ_MBX0_USER2	IRQ_CROSSBAR_285	N/A	EVE Mailbox 0 user 2
	EVE_IRQ_MBX0_USER3	IRQ_CROSSBAR_286	N/A	EVE Mailbox 0 user 3
	EVE_IRQ_MBX1_USER1	IRQ_CROSSBAR_287	N/A	EVE Mailbox 1 user 1
	EVE_IRQ_MBX1_USER2	IRQ_CROSSBAR_288	N/A	EVE Mailbox 1 user 2
	EVE_IRQ_MBX1_USER3	IRQ_CROSSBAR_289	N/A	EVE Mailbox 1 user 3
	EVE_IRQ_TPCC_REGION4	IRQ_CROSSBAR_398	N/A	EVE TPCC region interrupt4
No DMA Requests				

NOTE: The **Default Mapping** column in Table 6-4 shows the default mapping of module IRQ source signals. These IRQ source signals can also be mapped to other lines of each device INTC through the IRQ_CROSSBAR module.
For more information about the IRQ_CROSSBAR module, see Section 13.4.6.4, *IRQ_CROSSBAR Module Functional Description*, in Chapter 13, *Control Module*.

6.1.2.1 EVE Recommended Connections

Figure 6-3 show the interrupts, general purpose signals, and DMA requests for EVE instance.

Figure 6-3. EVE Signals



EVE INT[7:0] Usage:

EVE_INT[7:0] are not connected to any chip-level or inter-EVE interrupt. These interrupts can be tied to any chip-level mailboxes or any other interrupts that are intended as wake-up interrupts.

TPCC Interrupts Usage:

TPCC interrupts are used as region-based interrupts. The transfers for different processors are divided into separate regions and on completion of a transfer the associated TPCC_INT interrupts the corresponding processor. Region 0 interrupt is reserved for the internal ARP32 processor. Region 1 is reserved and not used. Region 2 is routed to DSP1, region 3 to DSP2, and region 4 to IPU.

6.1.3 EVE Functional Description

6.1.3.1 EVE Connection ID (ConnID) Mapping

The ConnID value is used for error reporting in the memory switch error register (EVE_MSW_ERR[24:16] CONNID). Table 1-5 lists the EVE internal initiators and their assigned 9-bit ConnID values.

Table 6-5. EVE Internal ConnID Mapping

Initiator	CONNID Value
System initiators (through OCP target bus)	0 to FFh
ARP32	100h
VCOP	101h
EDMA TC0	102h
EDMA TC1	103h

6.1.3.2 EVE Processors Overview

6.1.3.2.1 Scalar Core (ARP32)

The scalar core (ARP32) executes conventional RISC instructions. The scalar core is based on the ARP32 processor, which contains the following five functional units:

- The L-unit performs the logical, shift, rotation, extraction, reverse, clear, set, and equal operations.
- The S-unit performs the move operations.
- The D-unit performs arithmetic operations that include compare less than and greater than instructions, address calculation for load and store instructions, PC calculations for branch instructions and stack pointer increment/decrement for PUSH/POP and CALL/RETURN instructions.
- The M-unit performs multiplication.
- The DIV-unit is used for bit serial divide logic.

The ARP32 includes 32-bit program fetch and 32-bit data access bus. For additional information about ARP32 core see Section 6.2, *ARP32 CPU and Instruction Set*.

6.1.3.2.2 VCOP

The VCOP is an SIMD engine with built-in loop control and address generation. The VCOP is programmed in array of 2D block processing level. The vector core has the following resources:

- 4 nested for loops, with loop variables i1, i2, i3, and i4
- 8 address generators, each capable of 4-dimensional addressing; address pattern is *base + i1*const1 + i2*const2 + i3*const3 + i4*const4*
- 16-entry vector register file, each entry is 8-way SIMD × 40-bit signed (sign-extended or zero-padded from 8/16/32-bit signed/unused memory data, or zero-padded from operation upon register data)
- Two general-purpose functional units, each N-way SIMD, N = 2, 4, 8, 16, 32
- Table lookup unit supporting up to N parallel histogram operations
- 8 load units
- 8 store units
- Flat versus Aliased view of EVE memory

In addition, the VCOP supports the following functions:

- Generic compute
- Table lookup
- Histogram and weighted histogram

For additional information about VCOP, see Section 6.3, *VCOP CPU and Instruction Set*.

6.1.3.2.3 Scalar-Vector Interaction

The ARP32 scalar and vector cores share one program memory, but can execute in parallel on separate threads. The scalar core has sole control of the program memory, so all vector instructions are accessed by the scalar core. When a vector instruction is recognized as such it is relayed through the scalar-vector interface to the vector core for execution.

6.1.3.3 Internal Memory Overview

This section provides a functional overview for each of the memory endpoints listed in Table 6-6.

Table 6-6. Internal Memory Blocks

Memory	Organization	Size	Function
PMEM	1024 × 256	32KB	ARP32 program cache
DMEM	2048 × 128	32KB	ARP32 data memory
WBUF	8 × 1024 × 32	32KB	VCOP working buffer
IBUFLA	8 × 512 × 32	16KB	Image buffer low copy A
IBUFLB	8 × 512 × 32	16KB	Image buffer low copy B
IBUFHA	8 × 512 × 32	16KB	Image buffer high copy A
IBUFHB	8 × 512 × 32	16KB	Image buffer high copy B

NOTE: EVE internal memories, PMEM, DMEM, WBUFs and IBUFs, are visible in the SoC. Device host processors like IPU and DSP access the EVE memories through its OCP ports. For EVE memory mapping, see Table 6-25. For information about the address space of EVE subsystem modules and memories from device point of view see, Chapter 2, *Memory Mapping*.

6.1.3.3.1 Program Cache/Memory

For detailed explanation of the program cache operation, see Section 6.1.3.4, *Program Cache Architecture*.

6.1.3.3.2 ARP32 Data Memory (DMEM)

The ARP32 data memory is logically organized as four banks of 32-bit memories. It is the primary location for storing run-time variables and stack for the software execution environment for the ARP32 core. This memory is byte addressable, and is accessed with byte granularity by all valid initiators. DMEM is accessed at peak rate of up to 4B/cycle by the ARP32 core, or up to 16B/cycle by the system components (EVE EDMA, DMA target bus). The vector core cannot access this memory.

The arbitration between the ARP32 and DMA/System accesses to DMEM is done on round-robin scheme. That is a continuous stream of requests from two (or more) initiator ports results in each requestor gaining access on alternating burst boundaries, therefore DMA/System accesses to DMEM should be minimized in order to avoid impacting the ARP32 performance. Only a single access (ARP32 or DMA/System access) occurs in a single cycle.

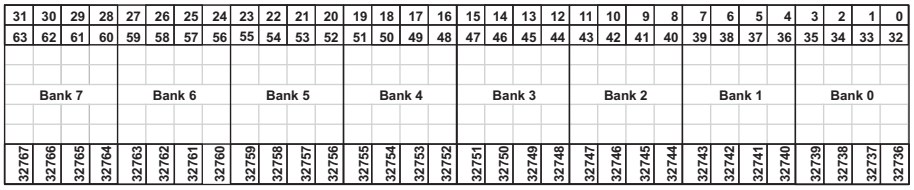
6.1.3.3.3 WBUF

VCOP WBUF is the primary location for VCOP look-up tables (LUTs) and other relatively long-lived data buffers. This memory is byte addressable. On a time-slot basis (as controlled by the ARP32 core) WBUF ownership is assigned to either the system (ARP32, EDMA, DMA target bus) or the VCOP.

WBUF is accessible by VCOP, ARP32, and system components (EDMA, DMA target bus) depending on the programmed ownership. The primary master of working buffer is the VCOP.

As shown in Figure 6-4, VCOP can access each of the eight banks of memory independently, giving a total of 256-bits per cycle. ARP32 core can access up to 32-bits per cycle in a given bank and EDMA (and other system components) can access up to 128-bits per cycle of contiguous banks of memory within a 128-bit aligned window (this is a case when there are no independent bank accesses, and single access does not cross 128-bit boundary).

Figure 6-4. WBUF Bank Organization



EDMA versus system accesses (through OCP target bus) are arbitrated dynamically, on OCP bus boundaries, with round-robin policy within the OCP high performance Interconnect. EDMA/System versus ARP32 accesses are arbitrated at dataphase boundaries.

The master that owns the WBUF is pseudostatically defined in the EVE_MSW_CTL[16] WBUF bit. When EVE_MSW_CTL[16] WBUF = 0x0 the static mux (SM) provides a connection from the OCP high-performance interconnect (EDMA, ARP32 or external-to-EVE initiated access) to the WBUF memory, and VCOP accesses are not allowed. Otherwise, (EVE_MSW_CTL[16] WBUF = 0x1) SM provides a connection from VCOP to the WBUF memory and system accesses are not allowed.

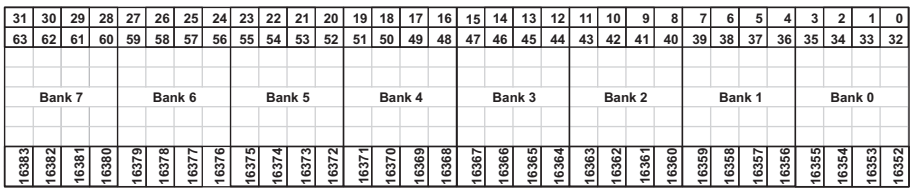
In case VCOP or system initiators access the WBUF address location while WBUF is not owned, an error is captured in the EVE memory switch error register (EVE_MSW_ERR) and EVE memory switch error the address register (EVE_MSW_ERRADDR) and the corresponding flag in EVE_MSW_ERR_IRQSTATUS_RAW register is set. Debug accesses do not set the error registers or interrupt, but result in OCP ERR response.

6.1.3.3.4 Image Buffers–IBUFLA, IBUFLB, IBUFHA, and IBUFHB

EVE image buffers are the primary location for the time-slot based data buffers. These buffers are built to facilitate ping-pong ownership and accesses by the EDMA and the VCOP. On a time-slot bases (as controlled by the ARP32 core), each of the respective image buffers can be assigned ownership to either the System (ARP32, EDMA, OCP target bus) or the VCOP.

IBUFLx and IBUFHx (x = A, B) are typically used as output buffers, and as such VCOP can concurrently access up to eight banks of memory in the VCOP-owned IBUFLA and IBUFLB banks and up to eight banks of memory in one of the VCOP owned IBUFHA and IBUFHB (see Figure 6-5). ARP32 accesses 32-bits per cycle in one of the system-owned IBUF banks.

Figure 6-5. IBUF Bank Organization



EDMA (and other EVE system components) access up to 128 bits per cycle of contiguous banks of memory (in case there are no independent bank accesses) within a 128-bit window (a single access does not cross 128-bit boundary) for each system-owned corresponding IBUF bank. For example, if all banks are system owned, EDMA TC0 (or any other EVE system resource) accesses IBUFL at 128-bits per cycle in parallel with EDMA TC1 accessing IBUFH at 128-bits per cycle, giving a total throughput of 256-bits per cycle.

EDMA versus system versus ARP32 accesses are arbitrated dynamically on OCP burst boundaries. The arbitration is done with round-robin policy inside the EVE high-performance interconnect.

The ownership of each IBUF is pseudostatically controlled by the EVE_MSW_CTL [12]IBUFHB, EVE_MSW_CTL [8]IBUFLB, EVE_MSW_CTL [4]IBUFHA, EVE_MSW_CTL [0]IBUFLA bits. Ownership is also internally changed by ARP32 custom instruction or through writes to the memory-mapped address.

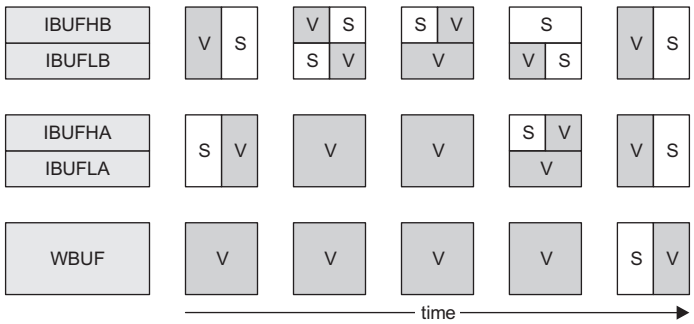
By setting EVE_MSW_CTL [12/8/4/0]IBUF = 0x1 the ownership of the IBUF memory is granted to VCOP, otherwise (EVE_MSW_CTL:IBUF = 0x0) the memory is owned by the EVE system resources. If IBUF is system owned, then the SM provides a connection from the EVE high-performance interconnect (for EDMA, ARP32, or external-to-EVE initiated access) to the corresponding IBUF memory, and VCOP accesses are not allowed. If IBUF is VCOP owned, the static mux provides a connection from VCOP to the IBUF memory, and system accesses are not allowed.

Ownership of each of the four IBUF regions (IBUFLA, IBUFLB, IBUFHA, and IBUFHB) is independently programmable with no restrictions. Aliased versus nonaliased mode is controlled by setting the EVE_MEMMAP register. In nonaliased mode it is possible for the VCOP core to own all four IBUFs, or for the system resources to own all four IBUFs or any combination. In aliased mode VCOP owns only one of the IBUFHx regions and one of the IBUFLx regions (x = A, B) because both A and B copies of the IBUFs are mapped to the same address.

In case VCOP or system initiators access the WBUF address location while IBUF is not owned, an error is captured in EVE memory switch error register (EVE_MSW_ERR) and EVE memory switch error address register (EVE_MSW_ERRADDR) and the corresponding flag in the EVE_MSW_ERR_IRQSTATUS_RAW register is set. Debug accesses do not cause the error registers or interrupt to be set, but result in OCP ERR response.

Figure 6-6 shows the typical ownership patterns that offer different amounts of memory/bandwidth to VCOP (V) versus system (S) accesses.

Figure 6-6. VCOP VersusSystem Memory Ownership Examples



6.1.3.3.5 Memory Switch Error Registers

The memory switch error registers capture the first occurrence of an error caused by a functional access. Software must service the error and clear the MMR to capture subsequent errors. The registers capture the identity of the specific requester through the OCP ConnID (see Table 6-5) field and the specific address that is requested.

The MSBs of the address is always 0x400, which matches the internal (local) address view. For system accesses, this does not necessarily match the system view base address. For IBUFL and IBUFH, the physical address offset is captured even in aliased mode (for example, IBUFLB address at 0x4007 0000 is used in both aliased and nonaliased mode).

Unlike the parity/error detect registers, that provide unique error registers per memory type, the memory switch error registers capture errors regardless of which specific memory is accessed.

Debug accesses do not cause the error or interrupt registers to be set, but result in OCP error response.

The memory switch error registers are:

- EVE_MSW_ERR– indicates whether the error is system-, EDMA-, VCOP-, or ARP32-initiated buffer ownership error and captures the corresponding ConnID

- EVE_MSW_ERRADDR– Captures the physical address of memory switch error

Associated error interrupts are mapped on:

- EVE_MSW_ERR_IRQSTATUS_RAW - indicates error interrupt
- EVE_MSW_ERR_IRQSTATUS - indicates error interrupt if event is enabled
- EVE_MSW_ERR_IRQENABLE_SET - enables the corresponding error interrupt
- EVE_MSW_ERR_IRQENABLE_CLR - disables the corresponding error interrupt

6.1.3.3.6 Memory Error Detection

EVE supports parity-based error detection for all internal data memories (including DMEM, WBUF, IBUFLA, IBUFLB, IBUFHA, and IBUFHB) on the minimum access size granularity of 8 bits (that is, there is 1 bit of parity per byte of memory). EVE subsystem also implements double bit error detection for program cache SRAM through a distance 3, 10-bit Hamming code. The 10-bit Hamming code is also applied to the tag and address for a particular cache line.

When parity is enabled, write accesses to any memory cause the parity/encoding bit(s) to be calculated for the specific write data (along with tag and address value for program cache) and written to the corresponding encoding location. For all read accesses from any data memory, the parity bit is read and compared against previously calculated parity for the current pattern in the memory. For program cache, only ARP32 program fetches perform stored versus expected encoding comparison (accesses through program cache OCP debug target port do not result in code calculation). If a mismatch occurs the error details are recorded in parity error MMR (different for each memory - IBUF, WBUF, DMEM) and the associated interrupt is asserted. At the same time, an OCP error response is generated.

The registers that capture program cache-related parity errors, ARP32 DMEM-related parity errors, WBUF-related parity errors, and IBUF-related parity errors are:

- EVE_PMEM_ED_CTL– Enables and disables error PMEM detection logic. Also lets the user invert error detection logic.
- EVE_PMEM_ED_STAT–Status register for the detected PMEM error
- EVE_PMEM_EDADDR– Physical address of parity error
- EVE_DMEM_ED_CTL– Enables/Disables DMEM error detection logic. Also lets the user invert error detection logic.
- EVE_DMEM_ED_STAT– Status register for the detected DMEM error
- EVE_DMEM_EDADDR– Physical address of parity error
- EVE_DMEM_EDADDR_BO– Address byte offset for parity error
- EVE_WBUF_ED_CTL– Enables/Disables WBUF error detection logic. Also lets the user invert error detection logic.
- EVE_WBUF_ED_STAT– Status register for the detected WBUF error
- EVE_WBUF_EDADDR– Physical address of parity error
- EVE_WBUF_EDADDR_BO– Address byte offset for parity error
- EVE_IBUF_ED_CTL– Enables/Disables IBUF error detection logic. Also lets the user invert error detection logic.
- EVE_IBUF_ED_STAT– Status register for the detected IBUF error
- EVE_IBUF_EDADDR– Physical address of parity error
- EVE_IBUF_EDADDR_BO– Address byte offset for parity error

Data returned to the requestor is not modified even in the case of error. The requestor consumes the data potentially causing execution or propagation of bad code and/or data. For a description of the recovery mechanism, see Section 6.1.3.3.6.4, *Parity Error Recovery*.

Each MMR captures the first occurrence of an error. Software then services the error and clears the MMR to capture subsequent errors (generation of OCP error responses continue if additional errors are detected). The parity error MMRs include bit fields to describe which initiator generated the memory access and the specific byte address. Debug accesses do not cause the error or interrupt registers to be set, but result in OCP error response.

Error interrupts are routed to both ARP32 and as EVE outputs to the device level host. ARP32 services interrupts caused by VCOP, EDMA, or OCP target accesses. However, for errors caused by ARP32 accesses, the ARP32 core is unable to service interrupt, and either the DSP or device local CPU detects and services those errors. This detect and service action is described in Section 6.1.3.9.1, *EVE Interrupt Sources – Memory Switch and Parity Error Interrrupts*.

6.1.3.3.6.1 Captured Address – EDADDR and EDADDR_BO

The EDADDR registers (EVE_PMEM_EDADDR, EVE_DMEM_EDADDR, EVE_WBUF_EDADDR, and EVE_IBUF_EDADDR) capture the memory width aligned address for the faulting address. A byte offset bit array for the corresponding memory (EVE_PMEM_EDADDR_BO, EVE_DMEM_EDADDR_BO, EVE_WBUF_EDADDR_BO, and EVE_IBUF_EDADDR_BO) is used to indicate which specific bytes relative to the aligned address is at fault. For most requestors this gives an accurate view of the specific byte for which parity error was detected.

For VCOP, because independent word addresses are generated to the eight banks of IBUF and WBUF, the BANK0 address is latched in the EDADDR register even if that bank is not at fault. In the case of linear addressing, BANK0 + byte_offset mask gives an accurate view. In the case of table-base loads (each bank uses a different address) the reported BANK0 address and the byte offset are used to post-mortem determine which specific address have the error.

The MSBs of the address are always 0x400, which matches the internal (local) address view. For system accesses this does not necessarily match the system view base address. For IBUFs the physical address offset is captured in both aliased and nonaliased mode (for example, IBUFLB address at 0x4007 is used in both aliased and nonaliased modes).

NOTE: EDADDR_BO does not exist for PMEM memory. PMEM only supports Hamming code-based error detection. PMEM does not support parity per 8-bit detection (unlike the data memories).

6.1.3.3.6.2 Modes of Operation

The EVE error detection includes two basic modes of operation by setting the EN bit in the EVE_<MEMORY>_ED_CTRL (EVE_PMEM_ED_CTL, EVE_DMEM_ED_CTL, EVE_WBUF_ED_CTL and EVE_IBUF_ED_CTL) register, as listed in Table 6-7.

Table 6-7. Error Detection Modes

Mode	Description
EN	Error detection logic is enabled. Writes update parity. Reads check parity.
DIS	Error detection logic is disabled and can be clock gated. Writes do not modify parity. Reads do not check parity.

NOTE: No parity valid bit exists in addition to a parity state bit. After an existing reset, the ED logic is in disabled state. In this mode the parity RAMs are clock gated to minimize power consumption (though state must be retained). When transitioning to enable state, the underlying SRAM + Parity is in a state that may result in parity mismatch.

For ARP32 program cache, the program cache automatically goes through an Invalidate All sequence; during this time the ARP32 program cache fetch path is stalled. After all lines are invalidated, any subsequent ARP32 request is serviced as cache miss. The newly fetched cacheline is written to PMEM along with valid hamming code bits, and subsequent cache hits check the parity for each instruction fetch. For debug and configuration accesses, the parity and encoding bits are not checked. Debug software uses the tag state to produce an appropriate visual for debug purposes.

For data memory (DMEM, WBUF, and IBUF), it is possible (though not recommended for functional and nontest software) for ARP32, VCOP, or DMA to read a memory location for which the parity bit is not set. After enabling EVE_<DMEM, WBUF, or IBUF>_ED_CTL for the corresponding memory, software must initialize all of the memory, which results in setting a valid parity bit. After this, any parity mismatch results in error signaling, as summarized in the previous sections.

6.1.3.3.6.3 Parity Error Testability

For the software to test their parity error interrupt service routine (ISRs), parity errors can be forced to occur. The EVE subsystem does this by providing an INVERT mode of operation per memory endpoint (by setting the INV bit in the EVE_<PMEM, DMEM, WBUF, or IBUF>_ED_CTL register). When INVERT mode is set (EVE_<MEMORY>_ED_CTL[1] INV = 0x1), and parity is enabled, all read accesses to the corresponding memory return the inverse value of the stored parity bit before the parity check comparison. As a result a parity error is latched and reported through an interrupt, as described in Section 6.1.3.3.5, *Memory Switch Error Registers.* For program memory, the INV bit causes the encoding calculation to return the inverted value for pass or fail.

For testing the ARP32 parity error ISR, these registers can be set asynchronously by the host processor or system EDMA controller. This setting mimics the random nature of radiation-induced soft error.

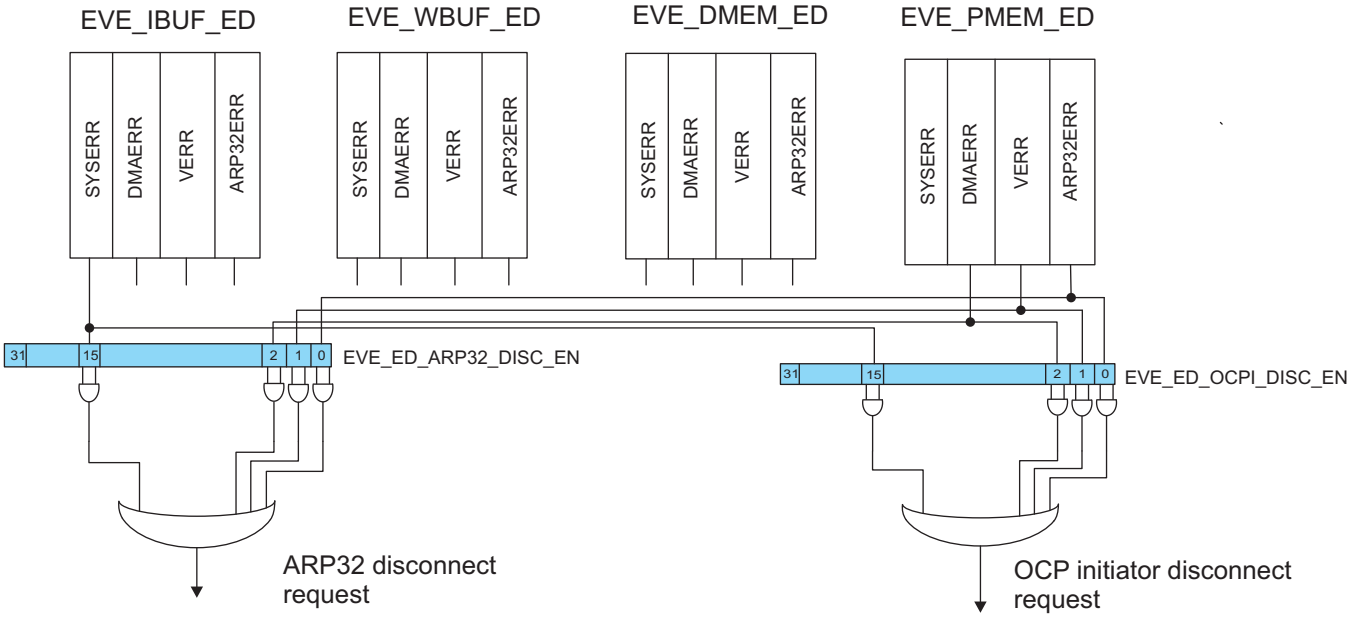
Parity or Hamming errors can be injected to a specific address after the initialization sequence. Parity or error detection is placed in disable mode (EVE_<MEMORY>_ED_CTL[0] EN = 0x0), and software can write an incorrect value to the desired address. Writes during disable mode do not modify the parity bit or encoding bits, thus the stored parity or encoding bits are considered incorrect. The memory then can be enabled. Any subsequent read access to that memory location results in a parity error mismatch.

6.1.3.3.6.4 Parity Error Recovery

When a parity error occurs, the faulty data is consumed by the requester. VCOP may fetch and process bad data from IBUF or WBUF. ARP32 may try to execute a faulty instruction from the program cache, or read a faulty pointer from DMEM. To avoid these problems, the subsystem parity error interrupt scheme notifies ARP32 or the host of any parity errors (see Section 6.1.3.9.1*, EVE Interrupt Sources – Memory Switch and Parity Error Interrupts*).

To minimize corruption of the system, and to let the system host query the error detection registers, a user selectable parity error disconnect scheme is provided. This mechanism lets the user disconnect the ARP32 from the neighboring EVE system, and/or disconnect the OCP initiator port from the device interconnect. Each of these disconnects can be manually initiated by setting the EVE_DISC_CONFIG register, or EVE_DISC_CONFIG can be initiated when a specific parity error is detected based on requestor and memory. The disconnect sequence is enabled in the EVE_ED_OCPI_DISC_EN and EVE_ED_ARP32_DISC_EN registers. Figure 6-7 shows an overview of the disconnection request.

Figure 6-7. Parity Error ARP32/OCP Disconnect



6.1.3.3.7 VCOP System Error Halt Conditions

VCOP has several internal error conditions that force the core to halt execution and freeze its internal state so that ARP32 or debugger can inspect the state of the VCOP. The vector core also provides an error interrupt that causes VCOP to halt based on detection of EVE-level errors. These are mapped to the VCOP memory switch error source status (EVE_MSW_ERR[1] VERR) and VCOP parity error source status (EVE_WBUF_ED_STAT[1] VERR or EVE_IBUF_ED_STAT[1] VERR). By default the halt signals are enabled. Halt signal generation can be modified by writing to the EVE_VCOP_HALT_CONFIG[1] MSW_EN and EVE_VCOP_HALT_CONFIG[0] ED_EN bits.

In addition, the EVE_VCOP_HALT_CONFIG[2] FORCE_ABORT bit is used to directly force the vector core to halt based on any unforseen error condition. The error is detected by ARP32 through a watchdog timer at which point ARP32 sets the FORCE_ABORT signal to halt or abort the current loop occurring in the VCOP core.

6.1.3.4 Program Cache Architecture

The main features of ARP32 program cache are:

- 32-KB direct mapped cache
- 32-B line size
- Hit throughput: 32-bits per cycle, 100% throughput
- Read-only (from ARP32 perspective), all accesses are cacheable
- Read and write by debug OCP port
- Software directed preload
- Demand based prefetch
- Software directed invalidate
  - Global
  - Range based
- Debug support:
  - Single register one-line invalidate (for breakpoint support)
  - Tags memory mapped
  - SRAM memory mapped

  - Cache miss signaling to SCTM
- Hamming code based error detection: 10-bits per 256-bit location + tag + address

Program cache is a direct-mapped cache, therefore software controls the linking of the functions in external memory to minimize conflict misses. The program cache supports error detection encoding and can correct two errors per 256 bits of program code.

6.1.3.4.1 Basic Operation

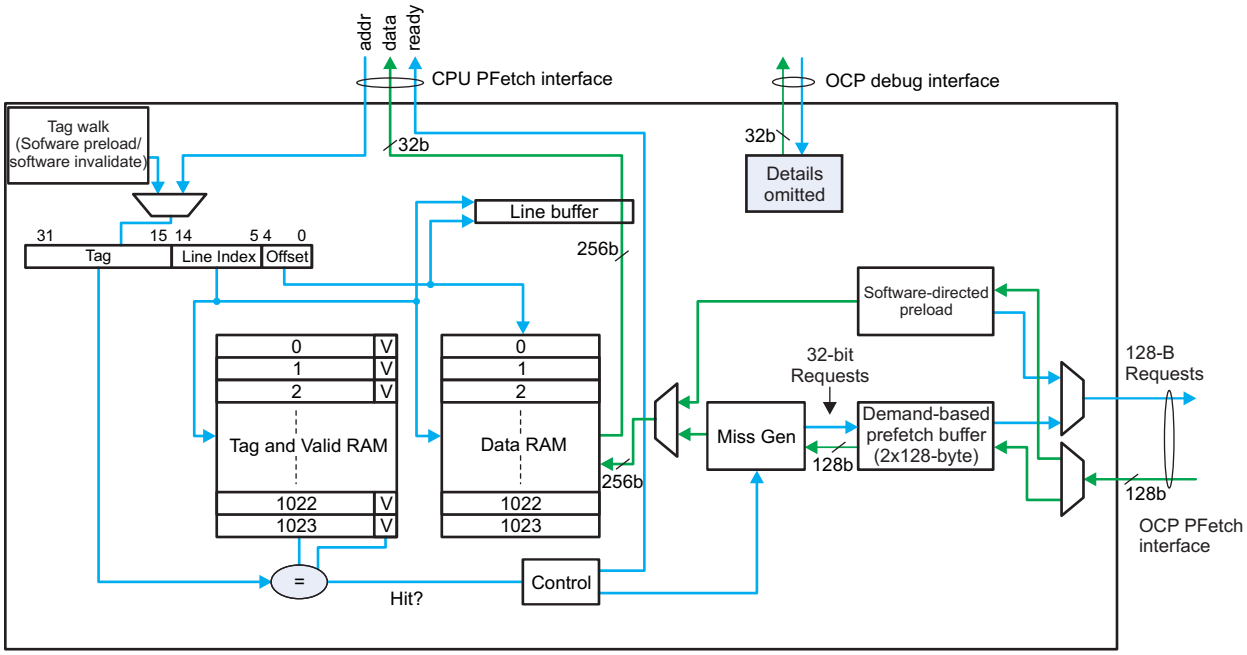
The program cache is always enabled and treats all ARP32 accesses as cacheable.

For cache hits the interface between ARP32 and program cache can handle back-to-back requests with 0 cycle latency to provide full throughput and program execution for the ARP32. For cache misses, the cache controller stalls ARP32, and internally submits a 32-bit OCP request (through the demand-based prefetch (DBP) block). After the 32-bit cacheline returns, the program cache returns the appropriate data to the ARP32 and starts the CPU.

For CPU-generated miss requests (not software-direct preload), the DBP block snoops the internal request and either services the request directly or schedules a miss+prefetch operation to the system.

Figure 6-8 shows the program cache block diagram.

Figure 6-8. EVE Program Cache Architecture



6.1.3.4.2 Line Buffer

The program cache includes a software and performance transparent 256-bit line buffer. The line buffer minimizes accesses to the underlying cache/SRAM in the case of back-to-back hits to the same line; doing so minimizes power consumption. The line buffer is allocated on any program cache access to a new line. As long as subsequent program fetches are to the same line, the data returns to ARP32 directly from the line buffer.

6.1.3.4.3 Software Direct Preload

Software Direct Preload SDP allows preloading of a range of system memory into the program cache. Software sets the preload base address register (EVE_PC_PBAR), along with a preload byte counter (EVE_PC_PBC) register (maximum = 0x8000). Hardware will issue a cache line fill request to the system for the associated line for the programmed address range. When the data is returned for each line, it is written into the cache (32-bits per write, to minimize program stalls) and the tag is marked as valid for the new address. This process is repeated for every cache line in the requested range. When the operation completes, the EVE_PC_PBC register is set to 0.

NOTE: Software must verify that the previous operation is completed before issuing another SDP operation.

Functional ARP32 requests are given a lower priority relative to software preload requests. The normal operation of the software preload results in many idle cycles while preload requests are in flight through the system. During that time ARP32 CPU accesses that result in cache hits proceed unhindered. If a CPU cache miss occurs while SDP operation is active, the two operations proceed in parallel.

The start address can be any arbitrary byte address, whereas the preload operation occurs on cache lines (32-bit aligned).Thus, the range preloaded is effectively rounded down to the nearest cache-line address relative to the start address, and rounded up to include the entire cache line relative to the end address.

Preload requests operate in parallel with the DBP block and do not use the buffering provided by the DBP block. Instead, the returned data for software direct preload is written directly to the program cache as highest priority, thus minimizing the required buffering and minimizing the impact on the system.

6.1.3.4.4 User Coherence Operation

The memory spaces cached in ARP32 program cache are not always coherent with the external memory. For example, if the copy of the code in external memory is modified, it is possible that the copy of the code that is cached in L1P is out of date with respect to the corresponding location contents in system memory. This may happen, for example, if software is implemented using code overlays. The application controlling such paging should take the appropriate steps to invalidate a cache that is rendered incoherent.

In addition, the ARP32 breakpoint methodology involves emulation and debug software inserting a breakpoint instruction in the program image in system memory (by replacing a valid instruction). Thus, there is a mechanism to update the cache to reflect the newly inserted breakpoint instruction. This is accomplished with the same software or hardware mechanism.

To allow for such software-directed invalidate, the cache controller supports three methods to invalidate the content of the cache:

- Global invalidate
- Range-based invalidate
- Single-address invalidate

6.1.3.4.4.1 Global Invalidate

The program cache clear register must be set (EVE_PC_INV[0] I = 0x1) to invalidate all the lines in the program cache. This is done by resetting all valid bits. Because EVE implements the valid bits with memory, the operation takes approximately a number of cycles equal to the number of tags. During invalidate-all operation, any CPU fetches are stalled until the invalidate-all operation completes, resulting in subsequent misses in the program cache.

6.1.3.4.4.2 Range-Based Invalidate

The program cache controller supports a programmable invalidate mechanism, with which the starting address and the number of words to invalidate can be specified to start an invalidation sequence. The programming model for the invalidation mechanism consist of two memory mapped registers: the start address register (EVE_PC_IBAR[31:0] ADDR) that holds the start address and the byte count register (EVE_PC_IBC[15:0] BC) that holds the number of bytes to be invalidated.

The invalidate operation begins immediately with the writing into EVE_PC_IBC[15:0] BC (maximum = 0x8000). The application first sets the EVE_PC_IBAR[31:0] ADDR bit field and then the byte count register to ensure correct operation. The operation involves cycling through addresses starting from the value of EVE_PC_IBAR[31:0] ADDR in increments of the cache line size, doing tag lookups to check if the line exists in the cache. If the line exists in cache, the corresponding valid bit is reset. The. EVE_PC_IBC[15:0] BC bit field is reset to 0 when the invalidate completes. As the byte count register is readable, a check for 0 provides a synchronization event for the application to execute from the region being invalidated or before issuing another range-based invalidate.

NOTE: EVE_PC_IBAR[31:0] ADDR can be any arbitrary byte address; whereas the invalidate operation occurs on cache-lines (that is, 32-bit aligned). Thus the range invalidated is effectively rounded down to the nearest cache-line address relative to the start address, and rounded up to include the entire cache line relative to the end address.

As in the global invalidate case, any CPU program fetches issued during the invalidate-range operation are stalled until the invalidation completes.

6.1.3.4.4.3 Single-Address Invalidate – For Breakpoint Operation

Because ARP32 supports breakpoint insertion, a single register version of the range-based invalidate is provided through the invalidate single address register (EVE_PC_ISAR). This is distinct from the previously defined range based invalidate, though the underlying hardware mechanism is the same. It is legal for the range-based invalidate and single-address invalidate to occur at the same time. Both commands must complete, though the order of completion is not explicitly defined.

Emulation software replaces the appropriate instruction in system memory with the breakpoint instruction. To ensure that future ARP32 execution of that line of code is fetched from system memory (as opposed to directly serviced from the cache), emulation software writes the address to EVE_PC_ISAR[31:0] ADDR.

The invalidate operation begins immediately on writing into the EVE_PC_ISAR register. The hardware operation involves checking the tag for the specified address. If the line exists in the cache, the corresponding valid bit is reset. Upon completion, the EVE_PC_ISAR[31:0] ADDR bit field is set to 0, thus informing the emulation software driver that the command is complete.

Operation is undefined if the application of emulation software tries to write to this register before the previous command completes.

6.1.3.4.5 Demand-Based Prefetch

The DBP block is used to improve performance for straight line cache misses, by snooping the program cache miss/software preload requests and issuing larger/pipelined burst requests to the system. The DBP block includes a 256-byte buffer that is logically composed of two 128-byte buffers.

When EVE is reset, the buffers are reset to an idle/empty state and DBP defaults to enabled (EVE_BUS_CONFIG[4] DBP_ENABLE = 0x1). When a new cache-line request is detected, the DBP initiates 128-byte burst to the system for the current 128-byte aligned range, and another 128-byte burst for the next 128-byte aligned range. DBP requests never cross the 128-byte boundary. Depending on the alignment of the new cache request, for initial requests with LSB = 0, 128-byte request is issued; for LSB = 0x20, a 96-byte burst is issued; for LSB = 0x40 a 64-byte burst is issued; and for LSB = 0x60, a 32-byte burst is issued. This last case does not allocate space in the prefetch buffer.

Any subsequent cache miss requests that match addresses for the previously prefetched data are returned directly from the prefetch buffer. When a cache request is received for address with LSB = 0x60 (last 32 bytes of a 128-byte aligned buffer), then that 128-byte buffer segment is marked invalid and the next sequential 128 bytes of data is prefetched. The prefetch buffer is always 256 bytes ahead of the program cache miss requests, and operates in ping-pong fashion with each of the 128-bytes buffer segments.

Example:

1. Buffer 0 holds addresses from 0x0 through 0x7F and buffer 1 holds addresses 0x80 through 0xFF. If the next miss request is to any location in buffer 1 (for example 0xA0), then buffer 0 is invalidated and prefetches range 0x100 through 0x17F.

2. Buffer 0 holds addresses from 0x0 through 0x7F and buffer 1 holds addresses 0x80 through 0xFF. If the next miss request is to the final quadrant of buffer 1 (0xE0), then return data from 0xE0 is returned. Both buffer 1 and buffer 0 are invalidated and addresses 0x100 and 0x180 are prefetched.

When a discontinuous cache-line request is received (that is, the request is not allocated [or in flight] in the prefetch buffer), then the entire prefetch buffer is invalidated, and the sequence begins from the start.

The prefetch buffer is invalidated when any user coherence operation is initiated.

DBP is disabled by clearing the EVE_BUS_CONFIG[4] DBP_ENABLE bit. When disabled, all cache miss requests bypass the DBP buffers and are issued to the system as 32-byte demand cache misses.

6.1.3.4.6 Debug Support

6.1.3.4.6.1 Read/Write Accessibility through OCP Debug Target Port

The OCP debug target port into the program cache provides full read and write of the cache SRAM contents, both for the data as well as tag contents, including parity/ecc and valid bits.

Writability is provided for memory self-test and parity error injection. ARP32 is in halted (for example in IDLE or debug halt) or reset state while write accesses are issued to program cache SRAM or tags. This is not intended as a code-preload feature. As such, if writes are performed to the tag value, 0 must be written to the valid bit before ARP32 can resume operation.

Relative to the EVE level base address, the data RAM resides at address 0x0 through 0x7FFF. The tag + ECC + valid bits reside at word-aligned addresses in the range of 0x8000 through 0x8FFF (base relative).

Table 6-. EVE Tag Visibility

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TAG																Reserved					HC										V	

Bit	Name	Description
31:15	TAG	Tag. When valid bit = 1, this bit represents the 17 MSBs of the cached address. Reads return tag; writes have no effect. Read access when LB = 0.
14:11	Reserved	Reserved. Read return 0s.
10:1	HC	Hamming code. Written by hardware based on computed Hamming code value calculated based on tag, address, and program cache data contents for this line. Read and compared by hardware to recalculated value when cache-line hit occurs. Reads return Hamming code; writes have no effect. Read accesses when LB = 0.
0	V	Valid bit. Set to 1 by hardware when tag + data line is valid/allocated. Cleared to 0 by hardware when tag + data line is invalid. Reads return valid bit; writes have no effect. Read access when LB = 0.

6.1.3.4.6.2 Breakpoint Support

Breakpoints are supported through the single-address invalidate operation summarized in Section 6.1.3.4.4.3, *Single-Address Invalidate – For Breakpoint Operation*. The emulation software replaces a functional instruction in system memory with the breakpoint opcode, and then invalidates that address in program cache through the range-based invalidate operation.

6.1.3.4.6.3 Cache Profiling

The program cache provides several output signals to the EVE SCTM block for the purpose of profiling and debugging cache performance issues.

Table 6-9 lists the cache profiling signals. The Type column of the table is either pulse or duration. The pulse signals are driven active one cycle for each occurrence. A sequence of consecutive active cycles represent multiple occurrences. It is possible that the duration signal stays active for multiple cycles on a given occurrence. These are stall signals that determine the total number of cycles in the stalled state. The SCTM Mode column describes the counter mode that is programmable in the SCTM module for a given counter. The SCTM Mode column is either event or duration. Event signals result in incrementing

the counter once per low-to-high transition on a given input. As such, the event mode is not overly useful for cache profiling. The duration mode measures the number of cycles during which a given signal is active. This is the mode used for most Cache profiling. In the case of counting pulses, this reports the total number of occurrences of a given state/signal (that is, cache hits). In the case of counting duration, this reports the total time in a given state.

Table 6-9. Cache Profiling Signal List

Name	Type	SCTM Mode	Description
cache_miss_count	Pulse	Duration	Per cache miss request occurrence
cache_hit_count	Pulse	Duration	Per cache hit request occurrence
line_buffer_hit_count	Pulse	Duration	Line buffer hit occurrence
cache_miss_stall	Duration	Duration (or event)	Driven high for the duration of a cache miss. Used by SCTM to measure the time the ARP32 is stalled due to a cache miss.
prefetch_compulsory_count	Pulse	Duration	Pulse on each or every compulsory DBP request to the system (that is, for the first 128-bit request after a buffer flush, or from IDLE state).
prefetch_lookahead_count	Pulse	Duration	Pulse on each or every lookahead DBP request to the system (that is, for the second 128-bit request after a flush).
prefetch_hit_count	Pulse	Duration	Pulse on each or every cache request that hits an already requested prefetch cache line, whether resident in the prefetch buffer or already requested but still in flight, and thus does not generate an additional prefetch request.
prefetch_line_count	Pulse	Duration	Pulse on every 32-bit cacheline that is requested/allocated into the prefetch buffer (including those that bypass the buffer due to starting at address 0x60). The difference between prefetch_line_count and prefetch_hit_count is equivalent to a prefetch_flush_line_count.
prefetch_discard_stall	Duration	Duration (or event)	Represents the amount of time that previously issued prefetch requests data is in flight (and is discarded) while a new compulsatory prefetch request begins.

6.1.3.4.7 Error Detection

For a description of the error detection mechanism, see Section 6.1.3.3.6, *Memory Error Detection*.

6.1.3.5 EDMA

NOTE: The device EVE integrated EDMA controller instance (EDMA3_0_CC0) which is functionally identical with the device EDMA controller instances (EDMA_TPCC, EDMA_TPTC0 and EDMA_TPTC1). The only difference is that the EDMA3_0_CC0 instance is located at different physical addresses.

EDMA is a DMA engine for transfers between the system memory (DDR and L3 SRAM) and EVE internal memories. For more detailed description of EDMA controller, see Chapter 11, *Enhanced DMA*, in .

Figure 6-9 is a high-level overview figure of the EDMA controller embedded in the EVE subsystem. The figure shows two major components:

- The Channel Controller (EDMACC or CC), which serves as the user interface of EDMA
- The Transfer Controllers (EDMATC or TC), which serve as the data transfer engine of EDMA

The ARP32 submits requests to the channel controller, which in turn submits transfer requests (TRs) to the appropriate transfer controller. Interrupts are posted in the EDMA CC upon transfer completion (if requested) and signaled to ARP32 core.

The EDMA is configured with two queues (in the CC) and two TC instances. The two TC instances offer maximum performance and concurrency for the case where simultaneous transfers occur between two different system-level endpoints, and two different EVE internal memories. For typical use cases, it is expected that transfers involving EMIF or DDR and L3 SRAM are mapped to two different TCs (TC0 for EMIF-related requests and TC1 for L3 SRAM related requests). For example, EMIF or DDR to IBUFL in parallel with IBUFH to L3 SRAM can be processed with maximum performance.

Table 6-10 shows the specific EDMA configuration for the available resources on EVE.

Figure 6-9. EDMA Block Diagram

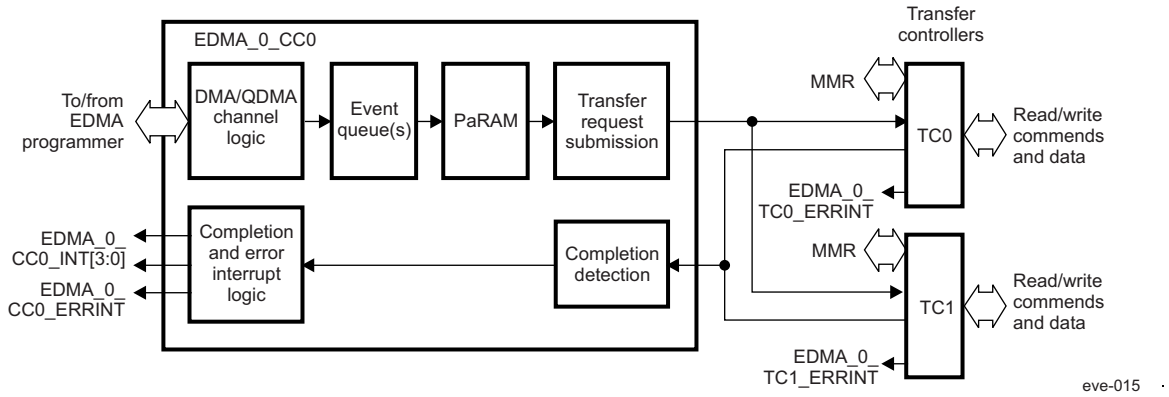


Table 6-10. EDMA Configuration

Name	Description	Configuration
NUM_DMACH	Number of EDMA channels	16
NUM_QDMACH	Number of QDMA channels	8
NUM_INTCH	Number of interrupt channels	16
NUM_PARAMENTRY	Number of PaRAM entries	128
NUM_EVQUE	Number of event queues	2
NUM_TC	Number of TPTC interfaces	2
MPEXIST	Memory protection exitsts	No
NUM_REGIONS	Number of MP and shadow regions	8
CHMAPEXIST	Channel mapping exists	Yes

6.1.3.5.1 DMA Channel Events

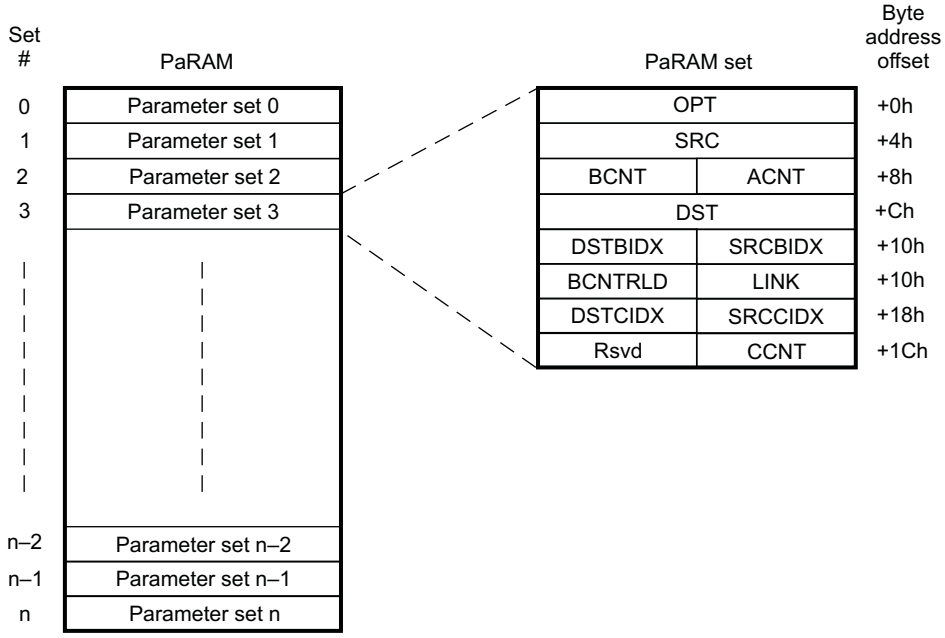
The EDMA input events/signals are driven by the evt_int[15:0] input signals. Those input interrupts are also connected to INTC1 of the EVE. A subset of these event inputs is mapped to ECM start events. Software must enable individual input interrupts as either events or ARP32 interrupts.

The EDMA channels can also be used under software control. In this case the DMA channels are triggered by writes to the event set register, or through internal chaining feature.

6.1.3.5.2 DMA Parameter Set

Figure 6-10 shows the layout of parameter RAM sets as a sequence of 128 entries which are adjacent in memory, and a detailed look into the contents of each parameter RAM.

Figure 6-10. Structure of Parameter RAM Sets and Contents



Note: *n* is the number of PaRAM sets supported in the EDMACC for a specific device.

Table 6-11. Fields in Parameter RAM

Acronym	Parameter
OPT	Channel Options
SRC	Channel Source Address
ACNT	Count for 1st Dimension
BCNT	Count for 2nd Dimension
DST	Channel Destination Address
SRCBIDX	Source BCNT Index
DSTBIDX	Destination BCNT Index
LINK	Link Address
BCNTRLD	BCNT Reload
SRCCIDX	Source CCNT Index
DSTCCIDX	Destination CCNT Index
CCNT	Count for 3rd Dimension
RSVD	Reserved

For a detailed explanation of the configuration of DMA PARAM set, see the *EVE Programmers Guide*, available through a TI representative.

6.1.3.5.3 Channel Controller

This section reviews various hardware resources of the channel controller that are programmed from the ARP32 controller of the EVE subsystem.

Parameter set: A parameter set is the program memory of the channel controller that describes the attributes of a transfer. Each parameter set describes one data transfer from a source address to a destination address. A parameter set is a group of eight contiguous 32-bit words and has an address that is aligned on a 256-bit boundary. There are 64 of these parameter sets, allowing the description of 64 transfers ahead of time. Individual parameter sets are used to describe isolated transfers, or they are linked together to describe a group of related transfers.

There are two types of DMA channels in the channel controller (CC) interface:

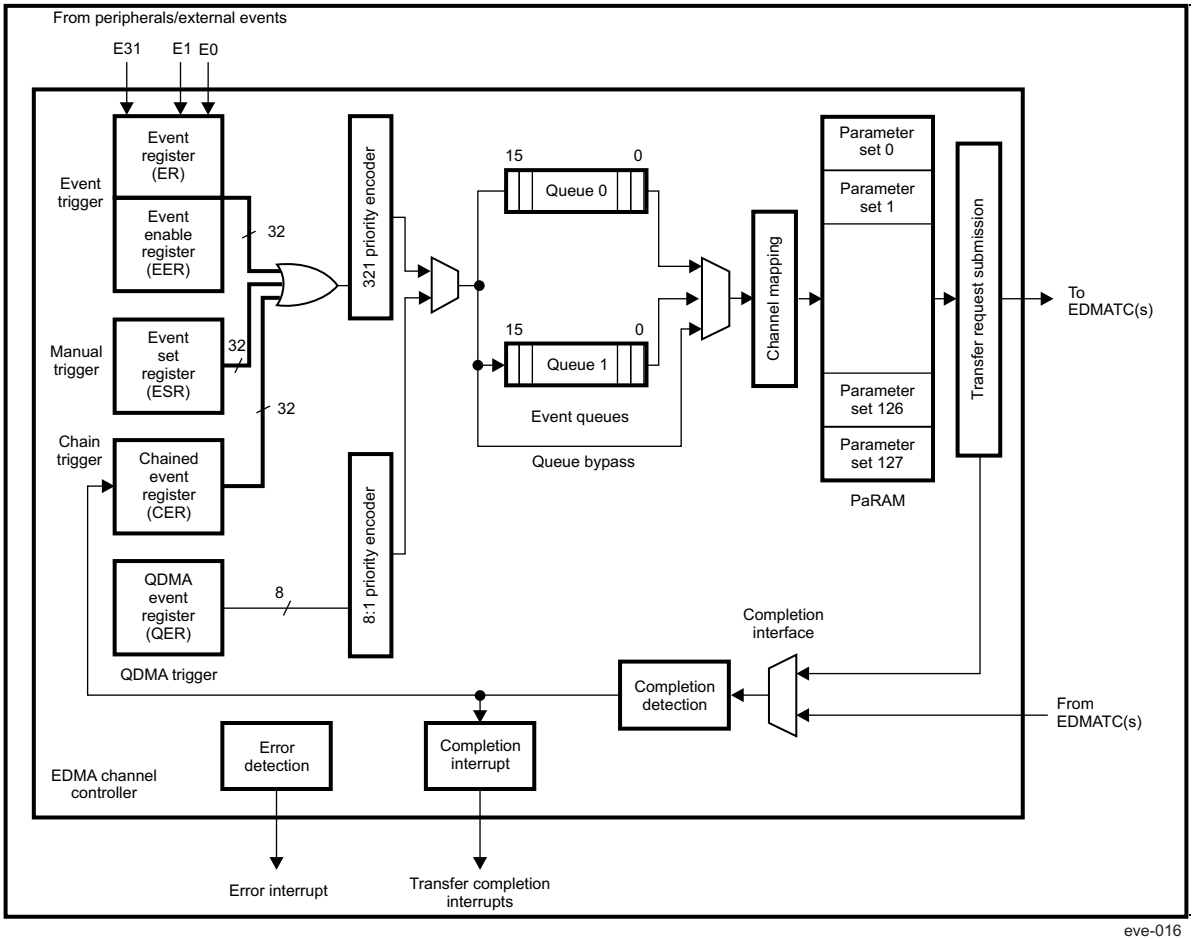
- EDMA channel: a data transfer channel that takes a parameter set as an input and requires that an event is triggered by a write to the corresponding event of the event set register (ESR), or by detecting assertion of a hardware event (Event Triggered), or by chain triggering. Software can designate a specific event as the trigger event by programming the mapping between the parameter set and the event in the DCHMAP register, and by mapping a DMA channel to a specific hardware event queue in the DMAQNUM register. EVE has 16 EDMA channels that must be assigned to 1 of 16 events of the ESR register, and mapped to 1 of 2 TC queues using DMAQNUM registers.
- QDMA channel: a data transfer channel that is similar to an EDMA channel, but is automatically triggered by the write to a specific word of parameter set (typically the eighth word). Eight QDMA channels are mapped to one of the two TC queues using the QDMAQNUM register.

An EDMA channel is used to describe a regular sequence of data transfers, in which the amount of data sent or received, and source and destination data address increments between successive transfers does not vary dynamically, but by regular amounts. A QDMA channel is used when it is needed to program in new values for the source and destination addresses. The amount of data to transfered varies from one transfer to another, and for other dynamic situations.

EVENT queue: data transfer events are different physical DMA channels (ESR writes for EDMA channels and parameter set trigger word writes for QDMA channels) and are detected simultaneously. The events are queued based on a fixed priority arbitration scheme; the EDMA channels are higher-priority events than the QDMA channels. Among the two groups of channels, the lowest-numbered channel is the highest priority.

Figure 6-11 shows the block diagram of the channel controller.

Figure 6-11. Channel Controller Block Diagram



6.1.3.5.4 EVE-Level Bus Width and Throughput

Table 6-12 shows the EVE-level effective bus width and the resulting peak bandwidth. In general, the effective bus width is the minimum of the bus width values between a given initiator and given target. For example, the DMEM port is effectively 4 bytes wide when accessed by ARP32 and is effectively 16 bytes wide when accessed by DMA. Similarly, VCOP memories are 4 bytes, 32 bytes and 16 bytes wide when accessed by ARP32, VCOP, and TCn, respectively.

Table 6-12. EVE-Level Effective Bus Width

		INITIATORS									
		ARP32		VCOP			TC0		TC1		OCP Target Port
		Program	Data	WBUF	IBUFL	IBUFH	Read	Write	Read	Write	
TARGETS	Program cache hit	4									
	Program cache OCP Target		4				4	4	4	4	4
	DMEM		4				16	16	16	16	16
	WBUF		4	32			16	16	16	16	16
	IBUFL		4		32		16	16	16	16	16
	IBUFH		4			32	16	16	16	16	16
	CC MMRs		4				4	4	4	4	4
	TC MMRs		4				4	4	4	4	4
	CFGMMRs		4				4	4	4	4	4
	OCP Init0	16	4				16	16			
	OCP Init1								16	16	

6.1.3.5.4.1 Concurrent Transfer Requirements

The two OCP ports are independent from one another to allow full bandwidth and nonblock requests, when EVE TC0 accesses one system-level memory at the same time as EVE TC1 accesses a different system-level memory and both internally access two different IBUF memories. The overall throughput is limited to only 128 bits per cycle in the case when the two TCs are accessing:

- The same system-level memory (EMIF, L3 SRAM)
- The same internal memory buffer

6.1.3.6 General-Purpose Inputs/Outputs

The general-purpose input signal state can be read directly through EVE_GPIN0 (for inputs 31:0) and EVE_GPIN1 (for inputs 63:32). These registers/GPIN signals are also mapped to the ARP32 INTC2 and INTC3 registers respectively.

The GPIs are used to convey state information (readable in the GPIN registers), or they are enabled as interrupts through the corresponding ARP32_INTn_IRQENABLE_SET register. A rising edge on the GP input is latched as a new interrupt in the corresponding ARP32_INTn_IRQSTATUS register.

The EVE subsystem also provides 64 GPOs. These outputs are manipulated by EVE_GPOUTm, EVE_GPOUTm_SET, EVE_GPOUTm_CLR, and EVE_GPOUTm_PULSE. The EVE_GPOUT0 register group controls eve_gpout[31:0] signals and the EVE_GPOUT1 register group controls eve_gpout[63:32] signals. The current state of the output is detected by reading any of the EVE_GPOUTm, EVE_GPOUTm_SET, EVE_GPOUTm_CLR, or EVE_GPOUTm_PULSE registers. The 64 GPOs are used to drive interrupts or state information to other EVEs in a multi-EVE system.

The GPOUTn register can be explicitly set with the corresponding write data word. Byte or halfword writes are permitted in this case (though support may depend on the specific initiator). A read-modify-write sequence can be required if different tasks own different bits within the GPOUT bank. In this case, access for the read-modify-write sequence must be assured. The EVE_GPOUTm_SET register unconditionally drives the corresponding output event to 1. GPIOm_CLR unconditionally drives the corresponding output event to 0. The EVE_GPOUTm_PULSE register unconditionally drives the output to one for four cycles, then deasserts the signal to 0. If the signal is 1, then the EVE_GPOUTm_PULSE simply deasserts the output signal. Software must ensure the output is low before using the EVE_GPOUTm_PULSE feature.

There are several GPOUT signals connected to arbitration/pressure bits (MFlag bits) in the device.

For EVE port 1 and EVE port 2 (EVE TC0 and EVE TC1) MFlag is driven by eve_gpout[63] and eve_gpout[62] respectively; eve_gpout[63] is connected to EMIF and eve_gpout[62] is connected to EMIF.

EVE signals are described in through .

EVE MFlag signals are not used inside L3_MAIN. Within the device main interconnect bandwidth regulators are used to control pressure/arbitration of EVE traffic.

6.1.3.7 CME Signaling

The credit management engine (CME) protocol is a credit-based data transfer scheme. A central agent asserts a START signal to a DMA engine or host processor. This signal indicates that a fixed-size data buffer is available to be read or written. Then the DMA engine or host processor is responsible for asserting a DONE signal back to the CME when the data buffer is read or written. The Central Agent uses this information to update its credit accounting (there is a consumer end and producer end for any buffer that is managed by CME). The EVE subsystem generically supports START and DONE signaling, and software must service the request accordingly.

CME START signal:

In a multi-EVE system the CME START signals are broadcast to the eve_evt_int[n:0] input signals of each EVE. These signals are mapped on both EDMA events or ARP32 interrupts. Software enables the appropriate event or input in the appropriate EVE, based on the overall system goals.

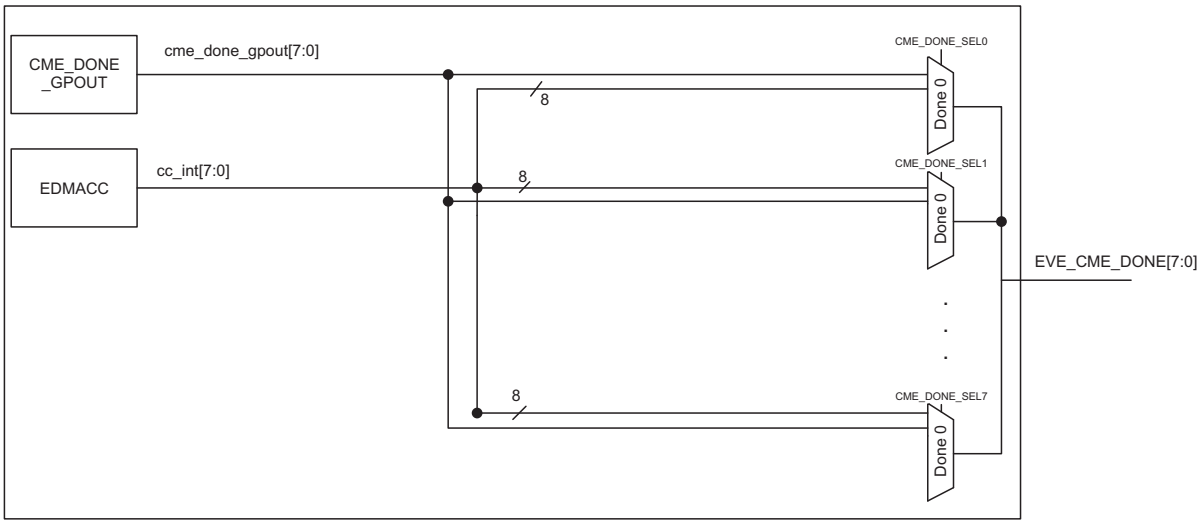
For a given EVE, a START signal must either be used to trigger an EDMA transfer directly for a known or agreed quantity of data, or it must trigger an ARP32 interrupt, which then manually sets up an EDMA transfer as necessary.

CME DONE signal:

EVE provides eight CME DONE signals that can be triggered from multiple internal conditions or sources. For any EVE in the system, the DONE output signals are driven either based on any one of the eight EDMA region interrupt outputs, or based on the EVE_CME_DONE_GPOUT register.

The EDMA region interrupt outputs are statically allocated in the device or host. As such, it is desirable to connect higher-numbered "done" bits to the CME to avoid resource allocation issues. The basic logic is essentially a mux select for each respective bit (see Figure 6-12). The selection is done through the EVE_CME_DONE_SEL[31:0] register.

Figure 6-12. CME Done Logic



CME registers in the EVE subsystem are:

- EVE_CME_DONE_GPOUT
- EVE_CME_DONE_GPOUT_SET
- EVE_CME_DONE_GPOUT_CLR
- EVE_CME_DONE_GPOUT_PULSE
- EVE_CME_DONE_SEL
- EVE_CME_DONE_EN

The DONE GPOUT registers are used to directly drive the state of the internal CME DONE output signals. The EVE_CME_DONE_SEL register sets whether the EVE output signals are driven from the EDMA region interrupts, the DONE GPOUT internal value, or a pass through from the eve_passthru signals.

NOTE: The device does not support hardware CME.

6.1.3.8 Memory Management Unit

There are two MMUs. Each of the two EDMA TCs is mapped to a different MMU. One of the MMUs is also shared with ARP32 program and data accesses. The ARP32 program cache and data accesses share an MMU to provide a convenient mechanism whereby ARP32 debug accesses have a single and consistent view of the system, which is equivalent to ARP32 software view. For the EDMA paths, the two MMUs provide maximum concurrency for each TC and its respective accesses to system memory. See Table 6-13 for MMU Configurations

Table 6-13. MMU Configuration

Name	Description	Configuration
MMU_Entries_Number	The number of entries in the TLB cache	32 entries
MMU_Command_Bypass	Bypass command register	2 cycle latency
MMU_Response_Bypass	Bypass response register(depends on clock speed versus technology)	2 cycle latency
MMU_LRU_Enable	LRU algorithm for TLB endry replacement	Disabled
MMU_RRB_Enable	Determines if Generic RRB module is instantiated (Zero only for DSPSS)	Present
LOG_DATA_WIDTH	data width configuration	128-bit data bus

For more information about MMU configuration see Chapter 15, *Memory Management Unit*

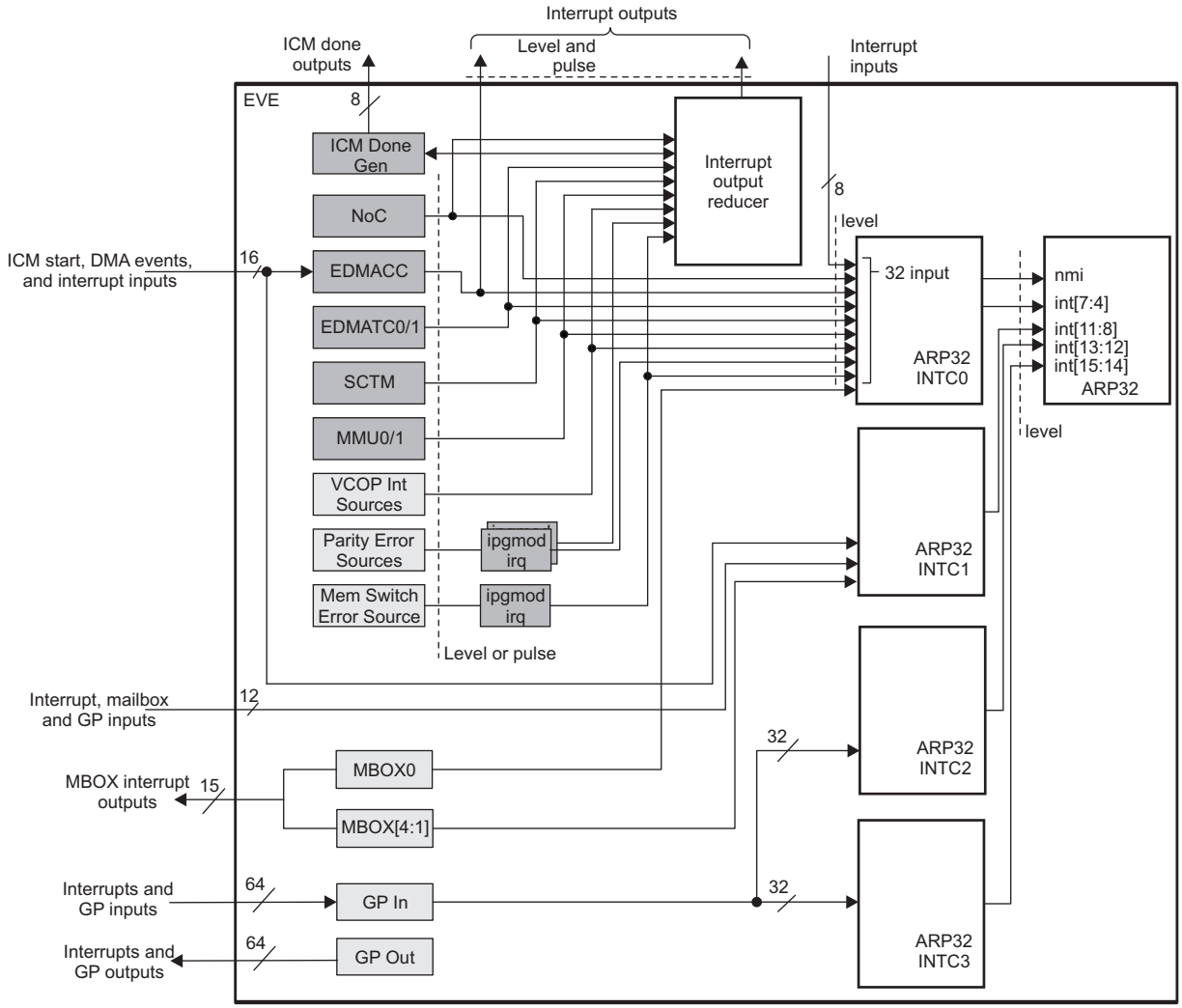
6.1.3.9 Interrupt Control

Each standard block (Interconnect, EDMA, SCTM) provides active high interrupts directly to the ARP32 INTC. In some cases, the submodule IP generated interrupts are mapped as outputs on the EVE boundary.

EVE interrupts (parity error interrupts and memory switch error interrupts) are summarized in Section 6.1.3.9.1, *EVE Interrupt Sources – Memory Switch and Parity Error Interrupts*. EVE accepts active-high input interrupts from the system and maps them to system-level mailbox interrupts.

Figure 6-13 is an overview of EVE interrupt topology.

Figure 6-13. EVE Interrupt Block Diagram



6.1.3.9.1 EVE Interrupt Sources – Memory Switch and Parity Error Interrupts

Figure 6-14 illustrates the mapping of EVE memory switch errors to a single interrupt output (EVE_MSW_ERR_INT). Figure 6-15 shows the mapping of EVE parity and error detect error interrupt outputs (EVE_ED_LCL_IRQ and EVE_ED_OUT_IRQ).

The errors are captured in the appropriate functional error bit field along with the additional error details (such as ConnID and specific address). Error status interrupts are readable in the EVE_MSW_ERR_IRQSTATUS_RAW registers. Software enables (or disables) interrupt generation for that source by writing to the appropriate bit in the interrupt enable set or clear register. When any enabled EVE_MSW_ERR_IRQSTATUS_RAW bit is set, then an interrupt is generated to the ARP32 INTC on the corresponding IRQ output (enabling of IRQSTATUS_RAW event is done through EVE_MSW_ERR_IRQENABLE_SET[3:0]ENABLE and EVE_MSW_ERR_IRQENABLE_CLR[3:0] ENABLE registers).

When the interrupt is served, software clears the interrupt status register fields and the source error register (through the EVE_MSW_ERR register or the EVE_<MEM>_ED register). New interrupt is latched into the EVE_MSW_ERR_IRQSTATUS register only after the register is cleared and new error occurs on the corresponding source error bit or signal. The appropriate bit in the EVE_MSW_ERR_IRQSTATUS_RAW register must be set on a low-to-high transition of the corresponding source error bit or signal. The EVE_MSW_ERR_IRQSTATUS register is routed to EVE-level interrupt output (active-high level and pulse signals are output) and as an active-high level interrupt to the ARP32 INTC.

Two interrupts are provided for EVE parity or error detect interrupt. The first is the local version (EVE_ED_LCL_IRQ) provided as an active-high level interrupt to the ARP32 INTC. The second is an output version (EVE_ED_OUT_IRQ) provided as an output on the EVE boundary as both active-high level and pulse version. This lets software selectively control which parity error sources are serviced by the ARP32, and which parity error sources are serviced by the external host. Typically, ARP32 services parity error interrupts generated by VCOP, EDMA, or system accesses and the system host services parity error interrupts generated by ARP32 accesses. The following registers control the detection and servicing of those interrupts (EVE_ED_LCL_IRQ and EVE_ED_OUT_IRQ):

- EVE_ED_LCL_IRQSTATUS_RAW
- EVE_ED_LCL_IRQSTATUS
- EVE_ED_LCL_IRQENABLE_SET
- EVE_ED_LCL_IRQENABLE_CLR
- EVE_ED_OUT_IRQSTATUS_RAW
- EVE_ED_OUT_IRQSTATUS
- EVE_ED_OUT_IRQENABLE_SET
- EVE_ED_OUT_IRQENABLE_CLR

The mechanism for detecting LCL and OUT interrupts is the same as the sequence described for the memory switch error interrupts. Any interrupt condition is captured in the IRQSTATUS_RAW registers (EVE_ED_LCL_IRQSTATUS_RAW[31:0] EVENT and EVE_ED_OUT_IRQSTATUS_RAW[32:0]EVENT), but only after enabling the appropriate IRQSTATUS_RAW bits (through EVE_ED_LCL_IRQENABLE_SET[3:0] ENABLE and EVE_ED_LCL_IRQENABLE_SET[3:0] ENABLE bit fields) the event is outputted for further processing (to the EVE parity or error detect error interrupts).

See Table 6-14,Figure 6-15, and Figure 6-15 for more information.

Figure 6-14. EVE Memory Switch Error Interrupt

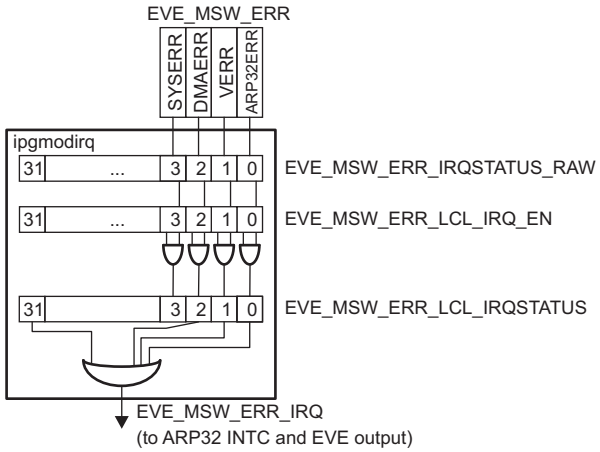


Table 6-14. EVE_MSW_ERR Register Interrupt Mapping

Bit Position	Source	Error
0	ARP32ERR	ARP32 initiated buffer ownership error
1	VERR	VCOP initiated buffer ownership error
2	DMAERR	EDMA initiated buffer ownership error
3	SYSERR	System-initiated ownership error

Figure 6-15. EVE Parity/Error Detect Interrupt

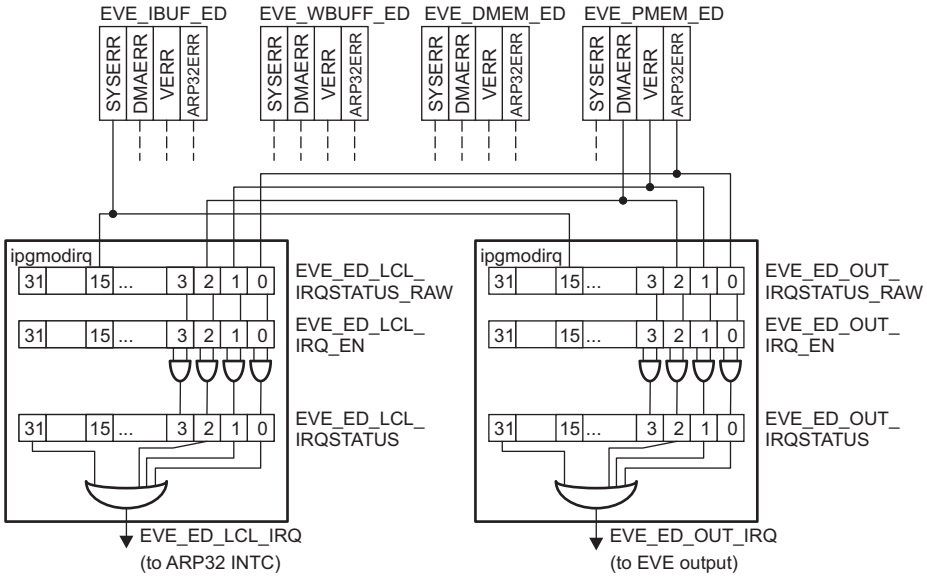


Table 6-15. EVE Local and Output Error Detect Error Interrupt Mapping

Bit Position	Source	Error
0	EVE_PMEM_ED_STAT[0] ARP32ERR	ARP32-initiated parity error
1	Reserved	Reserved
2	Reserved	Reserved
3	Reserved	Reserved
4	EVE_DMEM_ED_STAT[0] ARP32ERR	ARP32-initiated parity error
5	Reserved	Reserved

Table 6-15. EVE Local and Output Error Detect Error Interrupt Mapping (continued)

Bit Position	Source	Error
6	EVE_DMEM_ED_STAT[2] DMAERR	EDMA-initiated parity error
7	EVE_DMEM_ED_STAT[3] SYSERR	System-initiated parity error
8	EVE_WBUF_ED_STAT[0] ARP32ERR	ARP32-initiated parity error
9	EVE_WBUF_ED_STAT[1] VERR	VCOP-initiated parity error
10	EVE_WBUF_ED_STAT[2] DMAERR	EDMA-initiated parity error
11	EVE_WBUF_ED_STAT[3] SYSERR	System-initiated parity error
12	EVE_IBUF_ED_STAT[0] ARP32ERR	ARP32-initiated parity error
13	EVE_IBUF_ED_STAT[1] VERR	VCOP-initiated parity error
14	EVE_IBUF_ED_STAT[2] DMAERR	EDMA-initiated parity error
15	EVE_IBUF_ED_STAT[3] SYSERR	System-initiated parity error

6.1.3.9.2 ARP32 INTC

The EVE subsystem instantiates four INTC subblocks. Each ARP32 INTC supports up to 32 active-high level interrupt inputs, and outputs up to five active-high level interrupt outputs. NTC0 maps to NMI and INT[7:4]; INTC1 maps to INT[11:8]; INTC2 maps to INT[13:12]; and INTC3 maps to INT[15:14].

The ARP32_INTn_IRQSTATUS_RAW or ARP32_NMI_IRQSTATUS_RAW register is set when the input signal transitions from a low-to-high state. Software clears the ARP32_INTn_IRQSTATUS_RAW or ARP32_NMI_IRQSTATUS_RAW register by writing 1 into the appropriate bit position. To latch a new level interrupt, software must first clear the source so that a new low-to-high transition is generated on the INTC input. This is compatible with both level and source interrupt sources.

Upon clearing any bit in a given ARP32_[INTn(j)|NMI]_IRQSTATUS_RAW register, if any enabled interrupts are still set, that is ARP32_[INTn(j)|NMI] output signal pulse is low for two clock cycles and is about to transition back to high state. This results in resetting the ARP32 IFR and thus triggers a new ISR, protecting against race conditions. Any new interrupts must occur after the initial read of the ARP32_INTn_IRQSTATUS or ARP32_NMI_IRQSTATUS state.

Upon entering an ISR, software must first read from the EVE_INTk_OUT_IRQSTATUS register, which may show multiple enabled interrupts pending. Software clears the state of those enabled interrupts that are set (by writing 1 to EVE_INTk_OUT_IRQENABLE_CLR and IRQSTATUS bit) and that are dispatched for servicing. Software determines which interrupt source to service first through whatever scheme is convenient. Upon servicing the selected interrupt, software clears the original source interrupt status.

Figure 6-16 shows the INTC for ARP32.

Figure 6-16. EVE INTC for ARP32

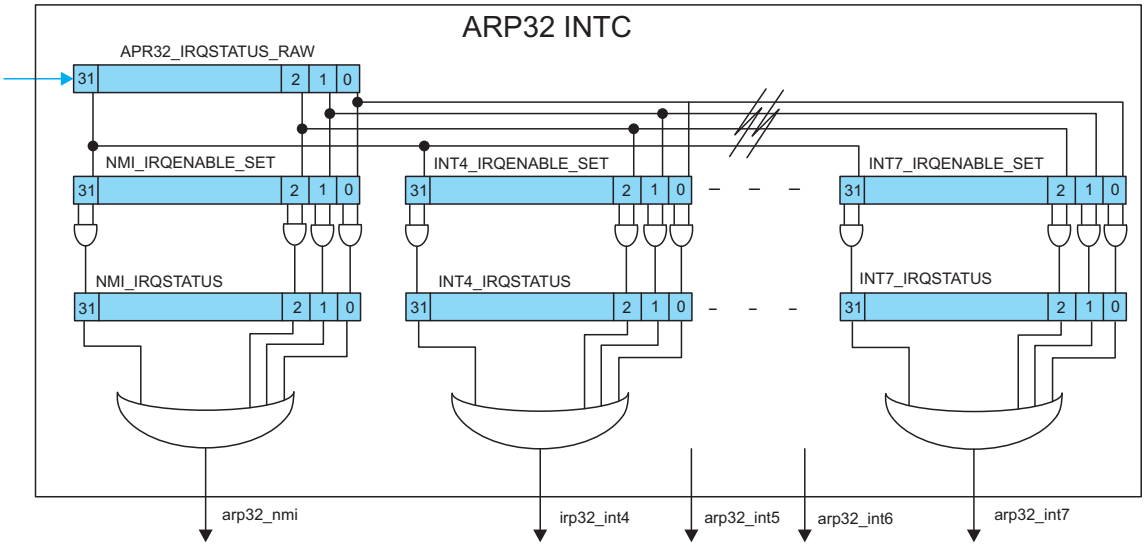


Table 6-16, Table 6-17, Table 6-18, and Table 6-19 summarize the ARP32 interrupt mapping.

Table 6-16. EVE ARP32 Interrupt Event Mapping Group0/INTC0

Interrupt	Name	Description	Source
0	eve_int00 (1)	Mapping to Mailbox 0	EVE input
1	eve_int01 (1)	Mapping to Mailbox 1	EVE input
2	eve_int02 (1)	Mapping to Mailbox 2	EVE input
3	eve_int03 (1)	Reserved	EVE input
4	eve_int04 (1)	Reserved	EVE input
5	eve_int05 (1)	Reserved	EVE input
6	eve_int06 (1)	Reserved	EVE input
7	eve_int07 (1)	Reserved	EVE input
8	tpcc_intg	EDMA CC global interrupt	EDMA CC
9	tpcc_int0	EDMA CC region 0 interrupt	EDMA CC
10	tpcc_int1	EDMA CC region 1 interrupt	EDMA CC
11	tpcc_int2	EDMA CC region 2 interrupt	EDMA CC
12	tpcc_int3	EDMA CC region 3 interrupt	EDMA CC
13	tpcc_int4	EDMA CC region 4 interrupt	EDMA CC
14	tpcc_int5	EDMA CC region 5 interrupt	EDMA CC
15	tpcc_int6	EDMA CC region 6 interrupt	EDMA CC
16	tpcc_int7	EDMA CC region 7 interrupt	EDMA CC
17	SCTM_TIMEVNTINT0	SCTM timer interrupt 0	SCTM
18	SCTM_TIMEVNTINT1	SCTM timer interrupt 1	SCTM
19	vcop_done (2)	VCOP done	VCOP
20	vcop_err_intn	VCOP error	VCOP
21	mmu0_int	MMU0 interrupt	MMU0

(1) eve_int[7:0] are available only to the ARP32 interrupt controller. They are unavailable and reserved to the output interrupt reducer.

(2) Vcop_done will deassert when new instructions are issued. Using the interrupt service routine, the interrupt is cleared in ARP32 INTC. Clearing of the actual source is with the functional mechanism of sending new instructions.

Table 6-16. EVE ARP32 Interrupt Event Mapping Group0/INTC0 (continued)

Interrupt	Name	Description	Source
22	mmu1_int	MMU1 interrupt	MMU1
23	tpcc_errint	EDMA CC error interrupt	EDMA CC
24	tptc_errint0	EDMA TC0 error interrupt	EDMA TC0
25	tptc_errint1	EDMA TC1 error interrupt	EDMA TC1
26	noc_errint	Interconnect error interrupt	Interconnect
27	EVE_MSW_ERR_INT	Buffer error interrupt	EVE top
28	EVE_ED_LCL_ERR_INT or EVE_ED_OUT_ERR_INT (3)	Parity error interrupt	EVE top
29	mailbox0_interrupt0	Mailbox 0 interrupt 0	Mailbox 0
30	mailbox1_interrupt0	Mailbox 1 interrupt 0	Mailbox 1
31	Reserved	Reserved	Reserved

(3) EVE_ED_LCL_ERR_INT (EVE_ED_LCL_IRQSTATUS_RAW, EVE_ED_LCL_IRQSTATUS) is used by the ARP32 INTC and EVE_ED_OUT_ERR_INT (EVE_ED_OUT_IRQSTATUS_RAW, EVE_ED_OUT_IRQSTATUS) is used for the output interrupts reducer.

Table 6-17. EVE ARP32 Interrupt Event Mapping Group1/INTC1

Interrupt	Name	INTC1 Mapping	Description	Source
0	eve_evt_int[0]	eve_intc1[00]	Require mapping to ICM_cstart0 signal or VIP interrupt	EVE input
1	eve_evt_int[1]	eve_intc1[01]	Require mapping to ICM_cstart1 signal or VIP interrupt	EVE input
2	eve_evt_int[2]	eve_intc1[02]	Require mapping to ICM_cstart2 signal or VIP interrupt	EVE input
3	eve_evt_int[3]	eve_intc1[03]	Require mapping to ICM_cstar3 signal or VIP interrupt	EVE input
4	eve_evt_int[4]	eve_intc1[04]	Require mapping to ICM_pstart0 signal or VIP interrupt	EVE input
5	eve_evt_int[5]	eve_intc1[05]	Require mapping to ICM_pstart1 signal or VIP interrupt	EVE input
6	eve_evt_int[6]	eve_intc1[06]	Require mapping to ICM_pstart2 signal or VIP interrupt	EVE input
7	eve_evt_int[7]	eve_intc1[07]	Require mapping to ICM_pstart3 signal or VIP interrupt	EVE input
8	eve_evt_int[8]	eve_intc1[08]	General purpose interrupt and EDMA event from EVE	EVE input
9	eve_evt_int[9]	eve_intc1[09]	Not used	Not used
10	eve_evt_int[10]	eve_intc1[10]	Not used	Not used
11	eve_evt_int[11]	eve_intc1[11]	Not used	Not used
12	eve_evt_int[12]	eve_intc1[12]	Not used	Not used
13	eve_evt_int[13]	eve_intc1[13]	Not used	Not used
14	eve_evt_int[14]	eve_intc1[14]	Not used	Not used
15	eve_evt_int[15]	eve_intc1[15]	Not used	Not used
16	eve_int1[0]	eve_intc1[16]	Require mapping to remote EVE Mailbox interrupt. Reserved.	EVE input
17	eve_int1[1]	eve_intc1[17]	Reserved	Reserved
18	eve_int1[2]	eve_intc1[18]	Reserved	Reserved
19	eve_int1[3]	eve_intc1[19]	Reserved	Reserved
20	eve_int1[4]	eve_intc1[20]	Not used	Not Used
21	eve_int1[5]	eve_intc1[21]	Not used	Not Used

Table 6-17. EVE ARP32 Interrupt Event Mapping Group1/INTC1 (continued)

Interrupt	Name	INTC1 Mapping	Description	Source
22	eve_int1[6]	eve_intc1[22]	Not used	Not Used
23	eve_int1[7]	eve_intc1[23]	Not used	Not Used
24	eve_int1[8]	eve_intc1[24]	General-purpose interrupt	EVE input
25	eve_int1[9]	eve_intc1[25]	General-purpose interrupt	EVE input
26	eve_int1[10]	eve_intc1[26]	General-purpose interrupt	EVE input
27	eve_int1[11]	eve_intc1[27]	General-purpose interrupt	EVE input
28	mailbox2_interrupt0	eve_intc1[28]	Mailbox 2 interrupt 0	Mailbox 2
29	Reserved	eve_intc1[29]	Reserved	Reserved
30	Reserved	eve_intc[30]	Reserved	Reserved
31	Reserved	Reserved	Reserved	Reserved

Table 6-18. ARP32 Interrupt Mapping for Group2/INTC2

Interrupt	Name	Description	Source
0	eve_gpin[00]	GP input 00	GPI register
1	eve_gpin[01]	GP input 01	GPI register
2	eve_gpin[02]	GP input 02	GPI register
...	...	...	...
31	eve_gpin[31]	GP input 31	GPI register

Table 6-19. ARP32 Interrupt Mapping for Group3/INTC3

Interrupt	Name	Description	Source
0	eve_gpin[32]	GP input 32	GPI register
1	eve_gpin[33]	GP input 33	GPI register
2	eve_gpin[34]	GP input 34	GPI register
...	...	...	...
31	eve_gpin[63]	GP input 63	GPI register

6.1.3.9.3 Output Interrupt Reduction

EVE provides four output interrupts based on the same approach shown previously for the ARP32 INTC. The EDMA region interrupts, as well as the four output interrupts, are provided at the EVE boundary as shown in Figure 6-13.

For more information about the IRQ_CROSSBAR module, see Section 13.4.6.4, *IRQ_CROSSBAR Module Functional Description*, in Chapter 13, *Control Module*.

Table 6-16 shows the interrupt mapping for the output interrupt reducer.

6.1.3.9.4 End of Interrupt Mapping

End of interrupt (EOI) is a feature used for software handshake with pulsed interrupts.

EOI MMR functionality is provided for EVE-generated interrupts that are mapped as outputs to the system. This feature is necessary when the external system uses the pulsed version of a given interrupt. Because all interrupts inside EVE are level interrupts, the EOI feature is not required for those interrupts.

NOTE: The EVE subsystem does not condition interrupts for internal submodules (EDMA, interconnect, MMU, or SCTM). Those interrupts are mapped directly as outputs to the system and no EOI functionality is provided for them.

Table 6-20 lists the EOI mapping for the required interrupts.

Table 6-20. EVE EOI Mapping

EVE MMR Value	EVE Interrupt Overview
1	eve_int0_out
2	eve_int1_out
3	eve_int2_out
4	eve_int3_out

6.1.3.10 Interprocessor Communication

This kind of communication is managed through mailboxes. The mailbox function supports 2-way communication between a maximum of four users. This function relies on internal submodules, each supporting 1-way communication between one user referred to as the sender and another user referred to as the receiver. Software allocates the mailbox submodule to the communication between two users. EVE includes internal mailbox to support synchronization and messages passing between EVE-, ARP32-, and system-level hosts.

6.1.3.10.1 Mailbox Configuration

The mailbox submodule includes support for four users and, therefore, provides four interrupt outputs. MAILBOX_Interrupt0 is mapped to the EVE INTC and MAILBOX_Interrupt[3:1] are mapped to three external host processors.

Six submailboxes are provided, allowing bidirectional communication between the ARP32 and the three external hosts:

- MAILBOX_MESSAGE0: HOST0 to EVE
- MAILBOX_MESSAGE1: HOST1 to EVE
- MAILBOX_MESSAGE2: HOST2 to EVE
- MAILBOX_MESSAGE3: EVE to HOST0
- MAILBOX_MESSAGE4: EVE to HOST1
- MAILBOX_MESSAGE5: EVE to HOST2

The listed allocation in this section is just an example. For more information about the Mailbox module see Chapter 14, *Mailbox*.

6.1.3.10.1.1 Mailbox 0 – EVE to DSP1 and DSP2

Mailbox 0 is dedicated to bidirectional communication between local EVE and two DSPs (DSP1 & DSP2). In this mailbox, 6 out of 16, submailboxes are used, 3 for senders and 3 for receivers. Additional mailboxes are available, if needed, to separate message traffic based on type or content between each sender and receiver pair (see Table 6-21).

Table 6-21. EVE to DSP1 and DSP2 Mapping

		Receiver		
		EVE	DSP1	DSP2
Sender	EVE	-	En_M0	En_M0
	DSP1	En_M0	-	-
	DSP2	En_M0	-	-

6.1.3.10.1.2 Mailbox 1 – EVE to Other Hosts

Mailbox 1 is dedicated to bidirectional communication with other system-level hosts . The actual usage and mapping varies on the different devices.

In a multicore system, additional mailbox modules are included to support generic and lower-frequency, higher-latency communication. This mailbox is used by host that needs tightly coupled interaction with the local EVE.

This mailbox has a similar usage as mailbox 0. Six sub-mailboxes are used, three for senders and three for receivers (see Table 6-22).

Table 6-22. EVE to Other Hosts Mapping

		Receiver			
		EVE	Host X	Host Y	Host Z
Sender	EVE	-	En_M1	En_M1	En_M1
	Host X	En_M1	-	-	-
	Host Y	En_M1	-	-	-
	Host Z	En_M1	-	-	-

6.1.3.11 Powerdown

The EVE subsystem supports two methods of power-minimization and clock gating:

- Active mode power minimization
- Extended duration sleep

6.1.3.11.1 Extended Duration Sleep

Extended duration sleep is an explicitly requested powerdown mode handshaking with the device-level PRCM. As a result, the EVE input clocks are gated. The EVE subsystem is able to latch new interrupts and wake-up requests while in this clock gated state. Internal state is retained, including ARP32 registers, ARP32 data SRAM and program cache contents, VCOP state, EDMA registers, PaRAM, EVE level registers, and so forth.

Since the EVE subsystem has only one logical clock domain (even though divided clocks are used) the PRCM only gates clocks to the EVE subsystem after all of the appropriate handshakes (that is IDLE, STANDBY, and OCP Disconnect) are concluded and the EVE subsystem is in a completely IDLE, STANDBY, and DISCONNECTED state.

6.1.3.11.1.1 Sequence Overview

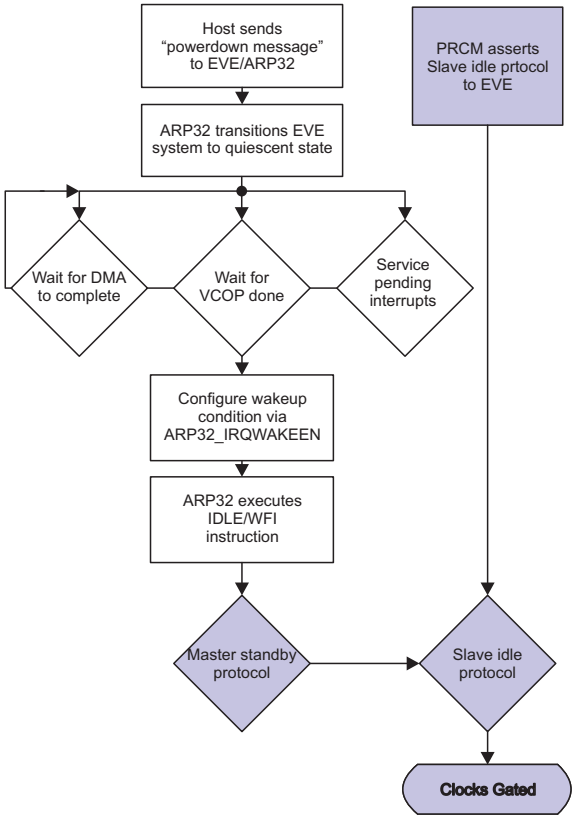
To enter extended duration sleep mode, the system host first signals the EVE subsystem (through systemlevel mailbox and interrupt) that it is entering this mode. In parallel, the PRCM asserts an SIdleReq request to the EVE subsystem. ARP32 performs any software bookkeeping necessary to transition the EVE subsystem to a quiescent state. This can include:

- Waiting for outstanding DMA transfers to complete
- Waiting for VCOP to finish the current task
- Servicing pending interrupts

If the EVE_SYSCONFIG[5:4] STANDBYMODE bit field is set accordingly, then the EVE hardware transitions to IDLE state and signals the desire to enter powerdown state to the device through the STANDBY and Idle protocols. After handshakes are completed internal EVE clocks are gated.

Figure 6-17 depicts the extended duration sleep software/hardware sequence.

Figure 6-17. Extended Duration Sleep Software/Hardware Sequence



6.1.3.11.1.2 Idle Protocol Overview

EVE IDLE protocol is associated with the EVE OCP target port. The configuration of the IDLE state is done by setting the EVE_SYSCONFIG[3:2] IDLEMODE bit field.

When the EVE subsystem is configured in Smart-Idle or in Smart-Idle-Wakeup, the following sequence must be performed upon an IDLE request from the PRCM:

- Acknowledge the receipt of the request by going into sleeptrans state.
- Wait for MSTANDBY state.
- Disconnect all OCP target ports using the OCP disconnect protocol.
- Gate EVE internal clocks.
- Acknowledge the IDLE request by going into IDLE state.

Externally generated OCP requests (through DMA OCP target bus) to EVE memory space are gated only after the MStandby state is entered. This is required because explicit messaging with the DMA may be necessary before ARP32 software can reach a state that allows entrance to MStandby.

6.1.3.11.1.3 Mstandby Protocol Overview

MSTANDBY protocol is initiated based on ARP32 executing idle instruction provided the other internal EVE masters are in idle state (assuming in Smart-Standby mode).

ARP32 software must wait until all internal operations are finished, before executing idle instruction.

The standby sequence follows:

1. ARP32 software waits for all activity to stop.
2. ARP32 issues IDLE instruction and all EVE submodules are in IDLE state.
3. Using the OCP disconnect protocol, deactivate all OCP initiator ports.

4. Assert the MStandby signal.

6.1.3.11.1.4 IDLE Wakeup

To facilitate auto-idle wakeup of the EVE subsystem, the wakeup capability of the idle protocol is used. A wake-up operation is executed when SYSCONFIG[3:2] IDLEMODE = 3h (SmartIdle-Wakeup).

In this mode, while in IDLE state, if an external input interrupt source that is enabled through the ARP32_IRQWAKEEN mask is active, then the SWakeup signal is asserted to the system. The device is expected to monitor the SWakeup request and enable clocks and exit the STANDBY and IDLE states. At this point, ARP32 can branch to the pending ISR.

The SWakeup signal is deasserted when all interrupt signals that are enabled in the IRQWAKEEN register are deasserted.

The logic controlling the assertion of SWakeup is completely asynchronous and does not rely on any input clock.

ARP32_IRQWAKEEN is not replicated per NMI, INTx. Any interrupt enabled in the ARP32_IRQWAKEEN mask must be enabled in one of the corresponding ARP32_INT_IRQENABLE registers.

6.1.3.12 Hardware-Assisted Software Self-Test – MISRs

To facilitate software self-test, MISRs are instantiated on address and data buses at key points in the system, such as ARP32 interfaces and the interconnect-WBUF interface. ARP32 is covered because it is the key control engine. WBUF coverage is provided as a convenient central destination that is used to indirectly provide coverage for a majority of EVE logic.

The MISRs monitor the address and data buses and calculate a signature based on the address or data pattern on a valid address or data phases. The signature registers reset to 0. A different speed value can be manually written through software to each signature register: MISR0_A, MISR0_D, MISR1_A, MISR1_D, and MISR2_Dk. Based on a known memory access data pattern, the MISR signature can be predicted or calculated and used as a reference for subsequent tests that occur at boot time or during runtime in a safety critical application.

The MISR calculation is a shift-register/XOR tree calculation using classical CRC algorithms.

The MISR calculation is initiated by setting the MISR_CTL[2:0] ENABLE bit field:

- bit 0: enable MISR 0 – ARP32 PMEM path
- bit 1: enable MISR 1 – ARP32 DMEM path
- bit 2: enable MISR2 – Interconnect WBUF path

The MISR_CLEAR[2:0] CLEAR register clears the MISR 0 and MISR 1 calculations for the corresponding path.

Table 6-23 describes the location and width of the various MISRs in the system and is shown as thick blue lines in the block diagram in Figure 6-1.

Table 6-23. MISR Mapping

		Address	Read Data	Write Data	Bus Width	MISR algorithm
MISR0_A	ARP32/PMEM	x			32	CRC-32
MISR0_D	ARP32/PMEM		x		32	CRC-32
MISR1_A	ARP32/DMEM	x			32	CRC-32
MISR1_D	ARP32/DMEM			x	32	CRC-32
MISR2_Dk (k = 0:3)	Interconnect/WBUF			x	128	CRC-32 (*4)

6.1.3.12.1 Mapping of MISRs to Different Width Buses

For WBUF data MISR, four instances of CRC-32 MISRs are instantiated in parallel to populate the width of the data bus. For ARP32 interfaces, a single instance of CRC-32 MISR is instantiated on each interface.

Each write data bus MISR (MISR1_D and MISR2_Dk) is activated based on the byte-enable pattern for the corresponding 32-bit word of memory. If any single byte-enable signal (of the 4 for the 4-bytes in the word) is active, then that MISR is updated. Byte-enables that are active use the corresponding data bus value. To avoid "don't care" signals from corrupting the MISR calculation, byte-enables that are inactive use a value of 0 on the input to the 32-bit MISR.

6.1.3.12.2 Detection of Valid Address and Data Cycles

For each MISR in the EVE subsystem, the update cycle occurs for functional accesses, once per read or write request or address phase for address based MISRs, and once per data phase for data-bus based MISRs. If a phase is extended due to "not ready" condition on the target, then the MISR is not updated on that cycle. MISR is updated only on the cycle where the REQUEST and READY signals are active.

6.1.3.12.3 Creating a Unique Signature – Software Self-Test Implications

To create tests that produce a unique signature, software must take care to ensure that the order of transactions (and resulting data/address phases) to a given MISR are consistent. Varying time between transactions does not affect the signature, because it is not updated on idle or not-ready cycles.

For cases that use a single initiator to access a given MISR, crossing asynchronous boundaries is acceptable. This is always true for ARP MISRs, but not necessarily for the WBUF MISRs.

For example if TC0 reads the OCP initiator bus and writes to WBUF MISR, the result is a consistent signature. And when TC0 reads EMIF address space (which is asynchronous) and writes to WBUF, while TC1 is also writing to WBUF results in different order of transactions.

For cases where all transactions/interactions are synchronous, using multiple initiators results in a consistent signature.

6.1.3.12.4 Multipass Tests Using WBUF MISR

The WBUF MISR is used to indirectly provide coverage for a majority of EVE logic. The EVE-level EDMA can be used to perform memcopy operation between a given source memory range and a WBUF memory address range and thus the WBUF MISR. The source buffer contains known values that are either the interim results of a VCOP self-test, or simply initialized to a known data pattern to provide interconnect level coverage. This is repeated for the necessary paths in the EVE subsystem to provide indirect coverage for those paths.

It is possible to reserve a portion of WBUF memory for the write data accesses during the self-test operation. The necessary reserved address space can be minimized by using EDMA constant mode transfers, which require a minimum of 32 bits of reserved space.

6.1.3.13 Error Recovery – ARP32 and OCP Disconnect

To prevent runaway code from corrupting the remainder of the system, and provide a clean reset/recovery mechanism, the EVE subsystem provides a mechanism to disconnect ARP32 from the remainder of the EVE subsystem, and to disconnect L3 initiator buses from the remainder of the device.

When ARP32 or OCP initiator buses are disconnected, the debugger can detect the EVE MMRs and memories through the interconnect target bus.

When the ARP32 and OCP buses are disconnected, a full reset and reboot cycle are issued in order to resume normal ARP32<->EVE operation. This is required in order to avoid any asynchronous timing paths due to asynchronous reset assertion to ARP32.

6.1.3.13.1 ARP32 Disconnect

ARP32 disconnect occurs when a specific parity error is detected and is enabled in the corresponding EVE_ED_ARP32_DISC_EN register. A disconnect request can also be issued by setting the EVE_DISC_CONFIG[0] ARP32_DISC bit to 1.

When the ARP32 core is in the process of disconnecting (waiting on in-flight request and response) the EVE_STAT[17:16] ARP32_DISC_STATUS bit field is set to 0x1, meaning "attempting to disconnect". When ARP32 is disconnected on both the program and data interfaces, then this bit field is set to 0x0 (disconnected).

NOTE: Software must wait for disconnected state before issuing a reset to the ARP32 core. This status ensures the neighboring system is not in a corrupted state

6.1.3.13.2 OCP Initiator Disconnect

The OCP initiator disconnect request occurs based on a specific parity error that is enabled in the corresponding EVE_ED_OCPI_DISC_EN register, or when the EVE_DISC_CONFIG[4] OCPI_DISC bit is set to 0x1.

When a disconnect request is issued both OCP initiator buses are disconnected. The disconnect logic forces the OCP initiator buses to stall at a clean boundary, such that EVE reset can be issued without violating the OCP protocol detected by the neighboring L3 interconnect. The disconnect logic also tries to fence any new requests, by allowing the current request handshake to complete, but gating any new OCP requests to the system. When the previously issued requests have received all responses, the interface is drained; at this point, the OCP disconnect protocol enters the M_OFF state.

When the OCP initiators are disconnecting, waiting on in-flight request/responses to complete, the EVE_STAT[21:20] OCPI_DISC_STATUS read value is 0x1 or "attempting to disconnect". Otherwise, when both initiators are disconnected, then the bit field contains the value 0, meaning that initiators are disconnected. In order to guarantee that the neighboring system is not left in a corrupted state, software must wait for disconnected state before issuing a reset to EVE

NOTE: Any requests into the OCP target bus must be stopped.

6.1.3.14 Lock and Unlock Feature

The EVE subsystem provides a Lock/Unlock mechanism that helps to prevent unintended access to the various control registers of the EVE control module, or EVE subcomponents. Ten lock and unlock registers are defined, where each register is used to lock and unlock access to a specific area of the memory map. Table 6-24 summarizes the logical groupings.

Table 6-24. Lock Register Mapping

Register Name	Mapping	Locked Registers
MMR_LOCK0	EVE memory configuration and standard EVE registers	EVE_REVISION, EVE_HWINFO, EVE_SYSCONFIG, EVE_STAT, EVE_DISC_CONFIG, EVE_BUS_CONFIG, EVE_VCOP_HALT_CONFIG, EVE_MMU_CONFIG
MMR_LOCK1	Memory switch registers and error registers	EVE_MEMMAP, EVE_MSW_CTL, EVE_MSW_ERR, EVE_MSW_ERRADDR
MMR_LOCK2	Program cache registers	EVE_PC_INV, EVE_PC_IBAR, EVE_PC_IBC, EVE_PC_ISAR, EVE_PC_ISAR_DONE, EVE_PC_PBAR, EVE_PC_PBC

Table 6-24. Lock Register Mapping (continued)

Register Name	Mapping	Locked Registers
MMR_LOCK3	Memory error detection registers	EVE_<Memory>_ED_CTL, EVE_<Memory>_ED_STAT, EVE_<Memory>_EDADDR, EVE_<Memory>_EDADDR_BO, EVE_ED_ARP32_DISC_EN, EVE_ED_OCPI_DISC_EN
MMR_LOCK4	Interrupt registers and GPout signals and CME signaling	EVE_MSW_ERR_X, EVE_ED_LCL_X, ARP32_NMI_X, ARP32_INTi_X (see (1), (2)), EVE_GPOUTm, EVE_GPOUTm_SET/CLR/PULSE, EVE_GPIN0, EVE_GPIN1, EVE_CME_DONE
MMR_LOCK5	MISR registers	See Section 6.1.5.
MMR_LOCK6	MMU, ARP32, VCOP	See appropriate chapter, MMU, ARP32 Reference Guide, VCOP Reference guide.
MMR_LOCK7	Debug	–
MMR_LOCK8	EDMA channel controller	–
MMR_LOCK9	EVE output interrupts	EVE_ED_OUT_IRQSTATUS_RAW, EVE_ED_OUT_IRQSTATUS, EVE_ED_OUT_IRQENABLE_SET, EVE_ED_OUT_IRQENABLE_CLR, EVE_INTk_OUT_IRQSTATUS_RAW, EVE_INTk_OUT_IRQSTATUS, EVE_INTk_OUT_IRQENABLE_SET, EVE_INTk_OUT_IRQENABLE_CLR (3)

(1) X = IRQSTATUS_RAW, IRQSTATUS, IRQENABLE_SET, or IRQENABLE_CLR
(2) i = 4 to 7 and 8 to 15
(3) k = 0 to 3

6.1.3.15 EVE Memory Map

For ARP32/EDMA view, the entire 32-bit address space is defined. Accesses to addresses less than 0x4000 0000 and greater than or equal to 0x4010 0000 are used to generate accesses to the external system. These accesses are managed through the MMU module. Accesses to addresses 0x4000 0000 to 0x400F FFFF that are generated directly by ARP32/EDMA are used to access the internal EVE address space. The MMU initiator is only able to access external-to-EVE address space, thus EVE logically remaps a virtual address (address generated by ARP32/EDMA below 0x4000 0000 and above 0x4010 0000) to any 32-bit physical address in the system, such that addresses mapped to 0x4000 0000 still occur. In other words, the EVE address space coexists with the same address space mapped for a different function at system memory through the MMU translation.

EDMA views a compressed/aliased version of the VCOP memories. This is dependent on the EVE_MEMMAP[4] LCL_EDMA_ALIAS bit.

VCOP address space is limited to the VCOP memories. The VCOP memory map depends on the setting of the EVE_MEMMAP[0] VCOP_ALIAS bit.

The system address space represents accesses to the external system through the OCP target bus. At the system level, this memory range is mapped to a fixed base address. Table 6-25 lists the address that represents the offset relative to that base address.

Table 6-25 shows the EVE subsystem memory map, with separate columns to show the ARP32/EDMA, VCOP, and SYS views.

Table 6-25. EVE Subsystem Memory Map

ARP32/EDMA View		VCOP View		SYS View		Size	Region	Function
Start Address	End Address	Start Address	End Address	Start Address	End Address			
0h	3FFF FFFFh					1024MB	MMU0/MMU1	System Memory through MMU

Table 6-25. EVE Subsystem Memory Map (continued)

ARP32/EDMA View		VCOP View		SYS View		Size	Region	Function
Start Address	End Address	Start Address	End Address	Start Address	End Address			
4000 0000h	4001 FFFFh			0	1 FFFFh	128KB	Reserved	–
4002 0000h	4002 7FFFh			2 0000h	2 7FFFh	32KB	DMEM	Data memory
4002 8000h	4003 FFFFh			2 8000h	3 FFFFh	96KB	Reserved	–
4004 0000h	4004 7FFFh	0	7FFFh	4 0000h	4 7FFFh	32KB	WBUF	VCOP working buffer
4004 8000h	4004 FFFFh	8000h	FFFFh	4 8000h	4 FFFFh	32KB	Reserved	–
4005 0000h	4005 3FFFh	1 0000h	1 3FFFh	5 0000h	5 3FFFh	16KB	IBUFLA (or IBUFLB)	Image buffer low copy A (or B for VCOP)
4005 4000h	4005 7FFFh	1 4000h	1 7FFFh	5 4000h	5 7FFFh	16KB	IBUFHA (or IBUFHB)	Image buffer high copy A (or B for VCOP)
4005 8000h	4006 FFFFh	1 8000h	2 FFFFh	5 8000h	6 FFFFh	96KB	Reserved	–
4007 0000h	4007 3FFFh	3 0000h	3 3FFFh	7 0000h	7 3FFFh	16KB	IBUFLB	Image buffer low copy B
4007 4000h	4007 7FFFh	3 4000h	3 7FFFh	7 4000h	7 7FFFh	16KB	IBUFHB	Image buffer high copy B
4007 8000h	4007 FFFFh	3 8000h	3 FFFFh	7 8000h	7 FFFFh	32KB	Reserved	–
4008 0000h	4008 0FFFh			8 0000h	8 0FFFh	4KB	SYSTEM	Interrupt, reset, clock, power, buffer switch
4008 1000h	4008 1FFFh			8 1000h	8 1FFFh	4KB	MMU0 CFG	MMU0 configuration and registers
4008 2000h	4008 2FFFh			8 2000h	8 2FFFh	4KB	MMU1 CFG	MMU1 configuration and registers
4008 3000h	4008 3FFFh			8 3000h	8 3FFFh	4KB	ARP32	ARP32 control, including debug
4008 4000h	4008 4FFFh			8 4000h	8 4FFFh	4KB	VCOPC	VCOP control, including debug
4008 5000h	4008 5FFFh			8 5000h	8 5FFFh	4KB	SCTM	Subsystem counter/timer module
4008 6000h	4008 6FFFh			8 6000h	8 6FFFh	4KB	EDMA_TC0	EDMA transfer controller 0
4008 7000h	4008 7FFFh			8 7000h	8 7FFFh	4KB	EDMA_TC1	EDMA transfer controller 1
4008 8000h	4008 8FFFh			8 8000h	8 8FFFh	4KB	SMSET CFG	SMSET configuration interface
4008 9000h	4008 9FFFh			8 9000h	8 9FFFh	4KB	SMSET MSG	SMSET messaging interface
4008 A000h	4008 AFFFh			8 A000h	8 AFFFh	4KB	NoC	Interconnect registers
4008 B000h	4008 BFFFh			8 B000h	8 BFFFh	4KB	MBOX	Mailbox
4008 C000h	4008 FFFFh			8 C000h	8 FFFFh	16KB	Reserved	–
4009 0000h	4009 7FFFh			9 0000h	9 7FFFh	32KB	P$ RAW	Program cache raw
4009 8000h	4009 FFFFh			9 8000h	9 FFFFh	32KB	P$ Tags	Program cache tag
400A 0000h	400A 7FFFh			A 0000h	A 7FFFh	32KB	EDMA_CC	EDMA channel controller
400A 8000h	400A FFFFh			A 8000h	A FFFFh	32KB	Reserved	–
400B 0000h	400B FFFFh			B 0000h	B FFFFh	64KB	Reserved	–
4010 0000h	FFFF FFFFh					3071MB	MMU0/MMU1	System memory through MMU

6.1.3.15.1 VCOP and Local EDMA: IBUF Memory Map Aliasing

To facilitate ping-pong buffer management, EVE allows the local EDMA and VCOP view of IBUFLA versus IBUFLB, and IBUFHA versus IBUFHB memories to be aliased into the same address ranges. ARP32 and System accesses (through OCP target port) always use the unique/256-KB address map for IBUF memory accesses.

Having a VCOP accesses and EVE_MEMMAP[0] VCOP_ALIAS = 0x1 limits VCOP data memory view to 128KB, with A and B sets aliased to the same address range, depending on the MMR ownership bits (EVE_MSW_CTL). When EVE_MEMMAP[0]VCOP_ALIAS = 0x0, VCOP uses the full 256 KB address. The EVE_MEMMAP[0]VCOP_ALIAS bit has no effect on how other initiators (ARP32, EDMA or SYS) see the VCOP memories. The VCOP view truth table is shown in Table 6-26.If IBUFLA/IBUFHA is set (while VCOP owned), then the memory switch points to the IBUFHA/IBUFLA memory and IBUFHB/IBUFLB is a "don't care". If VCOP accesses address that is shown in gray in Table 6-26, that is reported as an error.

Having local EDMA accesses, when EVE_MEMMAP[4] LCL_EDMA_ALIAS = 0x1, limits local EDMA data memory view to 128KB, with A and B sets aliased to the same address range, dependent on the MMR ownership bits (EVE_MSW_CTL). When EVE_MEMMAP[4] LCL_EDMA_ALIAS = 0x0, EDMA uses the full 256-KB address. The EVE_MEMMAP[4] LCL_EDMA_ALIAS bit has no effect on how other initiators detect the VCOP memories. Table 6-27 is the local EDMA truth table. If IBUFHA or IBUFLA is cleared (while EDMA/system owned), then the memory switch points to the IBUFHA or IBUFLA memory and IBUFHB/LB bit is a don't care bit. If ARP32, EDMA, or system accesses an address shown in gray, then that is reported as an error.

NOTE: The ALIAS and ownership register are modified when the system is initializated, or between major modes of operation when no memory transactions are ongoing. If the mode changes while accesses are in progress, then a given address may change from valid to reserved (or viceversa), or a given address can change from one memory to another.

NOTE: ARP32 and system accesses (including the system EDMA) always view the expanded 256-KB unique addresses for access to IBUF memories.

When a memory switch error is detected, the physical address is captured. IBUFLB address is 0x40007 0000 even if in aliased mode.

6.1.3.15.2 ARP32 Write Model – Avoiding Race Conditions

ARP32 uses posted writes exclusively. This implies that writes are "fire-and-forget" and that writes on a given path complete some variable number of cycles after being issued by ARP32. If ARP32 issues a write followed by either a read or write to the same endpoint, then those accesses complete in order. However, if ARP32 issues a write to a given endpoint target (for example MSW_CTL) and then issues a read or write to a different endpoint target (for example IBUF), then the IBUF access may occur before the MSW_CTL write. To assure that a write has landed, that ARP32 must issues a read to the same address range.

In case of memory switch ownership or aliasing modifications, software must ensure that the mode change is established. This is done by reading back the value of the ownership or alias registers.

Table 6-26. VCOP IBUF Aliasing Truth Table

EVE_MEMMAP: VCOP_ALIAS Bit	EVE_MSW_CTL Bit				VCOP Address			
	IBUFLA	IBUFHA	IBUFLB	IBUFHB	10000h	14000h	30000h	34000h
0	0	0	0	0				
0	0	0	0	1				IBUFHB
0	0	0	1	0			IBUFLB	
0	0	0	1	1			IBUFLB	IBUFHB
0	0	1	0	0		IBUFHA		
0	0	1	0	1		IBUFHA		IBUFHB
0	0	1	1	0		IBUFHA	IBUFLB	
0	0	1	1	1		IBUFHA	IBUFLB	IBUFHB
0	1	0	0	0	IBUFLA			
0	1	0	0	1	IBUFLA			IBUFHB
0	1	0	1	0	IBUFLA		IBUFLB	
0	1	0	1	1	IBUFLA		IBUFLB	IBUFHB
0	1	1	0	0	IBUFLA	IBUFHA		
0	1	1	0	1	IBUFLA	IBUFHA		IBUFHB
0	1	1	1	0	IBUFLA	IBUFHA	IBUFLB	
0	1	1	1	1	IBUFLA	IBUFHA	IBUFLB	IBUFHB
1	0	0	0	0			Reserved	Reserved
1	0	0	0	1		IBUFHB	Reserved	Reserved

Table 6-26. VCOP IBUF Aliasing Truth Table (continued)

EVE_MEMMAP: VCOP_ALIAS Bit	EVE_MSW_CTL Bit				VCOP Address			
	IBUFLA	IBUFHA	IBUFLB	IBUFHB	10000h	14000h	30000h	34000h
1	0	0	1	0	IBUFLB		Reserved	Reserved
1	0	0	1	1	IBUFLB	IBUFHB	Reserved	Reserved
1	0	1	0	X		IBUFHA	Reserved	Reserved
1	0	1	1	X	IBUFLB	IBUFHA	Reserved	Reserved
1	1	0	X	0	IBUFLA		Reserved	Reserved
1	1	0	X	1	IBUFLA	IBUFHB	Reserved	Reserved
1	1	1	X	X	IBUFLA	IBUFHA	Reserved	Reserved

Table 6-27. Local EDMA IBUF Aliasing Truth Table

EVE_MEMMAP: LCL_EDMA_ALIAS Bit	EVE_MSW_CTL Bit				EDMA Address			
	IBUFLA	IBUFHA	IBUFLB	IBUFHB	10000h	14000h	30000h	34000h
0	0	0	0	0	IBUFLA	IBUFHA	IBUFLB	IBUFHB
0	0	0	0	1	IBUFLA	IBUFHA	IBUFLB	
0	0	0	1	0	IBUFLA	IBUFHA		IBUFHB
0	0	0	1	1	IBUFLA	IBUFHA		
0	0	1	0	0	IBUFLA		IBUFLB	IBUFHB
0	0	1	0	1	IBUFLA		IBUFLB	
0	0	1	1	0	IBUFLA			IBUFHB
0	0	1	1	1	IBUFLA			
0	1	0	0	0		IBUFHA	IBUFLB	IBUFHB
0	1	0	0	1		IBUFHA	IBUFLB	
0	1	0	1	0		IBUFHA		IBUFHB
0	1	0	1	1		IBUFHA		
0	1	1	0	0			IBUFLB	IBUFHB
0	1	1	0	1			IBUFLB	
0	1	1	1	0				IBUFHB
0	1	1	1	1				
1	0	0	X	X	IBUFLA	IBUFHA	Reserved	Reserved
1	0	1	X	0	IBUFLA	IBUFHB	Reserved	Reserved
1	0	1	X	1	IBUFLA		Reserved	Reserved
1	1	0	0	X	IBUFLB	IBUFHA	Reserved	Reserved
1	1	0	1	X		IBUFHA	Reserved	Reserved
1	1	1	0	0	IBUFLB	IBUFHB	Reserved	Reserved
1	1	1	0	1	IBUFLB		Reserved	Reserved
1	1	1	1	0		IBUFHB	Reserved	Reserved
1	1	1	1	1			Reserved	Reserved

6.1.3.16 Debug Support

The SCTM and SMSET modules facilitate application and kernel-level performance tuning, as well as general debug. The EVE subsystem minimizes debug intrusion.

6.1.3.16.1 ARP32 Debug Support

The EVE subsystem provides the entire IcePick™ and cross-triggering interfaces at the EVE boundary for use at the system level. EVE ARP32 configuration supports two hardware watchpoints. There are no hardware breakpoints.

The ARP32 core supports the following debug features:

- A 32-bit OCP slave port as the debug interface
- Memory-mapped registers showing and controlling debug status of the core
- Ownership mechanism for managing use of all debug features by application code or debug software
- View of CPU resources (program memory, data memory, CPU architectural register, CPU control registers)
- Real-time debug access to CPU resources, if the CPU does not need to be halted to make a debug access
- Unlimited number of simultaneous software breakpoints using the BKPT instruction supported by the core
- Limited number of simultaneous hardware watchpoint (HWWP) units
- Run control:
  - Halting cpu at SWBP, HWBP, HWWP, incoming external Trigger
  - Single stepping in processor - instruction by instruction
  - Resuming CPU from halted state
- Cross triggering:
  - Halting the CPU on an incoming external trigger
  - Sending a trigger pulse out when the CPU halts as a result of SWBP, HWBP, HWWP trigger/match
  - Sending a trigger pulse out when HWBP, HWWP match is detected, even if the CPU is not configured to halt
- Debug control and status registers:
  - Shows the status of debug access, CPU core execution status
  - Shows miscellaneous status like RESET and IDLE state

6.1.3.16.2 SCTM

The SCTM provides general-purpose counters and timers. It also allows counting and measuring of specific internal EVE signals (for example, stall signals). The general-purpose counter/timer functionality is used by both BIOS software and application software. The signal-measuring functions are typically used for performance-related debug. See Section 6.1.6 for more details concerning SCTM.

6.1.3.16.2.1 SCTM Configuration

Table 6-28 shows the SCTM configuration for EVE. The SCTM module is configured for eight 32-bit counters, two of which can be configured as timers. Timers are counters that include a threshold MMR value and an interrupt. The interrupt is pulsed when the counter reaches the programmed threshold. It is possible for any two even and odd counter pairs to be chained (through MMR) to operate as a 64-bit counter. The SCTM configuration in EVE supports the atomic read feature on counter[3:0] pair and counter[5:6] pair.

Table 6-28. SCTM Configuration in EVE

Generic	SCTM Feature/Parameter Details	Value	Number on Device
CTM_NUMINPT	Number of event input signals supported	0 to 127	31
CTM_NUMCNTR	Number of counters in the module	1 to 32	8
CTM_NUMTIMR	Number of timers	0 to 8	2
CTM_TIMINTPOLARITY	Timer interrupt polarity	0 = Active low 1 = Active high	1
CTM_TIMINTWIDTH	Timer interrupt pulse width	1 or more	2
CTM_NUMSTM	Number of counters for STM export	0 to 32	0
CTM_CCMAVAIL	CCM frame export available	0 = No 1 = Yes	0
CTM_NUMDBGSGL	Number of debug event signals	0 to 8	0

Table 6-28. SCTM Configuration in EVE (continued)

Generic	SCTM Feature/Parameter Details	Value	Number on Device
CTM_ASYNCIDLEREQ	Whether IDLE request is asynchronous signal	0 = No 1 = Yes	0
CTM_CNTR_ CHAINSHADOW	Atomicity feature for even-indexed counters	One per counter. 0 = No chain shadow feature 1 = Chain shadow feature in counter chain mode	Counter numbers 2 and 4 have the chaining feature.

6.1.3.16.2.2 SCTM Resources Reserved for BIOS

Timer 0 (and associated interrupt) is exclusively reserved for BIOS software for driving the BIOS tick.

Timer 1 (and associated interrupt) is free to be used by application software.

BIOS also requires a 64-bit free-running counter with low-latency accessibility that is used during application logging. This is provided through the ARP32 free-running 64-bit counter.

6.1.3.16.2.3 SCTM Event Mapping

The SCTM events include low-level stall and duration signals sourced by various components of EVE, mainly program cache-related signals and VCOP-related signals. Table 6-29 summarized the event mapping. Event inputs start numbering at 1, because event0 is reserved and SCTM internally uses the functional clock as event0.

For more details on the functionality of each signal listed in the table, see *EVE Programmer's Guide*. Contact your TI representative for instructions how to access this document.

The SCTM module operates at half the clock rate (EVEx_GFCLK), CLK2 = 0.5 × CLK1. The event sources include CLK2 and CLK1 signals. The CLK2 relative signals are connected directly to SCTM. The CLK1 relative signals are conditioned by EVE-level logic to scale from CLK1 to CLK2. Because these signals are Duration type signals, EVE logic asserts a CLK2 pulse for every two CLK1 pulses detected. This results in, at most, 1 CLK1 cycle of inaccuracy in the CLK2 duration reported by the SCTM.

The SCTM allows measurement of active-high signals. The Type column represents the functional characteristics of the source signals, which can be Pulse, Duration or Edge. The SCTM Mode column describes the typical mode for the SCTM counter when used with a particular signal. The options are event or duration mode. The pulse signals are driven active one cycle for each occurrence, a sequence of consecutive active cycles represents multiple occurrences. For a type of signal where the duration implies the number of occurrences in particular state (for example, the number of cache hits) the furation mode of SCTM is used. The Duration type represents a signal that may stay active for multiple cycles on a given occurrence. These are stall signals, where software tried to determine the number of cycles in the stalled state. Duration mode indicates the time stalled. Event mode indicates the number of unique occurrences of stalls. Event is useful in this case since there is always a deassertion between occurrences. The Edge type goes active for undefined periods of time but must go inactive before the next occurrence. Event mode of the SCTM is used for these signals.

Table 6-29. SCTM Events

SCTM Event	Name	Source	Type	SCTM Mode	Clock
1	cache_miss_count	ARP32_Pcache	Pulse	Duration	CLK2
2	cache_hit_count	ARP32_Pcache	Pulse	Duration	CLK2
3	cache_miss_stall	ARP32_Pcache	Duration	Duration or Event	CLK2
4	prefetch_compulsory_count	ARP32_Pcache	Pulse	Duration	CLK2
5	Prefetch_lookahead_count	ARP32_Pcache	Pulse	Duration	CLK2
6	prefetch_hit_count	ARP32_Pcache	Pulse	Duration	CLK2
7	line_buffer_hit_count	ARP32_Pcache	Pulse	Duration	CLK2
8	Prefetch_line_count	ARP32_Pcache	Pulse	Duration	CLK2

Table 6-29. SCTM Events (continued)

SCTM Event	Name	Source	Type	SCTM Mode	Clock
9	prefetch_discard_stall	ARP32_Pcache	Duration	Duration or event	CLK2
10	tpcc_aet	EDMA	Duration	Duration or event	CLK2
11	arp32_int4	INTC	Duration	Duration or event	CLK2
12	arp32_int5	INTC	Duration	Duration or event	CLK2
13	arp32_int6	INTC	Duration	Duration or event	CLK2
14	arp32_int7	INTC	Duration	Duration or event	CLK2
15	vcop_busy	VCOP	Pulse	Duration	CLK2
16	vcop_idle_and_done	VCOP	Pulse	Duration	CLK2
17	vcop_wait_for_arp32	VCOP	Pulse	Duration	CLK2
18	vcop_arp32_awaits	VCOP	Pulse	Duration	CLK2
19	vcop_overhead	VCOP	Pulse	Duration	CLK1
20	vcop_ld_stall_by_st	VCOP	Pulse	Duration	CLK1
21	vcop_op_stall_by_ldst	VCOP	Pulse	Duration	CLK1
22	vcop_op_stall_by_dependency	VCOP	Pulse	Duration	CLK1
23	vcop_rd_ibufl	VCOP	Pulse	Duration	CLK1
24	vcop_rd_ibufh	VCOP	Pulse	Duration	CLK1
25	vcop_rd_wbuf	VCOP	Pulse	Duration	CLK1
26	vcop_wr_ibufl	VCOP	Pulse	Duration	CLK1
27	vcop_wr_ibufh	VCOP	Pulse	Duration	CLK1
28	vcop_wr_wbuf	VCOP	Pulse	Duration	CLK1
29	vcop_loop_start	VCOP	Edge	Event	CLK2
30	vcop_done	VCOP	Edge	Event	CLK2
31	arp32_nmi	INTC	Duration	Duration or event	CLK2
32	arp32_int8	INTC	Duration	Duration of event	CLK2
33	arp32_int9	INTC	Duration	Duration of event	CLK2
34	arp32_int10	INTC	Duration	Duration of event	CLK2
35	arp32_int11	INTC	Duration	Duration of event	CLK2
36	arp32_int12	INTC	Duration	Duration of event	CLK2
37	arp32_int13	INTC	Duration	Duration of event	CLK2
38	arp32_int14	INTC	Duration	Duration of event	CLK2
39	arp32_int15	INTC	Duration	Duration of event	CLK2

6.1.3.16.2.4 SCTM Halt and Idle Modes

The SCTM module has a method for enabling and disabling counters based on whether the CPU is in debug-halt (SUSPEND) state or idle state (IDLE). This state information is provided from the ARP32 CPU to the SCTM module through the SUSPEND and IDLE inputs.

6.1.3.16.3 SMSET

The SMSET allows tracing of key EVE subsystem events, as well as software messages from the ARP32 CPU. SMSET accepts software messages through its OCP target port, and accepts key system events through the system event input. These messages or events are queued locally in SMSET and written to the chip-level software trace module through the SMSET and EVE OCP debug initiator port. The system trace macrocell (STM) then traces these messages along with trace content with other chip-level agents. For details, see Chapter 26, *On-Chip Debug Support*, for details.

6.1.3.16.3.1 SMSET Configuration

Table 6-30 summarizes the SMSET configuration in EVEN. The SMSET module is configured to support both event and software message tracing, and supports cross-triggering. System events (that are not stallable and can overflow) include a 4-deep buffer; and software events (that can be stalled and never overflow) support a 2-deep buffer.

Table 6-30. SMSET Configuration in EVE

Generic	SCTM Feature/Parameter Details	Value	In EVE Subsystem	Number on Device
NB_EVENTS	Number of system events	1 to 255	52	9
SW_MESSAGE	Software messages support	0 to 1	1	1
TRIG_MIN_WIDTH	Minimum trigger input pulse (L4 cycles)	1 to 8	2	2
EVENTS_BUF_DEPTH	System events buffer depth	1 to 16	4	4
SW_BUF_DEPTH	Software message buffer depth	0 to 16	2	2

6.1.3.16.3.2 SMSET Event Mapping

The SMSET events include higher-level start and end task events. Table 6-31 summarize the event mapping. Refer to the *EVE Programmer's Guide* for details on the functionality of each signal listed. Contact your TI representative for instructions how to access the *EVE Programmer's Guide*.

The SMSET module operates at the CLK2 rate.

The SMSET traces events on either a high-to-low transition or low-to-high transition, but not both. Thus, the EDMAs tpcc_aet signal is not directly compatible with SMSET tracing. The tpcc_aet is an active-high signal that is used to measure the duration of a specific EDMA transfer. The tpcc_aet signal is programmed (in EDMA CC MMRs) to go high when a specific trigger is detected and to go low when a specific completion code is received. EVE logic implements custom tpcc_aet_start and tpcc_aet_end signals. The tpcc_aet_start signal pulses high (for one cycle) when tpcc_aet transitions to high, and the tpcc_aet_end signal pulses high (for one cycle) when tpcc_aet transitions to low.

The Type column in Table 6-31 lists the functional characteristics of the source signal (Pulse, Duration, or Edge).

The Pulse signals are expected to be driven active one cycle for each occurrence. A sequence of consecutive active cycles represent multiple occurrences. Duration mode of SCTM makes sense for this type of signal where the duration implies the number of occurrences in a particular state (for example, the number of cache hits).

The Duration type represents a signal that may stay active for multiple cycles on a given occurrence. These are typically stall signals where the user is trying to determine the number of cycles in the stalled state. Duration mode indicates the total time stalled. Event mode will indicate the number of unique occurrences of stalls. "Event" is useful in this case because there will always be a deassertion between occurrences.

An Edge signal goes active for undefined period of time but is goes inactive before the next occurrence. Event mode of the SCTM must be used for these signals.

Table 6-31. List of SMSET Events

SMSET Event	Name	Source	Type	SMSET Mode	Clock
0	tpcc_aet_start	EDMA	Pulse	Duration	CLK2
1	tpcc_aet_stop	EDMA	Pulse	Duration	CLK2
2	arp32_int4	INTC	Duration	Duration or event	CLK2
3	arp32_int5	INTC	Duration	Duration or event	CLK2
4	arp32_int6	INTC	Duration	Duration or event	CLK2
5	arp32_int7	INTC	Duration	Duration or event	CLK2

Table 6-31. List of SMSET Events (continued)

SMSET Event	Name	Source	Type	SMSET Mode	Clock
6	vcop_loop_start	VCOP	Edge	Event	CLK2
7	vcop_done	VCOP	Edge	Event	CLK2
8	arp32_nmi	INTC	Duration	Duration or event	CLK2
9	arp32_int8	INTC	Duration	Duration or event	CLK2
10	arp32_int9	INTC	Duration	Duration or event	CLK2
11	arp32_int10	INTC	Duration	Duration or event	CLK2
12	arp32_int11	INTC	Duration	Duration or event	CLK2
13	arp32_int12	INTC	Duration	Duration or event	CLK2
14	arp32_int13	INTC	Duration	Duration or event	CLK2
15	arp32_int14	INTC	Duration	Duration or event	CLK2
16	arp32_int15	INTC	Duration	Duration or event	CLK2

6.1.3.17 EVE L2_FNOC Interconnect

A level 2 (L2) Interconnect from Arteris - FlexNoC® is instantiated in the EVE subsystem.

6.1.3.17.1 EVE L2_FNOC Flag Mux and Error Log Registers

The EVE_L2_FNOC flagmux and error log registers are used for error logging and flag muxing purposes. The status information stored in them can be used to resolve issues related to EVE_NoC access coflicts, debugging, and so forth. The related registers are described in Section 6.1.5.3. For more information on flag muxing and error logging, see Section 9.2.3.5 and Section 9.2.3.8 in Section 9.2 *L3_MAIN Interconnect.*

6.1.4 EVE Programming Model

6.1.4.1 Boot

The EVE boot process follows:

1. EVE_RST and EVE_CPU_RST are asserted for a minimum of 16 cycles.
2. Host processor (through PRCM) deasserts eve_rst_n (eve_arp32vcop_rst_n remains asserted). This lets the system access EVE-level memories/MMRs through the OCP target bus while ARP32 remains in reset.
3. Host processor initializes EVE state:
   a. Enable parity generation for EVE memories: clear DMEM, WBUF, IBUF memories to initialize parity state (optional and can be handled at runtime by ARP32).
   b. Load ARP32 code and data into desired memories:
      i. Code is typically in EMIF/DDR memory, but can be any valid chip-level memory.
      ii. Data is typically in EVE DMEM, but can also be in EMIF/DDR or other chip level memory
   c. Initialize MMU TLB and related page tables: A translation must exist for address 0h because ARP32 boot vector is always at 0h (for MMU programming model see Chapter 15, *Memory Management Unit*).
   d. Load VCOP memories: Ownership defaults to non-VCOP so no extra steps are required (optional and can be handled at runtime by ARP32).
   e. Initialize EDMA, INTC (optional and can be handled at runtime by ARP32).
4. Host processor (through PRCM) deasserts eve_arp32vcop_rst_n
5. ARP32 initializes EVE state, for steps not handled by host processor:
   a. Configure EDMA.
   b. Configure interrupt controller: Enable all error-related interrupts.

c. Enable parity on all memories: Scrub memories to initialize parity state.
d. Initialize VCOP working buffers.

6. ARP32 manages ping-pong processing:
   a. Page in input buffers through EDMA.
   b. Assign memory ownership of I/O buffers to VCOP or EDMA/SYS.
   c. Page out output buffers through EDMA.
   d. Initialize VCOP program.
   e. Wait for VCOP completion, EDMA input paging completion, and EDMA output paging completion.

6.1.4.2 Task Change and Program Cache Prefetch

The ARP32 switches tasks in a number of ways. In this context, a task represents a unit of program flow that typically executes for many ping-pong time slices and represents a code section that is typically 16k to 32k long.

6.1.4.2.1 Simple or Unoptimized Branch to New Task

In the simplest case, the ARP32 branches to the new code section. The existence of program cache provides reasonable performance, especially if the new task program code is cache friendly and is executed a number of times before the next task switch occurs.

6.1.4.2.2 Prefetch, Wait, then Branch to New Task

To optimize the task switch efficiency, use the software-directed preload operation. In this case:

- Complete with the current task.
- Execute a Software directed preload command (as described in Section 6.1.3.4.3, Software Direct Preload).
- Wait for preload completion.
- Branch to the new address.

Given that the cache view and system view of the program address are still the same, there are no additional overhead and constraints.

6.1.4.2.3 Hidden Prefetch

If any two tasks are small enough to fit in program cache, software uses SDP to fetch the contents of the next tasks into program cache while operating on the current task. If that program cache is direct mapped, extra care must be taken to ensure that any two tasks do not reside in the same modulo-32k range of memory.

6.1.4.3 Interrupts

The following pseudocode describes a typical and simple workflow for ARP32 to service pending interrupts. This flow is not the only possible sequence and is not meant to represent a mandated software sequence:

1. ARP32 enables and maps interrupts:
   a. Enable ARP32 global interrupt enable (ARP32.CSR.GIE).
   b. Enable ARP32 nonmaskable interrupts (ARP32.IER.NMIE).
   c. Enable ARP32 interrupt inputs (ARP32.IER[*n*]).
   d. Map EVE-level interrupts to the appropriate ARP32 interrupt input (ARP32_INTn_IRQENABLE_SET).
2. When interrupt is asserted, ARP32 branches into interrupt vector code:
   a. ARP32 hardware clears the GIE bit, new interrupts are ignored
   b. ARP32 software reads ARP32_INTn_IRQSTATUS
   c. Clear to be serviced interrupts in ARP32_INTn_IRQSTATUS (by issuing write of 1)
   d. For each pending interrupt in ARP32_INTn_IRQSTATUS:
      i. ARP32 software reads IP interrupt status (<IP_INT>_IRQSTATUS)
      ii. For each pending interrupt in <IP_INT>_IRQSTATUS:
         1. May not be necessary for pulsed or timer type interrupts.
         2. Perform functional operations necessary to service interrupt.
         3. Clear IP source interrupt status (by setting correct bit in <IP_INT>_IRQSTATUS to 1).
         4. Ideally, only exit loop after <IP_INT>_IRQSTATUS is all 0s.
      iii. Only exit loop after <IP_INT>_IRQSTATUS is all 0s
   e. ARP32 software returns from interrupt through BIRP (branch interrupt return pointer) instruction: ARP32 hardware sets the GIE bit (by copy of SCSR to CSR, new interrupts are enabled) and clears IFR*n*.

For ARP32 register and state details (especially for CSR, GIE, and SCSR functionalities), see the *ARP32 CPU and Instruction Set Reference Guide*.

6.1.4.4 Safety Considerations

This section highlights the known hardware that must be enabled in a safety-conscious system, and specific software techniques that increases the level of safety of EVE.

6.1.4.4.1 Memory Error Detection

Memory error detection provides hardware to detect single-bit errors in data memories (DMEM, WBUF, and IBUF) and detect double-bit errors in program memories. This hardware can capable detect stuck-at faults, as well as radiation-induced soft errors.

- For data memories, to avoid false error signaling, memories must be initialized with known values before reading those memories. See Section 6.1.3.3.6, *Memory Error Detection*.
- All sources of error detection error interrupts must be enabled. See Section 6.1.3.3.6, *Memory Error Detection,* and Section 6.1.3.9.1, *EVE Interrupt Sources – Memory Switch and Parity Error Interrupts*.
- Software must be tested at development time to ensure robust response to random errors. See Section 6.1.3.3.6.3, *Parity Error Testability*.
- Software must be tested at runtime to ensure hardware is properly operating. See Section 6.1.3.3.6.3, *Parity Error Testability*.

6.1.4.4.2 MMU

The MMU is configured to minimize the view of system memory that ARP32 and EDMA are allowed to access. This prevents either coding bugs or faulty hardware from impacting other critical areas of the system, such as the system host code and data sections. For detailed explanation of the MMU and its programming, see Chapter 15, *Memory Management Unit (MMU)*.

6.1.4.4.3 Firewall

The system firewall is used to prevent access by unintended system initiators from accidentally modifying EVE memory; or accesses from intended system initiators to unintended EVE memory. For firewall configuration see Section 9.2, *Interconnect*.

6.1.4.4.4 Interconnect

The EVE detects accesses to reserved locations in the memory-map and informs the system.

6.1.4.4.5 Application Stability/Sequencing

The Mailbox module or the local EVE SMSET module can be leveraged to report either heartbeat or sequence messages to the system host. By sending deterministic messages, the host ARM verifies that EVE is still active and following a reasonable path of code execution.

6.1.4.4.6 Interrupt Servicing

ISRs are implemented to count the expected number of interrupts (for example, EDMA, mailbox, timer) and check for extra or missing interrupts where possible. For example, if the ARP32_INTn_IRQSTATUS[31:0] EVENT bit field is set, then at least one of the source registers is set as well. Or, if an EDMA interrupt is received, then at least one of the CIPR bits is set. For EDMA interrupts, software tracks DMAs submitted versus DMAs completed.

6.1.5 EVE Subsystem Register Manual

6.1.5.1 EVE Instance Summary

Table 6-32 lists EVE instances.

Table 6-32. EVE Instance Summary

Module Name	Base Address	Size
EVE	0x4200 0000	4KiB
EVE_DSP(1)	0x0200 0000	4KiB
EVE_L2_FNOC (2)	0x0208 A000	1024 Bytes

(1) Address space is NOT visible on L3_MAIN for the device. It is visible only within the DSP_ICFG internal configuration space, hence accessible only by the DSP C66x CPU.
(2) Address space is NOT visible on L3_MAIN for the device.

6.1.5.2 EVE Registers

6.1.5.2.1 EVE Register Summary

Table 6-33 summarizes the EVE registers.

Table 6-33. EVE Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	EVE Physical Address	EVE_DSP Physical Address DSP Private Access
EVE_REVISION	R	32	0x8 0000	0x4208 0000	0x0208 0000
EVE_HWINFO	R	32	0x8 0004	0x4208 0004	0x0208 0004
EVE_SYSCONFIG	RW	32	0x8 0008	0x4208 0008	0x0208 0008
EVE_STAT	R	32	0x8 000C	0x4208 000C	0x0208 000C
EVE_DISC_CONFIG	RW	32	0x8 0010	0x4208 0010	0x0208 0010
EVE_BUS_CONFIG	RW	32	0x8 0014	0x4208 0014	0x0208 0014
EVE_VCOP_HALT_CONFIG	RW	32	0x8 0018	0x4208 0018	0x0208 0018
EVE_MMU_CONFIG	RW	32	0x8 001C	0x4208 001C	0x0208 001C
EVE_MEMMAP	RW	32	0x8 0020	0x4208 0020	0x0208 0020
EVE_MSW_CTL	RW	32	0x8 0024	0x4208 0024	0x0208 0024
EVE_MSW_ERR	RW	32	0x8 0028	0x4208 0028	0x0208 0028
EVE_MSW_ERRADDR	R	32	0x8 002C	0x4208 002C	0x0208 002C
EVE_PC_INV	RW	32	0x8 0040	0x4208 0040	0x0208 0040
EVE_PC_IBAR	RW	32	0x8 0050	0x4208 0050	0x0208 0050
EVE_PC_IBC	RW	32	0x8 0054	0x4208 0054	0x0208 0054
EVE_PC_ISAR	RW	32	0x8 0058	0x4208 0058	0x0208 0058
EVE_PC_ISAR_DONE	R	32	0x8 005C	0x4208 005C	0x0208 005C
EVE_PC_PBAR	RW	32	0x8 0060	0x4208 0060	0x0208 0060
EVE_PC_PBC	RW	32	0x8 0064	0x4208 0064	0x0208 0064
EVE_PMEM_ED_CTL	RW	32	0x8 0080	0x4208 0080	0x0208 0080
EVE_PMEM_ED_STAT	RW	32	0x8 0084	0x4208 0084	0x0208 0084
EVE_PMEM_EDADDR	R	32	0x8 0088	0x4208 0088	0x0208 0088
EVE_DMEM_ED_CTL	RW	32	0x8 0090	0x4208 0090	0x0208 0090
EVE_DMEM_ED_STAT	RW	32	0x8 0094	0x4208 0094	0x0208 0094
EVE_DMEM_EDADDR	R	32	0x8 0098	0x4208 0098	0x0208 0098
EVE_DMEM_EDADDR_BO	R	32	0x8 009C	0x4208 009C	0x0208 009C
EVE_WBUF_ED_CTL	RW	32	0x8 00A0	0x4208 00A0	0x0208 00A0
EVE_WBUF_ED_STAT	RW	32	0x8 00A4	0x4208 00A4	0x0208 00A4
EVE_WBUF_EDADDR	R	32	0x8 00A8	0x4208 00A8	0x0208 00A8
EVE_WBUF_EDADDR_BO	R	32	0x8 00AC	0x4208 00AC	0x0208 00AC
EVE_IBUF_ED_CTL	RW	32	0x8 00B0	0x4208 00B0	0x0208 00B0
EVE_IBUF_ED_STAT	RW	32	0x8 00B4	0x4208 00B4	0x0208 00B4

Table 6-33. EVE Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	EVE Physical Address	EVE_DSP Physical Address DSP Private Access
EVE_IBUF_EDADDR	R	32	0x8 00B8	0x4208 00B8	0x0208 00B8
EVE_IBUF_EDADDR_BO	R	32	0x8 00BC	0x4208 00BC	0x0208 00BC
EVE_ED_ARP32_DISC_EN	RW	32	0x8 00F8	0x4208 00F8	0x0208 00F8
EVE_ED_OCPI_DISC_EN	RW	32	0x8 00FC	0x4208 00FC	0x0208 00FC
EVE_MSW_ERR_IRQSTATUS_RAW	RW	32	0x8 0110	0x4208 0110	0x0208 0110
EVE_MSW_ERR_IRQSTATUS	RW	32	0x8 0114	0x4208 0114	0x0208 0114
EVE_MSW_ERR_IRQENABLE_SET	RW	32	0x8 0118	0x4208 0118	0x0208 0118
EVE_MSW_ERR_IRQENABLE_CLR	RW	32	0x8 011C	0x4208 011C	0x0208 011C
EVE_ED_LCL_IRQSTATUS_RAW	RW	32	0x8 0120	0x4208 0120	0x0208 0120
EVE_ED_LCL_IRQSTATUS	RW	32	0x8 0124	0x4208 0124	0x0208 0124
EVE_ED_LCL_IRQENABLE_SET	RW	32	0x8 0128	0x4208 0128	0x0208 0128
EVE_ED_LCL_IRQENABLE_CLR	RW	32	0x8 012C	0x4208 012C	0x0208 012C
ARP32_NMI_IRQSTATUS_RAW	RW	32	0x8 0200	0x4208 0200	0x0208 0200
ARP32_NMI_IRQSTATUS	W	32	0x8 0204	0x4208 0204	0x0208 0204
ARP32_NMI_IRQENABLE_SET	RW	32	0x8 0208	0x4208 0208	0x0208 0208
ARP32_NMI_IRQENABLE_CLR	W	32	0x8 020C	0x4208 020C	0x0208 020C
ARP32_INTn_IRQSTATUS_RAW (1)	RW	32	0x8 01D0 + (0x10*n)	0x4208 01D0 + (0x10*n)	0x0208 01D0 + (0x10*n)
ARP32_INTn_IRQSTATUS (1)	W	32	0x8 01D4 + (0x10*n)	0x4208 01D4 + (0x10*n)	0x0208 01D4 + (0x10*n)
ARP32_INTn_IRQENABLE_SET (1)	RW	32	0x8 01D8 + (0x10*n)	0x4208 01D8 + (0x10*n)	0x0208 01D8 + (0x10*n)
ARP32_INTn_IRQENABLE_CLR (1)	W	32	0x8 01DC + (0x10*n)	0x4208 01DC + (0x10*n)	0x0208 01DC + (0x10*n)
ARP32_IRQWAKEEN	RW	32	0x8 02FC	0x4208 02FC	0x0208 02FC
MMR_LOCKi (2)	RW	32	0x8 0300 + (0x4*i)	0x4208 0300 + (0x4*i)	0x0208 0300 + (0x4*i)
MISR_CTL	RW	32	0x8 0400	0x4208 0400	0x0208 0400
MISR_CLEAR	RW	32	0x8 0404	0x4208 0404	0x0208 0404
MISR0_A	RW	32	0x8 0410	0x4208 0410	0x0208 0410
MISR0_D	RW	32	0x8 0414	0x4208 0414	0x0208 0414
MISR1_A	RW	32	0x8 0418	0x4208 0418	0x0208 0418
MISR1_D	RW	32	0x8 041C	0x4208 041C	0x0208 041C
MISR2_Dk (3)	RW	32	0x8 0420 + (0x4*k)	0x4208 0420 + (0x4*k)	0x0208 0420 + (0x4*k)
EVE_IRQ_EOI	RW	32	0x8 0500	0x4208 0500	0x0208 0500
EVE_ED_OUT_IRQSTATUS_RAW	RW	32	0x8 0510	0x4208 0510	0x0208 0510
EVE_ED_OUT_IRQSTATUS	RW	32	0x8 0514	0x4208 0514	0x0208 0514
EVE_ED_OUT_IRQENABLE_SET	RW	32	0x8 0518	0x4208 0518	0x0208 0518
EVE_ED_OUT_IRQENABLE_CLR	RW	32	0x8 051C	0x4208 051C	0x0208 051C
EVE_INTk_OUT_IRQSTATUS_RAW (3)	RW	32	0x8 0520 + (0x10*k)	0x4208 0520 + (0x10*k)	0x0208 0520 + (0x10*k)
EVE_INTk_OUT_IRQSTATUS (3)	RW	32	0x8 0524 + (0x10*k)	0x4208 0524 + (0x10*k)	0x0208 0524 + (0x10*k)

(1) n = 4 to 7 for EVE
n = 4 to 7 for EVE_DSP

(2) i = 0 to 9 for EVE
i = 0 to 9 for EVE_DSP

(3) k = 0 to 3 for EVE
k = 0 to 3 for EVE_DSP

Table 6-33. EVE Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	EVE Physical Address	EVE_DSP Physical Address DSP Private Access
EVE_INTk_OUT_IRQENABLE_SET [3]	RW	32	0x8 0528 + (0x10*k)	0x4208 0528 + (0x10*k)	0x0208 0528 + (0x10*k)
EVE_INTk_OUT_IRQENABLE_CLR [3]	RW	32	0x8 052C + (0x10*k)	0x4208 052C + (0x10*k)	0x0208 052C + (0x10*k)
ARP32_INTj_IRQSTATUS_RAW [4]	RW	32	0x8 0580 + (0x10*j)	0x4208 0580 + (0x10*j)	0x0208 0580 + (0x10*j)
ARP32_INTj_IRQSTATUS [4]	RW	32	0x8 0584 + (0x10*j)	0x4208 0584 + (0x10*j)	0x0208 0584 + (0x10*j)
ARP32_INTj_IRQENABLE_SET [4]	RW	32	0x8 0588 + (0x10*j)	0x4208 0588 + (0x10*j)	0x0208 0588 + (0x10*j)
ARP32_INTj_IRQENABLE_CLR [4]	RW	32	0x8 058C + (0x10*j)	0x4208 058C + (0x10*j)	0x0208 058C + (0x10*j)
ARP32_INT14_IRQSTATUS_RAW	RW	32	0x8 0680	0x4208 0680	0x0208 0680
ARP32_INT14_IRQSTATUS	RW	32	0x8 0684	0x4208 0684	0x0208 0684
ARP32_INT14_IRQENABLE_SET	RW	32	0x8 0688	0x4208 0688	0x0208 0688
ARP32_INT14_IRQENABLE_CLR	RW	32	0x8 068C	0x4208 068C	0x0208 068C
ARP32_INT15_IRQSTATUS_RAW	RW	32	0x8 0690	0x4208 0690	0x0208 0690
ARP32_INT15_IRQSTATUS	RW	32	0x8 0694	0x4208 0694	0x0208 0694
ARP32_INT15_IRQENABLE_SET	RW	32	0x8 0698	0x4208 0698	0x0208 0698
ARP32_INT15_IRQENABLE_CLR	RW	32	0x8 069C	0x4208 069C	0x0208 069C
EVE_GPOUTm [5]	RW	32	0x8 0700 + (0x10*m)	0x4208 0700 + (0x10*m)	0x0208 0700 + (0x10*m)
EVE_GPOUTm_SET [5]	RW	32	0x8 0704 + (0x10*m)	0x4208 0704 + (0x10*m)	0x0208 0704 + (0x10*m)
EVE_GPOUTm_CLR [5]	RW	32	0x8 0708 + (0x10*m)	0x4208 0708 + (0x10*m)	0x0208 0708 + (0x10*m)
EVE_GPOUTm_PULSE [5]	RW	32	0x8 070C + (0x10*m)	0x4208 070C + (0x10*m)	0x0208 070C + (0x10*m)
EVE_GPIN0	R	32	0x8 0740	0x4208 0740	0x0208 0740
EVE_GPIN1	R	32	0x8 0744	0x4208 0744	0x0208 0744
EVE_CME_DONE_GPOUT	RW	32	0x8 0780	0x4208 0780	0x0208 0780
EVE_CME_DONE_GPOUT_SET	RW	32	0x8 0784	0x4208 0784	0x0208 0784
EVE_CME_DONE_GPOUT_CLR	RW	32	0x8 0788	0x4208 0788	0x0208 0788
EVE_CME_DONE_GPOUT_PULSE	RW	32	0x8 078C	0x4208 078C	0x0208 078C
EVE_CME_DONE_SEL	RW	32	0x8 0790	0x4208 0790	0x0208 0790
EVE_CME_DONE_EN	RW	32	0x8 0794	0x4208 0794	0x0208 0794
EVE_PM_STAT0	R	32	0x8 0FE0	0x4208 0FE0	0x0208 0FE0
EVE_PM_STAT1	R	32	0x8 0FE4	0x4208 0FE4	0x0208 0FE4
EVE_DBGOUT	RW	32	0x8 0FE8	0x4208 0FE8	0x0208 0FE8
EVE_RSVD0	RW	32	0x8 0FF4	0x4208 0FF4	0x0208 0FF4
EVE_RSVD1	RW	32	0x8 0FF8	0x4208 0FF8	0x0208 0FF8
EVE_TEST	RW	32	0x8 0FFC	0x4208 0FFC	0x0208 0FFC

[4] j = 8 to 13 for EVE
j = 8 to 13 for EVE_DSP
[5] m = 0 to 1 for EVE
m = 0 to 1 for EVE_DSP

6.1.5.2.2 EVE Register Description

through describe the individual register bits.

Table 6-34. EVE_REVISION

Address Offset	0x0008 0000		
Physical Address	0x4208 0000 0x0208 0000	**Instance**	EVE EVE_DSP
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REVISION																															

Bits	Field Name	Description	Type	Reset
31:0	REVISION	Revision	R	0x0

Table 6-35. Register Call Summary for Register EVE_REVISION

Embedded Vision Engine (EVE) Subsystem
- Lock and Unlock Feature: [0]
- EVE Register Summary: [1]

Table 6-36. EVE_HWINFO

Address Offset	0x0000 0004		
Physical Address	0x4208 0004 0x0208 0004	**Instance**	EVE EVE_DSP
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INFO																												EVENUM			

Bits	Field Name	Description	Type	Reset
31: 4	INFO	0x0: No configurable options in EVE	R	0x00
3:0	EVENUM	EVE instance number set by eve_num inputs. In a multi-EVE system must be set to unique/incrementing values for each EVE.	R	0x0

Table 6-37. Register Call Summary for Register EVE_HWINFO

Embedded Vision Engine (EVE) Subsystem
- Lock and Unlock Feature: [0]
- EVE Register Summary: [1]

Table 6-38. EVE_SYSCONFIG

Address Offset	0x0000 0008		
Physical Address	0x4208 0008 0x0208 0008	**Instance**	EVE EVE_DSP
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								STANDBYMODE		IDLEMODE		FREEEMU		SOFTRESET	

Bits	Field Name	Description	Type	Reset
31:6	RESERVED	Reserved. Read returns 0s	RW	0x0000
5:4	STANDBYMODE	00: Force-Standby mode: This mode is a backup mode intended to be used only if the smart-idle mode is bugged. When in this mode / the SAF asserts with minimal hardware condition the status saying :I am in standby:. It is the responsibility of the software to ensure that the SAF is in a correct quiet state before programming this mode. Additionally when in this mode / the SAF is not allowed to generate wakeup reques . 01: No-Standby: This mode is a backup mode intended to be used only if the smart-idle mode is bugged. When in this mode / the SAF never asserts the status declaring that the system is in standby. 10: Smart-Standby: default. EVE generates the standby status based upon all hardware internal status / namely after having performed all hardware operations necessary to be in a correct quiet state. Additionally when in this mode / the SAF is not allowed to generate wakeup request. 11: Smart-Standby-Wkup: Same as Smart-Standby. (EVE generates the standby status based upon all hardware internal status / namely after having performed all hardware operations necessary to be in a correct quiet state). . Additionally when in this mode / the SAF is allowed to generate wakeup request	RW	0x0
3:2	IDLEMODE	00: Force-idle: This mode is a backup mode intended to be used only if the smart-idle mode is bugged. When in this mode the IAF acknowledges a request to go idle from the power manager with minimal hardware condition. It is the responsibility of the software to ensure that the IAF are in a correct quiet state before requesting a force-idle transition. Additionaly when in this mode the IAF is not allowed to generate any wakeup request. 01: No-idle: When in this mode the IAF disregards any request to go idle from the power manager. 10: Smart-idle: default mode. default. When in this mode / the IAF acknowledges a request to go idle from the power manager after having performed all hardware operations necessary to be in a correct quiet state. Additionally when in this mode / the IAF is not allowed to generate any wakeup reques 11: SmartIdleWkup : When in this mode / the IAF acknowledges a request to go idle from the power manager after having performed all hardware operations necessary for the IAF to be in a correct quiet state. Additionally when in this mode / the IAF is allowed to generate wakeup request	RW	0x0
1	FREEEMU	Resered. Note that SCTM has free control bit to define the timer operation during ARP32 debug halt mode	R	0x0

Bits	Field Name	Description	Type	Reset
0	SOFTRESET	Reserved	R	0x0

Table 6-39. Register Call Summary for Register EVE_SYSCONFIG

Embedded Vision Engine (EVE) Subsystem

- Extended Duration Sleep: [0][1]
- Lock and Unlock Feature: [2]
- EVE Register Summary: [3]

Table 6-40. EVE_STAT

Address Offset	0x0008 000C		
Physical Address	0x4208 000C 0x0208 000C	**Instance**	EVE EVE_DSP
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED										OCPI_DISC_STAT		RESERVED		ARP32_DISC_STATUS		RESERVED							INT_OUT_STAT	ARP32_INTC_STAT	RESERVED	TC1_STAT	TC0_STAT	RESERVED	PC_STAT	VCOP_STAT	ARP32_STAT

Bits	Field Name	Description	Type	Reset
31:22	RESERVED		R	0x0000
21:20	OCPI_DISC_STAT	OCP Initiator(s) Disconnect status2: 2: One or both initiators are active, no request to disconnect is pending 1: One or both initiators are attempting to disconnect 0: EVEs OCP initiators are disconnected	R	0x0000
19:18	RESERVED		R	0
17:16	ARP32_DISC_STATUS	ARP32 Program/Data Bus Disconnect Status 2: 2: ARP32 program and data busses are active, no request to disconnect is pending 1: ARP32 program and data buses are attempting to disconnect 0: ARP32 program and data buses are disconnected	R	0
15:9	RESERVED		R	0
8	INT_OUT_STAT	Interrupt Output status: 0: No edabled interrupts pending 1: Active (at least one enabled interrupt source is pending in the output reducer)	R	0
7	ARP32_INTC_STAT	Interrupt Controller Status: 0: No enabled interrupts pending 1: Active (at least one enabled interrupt source is pending in the ARP32 interrupt controller)	R	0
6	RESERVED		R	0

Bits	Field Name	Description	Type	Reset
5	TC1_STAT	Transfer Controller1 Status: 0: Idle 1: Active	R	0
4	TC0_STAT	Transfer Controller0 Status: 0: Idle 1: Active	R	0
3	RESERVED		R	0
2	PC_STAT	Program Cache Status: 0: Idle 1: Active (Program cache is either performing prefetch/preload/invalidation, or is servicing a CPU program fetch request (hit or miss)).	R	0
1	VCOP_STAT	VCOP Status : 0: Idle 1: Active (Program execution in progress. Based on inverse of vcop_done. Does not account for activity on VCOP OCP debug interface)	R	0
0	ARP32_STAT	Program Cache Status: 0: Idle 1: Active (based on inverse of arp32_stanby).	R	0

Table 6-41. Register Call Summary for Register EVE_STAT

Embedded Vision Engine (EVE) Subsystem

- ARP32 Disconnect: [0]
- OCP Initiator Disconnect: [1]
- Lock and Unlock Feature: [2]
- EVE Register Summary: [3]

Table 6-42. EVE_DISC_CONFIG

Address Offset	0x0008 0010		
Physical Address	0x4208 0010 0x0208 0010	**Instance**	EVE EVE_DSP
Description	Color 0 noise threshold		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																											OCPI_DISC	RESERVED			ARP32_DISC

Bits	Field Name	Description	Type	Reset
31:5	RESERVED		R	0x0
4	OCPI_DISC	OCP Initiator Disconnect request: Write 1: request for OCP initiator to disconnect and mask write byte enable signals. Writing 0 has no effect Read 0: Disconnect not in progress of has completed. Read 1: Disconnect request in progress.	RW	0x0
3:1	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
0	ARP32_DISC	ARP32 Initiator Disconnect request. Write 1: Request for ARP32 program and data buses to disconnect and mask write byte enable signals. Writing 0 has no effect. Read 1: disconnect request in progress. Read 0: disconnect not in progress or has completed.	RW	0x0

Table 6-43. Register Call Summary for Register EVE_DISC_CONFIG

Embedded Vision Engine (EVE) Subsystem

- Memory Error Detection: [0][1]
- ARP32 Disconnect: [2]
- OCP Initiator Disconnect: [3]
- Lock and Unlock Feature: [4]
- EVE Register Summary: [5]

Table 6-44. EVE_BUS_CONFIG

Address Offset	0x0008 0014		
Physical Address	0x4208 0014 0x0208 0014	**Instance**	EVE EVE_DSP
Description	Color 0 noise threshold		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																		TC1_DBS		RESERVED		TC0_DBS		RESERVED				DBP_ENABLE		MAX_IN_FLIGHT	

Bits	Field Name	Description	Type	Reset
31:14	RESERVED		R	0x0
13:12	TC1_DBS	TC1 default burst size. 00: 16 byte, 01: 32 byte, 10: 64 byte, 11: 128 byte (recommended)	RW	0x3
11:10	RESERVED		R	0x0
9:8	TC0_DBS	TC0 default burst size. 00: 16 byte, 01: 32 byte, 10: 64 byte, 11: 128 byte (recommended)	RW	0x3
7:5	RESERVED		R	
4	DBP_ENABLE	Program Cache Demand Based Prefetch enable: 0: DBP disabled 1: DBP enabled	RW	0x0
3:0	MAX_IN_FLIGHT	Defines maximum number of OCP requests in flight. Can be reduced to limit the peak bandwidth for software direct preload, which in turn may provide advantage to other EVE level (e.g. EDMA) or system-level initiators. 0: Reserved. 1: 1 request in flight allowed. 2: 2 requests in flight allowed.. F: 15 requests in flight allowed.	RW	0x4

Table 6-45. Register Call Summary for Register EVE_BUS_CONFIG

Embedded Vision Engine (EVE) Subsystem

- Demand-Based Prefetch: [0][1]
- Lock and Unlock Feature: [3]
- EVE Register Summary: [4]

Table 6-46. EVE_VCOP_HALT_CONFIG

Address Offset	0x0008 0018		
Physical Address	0x4208 0018 0x0208 0018	**Instance**	EVE EVE_DSP
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																													FORCE_ABORT	MSW_EN	ED_EN

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0
2	FORCE_ABORT	VCOP Force Abort Write: Read always returns 0s Write 0 has no effect. Write 1: issues force_abort command to VCOP (through pulse on vcop force abort input	RW	0x0000
1	MSW_EN	VCOP Memory Seitch Error Halt Enable: 0: Disabled 1: Enabled. VCOP halts on VCOP initiated memory swithc error	RW	0x00
0	ED_EN	VCOP Parity Error Detect Halt Enable: 0: Disabled 1: Enabled. VCOP halts on VCOP initiated parity error.	RW	0x000

Table 6-47. Register Call Summary for Register EVE_VCOP_HALT_CONFIG

Embedded Vision Engine (EVE) Subsystem

- VCOP System Error Halt Conditions: [0][1][2]
- Lock and Unlock Feature: [3]
- EVE Register Summary: [4]

Table 6-48. EVE_MMU_CONFIG

Address Offset	0x0008 001C		
Physical Address	0x4208 001C 0x0208 001C	**Instance**	EVE EVE_DSP
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																			MMU1_ABORT	RESERVED		MMU0_ABORT		RESERVED			MMU1_EN	RESERVED			MMU0_EN

Bits	Field Name	Description	Type	Reset
31:13	RESERVED		R	0x00
12	MMU1_ABORT	Causes the MMU to abort the current operation in case of lockup	RW	0x000
11:9	RESERVED		R	0x0
8	MMU0_ABORT	Causes the MMU to abort the current operation in case of lockup	RW	0x000
7:5	RESERVED		R	0x0
4	MMU1_EN	Clearing this bit disables MMU table lookup and causes accesses to use the non-translated address. This bit defaults to enabled but an identical bit within an MMU configuration register defaults to disabled and must be set after the page tables are programmed for MMU operation	RW	0x0
3:1	RESERVED		R	0x0
0	MMU0_EN	Clearing this bit disables MMU table lookup and causes accesses to use the non-translated address. This bit defaults to enabled but an identical bit within an MMU configuration register defaults to disabled and must be set after the page tables are programmed for MMU operation	RW	0x0

Table 6-49. Register Call Summary for Register EVE_MMU_CONFIG

Embedded Vision Engine (EVE) Subsystem

- Lock and Unlock Feature: [0]
- EVE Register Summary: [1]

Table 6-50. EVE_MEMMAP

Address Offset	0x0008 0020		
Physical Address	0x4208 0020 0x0208 0020	**Instance**	EVE EVE_DSP
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																											LCL_EDMA_ALIAS	RESERVED			VCOP_ALIAS

Bits	Field Name	Description	Type	Reset
31:5	RESERVED		R	0x0
4	LCL_EDMA_ALIAS	0: Local EDMA views full memory map 1: VCOP vies aliased memory map. In this mode VCOP views IBUFLA and IBUFLB at the same address and views IBUFHA and IBUFHB at the same address. In this mode only one of IBUFLA or IBUFLB and IBUFHA or IBUFHB can be owned by the system.Refer to Error: Reference source not found for full truth table. Software must poll for updated value to ensure mode change has taken effect.	RW	0x000
3:1	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
0	VCOP_ALIAS	0: VCOP views full memory map 1: VCOP views Aliased memory map. In this mode / VCOP views IBUFLA and IBUFLB at the same address / and views IBUFHA and IBUFHB at the same address. In this mode / only one of IBUFLA or IBUFLB can be :owned: by VCOP; and only one of IBUFHA or IBUFHB can be :owned: by VCOP. Refer to Error: Reference source not found for full truth table. Software must poll for updated value to ensure mode change has taken effect	RW	0x0000

Table 6-51. Register Call Summary for Register EVE_MEMMAP

Embedded Vision Engine (EVE) Subsystem

- Image Buffers–IBUFLA, IBUFLB, IBUFHA, and IBUFHB: [0]
- Lock and Unlock Feature: [1]
- EVE Memory Map: [2][3]
- VCOP and Local EDMA: IBUF Memory Map Aliasing: [4][5][6][7][8][9]
- ARP32 Write Model – Avoiding Race Conditions: [10][11]
- EVE Register Summary: [12]

Table 6-52. EVE_MSW_CTL

Address Offset	0x0008 0024		
Physical Address	0x4208 0024 0x0208 0024	**Instance**	EVE EVE_DSP
Description	Memory switch control register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED															WBUF	RESERVED				IBUFHB	RESERVED			IBUFLB	RESERVED			IBUFHA	RESERVED			IBUFLA

Bits	Field Name	Description	Type	Reset
31:17	RESERVED		R	0x0
16	WBUF	Working buffer onwership. Value can be modified through direct writes to the memory mapped register address or through ARP32 executing custom instruction __SwitchBuffer(ucst20). 0: System owned 1: VCOP owned	RW	0x0000
15:13	RESERVED		R	0x00
12	IBUFHB	Image buffer high B ownership. Value can be modified through direct writes to the memory mapped register address or through ARP32 executing custom instruction __SwitchBuffer(ucst20).: 0: System owned 1: VCOP owned	RW	0x000
11:9	RESERVED		R	0x0
8	IBUFLB	Image buffer low B ownership. Value can be modified through direct writes to the memory mapped register address or through ARP32 executing custom instruction __SwitchBuffer(ucst20).: 0: System owned 1: VCOP owned	RW	0x0

Bits	Field Name	Description	Type	Reset
7:5	RESERVED		R	0x0
4	IBUFHA	Image buffer high A ownership. Value can be modified through direct writes to the memory mapped register address or through ARP32 executing custom instruction __SwitchBuffer(ucst20).: 0: System owned 1: VCOP owned	RW	0x0
3:1	RESERVED		R	0x0
0	IBUFLA	Image buffer low A ownership. Value can be modified through direct writes to the memory mapped register address or through ARP32 executing custom instruction __SwitchBuffer(ucst20).: 0: System owned 1: VCOP owned	RW	0x0

Table 6-53. Register Call Summary for Register EVE_MSW_CTL

Embedded Vision Engine (EVE) Subsystem

- WBUF: [0][1][2]
- Image Buffers–IBUFLA, IBUFLB, IBUFHA, and IBUFHB: [3][4][5][6][7][8]
- Lock and Unlock Feature: [9]
- VCOP and Local EDMA: IBUF Memory Map Aliasing: [10][11]
- ARP32 Write Model – Avoiding Race Conditions: [12][13]
- EVE Register Summary: [14]

Table 6-54. EVE_MSW_ERR

Address Offset	0x0008 0028		
Physical Address	0x4208 0028 0x0208 0028	**Instance**	EVE EVE_DSP
Description	Memory Switch Error register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								CONNID								RESERVED												SYSERR	DMAERR	VERR	ARP32ERR

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x00
24:16	CONNID	Reads: OCP CONNID value for OCP request that results in buffer ownership error. CONNID[8] = 0 indicates system initiated request. CONNID[8] = 1 indicates internally initiated request. Valid only when SYSERR is set to 1 otherwise returns 0x0. Undefined when no error status bits are set. CONNID = 0x000 through 0x0FF / system initiator CONNID = 0x100 / ARP32 initiated error CONNID = 0x101 / VCOP initiated error CONNID = 0x102 / EDMA TC0 initiated error CONNID = 0x103 / EDMA TC1 initiated error. Cleared through write :1: to ERR bit field	R	0x000
15:4	RESERVED		R	0x0
3	SYSERR	0 - System initiated buffer ownership error not recorded 1 - System initiated buffer ownership error detected/recorded. Write 1 to clear.	WO	0x000

Bits	Field Name	Description	Type	Reset
2	DMAERR	0 : EDMA initiated buffer ownership error not recorded 1 : EDMA buffer ownership error detected/recorded Write 1 to clear.	WO	0x0
1	VERR	0 : VCOP initiated buffer ownership error not recorded 1 : VCOP initiated buffer ownership error detected/recorded Write 1 to clear.	WO	0x0
0	ARP32ERR	0 : ARP32 initiated buffer ownership error not recorded 1 : ARP32 initiated buffer ownership error detected/recorded Write 1 to clear.	WO	0x0

Table 6-55. Register Call Summary for Register EVE_MSW_ERR

Embedded Vision Engine (EVE) Subsystem

- EVE Connection ID (ConnID) Mapping: [0]
- WBUF: [1]
- Image Buffers–IBUFLA, IBUFLB, IBUFHA, and IBUFHB: [2]
- Memory Switch Error Registers: [3]
- VCOP System Error Halt Conditions: [4]
- EVE Interrupt Sources – Memory Switch and Parity Error Interrupts: [5]
- Lock and Unlock Feature: [6]
- EVE Register Summary: [7]

Table 6-56. EVE_MSW_ERRADDR

Address Offset	0x0008 002C		
Physical Address	0x4208 002C 0x0208 002C	**Instance**	EVE EVE_DSP
Description	Memory switch error address register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR																															

Bits	Field Name	Description	Type	Reset
31:0	ADDR	Physical Address (i.e. / not aliased address) of memory switch error. For VCOP accesses / indicates 32-B aligned address. For EDMA or System accesses / indicates 16-B aligned access. For ARP32 / indicates 4-B aligned address. Value is undefined when no ERR bits set.	R	0x0000 0000

Table 6-57. Register Call Summary for Register EVE_MSW_ERRADDR

Embedded Vision Engine (EVE) Subsystem

- WBUF: [0]
- Image Buffers–IBUFLA, IBUFLB, IBUFHA, and IBUFHB: [1]
- Memory Switch Error Registers: [2]
- Lock and Unlock Feature: [3]
- EVE Register Summary: [4]

Table 6-58. EVE_PC_INV

Address Offset	0x0008 0040		
Physical Address	0x4208 0040 0x0208 0040	**Instance**	EVE EVE_DSP
Description	Invalidate all register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	I	Invalidate all: Write 0: no effet. Write 1: initiate invalidate all command Read: 0 : Invalidate operation complete / or not in progress 1 : Invalidate operation still in progress	RW	0x0

Table 6-59. Register Call Summary for Register EVE_PC_INV

Embedded Vision Engine (EVE) Subsystem

- User Coherence Operation: [0]
- Lock and Unlock Feature: [1]
- EVE Register Summary: [2]

Table 6-60. EVE_PC_IBAR

Address Offset	0x0008 0050		
Physical Address	0x4208 0050 0x0208 0050	**Instance**	EVE EVE_DSP
Description	Invalidate Base Address register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR																															

Bits	Field Name	Description	Type	Reset
31:0	ADDR	Invalidate Base Address register. Defines system byte address base to be invalidated from L1P. The entire range to be invalidated is (loosely speaking) from the base address to the base address + byte count. The actual range is inclusive of cache line (32-B aligned) addresses containing the start address and end address	RW	0x0000 0000

Table 6-61. Register Call Summary for Register EVE_PC_IBAR

Embedded Vision Engine (EVE) Subsystem

- User Coherence Operation: [0][1][2][3]
- Lock and Unlock Feature: [4]
- EVE Register Summary: [5]

Table 6-62. EVE_PC_IBC

Address Offset	0x0008 0054		
Physical Address	0x4208 0054 0x0208 0054	**Instance**	EVE EVE_DSP
Description	Invalidate byte count register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																BC															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0
15:0	BC	Invalidate Byte Count register. Defines number of bytes relative to Invalidate Base Address (IBAR.ADDR) to be invalidated from L1P. Maximum of 32k (0x8000). Reads return 0x0 when invalidate range operation is complete.	RW	0x0000

Table 6-63. Register Call Summary for Register EVE_PC_IBC

Embedded Vision Engine (EVE) Subsystem

- User Coherence Operation: [0][1][2]
- Lock and Unlock Feature: [3]
- EVE Register Summary: [4]

Table 6-64. EVE_PC_ISAR

Address Offset	0x0008 0058		
Physical Address	0x4208 0058 0x0208 0058	**Instance**	EVE EVE_DSP
Description	Invalidate single address register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR																															

Bits	Field Name	Description	Type	Reset
31:0	ADDR	Invalidate Single Address register. Defines system byte address (1 line only) to be invalidated from L1P. Reads return 0x0 when the invalidate operation is complete.	RW	0x0000 0000

Table 6-65. Register Call Summary for Register EVE_PC_ISAR

Embedded Vision Engine (EVE) Subsystem

- User Coherence Operation: [0][1][2][3]
- Lock and Unlock Feature: [4]
- EVE Register Summary: [5]
- EVE Register Description: [6]

Table 6-66. EVE_PC_ISAR_DONE

Address Offset	0x0008 005C		
Physical Address	0x4208 005C 0x0208 005C	**Instance**	EVE EVE_DSP
Description	Invalidate single address done register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															DONE

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	DONE	Reads return 0x1 when the invalidate operation is complete. Cleared on the next write to the EVE_PC_ISAR register	R	0x0

Table 6-67. Register Call Summary for Register EVE_PC_ISAR_DONE

Embedded Vision Engine (EVE) Subsystem

- Lock and Unlock Feature: [0]
- EVE Register Summary: [1]

Table 6-68. EVE_PC_PBAR

Address Offset	0x0008 0060		
Physical Address	0x4208 0060 0x0208 0060	**Instance**	EVE EVE_DSP
Description	Program cache preload base address register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR																															

Bits	Field Name	Description	Type	Reset
31:0	ADDR	Preload Base Address register. Defines system byte address base to be Preloaded into L1P. The entire range to be Preloaded is (loosely speaking) from the base address to the base address + byte count. The actual range is inclusive of cache line (32-B aligned) addresses containing the start address and end address	RW	0x0000 0000

Table 6-69. Register Call Summary for Register EVE_PC_PBAR

Embedded Vision Engine (EVE) Subsystem

- Software Direct Preload: [0]
- Lock and Unlock Feature: [1]
- EVE Register Summary: [2]

Table 6-70. EVE_PC_PBC

Address Offset	0x0008 0064		
Physical Address	0x4208 0064 0x0208 0064	**Instance**	EVE EVE_DSP
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:0	BC	Preload Byte Count register. Defines number of bytes relative to Preload Base Address (PBAR.ADDR) to be Preloaded into L1P. Maximum of 32k (0x8000). Reads return 0x0 when Preload range operation is complete	RW	0x0000

Table 6-71. Register Call Summary for Register EVE_PC_PBC

Embedded Vision Engine (EVE) Subsystem

- Software Direct Preload: [0][1]
- Lock and Unlock Feature: [2]
- EVE Register Summary: [3]

Table 6-72. EVE_PMEM_ED_CTL

Address Offset	0x0008 0080		
Physical Address	0x4208 0080 0x0208 0080	**Instance**	EVE EVE_DSP
Description	Program Memory Error Detection Control register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														INV	EN

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x00
1	INV	Read 0: Error detection logic is not inverted. Read 1: Error detection logic is inverted. Writes to memory set parity as normal. Reds from meory return inverse of parity bit. Write 0 to clear, write 1 to set. Must be set when EN is set	RW	0x0
0	EN	Error detection logic enable. Writes update parity, reads check parity 0: Disabled 1: Enabled	RW	0x0

Table 6-73. Register Call Summary for Register EVE_PMEM_ED_CTL

Embedded Vision Engine (EVE) Subsystem

- Memory Error Detection: [0][1]
- EVE Register Summary: [2]

Table 6-74. EVE_PMEM_ED_STAT

Address Offset	0x0008 0084		
Physical Address	0x4208 0084 0x0208 0084	**Instance**	EVE EVE_DSP
Description	Error detection status register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								SYSCONNID								RESERVED												SYSERR	DMAERR	VERR	ARP32ERR

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	SYSCONNID	Reads: OCP CONNID value for OCP request that results in buffer ownership error. CONNID[8] = 0 indicates system initiated request. CONNID[8] = 1 indicates internally initiated request. Valid only when SYSERR is set to 1 otherwise returns 0x0. CONNID = 0x000 through 0x0FF / system initiator CONNID = 0x100 / ARP32 initiated error CONNID = 0x101 / VCOP initiated error CONNID = 0x102 / EDMA TC0 initiated error CONNID = 0x103 / EDMA TC1 initiated error. Cleared through write :1: to *ERR bit field.	RW	0x0000
15:4	RESERVED		R	0x000
3	SYSERR	0 - System initiated parity error not recorded 1 - System initiated parity error detected/recorded. Write 1 to clear.	RW	0x0
2	DMAERR	0 : EDMA initiated parity error not recorded 1 : EDMA parity error detected/recorded Write 1 to clear.	RW	0x0
1	VERR	0 : VCOP initiated parity error not recorded 1 : VCOP initiated parity error detected/recorded Write 1 to clear.	RW	0x0
0	ARP32ERR	0 : ARP32 initiated parity error not recorded 1 : ARP32 initiated parity error detected/recorded Write 1 to clear	RW	0x0

Table 6-75. Register Call Summary for Register EVE_PMEM_ED_STAT

Embedded Vision Engine (EVE) Subsystem

- Memory Error Detection: [0]
- EVE Interrupt Sources – Memory Switch and Parity Error Interrupts: [1]
- EVE Register Summary: [2]

Table 6-76. EVE_PMEM_EDADDR

Address Offset	0x0008 0088		
Physical Address	0x4208 0088 0x0208 0088	**Instance**	EVE EVE_DSP
Description	Program memory error detection address		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR																															

Bits	Field Name	Description	Type	Reset
31:0	ADDR	Physical Address (i.e. / not aliased address) of parity error. For VCOP accesses / indicates 32-B aligned address. For EDMA or System accesses / indicates 16-B aligned access. For ARP32 / indicates 4-B aligned address.	R	0x0

Table 6-77. Register Call Summary for Register EVE_PMEM_EDADDR

Embedded Vision Engine (EVE) Subsystem
- Memory Error Detection: [0][1]
- EVE Register Summary: [2]

Table 6-78. EVE_DMEM_ED_CTL

Address Offset	0x0008 0090		
Physical Address	0x4208 0090 0x0208 0090	**Instance**	EVE EVE_DSP
Description	DMEM error detection control		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														INV	EN

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x00
1	INV	Read 0: Error detection logic is not inverted. Read 1: Error detection logic is inverted. Writes to memory set parity as normal. Reads from meory return inverse of parity bit. Write 0 to clear, write 1 to set. Must be set when EN is set	RW	0x0
0	EN	Error detection logic enable. Writes update parity, reads check parity 0: Disabled 1: Enabled	RW	0x0

Table 6-79. Register Call Summary for Register EVE_DMEM_ED_CTL

Embedded Vision Engine (EVE) Subsystem
- Memory Error Detection: [0][1]
- EVE Register Summary: [2]

Table 6-80. EVE_DMEM_ED_STAT

Address Offset	0x0008 0094		
Physical Address	0x4208 0094 0x0208 0094	**Instance**	EVE EVE_DSP
Description	DMEM error detection status register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								SYSCONNID								RESERVED												SYSERR	DMAERR	VERR	ARP32ERR

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	SYSCONNID	Reads: OCP CONNID value for OCP request that results in buffer ownership error. CONNID[8] = 0 indicates system initiated request. CONNID[8] = 1 indicates internally initiated request. Valid only when SYSERR is set to 1 otherwise returns 0x0. CONNID = 0x000 through 0x0FF / system initiator CONNID = 0x100 / ARP32 initiated error CONNID = 0x101 / VCOP initiated error CONNID = 0x102 / EDMA TC0 initiated error CONNID = 0x103 / EDMA TC1 initiated error. Cleared through write :1: to *ERR bit field.	RW	0x0000
15:4	RESERVED		R	0x000
3	SYSERR	0 - System initiated parity error not recorded 1 - System initiated parity error detected/recorded. Write 1 to clear.	RW	0x0
2	DMAERR	0 : EDMA initiated parity error not recorded 1 : EDMA parity error detected/recorded Write 1 to clear.	RW	0x0
1	VERR	0 : VCOP initiated parity error not recorded 1 : VCOP initiated parity error detected/recorded Write 1 to clear.	RW	0x0
0	ARP32ERR	0 : ARP32 initiated parity error not recorded 1 : ARP32 initiated parity error detected/recorded Write 1 to clear	RW	0x0

Table 6-81. Register Call Summary for Register EVE_DMEM_ED_STAT

Embedded Vision Engine (EVE) Subsystem

- Memory Error Detection: [0]
- EVE Interrupt Sources – Memory Switch and Parity Error Interrupts: [1][2][3]
- EVE Register Summary: [4]

Table 6-82. EVE_DMEM_EDADDR

Address Offset	0x0000 0098		
Physical Address	0x4208 0098 0x0208 0098	**Instance**	EVE EVE_DSP
Description	DMEM error detection address register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR																															

Bits	Field Name	Description	Type	Reset
31:0	ADDR	Physical Address (i.e. / not aliased address) of parity error. For VCOP accesses / indicates 32-B aligned address. For EDMA or System accesses / indicates 16-B aligned access. For ARP32 / indicates 4-B aligned address.	R	0x0

Table 6-83. Register Call Summary for Register EVE_DMEM_EDADDR

Embedded Vision Engine (EVE) Subsystem

- Memory Error Detection: [0][1]
- EVE Register Summary: [2]

Table 6-84. EVE_DMEM_EDADDR_BO

Address Offset	0x0000 009C		
Physical Address	0x4208 009C 0x0208 009C	**Instance**	EVE EVE_DSP
Description	DMEM error detection address byte offset register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR																															

Bits	Field Name	Description	Type	Reset
31:0	BO	Address byte offset for parity error. Indicates that an error has occured in byte offset #n. 0: No error occuredin byte offset #n; 1: Error occured in byte offset #n. Write to clear any of MEM.SYSERR/DMARR/ARP32/or VERR.	R	0x0

Table 6-85. Register Call Summary for Register EVE_DMEM_EDADDR_BO

Embedded Vision Engine (EVE) Subsystem

- Memory Error Detection: [0][1]
- EVE Register Summary: [2]

Table 6-86. EVE_WBUF_ED_CTL

Address Offset	0x0008 00A0		
Physical Address	0x4208 00A0 0x0208 00A0	**Instance**	EVE EVE_DSP
Description	WBUF error detection control		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														INV	EN

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x00
1	INV	Read 0: Error detection logic is not inverted. Read 1: Error detection logic is inverted. Writes to memory set parity as normal. Reads from meory return inverse of parity bit. Write 0 to clea Write 1 to set. Must be set when EN is set	RW	0x0
0	EN	Error detection logic enable. Writes update parity, reads check parity 0: Disabled 1: Enabled	RW	0x0

Table 6-87. Register Call Summary for Register EVE_WBUF_ED_CTL

Embedded Vision Engine (EVE) Subsystem

- Memory Error Detection: [0][1]
- EVE Register Summary: [2]

Table 6-88. EVE_WBUF_ED_STAT

Address Offset	0x0008 00A4		
Physical Address	0x4208 00A4 0x0208 00A4	**Instance**	EVE EVE_DSP
Description	WBUF error detection status register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								SYSCONNID								RESERVED												SYSERR	DMAERR	VERR	ARP32ERR

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	SYSCONNID	Reads: OCP CONNID value for OCP request that results in buffer ownership error. CONNID[8] = 0 indicates system initiated request. CONNID[8] = 1 indicates internally initiated request. Valid only when SYSERR is set to 1 otherwise returns 0x0. CONNID = 0x000 through 0x0FF / system initiator CONNID = 0x100 / ARP32 initiated error CONNID = 0x101 / VCOP initiated error CONNID = 0x102 / EDMA TC0 initiated error CONNID = 0x103 / EDMA TC1 initiated error. Cleared through write :1: to *ERR bit field.	RW	0x0000
15:4	RESERVED		R	0x000
3	SYSERR	0 - System initiated parity error not recorded 1 - System initiated parity error detected/recorded. Write 1 to clear.	RW	0x0
2	DMAERR	0 : EDMA initiated parity error not recorded 1 : EDMA parity error detected/recorded Write 1 to clear.	RW	0x0
1	VERR	0 : VCOP initiated parity error not recorded 1 : VCOP initiated parity error detected/recorded Write 1 to clear.	RW	0x0
0	ARP32ERR	0 : ARP32 initiated parity error not recorded 1 : ARP32 initiated parity error detected/recorded Write 1 to clear	RW	0x0

Table 6-89. Register Call Summary for Register EVE_WBUF_ED_STAT

Embedded Vision Engine (EVE) Subsystem

- Memory Error Detection: [0]
- VCOP System Error Halt Conditions: [1]
- EVE Interrupt Sources – Memory Switch and Parity Error Interrupts: [2][3][4][5]
- EVE Register Summary: [6]

Table 6-90. EVE_WBUF_EDADDR

Address Offset	0x0008 00A8		
Physical Address	0x4208 00A8 0x0208 00A8	**Instance**	EVE EVE_DSP
Description	WBUF error detection address register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR																															

Bits	Field Name	Description	Type	Reset
31:0	ADDR	Physical Address (i.e. / not aliased address) of parity error. For VCOP accesses / indicates 32-B aligned address. For EDMA or System accesses / indicates 16-B aligned access. For ARP32 / indicates 4-B aligned address.	R	0x0

Table 6-91. Register Call Summary for Register EVE_WBUF_EDADDR

Embedded Vision Engine (EVE) Subsystem

- Memory Error Detection: [0][1]
- EVE Register Summary: [2]

Table 6-92. EVE_WBUF_EDADDR_BO

Address Offset	0x0008 00AC		
Physical Address	0x4208 00AC 0x0208 00AC	**Instance**	EVE EVE_DSP
Description	WBUF error detection address byte offset register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR																															

Bits	Field Name	Description	Type	Reset
31:0	BO	Address byte offset for parity error. Indicates that an error has occured in byte offset #n. 0: No error occuredin byte offset #n; 1: Error occured in byte offset #n. Write to clear any of MEM.SYSERR/DMARR/ARP32/or VERR.	R	0x0

Table 6-93. Register Call Summary for Register EVE_WBUF_EDADDR_BO

Embedded Vision Engine (EVE) Subsystem

- Memory Error Detection: [0][1]
- EVE Register Summary: [2]

Table 6-94. EVE_IBUF_ED_CTL

Address Offset	0x0008 00B0		
Physical Address	0x4208 00B0 0x0208 00B0	**Instance**	EVE EVE_DSP
Description	IBUF error detection control		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														INV	EN

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x00
1	INV	Read 0: Error detection logic is not inverted. Read 1: Error detection logic is inverted. Writes to memory set parity as normal. Reads from meory return inverse of parity bit. Write 0 to clear, write 1 to set. Must be set when EN is set	RW	0x0
0	EN	Error detection logic enable. Writes update parity, reads check parity 0: Disabled 1: Enabled	RW	0x0

Table 6-95. Register Call Summary for Register EVE_IBUF_ED_CTL

Embedded Vision Engine (EVE) Subsystem

- Memory Error Detection: [0][1]
- EVE Register Summary: [2]

Table 6-96. EVE_IBUF_ED_STAT

Address Offset	0x0008 00B4		
Physical Address	0x4208 00B4 0x0208 00B4	**Instance**	EVE EVE_DSP
Description	IBUF error detection status register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								SYSCONNID								RESERVED												SYSERR	DMAERR	VERR	ARP32ERR

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	SYSCONNID	Reads: OCP CONNID value for OCP request that results in buffer ownership error. CONNID[8] = 0 indicates system initiated request. CONNID[8] = 1 indicates internally initiated request. Valid only when SYSERR is set to 1 otherwise returns 0x0. CONNID = 0x000 through 0x0FF / system initiator CONNID = 0x100 / ARP32 initiated error CONNID = 0x101 / VCOP initiated error CONNID = 0x102 / EDMA TC0 initiated error CONNID = 0x103 / EDMA TC1 initiated error. Cleared through write :1: to *ERR bit field.	RW	0x0000

Bits	Field Name	Description	Type	Reset
15:4	RESERVED		R	0x000
3	SYSERR	0 - System initiated parity error not recorded 1 - System initiated parity error detected/recorded. Write 1 to clear.	RW	0x0
2	DMAERR	0 : EDMA initiated parity error not recorded 1 : EDMA parity error detected/recorded Write 1 to clear.	RW	0x0
1	VERR	0 : VCOP initiated parity error not recorded 1 : VCOP initiated parity error detected/recorded Write 1 to clear.	RW	0x0
0	ARP32ERR	0 : ARP32 initiated parity error not recorded 1 : ARP32 initiated parity error detected/recorded Write 1 to clear	RW	0x0

Table 6-97. Register Call Summary for Register EVE_IBUF_ED_STAT

Embedded Vision Engine (EVE) Subsystem

- Memory Error Detection: [0]
- VCOP System Error Halt Conditions: [1]
- EVE Interrupt Sources – Memory Switch and Parity Error Interrupts: [2][3][4][5]
- EVE Register Summary: [6]

Table 6-98. EVE_IBUF_EDADDR

Address Offset	0x0008 00B8		
Physical Address	0x4208 00B8 0x0208 00B8	**Instance**	EVE EVE_DSP
Description	IBUF error detection address register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR																															

Bits	Field Name	Description	Type	Reset
31:0	ADDR	Physical Address (i.e. / not aliased address) of parity error. For VCOP accesses / indicates 32-B aligned address. For EDMA or System accesses / indicates 16-B aligned access. For ARP32 / indicates 4-B aligned address.	R	0x0

Table 6-99. Register Call Summary for Register EVE_IBUF_EDADDR

Embedded Vision Engine (EVE) Subsystem

- Memory Error Detection: [0][1]
- EVE Register Summary: [2]

Table 6-100. EVE_IBUF_EDADDR_BO

Address Offset	0x0008 00BC		
Physical Address	0x4208 00BC 0x0208 00BC	**Instance**	EVE EVE_DSP
Description	IBUF error detection address byte offset register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR																															

Bits	Field Name	Description	Type	Reset
31:0	BO	Address byte offset for parity error. Indicates that an error has occured in byte offset #n. 0: No error occuredin byte offset #n; 1: Error occured in byte offset #n. Write to clear any of MEM.SYSERR/DMARR/ARP32/or VERR.	R	0x0

Table 6-101. Register Call Summary for Register EVE_IBUF_EDADDR_BO

Embedded Vision Engine (EVE) Subsystem

- Memory Error Detection: [0][1]
- EVE Register Summary: [2]

Table 6-102. EVE_ED_ARP32_DISC_EN

Address Offset	0x0008 00F8		
Physical Address	0x4208 00F8 0x0208 00F8	**Instance**	EVE EVE_DSP
Description	ARP32 disconnect enable register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																ENABLE															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:0	ENABLE	Disconnect Enable for Event #n 0: Disconnect disabled 1: Disconnect enabled	RW	0x0000

Table 6-103. Register Call Summary for Register EVE_ED_ARP32_DISC_EN

Embedded Vision Engine (EVE) Subsystem

- Memory Error Detection: [0]
- ARP32 Disconnect: [1]
- Lock and Unlock Feature: [2]
- EVE Register Summary: [3]

Table 6-104. EVE_ED_OCPI_DISC_EN

Address Offset	0x0008 00FC		
Physical Address	0x4208 00FC 0x0208 00FC	**Instance**	EVE EVE_DSP
Description	OCP interface disconnect enable register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																ENABLE															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:0	ENABLE	Disconnect Enable for Event #n. 0: Disconnect disabled 1: Disconnect enabled	RW	0x0000

Table 6-105. Register Call Summary for Register EVE_ED_OCPI_DISC_EN

Embedded Vision Engine (EVE) Subsystem

- Memory Error Detection: [0]
- OCP Initiator Disconnect: [1]
- Lock and Unlock Feature: [2]
- EVE Register Summary: [3]

Table 6-106. EVE_MSW_ERR_IRQSTATUS_RAW

Address Offset	0x0008 0110		
Physical Address	0x4208 0110 0x0208 0110	**Instance**	EVE EVE_DSP
Description	Per event memory switch error interrupt status register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												EVENT			

Bits	Field Name	Description	Type	Reset
31:4	RESERVED		R	0x0
3:0	EVENT	settable raw status for event #n Write 0: No action Read 0: No event pending Read 1: Event pending Write 1: Set event (for debug)	RW	0x0

Table 6-107. Register Call Summary for Register EVE_MSW_ERR_IRQSTATUS_RAW

Embedded Vision Engine (EVE) Subsystem

- WBUF: [0]
- Image Buffers–IBUFLA, IBUFLB, IBUFHA, and IBUFHB: [1]
- Memory Switch Error Registers: [2]
- EVE Interrupt Sources – Memory Switch and Parity Error Interrupts: [3][4][5]
- EVE Register Summary: [6]

Table 6-108. EVE_MSW_ERR_IRQSTATUS

Address Offset	0x0008 0114		
Physical Address	0x4208 0114 0x0208 0114	**Instance**	EVE EVE_DSP
Description	Memory switch error interrupt status register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												EVENT			

Bits	Field Name	Description	Type	Reset
31:4	RESERVED		R	0x0
3:0	EVENT	Clearable / enabled status for event #N Write 0: No action Read 0: No (enabled) event pending Read 1: Enabled Event pending Write 1: Clear raw event	RW	0x0

Table 6-109. Register Call Summary for Register EVE_MSW_ERR_IRQSTATUS

Embedded Vision Engine (EVE) Subsystem

- Memory Switch Error Registers: [0]
- EVE Interrupt Sources – Memory Switch and Parity Error Interrupts: [1][2]
- EVE Register Summary: [3]

Table 6-110. EVE_MSW_ERR_IRQENABLE_SET

Address Offset	0x0008 0118		
Physical Address	0x4208 0118 0x0208 0118	**Instance**	EVE EVE_DSP
Description	Memory switch error interrupt enable register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												ENABLE			

Bits	Field Name	Description	Type	Reset
31:4	RESERVED		R	0x0
3:0	ENABLE	Enable for event #n Write 0: No action Read 0: Interrupt disabled Read 1: Interrupt enabled Write 1: Enable interrupt	RW	0x0

Table 6-111. Register Call Summary for Register EVE_MSW_ERR_IRQENABLE_SET

Embedded Vision Engine (EVE) Subsystem

- Memory Switch Error Registers: [0]
- EVE Interrupt Sources – Memory Switch and Parity Error Interrupts: [1]
- EVE Register Summary: [2]

Table 6-112. EVE_MSW_ERR_IRQENABLE_CLR

Address Offset	0x0008 011C		
Physical Address	0x4208 011C 0x0208 011C	**Instance**	EVE EVE_DSP
Description	Memory switch error interrupt clear register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												ENABLE			

Bits	Field Name	Description	Type	Reset
31:4	RESERVED		R	0x0
3:0	ENABLE	Enable for event #n Write 0: No action Read 0: Interrupt disabled Read 1: Interrupt enabled Write 1: Disable interrupt (i.e. / clear ENABLEn bit)	RW	0x0

Table 6-113. Register Call Summary for Register EVE_MSW_ERR_IRQENABLE_CLR

Embedded Vision Engine (EVE) Subsystem

- Memory Switch Error Registers: [0]
- EVE Interrupt Sources – Memory Switch and Parity Error Interrupts: [1]
- EVE Register Summary: [2]

Table 6-114. EVE_ED_LCL_IRQSTATUS_RAW

Address Offset	0x0008 0120		
Physical Address	0x4208 0120 0x0208 0120	**Instance**	EVE EVE_DSP
Description	Per event error detection local interrupt status register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																EVENT															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:0	EVENT	settable raw status for event #n Write 0: No action Read 0: No event pending Read 1: Event pending Write 1: Set event (for debug)	RW	0x0000

Table 6-115. Register Call Summary for Register EVE_ED_LCL_IRQSTATUS_RAW

Embedded Vision Engine (EVE) Subsystem

- EVE Interrupt Sources – Memory Switch and Parity Error Interrupts: [0][1]
- ARP32 INTC: [2]
- EVE Register Summary: [3]

Table 6-116. EVE_ED_LCL_IRQSTATUS

Address Offset	0x0008 0124		
Physical Address	0x4208 0124 0x0208 0124	**Instance**	EVE EVE_DSP
Description	Error detection local interrupt status register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																EVENT															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:0	EVENT	Clearable / enabled status for event #N Write 0: No action Read 0: No (enabled) event pending Read 1: Enabled Event pending Write 1: Clear raw event	RW	0x0000

Table 6-117. Register Call Summary for Register EVE_ED_LCL_IRQSTATUS

Embedded Vision Engine (EVE) Subsystem

- EVE Interrupt Sources – Memory Switch and Parity Error Interrupts: [0]
- ARP32 INTC: [1]
- EVE Register Summary: [2]

Table 6-118. EVE_ED_LCL_IRQENABLE_SET

Address Offset	0x0008 0128		
Physical Address	0x4208 0128 0x0208 0128	**Instance**	EVE EVE_DSP
Description	Error detection local interrupt enable register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																ENABLE															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:0	ENABLE	Enable for event #n Write 0: No action Read 0: Interrupt disabled Read 1: Interrupt enabled Write 1: Enable interrupt	RW	0x0000

Table 6-119. Register Call Summary for Register EVE_ED_LCL_IRQENABLE_SET

Embedded Vision Engine (EVE) Subsystem

- EVE Interrupt Sources – Memory Switch and Parity Error Interrupts: [0][1][2]
- EVE Register Summary: [3]

Table 6-120. EVE_ED_LCL_IRQENABLE_CLR

Address Offset	0x0008 012C		
Physical Address	0x4208 012C 0x0208 012C	**Instance**	EVE EVE_DSP
Description	Error detection local interrupt clear register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																ENABLE															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:0	ENABLE	Enable for event #n Write 0: No action Read 0: Interrupt disabled Read 1: Interrupt enabled Write 1: Disable interrupt (i.e. / clear ENABLEn bit)	RW	0x0000

Table 6-121. Register Call Summary for Register EVE_ED_LCL_IRQENABLE_CLR

Embedded Vision Engine (EVE) Subsystem

- EVE Interrupt Sources – Memory Switch and Parity Error Interrupts: [0]
- EVE Register Summary: [1]

Table 6-122. ARP32_NMI_IRQSTATUS_RAW

Address Offset	0x0008 0200		
Physical Address	0x4208 0200 0x0208 0200	**Instance**	EVE EVE_DSP
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EVENT																															

Bits	Field Name	Description	Type	Reset
31:0	EVENT	settable raw status for event #n Write 0: No action Read 0: No event pending Read 1: Event pending Write 1: Set event (for debug)	RW	0x0000

Table 6-123. Register Call Summary for Register ARP32_NMI_IRQSTATUS_RAW

Embedded Vision Engine (EVE) Subsystem

- ARP32 INTC: [0][1]
- EVE Register Summary: [2]

Table 6-124. ARP32_NMI_IRQSTATUS

Address Offset	0x0008 0204		
Physical Address	0x4208 0204 0x0208 0204	**Instance**	EVE EVE_DSP
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EVENT																															

Bits	Field Name	Description	Type	Reset
31:0	EVENT	Clearable / enabled status for event #N Write 0: No action Read 0: No (enabled) event pending Read 1: Enabled Event pending Write 1: Clear raw event	RW	0x0000

Table 6-125. Register Call Summary for Register ARP32_NMI_IRQSTATUS

Embedded Vision Engine (EVE) Subsystem

- ARP32 INTC: [0]
- EVE Register Summary: [1]

Table 6-126. ARP32_NMI_IRQENABLE_SET

Address Offset	0x0008 0208		
Physical Address	0x4208 0208 0x0208 0208	**Instance**	EVE EVE_DSP
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ENABLE																															

Bits	Field Name	Description	Type	Reset
31:0	ENABLE	Enable for event #n Write 0: No action Read 0: Interrupt disabled Read 1: Interrupt enabled Write 1: Enable interrupt	RW	0x0000

Table 6-127. Register Call Summary for Register ARP32_NMI_IRQENABLE_SET

Embedded Vision Engine (EVE) Subsystem

- EVE Register Summary: [0]

Table 6-128. ARP32_NMI_IRQENABLE_CLR

Address Offset	0x0008 020C		
Physical Address	0x4208 020C 0x0208 020C	**Instance**	EVE EVE_DSP
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ENABLE																															

Bits	Field Name	Description	Type	Reset
31:0	ENABLE	Enable for event #n Write 0: No action Read 0: Interrupt disabled Read 1: Interrupt enabled Write 1: Disable interrupt (i.e. / clear ENABLEn bit)	RW	0x0000

Table 6-129. Register Call Summary for Register ARP32_NMI_IRQENABLE_CLR

Embedded Vision Engine (EVE) Subsystem
- EVE Register Summary: [0]

Table 6-130. ARP32_INTn_IRQSTATUS_RAW

Address Offset	0x0008 01D0 + (0x10*n)		
Physical Address	0x4208 01D0 + (0x10*n) 0x0208 01D0 + (0x10*n)	**Instance**	EVE EVE_DSP
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EVENT																															

Bits	Field Name	Description	Type	Reset
31:0	EVENT	settable raw status for event #n Write 0: No action Read 0: No event pending Read 1: Event pending Write 1: Set event (for debug)	RW	0x0000

Table 6-131. Register Call Summary for Register ARP32_INTn_IRQSTATUS_RAW

Embedded Vision Engine (EVE) Subsystem
- ARP32 INTC: [0][1]
- EVE Register Summary: [2]

Table 6-132. ARP32_INTn_IRQSTATUS

Address Offset	0x0008 01D4 + (0x10*n)		
Physical Address	0x4208 01D4 + (0x10*n) 0x0208 01D4 + (0x10*n)	**Instance**	EVE EVE_DSP
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EVENT																															

Bits	Field Name	Description	Type	Reset
31:0	EVENT	Clearable / enabled status for event #N Write 0: No action Read 0: No (enabled) event pending Read 1: Enabled Event pending Write 1: Clear raw event	RW	0x0000

Table 6-133. Register Call Summary for Register ARP32_INTn_IRQSTATUS

Embedded Vision Engine (EVE) Subsystem
- General-Purpose Inputs/Outputs: [0]
- ARP32 INTC: [1]
- Interrupts: [2][3][4]
- Interrupt Servicing: [5]
- EVE Register Summary: [6]

Table 6-134. ARP32_INTn_IRQENABLE_SET

Address Offset	0x0008 01D8 + (0x10*n)		
Physical Address	0x4208 01D8 + (0x10*n) 0x0208 01D8 + (0x10*n)	**Instance**	EVE EVE_DSP
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ENABLE																															

Bits	Field Name	Description	Type	Reset
31:0	ENABLE	Enable for event #n Write 0: No action Read 0: Interrupt disabled Read 1: Interrupt enabled Write 1: Enable interrupt	RW	0x0000

Table 6-135. Register Call Summary for Register ARP32_INTn_IRQENABLE_SET

Embedded Vision Engine (EVE) Subsystem

- General-Purpose Inputs/Outputs: [0]
- Interrupts: [1]
- EVE Register Summary: [2]

Table 6-136. ARP32_INTn_IRQENABLE_CLR

Address Offset	0x0008 01DC + (0x10*n)		
Physical Address	0x4208 01DC + (0x10*n) 0x0208 01DC + (0x10*n)	**Instance**	EVE EVE_DSP
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ENABLE																															

Bits	Field Name	Description	Type	Reset
31:0	ENABLE	Enable for event #n Write 0: No action Read 0: Interrupt disabled Read 1: Interrupt enabled Write 1: Disable interrupt (i.e. / clear ENABLEn bit)	RW	0x0000

Table 6-137. Register Call Summary for Register ARP32_INTn_IRQENABLE_CLR

Embedded Vision Engine (EVE) Subsystem

- EVE Register Summary: [0]

Table 6-138. ARP32_IRQWAKEEN

Address Offset	0x0000 02FC		
Physical Address	0x4208 02FC 0x0208 02FC	**Instance**	EVE EVE_DSP
Description	Wake enable register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								ENABLE																							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x00
23:0	ENABLE	Wakeup Enable for event EVE_EVT_INT #n 0: Interrupt #n disabled for wakeup 1: Interrupt #n enabled for wakeup	RW	0x00 0000

Table 6-139. Register Call Summary for Register ARP32_IRQWAKEEN

Embedded Vision Engine (EVE) Subsystem

- Extended Duration Sleep: [0][1][2]
- EVE Register Summary: [3]

Table 6-140. MMR_LOCKi

Address Offset	0x0008 0300 + (0x4*i)		
Physical Address	0x4208 0300 + (0x4*i) 0x0208 0300 + (0x4*i)	**Instance**	EVE EVE_DSP
Description	MMR Lock/Unlock register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MMR_LOCK																															

Bits	Field Name	Description	Type	Reset
31:0	MMR_LOCK	Lock/Unlock register for corresponding region	RW	0x0000 0000

Table 6-141. Register Call Summary for Register MMR_LOCKi

Embedded Vision Engine (EVE) Subsystem

- EVE Register Summary: [0]

Table 6-142. MISR_CTL

Address Offset	0x0008 0400		
Physical Address	0x4208 0400 0x0208 0400	**Instance**	EVE EVE_DSP
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														ENABLE	

Bits	Field Name	Description	Type	Reset
31:3	RESERVED			
2:0	ENABLE	MISR Enable #N 0: MISR #n disabled 1: MISR #n enabled Bit 0: MISR 0 - ARP32 PMEM path Bit 1: MISR 1 - ARP32 DMEM path Bit 2: MISR 2 - INTC WBUF path	RW	0x0000

Table 6-143. Register Call Summary for Register MISR_CTL

Embedded Vision Engine (EVE) Subsystem

- Hardware-Assisted Software Self-Test – MISRs: [0]
- EVE Register Summary: [1]

Table 6-144. MISR_CLEAR

Address Offset	0x0008 0404		
Physical Address	0x4208 0404 0x0208 0404	**Instance**	EVE EVE_DSP
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														CLEAR	

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0
2:0	CLEAR	MISR Clear #N Write 0: no effect Write 1: Clear MISR #n Read 0: Previous MISR clear command has completed. Read 1: MISR Clear in progress (this state may never actually be readable)	RW	0x00

Table 6-145. Register Call Summary for Register MISR_CLEAR

Embedded Vision Engine (EVE) Subsystem

- Hardware-Assisted Software Self-Test – MISRs: [0]
- EVE Register Summary: [1]

Table 6-146. MISR0_A

Address Offset	0x0008 0410		
Physical Address	0x4208 0410 0x0208 0410	**Instance**	EVE EVE_DSP
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SIGNATURE																															

Bits	Field Name	Description	Type	Reset
31:0	SIGNATURE	MISR Signature Value Write value to initialize to a desired seed. (only valid when disabled / undefined when enabled). Must be written as a 32-b write w/ all byte enable signals active. Read value returns current signature value.	RW	0x0000 0000

Table 6-147. Register Call Summary for Register MISR0_A

Embedded Vision Engine (EVE) Subsystem

- Hardware-Assisted Software Self-Test – MISRs: [0][1]
- EVE Register Summary: [2]

Table 6-148. MISR0_D

Address Offset	0x0008 0414		
Physical Address	0x4208 0414 0x0208 0414	**Instance**	EVE EVE_DSP
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SIGNATURE																															

Bits	Field Name	Description	Type	Reset
31:0	SIGNATURE	MISR Signature Value Write value to initialize to a desired seed. (only valid when disabled / undefined when enabled). Must be written as a 32-b write w/ all byte enable signals active. Read value returns current signature value.	RW	0x0000 0000

Table 6-149. Register Call Summary for Register MISR0_D

Embedded Vision Engine (EVE) Subsystem

- Hardware-Assisted Software Self-Test – MISRs: [0][1]
- EVE Register Summary: [2]

Table 6-150. MISR1_A

Address Offset	0x0008 0418		
Physical Address	0x4208 0418 0x0208 0418	**Instance**	EVE EVE_DSP
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SIGNATURE																															

Bits	Field Name	Description	Type	Reset
31:0	SIGNATURE	MISR Signature Value Write value to initialize to a desired seed. (only valid when disabled / undefined when enabled). Must be written as a 32-b write w/ all byte enable signals active. Read value returns current signature value.	RW	0x0000 0000

Table 6-151. Register Call Summary for Register MISR1_A

Embedded Vision Engine (EVE) Subsystem

- Hardware-Assisted Software Self-Test – MISRs: [0][1]
- EVE Register Summary: [2]

Table 6-152. MISR1_D

Address Offset	0x0008 041C		
Physical Address	0x4208 041C 0x0208 041C	**Instance**	EVE EVE_DSP
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SIGNATURE																															

Bits	Field Name	Description	Type	Reset
31:0	SIGNATURE	MISR Signature Value Write value to initialize to a desired seed. (only valid when disabled / undefined when enabled). Must be written as a 32-b write w/ all byte enable signals active. Read value returns current signature value.	RW	0x0000 0000

Table 6-153. Register Call Summary for Register MISR1_D

Embedded Vision Engine (EVE) Subsystem

- Hardware-Assisted Software Self-Test – MISRs: [0][1]
- Mapping of MISRs to Different Width Buses: [2]
- EVE Register Summary: [3]

Table 6-154. MISR2_Dk

Address Offset	0x0008 0420 + (0x4*k)		
Physical Address	0x4208 0420 + (0x4*k) 0x0208 0420 + (0x4*k)	**Instance**	EVE EVE_DSP
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SIGNATURE																															

Bits	Field Name	Description	Type	Reset
31:0	SIGNATURE	MISR Signature Value Write value to initialize to a desired seed. (only valid when disabled / undefined when enabled). Must be written as a 32-b write w/ all byte enable signals active. Read value returns current signature value.	RW	0x0000 0000

Table 6-155. Register Call Summary for Register MISR2_Dk

Embedded Vision Engine (EVE) Subsystem

- Hardware-Assisted Software Self-Test – MISRs: [0][1]
- Mapping of MISRs to Different Width Buses: [2]
- EVE Register Summary: [3]

Table 6-156. EVE_IRQ_EOI

Address Offset	0x0008 0500		
Physical Address	0x4208 0500 0x0208 0500	**Instance**	EVE EVE_DSP
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														LINE_NUMBER	

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0
2:0	LINE_NUMBER	Software End Of Interrupt (EOI) control. Write number of interrupt output to force re-evaluation of associated pending interrupts. Refer to Section Error: Reference source not found for EOI mapping. Reads always return 0x0. Write n : EOI for Interrupt output associated w/ EOI #n.	RW	0x0

Table 6-157. Register Call Summary for Register EVE_IRQ_EOI

Embedded Vision Engine (EVE) Subsystem

- EVE Register Summary: [0]

Table 6-158. EVE_ED_OUT_IRQSTATUS_RAW

Address Offset	0x0008 0510		
Physical Address	0x4208 0510 0x0208 0510	**Instance**	EVE EVE_DSP
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																EVENT															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:0	EVENT	Settable raw status for event #n Write 0: No action Read 0: No event pending Read 1: Event pending Write 1: Set event (for debug)	RW	0x0000

Table 6-159. Register Call Summary for Register EVE_ED_OUT_IRQSTATUS_RAW

Embedded Vision Engine (EVE) Subsystem

- EVE Interrupt Sources – Memory Switch and Parity Error Interrupts: [0][1]
- ARP32 INTC: [2]
- Lock and Unlock Feature: [3]
- EVE Register Summary: [4]

Table 6-160. EVE_ED_OUT_IRQSTATUS

Address Offset	0x0008 0514		
Physical Address	0x4208 0514 0x0208 0514	**Instance**	EVE EVE_DSP
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																EVENT															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:0	EVENT	Clearable / enabled status for event #N Write 0: No action Read 0: No (enabled) event pending Read 1: Enabled Event pending Write 1: Clear raw event	RW	0x0000

Table 6-161. Register Call Summary for Register EVE_ED_OUT_IRQSTATUS

Embedded Vision Engine (EVE) Subsystem

- EVE Interrupt Sources – Memory Switch and Parity Error Interrupts: [0]
- ARP32 INTC: [1]
- Lock and Unlock Feature: [2]
- EVE Register Summary: [3]

Table 6-162. EVE_ED_OUT_IRQENABLE_SET

Address Offset	0x0008 0518		
Physical Address	0x4208 0518 0x0208 0518	**Instance**	EVE EVE_DSP
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																ENABLE															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:0	ENABLE	Enable for event #n Write 0: No action Read 0: Interrupt disabled Read 1: Interrupt enabled Write 1: Enable interrupt	RW	0x0000

Table 6-163. Register Call Summary for Register EVE_ED_OUT_IRQENABLE_SET

Embedded Vision Engine (EVE) Subsystem

- EVE Interrupt Sources – Memory Switch and Parity Error Interrupts: [0]
- Lock and Unlock Feature: [1]
- EVE Register Summary: [2]

Table 6-164. EVE_ED_OUT_IRQENABLE_CLR

Address Offset	0x0008 051C		
Physical Address	0x4208 051C 0x0208 051C	**Instance**	EVE EVE_DSP
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																ENABLE															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:0	ENABLE	Enable for event #n Write 0: No action Read 0: Interrupt disabled Read 1: Interrupt enabled Write 1: Disable interrupt (i.e. / clear ENABLEn bit)	RW	0x0000

Table 6-165. Register Call Summary for Register EVE_ED_OUT_IRQENABLE_CLR

Embedded Vision Engine (EVE) Subsystem

- EVE Interrupt Sources – Memory Switch and Parity Error Interrupts: [0]
- Lock and Unlock Feature: [1]
- EVE Register Summary: [2]

Table 6-166. EVE_INTk_OUT_IRQSTATUS_RAW

Address Offset	0x0008 0520 + (0x10*k)		
Physical Address	0x4208 0520 + (0x10*k) 0x0208 0520 + (0x10*k)	**Instance**	EVE EVE_DSP
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EVENT																															

Bits	Field Name	Description	Type	Reset
31:0	EVENT	settable raw status for event #n Write 0: No action Read 0: No event pending Read 1: Event pending Write 1: Set event (for debug)	RW	0x0000 0000

Table 6-167. Register Call Summary for Register EVE_INTk_OUT_IRQSTATUS_RAW

Embedded Vision Engine (EVE) Subsystem

- Lock and Unlock Feature: [0]
- EVE Register Summary: [1]

Table 6-168. EVE_INTk_OUT_IRQSTATUS

Address Offset	0x0008 0524 + (0x10*k)		
Physical Address	0x4208 0524 + (0x10*k) 0x0208 0524 + (0x10*k)	**Instance**	EVE EVE_DSP
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EVENT																															

Bits	Field Name	Description	Type	Reset
31:0	EVENT	Clearable / enabled status for event #N Write 0: No action Read 0: No (enabled) event pending Read 1: Enabled Event pending Write 1: Clear raw event.	RW	0x0000 0000

Table 6-169. Register Call Summary for Register EVE_INTk_OUT_IRQSTATUS

Embedded Vision Engine (EVE) Subsystem

- ARP32 INTC: [0]
- Lock and Unlock Feature: [1]
- EVE Register Summary: [2]

Table 6-170. EVE_INTk_OUT_IRQENABLE_SET

Address Offset	0x0008 0528 + (0x10*k)		
Physical Address	0x4208 0528 + (0x10*k) 0x0208 0528 + (0x10*k)	**Instance**	EVE EVE_DSP
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ENABLE																															

Bits	Field Name	Description	Type	Reset
31:0	ENABLE	Enable for event #n Write 0: No action Read 0: Interrupt disabled Read 1: Interrupt enabled Write 1: Enable interrupt	RW	0x0000 0000

Table 6-171. Register Call Summary for Register EVE_INTk_OUT_IRQENABLE_SET

Embedded Vision Engine (EVE) Subsystem

- Lock and Unlock Feature: [0]
- EVE Register Summary: [1]

Table 6-172. EVE_INTk_OUT_IRQENABLE_CLR

Address Offset	0x0008 052C + (0x10*k)		
Physical Address	0x4208 052C + (0x10*k) 0x0208 052C + (0x10*k)	**Instance**	EVE EVE_DSP
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ENABLE																															

Bits	Field Name	Description	Type	Reset
31:0	ENABLE	Enable for event #n Write 0: No action Read 0: Interrupt disabled Read 1: Interrupt enabled Write 1: Disable interrupt (i.e. / clear ENABLEn bit)	RW	0x0000 0000

Table 6-173. Register Call Summary for Register EVE_INTk_OUT_IRQENABLE_CLR

Embedded Vision Engine (EVE) Subsystem

- ARP32 INTC: [0]
- Lock and Unlock Feature: [1]
- EVE Register Summary: [2]

Table 6-174. ARP32_INTj_IRQSTATUS_RAW

Address Offset	0x0008 0580 + (0x10*j)		
Physical Address	0x4208 0580 + (0x10*j) 0x0208 0580 + (0x10*j)	**Instance**	EVE EVE_DSP
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EVENT																															

Bits	Field Name	Description	Type	Reset
31:0	EVENT	settable raw status for event #n Write 0: No action Read 0: No event pending Read 1: Event pending Write 1: Set event (for debug)	R	0x0000 0000

Table 6-175. Register Call Summary for Register ARP32_INTj_IRQSTATUS_RAW

Embedded Vision Engine (EVE) Subsystem

- EVE Register Summary: [0]

Table 6-176. ARP32_INTj_IRQSTATUS

Address Offset	0x0008 0584 + (0x10*j)		
Physical Address	0x4208 0584 + (0x10*j) 0x0208 0584 + (0x10*j)	**Instance**	EVE EVE_DSP
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EVENT																															

Bits	Field Name	Description	Type	Reset
31:0	EVENT	Clearable / enabled status for event #N Write 0: No action Read 0: No (enabled) event pending Read 1: Enabled Event pending Write 1: Clear raw event	RW	0x0000 0000

Table 6-177. Register Call Summary for Register ARP32_INTj_IRQSTATUS

Embedded Vision Engine (EVE) Subsystem
- EVE Register Summary: [0]

Table 6-178. ARP32_INTj_IRQENABLE_SET

Address Offset	0x00008 0588 + (0x10*j)		
Physical Address	0x4208 0588 + (0x10*j) 0x0208 0588 + (0x10*j)	**Instance**	EVE EVE_DSP
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ENABLE																															

Bits	Field Name	Description	Type	Reset
31:0	ENABLE	Enable for event #n Write 0: No action Read 0: Interrupt disabled Read 1: Interrupt enabled Write 1: Enable interrupt	RW	0x0000

Table 6-179. Register Call Summary for Register ARP32_INTj_IRQENABLE_SET

Embedded Vision Engine (EVE) Subsystem
- EVE Register Summary: [0]

Table 6-180. ARP32_INTj_IRQENABLE_CLR

Address Offset	0x00008 058C + (0x10*j)		
Physical Address	0x4208 058C + (0x10*j) 0x0208 058C + (0x10*j)	**Instance**	EVE EVE_DSP
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ENABLE																															

Bits	Field Name	Description	Type	Reset
31:0	ENABLE	Enable for event #n Write 0: No action Read 0: Interrupt disabled Read 1: Interrupt enabled Write 1: Disable interrupt (i.e. / clear ENABLEn bit)	RW	0x0000

Table 6-181. Register Call Summary for Register ARP32_INTj_IRQENABLE_CLR

Embedded Vision Engine (EVE) Subsystem
- EVE Register Summary: [0]

Table 6-182. ARP32_INT14_IRQSTATUS_RAW

Address Offset	0x0008 0680		
Physical Address	0x4208 0680 0x0208 0680	**Instance**	EVE EVE_DSP
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EVENT																															

Bits	Field Name	Description	Type	Reset
31:0	EVENT	settable raw status for event #n Write 0: No action Read 0: No event pending Read 1: Event pending Write 1: Set event (for debug)	R	0x00

Table 6-183. Register Call Summary for Register ARP32_INT14_IRQSTATUS_RAW

Embedded Vision Engine (EVE) Subsystem
- EVE Register Summary: [0]

Table 6-184. ARP32_INT14_IRQSTATUS

Address Offset	0x0008 0684		
Physical Address	0x4208 0684 0x0208 0684	**Instance**	EVE EVE_DSP
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EVENT																															

Bits	Field Name	Description	Type	Reset
31:0	EVENT	Clearable / enabled status for event #N Write 0: No action Read 0: No (enabled) event pending Read 1: Enabled Event pending Write 1: Clear raw event	RW	0x0000 0000

Table 6-185. Register Call Summary for Register ARP32_INT14_IRQSTATUS

Embedded Vision Engine (EVE) Subsystem
- EVE Register Summary: [0]

Table 6-186. ARP32_INT14_IRQENABLE_SET

Address Offset	0x0008 0688		
Physical Address	0x4208 0688 0x0208 0688	**Instance**	EVE EVE_DSP
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ENABLE																															

Bits	Field Name	Description	Type	Reset
31:0	ENABLE	Enable for event #n Write 0: No action Read 0: Interrupt disabled Read 1: Interrupt enabled Write 1: Enable interrupt	RW	0x0000

Table 6-187. Register Call Summary for Register ARP32_INT14_IRQENABLE_SET

Embedded Vision Engine (EVE) Subsystem
- EVE Register Summary: [0]

Table 6-188. ARP32_INT14_IRQENABLE_CLR

Address Offset	0x0008 068C		
Physical Address	0x4208 068C 0x0208 068C	**Instance**	EVE EVE_DSP
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ENABLE																															

Bits	Field Name	Description	Type	Reset
31:0	ENABLE	Enable for event #n Write 0: No action Read 0: Interrupt disabled Read 1: Interrupt enabled Write 1: Disable interrupt (i.e. / clear ENABLEn bit)	RW	0x0

Table 6-189. Register Call Summary for Register ARP32_INT14_IRQENABLE_CLR

Embedded Vision Engine (EVE) Subsystem
- EVE Register Summary: [0]

Table 6-190. ARP32_INT15_IRQSTATUS_RAW

Address Offset	0x0008 0690		
Physical Address	0x4208 0690 0x0208 0690	**Instance**	EVE EVE_DSP
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EVENT																															

Bits	Field Name	Description	Type	Reset
31:0	EVENT	settable raw status for event #n Write 0: No action Read 0: No event pending Read 1: Event pending Write 1: Set event (for debug)	R	0x00

Table 6-191. Register Call Summary for Register ARP32_INT15_IRQSTATUS_RAW

Embedded Vision Engine (EVE) Subsystem
- EVE Register Summary: [0]

Table 6-192. ARP32_INT15_IRQSTATUS

Address Offset	0x0008 0694		
Physical Address	0x4208 0694 0x0208 0694	**Instance**	EVE EVE_DSP
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EVENT																															

Bits	Field Name	Description	Type	Reset
31:0	EVENT	Clearable / enabled status for event #N Write 0: No action Read 0: No (enabled) event pending Read 1: Enabled Event pending Write 1: Clear raw event	RW	0x0000 0000

Table 6-193. Register Call Summary for Register ARP32_INT15_IRQSTATUS

Embedded Vision Engine (EVE) Subsystem
- EVE Register Summary: [0]

Table 6-194. ARP32_INT15_IRQENABLE_SET

Address Offset	0x0008 0698		
Physical Address	0x4208 0698 0x0208 0698	**Instance**	EVE EVE_DSP
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ENABLE																															

Bits	Field Name	Description	Type	Reset
31:0	ENABLE	Enable for event #n Write 0: No action Read 0: Interrupt disabled Read 1: Interrupt enabled Write 1: Enable interrupt	RW	0x0000

Table 6-195. Register Call Summary for Register ARP32_INT15_IRQENABLE_SET

Embedded Vision Engine (EVE) Subsystem
- EVE Register Summary: [0]

Table 6-196. ARP32_INT15_IRQENABLE_CLR

Address Offset	0x0008 069C		
Physical Address	0x4208 069C 0x0208 069C	**Instance**	EVE EVE_DSP
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ENABLE																															

Bits	Field Name	Description	Type	Reset
31:0	ENABLE	Enable for event #n Write 0: No action Read 0: Interrupt disabled Read 1: Interrupt enabled Write 1: Disable interrupt (i.e. / clear ENABLEn bit)	RW	0x0

Table 6-197. Register Call Summary for Register ARP32_INT15_IRQENABLE_CLR

Embedded Vision Engine (EVE) Subsystem
- EVE Register Summary: [0]

Table 6-198. EVE_GPOUTm

Address Offset	0x0008 0700 + (0x10*m)		
Physical Address	0x4208 0700 + (0x10*m) 0x0208 0700 + (0x10*m)	**Instance**	EVE EVE_DSP
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EVENT																															

Bits	Field Name	Description	Type	Reset
31:0	EVENT	GP Output #n Write 0: Drive GP Output #n low/1 Read 0: GP Output #n is low/0. Write 1: Drive GP Output #n high/1. Read 1: GP Output is high/1.	RW	0x0000 0000

Table 6-199. Register Call Summary for Register EVE_GPOUTm

Embedded Vision Engine (EVE) Subsystem
- General-Purpose Inputs/Outputs: [0][1]
- Lock and Unlock Feature: [2]
- EVE Register Summary: [3]

Table 6-200. EVE_GPOUTm_SET

Address Offset	0x0008 0704 + (0x10*m)		
Physical Address	0x4208 0704 + (0x10*m) 0x0208 0704 + (0x10*m)	**Instance**	EVE EVE_DSP
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EVENT																															

Bits	Field Name	Description	Type	Reset
31:0	EVENT	GP Output #n Write 0: No action Read 0: GP Output #n is low/0. Write 1: Drive GP Output #n high/1. Read 1: GP Output is high/1.	RW	0x0000 0000

Table 6-201. Register Call Summary for Register EVE_GPOUTm_SET

Embedded Vision Engine (EVE) Subsystem

- General-Purpose Inputs/Outputs: [0][1][2]
- Lock and Unlock Feature: [3]
- EVE Register Summary: [4]

Table 6-202. EVE_GPOUTm_CLR

Address Offset	0x0008 0708 + (0x10*m)		
Physical Address	0x4208 0708 + (0x10*m) 0x0208 0708 + (0x10*m)	**Instance**	EVE EVE_DSP
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EVENT																															

Bits	Field Name	Description	Type	Reset
31:0	EVENT	GP Output #n Write 0: Drive GP Output #n low/1 Read 0: GP Output #n is low/0. Write 1: Drive GP Output #n high/1. Read 1: GP Output is high/1.	RW	0x0000 0000

Table 6-203. Register Call Summary for Register EVE_GPOUTm_CLR

Embedded Vision Engine (EVE) Subsystem

- General-Purpose Inputs/Outputs: [0][1]
- EVE Register Summary: [2]

Table 6-204. EVE_GPOUTm_PULSE

Address Offset	0x0008 070C + (0x10*m)		
Physical Address	0x4208 070C + (0x10*m) 0x0208 070C + (0x10*m)	**Instance**	EVE EVE_DSP
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EVENT																															

Bits	Field Name	Description	Type	Reset
31:0	EVENT	GP Output #n Write 0: No action Read 0: GP Output #n is low/0. Write 1: Drive GP Output #n high/1 for four cycles / then drive low/0. Read 1: GP Output #n is high/1. Note: Writing to GPOUT registers when the four cycles for the previous write to GPOUT_PULSE register have not been completed can result in unpredictable output.	RW	0x0000 0000

Table 6-205. Register Call Summary for Register EVE_GPOUTm_PULSE

Embedded Vision Engine (EVE) Subsystem

- General-Purpose Inputs/Outputs: [0][1][2][3][4]
- EVE Register Summary: [5]

Table 6-206. EVE_GPIN0

Address Offset	0x0008 0740		
Physical Address	0x4208 0740 0x0208 0740	**Instance**	EVE EVE_DSP
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EVENT																															

Bits	Field Name	Description	Type	Reset
31:0	EVENT	GP Input #n Read 0: GP Input #n is low/0. Read 1: GP Input is high/1.	R	0x0000 0000

Table 6-207. Register Call Summary for Register EVE_GPIN0

Embedded Vision Engine (EVE) Subsystem

- General-Purpose Inputs/Outputs: [0]
- Lock and Unlock Feature: [1]
- EVE Register Summary: [2]

Table 6-208. EVE_GPIN1

Address Offset	0x0008 0744		
Physical Address	0x4208 0744 0x0208 0744	**Instance**	EVE EVE_DSP
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EVENT																															

Bits	Field Name	Description	Type	Reset
31:0	EVENT	GP Input #n Read 0: GP Input #n is low/0. Read 1: GP Input is high/1.	R	0x0000 0000

Table 6-209. Register Call Summary for Register EVE_GPIN1

Embedded Vision Engine (EVE) Subsystem

- General-Purpose Inputs/Outputs: [0]
- Lock and Unlock Feature: [1]
- EVE Register Summary: [2]

Table 6-210. EVE_CME_DONE_GPOUT

Address Offset	0x0008 0780		
Physical Address	0x4208 0780 0x0208 0780	**Instance**	EVE EVE_DSP
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								EVENT							

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x000
7:0	EVENT	Internal CME Done Output #n. Write 0: Drive Internal CME Done #n low/1 Read 0: Drive Internal CME Done #n is low/0. Write 1: Drive Internal CME Done #n high/1. Read 1: Internal CME Done is high/1.	RW	0x00

Table 6-211. Register Call Summary for Register EVE_CME_DONE_GPOUT

Embedded Vision Engine (EVE) Subsystem

- CME Signaling: [0][1]
- EVE Register Summary: [2]

Table 6-212. EVE_CME_DONE_GPOUT_SET

Address Offset	0x0008 0784		
Physical Address	0x4208 0784 0x0208 0784	**Instance**	EVE EVE_DSP
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								EVENT							

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x000
7:0	EVENT	Internal CME Done #n Write 0: No action Read 0: Internal CME Done #n is low/0. Write 1: Drive Internal CME Done #n high/1. Read 1: Internal CME Done is high/1.	RW	0x00

Table 6-213. Register Call Summary for Register EVE_CME_DONE_GPOUT_SET

Embedded Vision Engine (EVE) Subsystem

- CME Signaling: [0]
- EVE Register Summary: [1]

Table 6-214. EVE_CME_DONE_GPOUT_CLR

Address Offset	0x0008 0788		
Physical Address	0x4208 0788 0x0208 0788	**Instance**	EVE EVE_DSP
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								EVENT							

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x000
7:0	EVENT	Internal CME Done #n Write 0: No action Read 0: Internal CME Done #n is low/0. Write 1: Drive Internal CME Done #n low/0. Read 1: Internal CME Done #n is high/1.	RW	0x00

Table 6-215. Register Call Summary for Register EVE_CME_DONE_GPOUT_CLR

Embedded Vision Engine (EVE) Subsystem

- CME Signaling: [0]
- EVE Register Summary: [1]

Table 6-216. EVE_CME_DONE_GPOUT_PULSE

Address Offset	0x0008 078C		
Physical Address	0x4208 078C 0x0208 078C	**Instance**	EVE EVE_DSP
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								EVENT							

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x000
7:0	EVENT	Internal CME Done #n Write 0: No action Read 0: Internal CME Done #n is low/0. Write 1: Drive Internal CME Done #n high/1 for four cycles / then drive low/0. Read 1: Internal CME Done #n is high/1. Note: Writing to GPOUT registers when the four cycles for the previous write to GPOUT_PULSE register have not been completed can result in unpredictable output.	RW	0x00

Table 6-217. Register Call Summary for Register EVE_CME_DONE_GPOUT_PULSE

Embedded Vision Engine (EVE) Subsystem

- CME Signaling: [0]
- EVE Register Summary: [1]

Table 6-218. EVE_CME_DONE_SEL

Address Offset	0x0008 0790		
Physical Address	0x4208 0790 0x0208 0790	**Instance**	EVE EVE_DSP
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SEL7				SEL6				SEL5				SEL4				SEL3				SEL2				SEL1				SEL0			

Bits	Field Name	Description	Type	Reset
31:28	SEL7	CME Done Output select for Bit #7 (n=7) .	RW	0x0
27:24	SEL6	CME Done Output select for Bit #6 (n=6) .	RW	0x0
23:20	SEL5	CME Done Output select for Bit #5 (n=5).	RW	0x0
19:16	SEL4	CME Done Output select for Bit #4 (n=4) .	RW	0x0
15:12	SEL3	CME Done Output select for Bit #3 (n=3) .	RW	0x0
11:8	SEL2	CME Done Output select for Bit #2 (n=2) .	RW	0x0
7:4	SEL1	CME Done Output select for Bit #1 (n=1) .	RW	0x0

Bits	Field Name	Description	Type	Reset
3:0	SEL0	CME Done Output select for Bit #0 (n=0) 0: Driven by EDMA cc_int0 1: Driven by EDMA cc_int1 2: Driven by EDMA cc_int2 3: Driven by EDMA cc_int3 4: Driven by EDMA cc_int4 5: Driven by EDMA cc_int5 6: Driven by EDMA cc_int6 7: Driven by EDMA cc_int7 8: Driven by EVE_CME_DONE_GPOUTn 9: driven by eve_cme_done_gpout[0+n] (from EVE) 10: driven by eve_cme_done_gpout[8+n] (from EVE2) 11: driven by eve_cme_done_gpout[16+n] (from EVE3) 12: driven by eve_cme_done_gpout[24+n] (from EVE4) 13: driven by eve_cme_done_gpout[32+n] (from EVE5) 14: driven by eve_cme_done_gpout[40+n] (from EVE6) 15: driven by eve_cme_done_gpout[48+n] (from EVE7)	RW	0x0

Table 6-219. Register Call Summary for Register EVE_CME_DONE_SEL

Embedded Vision Engine (EVE) Subsystem

- CME Signaling: [0][1][2]
- EVE Register Summary: [3]

Table 6-220. EVE_CME_DONE_EN

Address Offset	0x0008 794		
Physical Address	0x4208 0794 0x0208 0794	**Instance**	EVE EVE_DSP
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								EN							

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x0000 0000
7:0	EN	EVE CME Done EN #n Write 0: No action Read 0: EVE CME Done #n is disabled. Write 1: Enable EVE CME Done #n event. This allows the status of the CME Done #n to propagate to the output. Read 1: EVE CME Done #n is enabled.	RW	0x0

Table 6-221. Register Call Summary for Register EVE_CME_DONE_EN

Embedded Vision Engine (EVE) Subsystem

- CME Signaling: [0]
- EVE Register Summary: [1]

Table 6-222. EVE_PM_STAT0

Address Offset	0x0008 0FE0		
Physical Address	0x4208 0FE0 0x0208 0FE0	**Instance**	EVE EVE_DSP
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	OCPM1_SCONNECT			RESERVED		OCPM1_MCONNECT		RESERVED	OCPM0_SCONNECT			RESERVED		OCPM0_MCONNECT		RESERVED	OCPS_SCONNECT			RESERVED		OCPS_MCONNECT		RESERVED		MWAIT	MSTANDBY	SWAKEUP	SIDLEACK		SIDLEREQ

Bits	Field Name	Description	Type	Reset
31	RESERVED		R	0x0
30:28	OCPM1_SCONNECT	Readable state of OCP Power management handshake	R	0x0
27:26	RESERVED		R	0x0
25:24	OCPM1_MCONNECT	Readable state of OCP Power management handshake	R	0x0
23	RESERVED		R	0x0
22:20	OCPM0_SCONNECT	Readable state of OCP Power management handshake	R	0x0
19:18	RESERVED		R	0x0
17:16	OCPM0_MCONNECT	Readable state of OCP Power management handshake	R	0x0
15	RESERVED		R	0x0
14:12	OCPS_SCONNECT	Readable state of OCP Power management handshake	R	0x0
11:10	RESERVED		R	0x0
9:8	OCPS_MCONNECT	Readable state of OCP Power management handshake	R	0x0
7:6	RESERVED		R	0x0
5	MWAIT	Readable state of OCP Power management handshake	R	0x0
4	MSTANDBY	Readable state of OCP Power management handshake	R	0x0
3	SWAKEUP	Readable state of OCP Power management handshake	R	0x0
2:1	SIDLEACK	Readable state of OCP Power management handshake	R	0x0
0	SIDLEREQ	Readable state of OCP Power management handshake	R	0x0

Table 6-223. Register Call Summary for Register EVE_PM_STAT0

Embedded Vision Engine (EVE) Subsystem

- EVE Register Summary: [0]

Table 6-224. EVE_PM_STAT1

Address Offset	0x0008 0FE4		
Physical Address	0x4208 0FE4 0x0208 0FE4	**Instance**	EVE EVE_DSP
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								STBY_MDISCACK_OCPM1		STBY_MDISCACK_OCPM0		STBY_MDISCREQ_OCPM1	STBY_MDISCREQ_OCPM0	IDLE_SDISCONNECT_ACK	IDLE_SDISCONNECT_REQ	EVE_IDLE_INTR_DISABLE	TPTC1_MWAIT	TPTC0_MWAIT	EVE_PCACHE_OCP_BUSY	EVE_CONTROL_SIDLEACK	SMSET_SIDLEACK	L2_EVE_SIDLEACK	MMU1_CONFIG_SIDLEACK	MMU1_SIDLEACK	MMU0_CONFIG_SIDLEACK	MMU0_SIDLEACK	SCTM_SIDLEACK	TPCC_SIDLEACK	TPTC1_SIDLEACK	TPTC0_SIDLEACK	SUBMODULE_IDLE_REQ

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:22	STBY_MDISCACK_OCPM1	Readable state of internal power management handshake	R	0x0
21:20	STBY_MDISCACK_OCPM0	Readable state of internal power management handshake	R	0x0
19	STBY_MDISCREQ_OCPM1	Readable state of internal power management handshake	R	0x0
18	STBY_MDISCREQ_OCPM0	Readable state of internal power management handshake	R	0x0
17	IDLE_SDISCONNECT_ACK	Readable state of internal power management handshake	R	0x0
16	IDLE_SDISCONNECT_REQ	Readable state of internal power management handshake	R	0x0
15	EVE_IDLE_INTR_DISABLE	Readable state of internal power management handshake	R	0x0
14	TPTC1_MWAIT	Readable state of internal power management handshake	R	0x0
13	TPTC0_MWAIT	Readable state of internal power management handshake	R	0x0
12	EVE_PCACHE_OCP_BUSY	Readable state of internal power management handshake	R	0x0
11	EVE_CONTROL_SIDLEACK	Readable state of internal power management handshake	R	0x0
10	SMSET_SIDLEACK	Readable state of internal power management handshake	R	0x0
9	L2_EVE_SIDLEACK	Readable state of internal power management handshake	R	0x0
8	MMU1_CONFIG_SIDLEACK	Readable state of internal power management handshake	R	0x0
7	MMU1_SIDLEACK	Readable state of internal power management handshake	R	0x0
6	MMU0_CONFIG_SIDLEACK	Readable state of internal power management handshake	R	0x0
5	MMU0_SIDLEACK	Readable state of internal power management handshake	R	0x0
4	SCTM_SIDLEACK	Readable state of internal power management handshake	R	0x0

Bits	Field Name	Description	Type	Reset
3	TPCC_SIDLEACK	Readable state of internal power management handshake	R	0x0
2	TPTC1_SIDLEACK	Readable state of internal power management handshake	R	0x0
1	TPTC0_SIDLEACK	Readable state of internal power management handshake	R	0x0
0	SUBMODULE_IDLE_REQ			

Table 6-225. Register Call Summary for Register EVE_PM_STAT1

Embedded Vision Engine (EVE) Subsystem
- EVE Register Summary: [0]

Table 6-226. EVE_DBGOUT

Address Offset	0x0008 0FE8		
Physical Address	0x4208 0FE8 0x0208 0FE8	**Instance**	EVE EVE_DSP
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VALUE																								RESERVED				GROUP			

Bits	Field Name	Description	Type	Reset
31:8	VALUE	Read returns state of eve_dbgout bus.	R	0x000
7:4	RESERVED		R	0x0
3:0	GROUP	Debug Group Output control : mux select 0x0 : disabled / all debug outputs driven to 0x0. 0x1 : select output group1 0x2 : select output group2 : 0xN : select output groupN Others - reserved	RW	0x0

Table 6-227. Register Call Summary for Register EVE_DBGOUT

Embedded Vision Engine (EVE) Subsystem
- EVE Register Summary: [0]

Table 6-228. EVE_RSVD0

Address Offset	0x0008 0FF4		
Physical Address	0x4208 0FF4 0x0208 0FF4	**Instance**	EVE EVE_DSP
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VAL																															

Bits	Field Name	Description	Type	Reset
31:0	VAL	Value; This register is reserved for any necessary ECOs that may be required later in the design cycle.	RW	0x0000 0000

Table 6-229. Register Call Summary for Register EVE_RSVD0

Embedded Vision Engine (EVE) Subsystem • EVE Register Summary: [0]

Table 6-230. EVE_RSVD1

Address Offset	0x0008 0FF8		
Physical Address	0x4208 0FF8 0x0208 0FF8	**Instance**	EVE EVE_DSP
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															

Bits	Field Name	Description	Type	Reset
31:0	VAL	Value; This register is reserved for any necessary ECOs that may be required later in the design cycle.	RW	0x0000 0000

Table 6-231. Register Call Summary for Register EVE_RSVD1

Embedded Vision Engine (EVE) Subsystem • EVE Register Summary: [0]

Table 6-232. EVE_TEST

Address Offset	0x0008 0FFC		
Physical Address	0x4208 0FFC 0x0208 0FFC	**Instance**	EVE EVE_DSP
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VAL																															

Bits	Field Name	Description	Type	Reset
31:0	VAL	Value; This register is reserved for any necessary ECOs that may be required later in the design cycle.	RW	0x0000 0000

Table 6-233. Register Call Summary for Register EVE_TEST

Embedded Vision Engine (EVE) Subsystem • EVE Register Summary: [0]

6.1.5.3 EVE_L2_FNOC Registers

6.1.5.3.1 EVE_L2_FNOC Registers Mapping Summary

Table 6-234 summarizes the EVE_L2_FNOC registers.

Table 6-234. EVE_L2_FNOC Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	EVE_L2_FNOC Physical Address
ERRLOGGER_i_ID_COREID (1)	R	32	0x0000 0000 + (0x200*i)	0x0208 A000 + (0x200*i)
ERRLOGGER_i_ID_REVISION ID (1)	R	32	0x0000 0004 + (0x200*i)	0x0208 A004 + (0x200*i)
ERRLOGGER_i_FAULTEN (1)	RW	32	0x0000 0008 + (0x200*i)	0x0208 A008 + (0x200*i)
ERRLOGGER_i_ERRVLD (1)	R	32	0x0000 000C + (0x200*i)	0x0208 A00C + (0x200*i)
ERRLOGGER_i_ERRCLR (1)	RW	32	0x0000 0010 + (0x200*i)	0x0208 A010 + (0x200*i)
ERRLOGGER_i_ERRLOG0 (1)	R	32	0x0000 0014 + (0x200*i)	0x0208 A014 + (0x200*i)
ERRLOGGER_i_ERRLOG1 (1)	R	32	0x0000 0018 + (0x200*i)	0x0208 A018 + (0x200*i)
ERRLOGGER_i_ERRLOG3 (1)	R	32	0x0000 0020 + (0x200*i)	0x0208 A020 + (0x200*i)
ERRLOGGER_i_ERRLOG5 (1)	R	32	0x0000 0028 + (0x200*i)	0x0208 A028 + (0x200*i)
FLAGMUX_i_ID_COREID (1)	R	32	0x0000 0100 + (0x200*i)	0x0208 A100 + (0x200*i)
FLAGMUX_i_ID_REVISIONID (1)	R	32	0x0000 0104 + (0x200*i)	0x0208 A104 + (0x200*i)
FLAGMUX_i_FAULTEN (1)	RW	32	0x0000 0108 + (0x200*i)	0x0208 A108 + (0x200*i)
FLAGMUX_i_FAULTSTATUS (1)	R	32	0x0000 010C + (0x200*i)	0x0208 A10C + (0x200*i)
FLAGMUX_i_FLAGINEN0 (1)	RW	32	0x0000 0110 + (0x200*i)	0x0208 A110 + (0x200*i)
FLAGMUX_i_FLAGINSTATUS 0 (1)	R	32	0x0000 0114 + (0x200*i)	0x0208 A114 + (0x200*i)

(1) i = 0 to 1 for EVE_L2_FNOC

6.1.5.3.2 EVE_L2_FNOC Register Description

Table through describe the L2_FNOC registers in EVE.

Table 6-235. ERRLOGGER_i_ID_COREID

Address Offset	0x0000 0000		
Physical Address	0x0208 A000 + (0x200*i)	**Instance**	EVE_L2_FNOC
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CORECHECKSUM																								CORETYPEID							

Bits	Field Name	Description	Type	Reset
31:8	CORECHECKSUM	Field containing a checksum of the parameters of the IP.	R	0x220210(1)
7:0	CORETYPEID	Field identifying the type of IP.	R	0xd

(1) when i = 1 CHECKSUM reset value is 0x48443C

Table 6-236. Register Call Summary for Register ERRLOGGER_i_ID_COREID

Embedded Vision Engine (EVE) Subsystem

- EVE_L2_FNOC Registers Mapping Summary: [0]

Table 6-237. ERRLOGGER_i_ID_REVISIONID

Address Offset	0x0000 0004		
Physical Address	0x0208 A004 + (0x200*i)	**Instance**	EVE_L2_FNOC
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FLEXNOCID																								USERID							

Bits	Field Name	Description	Type	Reset
31:8	FLEXNOCID	Field containing the build revision of the software used to generate the IP HDL code.	R	0x10167
7:0	USERID	Field containing a user defined value, not used anywhere inside the IP itself.	R	0x0

Table 6-238. Register Call Summary for Register ERRLOGGER_i_ID_REVISIONID

Embedded Vision Engine (EVE) Subsystem

- EVE_L2_FNOC Registers Mapping Summary: [0]

Table 6-239. ERRLOGGER_i_FAULTEN

Address Offset	0x0000 0008		
Physical Address	0x0208 A008 + (0x200*i)	**Instance**	EVE_L2_FNOC
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															FAULTEN

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	FAULTEN	Enable Fault output	RW	0x0

Table 6-240. Register Call Summary for Register ERRLOGGER_i_FAULTEN

Embedded Vision Engine (EVE) Subsystem

- EVE_L2_FNOC Registers Mapping Summary: [0]

Table 6-241. ERRLOGGER_i_ERRVLD

Address Offset	0x0000 000C		
Physical Address	0x0208 A00C + (0x200*i)	**Instance**	EVE_L2_FNOC
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															ERRVLD

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	ERRVLD	Error logged Valid	R	0x0

Table 6-242. Register Call Summary for Register ERRLOGGER_i_ERRVLD

Embedded Vision Engine (EVE) Subsystem
- EVE_L2_FNOC Registers Mapping Summary: [0]

Table 6-243. ERRLOGGER_i_ERRCLR

Address Offset	0x0000 0010		
Physical Address	0x0208 A010 + (0x200*i)	**Instance**	EVE_L2_FNOC
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															ERRCLR

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	ERRCLR	Clr ErrVld status	RW	0x0

Table 6-244. Register Call Summary for Register ERRLOGGER_i_ERRCLR

Embedded Vision Engine (EVE) Subsystem
- EVE_L2_FNOC Registers Mapping Summary: [0]

Table 6-245. ERRLOGGER_i_ERRLOG0

Address Offset	0x0000 0014		
Physical Address	0x0208 A014 + (0x200*i)	**Instance**	EVE_L2_FNOC
Description	Header: Lock, Opcode, Len1, ErrCode values		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FORMAT	RESERVED							LEN1								RESERVED					ERRCODE			RESERVED			OPC				LOCK

Bits	Field Name	Description	Type	Reset
31	FORMAT	Format of ErrLog0 register	R	0x1
30:26	RESERVED		R	0x0
25:16	LEN1	Header: Len1 value	R	0x0
15:11	RESERVED		R	0x0
10:8	ERRCODE	Header: Error Code value	R	0x0
7:5	RESERVED		R	0x0
4:1	OPC	Header: Opcode value	R	0x0
0	LOCK	Header: Lock bit value	R	0x0

Table 6-246. Register Call Summary for Register ERRLOGGER_i_ERRLOG0

Embedded Vision Engine (EVE) Subsystem
- EVE_L2_FNOC Registers Mapping Summary: [0]

Table 6-247. ERRLOGGER_i_ERRLOG1

Address Offset	0x0000 0018		
Physical Address	0x0208 A018 + (0x200*i)	**Instance**	EVE_L2_FNOC
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																ERRLOG1															

Bits	Field Name	Description	Type	Reset
31:14	RESERVED		R	0x0
13:0	ERRLOG1	Header: RouteId lsb value	R	0x0

Table 6-248. Register Call Summary for Register ERRLOGGER_i_ERRLOG1

Embedded Vision Engine (EVE) Subsystem
- EVE_L2_FNOC Registers Mapping Summary: [0]

Table 6-249. ERRLOGGER_i_ERRLOG3

Address Offset	0x0000 0020		
Physical Address	0x0208 A020 + (0x200*i)	**Instance**	EVE_L2_FNOC
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ERRLOG3																															

Bits	Field Name	Description	Type	Reset
31:0	ERRLOG3	Header: Addr lsb value	R	0x0

Table 6-250. Register Call Summary for Register ERRLOGGER_i_ERRLOG3

Embedded Vision Engine (EVE) Subsystem
- EVE_L2_FNOC Registers Mapping Summary: [0]

Table 6-251. ERRLOGGER_i_ERRLOG5

Address Offset	0x0000 0028		
Physical Address	0x0208 A028 + (0x200*i)	**Instance**	EVE_L2_FNOC
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																ERRLOG5															

Bits	Field Name	Description	Type	Reset
31:17	RESERVED		R	0x0
16:0	ERRLOG5	Header: User lsb value	R	0x0

Table 6-252. Register Call Summary for Register ERRLOGGER_i_ERRLOG5

Embedded Vision Engine (EVE) Subsystem

- EVE_L2_FNOC Registers Mapping Summary: [0]

Table 6-253. FLAGMUX_i_ID_COREID

Address Offset	0x0000 0100		
Physical Address	0x0208 A100 + (0x200*i)	**Instance**	EVE_L2_FNOC
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CORECHECKSUM																								CORETYPEID							

Bits	Field Name	Description	Type	Reset
31:8	CORECHECKSUM	Field containing a checksum of the parameters of the IP.	R	0x2589ca (1)
7:0	CORETYPEID	Field identifying the type of IP.	R	0xb

(1) when i = 1 CHECKSUM reset value is 0x57c4a6

Table 6-254. Register Call Summary for Register FLAGMUX_i_ID_COREID

Embedded Vision Engine (EVE) Subsystem

- EVE_L2_FNOC Registers Mapping Summary: [0]

Table 6-255. FLAGMUX_i_ID_REVISIONID

Address Offset	0x0000 0104		
Physical Address	0x0208 A104 + (0x200*i)	**Instance**	EVE_L2_FNOC
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FLEXNOCID																								USERID							

Bits	Field Name	Description	Type	Reset
31:8	FLEXNOCID	Field containing the build revision of the software used to generate the IP HDL code.	R	0x10167
7:0	USERID	Field containing a user defined value, not used anywhere inside the IP itself.	R	0x0

Table 6-256. Register Call Summary for Register FLAGMUX_i_ID_REVISIONID

Embedded Vision Engine (EVE) Subsystem
- EVE_L2_FNOC Registers Mapping Summary: [0]

Table 6-257. FLAGMUX_i_FAULTEN

Address Offset	0x0000 0108		
Physical Address	0x0208 A108 + (0x200*i)	**Instance**	EVE_L2_FNOC
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															FAULTEN

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	FAULTEN	Global Fault Enable register	RW	0x0

Table 6-258. Register Call Summary for Register FLAGMUX_i_FAULTEN

Embedded Vision Engine (EVE) Subsystem
- EVE_L2_FNOC Registers Mapping Summary: [0]

Table 6-259. FLAGMUX_i_FAULTSTATUS

Address Offset	0x0000 010C		
Physical Address	0x0208 A10C + (0x200*i)	**Instance**	EVE_L2_FNOC
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															FAULTSTATUS

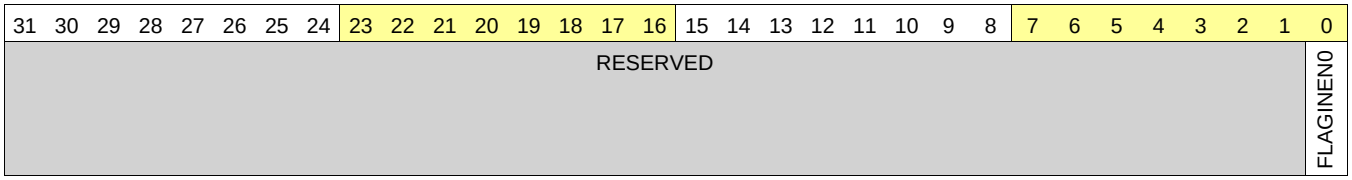
Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	FAULTSTATUS	Global Fault Status register	R	0x0

Table 6-260. Register Call Summary for Register FLAGMUX_i_FAULTSTATUS

Embedded Vision Engine (EVE) Subsystem • EVE_L2_FNOC Registers Mapping Summary: [0]

Table 6-261. FLAGMUX_i_FLAGINEN0

Address Offset	0x0000 0110		
Physical Address	0x0208 A110 + (0x200*i)	**Instance**	EVE_L2_FNOC
Description			
Type	RW		



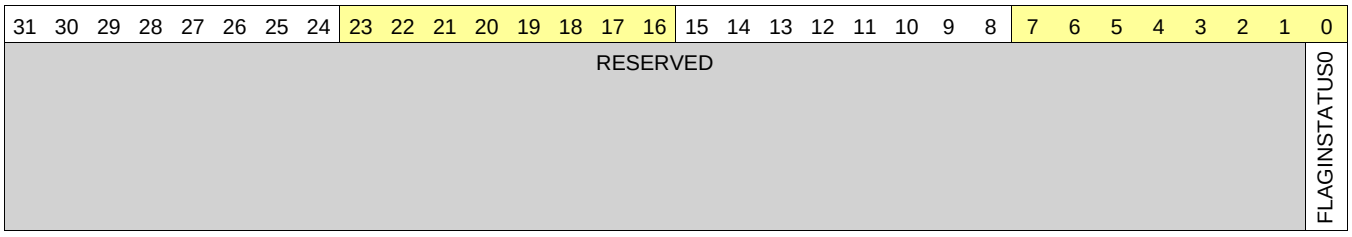
Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	FLAGINEN0	FlagIn Enable register #0	RW	0x0

Table 6-262. Register Call Summary for Register FLAGMUX_i_FLAGINEN0

Embedded Vision Engine (EVE) Subsystem • EVE_L2_FNOC Registers Mapping Summary: [0]

Table 6-263. FLAGMUX_i_FLAGINSTATUS0

Address Offset	0x0000 0114		
Physical Address	0x0208 A114 + (0x200*i)	**Instance**	EVE_L2_FNOC
Description			
Type	R		



Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	FLAGINSTATUS0	FlagIn Status register #0	R	0x0

Table 6-264. Register Call Summary for Register FLAGMUX_i_FLAGINSTATUS0

Embedded Vision Engine (EVE) Subsystem • EVE_L2_FNOC Registers Mapping Summary: [0]

6.1.6 Subsystem Counter Timer Module

6.1.6.1 Introduction

6.1.6.1.1 Overview

As the complexity of SoC architectures increases, system and subsystem optimization requires increased visibility into the frequency and duration of events within the system. One way to address these profiling requirements is to embed dedicated event or profile counters in all functional modules with profiling requirements. While this can address the fundamental requirements, there are several disadvantages to this approach:

- System architects must determine the detailed profiling requirements very early in the specification process.
- These dedicated profiling resources are generally quite similar across functional blocks so the redundant function introduces gate/area penalties in each functional module.
- Embedded profiling resources are harder to optimize out of subsequent system revisions when profiling requirements may be less prevalent.
- The embedded resources may have different configuration and control interfaces and this increases the difficulty of providing unified profiling data from all modules in a subsystem.

This section defines the functional specification for a generic counter timer module that can be instantiated within the processor subsystem and that also functions as a centralized profiling module for the entire subsystem. This module maps a large number of system and subsystem event signals to a smaller number of counter resources. Some of the counter resources in the module can be configured for timer functionality where system and/or debug events can be generated when designated intervals are matched.

In the EVE subsystem eight 32-bit counters are instantiated. Two of those eight counters can be configured as timers.

The generic centralized counter timer approach provides several advantages:

- Unified programmers view of all profiling resources within the subsystem
- Accessible by debug tools and application
- Parameterized synthesis for full configurability of the number of event inputs and counter resources
- A more flexible solution that lets designers tailor gate count to customer profiling requirements
- Standard configuration and control interfaces encourage reuse across subsystem modules within a complex SoC.
- In addition to profiling functions, resources in this module can be used to address normal application counter/timer functions, such as OS timers.

6.1.6.1.2 Top-Level Requirements

The SCTM has the following top-level functional requirements:

- Counter timer resources:
  - A maximum of 32 counters can be instantiated in a single SCTM.
  - A maximum of eight of the counters can be instantiated with timer (event generation) functionality in addition to the base counter function.
- System event signal inputs:
  - A maximum of 127 event signal inputs can be supported by the SCTM.
  - Any of the system event signal inputs can be routed to any of the counter timer resources in the SCTM.
- Counter functionality:
  - The module contains a collection of 32-bit counters that can be chained to an adjacent counter for 64-bit capability.
  - The counters can be configured to operate with any one of the event signal inputs.

  - The counters can be configured to operate in a free-running mode that counts the total number of clock cycles.
  - The counters can operate in duration mode that counts the total duration in cycles that the assigned input signal is asserted.
  - The counters can operate in event mode that counts the total number of times the assigned event signal is asserted.
  - The counters can be configured to continue to run or to temporarily stop when the CPU enters the debug halt state.
  - The counters can be configured to continue to run or to temporarily stop when the CPU enters the IDLE state.
- Timer functionality:
  - Timers have all the functionality of counters.
  - Timers have a 32-bit interval match register.
  - Timers can be configured to generate an interrupt event when the counter matches the interval register.
  - Timers can be configured to generate a debug event when the counter matches the interval register.
  - The timers can be configured to run to the interval register once or to reinitialize each time the interval counter is matched.
- System Trace Interface:
  - The counter state dump can be configured as periodic or may be generated under application control.

6.1.6.1.3 Configuration

Table 6-265 lists the configuration of the SCTM in EVE.

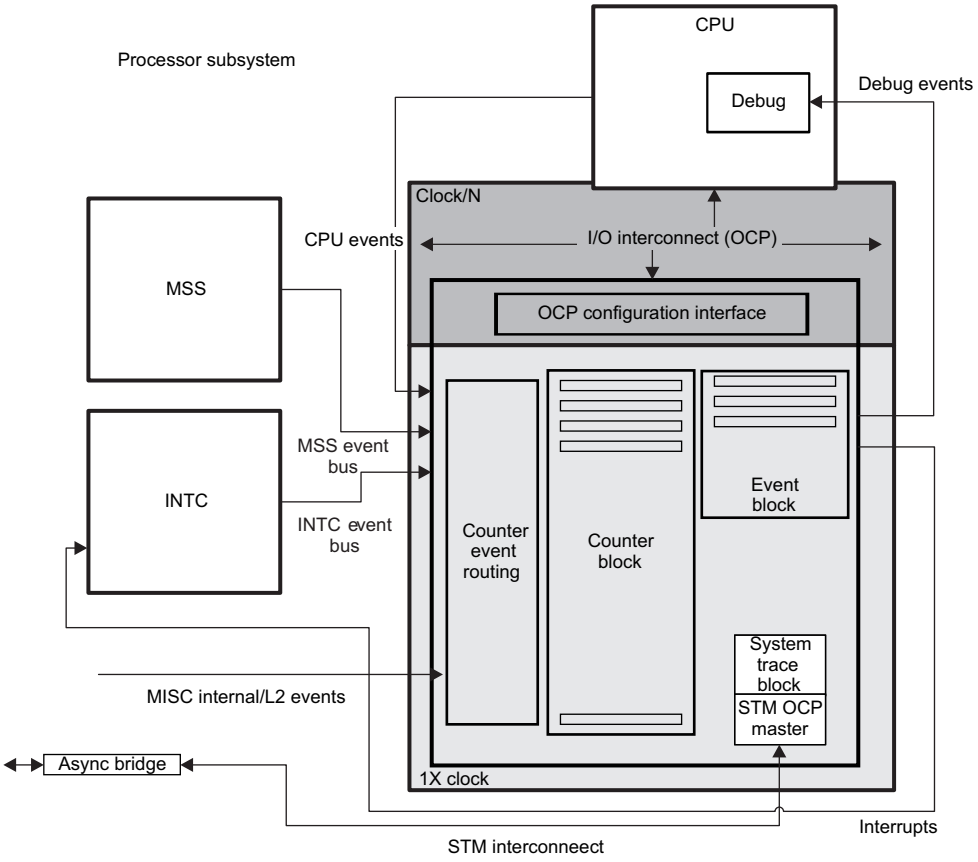
Table 6-265. SCTM Configuration

Generic	SCTM Feature/Parameter Details	Value	Number on Device
CTM_NUMINPT	Number of event input signals supported	0 to 127	31
CTM_NUMCNTR	Number of counters in the module	1 to 32	8
CTM_NUMTIMR	Number of timers	0 to 8	2
CTM_TIMINTPOLARITY	Timer interrupt polarity	0 = Active low 1 = Active high	1
CTM_TIMINTWIDTH	Timer interrupt pulse width	1 or more	2
CTM_NUMSTM	Number of counters for STM export	0-32	0
CTM_CCMAVAIL	CCM frame export available	0 = No 1 = Yes	0
CTM_NUMDBGSGL	Number of debug event signals	0-8	0
CTM_ASYNCIDLEREQ	Whether IDLE request is asynchronous signal	0 = No 1 = Yes	0
CTM_CNTR_CHAINSHADOW	Atomicity feature for even-indexed counters	One per counter. 0 = No chain shadow feature 1 = Chain shadow feature in counter chain mode	Counter numbers 2 and 4 have the chaining feature.

6.1.6.1.4 Block Diagram

Figure 6-18 represents a block diagram for the Counter Timer Module.

Figure 6-18. SCTM Block Diagram



6.1.6.2 Functional Description

The SCTM is a generic profile counter and timer module that can integrate these functions into a single module in a subsystem.

6.1.6.2.1 Configuration Interface

The SCTM has an OCP 2.1-compliant slave interface for configuring and controlling the module. For a detailed description of the register file and the individual registers, see Section 6.1.6.4, SCTM Register Manual. The configuration interface has the following basic functional requirements:

- Any unimplemented configuration memory space:
  - Generates a bus error on application read or write
  - Generates a bus error on debug right
  - Returns 0s and no bus error on debug read
- Any write to a read only register (application or debug) generates a bus error.

6.1.6.2.2 Counter Function

6.1.6.2.2.1 Input Events

Any signal of interest from within the subsystem can be routed to this module and used to control the counters and timers in the module. The routing of the input events from the module boundary to an individual counter is accomplished through an input event multiplexer and is controlled by the INPSEL bit field in the CTCR_WT_j or CTCR_WOT_j register.

The SCTM supports a maximum of 127 input events. The details of the input events for a particular device are detailed in a device-specific appendix to this specification. Application and debug software can determine the number of counters in the SCTM by reading the NUMINPT field in the SCTM_CTCNTL register.

6.1.6.2.2.2 Counters

The individual 32-bit counters in the SCTM count when the input event signals are asserted. The SCTM supports a maximum of 32 counters. The counter configuration for a particular device is detailed in a device-specific appendix to this specification. Application and debug software can determine the number of counters in the SCTM by reading the NUMCNTR field in the SCTM_CTCNTL register.

6.1.6.2.2.3 Counting Mode

Counters function in one of two mutually exclusive counting modes:

- Event mode – The counter increments each time a rising edge is detected on the designated input event signal.
- Duration mode – The counter continually increments when the event input is asserted.

The DURMOD bit in the SCTM_CTCR_WT_j or SCTM_CTCR_WOT_j register controls operating mode of the counter.

6.1.6.2.2.4 Counter Overflow

When the counter reaches the terminal value (FFFF FFFFh) it wraps and continues to increment. This condition is considered a timer overflow condition and the OVRFLW bit in the SCTM_CTCR_WT_j or SCTM_CTCR_WOT_j register indicates that overflow has occurred. The overflow bit can be cleared by reading the SCTM_CTCR_WT_j or SCTM_CTCR_WOT_j register . When chained, only the high-order counter (bits 63:32) can overflow.

6.1.6.2.2.5 Counters and Processor State

The counters can be configured to alter their behavior based on the state of the CPU in the subsystem supported by the SCTM. The FREE bit in the SCTM_CTCR_WT_j or SCTM_CTCR_WOT_j register determines if the counter continues to operate when the processor enters the debug-halt state. When this bit is 0 (the default), the counter stops incrementing while the debug-halt input from the CPU is asserted. Normal operation resumes when the processor exits the debug halt state and the debug-halt input is deasserted. When the FREE bit is 1, the state of the debug-halt input is not used to control counter operation.

The IDLE bit in the SCTM_CTCR_WT_j or SCTM_CTCR_WOT_j register determines if the counter continues to operate when the processor enters the IDLE state (the processor is no longer executing instructions and is waiting for a wakeup event). When this bit is 0 (the default), the counter stops incrementing while the IDLE state input from the CPU is asserted. Normal operation resumes when the processor exits the idle state and the idle state INPUT is deasserted. When the IDLE bit is 1, the state of the idle input does not control counter operation.

Not all subsystems may contain a processor or the subsystem processor may not support debug halt and/or idle modes. In this case, the processor state inputs to the SCTM are tied to the inactive value.

When there are multiple CPUs in a subsystem, the processor state inputs can be sourced from a single processor, or they can be the logical OR of the processor state signals from all CPUs in the subsystem.

6.1.6.2.2.6 Chaining Counters

The individual 32-bit counters in the SCTM can be chained with an adjacent counter to form a 64-bit counter. Counters can be chained to a counter across an even-odd index boundary with the even counter containing the least-significant 32-bits of the 64-bit pairing. For example, when counters 1 and 0 are paired, counter 1 contains bits 63:32 and counter 0 contains bits 31:00. The higher-order counter increments by 1 each time the lower order counter wraps.

Counters are chained by setting the CHAIN bit in the SCTM_CTCR_WT_j or SCTM_CTCR_WOT_j register for both of the counters. When chained, the counter control for both counters is taken from the lower-order SCTM_CTCR_WT_j or SCTM_CTCR_WOT_j register. Other than CHAIN, all the other bits in the higher-order SCTM_CTCR_WT_j or SCTM_CTCR_WOT_j register are ignored.

Chained counters can function in counter mode only. Timer mode is not supported.

6.1.6.2.2.6.1 Reading Chained Counters

Some applications may require atomic reads of the 64 bits of a chained counter pair. An SCTM instance can be configured at synthesis time to provide an atomic read capability for one or more 64-bit counter pairs. When this capability is present, the designated counter pair provides atomic access when the correct read sequence is performed.

So when a chained counter has atomic read capability, an atomic counter value can be obtained by first reading the high-order counter followed by the low-order counter. Always observe this ordering, to prevent reading stale counter values from the low-order counter.

The shadow feature for chained counters is active only when the low-order counter is chained to a high-order counter. When functioning independently, the shadow feature is deactivated and a read of the counter always returns the current value.

To prevent sync errors introduced by the debugger, only application accesses activate the shadow feature. Debug qualified accesses always return the current value of the low-order counter, regardless of configuration.

6.1.6.2.2.7 Enabling and Disabling Counters

After the counter is correctly configured, the counter can be started by setting the ENBL bit in the corresponding SCTM_CTCR_WT_j or SCTM_CTCR_WOT_j register . At this point, the counter begins incrementing under the control of the configured event input. The counter can be disabled (counting stops) at any time by clearing ENBL. Counters can be enabled and disabled dynamically during application flow.

Counters can also be enabled and disabled as groups through the SCTM_CTGNBL0 registers. These registers provide control of the individual counter enables in groups of 32. This allows an application to enable or disable groups of counter in lockstep.

6.1.6.2.2.8 Resetting Counters

The counters can be reset to their initial value (0000 0000h) by setting the RESET bit in the SCTM_CTCR_WT_j or SCTM_CTCR_WOT_j register to 1 . If the counter is chained, both the high-order and low-order counters are reset when the RESET bit is set in the CTCR*n* for the low-order counter.

Counters can also be reset as groups through the SCTM_CTGRSTL0 register. These registers provide control of the individual counter reset in groups of 32. This control letss an application reset groups of counters in lockstep.

6.1.6.2.3 Timer Function

A subset of the counters modules in the SCTM can function as timers. When operating as timers, interrupts and/or debug input events are generated when the counter value reaches a designated interval. The counters with timer capabilities always occupy the lowest indexed counters in the SCTM. A maximum of eight counters can function as timers in SCTM v1. The actual number for a particular implementation of the SCTM is configured at synthesis time. The number of timers in a particular device is detailed in a device-specific appendix to this specification. Application and debug software can determine the total number of counters with timer capabilities reading the NUMTIMR field in the SCTM_CTCNTL register.

6.1.6.2.3.1 Periodic Intervals

The interval for a timer is contained in the SCTM_TINTVLR_i register. There is a SCTM_TINTVLR_i register for every timer capable counter in the SCTM. Timers are initialized to 0. When the corresponding CTCNTRn increments and matches the values designated in SCTM_TINTVLR_i, the timer is considered to be triggered and events configured in SCTM_CTCR_WT_j or SCTM_CTCR_WOT_j are generated.

Timers can function in one of two mutually exclusive modes:

- Run once mode – The timer stops after the first interval match and is not re-enabled until the timer is reset to the initial value (0) by setting the RESET bit in the SCTM_CTCR_WT_j and SCTM_CTCR_WOT_j register to 1.
- Restart mode – The timer automatically resets to the initial value (0) each time the designated interval is reached.

6.1.6.2.3.2 Event Generation

Timers are able to generate both interrupts and debug events. Interrupts are routed from the module boundary to INTC in the subsystem. Debug events are routed as triggers to debug logic within the subsystem. It is assumed that these debug trigger events are merged with other debug resources and used to control debug activity like CPU control or trace.

The generation of the interrupts is controlled by the INT bit in the SCTM_CTCR_WT_j or SCTM_CTCR_WOT_j register. The generation of debug events is controlled by the DBG in the CTCR*n*. Both INT and DBG can be set simultaneously and both signals are generated on interval match.

For debug events, the pulse width is one cycle and the active polarity is high (a high going pulse of one cycle). For interrupts, the polarity and duration of the pulse are synthesizable options that can be tailored to the requirements of the INTC in the subsystem.

If neither the INT nor the DBG bit is set, the timer function is disabled and the counter functions as a regular counter.

The SCTM events include low-level stall and duration signals sourced by various components of the EVE subsystem. The primary categories include program cache-related signals and VCOP-related signals. Table 6-266 summarized the event mapping. Event inputs start numbering at 1, because event0 is reserved and the SCTM internally uses the functional clock as event 0.

The SCTM module operates at the CLK2 rate (nominally 250 to 300 MHz). The event sources include CLK2 and CLK1 (CLK1 = 2 × CLK2) signals. The CLK2 relative signals are connected directly to the SCTM. The CLK1 relative signals are conditioned by EVE level logic to scale from CLK1 to CLK2. Because these are duration type signals, EVE logic asserts a CLK2 pulse for every two CLK1 pulses detected. This results in, at most, one CLK1 cycle of inaccuracy in the CLK2 duration reported by SCTM.

Table 6-266. List of SCTM Events

SCTM Event	Name	Source	Type	SCTM Mode	Clock
1	cache_miss_count	ARP32_Pcache	Pulse	Duration	CLK2
2	cache_hit_count	ARP32_Pcache	Pulse	Duration	CLK2
3	cache_miss_stall	ARP32_Pcache	Duration	Duration or event	CLK2
4	prefetch_compulsory_count	ARP32_Pcache	Pulse	Duration	CLK2
5	Prefetch_lookahead_count	ARP32_Pcache	Pulse	Duration	CLK2
6	prefetch_hit_count	ARP32_Pcache	Pulse	Duration	CLK2
7	line_buffer_hit_count	ARP32_Pcache	Pulse	Duration	CLK2
8	Prefetch_line_count	ARP32_Pcache	Pulse	Duration	CLK2
9	prefetch_discard_stall	ARP32_Pcache	Duration	Duration or event	CLK2
10	tpcc_aet	EDMA	Duration	Duration or event	CLK2
11	arp32_int4	INTC	Duration	Duration or event	CLK2
12	arp32_int5	INTC	Duration	Duration or event	CLK2
13	arp32_int6	INTC	Duration	Duration or event	CLK2
14	arp32_int7	INTC	Duration	Duration or event	CLK2

Table 6-266. List of SCTM Events (continued)

SCTM Event	Name	Source	Type	SCTM Mode	Clock
15	vcop_busy	VCOP	Pulse	Duration	CLK2
16	vcop_idle_and_done	VCOP	Pulse	Duration	CLK2
17	vcop_wait_for_arp32	VCOP	Pulse	Duration	CLK2
18	vcop_arp32_awaits	VCOP	Pulse	Duration	CLK2
19	vcop_overhead	VCOP	Pulse	Duration	CLK1
20	vcop_ld_stall_by_st	VCOP	Pulse	Duration	CLK1
21	vcop_op_stall_by_ldst	VCOP	Pulse	Duration	CLK1
22	vcop_op_stall_by_dependency	VCOP	Pulse	Duration	CLK1
23	vcop_rd_ibufl	VCOP	Pulse	Duration	CLK1
24	vcop_rd_ibufh	VCOP	Pulse	Duration	CLK1
25	vcop_rd_wbuf	VCOP	Pulse	Duration	CLK1
26	vcop_wr_ibufl	VCOP	Pulse	Duration	CLK1
27	vcop_wr_ibufh	VCOP	Pulse	Duration	CLK1
28	vcop_wr_wbuf	VCOP	Pulse	Duration	CLK1
29	vcop_loop_start	VCOP	Edge	Event	CLK2
30	vcop_done	VCOP	Edge	Event	CLK2
31	arp32_nmi	INTC	Duration	Duration or event	CLK2
32	arp32_int8	INTC	Duration	Duration of event	CLK2
33	arp32_int9	INTC	Duration	Duration of event	CLK2
34	arp32_int10	INTC	Duration	Duration of event	CLK2
35	arp32_int11	INTC	Duration	Duration of event	CLK2
36	arp32_int12	INTC	Duration	Duration of event	CLK2
37	arp32_int13	INTC	Duration	Duration of event	CLK2
38	arp32_int14	INTC	Duration	Duration of event	CLK2
39	arp32_int15	INTC	Duration	Duration of event	CLK2

6.1.6.2.3.3 Watchdog Timer Function

A normally configured timer supports a software watchdog capability in which software can continually reset the timer by writing to the SCTM_CTCR_j[1] RESET bit or to the corresponding bit in the SCTM_CTGRST0 register before the interval expires (and the interrupt is generated).

The timers also support a hardware watchdog function where selected input events can start and reset the timers without software interaction.

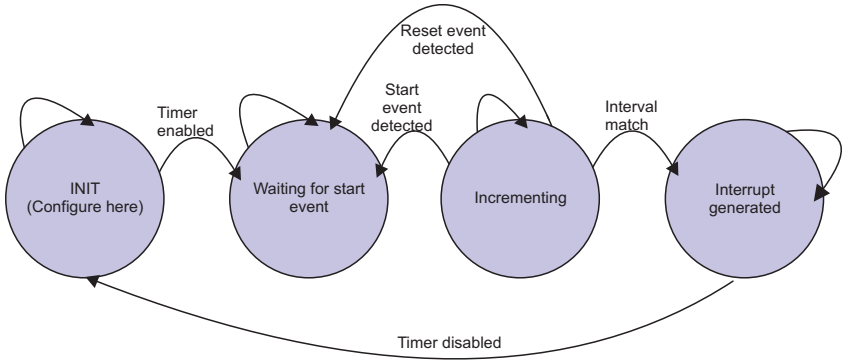
When hardware watchdog mode is enabled, the SCTM_CTCR_WT/WOT_j[23:16] INPSEL bit field selects the event that starts a WD timer. Once the input event is selected, the timer increments using the functional clock. If it reaches the interval match value before being reset, the interrupt is generated.

The WD timer is reset when an input event is detected. This resets the counter to 0. The counter does not restart until another input event is detected. The state-machine shown in Figure 6-19 illustrated the functional operation of the timer when configured for hardware watchdog mode.

Some important notes about WD timer mode operation:

- The INPSEL field selects the WD start event. This is always a rising edge event.
- The DURMODE bit is overridden. Once started, the timer continues to increment using the functional clock.

Figure 6-19. Watchdog Operation



6.1.6.2.4 System Trace Integration

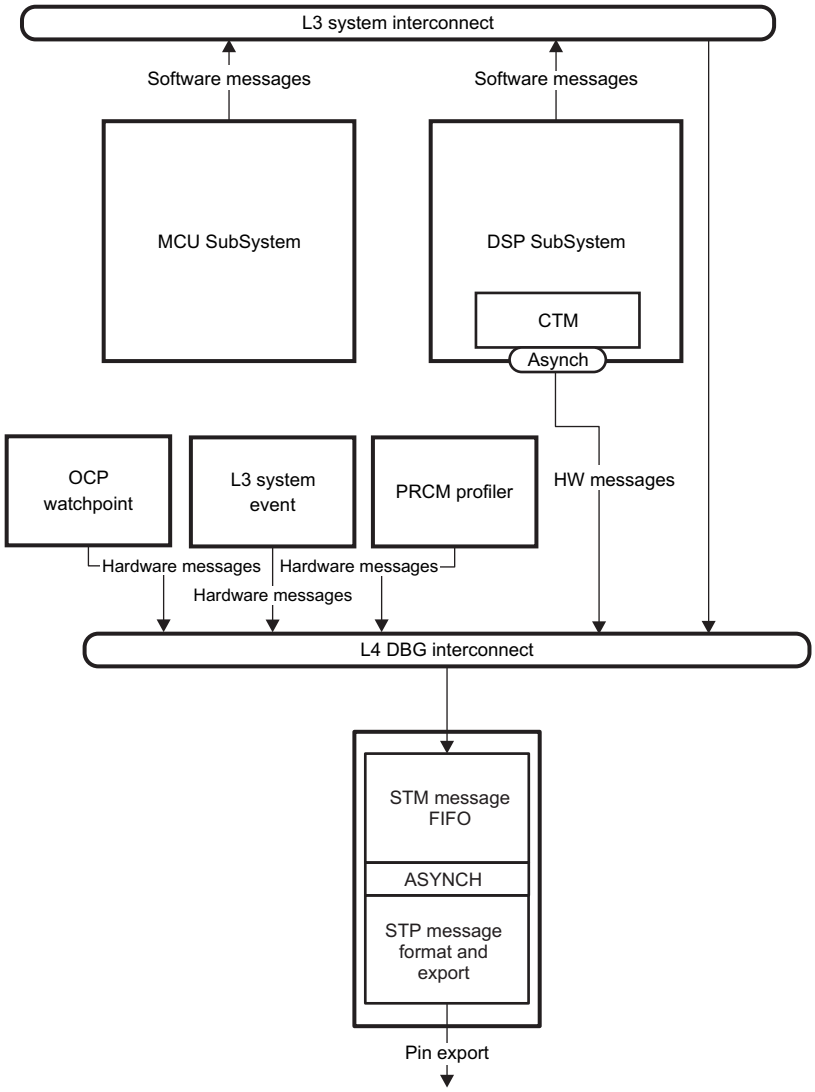
6.1.6.2.4.1 Overview

Many devices have system-level trace capabilities. This capability is generally implemented in a top-level module called the STM. The STM collects software and hardware messages (the trace data) from various other system modules and subsystems through a dedicated STM bus (OCP compliant). The STM then formats these high-level trace messages into streams of MIPI® system trace protocol (STP) compliant messages. The STP messages are then transmitted to a dedicated STP receiver over a narrow (generally four pins), high-speed export interface.

The SCTM has an optional mode that provides an STM OCP interface and a mode where the state of the counters can be transferred through the STM.

Figure 6-20 shows a typical STM system configuration with the SCTM as a hardware master. The hardware master messages from the SCTM STM bus enter the L4 debug interconnect through an asynchronous bridge. Other system-level hardware masters are also writing messages to the STM through this interconnect. Software messages from applications running on the MCU or DSP to reach the STM message buffer through a bridge from L3 interconnect to the L4 interconnect. The STM orders and formats the high-level hardware and software messages into STP protocol messages and exports them to an off-chip trace receiver.

Figure 6-20. Typical STM System Integration



6.1.6.2.4.2 STM Configuration

Because the STM function of the SCTM is an optional feature, the NUMSTM bit in the SCTM_CTCCNTL register is read to determine if any counters are available for STM export:

- If the value in NUMSTM is 0, there is no STM capability in the module.
- If the value is greater than 0 (N ≤ the total counters in the module), then the first N counters can also be designated for STM export.

The STM capability of the SCTM is enabled by setting the ENBL bit in the SCTM_CTSTMCNTL register to 1.

6.1.6.2.4.2.1 Periodic Counter State Export

The general operating mode of the SCTM STM function is to periodically export a message frame that provides a snapshot of the state of each counter designated for inclusion in the STM counter state message. The length of the snapshot interval is controlled by a programmable counter. The interval for this counter is determined by the value in the SCTM_CTSTMINTVL_i register. A periodic counter is reloaded with this value. The counter decrements using the functional clock of the SCTM. When the counter decrements to 0, a snapshot of all the designated counters is taken for STM export and the counter is reloaded using the value in CTSTMINTVL. If the value in CTSTMINTVL is 0, periodic message export is disabled.

Because an entire snapshot of data must be captured when an interval expires, the SCTM must have temporary storage for all the counter values that are designated for export.

6.1.6.2.4.2.2 Application Control of Counter State Export

Application software can also independently trigger the export of a CSM frame bsetting the CSMXPORT bit in the CTSTMCNTL register to 1.

- When the bit is set to 1, a snapshot is captured and a CSM frame is exported.
- Software-triggered export is available only when periodic export is not active; writing 1 when periodic export is active has no effect.
- The SW bit in the CSM header designates that this frame is generated by software.

6.1.6.2.4.2.3 Application Control of the Counter Configuration Export

The STM system also supports an optional counter configuration message (CCM). The capability for generating this message is a synthesizable option. If the module has been synthesized to support CCM, the SCTM_CTSTMCNTL[3] CCMAVAIL bit is asserted.

Export of the CCM message frame is only triggered by setting the CCMXPORT bit in the CTSTMCNTL register to 1.

- When 1 is written, a snapshot is captured and a CCM frame is exported.
- Software-triggered export is available only when periodic export is not active; writing 1 when periodic export is active has no effect.

6.1.6.3 Use Case Examples

This section contains high-level examples of the programming sequences for common SCTM user scenarios. The following examples assume that the counter timer module is already enabled by setting the ENBL bit in the SCTM_CTCNTL register to 1. The examples also assume that the CTCR*n*:ENBL bit is cleared before any configuration writes other than RESET.

6.1.6.3.1 Counter Enable

6.1.6.3.1.1 Enabling a Single Counter

Perform the following procedure to enable a single counter:

1. Reset the counter by setting the SCTM_CTCR_WOT_j[1] RESET bit to 1.
2. Select which signal drives the counter function by writing the correct index to the SCTM_CTCR_WOT_j[23:16]INPSEL bit field
3. Configure the sampling scheme to be used by the counter (edge/level) by writing to the SCTM_CTCR_WOT_j [3]DURMODE bit (if required).
4. Set the behavior for system state by writing to the SCTM_CTCR_WOT_j[5]IDLE and SCTM_CTCR_WOT_j[4] FREE bits (if required).
5. Start the counter function by setting the SCTM_CTCR_WOT_j[0] ENBL bit to 1.

6.1.6.3.1.2 Reading a Single Counter

Perform the following procedure to read a single counter:

1. Read the value of the counter through the CTCNTR_k register.
2. Read the corresponding SCTM_CTCR_WOT_j[6] OVRFLW bit to determine if the counter has wrapped since the last read of the registers.
3. (Optional) The counter can then be reset by setting the SCTM_CTCR_WOT_j[1] RESET bit to 1.

6.1.6.3.1.3 Enabling a Group of Counters Simultaneously

Perform the following procedure to enable a group of counters:

1. For each counter in the group:
   a. Enable access to the SCTM by setting the SCTM_CTCNTL[0] ENBL bit to 1.
   b. Select which signal drives the counter function by writing the correct index to the SCTM_CTCR_WOT_j [23:16] INPSEL bit field.
   c. Configure the sampling scheme to be used by the counter (edge/level) by writing to the SCTM_CTCR_WOT_j [3] DURMODE bit (if required).
   d. Set the behavior for system state by writing to the SCTM_CTCR_WOT_j [5] IDLE and [4]FREE bits (if required).
2. Start all counters in lockstep by writing to the CTGNBL0 register for all counters that are set to 1 in the group.

6.1.6.3.1.4 Reading a Group of Counters Simultaneously

Perform the following procedure to read a group of counters:

1. Read the SCTM_CTGNBL0 register and save.
2. Clear the corresponding enable bit of each counter in the group to 0.
3. Write back the SCTM_CTGNBL0 register content.
4. For each counter in the group:
   a. Read the value of the counter through the SCTM_CTCNTR_k register.
   b. Read the corresponding SCTM_CTCR_WOT_j[6] OVRFLW bit to determine if the counter has wrapped since the last read of the SCTM_CTCR_WOT_j register.
5. (Optional) Resume the group count by writing back the saved value of the SCTM_CTGNBL0 register.

6.1.6.3.1.5 Configuring a Chained Counter

Perform the following procedure to configure a chained counter:

1. Select the counter pair to be used for chaining (N and N + 1).
2. Set the SCTM_CTCR_WOT_j [2]CHAIN bit of counter N + 1 to 1.
3. For the SCTM_CTCR_WOT_j register for counter N in the counter pair:
   a. Select which signal drives the counter function by writing the correct index to the SCTM_CTCR_WOT_j[23:16]INPSEL bit field.
   b. Configure the sampling scheme to be used by the counter (edge/level) by writing to the SCTM_CTCR_WOT_j [3]DURMODE bit (if required).
   c. Set the behavior for system state by writing to the SCTM_CTCR_WOT_j [5]IDLE and SCTM_CTCR_WOT_j [4]FREE bits (if required).
   d. Start the counter function by setting the SCTM_CTCR_WOT_j[0] ENBL bit to 1.

6.1.6.3.2 Timer Enable

This example assumes interrupt generation capabilities. Perform the following procedure to enable the timer:

1. Disable interrupts globally using the CPU INTM mask or some other global interrupt masking in the

target subsystem.

2. Select which signal drives the counter function by writing the correct index to the SCTM_CTCR_WT_j[23:16] INPSEL bit field.
3. Configure the sampling scheme to be used by the counter (edge/level) by writing to the SCTM_CTCR_WT_j[3] DURMODE bit (if required).
4. Set the behavior for system state by writing to the SCTM_CTCR_WT_j [5]IDLE and [4] FREE bits (if required).
5. If the timer function is continuous, set the SCTM_CTCR_j[10] RESTART bit to 1.
6. Set the SCTM_CTCR_WT_j[8] INT bit to 1.
7. Set the interval match value in the corresponding SCTM_TINTVLR_i register.
8. Enable the corresponding interrupt in the subsystem INTC.
9. Enable interrupts globally.
10. Start the counter function by setting the SCTM_CTCR_WT_j[0] ENBL bit to 1.

6.1.6.3.3 Periodic STM Export Enable

Perform the following procedure to enable periodic STM export:

1. Configure, but do not enable, all the counters required for export per Section 6.1.6.3.1, Counter Enable.
2. Enable the STM configuration capability by setting the SCTM_CTSTMCNTL[0] ENBL bit to 1.
3. Disable periodic export by writing 0000 0000h to the SCTM_CTSTMINTVL register.
4. (Optional) Change the hardware master ID by writing the new value to the SCTM_CTSTMMSTID[6:0] MASTID bit.
5. Tag which counters are selected for export by setting the corresponding bits.
6. Set the total number of counters tagged for export through a write to the SCTM_CTSTMCNTL[11:6] NUMXPORT bit.
7. (Optional) To include overflow information in the CSM message, set the SCTM_CTSTMCNTL[1] SENDOVR bit to 1.
8. (Optional) To send a counter configuration message (CCM), set the SCTM_CTSTMCNTL[4] CCMXPORT bit to 1.
   a. This feature is only available if the SCTM_CTSTMCNTL[3]CCMAVAIL bit reads as 1.
   b. Test for completion of the CCM export through the SCTM_CTSTMCNTL[5] XPORTACT bit.
9. Enable periodic export by writing the interval value to the SCTM_CTSTMINTVL register.
10. Enable the counters per Section 6.1.6.3.1, *Counter Enable*.

6.1.6.3.4 Disabling the SCTM

Perform the following procedure to disable the SCTM:

1. When periodic STM export is enabled:
   a. Disable periodic export by writing 0000 0000h to the SCTM_CTSTMINTVL register.
   b. Test for completion of the CCM export through the SCTM_CTSTMCNTL[5] XPORTACT bit.
   c. Disable the STM function by writing 0 to the SCTM_CTSTMCNTL[0]ENBL bit.
2. Disable all counters by setting the SCTM_CTGNBL0 register to 0000 0000h.
3. Disable the SCTM by setting the SCTM_CTCNTL[0] ENBL bit to 0.

6.1.6.4 SCTM Register Manual

6.1.6.4.1 SCTM Instance Summary

Table 6-267. EVE_SCTM Instance Summary

Module Name	Physical Address	Size
EVE_SCTM	0x4208 5000	512 bytes

6.1.6.4.2 SCTM Registers

6.1.6.4.2.1 SCTM Registers Mapping Summary

Table 6-268 lists the SCTM registers

Table 6-268. EVE_SCTM Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	EVE_SCTM Base Address
SCTM_CTCNTL	RW	32	0x0000 0000	0x4208 5000
SCTM_CTSTMCNTL	RW	32	0x0000 0020	0x4208 5020
SCTM_CTSTMMSTID	RW	32	0x0000 0024	0x4208 5024
SCTM_CTSTMINTVL	RW	32	0x0000 0028	0x4208 5028
SCTM_CTSTMSEL	RW	32	0x0000 002C	0x4208 502C
SCTM_TINTVLR_i (1)	RW	32	0x0000 0040 + (0x4*i)	0x4208 5040 + (0x4*i)
SCTM_CTDBGNUM	R	32	0x0000 007C	0x4208 507C
SCTM_CTDBGEVT	RW	32	0x0000 0080	0x4208 5080
SCTM_CTGNBL	RW	32	0x0000 00F0	0x4208 50F0
SCTM_CTGRST	RW	32	0x0000 00F8	0x4208 50F8
SCTM_CTCR_WT_m (2)	RW	32	0x0000 0100 + (0x4*m)	0x4208 5100 + (0x4*m)
SCTM_CTCR_WOT_n (3)	RW	32	0x0000 0108 + (0x4*n)	0x4208 5108 + (0x4*n)
SCTM_CTCNTR_k (4)	R	32	0x0000 0180 + (0x4*k)	0x4208 5180 + (0x4*k)

(1) i = 0 to 7 for EVE_SCTM
(2) m = 0 to 1 for EVE_SCTM
(3) n = 0 to 5 for EVE_SCTM
(4) k = 0 to 7 for EVE_SCTM

6.1.6.4.2.2 SCTM Register Description

Table 6-269. SCTM_CTCNTL

Address Offset	0x0000 0000		
Physical Address	0x4208 5000	**Instance**	EVE_SCTM
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NUMSTM								NUMINPT								NUMTIMR				NUMCNTR				REVID					IDLEMODE		ENBL

Bits	Field Name	Description	Type	Reset
31:26	NUMSTM	Number of timers that can export through STM	R	0x0
25:18	NUMINPT	Number of event input signals	R	0x20
17:13	NUMTIMR	Number of timers in the module	R	0x2
12:7	NUMCNTR	Number of counters in the module	R	0x8
6:3	REVID	Revision ID of SCTM	R	0x1
2:1	IDLEMODE	Idle mode control 0x0: Force Idle mode 0x1: Ths SCTM will acknoledge the idle request, but never transition to the idle state 0x3: Since the SCTM does not support internal wakeup, this mode is identical to smart_idle 0x2: Ths SCTM uses the smart idle protocol. This is the default mode	RW	0x2
0	ENBL	SCTM global enable 0x0: DISABLE 0x1: ENABLE	RW	0x0

Table 6-270. Register Call Summary for Register SCTM_CTCNTL

Embedded Vision Engine (EVE) Subsystem

- Counter Function: [0][1]
- Timer Function: [2]
- Use Case Examples: [3]
- Counter Enable: [4]
- Disabling the SCTM: [5]
- SCTM Registers: [6]

Table 6-271. SCTM_CTSTMCNTL

Address Offset	0x0000 0020		
Physical Address	0x4208 5020	**Instance**	EVE_SCTM
Description	This register contains the control and status settings for STM export		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																					XPORTACT	NUMXPORT						CCMXPORT	CCMVAIL	CSMXPORT	SENDOVR	ENBL

Bits	Field Name	Description	Type	Reset
31:11	RESERVED		R	0x0
10	XPORTACT	Indicates if a frame is currently being written to the STM	R	0x0
9:5	NUMXPORT	The total number of counters designated for export. this will be used as the count in the CSM and CCM headers. The value written should be the total number of counters designated for export -1	RW	0x0
4	CCMXPORT	SW control of CCM message export	RW	0x0
3	CCMVAIL	SCTM supports CCM export	R	0x0
2	CSMXPORT	SW control of CSM message export	RW	0x0
1	SENDOVR	Send overflow data in CSM frame	RW	0x1
0	ENBL	STM global enable 0x0: DISABLE 0x1: ENABLE	RW	0x0

Table 6-272. Register Call Summary for Register SCTM_CTSTMCNTL

Embedded Vision Engine (EVE) Subsystem • System Trace Integration: [0][1] • Periodic STM Export Enable: [2][3][4][5][6][7] • Disabling the SCTM: [8][9] • SCTM Registers: [10]

Table 6-273. SCTM_CTSTMMSTID

Address Offset	0x0000 0024		
Physical Address	0x4208 5024	**Instance**	EVE_SCTM
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								MASTID							

Bits	Field Name	Description	Type	Reset
31:7	RESERVED		R	0x0
6:0	MASTID	HW Master ID for this module.	RW	0x0

Table 6-274. Register Call Summary for Register SCTM_CTSTMMSTID

Embedded Vision Engine (EVE) Subsystem • Periodic STM Export Enable: [0] • SCTM Registers: [1]

Table 6-275. SCTM_CTSTMINTVL

Address Offset	0x0000 0028		
Physical Address	0x4208 5028	**Instance**	EVE_SCTM
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																INTERVAL															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0
15:0	INTERVAL	Periodic export interval	RW	0x0

Table 6-276. Register Call Summary for Register SCTM_CTSTMINTVL

Embedded Vision Engine (EVE) Subsystem • Periodic STM Export Enable: [0][1] • Disabling the SCTM: [2] • SCTM Registers: [3]

Table 6-277. SCTM_CTSTMSEL

Address Offset	0x0000 002C		
Physical Address	0x4208 502C	**Instance**	EVE_SCTM
Description	These registers mark the counters selected for export in the CSM		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
COUNTSEL																															

Bits	Field Name	Description	Type	Reset
31:0	COUNTSEL	The counter selection bit field	RW	0x0

Table 6-278. Register Call Summary for Register SCTM_CTSTMSEL

Embedded Vision Engine (EVE) Subsystem
- SCTM Registers: [0]

Table 6-279. SCTM_TINTVLR_i

Address Offset	0x0000 0040		
Physical Address	0x4208 5040 + (0x4*i)	**Instance**	EVE_SCTM
Description	These registers contain the interval match value for the corresponding timers in the SCTM		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTERVAL																															

Bits	Field Name	Description	Type	Reset
31:0	INTERVAL	Interval match value for the timers in the SCTM	RW	0x0

Table 6-280. Register Call Summary for Register SCTM_TINTVLR_i

Embedded Vision Engine (EVE) Subsystem
- Timer Function: [0][1][2]
- Timer Enable: [3]
- SCTM Registers: [4]

Table 6-281. SCTM_CTDBGNUM

Address Offset	0x0000 007C		
Physical Address	0x4208 507C	**Instance**	EVE_SCTM
Description	Counter Timer Number Debug Event Register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														NUMEVT	

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0
2:0	NUMEVT	Number of input selectors for debug events	R	0x0

Table 6-282. Register Call Summary for Register SCTM_CTDBGNUM

Embedded Vision Engine (EVE) Subsystem • SCTM Registers: [0]

Table 6-283. SCTM_CTDBGEVT

Address Offset	0x0000 0080		
Physical Address	0x4208 5080	**Instance**	EVE_SCTM
Description	Counter Timer Debug Event Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								INPSEL							

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x0
7:0	INPSEL	Index of event input signal on the module boundary	RW	0x0

Table 6-284. Register Call Summary for Register SCTM_CTDBGEVT

Embedded Vision Engine (EVE) Subsystem • SCTM Registers: [0]

Table 6-285. SCTM_CTGNBL

Address Offset	0x0000 00F0		
Physical Address	0x4208 50F0	**Instance**	EVE_SCTM
Description	These registers provide for simultaneous enable/disable of 32 counters		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								ENABLE							

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x0
7:0	ENABLE	The counter enable bit field	RW	0x0

Table 6-286. Register Call Summary for Register SCTM_CTGNBL

Embedded Vision Engine (EVE) Subsystem • Counter Function: [0] • Counter Enable: [1][2][3] • Disabling the SCTM: [4] • SCTM Registers: [5]

Table 6-287. SCTM_CTGRST

Address Offset	0x0000 00F8		
Physical Address	0x4208 50F8	**Instance**	EVE_SCTM
Description	These registers provide for simultaneous reset of 32 counters		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								RESET							

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x0
7:0	RESET	The counter reset bit field	RW	0x0

Table 6-288. Register Call Summary for Register SCTM_CTGRST

Embedded Vision Engine (EVE) Subsystem
- Timer Function: [0]
- SCTM Registers: [1]

Table 6-289. SCTM_CTCR_WT_m

Address Offset	0x0000 0100		
Physical Address	0x4208 5100 + (0x4*m)	**Instance**	EVE_SCTM
Description	These registers contain the control and status settings for a single counter in the module. There will be a CTCR for every counter in the module (WT: with timer)		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								INPSEL								RESERVED					RESTART	DBG	INT	RESERVED	OVRFLW	IDLE	FREE	DURMODE	CHAIN	RESET	ENBL

Bits	Field Name	Description	Type	Reset
31:21	RESERVED		R	0x0
20:16	INPSEL	Counter Timer input selection	RW	0x0
15:11	RESERVED		R	0x0
10	RESTART	Restart the timer after an interval match	RW	0x0
9	DBG	Signal debug logic on interval match	RW	0x0
8	INT	Generate interrupt on interval match	RW	0x0
7	RESERVED		R	0x0
6	OVRFLW	Counter has wrapped since it was last read	R	0x0
5	IDLE	Counter ignores processor IDLE state	RW	0x0
4	FREE	Counter ignores processor debug halt state	RW	0x0
3	DURMODE	Counter is in duration or occurrence mode	RW	0x0
2	CHAIN	Counter is chained to an adjacent counter	RW	0x0
1	RESET	Counter reset control	RW	0x0
0	ENBL	Counter enable control	RW	0x0

Table 6-290. Register Call Summary for Register SCTM_CTCR_WT_m

Embedded Vision Engine (EVE) Subsystem
- SCTM Registers: [0]

Table 6-291. SCTM_CTCR_WOT_n

Address Offset	0x0000 0108		
Physical Address	0x4208 5108 + (0x4*n)	**Instance**	EVE_SCTM
Description	These registers contain the control and status settings for a single counter in the module. There will be a CTCR for every counter in the module (WOT: without timer)		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								INPSEL								RESERVED					RESTART	DBG	INT	RESERVED	OVRFLW	IDLE	FREE	DURMODE	CHAIN	RESET	ENBL

Bits	Field Name	Description	Type	Reset
31:21	RESERVED		R	0x0
20:16	INPSEL	Counter Timer input selection	RW	0x0
15:11	RESERVED		R	0x0
10	RESTART	Restart the timer after an interval match	RW	0x0
9	DBG	Signal debug logic on interval match	RW	0x0
8	INT	Generate interrupt on interval match	RW	0x0
7	RESERVED		R	0x0
6	OVRFLW	Counter has wrapped since it was last read	R	0x0
5	IDLE	Counter ignores processor IDLE state	RW	0x0
4	FREE	Counter ignores processor debug halt state	RW	0x0
3	DURMODE	Counter is in duration or occurrence mode	RW	0x0
2	CHAIN	Counter is chained to an adjacent counter	RW	0x0
1	RESET	Counter reset control	RW	0x0
0	ENBL	Counter enable control	RW	0x0

Table 6-292. Register Call Summary for Register SCTM_CTCR_WOT_n

Embedded Vision Engine (EVE) Subsystem

- SCTM Registers: [0]

Table 6-293. SCTM_CTCNTR_k

Address Offset	0x0000 0180		
Physical Address	0x4208 5180 + (0x4*k)	**Instance**	EVE_SCTM
Description	These registers contain the value of an individual counter in the moduel. There will be a CTCNTR for every counter in the module		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
COUNT																															

Bits	Field Name	Description	Type	Reset
31:0	COUNT	Counter value	R	0x0

Table 6-294. Register Call Summary for Register SCTM_CTCNTR_k

Embedded Vision Engine (EVE) Subsystem • Counter Enable: [0] • SCTM Registers: [1]

6.1.7 Software Message and System Event Trace

6.1.7.1 Introduction

6.1.7.1.1 Overview

The SMSET is a trace module used for monitoring key system events and transferring software messages. SMSET is performing arbitration between hardware messages and software messages written to the trace receiver through STM by the host processor. SMSET module provides various debug features such as: monitor system events, enable/disable event detection, Start/Stop event monitoring upon external trigger, time sampling, generating hardware message upon system event or every sampling window boundary and interleaving of hardware and software messages.

6.1.7.1.2 Configuration

Table 6-295 lists the configuration of the SMSET in the EVE subsystem.

Table 6-295. SMSET Configuration

Generic	SMSET Feature	Number on Device
NB_EVENTS	Number of system events	9
SW_MESSAGE	Software messages support	1
TRIG_MIN_WIDTH	Minimum trigger input pulse (L4 cycles)	2
EVENTS_BUF_DEPTH	System events buffer depth	4
SW_BUF_DEPTH	Software message buffer depth	2

6.1.7.1.3 Block Diagram

The block diagram shown in Figure 6-21 is a generic example to show the main concept of SMSET. Software messages are initiated by processors. Basically, a processor is writing a message to the trace receiver through STM. SMSET is performing arbitration between hardware messages and software messages.

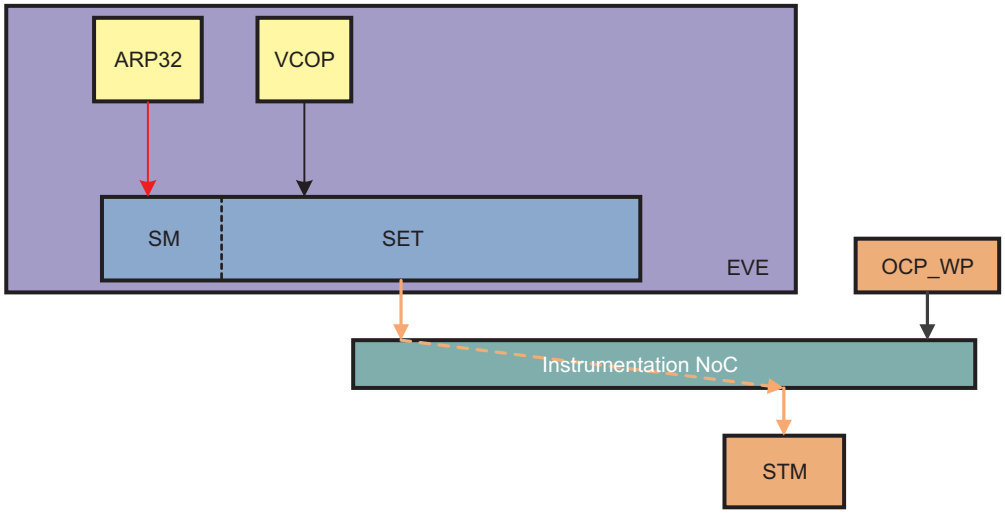
Software messages and system event messages can be interleaved. Software messages do not require a sequence of write accesses; therefore, the SMSET is can switch to the system event detection as soon as the event is detected.

In case of conflicting accesses between software message and detection of system events, the SMSET can switch to the system event detection as soon as the event is detected. Software messages are never lost. Therefore, an SMSET buffer allows absorbing instrumentation accesses peaks, which prevents stalling the processor originating the message.

In case of a platform implementing several SMSET instances, the priority is arbitrated by the instrumentation NoC at SoC level.

It is possible to program the sampling window size to exchange STM bandwidth for time-stamping accuracy. All the events detected within the sampling window are considered close enough and packed in the same message as concurrent events and therefore reported in the trace log with the same time stamp. In case a selected event has been captured but not yet written to the STM queue, the system event trace module shall not wait for the sampling window boundary to write the pending packet to the STM. This results in two separate messages reporting the first and second pulse of the selected event. System events are considered synchronous to the SMSET functional clock. The sampling window size granularity is one SMSET cycle. The size ranges from 1 to 256 cycles.

Figure 6-21. SMSET Block Diagram



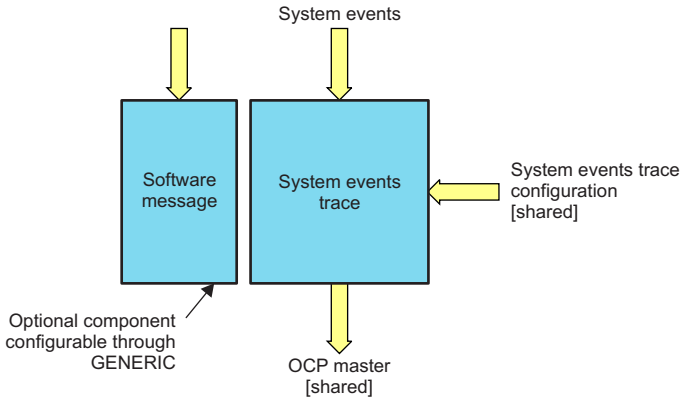
6.1.7.2 Functional Description

6.1.7.2.1 Connectivity

The SMSET component is interfaced to the following:

- Configuration OCP slave port to configure the SMSET parameters
- System events [x N] for monitoring
- Processors supporting software messages (through OCP interface)
- Instrumentation OCP master port to write hardware and software messages into the STM queue

Figure 6-22. SMSET Interfaces



6.1.7.2.2 SMSET Event Mapping

The SMSET events include higher-level start and end task events. Table 6-296 summarize the event mapping.

The SMSET module operates at the CLK2 rate.

SMSET traces events on either a high-to-low transition or low-to-high transition, but not both. Thus, the EDMA tpcc_aet signal is not directly compatible with SMSET tracing. The tpcc_aet is an active high signal that can be used to measure the duration of a specific edma transfer. The tpcc_aet signal can be programmed (in EDMA CC MMRs) to go high when a specific trigger is detected, and go low when a specific completion code is received. EVE logic will implement a custom tpcc_aet_start and tpcc_aet_end signals. The tpcc_aet_start signal will pulse high (for one cycle) when tpcc_aet transitions to high, and tpcc_aet_end signal will pulse high (for one cycle) when tpcc_aet_end signal when tpcc_aet transitions to low.

Table 6-296 details the various profiling signals.

The “pulse” signals are expected to be driven active one cycle for each occurrence. A sequence of consecutive active cycles represent multiple occurrences. "Duration" mode of the SMSET makes sense for this type of signal where the duration implies the number of occupancies in a particular state (for example the number of cache hits)

The “duration” type represents a signal that may stay active for multiple cycles on a given occurrence. These are typically stall signals where the user is attempting to determine the total number of cycles in the “stalled” state. “Duration” mode will indicate the total time stalled. “Event” mode will indicate the number of unique occurrences of stalls. “Event” is useful in this case since there will always be a deassertion between occurrences.

“Edge” type goes active for undefined period of time but is guaranteed to go inactive before the next occurrence. “Event” mode of the SCTM should be used for these signals.

Table 6-296. List of SMSET Events

SMSET Event	Name	Source	Type	SMSET Mode	Clock
0	tpcc_aet_start	EDMA	Pulse	Duration	CLK2
1	tpcc_aet_stop	EDMA	Pulse	Duration	CLK2
2	arp32_int4	INTC	Duration	Duration or event	CLK2
3	arp32_int5	INTC	Duration	Duration or event	CLK2
4	arp32_int6	INTC	Duration	Duration or event	CLK2
5	arp32_int7	INTC	Duration	Duration or event	CLK2
6	vcop_loop_start	VCOP	Edge	Event	CLK2
7	vcop_done	VCOP	Edge	Event	CLK2
8	arp32_nmi	INTC	Duration	Duration or event	CLK2
9	arp32_int8	INTC	Duration	Duration or event	CLK2
10	arp32_int9	INTC	Duration	Duration or event	CLK2
11	arp32_int10	INTC	Duration	Duration or event	CLK2
12	arp32_int11	INTC	Duration	Duration or event	CLK2
13	arp32_int12	INTC	Duration	Duration or event	CLK2
14	arp32_int13	INTC	Duration	Duration or event	CLK2
15	arp32_int14	INTC	Duration	Duration or event	CLK2
16	arp32_int15	INTC	Duration	Duration or event	CLK2

6.1.7.2.3 Software Messages

A GENERIC shall specify if software messages generated to STM through SMSET is supported. Software messages are initiated by processors. Basically a processor is writing a message to the trace receiver through STM. SMSET is just performing arbitration between hardware messages and software messages. Most of the subsystems cannot afford to have two OCP master ports for debug purpose. Interleaving is done in SMSET module.

6.1.7.2.4 SMSET Master Port

The SMSET master port is connected to an instrumentation OCP master port to export hardware and software messages through the STM module. The write accesses to STM are arbitrated at the instrumentation OCP interconnect level.

The SMSET master port supports burst transactions to reduce trace export surplus on the PTI interface; this surplus is due to instrumentation flows interleaving at instrumentation NoC level.

6.1.7.2.4.1 OCP Disconnect

The SMSET is supporting disconnect protocol on its master OCP interface, to make sure that software message transactions always complete. If software messages are initiated, while the instrumentation interconnect is disconnected, the access completes with a DVA response, and written data are lost.

6.1.7.2.5 SMSET Debug Features

The SMSET component provides the following system debug capabilities:

- Monitor events.
- Enable and disable event detection.
- Start event monitoring upon external trigger.
- Stop event monitoring upon external trigger.
- Time stamping.
- Generate hardware message upon system event.
- Generate hardware message every sampling window boundary.
- Interleave hardware messages and software messages.
- Program SMSET from debugger.
- Program SMSET from application.

6.1.7.2.6 Component Ownership

Some of the SMSET resources are owned either by the application or by the debugger. The ownership is required to configure or program SMSET. In other words, ownership determines if write access is granted to the SMSET control registers. The SMSET resource ownership is exclusive. Hence, simultaneous use of SMSET resources by both the debugger and the application is not permitted. However, the debugger can forcibly seize ownership of SMSET resources.

NOTE: A read access does not require ownership; therefore, either the debugger or the application can read any SMSET registers with or without ownership.

6.1.7.2.6.1 Ownership State

The ownership has three basic states:

- AVAILABLE: The unit has not been claimed.
- CLAIMED: The unit has been claimed.
- ENABLED: The unit is enabled by the owning party.

6.1.7.2.6.1.1 Available State

An SMSET is set to the AVAILABLE state when one of two conditions occurs:

- The state is available when power-on reset (POR) is asserted.
- The state is available if the SMSET is owned by the debugger and the emulator is disconnected.

NOTE: A warm reset has no effect on the operation of SMSET. An emulator disconnect has no effect when the application owns the unit.

6.1.7.2.6.1.2 Claimed State

An SMSET is in the CLAIMED state when either the debugger or the application has claimed it. The CLAIMED state provides exclusive access to the owner. The other party is not permitted write access to the unit until the owning party releases the claim. Once a unit is claimed, only the owner can write to the other resources controlled by the claim.

The debugger, however, is allowed to forcibly seize control of a unit, if it is already claimed by the application, by setting the DEBUGGEROVERRIDE bit to 1 in the SMSET configuration register (CFG).

6.1.7.2.6.1.3 Enabled State

An SMSET enters the ENABLED state when it is activated. The owner can continue writing to the other resources while SMSET is enabled.

6.1.7.2.6.2 Ownership Commands

Ownership of the SMSET resources is managed through the OWNERSHIP bit in the SMSET configuration register (CFG). Table 6-297 lists the ownership commands.

Table 6-297. Ownership Commands

OWNERSHIP Bits	Meaning	Description
00	Release ownership	Release ownership of system events trace. The Release command is accepted only from the owner.
01	Claim ownership	Claim access to system events trace. The claim command is successful only if the unit is available, or the requester is the debugger and the DEBUGGEROVERRIDE bit is high.
10	Enable unit	Activate system events trace for use. The enable command is accepted only from the owner.
11	No Operation	The NOP command does not affect ownership or claim state.

6.1.7.2.6.3 Claim Reset

When a resource is owned by the debugger, the resource is released if the debugger is disconnected. Debugger disconnection is accomplished by asserting PIDCON = 0.

If a resource is owned by the application and the processor enters the debug state, the resource continues to belong to the application. The SMSET is not sensitive to debug state.

If SMSET ownership is released while there are still data in the SMSET FIFO, the master port continues to export data to the STM until the FIFO is drained.

6.1.7.3 Use Case Examples

6.1.7.3.1 Procedure to Enable System Event Capture

Perform the following procedure to enable system event capture:

1. Claim the ownership by setting the OWNERSHIP bit in the SMSET configuration register (SMSET_CFG) to 01b.
2. Check the ownership and the current owner by reading the OWNERSHIP bit in the SMSET configuration register (SMSET_CFG).
3. Set the appropriate sampling window size value in the system event sampling window register (SMSET_SESW).
4. Enable the events needed to be detected by setting the appropriate bits in the programming of the

system event detection enable registers (SMSET_SEDEN_i) to 1.

5. Select the polarity of the event detection by writing the appropriate value to the EventPolarity bit in the SMSET configuration register (SMSET_CFG).
6. Select the message event generation by writing the appropriate value to the MessageEventGeneration bit in the SMSET configuration register (SMSET_CFG).
7. Enable system event capture by setting the SystemEventCapture bit in the SMSET configuration register (SMSET_CFG) to 1.

6.1.7.3.2 Procedure to Start and Stop System Event Capture from External Trigger Detection

Perform the following procedure to start and stop system event capture from external trigger detection:

1. Claim the ownership by setting the OWNERSHIP bit in the SMSET configuration register (SMSET_CFG) to 1.
2. Check the ownership and the current owner by reading the OWNERSHIP bit in the SMSET configuration register (SMSET_CFG).
3. Set the appropriate sampling window size value in the system event sampling window register (SMSET_SESW).
4. Enable the events to be detected by setting the appropriate bits in the programming of the system event detection enable registers (SMSET_SEDEN_i) to 1.
5. Select the polarity of the event detection by writing the appropriate value to the EventPolarity bit in the SMSET configuration register (SMSET_CFG).
6. Select the message event generation by writing the appropriate value to the MessageEventGeneration bit in the SMSET configuration register (SMSET_CFG).
7. Enable capturing system events from external trigger detection by setting the StopCapturingSystemEvents bit in the SMSET configuration register to 1.
8. Enable capturing system events from external trigger detection by setting the StartCapturingSystemEvents bit in the SMSET configuration register to 1.
9. Enable system event capture by writing a 1 to the SystemEventCapture bit in the SMSET configuration register.

6.1.7.3.3 Procedure to Disable System Event Capture

Perform the following procedure to disable system event capture:

1. Disable system event capture by setting the SystemEventCapture bit in the SMSET configuration register (SMSET_CFG) to 0.
2. Release the ownership by setting the OWNERSHIP bit in the SMSET configuration register (SMSET_CFG) to 0.
3. Check the ownership by reading the OWNERSHIP bit in the SMSET configuration register (SMSET_CFG). It must be available (OWNERSHIP = 0).

6.1.7.4 SMSET Register Manual

6.1.7.4.1 SMSET Instance Summary

Table 6-298. SMSET Instance Summary

Module Name	Physical Address	Size
EVE_SMSET	0x4208 8000	4k

6.1.7.4.2 SMSET Registers Mapping Summary

Table 6-299. EVE_SMSET Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	EVE_SMSET Base Address
SMSET_ID	R	32	0x0000 0000	0x4208 8000
SMSET_SCFG	RW	32	0x0000 0010	0x4208 8010
SMSET_SR	R	32	0x0000 0014	0x4208 8014
SMSET_CFG	RW	32	0x0000 0024	0x4208 8024
SMSET_SESW	RW	32	0x0000 0028	0x4208 8028
SMSET_SEDEN_i (1)	RW	32	0x0000 002C + (0x4*i)	0x4208 802C + (0x4*i)

(1) i = 1 to 8 for EVE_SMSET

6.1.7.4.3 SMSET Register Description

Table 6-300. SMSET_ID

Address Offset	0x0000 0000		
Physical Address	0x4208 8000	**Instance**	EVE_SMSET
Description	SMSET identification register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REVISION																															

	Bits	Field Name	Description	Type	Reset
(1)	31:0	REVISION	Revision	R	

(1) TI internal information

Table 6-301. Register Call Summary for Register SMSET_ID

Embedded Vision Engine (EVE) Subsystem
- SMSET Registers Mapping Summary: [0]

Table 6-302. SMSET_SCFG

Address Offset	0x0000 0010		
Physical Address	0x4208 8010	**Instance**	EVE_SMSET
Description	SMSET system configuration register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												IDLEMODE		RESERVED	SOFTRESET

Bits	Field Name	Description	Type	Reset
31:4	RESERVED		R	0x0
3:2	IDLEMODE	Configuration of the local target state management mode. 0x0: Force-idle; 0x1No-idle; 0x2 Smart-idle; 0x3 Smart-Idle wakeup-capable	R	0x2
1	RESERVED		R	0x0
0	SOFTRESET	Triggers System Event Trace module reset. This bit is automatically cleared by hardware	RW	0x0

Table 6-303. Register Call Summary for Register SMSET_SCFG

Embedded Vision Engine (EVE) Subsystem

- SMSET Registers Mapping Summary: [0]

Table 6-304. SMSET_SR

Address Offset	0x0000 0014		
Physical Address	0x4208 8014	**Instance**	EVE_SMSET
Description	SMSET Status Register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						SWFIFOEMPTY	HWFIFOEMPTY	RESERVED							RESETDONE

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	SWFIFOEMPTY	SW message FIFO empty	R	0x1
8	HWFIFOEMPTY	System event trace FIFO empty	R	0x1
7:1	RESERVED		R	0x0
0	RESETDONE	Reset completed	R	0x1

Table 6-305. Register Call Summary for Register SMSET_SR

Embedded Vision Engine (EVE) Subsystem

- SMSET Registers Mapping Summary: [0]

Table 6-306. SMSET_CFG

Address Offset	0x0000 0024		
Physical Address	0x4208 8024	**Instance**	EVE_SMSET
Description	SMSET Configuration register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OWNERSHIP		DEBUGGEROVERRIDE	CURRENTOWNER	RESERVED																				CAPTUREEN	RESERVED		EVENTLEVEL	EVENTMSG	STOP	START	RESERVED

Bits	Field Name	Description	Type	Reset
31:30	OWNERSHIP	Read to get current ownership status. The claim status encoding is (0=Available, 1=Claimed, 2=Enabled, 3=Reserved);Send command to modify ownership state: 00= Release ownership, 01 = Claim ownership, 10 = Enable unit, 11 = No operation	RW	0x0
29	DEBUGGEROVERRIDE	Reading from the DebuggerOverride bit returns a 1.	RW	0x1
28	CURRENTOWNER	This value reflects the SMSET ownership when the register is in a non-Available state.	R	0x0
27:8	RESERVED		R	0x0
7	CAPTUREEN	When high the sytem event capture is enabled	RW	0x0
6:5	RESERVED		R	0x0
4	EVENTLEVEL	This applies to all selected events: 0x0: low level event detection 0x1: high level evnet detection	RW	0x1
3	EVENTMSG	essage generated based on: 0x0: sampling window 0x1: event detection	RW	0x0
2	STOP	Stop capturing system events from external trigger detection [EMU1 HIGH to LOW]	RW	0x0
1	START	Start capturing system events from external trigger detection [EMU0 HIGH to LOW]	RW	0x0
0	RESERVED		R	0x0

Table 6-307. Register Call Summary for Register SMSET_CFG

Embedded Vision Engine (EVE) Subsystem

- Procedure to Enable System Event Capture: [0][1][2][3][4]
- Procedure to Start and Stop System Event Capture from External Trigger Detection: [5][6][7][8]
- Procedure to Disable System Event Capture: [9][10][11]
- SMSET Registers Mapping Summary: [12]

Table 6-308. SMSET_SESW

Address Offset	0x0000 0028		
Physical Address	0x4208 8028	**Instance**	EVE_SMSET
Description	System Event Sampling Window register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								SAMPLINGWINDOWSIZE							

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x0
7:0	SAMPLINGWINDOWSIZE	System events sampling window size expressed as SMSET cycles	RW	0x0

Table 6-309. Register Call Summary for Register SMSET_SESW

Embedded Vision Engine (EVE) Subsystem

- Procedure to Enable System Event Capture: [0]
- Procedure to Start and Stop System Event Capture from External Trigger Detection: [1]
- SMSET Registers Mapping Summary: [2]

Table 6-310. SMSET_SEDEN_i

Address Offset	0x0000 0030		
Physical Address	0x4208 802C + (0x4*i)	**Instance**	EVE_SMSET
Description	System Event Detection Enable register 1		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EVENT32EN	EVENT31EN	EVENT30EN	EVENT29EN	EVENT28EN	EVENT27EN	EVENT26EN	EVENT25EN	EVENT24EN	EVENT23EN	EVENT22EN	EVENT21EN	EVENT20EN	EVENT19EN	EVENT18EN	EVENT17EN	EVENT16EN	EVENT15EN	EVENT14EN	EVENT13EN	EVENT12EN	EVENT11EN	EVENT10EN	EVENT9EN	EVENT8EN	EVENT7EN	EVENT6EN	EVENT5EN	EVENT4EN	EVENT3EN	EVENT2EN	EVENT1EN

Bits	Field Name	Description	Type	Reset
31	EVENT32EN	Event 32 detection enable	RW	0x0
30	EVENT31EN	Event 31 detection enable	RW	0x0
29	EVENT30EN	Event 30 detection enable	RW	0x0
28	EVENT29EN	Event 29 detection enable	RW	0x0
27	EVENT28EN	Event 28 detection enable	RW	0x0
26	EVENT27EN	Event 27 detection enable	RW	0x0
25	EVENT26EN	Event 26 detection enable	RW	0x0
24	EVENT25EN	Event 25 detection enable	RW	0x0
23	EVENT24EN	Event 24 detection enable	RW	0x0
22	EVENT23EN	Event 23 detection enable	RW	0x0
21	EVENT22EN	Event 22 detection enable	RW	0x0
20	EVENT21EN	Event 21 detection enable	RW	0x0
19	EVENT20EN	Event 20 detection enable	RW	0x0
18	EVENT19EN	Event 19 detection enable	RW	0x0
17	EVENT18EN	Event 18 detection enable	RW	0x0
16	EVENT17EN	Event 17 detection enable	RW	0x0
15	EVENT16EN	Event 16 detection enable	RW	0x0
14	EVENT15EN	Event 15 detection enable	RW	0x0
13	EVENT14EN	Event 14 detection enable	RW	0x0
12	EVENT13EN	Event 13 detection enable	RW	0x0
11	EVENT12EN	Event 12 detection enable	RW	0x0
10	EVENT11EN	Event 11 detection enable	RW	0x0
9	EVENT10EN	Event 10 detection enable	RW	0x0
8	EVENT9EN	Event 9 detection enable	RW	0x0
7	EVENT8EN	Event 8 detection enable	RW	0x0
6	EVENT7EN	Event 7 detection enable	RW	0x0
5	EVENT6EN	Event 6 detection enable	RW	0x0
4	EVENT5EN	Event 5 detection enable	RW	0x0
3	EVENT4EN	Event 4 detection enable	RW	0x0
2	EVENT3EN	Event 3 detection enable	RW	0x0

Bits	Field Name	Description	Type	Reset
1	EVENT2EN	Event 2 detection enable	RW	0x0
0	EVENT1EN	Event 1 detection enable	RW	0x0

Table 6-311. Register Call Summary for Register SMSET_SEDEN_i

Embedded Vision Engine (EVE) Subsystem

- Procedure to Enable System Event Capture: [0]
- Procedure to Start and Stop System Event Capture from External Trigger Detection: [1]
- SMSET Registers Mapping Summary: [2]

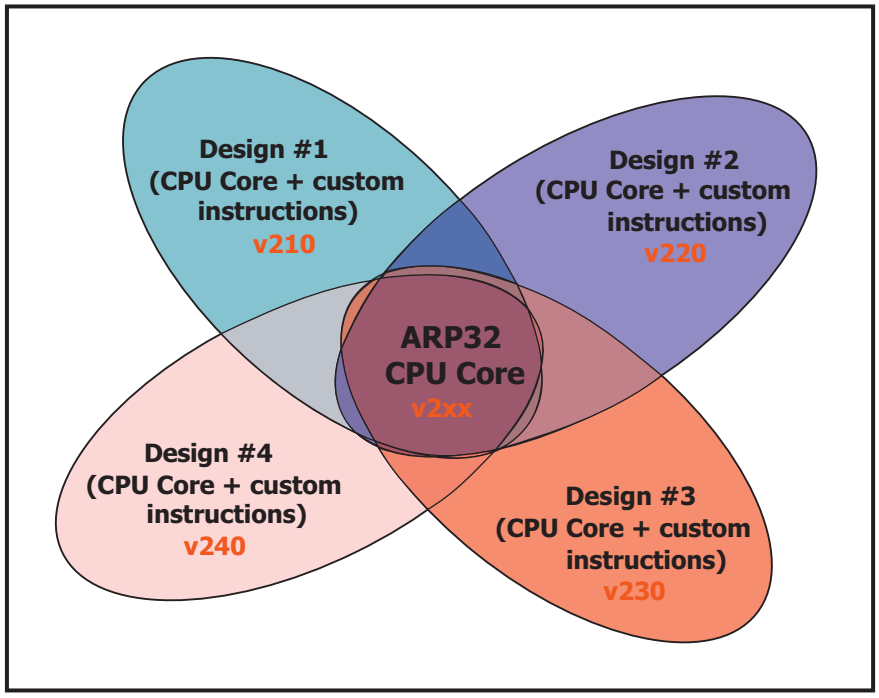
6.2 ARP32 CPU and Instruction Set

6.2.1 Overview

ARP32 refers to a family of customizable, embedded processors targeted for deeply embedded control applications that require high-performance at very-low system cost (die size) and dynamic power-consumption. Example of applications that can benefit using the ARP32 includes embedded video processing engine, computer vision, and video analytics processing engine.

Figure 6-23 illustrates the overlapping nature of the ARP32 CPU and feature set in different application contexts.

Figure 6-23. ARP32 Versions and ISA/Feature Space



6.2.2 Features

The main features of the ARP32 processor include:

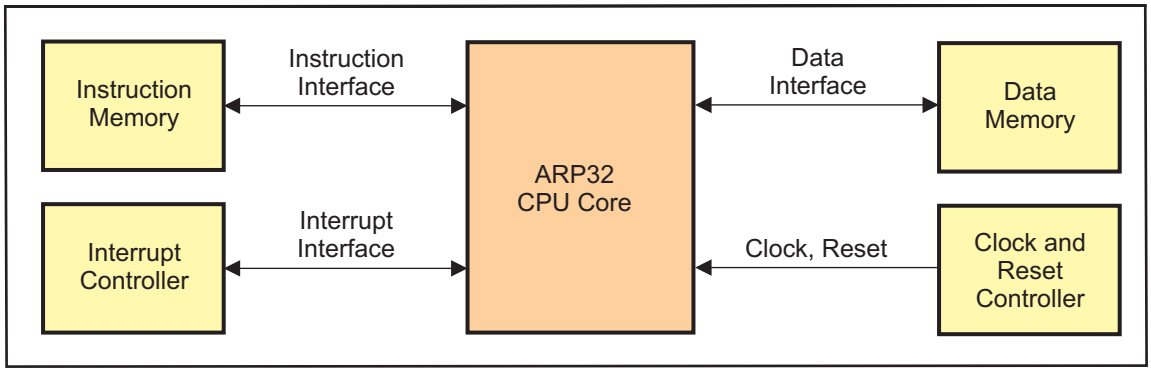
- Single issue RISC processor with Harvard memory architecture
- Highly efficient integer pipeline delivering high performance at low area and power cost
- Provides customization of instruction set to provide acceleration of a target application
- Supported by fully optimizing C/C++ compiler including intrinsic-driven usage of special custom instructions

It is recommended that ARP32 CPU users only use C/C++ language with the supported TI compiler and tools; assembly level programming is highly discouraged and may break future compatibility with the ARP32 processor family or its variants.

6.2.3 Block Diagram

Figure 6-24 shows the ARP32 core interfaces.

Figure 6-24. ARP32 CPU Block Diagram



6.2.4 Architecture

6.2.4.1 Interface Description

Table 6-312 provides the basic description of the interface signals of the core. The data and instruction memory interface implements a simple request-ready based handshake sufficient to directly interface TIs compiled SRAM memories.

There is no architectural clock-gating implemented within the ARP32 core. The clock control logic is implemented outside the ARP32 core assisted by signals driven by the CPU core via the **IDLE** instruction.

Table 6-312. Interface Signals

Signal	Direction	Width	Description
		Clock, Reset	
cpu_fclk	In	1	Functional clock input
cpu_porz_i	In	1	Power On reset (active low)
cpu_resetz_i	In	1	Functional reset (active low)
cpu_standby_o	Out	1	CPU standby status (active high

Table 6-312. Interface Signals (continued)

Signal	Direction	Width	Description
Data Interface			
cpu_dmem_enz_o	Out	1	Data memory request, active low
cpu_dmem_dbg_o	Out	1	Debug access qualifier for Data memory request
cpu_dmem_wrz_o	Out	1	Data memory write enable, active low (0: write, 1:read)
cpu_dmem_bez[3:0]	Out	4	Data memory byte enables, active low
cpu_dmem_addr_o[31:2]	Out	30	Data memory address (carries a word address)
cpu_dmem_wdata_o[31:0]	Out	32	Data memory write data
cpu_dmem_rdata_i[31:0]	In	32	Data memory read data
cpu_dmem_rdy_i	In	1	Data memory ready, active high
cpu_dmem_err_i	In	1	Data memory access error, active high, valid along with cpu_dmem_rdy_i
Instruction Interface			
cpu_imem_enz_o	Out	1	Instruction memory request, active low
cpu_imem_dbg_o	Out	1	Debug access qualifier for Instruction memory request
cpu_imem_addr_o[31:2]	Out	30	Instruction memory address (carries a word address)
cpu_imem_rdata_i[31:0]	In	32	Instruction memory read data
cpu_imem_rdy_i	In	1	Instruction memory ready, active high
cpu_imem_err_i	In	1	Instruction memory access error, active high, valid along with cpu_imem_rdy_i
Interrupt Interface			
cpu_nmi_i	In	1	Non-maskable interrupt, active high
cpu_int4_i	In	1	Maskable interrupt 4, active high
cpu_int5_i	In	1	Maskable interrupt 5, active high
cpu_int6_i	In	1	Maskable interrupt 6, active high
cpu_int7_i	In	1	Maskable interrupt 7, active high
cpu_int8_i	In	1	Maskable interrupt 8, active high
cpu_int9_i	In	1	Maskable interrupt 9, active high
cpu_int10_i	In	1	Maskable interrupt 10, active high
cpu_int11_i	In	1	Maskable interrupt 11, active high
cpu_int12_i	In	1	Maskable interrupt 12, active high
cpu_int13_i	In	1	Maskable interrupt 13, active high
cpu_int14_i	In	1	Maskable interrupt 14, active high
cpu_int15_i	In	1	Maskable interrupt 15, active high
cpu_iack_o	Out	1	Interrupt acknowledge, active high
cpu_inum_o	Out	4	Identifier of acknowledged interrupt

6.2.4.1.1 Data Memory Interface

- The address sent out to data memory is always a word address. Access sizes smaller than word size are supported via byte enables.
- The CPU core asserts the memory request signal (active low, cpu_dmem_enz_o) along with corresponding address (cpu_dmem_addr_o) and read/write qualifier (active low, cpu_dmem_wrz_o) until a ready (active high, cpu_dmem_rdy_i) is received from the memory sub-system.
- For a zero-wait state access, the ready is expected next cycle to the request. Ready must be de-asserted to extend the access cycle for a higher wait state memory subsystem.
- For read accesses, the CPU samples the read data bus (cpu_dmem_rdata_i[31:0]) when ready is asserted. For write accesses, ready indicates write completion whereas read data bus is a don't care.
- For writes accesses, appropriate byte enables (active low, cpu_dmem_bez[3:0]) are asserted in request phase. This is the only way for the memory sub-system to identify access size of a write access. Byte steering is done by the core, hence the memory sub-system can use the ANY lanes of write data bus for easy selections/multiplexing of write data - for byte writes, the write data is duplicated on all four byte lanes of write data bus whereas for halfword writes, the write data is duplicated on both the halfword lanes of write data bus.
- For read accesses, access size is always 32b - the memory sub-system always returns 32b at the requested word address. CPU core selects the appropriate bytes based on access size and access address.

6.2.4.1.2 Instruction Memory Interface

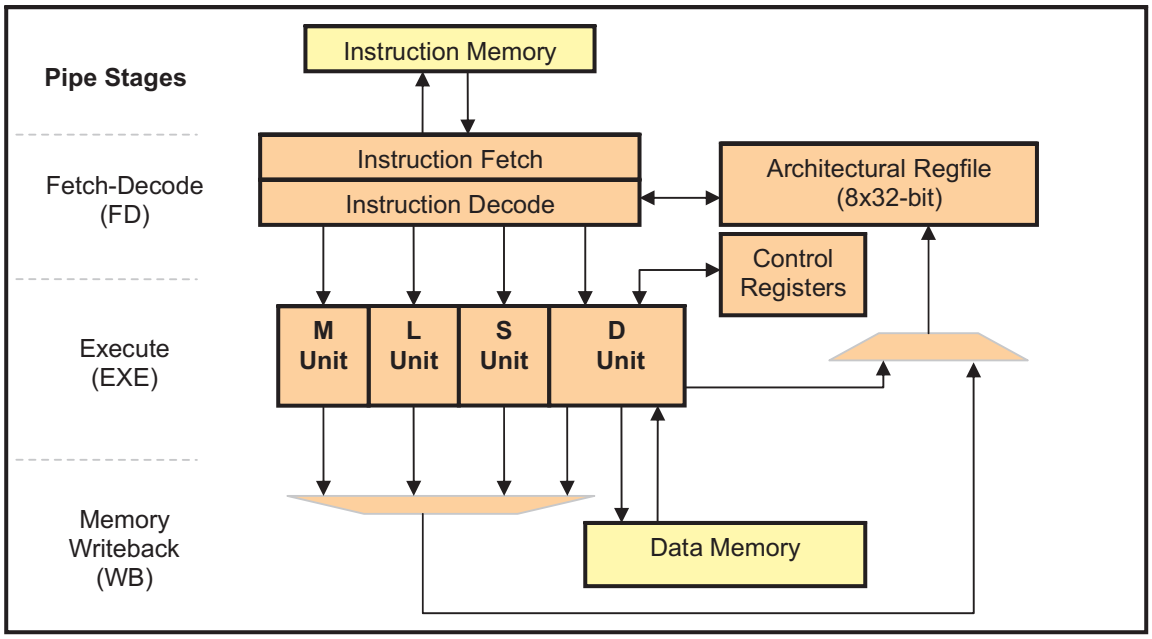
- The address sent out to the instruction memory is always a word address, access size is always a word (32 bits) and writes are not supported. Hence, byte enables and read/write qualifiers are not present for the instruction memory interface.
- The CPU core asserts the memory request signal (active low, cpu_imem_enz_o) along with the corresponding address (cpu_imem_addr_o) until a ready (active high, cpu_imem_rdy_i) is received from the memory subsystem.
- For a zero-wait state access, the ready is expected the next cycle to the request. Ready must be de-asserted to extend the access cycle for a higher wait state memory subsystem.
- The CPU samples the read data bus (cpu_imem_rdata_i[31:0]) when ready is asserted.

6.2.4.2 Pipeline

6.2.4.2.1 Overview

The ARP32 CPU implements a three-deep pipeline (see Figure 6-25): merged fetch-decode (FD), Execute (EXE), and Writeback (WB). The fetch-decode and writeback stages are overlapped.

Figure 6-25. ARP32 CPU Pipeline



6.2.4.2.2 Pipeline Operation

The first stage (FetchDecode/FD) of the pipe operates directly on the instruction memory read data input. This stage consists of the following:

- Instructions are decoded to determine
  - size of the instruction (32 bit/16 bit)
  - source operand, destination operand (in architectural and control register file)
  - if it is a blocking multi-execute cycle instruction
- Source operands are read from architectural register file

In the second stage (Execute/EXE) of the pipe:

- All arithmetic/data processing instructions are executed.
- For a single execute cycle instruction, the results are available at the end of this stage and are written back to architectural register file and control register file (if required).
- Instructions requiring data memory access (load, store, call, return etc) compute the access address in this cycle and place the memory access request on data memory bus (along with all other qualifiers). For write accesses, write data is also provided in this stage.

In the last stage (MemoryWriteback/WB) of the pipe:

- All instructions having data memory access request issued in EXE stage waits for the read data (for read accesses) or write completion notification (for write accesses) via the data memory 'ready' signal.
- If memory subsystem asserts the ready signal, the read data is updated to architectural register file

NOTE: The Writeback stage is really outside the processor core and is used for load/store instructions only. All other instructions complete at EXE stage updating the result. Hence, in some sense the pipe depth of the ARP32 processor is really 2 (Fetch-Decode, Execute).

Consider the following example code sequence in Example 6-1. Table 6-313 shows how the instructions flows through the pipe, highlighting the fact that only load/store instructions use the WB stage (others complete/retire at EXE).

Example 6-1. ARP32 CPU Pipeline Operation

```
MVK   0x100, R0         ; (I1) Move 0x100 into R0
MVKH  0x02000000, R0    ; (I2) Move 0x0200 into R0[31:16]
ADD   100, R0, R0       ; (I3) Add 100 to R0
LDW   *+R2(0), R3       ; (I4) Load a word to R3
ADD   4, R2, R2         ; (I5) Increment R2 to next word address
STW   R3, *+R4(0)       ; (I6) Store R3 to address pointed by R4
ADD   4, R4, R4         ; (I7) Increment R4 to next word address
```

Table 6-313. ARP32 CPU Pipeline Operation

Cycle	1	2	3	4	5	6	7	8	9
FD	MVK (I1)	MVKH (I2)	ADD (I3)	LDW (I4)	ADD (I5)	STW (I6)	ADD (I7)		
EXE		MVK (I1)	MVKH (I2)	ADD (I3)	LDW (I4)	ADD (I5)	STW (I6)	ADD (I7)	
WB						LDW (I4)		STW (I6)	

6.2.4.2.3 Pipeline Interlocks

The pipeline is fully interlocked – the CPU stalls in case of source operand registers have pending loads. Data bypassing for read-after-write dependency is implemented at the end of EXE and WB stages to increase instructions-per-cycle (IPC).

The load data has a single-cycle load use penalty since the load data is written back to register file at WB stage. In the following example, the ADD instruction stalls for a cycle to allow load to complete:

```
LDW   *+R0(0), R0       ; Load a word into R0
ADD   4, R0, R0         ; Increment R0 to the next word address
MVK   100, R1           ; Move a value 100 into R1
```

Since the CPU allows a nondependent instruction to continue executing, this stall is avoided, if the MVK (no dependency on the load data) instruction is placed in the load delay slot – the CPU executes all three instructions without a stall.

```
LDW   *+R0(0), R0       ; Load a word into R0
MVK   100, R1           ; Move a value 100 into R1
ADD   4, R0, R0         ; Increment R0 to the next word address
```

6.2.4.3 Data Format

The ARP32 CPU is an integer machine supporting three different data width/formats: byte, halfword, and word. A byte is 8 bits, a halfword is 16 bits, and a word is 32 bits – they directly map to native C data types: char, short, and int, respectively.

Each type is aligned with the corresponding size, that is, byte aligned to byte boundary, halfword aligned to 2-byte boundary, word aligned to 4-byte boundary. It is the compiler/toolchain and the programmers responsibility to maintain the access alignment all the time (for global and/or local variables).

The ARP32 core does not support unaligned access. If an unaligned access is attempted by software (via an appropriate instruction), the CPU ignores the lower 2 bits of the data memory (byte) address, thus forcing an alignment to the (truncated, floored) word address.

6.2.4.4 Endian Support

The ARP32 CPU supports only little-endian byte ordering in the memory subsystem (see Figure 6-26). A word loaded from the memory is loaded into a CPU register as follows:

- a byte or half word at address A is the least significant byte or half word within the word at that address
- a byte at address A is the least significant byte within the half word at that address

Figure 6-26. ARP32 CPU Little Endianness

	31 24	23 16	15 8	7 0
Word at address A	Byte [Addr + 3]	Byte [Addr + 2]	Byte [Addr + 1]	Byte [Addr + 0]
Halfword at address A			Byte [Addr + 1]	Byte [Addr + 0]

6.2.4.5 Architectural Register File

The ARP32 CPU contains an 8-entry, 32-bit architectural register file named R0-R7.

The architectural registers are read (for source operands) at the end of Fetch-decode (2 dedicated read ports available at fetch-decode for this) and written back at the end of execute stage (1 dedicated write port available at EXE for this). Additionally, for load instructions, the memory read data is written back to architectural register file at the end of writeback stage (1 dedicated write port available at WB for this) and for store instructions, the architectural register file is read (for write data) at the execute stage (via a dedicated read port available in EXE phase).

The register file contains logic to detect matching of write address in the execute stage and two read addresses in the decode stage for bypassing of result data to read data buses. The read data is flopped in the register file and routed to the execution unit in the execute stage.

6.2.4.6 CPU Control Registers

The control register file (CREG) contains registers that control or report status for the ARP32 CPU. Table 6-314 lists the control registers contained in the control register file.

Control registers are accessed via **MVC**, **MVCH** instructions – this is the only mechanism for moving the contents of registers between the architectural register file and the control register file. Control registers are addressable via 5 bits in the *creg* field of the **MVC/MVCH** instructions. All control registers are defined as 32 bits – **MVC** + **MVCH** pair should be used to move immediate values to any control registers.

The following instructions are available in the **MVC/MVCH** family:

- **MVC** *ucst16, creg*
- **MVC** *areg, creg*
- **MVC** *creg, areg*
- **MVCH** *ucst16, creg*

Table 6-314. Control Registers

Acronym	Register Name	Section
CSR	Control Status Register	Section 6.2.4.6.1
IER	Interrupt Enable Register	Section 6.2.4.6.2
IFR	Interrupt Flag Register	Section 6.2.4.6.3
ISR	Interrupt Set Register	Section 6.2.4.6.4
ICR	Interrupt Clear Register	Section 6.2.4.6.5
NRP	Nonmaskable Interrupt Return Pointer Register	Section 6.2.4.6.6
IRP	Interrupt Return Pointer Register	Section 6.2.4.6.7
SP	Stack Pointer Register	Section 6.2.4.6.8
GDP	Global Data Pointer Register	Section 6.2.4.6.9
LR	Link Register	Section 6.2.4.6.10
LSA0	Loop 0 Start Address Register	Section 6.2.4.6.11
LEA0	Loop 0 End Address Register	Section 6.2.4.6.12
LCNT0	Loop 0 Iteration Count Register	Section 6.2.4.6.13
LSA1	Loop 1 Start Address Register	Section 6.2.4.6.14
LEA1	Loop 1 End Address Register	Section 6.2.4.6.15
LCNT1	Loop 1 Iteration Count Register	Section 6.2.4.6.16
LCNT0RLD	Loop 0 Iteration Count Reload Value Register	Section 6.2.4.6.17
SCSR	Shadow Control Status Register	Section 6.2.4.6.18
NMISCSR	NMI Shadow Control Status Register	Section 6.2.4.6.19
CPUID	CPU ID Register	Section 6.2.4.6.20
DPC	Decode Program Counter	Section 6.2.4.6.21
TSCH	Time-Stamp Counter (high 32 bits) Register	Section 6.2.4.6.22
TSCL	Time-Stamp Counter (low 32 bits) Register	Section 6.2.4.6.22

Direct modification of the CREG entries is limited to a few special case instructions. For example, some forms of the **ADD** and **SUB** instructions directly modify the stack pointer (SP) to improve code execution performance.

A read of a control register (by **MVC** *creg, areg*) can be immediately followed by any other instruction using the same *areg* without causing a stall – the read data is forwarded to the next instruction in the pipeline. The following example illustrates the behavior:

```
// Read from IRP and push it stack
MVC   IRP, R0          ; Read IRP and place it is R0
STRF  R0, R0           ; No stall, content of IRP is pushed to stack
```

A write to a control register is also bypassed to another instruction immediately following it and using the same *creg*. The following example illustrates the behavior:

```
// Update SP before a register push
MVC   5000, SP         ; Update SP to 5000
STRF  R0, R0           ; No stall, R0 is pushed @5000
```

However, a write to a control register (by **MVC** *areg, creg*) has one exposed effective delay slot for an immediately following read from the same control register (by **MVC** *creg, areg*). The following examples illustrate the behavior:

```
// Write to LCNT0 and read it back
MVC   2000, LCNT0      ; Update LCNT0 again
NOP                    ; without this NOP the next MVC will read UNPREDICTABLE value
MVC   LCNT0, R0        ; R0=2000
```

6.2.4.6.1 Control Status Register (CSR)

The control status register (CSR) contains control and status bits. The CSR is shown in Figure 6-27 and described in Table 6-315.

The condition code bits GT, LT, and EQ are not required to be one-hot. They may be set in any combinations using the **MVC** instruction or by combinations of **CMP** and instructions that update the EQ bit. However, execution of compare instructions enforces a one-hot condition for GT/LT/EQ.

Having more than one bit set does not affect conditional branch execution, as each branch compares only the respective condition bits, that is, the **BGE** instruction uses the CSR[4] and CSR[2] bits to determine if the branch is taken. The remaining condition bits have no effect on the **BGE** instruction.

Figure 6-27. Control Status Register (CSR)

31														16
Reserved														
R-0														
15	14	13	12	11		8	7	6	5	4	3	2	1	0
Reserved		DBZ	DBZE	INUM			V	SAT	C	GT	LT	EQ	Rsvd	GIE
R-0		R/W-0	R/W-0	R/W-0			R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-0	R/W-0

LEGEND: R = Readable by the **MVC** instruction; W = Writeable by the **MVC** instruction; -*n* = value after reset

Table 6-315. Control Status Register (CSR) Field Descriptions

Bit	Field	Value	Description
31-14	Reserved	0	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.
13	DBZ	0-1	Divide-By-Zero. Status indicating if a Divide-By-Zero condition occurred since last time it was cleared. Once set, software MUST clear this bit. It is set only if the DBZE bit is set.
12	DBZE	0-1	Divide-By-Zero exception enable. When set to 1, CPU takes an UNDEF interrupt when a divide-by-zero condition is detected by **DIV, DIVU, MOD,** or **MODU** instructions. By default this bit is disabled.
11-8	INUM		Interrupt Number. Indicates the interrupt ID of the last taken interrupt.
		0	Reset
		1h	NMI
		2h	SWI
		3h	UNDEF
		4h	INT4
		5h	INT5
		6h	INT6
		7h	INT7
		8h	INT8
		9h	INT9
		Ah	INT10
		Bh	INT11
		Ch	INT12
		Dh	INT13
		Eh	INT14
		Fh	INT15
7	V	0-1	Overflow bit. Arithmetic operations that results in overflow or borrow set this bit. See individual instruction descriptions for instructions that modify the V bit.
6	SAT	0-1	Saturation bit. Arithmetic operations whose results have been saturated set this bit. See individual instruction descriptions for instructions that modify the SAT bit.
5	C	0-1	Carry bit. Arithmetic operations that results in carry out or borrow set this bit. See individual instruction descriptions for instructions that modify the C bit.

Table 6-315. Control Status Register (CSR) Field Descriptions (continued)

Bit	Field	Value	Description
4	GT	0-1	Greater-than bit. This bit is set or cleared based on the result of a **CMP** instruction. (GT = 1 if Rx > Ry; else GT = 0). See individual instruction descriptions for instructions that modify the GT bit.
3	LT	0-1	Less-than bit. This bit is set or cleared based on the result of a **CMP** instruction. (LT = 1 if Rx < Ry; else LT = 0). See individual instruction descriptions for instructions that modify the LT bit.
2	EQ	0-1	Equal bit. This bit is set to 1 if the result of an instruction execution results in a zero result or the result of a **CMP** instruction returns equality (EQ = 1 if Rx == Ry; else EQ = 0). See individual instruction descriptions for instructions that modify the EQ bit.
1	Reserved	0	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.
0	GIE		Global interrupt enable.
		0	Disables all interrupts, except the reset interrupt and NMI (nonmaskable interrupt).
		1	Enables all interrupts.

6.2.4.6.2 Interrupt Enable Register (IER)

The interrupt enable register (IER) enables and disables individual interrupts. The IER is shown in Figure 6-28 and described in Table 6-316.

Figure 6-28. Interrupt Enable Register (IER)

31															16
Reserved															
R-0															
15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IE15	IE14	IE13	IE12	IE11	IE10	IE9	IE8	IE7	IE6	IE5	IE4	Reserved		NMIE	Rsvd
R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-0		R/W-0	R-1

LEGEND: R = Readable by the **MVC** instruction; W = Writeable by the **MVC** instruction; -*n* = value after reset

Table 6-316. Interrupt Enable Register (IER) Field Descriptions

Bit	Field	Value	Description
31-16	Reserved	0	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.
15-4	IE*n*		Interrupt enable. An interrupt triggers interrupt processing only if the corresponding bit is set to 1.
		0	Interrupt is disabled.
		1	Interrupt is enabled.
3-2	Reserved	0	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.
1	NMIE		Nonmaskable interrupt enable. An interrupt triggers interrupt processing only if the bit is set to 1.
			The NMIE bit is cleared at reset. After reset, software must set the NMIE bit to enable the NMI and to allow INT15-INT4 to be enabled by the GIE bit in CSR and the corresponding IER bit. The NMIE bit cannot be cleared manually; a write of 0 has no effect. The NMIE bit is also cleared by the occurrence of an NMI.
		0	All nonreset interrupts are disabled.
		1	All nonreset interrupts are enabled.
0	Reserved	1	Reserved. The reserved bit location is always read as 1. A value written to this field has no effect.

6.2.4.6.3 Interrupt Flag Register (IFR)

The interrupt flag register (IFR) contains the status of the maskable interrupts (INT15-INT4) and the NMI interrupt. Each corresponding bit in the IFR is set to 1 when that interrupt occurs; otherwise, the bits are cleared to 0. The IFR is shown in Figure 6-29 and described in Table 6-317.

The update to IFR via a write to the interrupt set register (ISR)/interrupt clear register (ICR) has one effective delay slot; IFR cannot be set/cleared using one **MVC** instruction (a write to ISR/ICR) and read the corresponding changed value of IFR via a very next **MVC** instruction. Software should use a **NOP** after a write to ISR/ICR, if the changed value has to be read immediately.

```
MVC   0x10, ISR         ; Set IFR[4]
NOP                     ; a NOP to fill the delay slot
MVC   IFR, R0           ; R0 would contain 0x10
```

Figure 6-29. Interrupt Flag Register (IFR)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
R-0															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IF15	IF14	IF13	IF12	IF11	IF10	IF9	IF8	IF7	IF6	IF5	IF4	Reserved		NMIF	Rsvd
R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0	R-0		R-0	R-0

LEGEND: R = Readable by the **MVC** instruction; -*n* = value after reset

Table 6-317. Interrupt Flag Register (IFR) Field Descriptions

Bit	Field	Value	Description
31-16	Reserved	0	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.
15-4	IF*n*		Interrupt flag. Indicates the status of the corresponding maskable interrupt. An interrupt flag may be manually set by setting the corresponding bit (IS*n*) in the interrupt set register (ISR) or manually cleared by setting the corresponding bit (IC*n*) in the interrupt clear register (ICR).
		0	Interrupt has not occurred.
		1	Interrupt has occurred.
3-2	Reserved	0	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.
1	NMIF		Nonmaskable interrupt flag.
		0	Interrupt has not occurred.
		1	Interrupt has occurred.
0	Reserved	0	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.

6.2.4.6.4 Interrupt Set Register (ISR)

The interrupt set register (ISR) allows to manually set the maskable interrupts (INT15-INT4) in the interrupt flag register (IFR). Writing a 1 to any of the bits in ISR causes the corresponding interrupt flag (IF*n*) to be set in IFR. Writing a 0 to any bit in ISR has no effect. It is not possible to set any bit in ISR to affect NMI or reset. The ISR is shown in Figure 6-30 and described in Table 6-318.

NOTE: Any write to ISR (by the **MVC** instruction) effectively has one delay slot because the results cannot be read (by the **MVC** instruction) in IFR until two cycles after the write to ISR.

Any write to the interrupt clear register (ICR) is ignored by a simultaneous write to the same bit in ISR.

Figure 6-30. Interrupt Set Register (ISR)

31													16
Reserved													
R-0													
15	14	13	12	11	10	9	8	7	6	5	4	3	0
IS15	IS14	IS13	IS12	IS11	IS10	IS9	IS8	IS7	IS6	IS5	IS4	Reserved	
W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	R-0	

LEGEND: R = Read only; W = Writeable by the **MVC** instruction; *-n* = value after reset

Table 6-318. Interrupt Set Register (ISR) Field Descriptions

Bit	Field	Value	Description
31-16	Reserved	0	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.
15-4	IS*n*		Interrupt set.
		0	Corresponding interrupt flag (IF*n*) in IFR is not set.
		1	Corresponding interrupt flag (IF*n*) in IFR is set.
3-0	Reserved	0	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.

6.2.4.6.5 Interrupt Clear Register (ICR)

The interrupt clear register (ICR) allows to manually clear the maskable interrupts (INT15-INT4) in the interrupt flag register (IFR). Writing a 1 to any of the bits in ICR causes the corresponding interrupt flag (IF*n*) to be cleared in IFR. Writing a 0 to any bit in ICR has no effect. Incoming interrupts have priority and override any write to ICR. It is not possible to set any bit in ICR to affect NMI or reset. The ISR is shown in Figure 6-31 and described in Table 6-319.

NOTE: Any write to ICR (by the **MVC** instruction) effectively has one delay slot because the results cannot be read (by the **MVC** instruction) in IFR until two cycles after the write to ICR.

Any write to ICR is ignored by a simultaneous write to the same bit in the interrupt set register (ISR).

Figure 6-31. Interrupt Clear Register (ICR)

31													16
Reserved													
R-0													
15	14	13	12	11	10	9	8	7	6	5	4	3	0
IC15	IC14	IC13	IC12	IC11	IC10	IC9	IC8	IC7	IC6	IC5	IC4	Reserved	
W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	W-0	R-0	

LEGEND: R = Read only; W = Writeable by the **MVC** instruction; *-n* = value after reset

Table 6-319. Interrupt Clear Register (ICR) Field Descriptions

Bit	Field	Value	Description
31-16	Reserved	0	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.
15-4	IC*n*		Interrupt clear.
		0	Corresponding interrupt flag (IF*n*) in IFR is not cleared.
		1	Corresponding interrupt flag (IF*n*) in IFR is cleared.
3-0	Reserved	0	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.

6.2.4.6.6 Nonmaskable Interrupt (NMI) Return Pointer Register (NRP)

The NMI return pointer register (NRP) contains the return pointer that directs the CPU to the proper location to continue program execution after NMI processing. A branch using the address in NRP (**BNRP**) in your interrupt service routine returns to the program flow when NMI servicing is complete. The NRP is shown in Figure 6-32 and described in Table 6-320.

Figure 6-32. NMI Return Pointer Register (NRP)

31	30	0
Rsvd	Return Address	
R-0	R/W-x	

LEGEND: R = Readable by the **MVC** instruction; W = Writeable by the **MVC** instruction; -x = value is indeterminate after reset

Table 6-320. NMI Return Pointer Register (NRP) Field Descriptions

Bit	Field	Value	Description
31	Reserved	0	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.
30-0	Return Address	0-7FFF FFFFh	Contains the return address. It is written by the CPU while taking an NMI. It is read by the CPU to execute the **BNRP** instruction.

6.2.4.6.7 Interrupt Return Pointer Register (IRP)

The interrupt return pointer register (IRP) contains the return pointer that directs the CPU to the proper location to continue program execution after processing a maskable interrupt (INT15-INT4, SWI, or UNDEF). A branch using the address in IRP (**BIRP**) in your interrupt service routine returns to the program flow when interrupt servicing is complete. The IRP is shown in Figure 6-33 and described in Table 6-321.

Figure 6-33. Interrupt Return Pointer Register (IRP)

31	30	0
Rsvd	Return Address	
R-0	R/W-x	

LEGEND: R = Readable by the **MVC** instruction; W = Writeable by the **MVC** instruction; -x = value is indeterminate after reset

Table 6-321. Interrupt Return Pointer Register (IRP) Field Descriptions

Bit	Field	Value	Description
31	Reserved	0	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.
30-0	Return Address	0-7FFF FFFFh	Contains the return address. It is written by the CPU while taking an interrupt. It is read by the CPU to execute the **BIRP** instruction.

6.2.4.6.8 Stack Pointer Register (SP)

The stack pointer register (SP) may be directly updated by software using the **MVC** instruction. The SP points to the top of the stack and always contains a byte address. The programmer is responsible for ensuring the correct alignment of the SP. See Section 6.2.4.14 for more details on stack pointer convention that must be followed. The SP is shown in Figure 6-34.

Figure 6-34. Stack Pointer Register (SP)

31	0
Stack Pointer	
R/W-0	

LEGEND: R = Readable by the **MVC** instruction; W = Writeable by the **MVC** instruction; -*n* = value after reset

6.2.4.6.9 Global Data Pointer Register (GDP)

The GDP may be directly updated by software using the **MVC/MVCH** instruction. The GDP points to the start of the global data section (.bss section) and always contains a byte address. The programmer is responsible for ensuring that the convention described in Section 6.2.4.15 is maintained. The GDP is shown in Figure 6-35.

Figure 6-35. Global Data Pointer Register (GDP)

31	0
Global Data Pointer	
R/W-0	

LEGEND: R = Readable by the **MVC** instruction; W = Writeable by the **MVC** instruction; -*n* = value after reset

6.2.4.6.10 Link Register (LR)

The link register (LR) contains the return address (a halfword address) for a **RET** instruction. A **CALL** instruction saves the last return address into the stack and updates LR with the return address. A **RET** instruction uses the current value of LR as the target address and restores the last return address into LR from the stack. The LR is shown in Figure 6-36 and described in Table 6-322.

Figure 6-36. Link Register (LR)

31	30	0
Rsvd	Return Address	
R-0	R/W-0	

LEGEND: R = Readable by the **MVC** instruction; W = Writeable by the **MVC** instruction; -*n* = value after reset

Table 6-322. Link Register (LR) Field Descriptions

Bit	Field	Value	Description
31	Reserved	0	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.
30-0	Return Address	0-7FFF FFFFh	Contains the return address (a halfword address) for a **RET** instruction.

6.2.4.6.11 Loop 0 Start Address Register (LSA0)

The loop 0 start address register (LSA0) contains the start address (byte address) of Loop 0. Note that Loop 0 is the inner loop. The LSA0 is shown in Figure 6-37 and described in Table 6-323.

See Section 6.2.4.17 for more details on hardware loop mechanism and usage of this register.

Figure 6-37. Loop 0 Start Address Register (LSA0)

31	0
LSA0	
R/W-0	

LEGEND: R = Readable by the **MVC** instruction; W = Writeable by the **MVC** instruction; -*n* = value after reset

Table 6-323. Loop 0 Start Address Register (LSA0) Field Descriptions

Bit	Field	Value	Description
31-0	LSA0	0-FFFF FFFFh	Start address of Loop 0 (byte address).

6.2.4.6.12 Loop 0 End Address Register (LEA0)

The loop 0 end address register (LEA0) contains the end address (byte address) of Loop 0. Note that Loop 0 is the inner loop. The LEA0 is shown in Figure 6-38 and described in Table 6-324.

See Section 6.2.4.17 for more details on hardware loop mechanism and usage of this register.

Figure 6-38. Loop 0 End Address Register (LEA0)

31	0
LEA0	
R/W-0	

LEGEND: R = Readable by the **MVC** instruction; W = Writeable by the **MVC** instruction; -*n* = value after reset

Table 6-324. Loop 0 End Address Register (LEA0) Field Descriptions

Bit	Field	Value	Description
31-0	LEA0	0-FFFF FFFFh	End address of Loop 0 (byte address).

6.2.4.6.13 Loop 0 Iteration Count Register (LCNT0)

The loop 0 iteration count register (LCNT0) contains the iteration count of Loop 0. Note that Loop 0 is the inner loop. The LCNT0 is shown in Figure 6-39 and described in Table 6-325.

See Section 6.2.4.17 for more details on hardware loop mechanism and usage of this register.

Figure 6-39. Loop 0 Iteration Count Register (LCNT0)

31	0
LCNT0	
R/W-0	

LEGEND: R = Readable by the **MVC** instruction; W = Writeable by the **MVC** instruction; -*n* = value after reset

Table 6-325. Loop 0 Iteration Count Register (LCNT0) Field Descriptions

Bit	Field	Value	Description
31-0	LCNT0	0-FFFF FFFFh	Iteration count of Loop 0.

6.2.4.6.14 Loop 1 Start Address Register (LSA1)

The loop 1 start address register (LSA1) contains the start address (byte address) of Loop 1. Note that Loop 1 is the outer loop. The LSA1 is shown in Figure 6-40 and described in Table 6-326.

See Section 6.2.4.17 for more details on hardware loop mechanism and usage of this register.

Figure 6-40. Loop 1 Start Address Register (LSA1)

31	0
LSA1	
R/W-0	

LEGEND: R = Readable by the **MVC** instruction; W = Writeable by the **MVC** instruction; -*n* = value after reset

Table 6-326. Loop 1 Start Address Register (LSA1) Field Descriptions

Bit	Field	Value	Description
31-0	LSA1	0-FFFF FFFFh	Start address of Loop 1 (byte address).

6.2.4.6.15 Loop 1 End Address Register (LEA1)

The loop 1 end address register (LEA1) contains the end address (byte address) of Loop 1. Note that Loop 1 is the outer loop. The LEA1 is shown in Figure 6-41 and described in Table 6-327.

See Section 6.2.4.17 for more details on hardware loop mechanism and usage of this register.

Figure 6-41. Loop 1 End Address Register (LEA1)

31	0
LEA1	
R/W-0	

LEGEND: R = Readable by the **MVC** instruction; W = Writeable by the **MVC** instruction; -*n* = value after reset

Table 6-327. Loop 1 End Address Register (LEA1) Field Descriptions

Bit	Field	Value	Description
31-0	LEA1	0-FFFF FFFFh	End address of Loop 1 (byte address).

6.2.4.6.16 Loop 1 Iteration Count Register (LCNT1)

The loop 1 iteration count register (LCNT1) contains the iteration count of Loop 1. Note that Loop 1 is the outer loop. The LCNT1 is shown in Figure 6-42 and described in Table 6-328.

See Section 6.2.4.17 for more details on hardware loop mechanism and usage of this register.

Figure 6-42. Loop 1 Iteration Count Register (LCNT1)

31	0
LCNT1	
R/W-0	

LEGEND: R = Readable by the **MVC** instruction; W = Writeable by the **MVC** instruction; -*n* = value after reset

Table 6-328. Loop 1 Iteration Count Register (LCNT1) Field Descriptions

Bit	Field	Value	Description
31-0	LCNT1	0-FFFF FFFFh	Iteration count of Loop 1.

6.2.4.6.17 Loop 0 Iteration Count Reload Value Register (LCNT0RLD)

The loop 0 iteration count reload value register (LCNT0RLD) contains the value that the loop 0 iteration count register (LCNT0) is reloaded during the outer loop rewind. LCNT0RLD is automatically updated by hardware when a write to LCNT0 is performed (via **MVC** instruction), thus LCNT0RLD does not need to be explicitly programmed during a loop setup. The LCNT0RLD is shown in Figure 6-43 and described in Table 6-329.

See Section 6.2.4.17 for more details on hardware loop mechanism and usage of this register.

Figure 6-43. Loop 0 Iteration Count Reload Value Register (LCNT0RLD)

31	0
LCNT0RLD	
R/W-0	

LEGEND: R = Readable by the **MVC** instruction; W = Writeable by the **MVC** instruction; -*n* = value after reset

Table 6-329. Loop 0 Iteration Count Reload Value Register (LCNT0RLD) Field Descriptions

Bit	Field	Value	Description
31-0	LCNT0RLD	0-FFFF FFFFh	Reload value of the loop 0 iteration count register (LCNT0).

6.2.4.6.18 Shadow Control Status Register (SCSR)

The shadow control status register (SCSR) contains a copy of the control status register (CSR) (of background code) in the interrupt context (except for an NMI). The SCSR is shown in Figure 6-44 and described in Table 6-330.

On acceptance of an interrupt (except for an NMI), the current state of the control status register (CSR) is copied to SCSR. On execution of a **BIRP** instruction, SCSR is copied back to CSR. SCSR facilitates fast interrupt response. A write to SCSR has no effect on machine operation.

Figure 6-44. Shadow Control Status Register (SCSR)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
R-0															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		DBZ	DBZE	INUM				V	SAT	C	GT	LT	EQ	Rsvd	GIE
R-0		R/W-0	R/W-0	R/W-0				R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-0	R/W-0

LEGEND: R = Readable by the **MVC** instruction; W = Writeable by the **MVC** instruction; -*n* = value after reset

Table 6-330. Shadow Control Status Register (SCSR) Field Descriptions

Bit	Field	Value	Description
31-14	Reserved	0	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.
13	DBZ	0-1	Shadow copy of CSR:DBZ bit.
12	DBZE	0-1	Shadow copy of CSR:DBZE bit.
11-8	INUM	0-Fh	Shadow copy of CSR:INUM field.
7	V	0-1	Shadow copy of CSR:V bit.
6	SAT	0-1	Shadow copy of CSR:SAT bit.
5	C	0-1	Shadow copy of CSR:C bit.
4	GT	0-1	Shadow copy of CSR:GT bit.
3	LT	0-1	Shadow copy of CSR:LT bit.
2	EQ	0-1	Shadow copy of CSR:EQ bit.
1	Reserved	0	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.
0	GIE	0-1	Shadow copy of CSR:GIE bit.

6.2.4.6.19 NMI Shadow Control Status Register (NMISCSR)

The NMI shadow control status register (NMISCSR) contains a copy of the control status register (CSR) (of background code) in the nonmaskable interrupt context. The NMISCSR is shown in Figure 6-45 and described in Table 6-331.

On acceptance of an NMI, the current state of the control status register (CSR) is copied to NMISCSR. On execution of a **BNRP** instruction, NMISCSR is copied back to CSR. NMISCSR facilitates correct servicing of an NMI even in a nested context. A write to NMISCSR has no effect on machine operation.

Figure 6-45. NMI Shadow Control Status Register (NMISCSR)

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16
Reserved															
R-0															

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved		DBZ	DBZE	INUM				V	SAT	C	GT	LT	EQ	Rsvd	GIE
R-0		R/W-0	R/W-0	R/W-0				R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R/W-0	R-0	R/W-0

LEGEND: R = Readable by the **MVC** instruction; W = Writeable by the **MVC** instruction; -*n* = value after reset

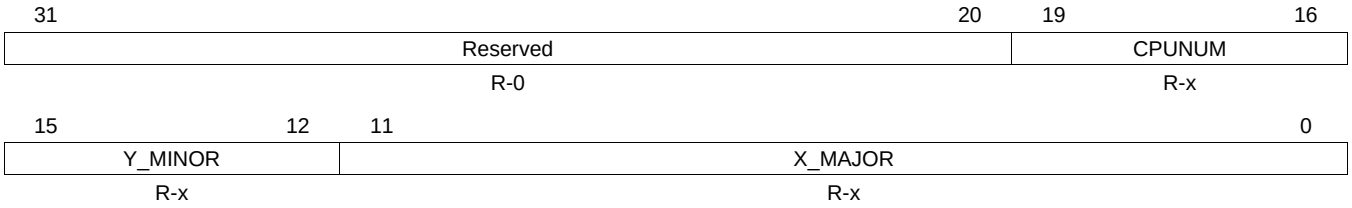
Table 6-331. NMI Shadow Control Status Register (NMISCSR) Field Descriptions

Bit	Field	Value	Description
31-14	Reserved	0	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.
13	DBZ	0-1	Shadow copy of CSR:DBZ bit.
12	DBZE	0-1	Shadow copy of CSR:DBZE bit.
11-8	INUM	0-Fh	Shadow copy of CSR:INUM field.
7	V	0-1	Shadow copy of CSR:V bit.
6	SAT	0-1	Shadow copy of CSR:SAT bit.
5	C	0-1	Shadow copy of CSR:C bit.
4	GT	0-1	Shadow copy of CSR:GT bit.
3	LT	0-1	Shadow copy of CSR:LT bit.
2	EQ	0-1	Shadow copy of CSR:EQ bit.
1	Reserved	0	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.
0	GIE	0-1	Shadow copy of CSR:GIE bit.

6.2.4.6.20 CPU Identification Register (CPUID)

The CPU identification register (CPUID) contains the major and minor version ID. The CPUID is shown in Figure 6-46 and described in Table 6-332.

Figure 6-46. CPU Identification Register (CPUID)



LEGEND: R = Readable by the **MVC** instruction; W = Writeable by the **MVC** instruction; -*n* = value after reset; -x = value is indeterminate after reset

Table 6-332. CPU Identification Register (CPUID) Field Descriptions

Bit	Field	Value	Description
31-20	Reserved	0	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.
19-16	CPUNUM	0-Fh	Reflects the value on the t_cpunum_i[3:0] input port. The input port is sampled on reset release and kept constant – it is not sampled continuously.
15-12	Y_MINOR	0-Fh	Minor revision.
11-0	X_MAJOR	0-FFFh	Major revision.

6.2.4.6.21 Decode Program Counter Register (DPC)

The decode program counter register (DPC) contains the halfword address of the next instruction to be executed (or instruction currently at DEC stage of the pipe). The DPC is shown in Figure 6-47 and described in Table 6-333.

The DPC is a read-only register provided for debug purposes only. A read of DPC while the CPU is halted at a debug event shows the PC of the next instruction to be executed (after the CPU is resumed from a halted state). The debugger can use DPC to determine the halt location. Reading DPC from the debugger while the CPU is running has limited or no use.

An **MVC** instruction with DPC as write address is ignored by the CPU, no errors are generated.

A read of DPC via the **MVC** instruction returns the address of the **MVC** instruction itself. For example, in the following assembly instruction sequence, the read of DPC via the **MVC** instruction returns 0x101 in R1:

```
0x100    ADDD 1, R0, R0
0x101    MVC  DPC, R1   ; R1 gets 0x101
```

Figure 6-47. Decode Program Counter Register (DPC)

31	30	0
Rsvd	Decode Program Counter	
R-0	R-0	

LEGEND: R = Read only; -*n* = value after reset

Table 6-333. Decode Program Counter Register (DPC) Field Descriptions

Bit	Field	Value	Description
31	Reserved	0	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.
30-0	Decode Program Counter	0-7FFF FFFFh	Contains the address of the next instruction to be executed.

6.2.4.6.22 Time Stamp Counter Registers (TSCL and TSCH)

The CPU contains a free running 64-bit counter that advances each CPU clock under normal operation. The counter is accessed as two 32-bit read-only control registers, TSCL (Figure 6-48) and TSCH (Figure 6-49).

Figure 6-48. Time Stamp Counter Register - Low Half (TSCL)

31	0
CPU clock count (32 LSBs of 64-bit value)	
R-0	

LEGEND: R = Readable by the **MVC** instruction; -*n* = value after reset

Figure 6-49. Time Stamp Counter Register - High Half (TSCH)

31	0
CPU clock count (32 MSBs of 64-bit value)	
R-0	

LEGEND: R = Readable by the **MVC** instruction; -*n* = value after reset

6.2.4.6.22.1 Initialization

The counter is cleared to 0 after reset, and counting is disabled. Subsequently, any write to TSCL initializes the counter to 0 (the write data is ignored) and enables the counter. A write to TSCH is completely ignored, without any side effect of enabling the time stamp counter (like that with TSCL). The counter remains enabled until the next CPU functional reset. Once enabled, there is no way to disable the counter or reinitialize the counter to 0.

6.2.4.6.22.2 Enabling Counting

The counter is enabled by writing to TSCL. The value written is ignored. Counting begins in the cycle after the **MVC** instruction executes. If executed with the count disabled, the following code sequence shows the timing of the count starting (assuming no stalls occur in the three cycles shown).

```
MVC   0, TSCL          ; Start the counter
MVC   TSCL, R0         ; R0 = 0
MVC   TSCL, R1         ; R1 = 1
```

6.2.4.6.22.3 Disabling Counting

Once enabled, counting cannot be disabled under program control. Counting is disabled in the following cases:

- After exiting the reset state.
- When the CPU is fully powered down.

6.2.4.6.22.4 Reading the Counter

Reading the full 64-bit count takes two sequential **MVC** instructions. A read from TSCL causes the upper 32 bits of the current count (the value of the counter when the corresponding **MVC** instruction is at Fetch/DEC stage) to be copied into TSCH. In normal operation, only this snapshot of the upper half of the 64-bit count is available to the programmer. The value read will always be the value copied at the cycle of the last **MVC TSCL,** *reg* instruction. If it is read with no TSCL reads having taken place since reset, then the reset value of 0 is read.

When reading the full 64-bit value, it must be ensured that no interrupts are serviced between the two **MVC** instructions if an ISR is allowed to make use of the time stamp counter. There is no way for an ISR to restore the previous value of TSCH (snapshot) if it reads TSCL, since a new snapshot is performed.

The following example shows the correct and recommended way of obtaining the full 64-bit value of the time stamp counter:

```
// Disable interrupts
MVC   CSR, R7          ; Get CSR into R7
CLR   0, 0, R7, R7     ; Clear R7[0]
MVC   R7, CSR          ; Copy R7 back to CSR (clears GIE)

// Read Counter
MVC   TSCL, R0         ; read lower 32b into R0
MVC   TSCH, R1         ; read upper 32b into R1

// Enable interrupts
MVC   CSR, R7          ; Get CSR into R7
SET   0, 0, R7, R7     ; Set R7[0]
MVC   R7, CSR          ; Copy R7 back to CSR (sets GIE)
```

6.2.4.7 CPU Shadow Registers

The ARP32 CPU implements a shadow copy for the following registers:

- All architectural registers: R0-R7
- Loop control registers: LSA*n*, LEA*n*, LCNT*n*

The shadow registers are used to save-off the main registers during an interrupt processing. When an interrupt is taken, the main registers are copied onto the corresponding shadow register by the CPU along with saving the interrupt return address in IRP/NRP. The original registers are restored back upon a return from interrupt – as a result of a **BIRP** or **BNRP** instruction execution.

As a result of this, as long as interrupts are not allowed to be nested, interrupt handler routines in the ARP32 CPU do not need to save of any registers on the stack, this significantly reduces the interrupt processing latency.

Table 6-334 provides the list of the shadow registers in the ARP32 CPU.

Table 6-334. CPU Shadow Registers

Acronym	Register Name	Description
SR0	Shadow R0	Shadow copy of R0
SR1	Shadow R1	Shadow copy of R1
SR2	Shadow R2	Shadow copy of R2
SR3	Shadow R3	Shadow copy of R3
SR4	Shadow R4	Shadow copy of R4
SR5	Shadow R5	Shadow copy of R5
SR6	Shadow R6	Shadow copy of R6
SR7	Shadow R7	Shadow copy of R7
SLSA0	Shadow Loop 0 Start Address	Shadow copy of LSA0
SLEA0	Shadow Loop 0 End Address	Shadow copy of LEA0
SLCNT0	Shadow Loop 0 Iteration Count	Shadow copy of LCNT0
SLSA1	Shadow Loop 1 Start Address	Shadow copy of LSA1
SLEA1	Shadow Loop 1 End Address	Shadow copy of LEA1
SLCNT1	Shadow Loop 1 Iteration Count	Shadow copy of LCNT1
SLCNT0RLD	Shadow Loop 0 Iteration Count Reload	Shadow copy of LCNT0RLD

Shadow registers are access via the **MVS** instruction:

- **MVS** *sreg, areg* - Read shadow register into an architectural register
- **MVS** *areg, sreg* - Write shadow register from an architectural register

The **MVS** instruction is executed by the S unit. Any write to the shadow registers (via a **MVS** *areg, sreg* instruction) has two exposed delay slots - a subsequent read from the same *sreg* (via a **MVS** *areg, sreg* instruction) must be separated by at least two instructions. Otherwise, the old value of *sreg* is read.

The read result (of a **MVS** *sreg, areg* instruction) is bypassed appropriately to a subsequent instruction using the same *areg* – there is no delay slot or stall in this case.

The following examples illustrate the behavior:

```
// Update SR0 and then read back (assume SR0 = 0x100, R0 = 0x200)
MVS   R0, SR0         ; Move R0 to SR0
NOP                   ; NOP to fill delay slot #1
MVS   SR0, R7         ; Move SR0 to R7, R7 = 0x100 (old value)


// Update SR0 and then read back (assume SR0 = 0x100, R0 = 0x200)
MVS   R0, SR0         ; Move R0 to SR0
NOP                   ; NOP to fill delay slot #1
NOP                   ; NOP to fill delay slot #2
MVS   SR0, R7         ; Move SR0 to R7, R7 = 0x200 (new value)
```

6.2.4.8 Functional Units

The ARP32 CPU contains the following main functional units:

- **L-Unit:** Performs all logical operations (AND/OR/XOR/NOT/shift and minimum/maximum), some bit level operations like bit rotation, signed/unsigned bit extraction, bit reversal, bit clear, bit set operations, left most bit detection, and saturation operation.
- **S-Unit:** Performs the move operations.
- **D-Unit:** Performs arithmetic operations that include compare less/greater than instructions, address calculation for load/store instructions, PC calculations for branch instructions, and stack pointer increment/decrement for **PUSH/POP** and **CALL/RETURN** instructions.
- **M-Unit:** Performs multiplication, division, and modulo operations.

6.2.4.9 Instruction Fetch

The ARP32 ISA has instructions that are either 16 bit or 32 bit; the mix being chosen to achieve optimal code size without sacrificing the performance (of the amount of actual work done per instruction per cycle).

The ARP32 CPU allows 16/32-bit instructions to be mixed freely without any overhead or switching between instruction decoding mode. The program counter (PC) always points to a halfword address aligned with an instruction boundary. However, the program fetch is always aligned with a word boundary. The ARP32 CPU always requests a 32-bit word from a word-aligned address over its 32-bit wide program interface. Internally, the CPU maintains a 16 bit single entry fetch buffer (FetchBuffer[15:0]) to store an unused instruction word or part of an instruction word.

This allows a simpler program memory subsystem design and allows area, power efficient implementation of tightly coupled program memory subsystem without incurring any overhead on code density.

Since instructions in the ARP32 CPU are either 16 bit or 32 bit and they are freely mixable, any 32-bit word fetched from the program memory (FetchWord[31:0]) may have the following cases:

- **The FetchWord contains two 16-bit instructions:** the first instruction (FetchWord[15:0]) is sent to the instruction decoder and the second instruction (FetchWord[31:16]) is stored in the FetchBuffer. The stored instruction is then sent to the instruction decoder at the next decode cycle. No program request is sent for a cycle (as the fetch buffer already contains the next instruction to be executed) unless there is a branch.
- **The FetchWord contains one 32-bit instruction:** the entire FetchWord is sent to the instruction decoder and the FetchBuffer is invalidated.
- **The FetchWord contains one 16-bit instruction and the lower halfword of the next 32-bit instruction:** the valid 16-bit instruction (in FetchWord[15:0]) is sent to the instruction decoder and the lower halfword of the next 32-bit instruction is stored in the FetchBuffer. A second fetch must be performed to fetch the upper halfword of the 32-bit instruction before it is decoded. The request for this second fetch is sent in the current cycle. When the corresponding FetchWord is available (for example, at the next cycle), its lower halfword constitutes the full 32-bit instruction word and is sent to the instruction decoder. The unused upper halfword of the current FetchWord is stored back to the FetchBuffer again.

Thus, for a sequential program with an arbitrary mix of 16-bit and 32-bit instructions, the instruction is executed without a single cycle of stall (due to unavailability of a whole instruction word for decoding) in the pipeline. However, if the target address of a program discontinuity (due to branch, call, return interrupt) is not word aligned and there exists a 32-bit instruction at the target address, a single 32-bit aligned fetch is not sufficient to fetch a full 32-bit instruction word. In this case, there is a stall/bubble cycle in the CPU pipeline. The fetch engine of the CPU issues an additional instruction fetch request at the next cycle to fulfill the required 32-bit instruction fetch and the CPU pipeline continues normally.

6.2.4.10 Alignment of 32-bit Instructions

The fetch engine of the ARP32 CPU supports 16-bit and 32-bit instructions to be aligned at any halfword boundary. For a sequential execution, CPU fetches and executes from such instruction stream without incurring any stall cycles. However, if a program discontinuity target contains an unaligned 32-bit instruction, the CPU stalls for a cycle.

NOTE: The following requirements are for efficient code generation, not functionally correct code generation. The ARP32 CPU supports having a 32-bit instruction at an unaligned discontinuity target – with the associated stall overhead. There are cases where this cannot be avoided – for example, an interrupt return to an unaligned 32-bit instruction.

This requires that for the most efficient programming of the ARP32 CPU, all programs contain 32-bit instructions to be aligned on a 32-bit boundary in the program memory. For the ARP32 C/C++ compiler/toolchain, this means that a 32-bit alignment is maintained for the following cases:

- All assembly level basic blocks that start with a 32-bit instruction
- All functions that start with a 32-bit instruction
- All HLA loop rewind targets (content of LSA*n* registers) that contains a 32-bit instruction

If the previous conditions are met, the following most frequent cases where a program discontinuity happens, do not incur any additional stalls:

- Use of branch/call immediate instructions (**Bcc** *scst9*, **Bcc** *scst16*, or **CALL** *scst22*) for general program flow control
- Use of register call (**CALL** *src1*) for long calls to labels or function calls using function pointers
- Use of HLA for loop constructs and corresponding branch

6.2.4.11 Instruction Execution in Branch Delay Slot

The ARP32 CPU allows one delay slot for the following control flow instructions:

- branch (**Bcc** *scst9*, **Bcc** *scst16*, or **Bcc** *dst*)
- call (**CALL** *scst22* or **CALL** *src1*)
- return (**RET**)

There is no delay slot for SWI.

An instruction is placed in a branch delay slot as long as the restrictions mentioned in Section 6.2.5.3 are maintained.

6.2.4.12 Address Space

The ARP32 CPU uses the convention that both instruction and data memory is byte addressable. The ARP32 CPU is capable of supporting full 32-bit instruction and data memory address space.

The ARP32 core is a pure Harvard architecture, separate instruction and data bus access the program and data memory space. However, while integrating the core, it is possible to unify the program and data space to make it a unified (modified Harvard) memory architecture.

6.2.4.13 Program Counter Convention

In the ARP32 CPU, the PC (program counter register) always contains halfword addresses. Correspondingly, all PC related registers (IRP, NRP) also contain a halfword address. The PC conventions used in pseudo code and other program snippets presume this convention, PC = PC + 1 indicates advancement of the PC by 1 entry, with each entry containing 2 bytes.

However, since program fetch is always of word size and word aligned, a word address is sent over the instruction fetch address bus (cpu_imem_raddr_o[31:2]).

This convention is used in most of the cases in the ISA where a PC related operation is implied. However, there are some use cases where an absolute byte address is needed to reference a location in program memory. Based on these considerations, the following rules/conventions are applied to PC related instructions and operations; the ARP32 compiler/toolchain and assembly code must maintain the following rules/conventions:

- For immediate branch instructions (**Bcc** *scst9* or **Bcc** *scst16*), the branch offset is treated as halfword offsets. This offset is directly added to the PC to calculate effective branch address (see immediate branch instructions).
- For immediate call instructions (**CALL** *scst22*), the call address is treated as PC relative halfword offset. This offset is directly added to the PC to calculate effective call address (see immediate call instructions).
- For register branch or call instructions (**Bcc** *src1* or **CALL** *src1*), the register content is treated as the absolute byte address of the branch/call destination. This address is first converted to a halfword address and then loaded to the PC.
- For set loop address instruction (**SLA** *ucst16, creg*), the immediate offset is treated as a PC relative halfword offset. This offset is directly added to the PC to calculate effective branch address (see immediate branch instructions).
- While setting up address in loop address registers (LSA*n*, LEA*n*) using the **MVC/MVCH** instructions (immediate or register forms), the value set in the loop address register is the absolute halfword address (of loop start/end instruction).
- A call instruction (**CALL** *src1* or **CALL** *scst22*) saves off a return address to the stack that is a halfword address. A corresponding return instruction (**RET**) loads this address directly onto the PC to return to the correct location.
- An interrupt saves off a return address to the IRP/NRP register that is a halfword address. A corresponding return instruction (**BIRP** or **BNRP**) loads this address directly onto the PC to return to the correct location.
- Interrupt Service Table (IST) entries are treated as a halfword address to the interrupt handler routine.

The PC is an architectural register, that is, it contains machine state, but is not directly accessible through the instruction set. Instruction execution has an effect on the PC, but the current PC value can not be read or written explicitly.

6.2.4.14 Stack Pointer Convention

The ARP32 CPU has a 32-bit stack pointer register (SP) and its content is always assumed to be a byte address. Any address computation using the SP value considers content of the SP as a byte address while computing an effective address of a byte/halfword/word access.

The ARP32 CPU is a descending stack machine. The ARP32 CPU programming model requires that the programmer and/or the compiler/toolchain maintain the following conventions:

- Stack pointer (SP) is initialized to a high address, it reduces to a lower address as it proceeds through function calls during program execution.
- Stack pointer (SP) must be always aligned to a 4-byte word address. It is software responsibility to keep the SP always aligned to 4-byte word address when allocating/de-allocating stack – there is no hardware support to do this alignment automatically. Since the ARP32 CPU does not support unaligned access, a stack pointer value not aligned to 4-byte boundary may result in access to unaligned addresses and thus produce unexpected result.
- In all cases where the CPU itself modifies the SP (call, return, **LDRF/STRF**), it always increments/decrements it by 4 (word aligned).

- Stack pointer (SP) always points to the next free location where an entry is stacked. For pushing anything onto stack, the CPU writes to the location the SP is currently pointing to and then post-decrements the SP. Similarly, while popping anything from the stack, the CPU pre-decrements the SP and then reads from the location (pre-decremented) the SP points to. For example:
  - Every **CALL** saves the return address to the location the SP is currently pointing to and then post-decrements the SP.
  - Every **RET** pre-decrements the SP and then loads the return address from the location updated SP points to.
- To allocate stack in a function (in the function prolog), reduce the SP by an amount equal to local stack frame size.
- Local variables, input arguments to functions and spilled variables all are accessed via the SP relative positive offset load/store instructions.
- To de-allocate stack in a function (in the function epilog), increase the SP by an amount equal to local stack frame size.

The ARP32 core does not save any context on stack during interrupt processing. The same stack layout and convention is applied to the interrupt handler function/procedure – except for additional register save requirements.

6.2.4.15 Global Data Pointer Convention

Global data pointer register (GDP) contains a byte address of the start of global data section. It is computed by the linker and assigned to a linker symbol that the boot routine must be used to initialize GDP.

6.2.4.16 Conditional Execution

On the ARP32 CPU, conditional execution is available only by using conditional branches.

Most data processing instructions and all compare instructions update the condition flags in control status register: CSR[2]EQ, CSR[3]LT, CSR[4]GT. Some instructions update all flags and some instructions only update a subset. See the instruction descriptions for the flags they affect. Instructions update the status bit relevant to it; other status bits are left unchanged.

Conditional branch instructions are executed, based on the condition flags set in another instruction, either:

- immediately after the instruction that updated the flags
- after any number of intervening instructions that have not updated the flags

See conditional branch instruction for more details.

6.2.4.17 Hardware Loop Acceleration

6.2.4.17.1 Overview

The ARP32 CPU implements a hardware loop assist (HLA) function to reduce cycles in critical inner loops that are typically spent in loop control operations such as loop index increment, decrement, compare, branch operations. Using an HLA mechanism, a zero-overhead rewind to the top of the loop is achieved for up to two levels of nested loops. The structure and operation of the HLA is conformable for easy mapping of loop constructs (for, while, do, etc.) available in C/C++.

Up to two levels of nested loops are supported. Additional levels of nesting are possible through standard software techniques.

In a two level nested loop construct, the inner most level is termed as Loop 0, while the outer level is termed Loop 1. For a single-level nested loop, only Loop 0 is used. Control registers associated with Loop 0 and Loop 1 control the operation of each level.

6.2.4.17.2 Loop Registers

HLA operation is setup and controlled by six control registers listed in Table 6-335. Note that LSA*n* and LEA*n* registers contain/reflect the byte address.

Table 6-335. Hardware Loop Control Registers

Acronym	Register Name	Description
LSA0	Loop 0 Start Address Register	Contains the Start Address of Loop 0
LEA0	Loop 0 End Address Register	Contains the End Address of Loop 0
LCNT0	Loop 0 Iteration Count Register	Contains the Iteration Count of Loop 0
LSA1	Loop 1 Start Address Register	Contains the Start Address of Loop 1
LEA1	Loop 1 End Address Register	Contains the End Address of Loop 1
LCNT1	Loop 1 Iteration Count Register	Contains the Iteration Count of Loop 1
LCNT0RLD	Loop 0 Iteration Count Reload Value Register	Contains the LCNT0 reload value during outer loop rewind

6.2.4.17.3 Loop Setup Instructions

Loop control registers are accessed via the **MVC/MVCH** instructions like any other control registers.

To simplify and optimize loop setup in the most frequent compiler generated use cases, a separate instruction called set loop address (**SLA** *ucst16, creg*) is provided. The **SLA** instruction takes a PC-relative immediate positive halfword offset (*ucst16*) and writes the byte address of the location indicated by the PC + *ucst16* value to the LSA*n* and LEA*n* registers.

Loop setup occurs in the following order:

- For a single-level loop (only Loop 0):
  1. Setup LSA0
  2. Setup LEA0
  3. Setup LCNT0
- For a two-level loop (Loop 0 and Loop 1):
  1. Setup LSA1
  2. Setup LEA1
  3. Setup LCNT1
  4. Setup LSA0
  5. Setup LEA0
  6. Setup LCNT0

HLA is not assured to work if the above order is not maintained.

6.2.4.17.4 Loop Operation

The loop setup process for each level of nesting consists of the following (in the specified order):

1. Setup loop start address register (LSAn) with the byte address of the first instruction in the corresponding loop.
2. Setup loop end address register (LEAn) with the byte address of the last instruction in the corresponding loop.
3. Setup loop iteration count register (LCNTn) with the intended loop iteration count.

A loop becomes active as soon as a value greater than 1 is programmed in LCNTn. The LSAn and LEAn registers are written when the corresponding **MVC/SLA** instruction is executed but the values are not used by the processor (for detecting loop rewind condition) until the corresponding loop becomes active.

When a loop becomes active (LCNTn > 1), the PC of the instruction at DEC phase is checked with the LEAn register. If they match, the CPU takes a zero-cycle overhead branch to the top of the loop provided by the LSAn registers. The program fetch request to LSAn is placed when the last instruction of a loop is at DEC. Thus there is no delay slot for the loop rewind branch, neither any stall or null cycle.

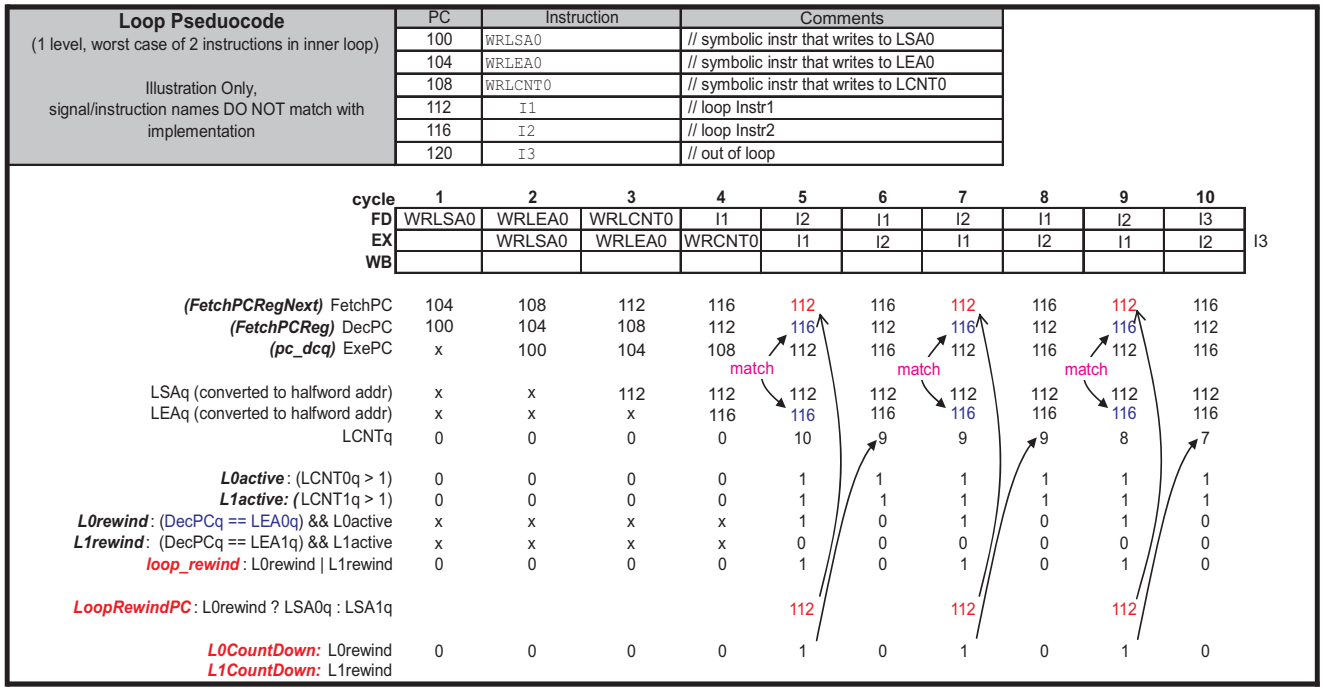
For LCNTn = x, the loop rewinds x - 1 times, effectively the loop body executes total x times (x - 1 rewinds and the last fall through pass). For LCNTn = 1, the loop body executes once (zero rewind). For LCNTn = 0, the loop is not considered active – the CPU just linearly executes – however, as a result, the loop body executes once (linear fall through). Note that LCNTn = 0 does not mean loop body would not be executed.

During operation, the LCNTn registers are directly updated (decremented) with the remaining iteration count. The loop counters are updated along with the corresponding rewind-branch. At any time LCNTn is read to determine how many iterations of the corresponding loop are still to be executed. When each level completes, the corresponding LCNTn registers become zero, the LSAn/LEAn registers are left unchanged. Note that for LCNTn = 1, the loop body executes once and the loop counter is decremented when the last instruction of the corresponding loop is executed.

The LCNT0RLD register is updated automatically by the CPU whenever a value is written to LCNT0 (for example, during loop setup). For a two-level nested loop, once the inner most level iteration completes and the CPU rewinds to the top of the outer loop, the LCNT0 is reloaded with the content of LCNT0RLD. Note that even if LCNT0RLD is a programmer visible control register, it acts as a shadow register of LCNT0 (updated on any write to LCNT0) and must not be explicitly written during a loop setup.

There are no restrictions in embedding an HLA construct or instruction sequence, consisting of one or two levels of nested loops, into a larger level of nested loop implemented by normal software techniques.

Figure 6-50. Loop Operation



6.2.4.17.5 Call and Branch within Loop Context

Branches (**Bcc** *scst9*, **Bcc** *scst16*, or **Bcc** *dst*) within the loop are allowed, provided the conditions explicitly mentioned in Section 6.2.5.3.2 are met.

Calls (**CALL** *scst22* or **CALL** *dst*) within the loop are generally not supported as the loop registers are not explicitly saved off when a CALL is executed within the loop context. However, software uses **CALL** instructions if the call tree does not contain any HLA usage within the caller code.

6.2.4.17.6 Dynamic Changes to Loop Iteration Count

A write to the LCNTn register within the loop body (of either of the loops) is used to dynamically change the iteration count, including pre-maturely terminating the corresponding level. Such a write to LCNTn instruction must be executed at least 2 instructions before the last (ending) instruction of the corresponding loop (x).

It is also possible to skip inner loop dynamically (per iteration of the outer loop) by simply branching across the inner loop within the outer loop context. It is not necessary to explicitly set the inner loop count to 0/1 (loop inactive values). Similarly, a branch out of the outer loop context is possible without setting the outer loop count to inactive values. This relies on the fact that even if the inner or outer loop count has not been set to an inactive value, a loop rewind des not happen unless the loop end PC (LEAn) is encountered.

These features (dynamically changing the loop iteration count, skipping inner/outer loop via branches) are used by the compiler to implement constructs such as last/continue/break and alike.

6.2.4.17.7 Interrupt Processing During HLA

Interrupts are not taken while the last instruction of a loop level is at DEC. Interrupts are taken after the loop rewind happens. The return address saved off during the interrupt processing points to the loop rewind address (LSAn). Consequently, a return from interrupt returns to the last rewind address (LSAn).

When taking an interrupt (irrespective if an HLA is active), the CPU saves off the six loop registers in the corresponding shadow registers. They are restored when interrupts are returned via the **BIRP/BNRP** instructions.

6.2.4.17.8 HLA Usage in Interrupt Context

Since loop context is fully saved by hardware while taking an interrupt, HLA is used within an ISR context freely without requiring any additional context save. For allowing nested interrupts, the shadow copies of HLA registers must be saved off to the stack explicitly along with other shadow registers. This helps achieve very-low interrupt latency in the ARP32 CPU under a single-level-nested-interrupt use model.

6.2.4.17.9 HLA Usage Restrictions

There are certain restrictions on HLA usage, see Section 6.2.5.3.2.

6.2.4.17.10 HLA Mapping Examples

6.2.4.17.10.1 Loops With Single Level of Nesting

Example 6-2. C memset-like Loop, Single Level, Minimum Instructions

Pseudo C code:

```
int   i, myarray[100];
for (i=0; i< 100; i++) {
    myarray[i] = 1;
}
```

Table 6-336. Example 1 of Generated Assembly Code (relevant instructions only)

Byte Address	Instruction	Comment
0x100	`MVKS 1, R0`	Write data
0x102	`SLA 6, LSA0`	First instr of loop0 is at #6 halfword offset `LSA0 = 0x102 + hex(6 x 2) = 0x10E`
0x106	`SLA 5, LEA0`	Last instr of loop0 is at #5 halfword offset `LEA0 = 0x106 + hex(5 x 2) = 0x110`
0x10A	`MVC 100, LCNT0`	iter count = 100
0x10E	`STW R0, *++R1(1)`	First instr of Loop 0
0x110	`NOP`	Last instr of Loop 0 (padding needed)

Table 6-337. Example 2 of Generated Assembly Code (relevant instructions only)

Byte Address	Instruction	Comment
0x100	`MVKS 1, R0`	Write data
0x102	`SLA 6, LSA0`	First instr of loop0 is at #6 halfword offset `LSA0 = 0x102 + hex(6 x 2) = 0x10E`
0x106	`SLA 5, LEA0`	Last instr of loop0 is at #5 halfword offset `LEA0 = 0x106 + hex(5 x 2) = 0x110`
0x10A	`MVC 100, LCNT0`	iter count = 100
0x10E	`STW R0, *+R1(0)`	First instr of Loop 0
0x110	`ADD 4, R1, R1`	Last instr of Loop 0 (padding needed)

Example 6-3. C memcpy-like Loop, Single Level, Minimum Instructions

Pseudo C code:

```
void MyMemcpyByWord (void *ptr1, void *ptr2, int size){
  int ii, dbwSize = size>>2;
  unsigned int *dst = ptr1;
  unsigned int *src = ptr2;
  for (ii=0 ; ii< dbwSize ; ii++)
  dst[ii] = src[ii];
}
```

Table 6-338. Example 1 of Generated Assembly Code (relevant instructions only)

Byte Address	Instruction	Comment
0x100	`SLA 5, LSA0`	First instr of loop0 is at #5 halfword offset `LSA0 = 0x100 + hex(5 x 2) = 0x10A`
0x104	`SLA 4, LEA0`	Last instr of loop0 is at #4 halfword offset `LEA0 = 0x104 + hex(4 x 2) = 0x10C`
0x108	`MVC R1, LCNT0`	iter count = “size” (passed via R1)
0x10A	`LDW *++R2(1), R3`	Load word from source (Loop 0, instr 1)
0x10C	`STW R3, *++R4(1)`	Store word to dest (Loop 0, instr 2)

Table 6-339. Example 2 of Generated Assembly Code (relevant instructions only)

Byte Address	Instruction	Comment
0x100	`SLA 5, LSA0`	First instr of loop0 is at #5 halfword offset `LSA0 = 0x100 + hex(5 x 2) = 0x10A`
0x104	`SLA 6, LEA0`	Last instr of loop0 is at #6 halfword offset `LEA0 = 0x104 + hex(6 x 2) = 0x110`
0x108	`MVC R1, LCNT0`	iter count = “size” (passed via R1)
0x10A	`LDW *+R2(0), R3`	Load word from source (Loop 0, instr 1)
0x10C	`ADD 4, R2, R2`	Increment src pointer
0x10E	`STW R3, *+R4(0)`	Store word to dest
0x110	`ADD 4, R4, R4`	Increment dest pointer (Loop 0, instr 4)

6.2.4.17.10.2 Loops With Two Levels of Nesting

Example 6-4. Two-level Nesting, Both Loops Ending at Same Instruction

Pseudo C code:

```
For (i=0; i<20; i++) {
   Instruction1      // size 16b
   Instruction2      // size 32b
   Instruction3      // size 32b
   Instruction4      // size 16b
   For (j=0; j<10; j++) {
      Instruction5   // size 16b
      Instruction6   // size 32b
      Instruction7   // size 16b
   }
}
```

Table 6-340. Example of Generated Assembly Code (relevant instructions only)

Byte Address	Instruction	Comment
0x100	`SLA 18, LSA1`	First instr of loop1 is at #18 halfword offset `LSA1 = 0x100 + hex(18 x 2) = 0x124`
0x104	`SLA 19, LEA1`	Last instr of loop0 is at #19 halfword offset `LEA0 = 0x104 + hex(19 x 2) = 0x12A`
0x108	`MVC 20, LCNT1`	iter count (=20) of loop1
0x10C	`SLA 6, LSA0`	First instr of loop0 is at #6 halfword offset `LSA1 = 0x10C + hex(6 x 2) = 0x118`
0x110	`SLA 9, LEA0`	First instr of loop0 is at #9 halfword offset `LEA0 = 0x110 + hex(9 x 2) = 0x122`
0x114	`MVC 10, LCNT0`	iter count (=10) of loop0
0x118	`Instruction1`	Loop1 start
0x11A	`Instruction2`	
0x11E	`Instruction3`	
0x122	`Instruction4`	
0x124	`Instruction5`	Loop0 start
0x126	`Instruction6`	
0x12A	`Instruction7`	Loop0 end, Loop1 end

Example 6-5. Two-level Nesting, Different Ending Instructions for Two Levels

Pseudo C code:

```
For (i=0; i<20; i++) {
   Instruction1      // size 16b
   Instruction2      // size 32b
   Instruction3      // size 32b
   Instruction4      // size 16b
   For (j=0; j<10; j++) {
      Instruction5   // size 16b
      Instruction6   // size 32b
      Instruction7   // size 16b
   }
   Instruction8      // size 32b
   Instruction9      // size 32b
   Instruction10     // size 16b
}
```

Table 6-341. Example of Generated Assembly Code (relevant instructions only)

Byte Address	Instruction	Comment
0x100	`SLA 18, LSA1`	First instr of loop1 is at #18 halfword offset `LSA1 = 0x100 + hex(18 x 2) = 0x124`
0x104	`SLA 19, LEA1`	Last instr of loop1 is at #19 halfword offset `LEA0 = 0x104 + hex(19 x 2) = 0x12A`
0x108	`MVC 20, LCNT1`	iter count (=20) of loop1
0x10C	`SLA 6, LSA0`	First instr of loop0 is at #6 halfword offset `LSA1 = 0x10C + hex(6 x 2) = 0x118`
0x110	`SLA 18, LEA0`	First instr of loop0 is at #18 halfword offset `LEA0 = 0x110 + hex(18 x 2) = 0x134`
0x114	`MVC 10, LCNT0`	iter count (=10) of loop0
0x118	`Instruction1`	Loop1 start
0x11A	`Instruction2`	
0x11E	`Instruction3`	
0x122	`Instruction4`	
0x124	`Instruction5`	Loop0 start
0x126	`Instruction6`	
0x12A	`Instruction7`	Loop0 end
0x12C	`Instruction8`	
0x130	`Instruction9`	
0x134	`Instruction10`	Loop1 end

6.2.4.18 Interrupts

6.2.4.18.1 Overview

The ARP32 CPU supports various types of interrupts that include: reset, a non-maskable interrupt (NMI), 12 maskable interrupts (INT15-INT4), an undefined instruction interrupt (UNDEF), and a software interrupt (SWI). The following registers control the CPU behavior on receipt of an interrupt:

- Control Status Register (CSR)
- Interrupt Enable Register (IER)
- Interrupt Flag Register (IFR)
- Interrupt Set Register (ISR)
- Interrupt Clear Register (ICR)
- Nonmaskable Interrupt Return Pointer Register (NRP)
- Interrupt Return Pointer Register (IRP)

The SWI and UNDEF interrupts are special cases that do not have an associated input pin. The SWI interrupt mechanism is activated by decoding of the **SWI** instruction. The UNDEF interrupt is activated by detection of an undefined instruction.

On acknowledge of an enabled interrupt, the ARP32 CPU loads the contents of the interrupt service table entry associated with the interrupt into the PC.

Each interrupt type is discussed in the following sections; Table 6-342 summarizes the ARP32 CPU interrupts.

Table 6-342. Interrupt Summary

Interrupt Name	Vector Address (byte)	Input Pin	Enable Control	Interrupt Return Pointer Register	Interrupt Return Instruction to be Used	Context Save and Restore Actions	
						By Hardware	By Software
Reset	00h	cpu_reset_i	Always Enabled	NA	NA	NA	NA
NMI	04h	cpu_nmi_i	IER:NMIE	NRP	BNRP	CSR to NMISCSR	R0-R7 to Stack, LSA*n* to Stack, LEA*n* to Stack, LCNT*n* to Stack
SWI	08h	None	Always Enabled	IRP	BIRP	CSR to SCSR, R0-R7 to SR0-SR7, LSA*n* to SLSA*n*, LEA*n* to SLEA*n*, LCNT*n* to SLCNT*n*	None
UNDEF	0Ch	None	Always Enabled	IRP	BIRP	CSR to SCSR, R0-R7 to SR0-SR7, LSA*n* to SLSA*n*, LEA*n* to SLEA*n*, LCNT*n* to SLCNT*n*	None
INT4	10h	cpu_int4_i	CSR:GIE; IER:NMIE; IER:IE4	IRP	BIRP	CSR to SCSR, R0-R7 to SR0-SR7, LSA*n* to SLSA*n*, LEA*n* to SLEA*n*, LCNT*n* to SLCNT*n*	None
INT5	14h	cpu_int5_i	CSR:GIE; IER:NMIE; IER:IE5	IRP	BIRP	CSR to SCSR, R0-R7 to SR0-SR7, LSA*n* to SLSA*n*, LEA*n* to SLEA*n*, LCNT*n* to SLCNT*n*	None
INT6	18h	cpu_int6_i	CSR:GIE; IER:NMIE; IER:IE6	IRP	BIRP	CSR to SCSR, R0-R7 to SR0-SR7, LSA*n* to SLSA*n*, LEA*n* to SLEA*n*, LCNT*n* to SLCNT*n*	None

Table 6-342. Interrupt Summary (continued)

Interrupt Name	Vector Address (byte)	Input Pin	Enable Control	Interrupt Return Pointer Register	Interrupt Return Instruction to be Used	Context Save and Restore Actions	
						By Hardware	By Software
INT7	1Ch	cpu_int7_i	CSR:GIE; IER:NMIE; IER:IE7	IRP	BIRP	CSR to SCSR, R0-R7 to SR0-SR7, LSA*n* to SLSA*n*, LEA*n* to SLEA*n*, LCNT*n* to SLCNT*n*	None
INT8	20h	cpu_int8_i	CSR:GIE; IER:NMIE; IER:IE8	IRP	BIRP	CSR to SCSR, R0-R7 to SR0-SR7, LSA*n* to SLSA*n*, LEA*n* to SLEA*n*, LCNT*n* to SLCNT*n*	None
INT9	24h	cpu_int9_i	CSR:GIE; IER:NMIE; IER:IE9	IRP	BIRP	CSR to SCSR, R0-R7 to SR0-SR7, LSA*n* to SLSA*n*, LEA*n* to SLEA*n*, LCNT*n* to SLCNT*n*	None
INT10	28h	cpu_int10_i	CSR:GIE; IER:NMIE; IER:IE10	IRP	BIRP	CSR to SCSR, R0-R7 to SR0-SR7, LSA*n* to SLSA*n*, LEA*n* to SLEA*n*, LCNT*n* to SLCNT*n*	None
INT11	2Ch	cpu_int11_i	CSR:GIE; IER:NMIE; IER:IE11	IRP	BIRP	CSR to SCSR, R0-R7 to SR0-SR7, LSA*n* to SLSA*n*, LEA*n* to SLEA*n*, LCNT*n* to SLCNT*n*	None
INT12	30h	cpu_int12_i	CSR:GIE; IER:NMIE; IER:IE12	IRP	BIRP	CSR to SCSR, R0-R7 to SR0-SR7, LSA*n* to SLSA*n*, LEA*n* to SLEA*n*, LCNT*n* to SLCNT*n*	None
INT13	34h	cpu_int13_i	CSR:GIE; IER:NMIE; IER:IE13	IRP	BIRP	CSR to SCSR, R0-R7 to SR0-SR7, LSA*n* to SLSA*n*, LEA*n* to SLEA*n*, LCNT*n* to SLCNT*n*	None
INT14	38h	cpu_int14_i	CSR:GIE; IER:NMIE; IER:IE14	IRP	BIRP	CSR to SCSR, R0-R7 to SR0-SR7, LSA*n* to SLSA*n*, LEA*n* to SLEA*n*, LCNT*n* to SLCNT*n*	None
INT15	3Ch	cpu_int15_i	CSR:GIE; IER:NMIE; IER:IE15	IRP	BIRP	CSR to SCSR, R0-R7 to SR0-SR7, LSA*n* to SLSA*n*, LEA*n* to SLEA*n*, LCNT*n* to SLCNT*n*	None

6.2.4.18.2 Interrupt Processing

Figure 6-51 provides an illustration of different actions and events of interest during an interrupt processing. In this case, both INT4 and INT5 were asserted simultaneously, INT4 was selected as the winner and was taken after few cycles when all enable conditions (CSR[0]GIE, IER[1]NMIE, IER[n]) were met.

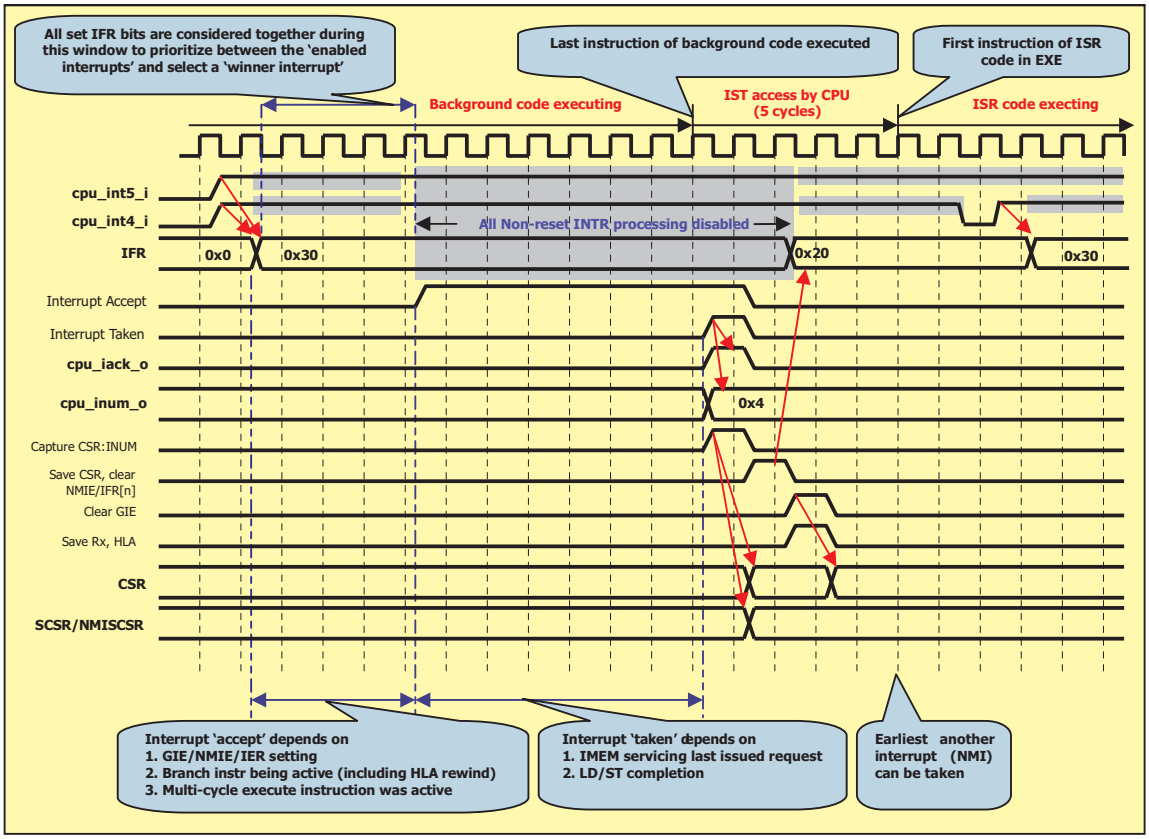
Note that the CPU entity signals/registers are shown in **bold** font; internal (and sometimes just conceptual) signals are shown in normal font.

The conceptual signal “Interrupt Accept” shows when a particular interrupt (in this case, INT4) is accepted to be taken next. This is generated by considering that all set bits of IFR, the interrupt priority, the enable conditions, and a safe cycle boundary within the CPU, where accepting an interrupt is safe.

The conceptual signal “Interrupt Taken” shows when a particular interrupt (in this case, INT4) is actually taken. The actions corresponding to an interrupt processing (IST access, context save, etc.) starts to take place within the CPU after this condition is achieved.

The delay between ‘Interrupt Accept’ and ‘Interrupt Taken’ is due to pending memory transaction on CPU instruction and data interfaces. The CPU waits for all outstanding instruction and data memory requests to be serviced before staring interrupt context save or IST access process.

Figure 6-51. Interrupt Processing



6.2.4.18.3 *Interrupt Acknowledgment*

The ARP32 CPU interrupts are sensitive to rising edge detected on the input interrupt pins (detected synchronously detected on the rising edge of the input clock). All interrupt pins are active high, with the exception of reset, which is active low and asynchronous.

The cpu_iack_o output signal is asserted high for a single cycle to indicate (to hardware external to the CPU core) that the CPU has begun processing an interrupt. The cpu_inum_o signal indicates the number of the interrupt that is being processed. The value driven on cpu_inum_o signal is also captured in the INUM field of the control status register (CSR).

Note that cpu_iack_o and cpu_inum_o are asserted for all interrupts (including reset). Table 6-343 summarizes the cpu_inum_o values.

Table 6-343. cpu_inum_o Values

cpu_inum_o[3:0]	Interrupt
0h	Reset
1h	NMI
2h	SWI
3h	UNDEF
4h	INT4
5h	INT5
6h	INT6
7h	INT7
8h	INT8
9h	INT9
Ah	INT10
Bh	INT11
Ch	INT12
Dh	INT12
Eh	INT14
Fh	INT15

6.2.4.18.4 Interrupt Priorities

When two interrupts happen at the same time, they are serviced by the ARP32 in order of their priority. The priority level for each interrupt is listed in Table 6-344

Table 6-344. Interrupt Priorities

Priority	Interrupt
Highest	Reset
	NMI
	SWI
	INT4
	INT5
	INT6
	INT7
	INT8
	INT9
	INT10
	INT11
	INT12
	INT13
	INT14
	INT15
Lowest	UNDEF

6.2.4.18.5 Interrupt Service Table (IST)

Each entry of the interrupt service table (IST) contains a byte address that points to the beginning of the interrupt service routine for that interrupt. When the ARP32 CPU accepts an interrupt, the contents of the associated IST entry is read (IST[n]), converted to the corresponding halfword address, and then loaded into the PC creating an effective branch.

The starting address of the IST is always the address 00h in the instruction memory; the IST can not be relocated. The addresses (byte address) of each entry is shown in Table 6-345.

IST entries are byte address(s) of the interrupt handler (interrupt service routine) functions. Since 4 bytes are allocated for each IST entry, the interrupt handlers are placed anywhere in the 32-bit program memory space and the ARP32 CPU directly branches to that address. This reduces interrupt latency significantly since an instruction need not be fetched/decoded/executed from an IST location, nor an intermediate routine/call/branch needs to be processed for a branch to relatively large distance.

In fact, the IST is an “array of pointers (each entry 32 bits) to interrupt handler functions in C/C++” placed at the instruction memory address 0, which is remapped to a physical address by using the memory management unit (MMU0) that services program cache requests.

Table 6-345. Interrupt Service Table (IST)

Byte Address	IST Entry
00h	Reset
04h	NMI
08h	SWI
0Ch	UNDEF
10h	INT4
14h	INT5
18h	INT6
1Ch	INT7
20h	INT8
24h	INT9
28h	INT10
2Ch	INT11
30h	INT12
34h	INT13
38h	INT14
3Ch	INT15

6.2.4.18.6 Interrupt Flags

The ARP32 CPU maintains the pending status of all non-reset external interrupts (NMI, INT15-INT4) in the corresponding bit in the interrupt flag register (IFR). A 1 indicates the corresponding interrupt is pending, reading a 0 indicates the corresponding interrupt is not pending (either is being processed or it has not occurred).

When a rising edge is detected on the input pins of external interrupts (NMI, INT15-INT4), the corresponding bits in IFR are set unconditionally in the next CPU clock cycle. When the CPU actually takes a particular interrupt, the corresponding IFR bit is cleared by the CPU. Thus at any point, reading IFR using the **MVC** instruction shows which input external interrupts is asserted or is not yet processed by the CPU (or in pending state).

The CPU checks the status of IFR bits to understand that an interrupt is asserted, and starts processing this interrupt. Processing of an interrupt involves checking for all associated enable conditions, prioritization between simultaneously asserted interrupts and actions to perform interrupt service table look up to finally achieve a branch to the interrupt service routine of the corresponding interrupt. The exact behavior and actions taken on each kind of interrupts are described in Section 6.2.4.18.7.

It is possible to set one or more interrupt flags of maskable interrupts in IFR - by writing to the corresponding bit in the interrupt set register (ISR) - creating the same effect of an external interrupt assertion. This makes it possible for the software to raise a maskable interrupt. The following examples illustrate this:

Example 6-6. Setting Interrupt Flag

```
MVC   0x10, ISR      ; Set IFR[4] to raise INT4
```

Example 6-7. Setting Interrupt Flag

```
MVC   IFR,  R0       ; Get current state
SET   5,5,  R0, R0   ; Set R0:bit[5]
MVC   R0,   ISR      ; Set IFR[5] without changing other bits
```

NOTE: Setting of IFR by an external interrupt assertion has a higher priority than writing or clearing the IFR bits via the **MVC** instruction.

6.2.4.18.7 Interrupt Behavior

All maskable (INT15-INT4) and non-maskable (NMI) interrupt processing is triggered by the corresponding interrupt flag bit in the interrupt flag register (IFR). For SWI and UNDEF, a direct instruction decode (or non-decode) triggers interrupt processing. Once an interrupt is triggered, the CPU checks the corresponding enabling conditions (if any), does a priority resolution among simultaneously asserted interrupts, and then starts processing the interrupt.

The details of conditions for processing each type of interrupts and the actions taken by the CPU during and returning from the interrupts are described in the following subsections.

6.2.4.18.7.1 Reset Interrupt

Reset is the highest priority interrupt and is used to initialize the CPU to a known state. The reset interrupt is unique in a number of ways:

- cpu_resetz_i is an active-low signal and is treated asynchronously. All other interrupts are active-high signals, or activated via the instruction decoder.
- cpu_resetz_i must be held low for 4 clock cycles before it goes high again to reinitialize the CPU properly.
- cpu_resetz_i is not affected by branches or pending loads or any enable bits or any other condition applicable for processing other interrupts. The current instruction execution is aborted and the CPU registers are returned to their default state.
- Reset uses interrupt semantics, that is, loading of the IST table entry; however, it is not required to issue a **BIRP** instruction to exit reset processing.

6.2.4.18.7.2 Non-maskable Interrupt (NMI)

NMI is the second-highest priority interrupt and is generally used to alert the CPU of a serious hardware problem. For an NMI to be detected (and processed) by the ARP32 CPU, the non-maskable interrupt enable (NMIE) bit in the interrupt enable register (IER) must be set to 1. The NMIE bit is cleared to 0 at reset to prevent an NMI being taken prematurely (before system initialization). The NMI bit is set (to enable NMI) by software only after the system has been properly initialized and is ready to accept NMIs. The NMIE bit is also cleared at the occurrence of an NMI to prevent another NMI from being processed; it may be set again in the interrupt set register (ISR), to allow nested NMIs, by software after the CPU state has been saved properly in the previous NMI ISR. While the NMIE bit is cleared, all maskable interrupts (INT15–INT4) are also disabled.

When the NMIF bit is set, (as a result of an NMI assertion via cpu_nmi_i input pin), assuming the previous conditions are met, the ARP32 CPU is said to accept the interrupt and performs the following actions to process the NMI:

- CPU stops fetching further instruction from instruction memory
- CPU execute pipe is allowed to drain
  - Any load/store instructions (including **LDRF, STRF, CALL, RET**) currently executing are allowed to complete
  - Any instruction in the branch/call/return delay slot is allowed to complete
- The CSR[11:8]INUM field is updated with the NMI interrupt ID.
- The CSR content is copied to the NMISCSR.
- The IER[1]NMIE bit is cleared.
- The CSR[0]GIE bit is cleared.
- The IFR[1]NMIF bit in the IFR is cleared.
- The PC value of the next instruction to execute (after completion of the interrupt service routine) is stored to the NRP register. This is the interrupt return address.
- The cpu_iack_o is asserted for a cycle along with the corresponding interrupt ID on cpu_inum_o.
- An instruction fetch request to NMI IST location is sent and eventually the NMI IST entry is loaded into the PC. As a result, the CPU begins executing the NMI interrupt service routine.

To exit an NMI service routine, the **BNRP** instruction must be used. Execution of the **BNRP** instruction causes:

- The NMISCSR content is copied to the CSR.
- The PC loaded with the contents of the NRP.
- IER[1]NMIE bit is set.

6.2.4.18.7.3 SWI Interrupt

The **SWI** instruction is used to trigger the software interrupt. Software interrupts are always enabled. The SWI interrupt does not have a corresponding entry in the IFR, ICR, ISR, or IER, and it is not affected by the state of the CSR:GIE or IER[1]NMIE bits. An SWI does not have any delay slot. Decoding of SWI causes the following actions to be taken by the ARP32 CPU:

- The CSR:INUM field is updated with the SWI interrupt ID.
- The CSR content is copied to the SCSR.
- All architectural registers (R0-R7) and HLA registers (LSAn, LEAn, and LCNTn) are saved to the corresponding shadow registers.
- The CSR[0]GIE bit is cleared.
- The address of the next instruction (interrupt return address) to be stored to the IRP register.
- The cpu_iack_o is asserted for a cycle along with the corresponding interrupt ID on cpu_inum_o.
- An instruction fetch request to SWI IST location is sent and eventually the SWI IST entry is loaded into the PC. As a result, the CPU begins executing the SWI ISR.

To exit a SWI service routine, the **BIRP** instruction must be used. Execution of the **BIRP** instruction causes:

- The SCSR content is copied to the CSR.
- All architectural registers (R0-R7) and HLA registers (LSAn, LEAn, and LCNTn) are restored from the corresponding shadow registers.
- The PC loaded with the contents of the IRP.

6.2.4.18.7.4 Maskable Interrupts

The ARP32 CPU supports 12 maskable interrupts (INT15-INT4). For a maskable interrupt to be detected (and processed) by the ARP32 CPU:

- The global (maskable) interrupt enable (GIE) bit in the control status register (CSR) must be set to 1. The GIE bit is cleared to 0 at reset to prevent a maskable interrupt being taken prematurely (before system initialization). The GIE bit is set (to enable maskable interrupts) only after the system has been properly initialized and is ready to accept interrupts. The GIE bit is also cleared at the occurrence of an interrupt (NMI, INT15-INT4, SWI, and UNDEF) to prevent another interrupt from being processed; it may be set again in the ISR (to enable nested maskable interrupts) by software after the CPU state has been saved off properly.
- The IER:NMIE bit must be set.
- The corresponding enable bit in the interrupt enable register (IER) must also be set. IER bits allow enabling/disabling single maskable interrupt at a time.

When any of the IFR flag bits (IFR[15-4]) of maskable interrupts are set, (as a result of an maskable interrupt assertion via cpu_int[15-4]_i input pins), assuming the previous conditions are met, the ARP32 CPU accepts the highest priority interrupt and performs the following actions to process the interrupt:

- CPU stops fetching further instruction from instruction memory
- CPU execute pipe is allowed to drain
  - Any load/store instructions (including **LDRF, STRF, CALL, RET**) currently executing are allowed to complete
  - Any instruction in the branch/call/return delay slot is allowed to complete
- The CSR[11:8]INUM field is updated with the corresponding maskable interrupt ID.
- The CSR content is copied to the SCSR.
- All architectural registers (R0-R7) and HLA registers (LSAn, LEAn, and LCNTn) are saved to the corresponding shadow registers.
- The CSR[0]GIE bit is cleared.
- The associated bit in the IFR is cleared.
- The PC value of the next instruction to execute (after completion of the interrupt service routine) is stored to the IRP register. This is the interrupt return address.
- The cpu_iack_o is asserted for a cycle along with the corresponding interrupt ID on cpu_inum_o.
- An instruction fetch request to the corresponding IST location is sent and eventually the IST entry is loaded into the PC. As a result, the CPU begins executing the corresponding interrupt service routine.

To exit a maskable service routine, the **BIRP** instruction must be used. Execution of the **BIRP** instruction causes:

- The SCSR content is copied to the CSR.
- All architectural registers (R0-R7) and HLA registers (LSAn, LEAn, and LCNTn) are restored from the corresponding shadow registers.
- The PC loaded with the contents of the IRP.

6.2.4.18.7.5 UNDEF Interrupt

An UNDEF interrupt is triggered by either of the following conditions:

- The CPU decodes an undefined/unsupported instruction, an unmapped primary opcode in instruction word 7:0. This normally happen as a result of instruction memory getting corrupted. Note that not all kinds of invalid opcodes are detected by the CPU; for example, invalid fields other than the primary opcode of a valid instruction.
- The CPU executes a **DIV/DIVU/MOD/MODU** instruction where the divisor operand (*src1*) is 0.

An UNDEF interrupt causes the UNDEF IST entry to be loaded into the PC. UNDEF interrupts are always enabled, it does not have a corresponding entry in the IFR, ICR, ISR, or IER, and is unaffected by the state of the CSR[0]GIE or IER[1]NMIE bits.

The ARP32 CPU treats an UNDEF as an irrecoverable exception. It is not possible to correctly return from an UNDEF interrupt. The only way to recover from an UNDEF interrupt is to reset the CPU.

For example, when UNDEF is caused by an invalid instruction decode, the size of (undecoded) instruction under consideration is not determined unambiguously (since the ARP32 CPU supports freely mixable 16-bit and 32-bit instructions) and hence a recoverable return address is not saved to the IRP while taking an UNDEF interrupt.

The purpose of the UNDEF interrupt is to provide just the basic detection and diagnostic mechanism when such an irrecoverable fault occurs in the system. It is software responsibility to design such an UNDEF handler such that within the UNDEF handler context (without requiring to return from the UNDEF interrupt service routine), it can do diagnostic check(s), if any, and inform the external world (for example, a host CPU) that an irrecoverable fault/exception has happened within the ARP32 CPU that cannot be handled.

It is possible an **IDLE** instruction to be used in the UNDEF interrupt service routine to hold the processor in the UNDEF service routine, for debug purposes, and avoid returning to an incorrect instruction boundary (because of incorrect return address saved to IRP/NRP) causing further UNDEF(s).

Also a manipulation of the IRP to return to some known address (after clearing all machine state correctly, for example, CSR, SCSR), for example the C entry point (__cinit_00) with the expectation that the CPU practically re-executes from reset with all ISA initialization done freshly by the boot code at the entry point. However, it is still just a workaround considering that the root cause of instruction memory corrupt remains uncorrected.

When an undefined instruction is decoded by the ARP32 CPU it performs the following action:

- The CSR[11:8]INUM field is updated with the UNDEF interrupt ID.
- The CSR content is copied to the SCSR.
- All architectural registers (R0-R7) and HLA registers (LSAn, LEAn, and LCNTn) are saved to the corresponding shadow registers.
- The CSR[0]GIE bit is cleared.
- The address of the next instruction (interrupt return address) to be stored to the IRP register.
- The cpu_iack_o is asserted for a cycle along with the corresponding interrupt ID on cpu_inum_o.
- An instruction fetch request to the UNDEF IST location is sent and eventually the UNDEF IST entry is loaded into the PC. As a result, the CPU begins executing the UNDEF interrupt service routine.

To exit an UNDEF service routine, the **BIRP** instruction is used. Execution of the **BIRP** instruction causes:

- The SCSR content is copied to the CSR.
- All architectural registers (R0-R7) and HLA registers (LSAn, LEAn, and LCNTn) are restored from the corresponding shadow registers.
- The PC loaded with the contents of the IRP.

However, it is not assured to result in successfully returning to a correct instruction boundary. More UNDEF may follow immediately.

6.2.4.18.8 Interrupt Context Save and Restore

The ARP32 CPU supports extensive automatic context save and restore during interrupt processing to reduce the effective interrupt latency significantly.

While processing all maskable interrupts (INT15-INT4) , SWI and UNDEF, the ARP32 CPU saves off all key registers into shadow registers and during an interrupt return, via the **BIRP** instruction, restores them back:

- The Control Status Register (CSR) is saved to Shadow Control Status Register (SCSR)
- All architectural registers (R0-R7) are saved to corresponding shadow registers (SR0-SR7)
- The HLA setup registers (LSA*n*, LEA*n*, LCNT*n*, LCNT0RLD) are saved to the corresponding shadow registers (SLSAn, SLEAn, SLCNTn, SLCNT0RLD)

Both saving and restoring happens in a single cycle in parallel to interrupt vector fetch or the return branch execution. Also, as long as interrupt nesting is not enabled, there is absolutely no need of any additional context save (or restore) by the software at the start (or end) of the interrupt service routine. Hence, these interrupts have extremely low effective latency and are considered fast.

While processing the non-maskable interrupt (NMI), however, the CPU does not do extensive context save and restore. Only the CSR register is saved off to NMISCSR and during an NMI return, via the **BNRP** instruction, CSR is restored back. So, at the beginning of an NMI ISR, the software ensures to save off the complete CPU state to stack and restore them back before issuing a **BNRP** instruction (to accomplish a return from NMI):

- All architectural registers (R0-R7)
- The HLA setup registers (LSAn, LEAn, LCNTn, LCNT0RLD)

Thus, the interrupt behavior of the ARP32 CPU makes all maskable interrupts (INT15-INT4), SWI, and UNDEF as fast, and non-maskable interrupt (NMI) as just correct. The maskable interrupts (INT15-INT4) are used as the functional interrupt(s) of the system and the NMI are used to handle/service System Error, Exception scenarios (for example, a bus fault on program or data access).

The following example illustrates the process of context save and restore within an NMI service routine. In this example, the `NMIHandler()` is the actual software NMI handler function and a single unified stack across background and interrupt handler code is intended.

```
__NonMaskableInterruptHandlerWrapper:

        ; Save off Arch regs first
          STRF   R7, R0

        ; Save off HLA regs
          MVC    LSA0,     R0
          MVC    LEA0,     R1
          MVC    LCNT0,    R2
          MVC    LSA1,     R3
          MVC    LEA1,     R4
          MVC    LCNT1,    R5
          MVC    LCNT0RLD, R6
          STRF   R6, R0

        ; Now call the actual handler function
          CALL   NMIHandler
          NOP

        ; Restore HLA regs
          LDRF   R0, R6
          MVC    R0, LSA0
          MVC    R1, LEA0
          MVC    R2, LCNT0
          MVC    R3, LSA1
          MVC    R4, LEA1
          MVC    R5, LCNT1
          MVC    R6, LCNT0RLD

        ; Restore Arch regs
          LDRF   R0, R7

        ; Return from NMI ISR
          BNRP
          NOP
```

Note that in the example, during context restore, a write to LCNT0 also gets written to LCNT0RLD. Hence, the LCNT0RLD register is restored after LCNT0 is restored.

6.2.4.18.9 Nested Interrupts

When an interrupt (INT15–INT4, SWI, and UNDEF) is taken, the CSR[0]GIE bit is cleared by the hardware and subsequently the GIE bit is restored on interrupt return via the **BIRP** instruction. Thus all maskable interrupts (INT15–INT4) are disabled within the interrupt service routine of maskable interrupts.

Also, when an NMI is taken, the IER[1]NMIE bit is cleared by the hardware and subsequently the IER[1]NMIE bit is restored on interrupt return via the **BNRP** instruction. Thus NMI and all maskable interrupts (INT15–INT4) are disabled within the NMI service routine.

So in general, nesting of interrupts (except the case of NMI within a maskable interrupt routine) is disabled. However, under software control, it is possible that the ARP32 CPU is configured to take fully nested interrupts.

The ARP32 CPU offers two programming models with respect to usage of interrupts in a nested or non-nested manner.

6.2.4.18.9.1 Non-nested Interrupt Model

In this programming model, the CSR[0]GIE (or IER[1]NMIE) bit is not enabled within interrupts service routine of maskable interrupts (INT15–INT4), SWI, or UNDEF interrupt (or NMI). Hence, maskable interrupts (INT15–INT4), SWI, and UNDEF are not nested. However, an NMI is taken as a nested interrupt within a maskable interrupt (INT15–INT4), SWI, or UNDEF.

While adopting this programming model, the following points must be noted:

- It is the software responsibility to not use an **SWI** instruction within an ISR context. Otherwise, SWI corrupts the shadow registers and results in an unpredictable behavior.
- An UNDEF interrupts may be taken. UNDEF interrupts are irrecoverable exception and cannot be recovered from (see Section 6.2.4.18.7.5)
- NMI service routine follows the procedure described in Section 6.2.4.18.8. An NMIE must not be enabled within an NMI service routine as nesting of NMI itself is not supported.

Note that this programming model offers fast response time to system events via the maskable interrupt lines (INT15–INT4) by fully exploiting the extensive automatic context save/restore feature of the ARP32 CPU (see Section 6.2.4.18.8).

6.2.4.18.9.2 Nested Interrupt Model

In this programming model, the CSR[0]GIE bit is enabled within the interrupts service routine of maskable interrupts (INT15–INT4), SWI, or UNDEF interrupt. Hence, it is possible that maskable interrupts are nested. An NMI can always be taken with all other non-NMI interrupts.

Before enabling the CSR[0]GIE bit (in ISR), the software must ensure proper context save of all CPU registers onto the stack:

- Architectural registers (R0-R7)
- Shadow registers (SR0-SR7)
- Key control registers: SCSR, IRP/NRP
- Optional control register: SP, IER

While adopting this programming model, the following points must be noted:

- The SP needs to be saved, if a separate stack for the interrupt service routine is required
- IER needs to be saved, if a different interrupt mask is expected within an ISR
- Once the GIE bit is enabled, even the current interrupt (the once being serviced) becomes enabled to be re-entered. If reentrant behavior is not intended, IER must be appropriately modified after saving it off to the stack.
- NMI service routine must follow the procedure described in Section 6.2.4.18.8. An NMIE must not be enabled within an NMI service routine as nesting of NMI itself is not supported. All maskable interrupts (INT15–INT4) and NMI remain disabled within an NMI service routine.

The following example illustrates the process of context save and restore within a maskable interrupt service routine. In this example, it is assumed that the `InterruptHandler()` is the actual software interrupt handler function and a single unified stack across background and interrupt handler code is intended.

```
__NonMaskableInterruptHandlerWrapper:

        ; Save off Arch regs first
          STRF   R7, R0

        ; Save off HLA shadow regs
          MVS    SLSA0,     R0
          MVS    SLEA0,     R1
          MVS    SLCNT0,    R2
          MVS    SLSA1,     R3
          MVS    SLEA1,     R4
          MVS    SLCNT1,    R5
          MVS    SLCNT0RLD, R6
          STRF   R6, R0

        ; Save off SCSR, IRP
          MVC    SCSR,     R0
          MVC    IRP,      R1
          STRF   R1, R0

        ; Now call the actual handler function
          CALL   InterruptHandler
          NOP

        ; Restore SCSR, IRP
          LDRF   R0, R1
          MVC    R0, SCSR
          MVC    R1, IRP

        ; Restore HLA shadow regs
          LDRF   R0, R6
          MVS    R0, SLSA0
          MVS    R1, SLEA0
          MVS    R2, SLCNT0
          MVS    R3, SLSA1
          MVS    R4, SLEA1
          MVS    R5, SLCNT1
          MVS    R6, SLCNT0RLD

        ; Restore Arch regs into shadow copies
          LDRF   R0, R7
          MVS    R0, SR0
          MVS    R0, SR1
          MVS    R0, SR2
          MVS    R0, SR3
          MVS    R0, SR4
          MVS    R0, SR5
          MVS    R0, SR6
          MVS    R0, SR7

        ; Return from NMI ISR
          BIRP
          NOP
```

6.2.4.18.10 Non-nested Interrupt Latency

This section describes the best and worst case interrupt latencies in the non-nested interrupt programming model (described in Section 6.2.4.18.9.1).

6.2.4.18.10.1 Best Case Interrupt Latency

In the best case (and the most frequent case), the ARP32 CPU takes an interrupt as soon as (very next cycle) it is signaled at its boundary via the cpu_int[15-4]_i or cpu_nmi_i input ports. In this case, the ARP32 CPU takes as little as 5 cycles (from the cycle when the IFR bit gets set) to start executing the first instruction of the ISR:

- **Cycle 0:** cpu_int[15-4]_i or cpu_nmi_i assertion
- **Cycle 1:** corresponding IFR bit is set, interrupt enable conditions checked, interrupt priorities resolved, a winner interrupt is selected to be taken
- **Cycle 2:** cpu_iack_o, cpu_inum_o are asserted, program read request is sent to the corresponding IST address associate with the interrupt being serviced
- **Cycle 3:** program read data (IST content) is received and is loaded to PC
- **Cycle 4:** program read request is sent to the first instruction of the ISR routine
- **Cycle 5:** first instruction of the ISR routine is decoded
- **Cycle 6:** first instruction of the ISR routine is executed

Note that the ARP32 CPU saves off all architectural registers and other required CPU registers while taking the interrupt (in cycle 1) and as a result the ISR routine does not need to have any instructions to save off the machine state. The ISR routine directly starts executing code related to actual actions to be taken to service the interrupt.

6.2.4.18.10.2 Worst Case Interrupt Latency

Under certain cases, the ARP32 CPU blocks taking an interrupt, thus introducing variable interrupt latency. The following are the worst cases:

- When a load/store multiple (**LDRF, STRF**) instruction is at the EXE stage of the CPU pipeline and an interrupt is asserted. The interrupt is not taken until the **LDRF/STRF** instruction is completed, which takes several (variable) cycles.

  In compiler-generated code, **LDRF/STRF** is normally used as a function context save/restore across function calls; a maximum of three save-on-entry (SOE) registers as per the ARP32 CPU EABI function calling convention. Thus, three load/stores would need to finish before the CPU takes an interrupt. Moreover, if the stack pointer points to a wait-stated memory, this register save/restore takes even longer. Also, the program fetch request to the IST is stalled if the corresponding memory region is not zero-wait state, thus, increasing interrupt latency further.
- When a blocking, multicycle execute (**DIV/DIVU/MOD/MODU**) instruction is at the EXE stage of the CPU pipeline and an interrupt is asserted. The interrupt is not taken until the instruction at EXE is completed, which takes 13 (fixed) cycles.

6.2.5 Instruction Set

This section provides a description of the instruction set. Unless otherwise explicitly mentioned, all instructions are a single execute cycle instruction. All instructions that are multicycle execute are mentioned explicitly.

6.2.5.1 Instruction Operation and Execution Notations

Table 6-346 explains the symbols used in the instruction Pseudo Code descriptions.

All immediate values are unsigned (zero-extended) or signed (sign-extended) to match the relevant associated operand size. For example, zero (or sign) extended to 32 bits for operations with architectural registers (32 bit in size), zero-extended to 32 bits for operation with SP (32 bits in size), and zero-extended to the width of the PC for operations with the PC.

Although the instruction set architecture (ISA) does not contain a writable program counter (PC) register, in the description of some instructions (especially in the pseudo code), a conceptual register, PC, has been used. This is to simplify the description and does not imply existence of such a physical register.

Table 6-346. Instruction Pseudo Code Notations

Symbol	Meaning
AND	bitwise AND
areg	architectural register (R0-R7)
baseR	base address register
creg	control register
dst	destination operand
GDP	global data pointer register
lmb0(x)	leftmost 0 bit search of x
lmb1(x)	leftmost 1 bit search of x
lsb*n*	n least-significant bits (for example, lsb32)
op*n*	field within opcode that specifies a unique instruction
OR	bitwise OR
PC	program counter
sat	saturate
SP	stack pointer register
src1	source operand 1
src2	source operand 2
sreg	shadow register
ucst*n*	n-bit unsigned immediate constant field (for example, ucst6)
XOR	bitwise exclusive-OR
+	addition
-	subtraction
×	multiplication
/	division
%	modulo
~	logical inverse
++	increment by 1
==	equal to
>	greater than
>=	greater than or equal to
<	less than
<=	less than or equal to
<<	shift left
>>	shift right
&&	logical AND

6.2.5.2 Instruction Syntax and Opcode Notations

Table 6-347 defines the syntaxes and opcode fields used in the instruction descriptions.

Table 6-347. Instruction Syntax and Opcode Notations

Symbol	Meaning
areg	architectural register (R0-R7)
baseR	base address register
cc	condition code (LT, GT, EQ, LE, GE, NE)
creg	control register
dst	destination
GDP	global data pointer register
opn	opfield; field within opcode that specifies a unique instruction
scstn	n-bit signed constant field
sreg	shadow register
SP	stack pointer register
src1	source 1
src2	source 2
ucstn	n-bit unsigned constant field
x	don't care

6.2.5.3 Instruction Scheduling Restrictions

The ARP32 CPU architecture places the following restrictions on instruction scheduling that must be maintained. Whereas the compiler maintains these restrictions while generating assembly code for C/C++ programs, it is the responsibility of the programmer to maintain the restrictions for codes written directly in assembly.

6.2.5.3.1 Restrictions Applicable to a Branch Delay Slot

The following restrictions are applicable for an instruction in the delay slot of a branch (**Bcc** *scst9*, **Bcc** *scst16*, or **Bcc** *dst*), call (**CALL** *scst22* or **CALL** *src1*), and return (**RET**) instruction:

- 32-bit instructions are not allowed
- Another branch/call/return is not allowed
- **IDLE** instruction is not allowed
- **SWI** instruction is not allowed
- **LDRF/STRF** instructions with more than one register operand

Also, a branch delay slot cannot be the target of any branch/call/return.

6.2.5.3.2 Restrictions on Loops Using Hardware Loop Assist (HLA)

Loops setup using the HLA mechanism (Section 6.2.4.17) are required to maintain the following restrictions:

- Minimum number of instructions in Loop 0 is 2
- Branches (**Bcc/CALL/RET** + the delay slot instruction) cannot be placed at the end of a loop. This makes the loop rewind and branch target unambiguous.
- Last two instruction of any level cannot be a multi-slot load/store multiple (**LDRF/STRF**)
- Software cannot branch (**Bcc/CALL/RET**) to the last instruction of any loop, that is, the last instruction of any loop cannot be a branch target.

6.2.5.3.3 Restrictions on Other Types of Control Flow Instructions

The following restrictions are applicable for other types of control flow instructions:

- **SWI** instruction must be followed by a **NOP**
- **BIRP/BNRP** instructions must be followed by a **NOP**

6.2.5.3.4 Restrictions for Write Data Bypass to Control Register Reads

The following restrictions are applicable for write data bypass to control register reads:

- An **MVC** instruction that writes to a control register (**MVC** *areg, creg*; **MVC** *ucst16, creg*; or **MVCH** *ucst16, creg*), must not be followed by an **MVC** instruction (**MVC** *creg, areg*) that reads the same control register.
- An **SLA** instruction that writes to a control register (**SLA** *ucst16*, LSA*n* or **SLA** *ucst16*, LEA*n*) must not be followed by an **MVC** instruction (**MVC** *creg, areg*) that reads the same control register.
- An **SLA** instruction that writes to a control register (**SLA** *ucst16*, LSA*n* or **SLA** *ucst16*, LEA*n*) must not be followed by an **MVCH** instruction (**MVCH** *creg, areg*) that reads the same control register.

6.2.5.3.5 Restrictions for Write Data Bypass to Shadow Register Reads

The following restriction is applicable for write data bypass to shadow register reads:

- Thereare at least two instructions separating an **MVS** instruction writing to a shadow register (**MVS** *areg, sreg*) and another **MVS** instruction reading the same shadow register (**MVS** *sreg, areg*).

6.2.5.3.6 Restrictions for Link Register Update

The following restriction is applicable for link register update:

- Link registers cannot be read or written in the delay slot of a call (**CALL** *scst22* or **CALL** *src1*) and return (**RET**) instruction.

6.2.5.4 Instruction Set Encoding

The ARP32 ISA comprises of a mix of 16-bit and 32-bit instructions. An instruction packet is defined as a 32-bit field for a 32-bit instruction and a 16-bit field for a 16-bit instruction. The instruction decoder is sent an instruction packet every cycle.

6.2.5.5 Instruction Descriptions

Table 6-348 lists the instructions for the ARP32 CPU.

Table 6-348. Instruction Summary

Instruction	Description	Size
ABS src2, dst	Absolute value of a register	16-bit
ADD ucst16, src2, dst	Add 16-bit unsigned constant to a register	32-bit
ADD ucst16, SP	Add 16-bit unsigned constant to stack pointer, result to stack pointer	32-bit
ADD ucst16, SP, dst	Add 16-bit unsigned constant to stack pointer, result to register	32-bit
ADD ucst3, src2, dst	Add 3-bit unsigned constant to a register	16-bit
ADD src1, src2, dst	Signed addition of two register values	16-bit
AND ucst16, src2, dst	Bitwise AND 16-bit unsigned constant with a register	32-bit
AND src1, src2, dst	Bitwise AND two registers	16-bit
B dst	Indirect unconditional branch using a register	16-bit
BEQ dst	Indirect conditional branch using a register	16-bit
BGE dst	Indirect conditional branch using a register	16-bit
BGT dst	Indirect conditional branch using a register	16-bit
BLE dst	Indirect conditional branch using a register	16-bit

Table 6-348. Instruction Summary (continued)

Instruction	Description	Size
BLT dst	Indirect conditional branch using a register	16-bit
BNE dst	Indirect conditional branch using a register	16-bit
B scst9	Direct unconditional branch using a 9-bit signed constant offset	16-bit
BEQ scst9	Direct conditional branch using a 9-bit signed constant offset	16-bit
BGE scst9	Direct conditional branch using a 9-bit signed constant offset	16-bit
BGT scst9	Direct conditional branch using a 9-bit signed constant offset	16-bit
BLE scst9	Direct conditional branch using a 9-bit signed constant offset	16-bit
BLT scst9	Direct conditional branch using a 9-bit signed constant offset	16-bit
BNE scst9	Direct conditional branch using a 9-bit signed constant offset	16-bit
B scst16	Direct unconditional branch using a 16-bit signed constant offset	32-bit
BEQ scst16	Direct conditional branch using a 16-bit signed constant offset	32-bit
BGE scst16	Direct conditional branch using a 16-bit signed constant offset	32-bit
BGT scst16	Direct conditional branch using a 16-bit signed constant offset	32-bit
BLE scst16	Direct conditional branch using a 16-bit signed constant offset	32-bit
BLT scst16	Direct conditional branch using a 16-bit signed constant offset	32-bit
BNE scst16	Direct conditional branch using a 16-bit signed constant offset	32-bit
BIRP	Unconditional branch to interrupt return pointer	16-bit
BKPT	Software breakpoint	16-bit
BNRP	Unconditional branch to NMI return pointer	16-bit
CALL src1	Unconditional call using a register	16-bit
CALL scst22	Unconditional call using a 22-bit signed constant offset	32-bit
CLR ucst5_1, ucst5_2, src2, dst	Clear bit field bounded by two immediate values	32-bit
CLR src1, src2, dst	Clear bit field bounded by two register values	32-bit
CMP scst16, src2	Compare for equality, less than, greater than, 16-bit signed constant to a register	32-bit
CMP scst3, src2	Compare for equality, less than, greater than, 3-bit signed constant to a register	16-bit
CMP src1, src2	Signed compare for equality, less than, greater than, two register values	16-bit
CMPU ucst16, src2	Compare for equality, less than, greater than, 16-bit unsigned constant to a register	32-bit
CMPU ucst3, src2	Compare for equality, less than, greater than, 3-bit unsigned constant to a register	16-bit
CMPU src1, src2	Unsigned compare for equality, less than, greater than, two register values	16-bit
DIV src1, src2, dst	Signed division of two register values	32-bit
DIVU src1, src2, dst	Unsigned division of two register values	32-bit
EXT ucst5_1, ucst5_2, src2, dst	Extract and sign-extend a bit field bounded by two immediate values	32-bit
EXT src1, src2, dst	Extract and sign-extend a bit field bounded by two register values	32-bit
EXTU ucst5_1, ucst5_2, src2, dst	Extract and zero-extend a bit field bounded by two immediate values	32-bit
EXTU src1, src2, dst	Extract and zero-extend a bit field bounded by two register values	32-bit
IDLE	Idle until interrupt or reset	16-bit
LDB *+baseR[ucst3], dst	Load signed byte from memory with a 3-bit unsigned constant offset	16-bit
LDB *+baseR[ucst16], dst	Load signed byte from memory with a 16-bit unsigned constant offset	32-bit
LDB *+baseR[src1], dst	Load signed byte from memory with a register offset	16-bit
LDB *baseR++[ucst3], dst	Load signed byte from memory, postincrement memory with a 3-bit unsigned constant offset	16-bit
LDB *baseR++[src1], dst	Load signed byte from memory, postincrement memory with a register offset	16-bit

Table 6-348. Instruction Summary (continued)

Instruction	Description	Size
LDB *+SP[ucst6], dst	Load signed byte from memory with a SP-relative 6-bit unsigned constant offset	16-bit
LDB *+SP[ucst19], dst	Load signed byte from memory with a SP-relative 19-bit unsigned constant offset	32-bit
LDB *+GDP[ucst19], dst	Load signed byte from memory with a GDP-relative 19-bit unsigned constant offset	32-bit
LDBU *+baseR[ucst3], dst	Load unsigned byte from memory with a 3-bit unsigned constant offset	16-bit
LDBU *+baseR[ucst16], dst	Load unsigned byte from memory with a 16-bit unsigned constant offset	32-bit
LDBU *+baseR[src1], dst	Load unsigned byte from memory with a register offset	16-bit
LDBU *baseR++[ucst3], dst	Load unsigned byte from memory, postincrement memory with a 3-bit unsigned constant offset	16-bit
LDBU *baseR++[src1], dst	Load unsigned byte from memory, postincrement memory with a register offset	16-bit
LDBU *+SP[ucst6], dst	Load unsigned byte from memory with a SP-relative 6-bit unsigned constant offset	16-bit
LDBU *+SP[ucst19], dst	Load unsigned byte from memory with a SP-relative 19-bit unsigned constant offset	32-bit
LDBU *+GDP[ucst19], dst	Load unsigned byte from memory with a GDP-relative 19-bit unsigned constant offset	32-bit
LDH *+baseR[ucst3], dst	Load signed halfword from memory with a 3-bit unsigned constant offset	16-bit
LDH *+baseR[ucst16], dst	Load signed halfword from memory with a 16-bit unsigned constant offset	32-bit
LDH *+baseR[src1], dst	Load signed halfword from memory with a register offset	16-bit
LDH *baseR++[ucst3], dst	Load signed halfword from memory, postincrement memory with a 3-bit unsigned constant offset	16-bit
LDH *baseR++[src1], dst	Load signed halfword from memory, postincrement memory with a register offset	16-bit
LDH *+SP[ucst6], dst	Load signed halfword from memory with a SP-relative 6-bit unsigned constant offset	16-bit
LDH *+SP[ucst19], dst	Load signed halfword from memory with a SP-relative 19-bit unsigned constant offset	32-bit
LDH *+GDP[ucst19], dst	Load signed halfword from memory with a GDP-relative 19-bit unsigned constant offset	32-bit
LDHU *+baseR[ucst3], dst	Load unsigned halfword from memory with a 3-bit unsigned constant offset	16-bit
LDHU *+baseR[ucst16], dst	Load unsigned halfword from memory with a 16-bit unsigned constant offset	32-bit
LDHU *+baseR[src1], dst	Load unsigned halfword from memory with a register offset	16-bit
LDHU *baseR++[ucst3], dst	Load unsigned halfword from memory, postincrement memory with a 3-bit unsigned constant offset	16-bit
LDHU *baseR++[src1], dst	Load unsigned halfword from memory, postincrement memory with a register offset	16-bit
LDHU *+SP[ucst6], dst	Load unsigned halfword from memory with a SP-relative 6-bit unsigned constant offset	16-bit
LDHU *+SP[ucst19], dst	Load unsigned halfword from memory with a SP-relative 19-bit unsigned constant offset	32-bit
LDHU *+GDP[ucst19], dst	Load unsigned halfword from memory with a GDP-relative 19-bit unsigned constant offset	32-bit
LDW *+baseR[ucst3], dst	Load word from memory with a 3-bit unsigned constant offset	16-bit
LDW *+baseR[ucst16], dst	Load word from memory with a 16-bit unsigned constant offset	32-bit
LDW *+baseR[src1], dst	Load word from memory with a register offset	16-bit
LDW *baseR++[ucst3], dst	Load word from memory, postincrement memory with a 3-bit unsigned constant offset	16-bit

Table 6-348. Instruction Summary (continued)

Instruction	Description	Size
LDW *baseR++[src1], dst	Load word from memory, postincrement memory with a register offset	16-bit
LDW *+SP[ucst6], dst	Load word from memory with a SP-relative 6-bit unsigned constant offset	16-bit
LDW *+SP[ucst19], dst	Load word from memory with a SP-relative 19-bit unsigned constant offset	32-bit
LDW *+GDP[ucst19], dst	Load word from memory with a GDP-relative 19-bit unsigned constant offset	32-bit
LDRF op1, op2	Load register file from stack pointer	16-bit
LMBD ucst3, src2, dst	Left most bit detection	32-bit
MAX src1, src2, dst	Maximum of two signed register values	32-bit
MAXU src1, src2, dst	Maximum of two unsigned register values	32-bit
MIN src1, src2, dst	Minimum of two signed register values	32-bit
MINU src1, src2, dst	Minimum of two unsigned register values	32-bit
MOD src1, src2, dst	Signed modulo (32 % 32 = 32)	32-bit
MODU src1, src2, dst	Unsigned modulo (32 % 32 = 32)	32-bit
MPY src1, src2, dst	Signed multiplication of two register values (32-bit × 32-bit = 32-bit)	32-bit
MPYU src1, src2, dst	Unsigned multiplication of two register values (32-bit × 32-bit = 32-bit)	32-bit
MV src1, dst	Move register to register	16-bit
MVC areg, creg	Move architectural register to control register	16-bit
MVC creg, areg	Move control register to architectural register	16-bit
MVC ucst16, creg	Move 16-bit unsigned constant to control register	32-bit
MVCH ucst16, creg	Move 16-bit unsigned constant to upper bits of control register	32-bit
MVK ucst16, dst	Move 16-bit unsigned constant to register	32-bit
MVKH ucst16, dst	Move 16-bit unsigned constant to upper bits of register	32-bit
MVKLS scst16, dst	Move 16-bit signed constant to register	32-bit
MVKS scst6, dst	Move 6-bit signed constant to register	16-bit
MVS areg, sreg	Move architectural register to shadow register	16-bit
MVS sreg, areg	Move shadow register to architectural register	16-bit
NEG src2, dst	Negation	16-bit
NOP	No operation	16-bit
NOT src2, dst	Bitwise NOT	16-bit
OR ucst16, src2, dst	Bitwise OR 16-bit unsigned constant with a register	32-bit
OR src1, src2, dst	Bitwise OR two registers	16-bit
RET	Return from subroutine	16-bit
REV src1, src2, dst	Reverse a bit field bounded by two register values	32-bit
ROT src1, src2, dst	Rotate right	32-bit
ROTC src1, src2, dst	Rotate right through carry bit	32-bit
SADD src1, src2, dst	Signed addition of two register values with saturation	16-bit
SATN ucst3, src2, dst	Saturation to signed/unsigned *n*-bit value with sign/zero extend	32-bit
SET ucst5_1, ucst5_2, src2, dst	Set bit field bounded by two immediate values	32-bit
SET src1, src2, dst	Set bit field bounded by two register values	32-bit
SHL ucst5, src2, dst	Logical shift left by 5-bit unsigned constant	32-bit
SHL src1, src2, dst	Logical shift left by register value	16-bit
SHRA ucst5, src2, dst	Arithmetic shift right by 5-bit unsigned constant	32-bit
SHRA src1, src2, dst	Arithmetic shift right by register value	16-bit
SHRU ucst5, src2, dst	Logical shift right by 5-bit unsigned constant	32-bit
SHRU src1, src2, dst	Logical shift right by register value	16-bit
SLA ucst16, creg	Set loop address	32-bit

Table 6-348. Instruction Summary (continued)

Instruction	Description	Size
SSUB src1, src2, dst	Subtraction of two register values with saturation	16-bit
STB dst, *+baseR[ucst3]	Store byte to memory with a 3-bit unsigned constant offset	16-bit
STB dst, *+baseR[ucst16]	Store byte to memory with a 16-bit unsigned constant offset	32-bit
STB dst, *+baseR[src1]	Store byte to memory with a register offset	16-bit
STB dst, *baseR++[ucst3]	Store byte to memory, postincrement memory with a 3-bit unsigned constant offset	16-bit
STB dst, *baseR++[src1]	Store byte to memory, postincrement memory with a register offset	16-bit
STB dst, *+SP[ucst6]	Store byte to memory with a SP-relative 6-bit unsigned constant offset	16-bit
STB dst, *+SP[ucst19]	Store byte to memory with a SP-relative 19-bit unsigned constant offset	32-bit
STB dst, *+GDP[ucst19]	Store byte to memory with a GDP-relative 19-bit unsigned constant offset	32-bit
STH dst, *+baseR[ucst3]	Store halfword to memory with a 3-bit unsigned constant offset	16-bit
STH dst, *+baseR[ucst16]	Store halfword to memory with a 16-bit unsigned constant offset	32-bit
STH dst, *+baseR[src1]	Store halfword to memory with a register offset	16-bit
STH dst, *baseR++[ucst3]	Store halfword to memory, postincrement memory with a 3-bit unsigned constant offset	16-bit
STH dst, *baseR++[src1]	Store halfword to memory, postincrement memory with a register offset	16-bit
STH dst, *+SP[ucst6]	Store halfword to memory with a SP-relative 6-bit unsigned constant offset	16-bit
STH dst, *+SP[ucst19]	Store halfword to memory with a SP-relative 19-bit unsigned constant offset	32-bit
STH dst, *+GDP[ucst19]	Store halfword to memory with a GDP-relative 19-bit unsigned constant offset	32-bit
STW dst, *+baseR[ucst3]	Store word to memory with a 3-bit unsigned constant offset	16-bit
STW dst, *+baseR[ucst16]	Store word to memory with a 16-bit unsigned constant offset	32-bit
STW dst, *+baseR[src1]	Store word to memory with a register offset	16-bit
STW dst, *baseR++[ucst3]	Store word to memory, postincrement memory with a 3-bit unsigned constant offset	16-bit
STW dst, *baseR++[src1]	Store word to memory, postincrement memory with a register offset	16-bit
STW dst, *+SP[ucst6]	Store word to memory with a SP-relative 6-bit unsigned constant offset	16-bit
STW dst, *+SP[ucst19]	Store word to memory with a SP-relative 19-bit unsigned constant offset	32-bit
STW dst, *+GDP[ucst19]	Store word to memory with a GDP-relative 19-bit unsigned constant offset	32-bit
STHI ucst16, *+baseR[ucst6]	Store 16-bit halfword to memory with a 6-bit unsigned constant offset	32-bit
STRF op1, op2	Store register file to stack pointer	16-bit
SUB ucst16, src2, dst	Subtract 16-bit unsigned constant from a register	32-bit
SUB ucst16, SP	Subtract 16-bit unsigned constant from stack pointer, result to stack pointer	32-bit
SUB ucst16, SP, dst	Subtract 16-bit unsigned constant from stack pointer, result to register	32-bit
SUB ucst3, src2, dst	Subtract 3-bit unsigned constant from a register	16-bit
SUB src1, src2, dst	Signed subtraction of two register values	16-bit
SWI	Software interrupt	16-bit
XOR ucst16, src2, dst	Bitwise XOR 16-bit unsigned constant with a register	32-bit
XOR src1, src2, dst	Bitwise XOR two registers	16-bit

ABS *Absolute Value*

Syntax

ABS *src2, dst*

Functional unit = L

Opcode

16 bit

15 13	12 10	9									0
dst	*src2*	0	0	0	1	0	0	0	0	0	0
3	3	*opcode*									

Description

The absolute value of *src2* is placed in *dst*. Note that *src2* is considered a signed 32-bit value.

Condition Codes

CSR[2]EQ = (dst == 0)

Pseudo Code

```
dst = (src2 < 0)? -src2:src2
```

ADD *Add 16-Bit Unsigned Constant to Register*

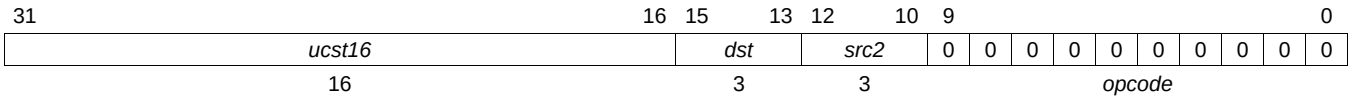
Syntax

ADD *ucst16, src2, dst*

Functional unit = D

Opcode

32 bit



Description

Addition of a 16-bit unsigned constant (*ucst16*) with *src2* and store result to *dst*.

Condition Codes

CSR[2]EQ = (dst == 0)

CSR[5]C = {carry out} from (dst + {zero extend}ucst16)

CSR[7]V = {overflow} from (dst + {zero extend}ucst16)

Note that the carry (CSR:C) and overflow (CSR:V) bits are relevant to unsigned and signed operations, respectively. Appropriate bits are checked depending on if the operands are represented (or, interpreted) as unsigned or signed numbers.

Pseudo Code

```
dst = src2 + {zero extend}ucst16
```

ADD *Add 16-Bit Unsigned Constant to Stack Pointer, Result to Stack Pointer*

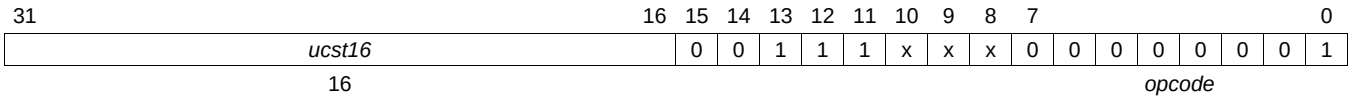
Syntax

ADD *ucst16,* **SP**

Functional unit = D

Opcode

32 bit



Description

Addition of a 16-bit unsigned constant (*ucst16*) with the stack pointer (SP) and store result to SP.

Condition Codes

None

Pseudo Code

```
SP = SP + ucst16
```

ADD *Add 16-Bit Unsigned Constant to Stack Pointer, Result to Register*

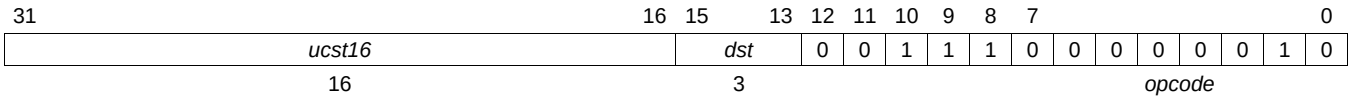
Syntax

ADD *ucst16,* **SP**, *dst*

Functional unit = D

Opcode

32 bit



31 … 16	15 … 13	12	11	10	9	8	7							0
ucst16	*dst*	0	0	1	1	1	0	0	0	0	0	0	1	0
16	3	*opcode*												

Description

Addition of a 16-bit unsigned constant (*ucst16*) with the stack pointer (SP) and store result to *dst*.

Condition Codes

CSR[2]EQ = (dst == 0)

CSR[5]C = {carry out} from (SP + ucst16)

CSR[7]V = {overflow} from (SP + ucst16)

Note that the carry (CSR:C) and overflow (CSR:V) bits are relevant to unsigned and signed operations, respectively. Appropriate bits are checked depending on if the operands are represented (or, interpreted) as unsigned or signed numbers.

Pseudo Code

```
dst = SP + ucst16
```

ADD *Add 3-Bit Unsigned Constant to Register*

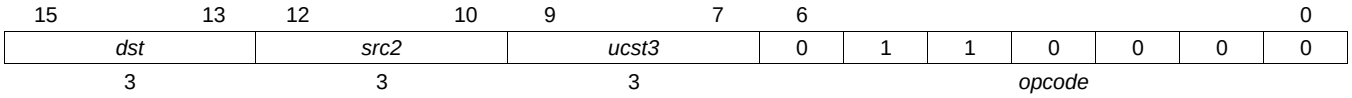
Syntax

ADD *ucst3, src2, dst*

Functional unit = D

Opcode

16 bit



Description

Addition of a 3-bit unsigned constant (*ucst3*) with *src2* and store result to *dst*.

Condition Codes

CSR[2]EQ = (dst == 0)

CSR[5]C = {carry out} from (src2 + {zero extend}ucst3)

CSR[7]V = {overflow} from (src2 + {zero extend}ucst3)

Note that the carry (CSR:C) and overflow (CSR:V) bits are relevant to unsigned and signed operations, respectively. Appropriate bits are checked depending on if the operands are represented (or, interpreted) as unsigned or signed numbers.

Pseudo Code

```
dst = src2 + {zero extend}ucst3
```

ADD *Signed Addition of Two Register Values*

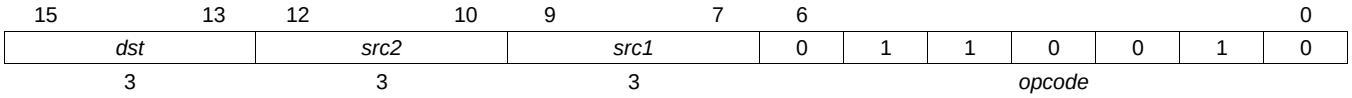
Syntax

ADD *src1, src2, dst*

Functional unit = D

Opcode

16 bit



15–13	12–10	9–7	6			5–0			
dst	*src2*	*src1*	0	1	1	0	0	1	0
3	3	3	*opcode*						

Description

Signed addition of *src1* with *src2* and store result to *dst*.

Condition Codes

CSR[2]EQ = (dst == 0)

CSR[5]C = {carry out} from (src2 + src1)

CSR[7]V = {overflow} from (src2 + src1)

Note that the carry (CSR[5]C) and overflow (CSR[7]V) bits are relevant to unsigned and signed operations, respectively. Appropriate bits are checked depending on if the operands are represented (or, interpreted) as unsigned or signed numbers.

The status of the CSR[5]C bit is used to synthesize multiword (wider) addition. To synthesize a multiword addition, subsequent instructions use the CSR[5]C bit as a carry in operand, performing a normal addition if CSR[5]C == 0 and adding one more than usual if CSR[5]C == 1.

For example, if register pairs R0/R1 and R2/R3 hold 64-bit values (where R0 and R2 hold the least-significant words), the following instructions leave the 64-bit sum in the register pair R4/R5:

```
ADD   R2, R0, R4       ; R4=R2+R0, sets CSR[C]
MVC   CSR, R6          ; leaves CSR[C] untouched
EXTU  5, 5, R6, R6     ; if (CSR[C]) R6=0x1 else R6=0x0
ADD   R3, R1, R5       ; R5=R3+R1
ADD   R5, R6, R5       ; R5=R5+1 if CSR[C] was set, else R5=R5
```

Note that the above example needs appropriate modification in order to generate correct carry/overflow applicable for the total 64-bit result.

Pseudo Code

```
dst = src2 + src1
```

AND — *Bitwise AND 16-Bit Unsigned Constant with Register*

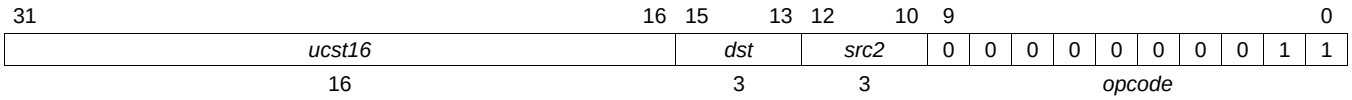
Syntax

AND *ucst16, src2, dst*

Functional unit = L

Opcode

32 bit



Description

Bitwise AND of a zero-extended 16-bit unsigned constant (*ucst16*) with *src2* and store result to *dst*.

Condition Codes

CSR:EQ = (dst == 0)

Pseudo Code

```
dst = src2 AND {zero extend}ucst16
```

AND — *Bitwise AND Two Registers*

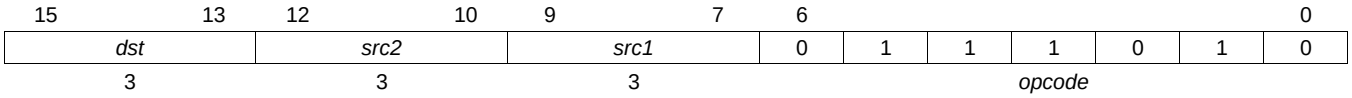
Syntax

AND *src1, src2, dst*

Functional unit = L

Opcode

16 bit



15–13	12–10	9–7	6			5–0
dst	*src2*	*src1*	0	1	1	1 0 1 0
3	3	3	*opcode*			

Description

Bitwise AND of *src1* with *src2* and store result to *dst*.

Condition Codes

CSR:EQ = (dst == 0)

Pseudo Code

```
dst = src2 AND src1
```

B(cc) *Indirect Branch Using a Register*

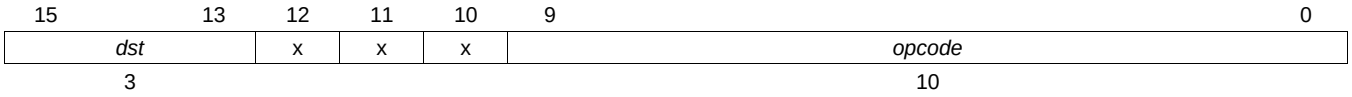
Syntax

B(cc) *dst*

Functional unit = D

Opcode

16 bit



Syntax	Opcode
B *dst*	00 0111 1110
BLT *dst*	01 0111 1110
BGT *dst*	10 1111 1110
BEQ *dst*	01 1111 1110
BLE *dst*	11 0111 1110
BGE *dst*	11 1111 1110
BNE *dst*	10 0111 1110

Description

Performs a branch to the address specified by the *dst* register. The contents of *dst* is treated as a byte address, it is converted to a halfword address before assigning to the PC. Note that the PC always contains a halfword address.

This instruction has one delay slot. See Section 6.2.5.3.1 for restrictions on scheduling an instruction in this delay slot.

For conditional branches, if the associated conditional (cc) bit is set in the control status register (CSR), the branch is taken; otherwise, the PC is incremented by 1. Valid values for ***cc*** are:

- <blank> = unconditional
- LT = CSR[3]LT bit is set to 1
- GT = CSR[4]GT bit is set to 1
- EQ = CSR[2]EQ bit is set to 1
- LE = CSR[3]LT and CSR:EQ bits are set to 1
- GE = CSR[4]GT and CSR:EQ bits are set to 1
- NE = CSR[2]EQ bit is cleared to 0

Condition Codes

None

Pseudo Code

```
if (cond)     PC = (*dst >> 1)
else          PC = PC + 1
```

B(cc) *Direct Branch Using a 9-Bit Signed Constant Offset*

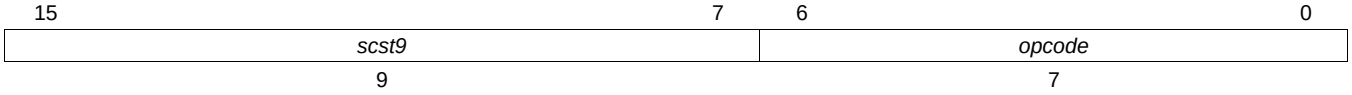
Syntax

B(cc) *scst9*

Functional unit = D

Opcode

16 bit



Syntax	Opcode
B *scst9*	100 1010
BLT *scst9*	100 1011
BGT *scst9*	100 1110
BEQ *scst9*	100 1100
BLE *scst9*	100 1111
BGE *scst9*	101 0000
BNE *scst9*	100 1101

Description

Performs a branch to the effective address formed by the signed addition of *scst9* with the current PC value. The offset is considered as a halfword offset. Note that the PC always contains a halfword address.

This instruction has one delay slot. See Section 6.2.5.3.1 for restrictions on scheduling an instruction in this delay slot.

For conditional branches, if the associated conditional (cc) bit is set in the control status register (CSR), the branch is taken; otherwise, the PC is incremented by 1. Valid values for ***cc*** are:

- <blank> = unconditional
- LT = CSR[3]LT bit is set to 1
- GT = CSR[4]GT bit is set to 1
- EQ = CSR[2]EQ bit is set to 1
- LE = CSR[3]LT and CSR:EQ bits are set to 1
- GE = CSR[4]GT and CSR:EQ bits are set to 1
- NE = CSR[2]EQ bit is cleared to 0

Condition Codes

None

Pseudo Code

```
if (cond)     PC = (PC + scst9)
else          PC = PC + 1
```

B(cc) *Direct Branch Using a 16-Bit Signed Constant Offset*

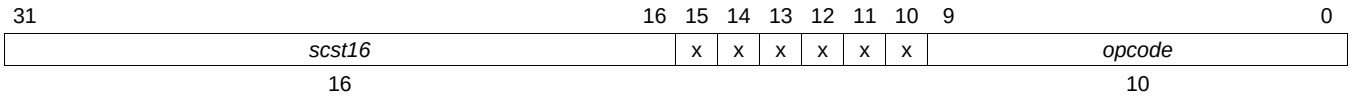
Syntax

B(cc) *scst16*

Functional unit = D

Opcode

32 bit



Syntax	Opcode
B *scst16*	00 1000 1001
BLT *scst16*	01 0000 1001
BGT *scst16*	10 1000 1001
BEQ *scst16*	01 1000 1001
BLE *scst16*	11 0000 1001
BGE *scst16*	11 1000 1001
BNE *scst16*	10 0000 1001

Description

Performs a branch to the effective address formed by the signed addition of *scst16* with the current PC value. The offset is considered as a halfword offset. Note that the PC always contains a halfword address.

This instruction has one delay slot. See Section 6.2.5.3.1 for restrictions on scheduling an instruction in this delay slot.

For conditional branches, if the associated conditional (cc) bit is set in the control status register (CSR), the branch is taken; otherwise, the PC is incremented by 1. Valid values for ***cc*** are:

- <blank> = unconditional
- LT = CSR[3]LT bit is set to 1
- GT = CSR[4]GT bit is set to 1
- EQ = CSR[2]EQ bit is set to 1
- LE = CSR[3]LT and CSR:EQ bits are set to 1
- GE = CSR[4]GT and CSR:EQ bits are set to 1
- NE = CSR[2]EQ bit is cleared to 0

Condition Codes

None

Pseudo Code

```
if (cond)     PC = (PC + scst16)
else          PC = PC + 1
```

BIRP *Branch Using an Interrupt Return Pointer*

Syntax

BIRP

Functional unit = D

Opcode

16 bit

15	14	13	12	11	10	9									0
x	x	x	x	x	x	1	0	0	1	1	1	1	1	1	1
										opcode					

Description

Performs an unconditional branch to the address in the interrupt return pointer register (IRP). The content of IRP is treated as a halfword address. The shadow control status register (SCSR) content is copied back to the control status register (CSR) as a result of this instruction.

This instruction has one delay slot. Only a NOP instruction is supported in a BIRP delay slot.

Condition Codes

For context restore: CSR = SCSR

Pseudo Code

```
PC = *IRP
CSR = SCSR, Rn = SRn, LSAn = SLSAn, LEAn = SLEAn, LCNTn = SLCNTn
```

BKPT *Software Breakpoint*

Syntax

BKPT

Functional unit = S

Opcode

16 bit

15	14	13	12	11	10	9									0
x	x	x	x	x	x	0	0	1	1	1	1	1	1	1	1
										opcode					

Description

The **BKPT** instruction is equivalent to a single instruction endless loop. The ARP32 CPU re-fetches this same instruction again until released through the debug interface.

Condition Codes

None

Pseudo Code

None

BNRP *Branch Using NMI Return Pointer*

Syntax

BNRP

Functional unit = D

Opcode

16 bit

15	14	13	12	11	10	9									0
x	x	x	x	x	x	0	1	0	1	1	1	1	1	1	1
										opcode					

Description

Performs an unconditional branch to the address in the nonmaskable interrupt return pointer register (NRP). The content of NRP is treated as a halfword address. The IER1 is set to 1 upon executing this instruction. The shadow control status register (SCSR) content is copied back to the control status register (CSR) as a result of this instruction.

This instruction has one delay slot. Only a NOP instruction is supported in a BNRP delay slot.

Condition Codes

For context restore: CSR = NMISCSR

Pseudo Code

```
PC = *NRP
CSR = NMISCSR
```

CALL *Call Using a Register*

Syntax

CALL *src1*

Functional unit = D

Opcode

16 bit

15 13	12	11	10	9									0
src1	x	x	x	0	0	0	1	1	1	1	1	1	0
3								*opcode*					

Description

Store the link register (LR) to the location pointed to by the stack pointer (SP), decrement the SP, store PC + 2 into LR, then perform an unconditional branch to the address contained in *src1*. The contents of *src1* is treated as a byte address.

This instruction has one delay slot. See Section 6.2.5.3.1 for restrictions on scheduling an instruction in this delay slot.

Condition Codes

None

Pseudo Code

```
*(SP) = LR; SP -=4; LR = PC + 2; PC = *src1 >> 1
```

CALL *Call Using a 22-Bit Signed Constant Offset*

Syntax

CALL *scst22*

Functional unit = D

Opcode

32 bit

Description

Store the link register (LR) to the location pointed to by the stack pointer (SP), decrement the SP, store PC + 3 into LR, then perform an unconditional branch to the address specified by the PC relative immediate signed halfword offset (*scst22*).

This instruction has one delay slot. See Section 6.2.5.3.1 for restrictions on scheduling an instruction in this delay slot.

Condition Codes

None

Pseudo Code

```
*(SP) = LR; SP -=4; LR = PC + 3; PC = PC + scst22
```

CLR *Clear Bit Field Bounded by Two Immediate Values*

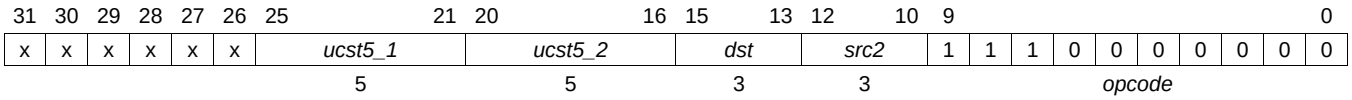
Syntax

CLR *ucst5_1, ucst5_2, src2, dst*

Functional unit = L

Opcode

32 bit



Description

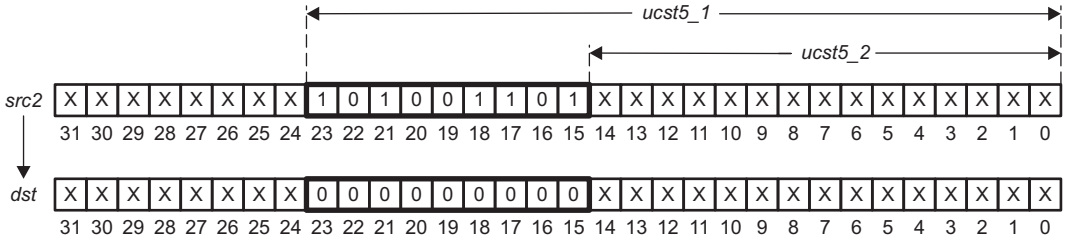
For *ucst5_1* ≥ *ucst5_2*, the field in *src2* as specified by *ucst5_2* to *ucst5_1* is cleared to all 0s and the resulting value of *src2* is written to *dst*. *src2* is left unchanged. *ucst5_2* is the LSB of the field and *ucst5_1* is the MSB of the field. In other words, *ucst5_2* and *ucst5_1* represent the beginning and ending bits, respectively, of the field to be cleared to all 0s. The LSB location of *src2* is bit 0 and the MSB location of *src2* is bit 31.

Valid values of *ucst5_1* and *ucst5_2* are:

- 0 ≤ *ucst5_1* ≤ 31; 0 ≤ *ucst5_2* ≤ 31
- *ucst5_1* ≥ *ucst5_2*

For *ucst5_1* < *ucst5_2*, the result is undefined.

In the following example, *ucst5_2* is 15 and *ucst5_1* is 23.



Condition Codes

CSR:EQ = (dst == 0)

Pseudo Code

```
dst = src2
for( i = 0; i < 32; i++) {
 if ( i >= ucst5_2 && i <= ucst5_2) dst[i] = 0;
};
```

CLR *Clear Bit Field Bounded by Two Register Values*

Syntax

CLR *src1, src2, dst*

Functional unit = L

Opcode

32 bit

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	13	12	10	9	7	6						0
x	x	x	x	x	x	x	x	x	x	x	1	0	0	0	1	*dst*		*src2*		*src1*		0	0	0	0	1	0	0
																3		3		3					*opcode*			

Description

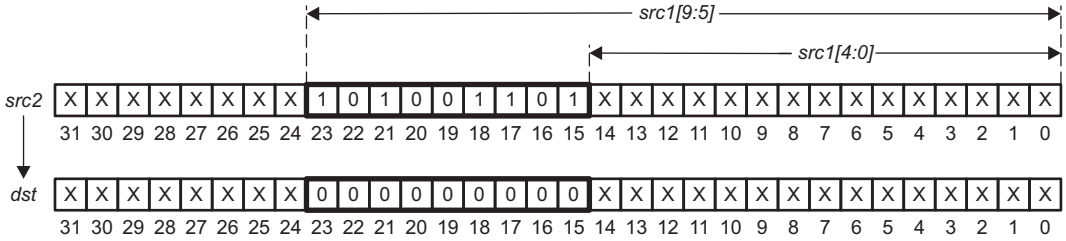
For *src1[9:5]* ≥ *src1[4:0*, the field in *src2* as specified by *src1[4:0]* to *src1[9:5]* is cleared to all 0s and the resulting value of *src2* is written to *dst*. *src2* is left unchanged. *src1[4:0]* is the LSB of the field and *src1[9:5]* is the MSB of the field. In other words, *src1[4:0]* and *src1[9:5]* represent the beginning and ending bits, respectively, of the field to be cleared to all 0s. The LSB location of *src2* is bit 0 and the MSB location of *src2* is bit 31.

Valid values of *src1[9:5* and *src1[4:0]* are:

- 0 ≤ *src1[9:5]* ≤ 31; 0 ≤ *src1[4:0]* ≤ 31
- *src1[9:5]* ≥ *src1[4:0]*

For *src1[9:5]* < *src1[4:0]*, the result is undefined.

In the following example, *src1[4:0]* is 15 and *src1[9:5]* is 23.



Condition Codes

CSR[2]EQ = (dst == 0)

Pseudo Code

```
dst = src2
for( i = 0; i < 32; i++) {
 if ( i >= src1[4:0] && i <= src1[9:5]) dst[i] = 0;
};
```

CMP *Compare for Equality, Less Than, Greater Than, 16-Bit Signed Constant to Register*

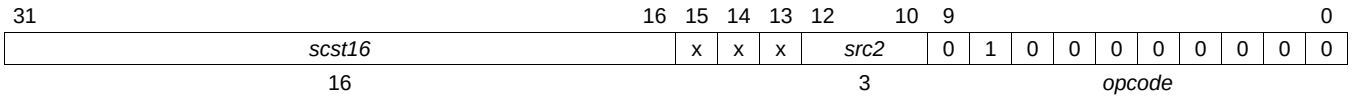
Syntax

CMP *scst16, src2*

Functional unit = D

Opcode

32 bit



Description

Compares *src2* to a sign-extended 16-bit constant (*scst16*). The condition bits CSR:EQ, CSR:LT, and CSR:GT are updated based on the comparison.

Condition Codes

CSR[2]EQ = (src2 == {sign extend}(scst16)

CSR[3]LT = (src2 < {sign extend}(scst16)

CSR[4]GT = (src2 > {sign extend}(scst16)

Pseudo Code

See Condition Codes

CMP *Compare for Equality, Less Than, Greater Than, 3-Bit Signed Constant to Register*

Syntax

CMP *scst3, src2*

Functional unit = D

Opcode

16 bit

15	14	13	12 10	9 7	6						0
x	x	x	*src2*	*scst3*	0	1	1	0	1	0	0
			3	3				*opcode*			

Description

Compares *src2* to a sign-extended 3-bit constant (*scst3*). The condition bits CSR:EQ, CSR:LT, and CSR:GT are updated based on the comparison.

Condition Codes

CSR[2]EQ = (src2 == {sign extend}(scst3)

CSR[3]LT = (src2 < {sign extend}(scst3)

CSR[4]GT = (src2 > {sign extend}(scst3)

Pseudo Code

See Condition Codes

CMP *Signed Compare for Equality, Less Than, Greater Than, Two Register Values*

Syntax

CMP *src1, src2*

Functional unit = D

Opcode

16 bit

15	14	13	12 10	9 7	6						0
x	x	x	*src2*	*src1*	0	1	1	0	1	1	0
			3	3				*opcode*			

Description

Compares *src2* to *src1*. The condition bits CSR:EQ, CSR:LT, and CSR:GT are updated based on the comparison.

Condition Codes

CSR[2]EQ = (src2 == src1)

CSR[3]LT = (src2 < src1)

CSR[4]GT = (src2 > src1)

Pseudo Code

See Condition Codes

CMPU *Compare for Equality, Less Than, Greater Than, 16-Bit Unsigned Constant to Register*

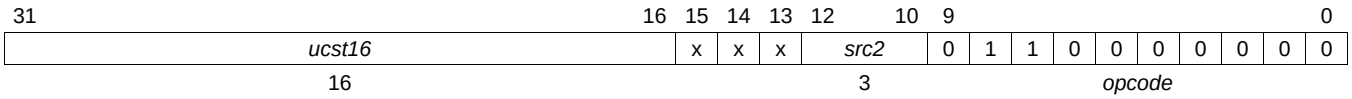
Syntax

CMPU *ucst16, src2*

Functional unit = D

Opcode

32 bit



31 16	15	14	13	12 10	9								0
ucst16	x	x	x	*src2*	0	1	1	0	0	0	0	0	0
16				3	*opcode*								

Description

Compares *src2* to a zero-extended 16-bit constant (*ucst16*). The condition bits CSR:EQ, CSR:LT, and CSR:GT are updated based on the comparison. This is an unsigned comparison.

Condition Codes

CSR[2]EQ = (src2 == {zero extend}(ucst16)

CSR[3]LT = (src2 < {zero extend}(ucst16)

CSR[4]GT = (src2 > {zero extend}(ucst16)

Pseudo Code

See Condition Codes

CMPU

Compare for Equality, Less Than, Greater Than, 3-Bit Unsigned Constant to Register

Syntax

CMPU *ucst3, src2*

Functional unit = D

Opcode

16 bit

15	14	13	12 10	9 7	6						0
x	x	x	*src2*	*ucst3*	0	1	1	0	1	0	1
			3	3				*opcode*			

Description

Compares *src2* to a zero-extended 3-bit constant (*ucst3*). The condition bits CSR:EQ, CSR:LT, and CSR:GT are updated based on the comparison. This is an unsigned comparison.

Condition Codes

CSR[2]EQ = (src2 == {zero extend}(ucst3)

CSR[3]LT = (src2 < {zero extend}(ucst3)

CSR[4]GT = (src2 > {zero extend}(ucst3)

Pseudo Code

See Condition Codes

CMPU *Unsigned Compare for Equality, Less Than, Greater Than, Two Register Values*

Syntax

CMPU *src1, src2*

Functional unit = D

Opcode

16 bit

15	14	13	12 10	9 7	6						0
x	x	x	*src2*	*src1*	0	1	1	0	1	1	1
			3	3				*opcode*			

Description

Compares *src2* to *src1*. The condition bits CSR:EQ, CSR:LT, and CSR:GT are updated based on the comparison. This is an unsigned comparison.

Condition Codes

CSR[2]EQ = (src2 == src1)

CSR[3]LT = (src2 < src1)

CSR[4]GT = (src2 > src1)

Pseudo Code

See Condition Codes

DIV *Signed Division of Two Register Values*

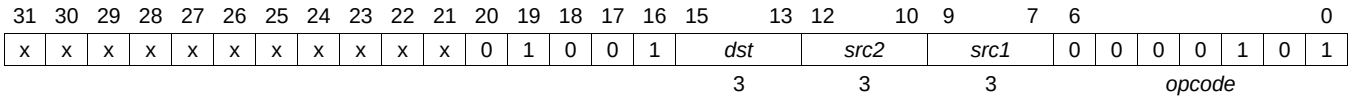
Syntax

DIV *src1, src2, dst*

Functional unit = M

Opcode

32 bit



31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
x	x	x	x	x	x	x	x	x	x	x	0	1	0	0	1	*dst*			*src2*			*src1*			0	0	0	0	1	0	1
																3			3			3			*opcode*						

Description

Signed division of *src1* and *src2* and store result to *dst*. Divide by 0 raises UNDEF interrupt, *dst* is written with 0, CSR:EQ gets set.

Using **DIV** (integer division) and **SHRA** (arithmetic right shift) does not produce the same result for negative numbers. The quotient of **DIV** is rounded towards zero, whereas the quotient of **SHRA** is rounded towards negative infinity. For example, using the **DIV** instruction: -9/4 = -2, whereas using the **SHRA** instruction: -9/4 = -3.

Condition Codes

CSR:EQ = (dst == 0)

Pseudo Code

```
dst = src2 / src1
```

Cycles

14 execute cycles. This is a blocking multi-execute cycle instruction.

DIVU *Unsigned Division of Two Register Values*

Syntax

DIVU *src1, src2, dst*

Functional unit = M

Opcode

32 bit

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
x	x	x	x	x	x	x	x	x	x	x	0	1	1	0	1	*dst*			*src2*			*src1*			0	0	0	0	1	0	1
																3			3			3			*opcode*						

Description

Unsigned division of *src1* and *src2* and store result to *dst*. Divide by 0 raises UNDEF interrupt, *dst* is written with 0, CSR:EQ gets set.

Condition Codes

CSR[2]EQ = (dst == 0)

Pseudo Code

```
dst = src2 / src1
```

Cycles

14 execute cycles. This is a blocking multi-execute cycle instruction.

EXT *Extract and Sign-Extend a Bit Field Bounded by Two Immediate Values*

Syntax

EXT *ucst5_1, ucst5_2, src2, dst*

Functional unit = L

Opcode

32 bit

31	30	29	28	27	26	25 21	20 16	15 13	12 10	9									0
x	x	x	x	x	x	*ucst5_1*	*ucst5_2*	*dst*	*src2*	1	0	0	0	0	0	0	0	0	0
						5	5	3	3	*opcode*									

Description

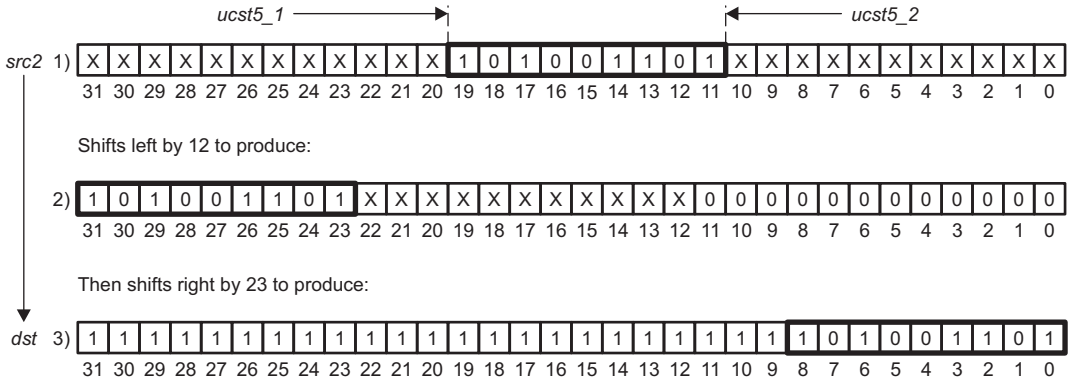
For *ucst5_1* ≥ *ucst5_2*, the field in *src2* as specified by *ucst5_2* to *ucst5_1* is extracted and sign-extended to 32 bits and is written to *dst*. *src2* is left unchanged. *ucst5_2* is the LSB of the field and *ucst5_1* is the MSB of the field to be extracted. In other words, *ucst5_2* and *ucst5_1* represent the beginning and ending bits, respectively, of the field to be extracted. The LSB location of *src2* is bit 0 and the MSB location of *src2* is bit 31.

Valid values of *ucst5_1* and *ucst5_2* are:

- 0 ≤ *ucst5_1* ≤ 31; 0 ≤ *ucst5_2* ≤ 31
- *ucst5_1* ≥ *ucst5_2*

For *ucst5_1* < *ucst5_2*, the result is undefined.

In the following example, *ucst5_2* is 11 and *ucst5_1* is 19.



Condition Codes

CSR:EQ = (dst == 0)

Pseudo Code

```
tmpdst = 0; j = 0;
for( i = 0; i < 32; i++) {
 if ( i >= ucst5_2 && i <= ucst5_1) tmpdst[j++] = src2[i];
};
dst = tmpdst[j] ? signext(tmpdst) : tmpdst ;
```

EXT *Extract and Sign-Extend a Bit Field Bounded by Two Register Values*

Syntax

EXT *src1, src2, dst*

Functional unit = L

Opcode

32 bit

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
x	x	x	x	x	x	x	x	x	x	x	1	0	0	1	0	*dst*			*src2*			*src1*			0	0	0	0	1	0	0
																3			3			3			*opcode*						

Description

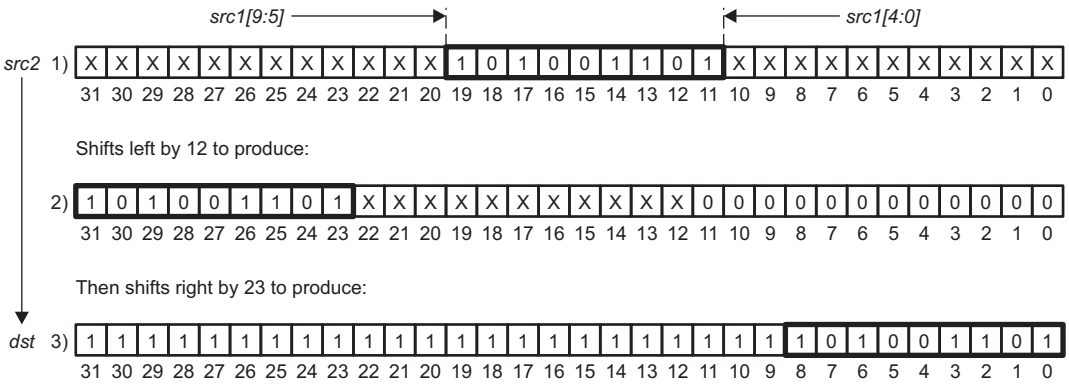
For *src1[9:5]* ≥ *src1[4:0]*, the field in *src2* as specified by *src1[4:0]* to *src1[9:5]* is extracted and sign-extended to 32 bits and is written to *dst*. *src2* is left unchanged. *src1[4:0]* is the LSB of the field and *src1[9:5]* is the MSB of the field to be extracted. In other words, *src1[4:0]* and *src1[9:5]* represent the beginning and ending bits, respectively, of the field to be extracted. The LSB location of *src2* is bit 0 and the MSB location of *src2* is bit 31.

Valid values of *src1[9:5]* and *src1[4:0]* are:

- 0 ≤ *src1[9:5]* ≤ 31; 0 ≤ *src1[4:0]* ≤ 31
- *src1[9:5]* ≥ *src1[4:0]*

For *src1[9:5]* < *src1[4:0]*, the result is undefined.

In the following example, *src1[4:0]* is 11 and *src1[9:5]* is 19.



Condition Codes

CSR:EQ = (dst == 0)

Pseudo Code

```
tmpdst = 0; j = 0;
for( i = 0; i < 32; i++) {
 if ( i >= src1[4:0] && i <= src1[9:5]) tmpdst[j++] = src2[i];
};
dst = tmpdst[j] ? signext(tmpdst) : tmpdst ;
```

EXTU *Extract and Zero-Extend a Bit Field Bounded by Two Immediate Values*

Syntax

EXTU *ucst5_1, ucst5_2, src2, dst*

Functional unit = L

Opcode

32 bit

31	30	29	28	27	26	25 – 21	20 – 16	15 – 13	12 – 10	9								0
x	x	x	x	x	x	*ucst5_1*	*ucst5_2*	*dst*	*src2*	1	0	1	0	0	0	0	0	0
						5	5	3	3	*opcode*								

Description

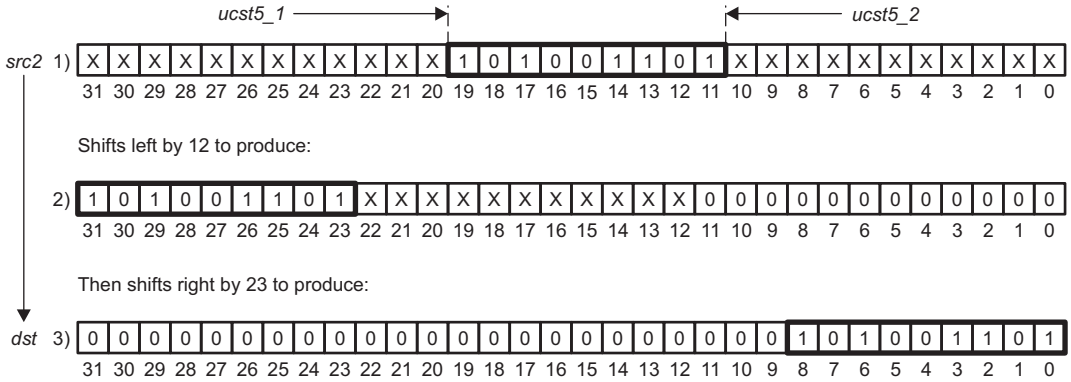
For *ucst5_1* ≥ *ucst5_2*, the field in *src2* as specified by *ucst5_2* to *ucst5_1* is extracted and zero-extended to 32 bits and is written to *dst*. *src2* is left unchanged. *ucst5_2* is the LSB of the field and *ucst5_1* is the MSB of the field to be extracted. In other words, *ucst5_2* and *ucst5_1* represent the beginning and ending bits, respectively, of the field to be extracted. The LSB location of *src2* is bit 0 and the MSB location of *src2* is bit 31.

Valid values of *ucst5_1* and *ucst5_2* are:

- 0 ≤ *ucst5_1* ≤ 31; 0 ≤ *ucst5_2* ≤ 31
- *ucst5_1* ≥ *ucst5_2*

For *ucst5_1* < *ucst5_2*, the result is undefined.

In the following example, *ucst5_2* is 11 and *ucst5_1* is 19.



Condition Codes

CSR[2]EQ = (dst == 0)

Pseudo Code

```
tmpdst = 0; j = 0;
for( i = 0; i < 32; i++) {
 if ( i >= ucst5_2 && i <= ucst5_1) tmpdst[j++] = src2[i];
};
dst = tmpdst ;
```

EXTU *Extract and Zero-Extend a Bit Field Bounded by Two Register Values*

Syntax

EXTU *src1, src2, dst*

Functional unit = L

Opcode

32 bit

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
x	x	x	x	x	x	x	x	x	x	x	1	0	0	1	1	*dst*			*src2*			*src1*			0	0	0	0	1	0	0
																3			3			3			*opcode*						

Description

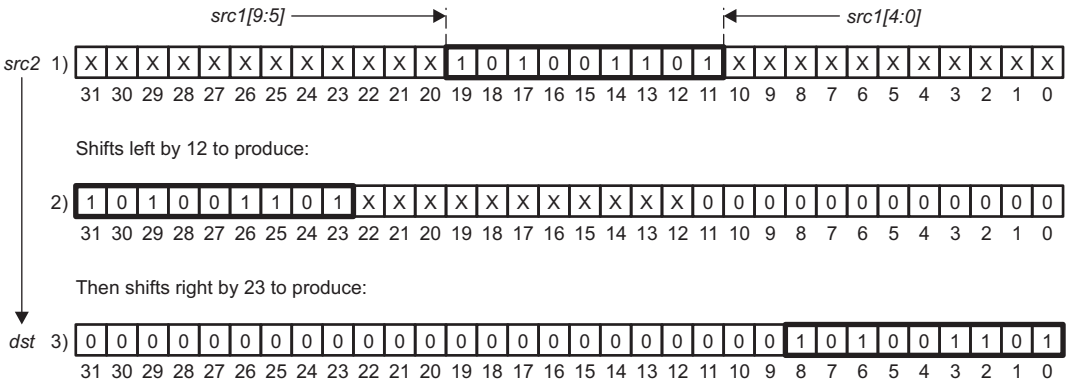
For *src1[9:5]* ≥ *src1[4:0]*, the field in *src2* as specified by *src1[4:0]* to *src1[9:5]* is extracted and zero-extended to 32 bits and is written to *dst*. *src2* is left unchanged. *src1[4:0]* is the LSB of the field and *src1[9:5]* is the MSB of the field to be extracted. In other words, *src1[4:0]* and *src1[9:5]* represent the beginning and ending bits, respectively, of the field to be extracted. The LSB location of *src2* is bit 0 and the MSB location of *src2* is bit 31.

Valid values of *src1[9:5]* and *src1[4:0]* are:

- 0 ≤ *src1[9:5]* ≤ 31; 0 ≤ *src1[4:0]* ≤ 31
- *src1[9:5]* ≥ *src1[4:0]*

For *src1[9:5]* < *src1[4:0]*, the result is undefined.

In the following example, *src1[4:0]* is 11 and *src1[9:5]* is 19.



Condition Codes

CSR:EQ = (dst == 0)

Pseudo Code

```
tmpdst = 0; j = 0;
for( i = 0; i < 32; i++) {
 if ( i >= src1[4:0] && i <= src1[9:5]) tmpdst[j++] = src2[i];
};
dst = tmpdst ;
```

IDLE *Idle Until Interrupt or Reset*

Syntax

IDLE

Functional unit = S

Opcode

16 bit

15	14	13	12	11	10	9									0
x	x	x	x	x	x	1	1	0	1	1	1	1	1	1	1
										opcode					

Description

The **IDLE** instruction causes the CPU to wait for all pending instruction and data memory transactions to complete and then go to an idle state. The idle state is the equivalent of an endless loop. Termination of this loop occurs on an interrupt or reset. However, the CPU does not actually continue fetch/executing any actual instructions. On acceptance of an interrupt, PC + 1 is stored into the interrupt return pointer register (IRP), subsequent instruction execution after return from the interrupt begins at the instruction following the **IDLE** instruction.

For more details on usage of IDLE instruction, see Section 6.2.6.3.

Condition Codes

None

Pseudo Code

None

LDB(U) *Load Byte from Memory with a 3-Bit Unsigned Constant Offset*

Syntax

LDB **+baseR[ucst3], dst*

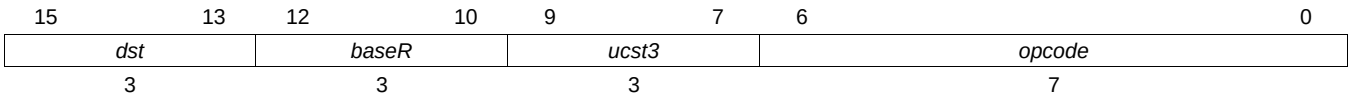
or

LDBU **+baseR[ucst3], dst*

Functional unit = D

Opcode

16 bit



Syntax	Opcode	Load Type
LDB **+baseR[ucst3], dst*	101 1010	Load byte (sign-extended)
LDBU **+baseR[ucst3], dst*	101 1000	Load byte unsigned (zero-extended)

Description

Loads a byte (signed or unsigned) from memory (effective address) to *dst*. The memory address is formed from a base address register (*baseR*) and an offset (number of bytes) that is a 3-bit unsigned constant (*ucst3*). If an offset is not given, the assembler assigns an offset of zero.There must be brackets, [ ], around the specified offset, if using the optional offset parameter.

The square brackets, [ ], indicate that the *ucst3* is scaled by a left-shift of 0 bits. After scaling, *ucst3* is added to *baseR*. The result of the calculation is the effective address to be accessed in memory and the content loaded into *dst*.

The content is loaded into the 8 LSBs of *dst*. For the **LDB** instruction, the upper 24 bits of *dst* are sign-extended; for the **LDBU** instruction, the upper 24 bits of *dst* are zero-extended.

Condition Codes

None

Pseudo Code

```
dst = *(baseR[ucst3])
```

LDB(U) *Load Byte from Memory with a 16-Bit Unsigned Constant Offset*

Syntax

LDB **+baseR[ucst16], dst*

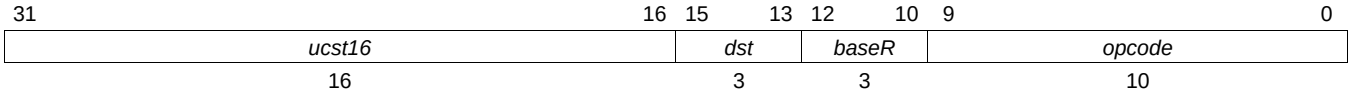
or

LDBU **+baseR[ucst16], dst*

Functional unit = D

Opcode

32 bit



Syntax	Opcode	Load Type
LDB **+baseR[ucst16], dst*	01 0000 1000	Load byte (sign-extended)
LDBU **+baseR[ucst16], dst*	00 0000 1000	Load byte unsigned (zero-extended)

Description

Loads a byte (signed or unsigned) from memory (effective address) to *dst*. The memory address is formed from a base address register (*baseR*) and an offset (number of bytes) that is a 16-bit unsigned constant (*ucst16*). If an offset is not given, the assembler assigns an offset of zero. Brackets,[], must surround the specified offset, if using the optional offset parameter.

The square brackets, [ ], indicate that the *ucst16* is scaled by a left-shift of 0 bits. After scaling, *ucst16* is added to *baseR*. The result of the calculation is the effective address to be accessed in memory and the content loaded into *dst*.

The content is loaded into the 8 LSBs of *dst*. For the **LDB** instruction, the upper 24 bits of *dst* are sign-extended; for the **LDBU** instruction, the upper 24 bits of *dst* are zero-extended.

Condition Codes

None

Pseudo Code

```
dst = *(baseR[ucst16])
```

LDB(U) *Load Byte from Memory with a Register Offset*

Syntax

LDB **+baseR[src1], dst*

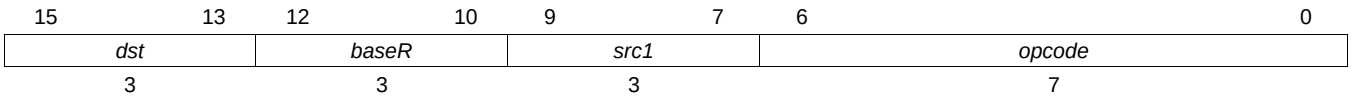
or

LDBU **+baseR[src1], dst*

Functional unit = D

Opcode

16 bit



Syntax	Opcode	Load Type
LDB **+baseR[src1], dst*	101 0101	Load byte (sign-extended)
LDBU **+baseR[src1], dst*	101 0011	Load byte unsigned (zero-extended)

Description

Loads a byte (signed or unsigned) from memory (effective address) to *dst*. The memory address is formed from a base address register (*baseR*) and an offset (number of bytes) that is a register (*src1*). If an offset is not given, the assembler assigns an offset of zero. Brackets, [ ], must surround the specified offset, if using the optional offset parameter.

The square brackets, [ ], indicate that the *src1* is scaled by a left-shift of 0 bits. After scaling, *src1* is added to *baseR*. The result of the calculation is the effective address to be accessed in memory and the content loaded into *dst*.

The content is loaded into the 8 LSBs of *dst*. For the **LDB** instruction, the upper 24 bits of *dst* are sign-extended; for the **LDBU** instruction, the upper 24 bits of *dst* are zero-extended.

Condition Codes

None

Pseudo Code

```
dst = *(baseR[src1])
```

LDB(U) *Load Byte from Memory, Postincrement with a 3-Bit Unsigned Constant Offset*

Syntax

LDB **baseR++[ucst3], dst*

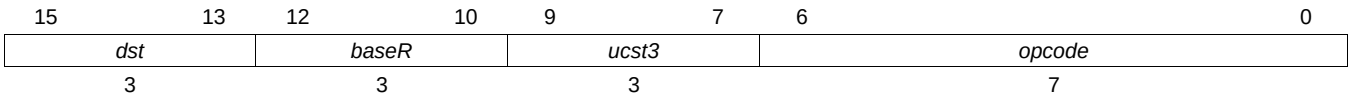
or

LDBU **baseR++[ucst3], dst*

Functional unit = D

Opcode

16 bit



Syntax	Opcode	Load Type
LDB **baseR++[ucst3], dst*	110 0100	Load byte (sign-extended)
LDBU **baseR++[ucst3], dst*	110 0010	Load byte unsigned (zero-extended)

Description

Loads a byte (signed or unsigned) from memory (effective address) to *dst*. The value of *baseR* is the effective address to be accessed in memory and the content loaded into *dst*. After accessing, an offset (number of bytes) that is a 3-bit unsigned constant (*ucst3*) is added to *baseR* at the end of the EXE phase. Brackets, [ ],must be around the specified offset, if using the optional offset parameter.

The content is loaded into the 8 LSBs of *dst*. For the **LDB** instruction, the upper 24 bits of *dst* are sign-extended; for the **LDBU** instruction, the upper 24 bits of *dst* are zero-extended.

Condition Codes

None

Pseudo Code

```
dst = *(baseR++[ucst3])
```

LDB(U) *Load Byte from Memory, Postincrement with a Register Offset*

Syntax

LDB **baseR++[src1], dst*

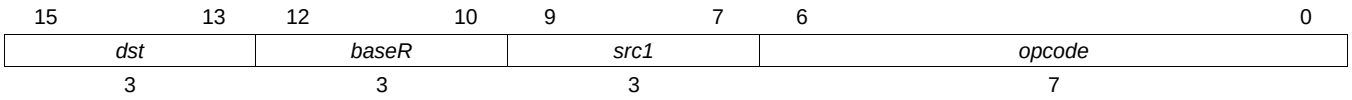
or

LDBU **baseR++[src1], dst*

Functional unit = D

Opcode

16 bit



Syntax	Opcode	Load Type
LDB **baseR++[src1], dst*	101 1111	Load byte (sign-extended)
LDBU **baseR++[src1], dst*	101 1101	Load byte unsigned (zero-extended)

Description

Loads a byte (signed or unsigned) from memory (effective address) to *dst*. The value of *baseR* is the effective address to be accessed in memory and the content loaded into *dst*. After accessing, an offset (number of bytes) that is a register (*src1*) is added to *baseR* at the end of the EXE phase. Brackets, [ ],must surround the specified offset, if using the optional offset parameter.

The content is loaded into the 8 LSBs of *dst*. For the **LDB** instruction, the upper 24 bits of *dst* are sign-extended; for the **LDBU** instruction, the upper 24 bits of *dst* are zero-extended.

Condition Codes

None

Pseudo Code

```
dst = *(baseR++[src1])
```

LDB(U) *Load Byte from Memory with a SP-Relative 6-Bit Unsigned Constant Offset*

Syntax

LDB *+**SP***[ucst6], dst*

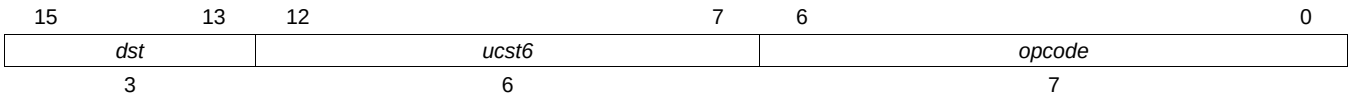
or

LDBU *+**SP***[ucst6], dst*

Functional unit = D

Opcode

16 bit



Syntax	Opcode	Load Type
LDB *+**SP***[ucst6], dst*	110 1001	Load byte (sign-extended)
LDBU *+**SP***[ucst6], dst*	110 0111	Load byte unsigned (zero-extended)

Description

Loads a byte (signed or unsigned) from memory (effective address) to *dst*. The memory address is formed from the stack pointer (SP) and an offset (number of bytes) that is a 6-bit unsigned constant (*ucst6*). If an offset is not given, the assembler assigns an offset of zero. Brackets, [ ],must surround the specified offset, if using the optional offset parameter.

The square brackets, [ ], indicate that the *ucst6* is scaled by a left-shift of 0 bits. After scaling, *ucst6* is added to SP. The result of the calculation is the effective address to be accessed in memory and the content loaded into *dst*.

The content is loaded into the 8 LSBs of *dst*. For the **LDB** instruction, the upper 24 bits of *dst* are sign-extended; for the **LDBU** instruction, the upper 24 bits of *dst* are zero-extended.

Condition Codes

None

Pseudo Code

```
dst = *(SP[ucst6])
```

LDB(U) *Load Byte from Memory with a SP-Relative 19-Bit Unsigned Constant Offset*

Syntax

LDB *+**SP***[ucst19], dst*

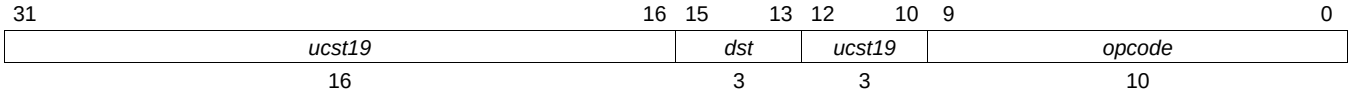
or

LDBU *+**SP***[ucst19], dst*

Functional unit = D

Opcode

32 bit



Syntax	Opcode	Load Type
LDB *+**SP***[ucst19], dst*	01 0000 0111	Load byte (sign-extended)
LDBU *+**SP***[ucst19], dst*	00 0000 0111	Load byte unsigned (zero-extended)

Description

Loads a byte (signed or unsigned) from memory (effective address) to *dst*. The memory address is formed from the stack pointer (SP) and an offset (number of bytes) that is a 19-bit unsigned constant (*ucst19*). If an offset is not given, the assembler assigns an offset of zero. Brackets, [ ],must surround the specified offset, if using the optional offset parameter.

The square brackets, [ ], indicate that the *ucst19* is scaled by a left-shift of 0 bits. After scaling, *ucst19* is added to SP. The result of the calculation is the effective address to be accessed in memory and the content loaded into *dst*.

The content is loaded into the 8 LSBs of *dst*. For the **LDB** instruction, the upper 24 bits of *dst* are sign-extended; for the **LDBU** instruction, the upper 24 bits of *dst* are zero-extended.

Condition Codes

None

Pseudo Code

```
dst = *(SP[ucst19])
```

LDB(U) *Load Byte from Memory with a GDP-Relative 19-Bit Unsigned Constant Offset*

Syntax

LDB *+**GDP***[ucst19], dst*

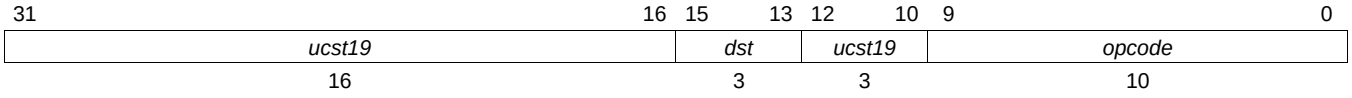
or

LDBU *+**GDP***[ucst19], dst*

Functional unit = D

Opcode

32 bit



Syntax	Opcode	Load Type
LDB *+**GDP***[ucst19], dst*	01 0000 0110	Load byte (sign-extended)
LDBU *+**GDP***[ucst19], dst*	00 0000 0110	Load byte unsigned (zero-extended)

Description

Loads a byte (signed or unsigned) from memory (effective address) to *dst*. The memory address is formed from the global data pointer register (GDP) and an offset (number of bytes) that is a 19-bit unsigned constant (*ucst19*). If an offset is not given, the assembler assigns an offset of zero. Brackets, [ ],must surround the specified offset, if using the optional offset parameter.

The square brackets, [ ], indicate that the *ucst19* is scaled by a left-shift of 0 bits. After scaling, *ucst19* is added to GDP. The result of the calculation is the effective address to be accessed in memory and the content loaded into *dst*.

The content is loaded into the 8 LSBs of *dst*. For the **LDB** instruction, the upper 24 bits of *dst* are sign-extended; for the **LDBU** instruction, the upper 24 bits of *dst* are zero-extended.

Condition Codes

None

Pseudo Code

```
dst = *(GDP[ucst19])
```

LDH(U) *Load Halfword from Memory with a 3-Bit Unsigned Constant Offset*

Syntax

LDH **+baseR[ucst3], dst*

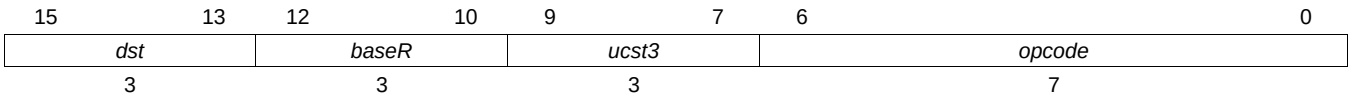
or

LDHU **+baseR[ucst3], dst*

Functional unit = D

Opcode

16 bit



Syntax	Opcode	Load Type
LDH **+baseR[ucst3], dst*	101 1011	Load halfword (sign-extended)
LDHU **+baseR[ucst3], dst*	101 1001	Load halfword unsigned (zero-extended)

Description

Loads a halfword (signed or unsigned) from memory (effective address) to *dst*. The memory address is formed from a base address register (*baseR*) and an offset (number of halfwords) that is a 3-bit unsigned constant (*ucst3*). If an offset is not given, the assembler assigns an offset of zero. Brackets, [ ],must surround the specified offset, if using the optional offset parameter.

The square brackets, [ ], indicate that the *ucst3* is scaled by a left-shift of 1 bit. After scaling, *ucst3* is added to *baseR*. The result of the calculation is the effective address to be accessed in memory and the content loaded into *dst*.

The content is loaded into the 16 LSBs of *dst*. For the **LDH** instruction, the upper 16 bits of *dst* are sign-extended; for the **LDHU** instruction, the upper 16 bits of *dst* are zero-extended.

Halfword addresses must be aligned on halfword (LSB is 0) boundaries.

Condition Codes

None

Pseudo Code

```
dst = *(baseR[ucst3])
```

LDH(U) *Load Halfword from Memory with a 16-Bit Unsigned Constant Offset*

Syntax

LDH **+baseR[ucst16], dst*

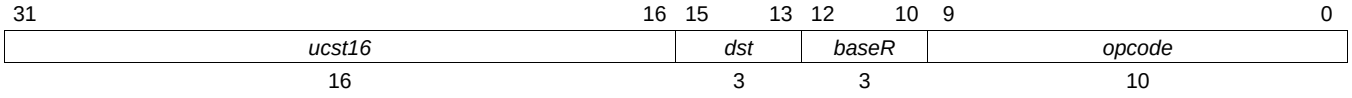
or

LDHU **+baseR[ucst16], dst*

Functional unit = D

Opcode

32 bit



Syntax	Opcode	Load Type
LDH **+baseR[ucst16], dst*	01 1000 1000	Load halfword (sign-extended)
LDHU **+baseR[ucst16], dst*	00 1000 1000	Load halfword unsigned (zero-extended)

Description

Loads a halfword (signed or unsigned) from memory (effective address) to *dst*. The memory address is formed from a base address register (*baseR*) and an offset (number of halfwords) that is a 16-bit unsigned constant (*ucst16*). If an offset is not given, the assembler assigns an offset of zero. Brackets, [ ],must surround the specified offset, if using the optional offset parameter.

The square brackets, [ ], indicate that the *ucst16* is scaled by a left-shift of 1 bit. After scaling, *ucst16* is added to *baseR*. The result of the calculation is the effective address to be accessed in memory and the content loaded into *dst*.

The content is loaded into the 16 LSBs of *dst*. For the **LDH** instruction, the upper 16 bits of *dst* are sign-extended; for the **LDHU** instruction, the upper 16 bits of *dst* are zero-extended.

Halfword addresses must be aligned on halfword (LSB is 0) boundaries.

Condition Codes

None

Pseudo Code

```
dst = *(baseR[ucst16])
```

LDH(U) *Load Halfword from Memory with a Register Offset*

Syntax

LDH **+baseR[src1], dst*

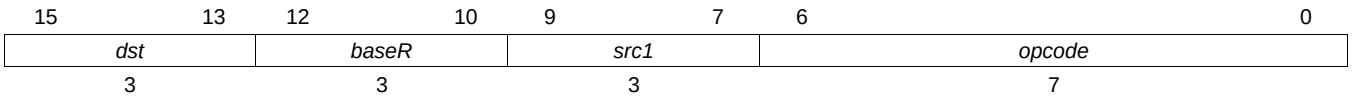
or

LDHU **+baseR[src1], dst*

Functional unit = D

Opcode

16 bit



Syntax	Opcode	Load Type
LDH **+baseR[src1], dst*	101 0110	Load halfword (sign-extended)
LDHU **+baseR[src1], dst*	101 0100	Load halfword unsigned (zero-extended)

Description

Loads a halfword (signed or unsigned) from memory (effective address) to *dst*. The memory address is formed from a base address register (*baseR*) and an offset (number of halfwords) that is a register (*src1*). If an offset is not given, the assembler assigns an offset of zero. Brackets, [ ],must surround the specified offset, if using the optional offset parameter.

The square brackets, [ ], indicate that the *src1* is scaled by a left-shift of 1 bit. After scaling, *src1* is added to *baseR*. The result of the calculation is the effective address to be accessed in memory and the content loaded into *dst*.

The content is loaded into the 16 LSBs of *dst*. For the **LDH** instruction, the upper 16 bits of *dst* are sign-extended; for the **LDHU** instruction, the upper 16 bits of *dst* are zero-extended.

Halfword addresses must be aligned on halfword (LSB is 0) boundaries.

Condition Codes

None

Pseudo Code

```
dst = *(baseR[src1])
```

LDH(U) *Load Halfword from Memory, Postincrement with a 3-Bit Unsigned Constant Offset*

Syntax

LDH **baseR++[ucst3], dst*

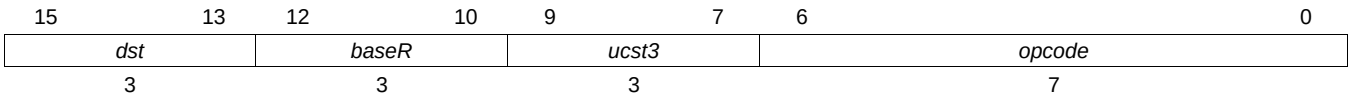
or

LDHU **baseR++[ucst3], dst*

Functional unit = D

Opcode

16 bit



Syntax	Opcode	Load Type
LDH **baseR++[ucst3], dst*	110 0101	Load halfword (sign-extended)
LDHU **baseR++[ucst3], dst*	110 0011	Load halfword unsigned (zero-extended)

Description

Loads a halfword (signed or unsigned) from memory (effective address) to *dst*. The value of *baseR* is the effective address to be accessed in memory and the content loaded into *dst*. After accessing, an offset (number of halfwords) that is a 3-bit unsigned constant (*ucst3*) is added to *baseR* at the end of the EXE phase. Brackets, [ ],must surround the specified offset, if using the optional offset parameter.

The content is loaded into the 16 LSBs of *dst*. For the **LDH** instruction, the upper 16 bits of *dst* are sign-extended; for the **LDHU** instruction, the upper 16 bits of *dst* are zero-extended.

Halfword addresses must be aligned on halfword (LSB is 0) boundaries.

Condition Codes

None

Pseudo Code

```
dst = *(baseR++[ucst3])
```

LDH(U) *Load Halfword from Memory, Postincrement with a Register Offset*

Syntax

LDH **baseR++[src1], dst*

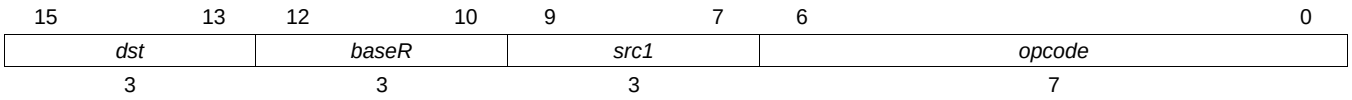
or

LDHU **baseR++[src1], dst*

Functional unit = D

Opcode

16 bit



Syntax	Opcode	Load Type
LDH **baseR++[src1], dst*	110 0000	Load halfword (sign-extended)
LDHU **baseR++[src1], dst*	101 1110	Load halfword unsigned (zero-extended)

Description

Loads a halfword (signed or unsigned) from memory (effective address) to *dst*. The value of *baseR* is the effective address to be accessed in memory and the content loaded into *dst*. After accessing, an offset (number of halfwords) that is a register (*src1*) is added to *baseR* at the end of the EXE phase. Brackets, [ ],must surround the specified offset, if using the optional offset parameter.

The content is loaded into the 16 LSBs of *dst*. For the **LDH** instruction, the upper 16 bits of *dst* are sign-extended; for the **LDHU** instruction, the upper 16 bits of *dst* are zero-extended.

Halfword addresses must be aligned on halfword (LSB is 0) boundaries.

Condition Codes

None

Pseudo Code

```
dst = *(baseR++[src1])
```

LDH(U) *Load Halfword from Memory with a SP-Relative 6-Bit Unsigned Constant Offset*

Syntax

LDH *+**SP***[ucst6], dst*

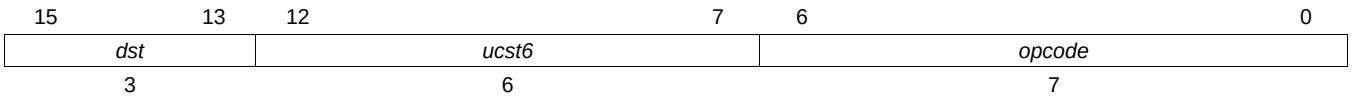
or

LDHU *+**SP***[ucst6], dst*

Functional unit = D

Opcode

16 bit



Syntax	Opcode	Load Type
LDH *+**SP***[ucst6], dst*	110 1010	Load halfword (sign-extended)
LDHU *+**SP***[ucst6], dst*	110 1000	Load halfword unsigned (zero-extended)

Description

Loads a halfword (signed or unsigned) from memory (effective address) to *dst*. The memory address is formed from the stack pointer (SP) and an offset (number of halfwords) that is a 6-bit unsigned constant (*ucst6*). If an offset is not given, the assembler assigns an offset of zero. Brackets, [ ],must surround the specified offset, if using the optional offset parameter.

The square brackets, [ ], indicate that the *ucst6* is scaled by a left-shift of 1 bit. After scaling, *ucst6* is added to SP. The result of the calculation is the effective address to be accessed in memory and the content loaded into *dst*.

The content is loaded into the 16 LSBs of *dst*. For the **LDH** instruction, the upper 16 bits of *dst* are sign-extended; for the **LDHU** instruction, the upper 16 bits of *dst* are zero-extended.

Halfword addresses must be aligned on halfword (LSB is 0) boundaries.

Condition Codes

None

Pseudo Code

```
dst = *(SP[ucst6])
```

LDH(U) *Load Halfword from Memory with a SP-Relative 19-Bit Unsigned Constant Offset*

Syntax

LDH *+**SP***[ucst19], dst*

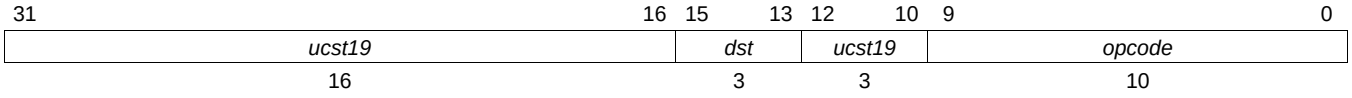
or

LDHU *+**SP***[ucst19], dst*

Functional unit = D

Opcode

32 bit



Syntax	Opcode	Load Type
LDH *+**SP***[ucst19], dst*	01 1000 0111	Load halfword (sign-extended)
LDHU *+**SP***[ucst19], dst*	00 1000 0111	Load halfword unsigned (zero-extended)

Description

Loads a halfword (signed or unsigned) from memory (effective address) to *dst*. The memory address is formed from the stack pointer (SP) and an offset (number of halfwords) that is a 19-bit unsigned constant (*ucst19*). If an offset is not given, the assembler assigns an offset of zero. Brackets, [ ],must surround the specified offset, if using the optional offset parameter.

The square brackets, [ ], indicate that the *ucst19* is scaled by a left-shift of 1 bit. After scaling, *ucst19* is added to SP. The result of the calculation is the effective address to be accessed in memory and the content loaded into *dst*.

The content is loaded into the 16 LSBs of *dst*. For the **LDH** instruction, the upper 16 bits of *dst* are sign-extended; for the **LDHU** instruction, the upper 16 bits of *dst* are zero-extended.

Halfword addresses must be aligned on halfword (LSB is 0) boundaries.

Condition Codes

None

Pseudo Code

```
dst = *(SP[ucst19])
```

LDH(U) *Load Halfword from Memory with a GDP-Relative 19-Bit Unsigned Constant Offset*

Syntax

LDH *+**GDP***[ucst19], dst*

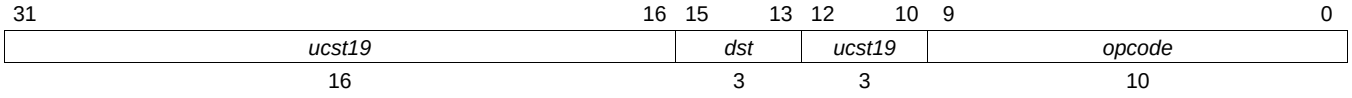
or

LDHU *+**GDP***[ucst19], dst*

Functional unit = D

Opcode

32 bit



Syntax	Opcode	Load Type
LDH *+**GDP***[ucst19], dst*	01 1000 0110	Load halfword (sign-extended)
LDHU *+**GDP***[ucst19], dst*	00 1000 0110	Load halfword unsigned (zero-extended)

Description

Loads a halfword (signed or unsigned) from memory (effective address) to *dst*. The memory address is formed from the global data pointer register (GDP) and an offset (number of halfwords) that is a 19-bit unsigned constant (*ucst19*). If an offset is not given, the assembler assigns an offset of zero. Brackets, [ ],must surround the specified offset, if using the optional offset parameter.

The square brackets, [ ], indicate that the *ucst19* is scaled by a left-shift of 1 bit. After scaling, *ucst19* is added to GDP. The result of the calculation is the effective address to be accessed in memory and the content loaded into *dst*.

The content is loaded into the 16 LSBs of *dst*. For the **LDH** instruction, the upper 16 bits of *dst* are sign-extended; for the **LDHU** instruction, the upper 16 bits of *dst* are zero-extended.

Halfword addresses must be aligned on halfword (LSB is 0) boundaries.

Condition Codes

None

Pseudo Code

```
dst = *(GDP[ucst19])
```

LDW *Load Word from Memory with a 3-Bit Unsigned Constant Offset*

Syntax

LDW **+baseR[ucst3], dst*

Functional unit = D

Opcode

16 bit

15 13	12 10	9 7	6						0
dst	*baseR*	*ucst3*	1	0	1	1	1	0	0
3	3	3	*opcode*						

Description

Loads a word from memory (effective address) to *dst*. The memory address is formed from a base address register (*baseR*) and an offset (number of words) that is a 3-bit unsigned constant (*ucst3*). If an offset is not given, the assembler assigns an offset of zero. Brackets, [ ],must surround the specified offset, if using the optional offset parameter.

The square brackets, [ ], indicate that the *ucst3* is scaled by a left-shift of 2 bits. After scaling, *ucst3* is added to *baseR*. The result of the calculation is the effective address to be accessed in memory and the content loaded into *dst*.

The entire content is loaded into *dst*.

Word addresses must be aligned on word (two LSBs are 0) boundaries.

Condition Codes

None

Pseudo Code

```
dst = *(baseR[ucst3])
```

LDW *Load Word from Memory with a 16-Bit Unsigned Constant Offset*

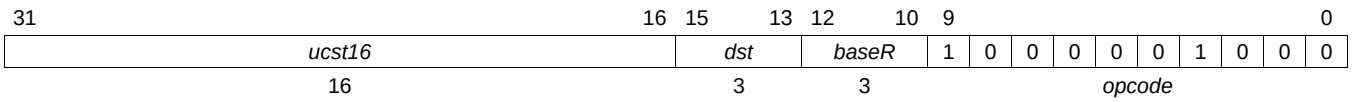
Syntax

LDW **+baseR[ucst16], dst*

Functional unit = D

Opcode

32 bit



Description

Loads a word from memory (effective address) to *dst*. The memory address is formed from a base address register (*baseR*) and an offset (number of words) that is a 16-bit unsigned constant (*ucst16*). If an offset is not given, the assembler assigns an offset of zero. Brackets, [ ],must surround the specified offset, if using the optional offset parameter.

The square brackets, [ ], indicate that the *ucst16* is scaled by a left-shift of 2 bits. After scaling, *ucst16* is added to *baseR*. The result of the calculation is the effective address to be accessed in memory and the content loaded into *dst*.

The entire content is loaded into *dst*.

Word addresses must be aligned on word (two LSBs are 0) boundaries.

Condition Codes

None

Pseudo Code

```
dst = *(baseR[ucst16])
```

LDW *Load Word from Memory with a Register Offset*

Syntax

LDW **+baseR[src1], dst*

Functional unit = D

Opcode

16 bit

15 13	12 10	9 7	6						0
dst	*baseR*	*src1*	1	0	1	0	1	1	1
3	3	3	*opcode*						

Description

Loads a word from memory (effective address) to *dst*. The memory address is formed from a base address register (*baseR*) and an offset (number of words) that is a register (*src1*). If an offset is not given, the assembler assigns an offset of zero. Brackets, [ ],must surround the specified offset, if using the optional offset parameter.

The square brackets, [ ], indicate that the *src1* is scaled by a left-shift of 2 bits. After scaling, *src1* is added to *baseR*. The result of the calculation is the effective address to be accessed in memory and the content loaded into *dst*.

The entire content is loaded into *dst*.

Word addresses must be aligned on word (two LSBs are 0) boundaries.

Condition Codes

None

Pseudo Code

```
dst = *(baseR[src1])
```

LDW *Load Word from Memory, Postincrement with a 3-Bit Unsigned Constant Offset*

Syntax

LDW **baseR++[ucst3], dst*

Functional unit = D

Opcode

16 bit

15 13	12 10	9 7	6						0
dst	*baseR*	*ucst3*	1	1	0	0	1	1	0
3	3	3	*opcode*						

Description

Loads a word from memory (effective address) to *dst*. The value of *baseR* is the effective address to be accessed in memory and the content loaded into *dst*. After accessing, an offset (number of words) that is a 3-bit unsigned constant (*ucst3*) is added to *baseR* at the end of the EXE phase. Brackets, [ ],must surround the specified offset, if using the optional offset parameter.

The entire content is loaded into *dst*.

Word addresses must be aligned on word (two LSBs are 0) boundaries.

Condition Codes

None

Pseudo Code

```
dst = *(baseR++[ucst3])
```

LDW *Load Word from Memory, Postincrement with a Register Offset*

Syntax

LDW **baseR++[src1], dst*

Functional unit = D

Opcode

16 bit

15 13	12 10	9 7	6						0
dst	*baseR*	*src1*	1	1	0	0	0	0	1
3	3	3	*opcode*						

Description

Loads a word from memory (effective address) to *dst*. The value of *baseR* is the effective address to be accessed in memory and the content loaded into *dst*. After accessing, an offset (number of words) that is a register (*src1*) is added to *baseR* at the end of the EXE phase. Brackets, [ ],must surround the specified offset, if using the optional offset parameter.

The entire content is loaded into *dst*.

Word addresses must be aligned on word (two LSBs are 0) boundaries.

Condition Codes

None

Pseudo Code

```
dst = *(baseR++[src1])
```

LDW *Load Word from Memory with a SP-Relative 6-Bit Unsigned Constant Offset*

Syntax

LDW *+**SP***[ucst6], dst*

Functional unit = D

Opcode

16 bit

15 13	12 7	6						0
dst	*ucst6*	1	1	0	1	0	1	1
3	6	*opcode*						

Description

Loads a word from memory (effective address) to *dst*. The memory address is formed from the stack pointer (SP) and an offset (number of words) that is a 6-bit unsigned constant (*ucst6*). If an offset is not given, the assembler assigns an offset of zero. Brackets, [ ],must surround the specified offset, if using the optional offset parameter.

The square brackets, [ ], indicate that the *ucst6* is scaled by a left-shift of 2 bits. After scaling, *ucst6* is added to SP. The result of the calculation is the effective address to be accessed in memory and the content loaded into *dst*.

The entire content is loaded into *dst*.

Word addresses must be aligned on word (two LSBs are 0) boundaries.

Condition Codes

None

Pseudo Code

```
dst = *(SP[ucst6])
```

LDW *Load Word from Memory with a SP-Relative 19-Bit Unsigned Constant Offset*

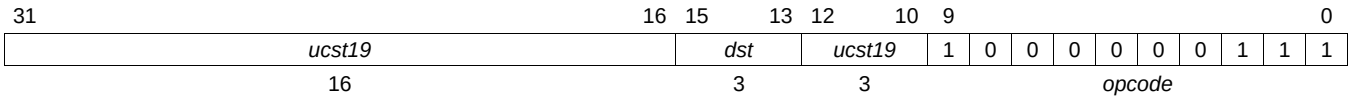
Syntax

LDW *+**SP***[ucst19], dst*

Functional unit = D

Opcode

32 bit



Description

Loads a word from memory (effective address) to *dst*. The memory address is formed from the stack pointer (SP) and an offset (number of words) that is a 19-bit unsigned constant (*ucst19*). If an offset is not given, the assembler assigns an offset of zero. Brackets, [ ],must surround the specified offset, if using the optional offset parameter.

The square brackets, [ ], indicate that the *ucst19* is scaled by a left-shift of 2 bits. After scaling, *ucst19* is added to SP. The result of the calculation is the effective address to be accessed in memory and the content loaded into *dst*.

The entire content is loaded into *dst*.

Word addresses must be aligned on word (two LSBs are 0) boundaries.

Condition Codes

None

Pseudo Code

```
dst = *(SP[ucst19])
```

LDW *Load Word from Memory with a GDP-Relative 19-Bit Unsigned Constant Offset*

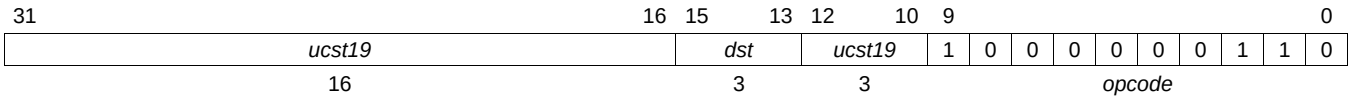
Syntax

LDW *+**GDP***[ucst19], dst*

Functional unit = D

Opcode

32 bit



Description

Loads a word from memory (effective address) to *dst*. The memory address is formed from the global data pointer register (GDP) and an offset (number of words) that is a 19-bit unsigned constant (*ucst19*). If an offset is not given, the assembler assigns an offset of zero. Brackets, [ ],must surround the specified offset, if using the optional offset parameter.

The square brackets, [ ], indicate that the *ucst19* is scaled by a left-shift of 2 bits. After scaling, *ucst19* is added to GDP. The result of the calculation is the effective address to be accessed in memory and the content loaded into *dst*.

The entire content is loaded into *dst*.

Word addresses must be aligned on word (two LSBs are 0) boundaries.

Condition Codes

None

Pseudo Code

```
dst = *(GDP[ucst19])
```

LDRF *Load Register File from Stack Pointer*

Syntax

LDRF *op1, op2*

Functional unit = D

Opcode

16 bit

15 13	12 10	9	8	7	6						0
op2	*op1*	x	x	x	1	0	1	0	0	0	1
3	3							*opcode*			

Description

The current base register file is loaded from the starting address specified by the preincremented value of the stack pointer (SP). *op1* specifies the ending register (last register loaded) and *op2* specifies the beginning register (first register loaded). All registers between *op1* and *op2* are loaded inclusive. The registers are loaded in order from highest register (*op2*) to lowest register (*op1*).

Valid values of *op1* and *op2* are: *op1* ≤ *op2*

For *op1* > *op2*, results in unspecified behavior.

This is a multicycle blocking instruction. The number of cycles is determined by the registers, for example, **LDRF** R0, R5 requires 6 cycles to complete.

Condition Codes

None

Pseudo Code

```
for ( i = op2; i <= op1 ; i--) {
SP += 4;
Ri = *SP;
}
```

LMBD *Left Most Bit Detection*

Syntax

LMBD *ucst3, src2, dst*

Functional unit = L

Opcode

32 bit

31	30	29	28	27	26	25	24	23	22	21	20	19	18 16	15 13	12 10	9								0
x	x	x	x	x	x	x	x	x	x	x	x	x	*ucst3*	*dst*	*src2*	1	1	0	0	0	0	0	0	1	1
													3	3	3					*opcode*					

Description

The LSB of *ucst3* determines whether to search for a leftmost 1 or 0 in *src2*. The number of bits to the left of the first 1 or 0 when searching for a 1 or 0, respectively, is placed in *dst*.

- *ucst3* = 0, searches for leftmost 0 (count leading ones)
- *ucst3* = 1, searches for leftmost 1 (count leading zeros)
- Other values of *ucst3* are reserved. The result is undefined if reserved values are used.

The following diagrams illustrate the operation of **LMBD** for several cases.

When searching for 0 in *src2*, **LMBD** returns 0:

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	1	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

When searching for 1 in *src2*, **LMBD** returns 4:

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	0	0	0	1	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x	x

When searching for 0 in *src2*, **LMBD** returns 32:

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1

Condition Codes

CSR:EQ = (dst == 0)

Pseudo Code

```
dst = if (ucst3 == 0), lmb0(src2)
dst = if (ucst3 == 1), lmb1(src2)
```

MAX *Maximum of Two Signed Register Values*

Syntax

MAX *src1, src2, dst*

Functional unit = L

Opcode

32 bit

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	13	12	10	9	7	6						0
x	x	x	x	x	x	x	x	x	x	x	1	1	0	0	0	*dst*		*src2*		*src1*		0	0	0	0	1	0	0
																3		3		3					*opcode*			

Description

Performs a maximum operation on the signed values in *src1* and *src2*. The larger value is loaded in *dst*.

Condition Codes

CSR:EQ = (dst == 0)

Pseudo Code

```
dst = (src2 > src1) ? src2:src1
```

MAXU *Maximum of Two Unsigned Register Values*

Syntax

MAXU *src1, src2, dst*

Functional unit = L

Opcode

32 bit

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	13	12	10	9	7	6						0
x	x	x	x	x	x	x	x	x	x	x	1	1	0	1	0	*dst*		*src2*		*src1*		0	0	0	0	1	0	0
																3		3		3					*opcode*			

Description

Performs a maximum operation on the unsigned values in *src1* and *src2*. The larger value is loaded in *dst*.

Condition Codes

CSR:EQ = (dst == 0)

Pseudo Code

```
dst = (src2 > src1) ? src2:src1
```

MIN *Minimum of Two Signed Register Values*

Syntax

MIN *src1, src2, dst*

Functional unit = L

Opcode

32 bit

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
x	x	x	x	x	x	x	x	x	x	x	1	0	1	1	1	*dst*			*src2*			*src1*			0	0	0	0	1	0	0
																3			3			3			*opcode*						

Description

Performs a minimum operation on the signed values in *src1* and *src2*. The smaller value is loaded in *dst*.

Condition Codes

CSR:EQ = (dst == 0)

Pseudo Code

```
dst = (src2 < src1) ? src2:src1
```

MINU *Minimum of Two Unsigned Register Values*

Syntax

MINU *src1, src2, dst*

Functional unit = L

Opcode

32 bit

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
x	x	x	x	x	x	x	x	x	x	x	1	1	0	0	1	*dst*			*src2*			*src1*			0	0	0	0	1	0	0
																3			3			3			*opcode*						

Description

Performs a minimum operation on the unsigned values in *src1* and *src2*. The smaller value is loaded in *dst*.

Condition Codes

CSR:EQ = (dst == 0)

Pseudo Code

```
dst = (src2 < src1) ? src2:src1
```

MOD *Signed Modulo*

Syntax

MOD *src1, src2, dst*

Functional unit = M

Opcode

32 bit

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
x	x	x	x	x	x	x	x	x	x	x	0	1	0	1	0	*dst*			*src2*			*src1*			0	0	0	0	1	0	1
																3			3			3			*opcode*						

Description

Performs a signed modulo on the values in *src1* and *src2* and the resulting value is written to *dst*. Sign of the result is the same as *src2* (the dividend). *src1* = 0 (that is, a modulo 0 case), raises the UNDEF interrupt, *dst* is written with 0, and CSR:EQ is set.

Condition Codes

CSR:EQ = (dst == 0)

Pseudo Code

```
dst = src2 % src1
```

Cycles

14 execute cycles. This is a blocking multi-execute cycle instruction.

MODU *Unsigned Modulo*

Syntax

MODU *src1, src2, dst*

Functional unit = M

Opcode

32 bit

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15 13	12 10	9 7	6					0
x	x	x	x	x	x	x	x	x	x	x	0	1	1	1	0	*dst*	*src2*	*src1*	0	0	0	0	1	0	1
																3	3	3				*opcode*			

Description

Performs an unsigned modulo on the values in *src1* and *src2* and the resulting value is written to *dst*. *src1* = 0 (that is, a modulo 0 case), raises the UNDEF interrupt, *dst* is written with 0, and CSR:EQ is set.

Condition Codes

CSR:EQ = (dst == 0)

Pseudo Code

```
dst = src2 % src1
```

Cycles

14 execute cycles. This is a blocking multi-execute cycle instruction.

MPY *Signed Multiplication of Two Register Values*

Syntax

MPY *src1, src2, dst*

Functional unit = M

Opcode

32 bit

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
x	x	x	x	x	x	x	x	x	x	x	0	1	0	0	0	*dst*			*src2*			*src1*			0	0	0	0	1	0	1
																3			3			3			*opcode*						

Description

Performs a multiply on the signed values in *src1* and *src2*. Only the 32 least-significant bits of the result are retained and written to *dst*.

Condition Codes

CSR:EQ = (dst == 0)

Pseudo Code

```
dst = lsb32(src2 × src1)
```

MPYU *Unsigned Multiplication of Two Register Values*

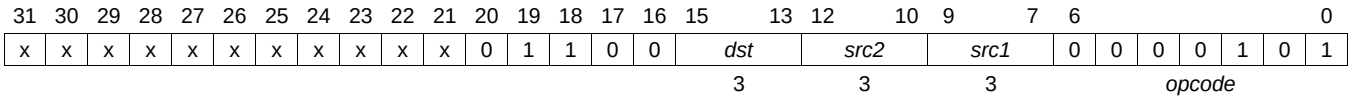
Syntax

MPYU *src1, src2, dst*

Functional unit = M

Opcode

32 bit



31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
x	x	x	x	x	x	x	x	x	x	x	0	1	1	0	0	*dst*			*src2*			*src1*			0	0	0	0	1	0	1
																3			3			3			*opcode*						

Description

Performs a multiply on the unsigned values in *src1* and *src2*. Only the 32 least-significant bits of the result are retained and written to *dst*.

Condition Codes

CSR:EQ = (dst == 0)

Pseudo Code

```
dst = lsb32(src2 × src1)
```

MV *Move Register to Register*

Syntax

MV *src1, dst*

Functional unit = S

Opcode

16 bit

15 13	12	11	10	9 7	6						0
dst	x	x	x	*src1*	1	0	0	1	0	0	1
3				3				*opcode*			

Description

The contents of *src1* are copied to *dst*.

Condition Codes

None

Pseudo Code

```
dst = src1
```

MVC *Move Architectural Register to Control Register*

Syntax

MVC *areg, creg*

Functional unit = S

Opcode

16 bit

15	11	10	9	7	6						0
creg		0	*areg*		1	0	0	1	0	0	0
5			3					*opcode*			

Description

The content of the architectural register (*areg*) is moved to the control register (*creg*).

Condition Codes

None

Pseudo Code

```
creg = areg
```

MVC *Move Control Register to Architectural Register*

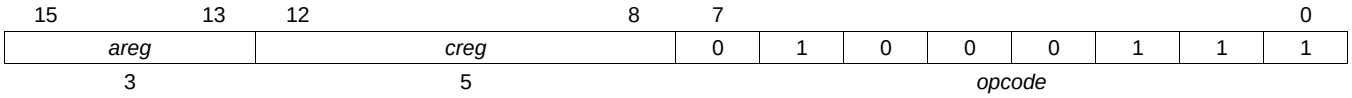
Syntax

MVC *creg, areg*

Functional unit = S

Opcode

16 bit



Description

The content of the control register (*creg*) is moved to the architectural register (*areg*).

Condition Codes

None

Pseudo Code

```
areg = creg
```

MVC *Move 16-Bit Unsigned Constant to Control Register*

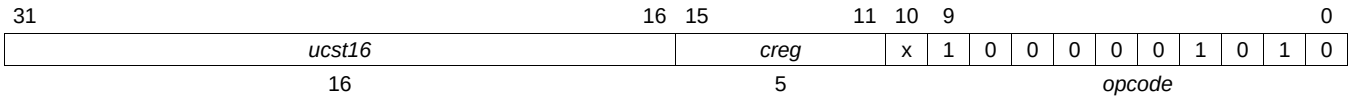
Syntax

MVC *ucst16, creg*

Functional unit = S

Opcode

32 bit



Description

A 16-bit unsigned constant (*ucst16*) is moved to the control register (*creg*).

Condition Codes

None

Pseudo Code

```
creg = {zero extend}ucst16
```

MVCH *Move 16-Bit Unsigned Constant to Upper Bits of Control Register*

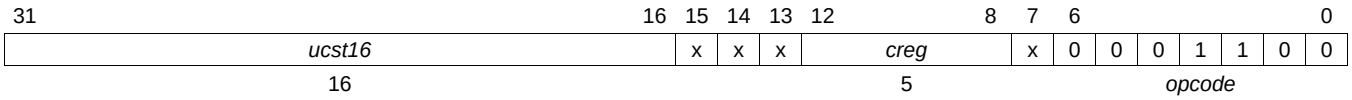
Syntax

MVCH *ucst16, creg*

Functional unit = S

Opcode

32 bit



Description

A 16-bit unsigned constant (*ucst16*) is moved to the upper halfword (bits 31-16) of the control register (*creg*). The lower halfword (bits 15-0) of *creg* is left unchanged.

Condition Codes

None

Pseudo Code

```
creg[31:16] = ucst16
```

MVK *Move 16-Bit Unsigned Constant to Register*

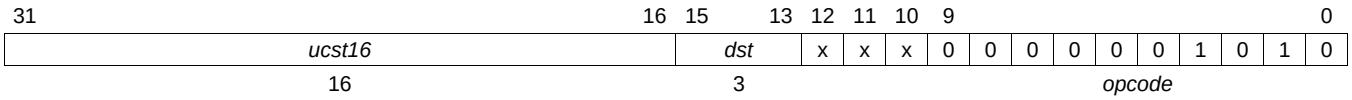
Syntax

MVK *ucst16, dst*

Functional unit = S

Opcode

32 bit



31	16	15	13	12	11	10	9									0
ucst16		*dst*		x	x	x	0	0	0	0	0	0	1	0	1	0
16		3									*opcode*					

Description

A 16-bit unsigned constant (*ucst16*) is moved to *dst*.

Condition Codes

CSR:EQ = (dst == 0)

Pseudo Code

```
dst = {zero extend}ucst16
```

MVKH *Move 16-Bit Unsigned Constant to Upper Bits of Register*

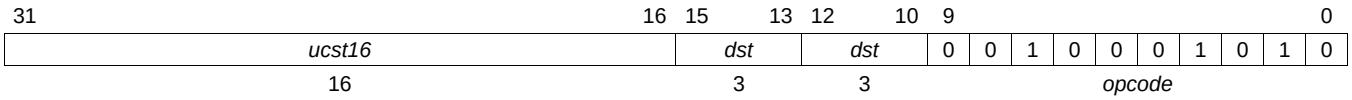
Syntax

MVKH *ucst16, dst*

Functional unit = S

Opcode

32 bit



Description

A 16-bit unsigned constant (*ucst16*) is moved to the upper halfword (bits 31-16) of *dst*. The lower halfword (bits 15-0) of *dst* is left unchanged.

Condition Codes

CSR:EQ = (dst == 0)

Pseudo Code

```
dst[31:16] = ucst16
```

MVKLS *Move 16-Bit Signed Constant to Register*

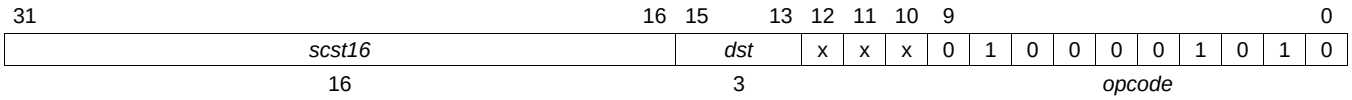
Syntax

MVKLS *scst16, dst*

Functional unit = S

Opcode

32 bit



Description

A 16-bit constant (*scst16*) is sign extended and is moved to *dst*.

Condition Codes

CSR:EQ = (dst == 0)

Pseudo Code

```
dst = {sign extend}scst16
```

MVKS *Move 6-Bit Signed Constant to Register*

Syntax

MVKS *scst6, dst*

Functional unit = S

Opcode

16 bit

15 13	12 7	6						0
dst	*scst6*	1	0	0	0	1	1	0
3	6	*opcode*						

Description

A 6-bit signed constant (*scst6*) is sign extended and is moved to *dst*.

Condition Codes

CSR:EQ = (dst == 0)

Pseudo Code

```
dst = {sign extend}scst6
```

MVS *Move Architectural Register to Shadow Register*

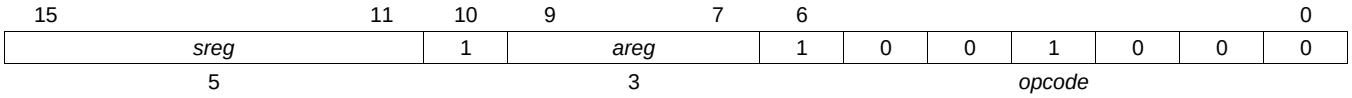
Syntax

MVS *areg, sreg*

Functional unit = S

Opcode

16 bit



Description

The content of the architectural register (*areg*) is moved to the shadow register (*sreg*). This instruction has two exposed delay slots, the data written cannot be read back until the third cycle from the cycle when this instruction enters the EXE phase.

Condition Codes

None

Pseudo Code

```
sreg = areg
```

MVS *Move Shadow Register to Architectural Register*

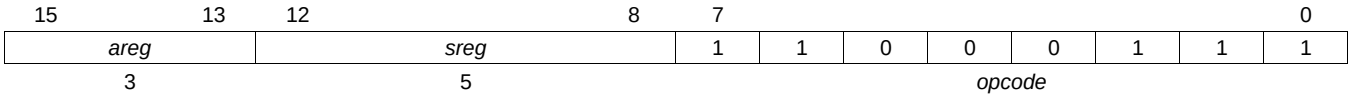
Syntax

MVS *sreg, areg*

Functional unit = S

Opcode

16 bit



Description

The content of the shadow register (*sreg*) is moved to the architectural register (*areg*).

Condition Codes

None

Pseudo Code

```
areg = sreg
```

NEG *Negation*

Syntax

NEG *src2, dst*

Functional unit = L

Opcode

16 bit

15 13	12 10	9									0
dst	*src2*	0	1	0	1	0	0	0	0	0	0
3	3	*opcode*									

Description

src2 is subtracted from zero and the result is stored in *dst*.

Condition Codes

CSR[2]EQ = (dst == 0)

Pseudo Code

```
dst = 0 - src2
```

NOP *No Operation*

Syntax

NOP

Functional unit = None

Opcode

16 bit

15	14	13	12	11	10	9									0
x	x	x	x	x	x	1	1	1	1	1	1	1	1	1	1
										opcode					

Description

No operation.

Condition Codes

None

Pseudo Code

None

NOT — *Bitwise NOT*

Syntax

NOT *src2, dst*

Functional unit = L

Opcode

16 bit

15 13	12 10	9									0
dst	*src2*	0	0	1	1	0	0	0	0	0	0
3	3	*opcode*									

Description

Performs a bitwise inversion of *src2* and stores the result in *dst*.

Condition Codes

CSR[2]EQ = (dst == 0)

Pseudo Code

```
dst = ~src2
```

OR — *Bitwise OR 16-Bit Unsigned Constant with Register*

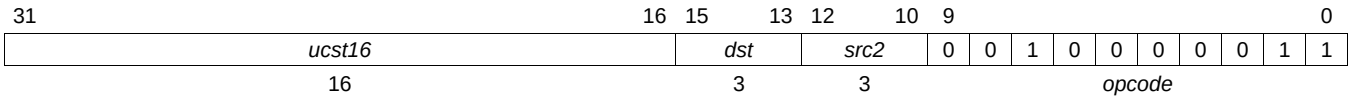
Syntax

OR *ucst16, src2, dst*

Functional unit = L

Opcode

32 bit



Description

Bitwise OR of a zero-extended 16-bit unsigned constant (*ucst16*) with *src2* and store result to *dst*.

Condition Codes

CSR[2]EQ = (dst == 0)

Pseudo Code

```
dst = src2 OR {zero extend}ucst16
```

OR — *Bitwise OR Two Registers*

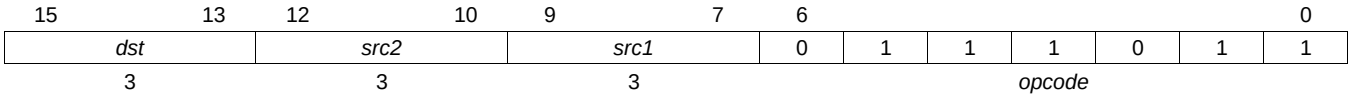
Syntax

OR *src1, src2, dst*

Functional unit = L

Opcode

16 bit



15–13	12–10	9–7	6						0
dst	*src2*	*src1*	0	1	1	1	0	1	1
3	3	3	*opcode*						

Description

Bitwise OR of *src1* with *src2* and store result to *dst*.

Condition Codes

CSR[2]EQ = (dst == 0)

Pseudo Code

```
dst = src2 OR src1
```

RET *Return from Subroutine*

Syntax

RET

Functional unit = D

Opcode

16 bit

15	14	13	12	11	10	9									0
x	x	x	x	x	x	0	1	1	1	1	1	1	1	1	1
										opcode					

Description

Load the link register (LR) into the program counter (PC), increment the stack pointer (SP), then load the contents of the memory location pointed to by SP into LR. The content of LR and the content of the memory location pointed to by SP in this case is treated as an absolute halfword address.

This instruction has one delay slot. See Section 6.2.5.3.1 for restrictions on scheduling an instruction in this delay slot.

Condition Codes

None

Pseudo Code

```
PC = LR; SP += 4; LR = *(SP)
```

REV *Reverse a Bit Field Bounded by Two Register Values*

Syntax

REV *src1, src2, dst*

Functional unit = L

Opcode

32 bit

31	30	29	28	27	26	25	24	23	22	21	20				16	15		13	12		10	9		7	6					0
x	x	x	x	x	x	x	x	x	x	x	1	0	1	0	0	*dst*			*src2*			*src1*			0	0	0	0	1	0	0
																3			3			3						*opcode*			

Description

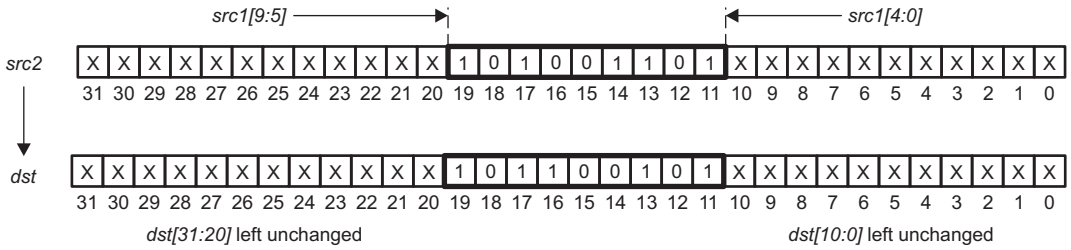
For *src1[9:5]* ≥ *src1[4:0]*, the field in *src2* as specified by *src1[4:0]* to *src1[9:5]* is bit reversed. The reversed bits are merged with the remaining (unchanged) bits in *dst* and are stored in *dst*. *src2* is left unchanged. *src1[4:0]* is the LSB of the field and *src1[9:5]* is the MSB of the field to be bit reversed. In other words, *src1[4:0]* and *src1[9:5]* represent the beginning and ending bits, respectively, of the field to be bit reversed. The LSB location of *src2* is bit 0 and the MSB location of *src2* is bit 31.

Valid values of *src1[9:5]* and *src1[4:0]* are:

- 0 ≤ *src1[9:5]* ≤ 31; 0 ≤ *src1[4:0]* ≤ 31
- *src1[9:5]* ≥ *src1[4:0]*

For *src1[9:5]* < *src1[4:0]*, the result is undefined.

In the following example, *src1[4:0]* is 11 and *src1[9:5]* is 19.



Condition Codes

CSR[2]EQ = (dst == 0)

Pseudo Code

```
dst = src2
pntr = src1[9:5];
for( i = 0; i < 32; i++) {
  if ( i >= src1[4:0] && i <= src1[9:5]){
  dst[i] = dst[pntr]
  --pntr;
} else
  dst[i] = dst[i];
}
```

ROT *Rotate Right*

Syntax

ROT *src1, src2, dst*

Functional unit = L

Opcode

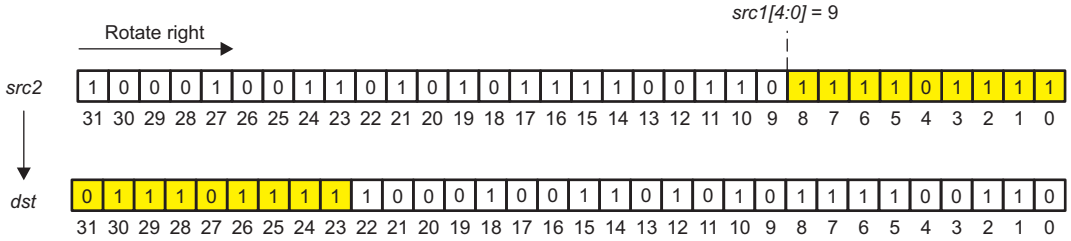
32 bit

31	30	29	28	27	26	25	24	23	22	21	20				16	15		13	12		10	9		7	6						0
x	x	x	x	x	x	x	x	x	x	x	1	0	1	0	1	*dst*			*src2*			*src1*			0	0	0	0	1	0	0
																3			3			3						*opcode*			

Description

src1[4:0] contains the number of bit positions that *src2* is to be rotated right. The new MSB value is defined by the previous LSB value of *src2*. The result is stored in *dst*. A *src1[4:0]* = 0, results in the **MV** *src1, dst* instruction.

In the following example, *src1[4:0]* is 9.



Condition Codes

CSR[2]EQ = (dst == 0)

Pseudo Code

```
dst = src2
for( i = 1; i < src1; i++) {
  dst = { dst[0], dst[32-2: 1] };
}
```

ROTC *Rotate Right Through Carry Bit*

Syntax

ROTC *src1, src2, dst*

Functional unit = L

Opcode

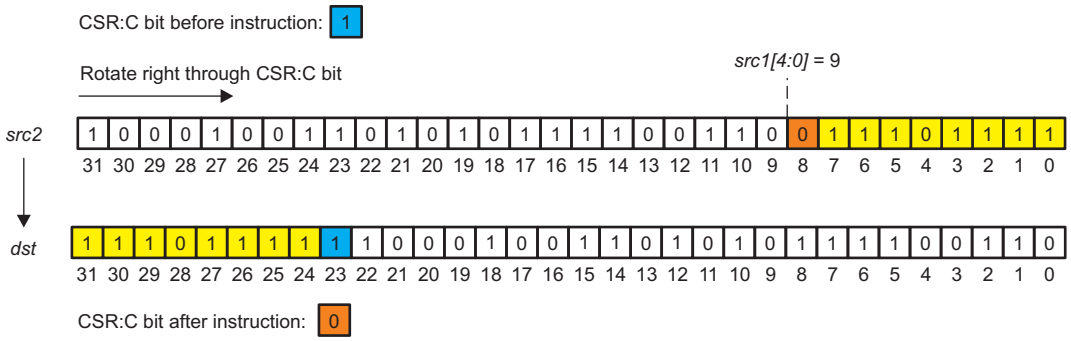
32 bit

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
x	x	x	x	x	x	x	x	x	x	x	1	0	1	1	0	*dst*			*src2*			*src1*			0	0	0	0	1	0	0
																3			3			3			*opcode*						

Description

src1[4:0] contains the number of bit positions that *src2* is to be rotated right through the Carry bit (CSR[5]C). The new MSB value is defined by the current SAT bit in the control status register (CSR). The rotated LSB of *src2* is moved to CSR[5]C. The result is stored in *dst*. A *src1[4:0]* = 0, results in the **MV** *src1, dst* instruction.

In the following example, *src1[4:0]* is 9.



Condition Codes

CSR[2]EQ = (dst == 0)

CSR[5]C: see the Pseudo Code

Pseudo Code

```
tmp = src2
for( i = 1; i < src1; i++) {
  tmp = { CSR[C], tmp[32-2: 1] };
  CSR[C] = tmp[0] ;
}
dst = tmp;
```

SADD *Signed Addition of Two Register Values with Saturation*

Syntax

SADD *src1, src2, dst*

Functional unit = D

Opcode

16 bit

15 13	12 10	9 7	6						0
dst	*src2*	*src1*	0	1	1	1	0	0	0
3	3	3	*opcode*						

Description

Signed addition of *src1* with *src2* with result saturation and stored to *dst*. If a saturate occurs, the SAT bit in the control status register (CSR) is set after *dst* is written.

Condition Codes

CSR[2]EQ = (dst == 0)

CSR[5]SAT = satP OR satN

Pseudo Code

```
tmp = src2 + src1
satP = tmp > 2^31 -     1
satN = tmp <     -2^31
dst = satP ? (2^31 - 1)     : (satN ? -2^31 : tmp)
```

SATN *Saturation to Signed/Unsigned N-Bit Value with Sign/Zero Extend*

Syntax

SATN *ucst3, src2, dst*

Functional unit = L

Opcode

32 bit

31	30	29	28	27	26	25	24	23	22	21	20	19	18 16	15 13	12 10	9								0
x	x	x	x	x	x	x	x	x	x	x	x	x	*ucst3[2:0]*	*dst*	*src2*	1	1	1	0	0	0	0	1	1
													3	3	3					*opcode*				

Description

Saturation to signed/unsigned N-bit value and sign/zero extend to 32 bits. *src2* is saturated based on *ucst3*:

- *ucst3[2]* = 1 indicates signed saturation
- *ucst3[2]* = 0 indicates unsigned saturation
- *ucst3[1:0]* = 0 indicates saturation to 8 bit value
- *ucst3[1:0]* = 1 indicates saturation to 16 bit value
- *ucst3[1:0]* = other values (2 or 3) are reserved

src2 is considered a signed value that needs to be saturated to a signed or unsigned range. The result is undefined, if reserved values in *ucst3* are used.

Condition Codes

CSR[2]EQ = (dst == 0)

Pseudo Code

dst = (src2 <= $-2^{(N-1)}$) ? $-2^{(N-1)}$: ((src2 >= $2^{(N-1)}$ - 1) ? $2^{(N-1)}$ - 1 : src2);

SET — *Set Bit Field Bounded by Two Immediate Values*

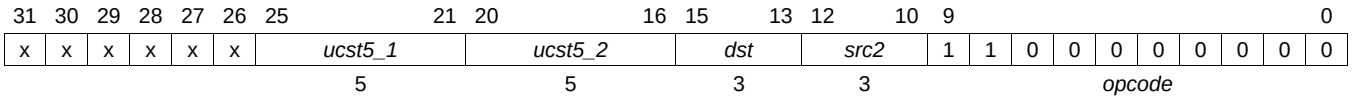
Syntax

SET *ucst5_1, ucst5_2, src2, dst*

Functional unit = L

Opcode

32 bit



Description

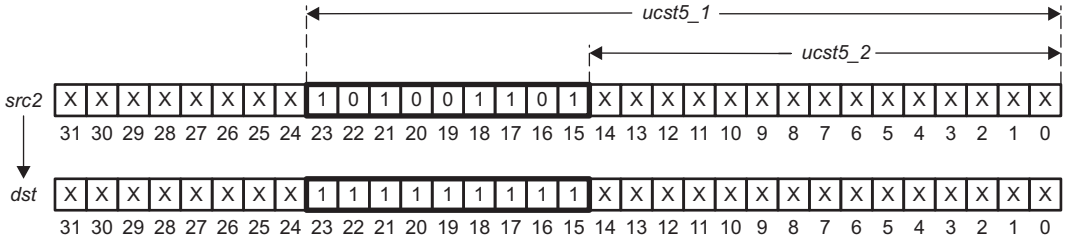
For *ucst5_1* ≥ *ucst5_2*, the field in *src2* as specified by *ucst5_2* to *ucst5_1* is set to all 1s and the resulting value of *src2* is written to *dst*. *src2* is left unchanged. *ucst5_2* is the LSB of the field and *ucst5_1* is the MSB of the field. In other words, *ucst5_2* and *ucst5_1* represent the beginning and ending bits, respectively, of the field to be set to all 1s. The LSB location of *src2* is bit 0 and the MSB location of *src2* is bit 31.

Valid values of *ucst5_1* and *ucst5_2* are:

- 0 ≤ *ucst5_1* ≤ 31; 0 ≤ *ucst5_2* ≤ 31
- *ucst5_1* ≥ *ucst5_2*

For *ucst5_1* < *ucst5_2*, the result is undefined.

In the following example, *ucst5_2* is 15 and *ucst5_1* is 23.



Condition Codes

CSR[2]EQ = (dst == 0)

Pseudo Code

```
dst = src2
for( i = 0; i < 32; i++) {
  if ( i >= ucst5_2 && i <= ucst5_2) dst[i] = 1;
};
```

SET *Set Bit Field Bounded by Two Register Values*

Syntax

SET *src1, src2, dst*

Functional unit = L

Opcode

32 bit

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	13	12	10	9	7	6						0
x	x	x	x	x	x	x	x	x	x	x	1	0	0	0	0	*dst*		*src2*		*src1*		0	0	0	0	1	0	0
																3		3		3					*opcode*			

Description

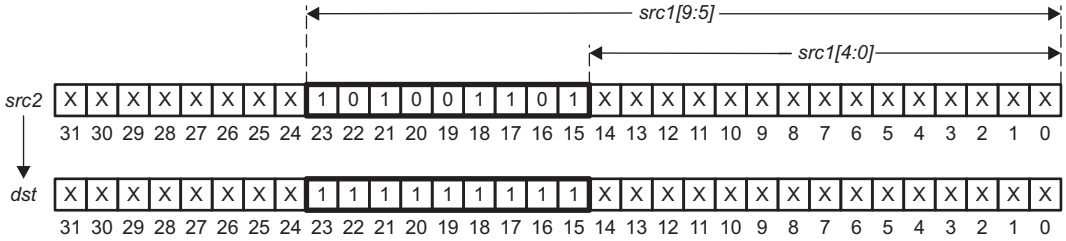
For *src1[9:5]* ≥ *src1[4:0*, the field in *src2* as specified by *src1[4:0]* to *src1[9:5]* is set to all 1s and the resulting value of *src2* is written to *dst*. *src2* is left unchanged. *src1[4:0]* is the LSB of the field and *src1[9:5]* is the MSB of the field. In other words, *src1[4:0]* and *src1[9:5]* represent the beginning and ending bits, respectively, of the field to be set to all 1s. The LSB location of *src2* is bit 0 and the MSB location of *src2* is bit 31.

Valid values of *src1[9:5* and *src1[4:0]* are:

- 0 ≤ *src1[9:5]* ≤ 31; 0 ≤ *src1[4:0]* ≤ 31
- *src1[9:5]* ≥ *src1[4:0]*

For *src1[9:5]* < *src1[4:0]*, the result is undefined.

In the following example, *src1[4:0]* is 15 and *src1[9:5]* is 23.



Condition Codes

CSR[2]EQ = (dst == 0)

Pseudo Code

```
dst = src2
for( i = 0; i < 32; i++) {
  if ( i >= src1[4:0] && i <= src1[9:5]) dst[i] = 1;
};
```

SHL *Logical Shift Left by 5-Bit Unsigned Constant*

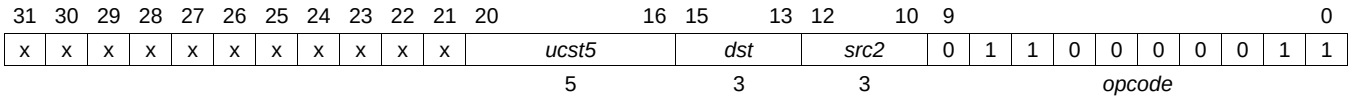
Syntax

SHL *ucst5, src2, dst*

Functional unit = L

Opcode

32 bit



31	30	29	28	27	26	25	24	23	22	21	20 – 16	15 – 13	12 – 10	9	8	7	6	5	4	3	2	1	0
x	x	x	x	x	x	x	x	x	x	x	*ucst5*	*dst*	*src2*	0	1	1	0	0	0	0	0	1	1
											5	3	3	*opcode*									

Description

src2 is left shifted by a 5-bit unsigned constant (*ucst5*) and the result is stored to *dst*. The shift in value is 0.

Condition Codes

CSR[2] EQ = (dst == 0)

Pseudo Code

```
dst = src2 << ucst5
```

SHL *Logical Shift Left by Register Value*

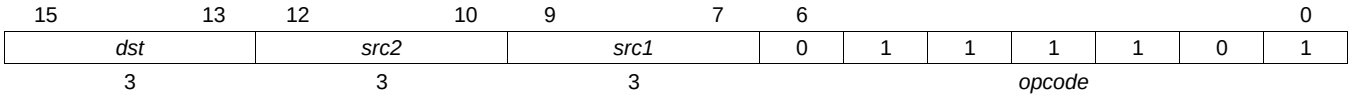
Syntax

SHL *src1, src2, dst*

Functional unit = L

Opcode

16 bit



15–13	12–10	9–7	6						0
dst	*src2*	*src1*	0	1	1	1	1	0	1
3	3	3	*opcode*						

Description

src2 is left shifted by an unsigned value (*src1*) and the result is stored to *dst*. The shift in value is 0. If the *src1* value is greater than 31, *dst* is cleared to 0.

Condition Codes

CSR[2] EQ = (dst == 0)

Pseudo Code

```
dst = (src1 > 31) ? 0: (src2 << src1)
```

SHRA *Arithmetic Shift Right by 5-Bit Unsigned Constant*

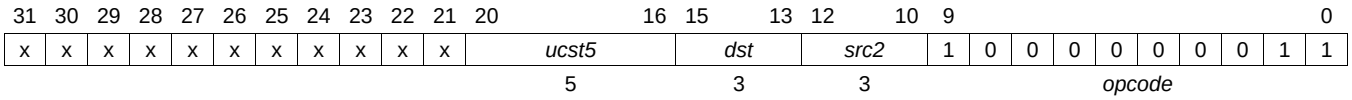
Syntax

SHRA *ucst5, src2, dst*

Functional unit = L

Opcode

32 bit



31	30	29	28	27	26	25	24	23	22	21	20 – 16	15 – 13	12 – 10	9	8	7	6	5	4	3	2	1	0
x	x	x	x	x	x	x	x	x	x	x	*ucst5*	*dst*	*src2*	1	0	0	0	0	0	0	0	1	1
											5	3	3	*opcode*									

Description

src2 is right shifted by a 5-bit unsigned constant (*ucst5*) and the result is stored to *dst*. The shift in value is the sign bit (bit 31) of *src2*.

Using **DIV** (integer division) and **SHRA** (arithmetic right shift) does not produce the same result for negative numbers. The quotient of **DIV** is rounded towards zero, whereas the quotient of **SHRA** is rounded towards negative infinity. For example, using the **DIV** instruction: -9/4 = -2, whereas using the **SHRA** instruction: -9/4 = -3.

Condition Codes

CSR[2] EQ = (dst == 0)

Pseudo Code

```
dst = {sign extend}{src2 >> ucst5}
```

SHRA *Arithmetic Shift Right by Register Value*

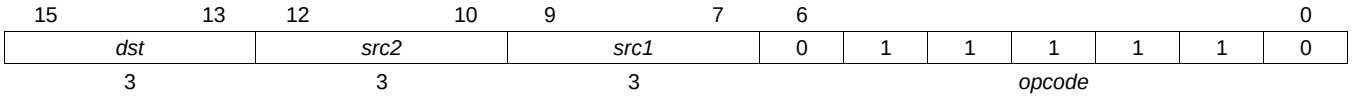
Syntax

SHRA *src1, src2, dst*

Functional unit = L

Opcode

16 bit



15–13	12–10	9–7	6						0
dst	*src2*	*src1*	0	1	1	1	1	1	0
3	3	3	*opcode*						

Description

src2 is right shifted by an unsigned value (*src1*) and the result is stored to *dst*. The shift in value is the sign bit (bit 31) of *src2* . If the *src1* value is greater than 31, *dst* is -1.

Using **DIV** (integer division) and **SHRA** (arithmetic right shift) does not produce the same result for negative numbers. The quotient of **DIV** is rounded towards zero, whereas the quotient of **SHRA** is rounded towards negative infinity. For example, using the **DIV** instruction: -9/4 = -2, whereas using the **SHRA** instruction: -9/4 = -3.

Condition Codes

CSR[2] EQ = (dst == 0)

Pseudo Code

```
dst = (src1 > 31) ? FFFF FFFFh: {sign extend}{src2 >> src1}
```

SHRU *Logical Shift Right by 5-Bit Unsigned Constant*

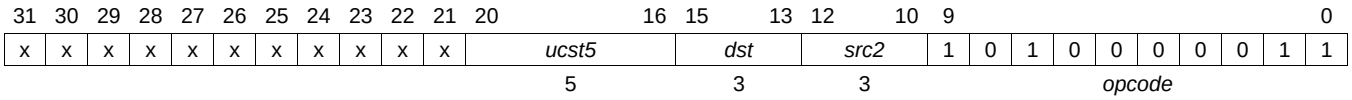
Syntax

SHRU *ucst5, src2, dst*

Functional unit = L

Opcode

32 bit



31	30	29	28	27	26	25	24	23	22	21	20 – 16	15 – 13	12 – 10	9	8	7	6	5	4	3	2	1	0
x	x	x	x	x	x	x	x	x	x	x	*ucst5*	*dst*	*src2*	1	0	1	0	0	0	0	0	1	1
											5	3	3	*opcode*									

Description

src2 is right shifted by a 5-bit unsigned constant (*ucst5*) and the zero-extended result is stored to *dst*.

Condition Codes

CSR[2] EQ = (dst == 0)

Pseudo Code

```
dst = {zero extend}{src2 >> ucst5}
```

SHRU *Logical Shift Right by Register Value*

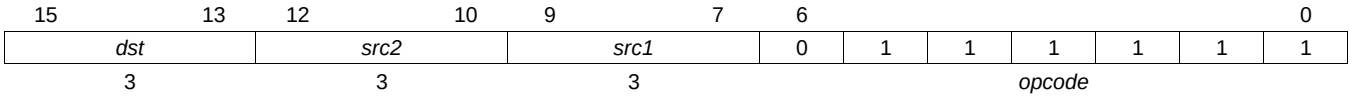
Syntax

SHRU *src1, src2, dst*

Functional unit = L

Opcode

16 bit



Description

src2 is right shifted by an unsigned value (*src1*) and the result is stored to *dst*. The shift in value is 0. If the *src1* value is greater than 31, *dst* is cleared to 0.

Condition Codes

CSR[2] EQ = (dst == 0)

Pseudo Code

```
dst = (src1 > 31) ? 0: {zero extend}{src2 >> src1}
```

SLA *Set Loop Address*

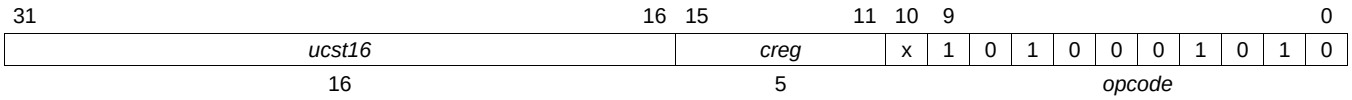
Syntax

SLA *ucst16, creg*

Functional unit = S

Opcode

32 bit



Description

Sets up loop bound address using a program counter (PC)-relative immediate positive offset. A zero-extended 16-bit unsigned immediate value (*ucst16*) is added to the PC of the instruction and is written to *creg*. Note that the offset is treated as a halfword offset. The only allowed *creg* registers for this instruction are: LSA0, LEA0, LSA1, and LEA1.

Condition Codes

None

Pseudo Code

```
creg = PC + {zero extend}ucst16
```

SSUB *Subtraction of Two Register Values with Saturation*

Syntax

SSUB *src1, src2, dst*

Functional unit = D

Opcode

16 bit

15 13	12 10	9 7	6						0
dst	*src2*	*src1*	0	1	1	1	0	0	1
3	3	3	*opcode*						

Description

src1 is subtracted from *src2* with result saturation and stored to *dst*. If a saturate occurs, the SAT bit in the control status register (CSR) is set after *dst* is written.

Condition Codes

CSR[2] EQ = (dst == 0)

CSR[6] SAT = satP OR satN

Pseudo Code

```
tmp = src2 - src1
satP = tmp > 2^31 -     1
satN = tmp <     -2^31
dst = satP ? (2^31 - 1)     : (satN ? -2^31 : tmp)
```

STB *Store Byte to Memory with a 3-Bit Unsigned Constant Offset*

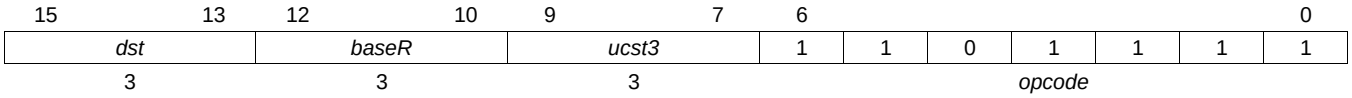
Syntax

STB *dst, *+baseR[ucst3]*

Functional unit = D

Opcode

16 bit



15–13	12–10	9–7	6						0
dst	*baseR*	*ucst3*	1	1	0	1	1	1	1
3	3	3	*opcode*						

Description

The 8 LSBs (byte) of *dst* are stored to memory (effective address). The memory address is formed from a base address register (*baseR*) and an offset (number of bytes) that is a 3-bit unsigned constant (*ucst3*). If an offset is not given, the assembler assigns an offset of zero. Brackets, [ ], must surround the specified offset, if using the optional offset parameter.

The square brackets, [ ], indicate that the *ucst3* is scaled by a left-shift of 0 bits. After scaling, *ucst3* is added to *baseR*. The result of the calculation is the effective address in memory that contains the content from *dst*.

Condition Codes

None

Pseudo Code

```
*(baseR[ucst3]) = *dst
```

STB *Store Byte to Memory with a 16-Bit Unsigned Constant Offset*

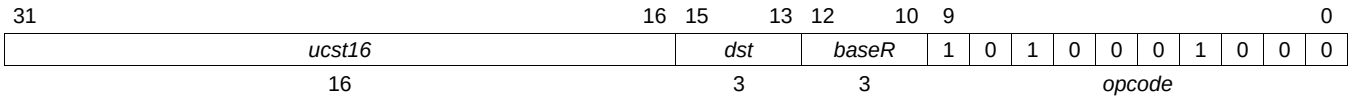
Syntax

STB *dst, *+baseR[ucst16]*

Functional unit = D

Opcode

32 bit



Description

The 8 LSBs (byte) of *dst* are stored to memory (effective address). The memory address is formed from a base address register (*baseR*) and an offset (number of bytes) that is a 16-bit unsigned constant (*ucst16*). If an offset is not given, the assembler assigns an offset of zero. Brackets, [ ], must surround the specified offset, if using the optional offset parameter.

The square brackets, [ ], indicate that the *ucst16* is scaled by a left-shift of 0 bits. After scaling, *ucst16* is added to *baseR*. The result of the calculation is the effective address in memory that contains the content from *dst*.

Condition Codes

None

Pseudo Code

```
*(baseR[ucst16]) = *dst
```

STB *Store Byte to Memory with a Register Offset*

Syntax

STB *dst, *+baseR[src1]*

Functional unit = D

Opcode

16 bit

15 13	12 10	9 7	6						0
dst	*baseR*	*src1*	1	1	0	1	1	0	0
3	3	3	*opcode*						

Description

The 8 LSBs (byte) of *dst* are stored to memory (effective address). The memory address is formed from a base address register (*baseR*) and an offset (number of bytes) that is a register (*src1*). If an offset is not given, the assembler assigns an offset of zero. Brackets, [ ], must surround the specified offset, if using the optional offset parameter.

The square brackets, [ ], indicate that the *src1* is scaled by a left-shift of 0 bits. After scaling, *src1* is added to *baseR*. The result of the calculation is the effective address in memory that contains the content from *dst*.

Condition Codes

None

Pseudo Code

```
*(baseR[src1]) = *dst
```

STB *Store Byte to Memory, Postincrement with a 3-Bit Unsigned Constant Offset*

Syntax

STB *dst, *baseR++[ucst3]*

Functional unit = D

Opcode

16 bit

15 13	12 10	9 7	6						0
dst	*baseR*	*ucst3*	1	1	1	0	1	0	1
3	3	3	*opcode*						

Description

The 8 LSBs (byte) of *dst* are stored to memory (effective address). The memory address is the value of the base address register (*baseR*). After accessing, an offset (number of bytes) that is a 3-bit unsigned constant (*ucst3*) is added to *baseR* at the end of the EXE phase. Brackets, [ ], must surround the specified offset, if using the optional offset parameter.

Condition Codes

None

Pseudo Code

```
*(baseR++[ucst3]) = *dst
```

STB *Store Byte to Memory, Postincrement with a Register Offset*

Syntax

STB *dst, *baseR++[src1]*

Functional unit = D

Opcode

16 bit

15 13	12 10	9 7	6						0
dst	*baseR*	*src1*	1	1	1	0	0	1	0
3	3	3				*opcode*			

Description

The 8 LSBs (byte) of *dst* are stored to memory (effective address). The memory address is the value of the base address register (*baseR*). After accessing, an offset (number of bytes) that is a register (*src1*) is added to *baseR* at the end of the EXE phase. Brackets, [ ], must surround the specified offset, if using the optional offset parameter.

Condition Codes

None

Pseudo Code

```
*(baseR++[src1]) = *dst
```

STB *Store Byte to Memory with a SP-Relative 6-Bit Unsigned Constant Offset*

Syntax

STB *dst, *+***SP***[ucst6]*

Functional unit = D

Opcode

16 bit

15 13	12 7	6						0
dst	*ucst6*	1	1	1	1	0	0	0
3	6	*opcode*						

Description

The 8 LSBs (byte) of *dst* are stored to memory (effective address). The memory address is formed from the stack pointer (SP) and an offset (number of bytes) that is a 6-bit unsigned constant (*ucst6*). If an offset is not given, the assembler assigns an offset of zero. Brackets, [ ], must surround the specified offset, if using the optional offset parameter.

The square brackets, [ ], indicate that the *ucst6* is scaled by a left-shift of 0 bits. After scaling, *ucst6* is added to SP. The result of the calculation is the effective address in memory that contains the content from *dst*.

Condition Codes

None

Pseudo Code

```
*(SP[ucst6]) = *dst
```

STB *Store Byte to Memory with a SP-Relative 19-Bit Unsigned Constant Offset*

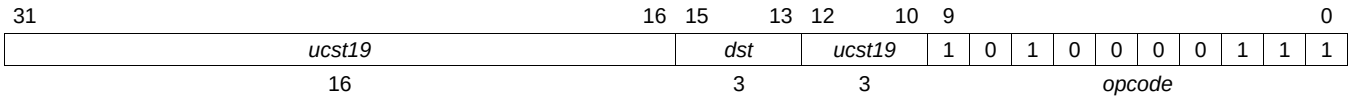
Syntax

STB *dst, **+**SP***[***ucst19***]*

Functional unit = D

Opcode

32 bit



Description

The 8 LSBs (byte) of *dst* are stored to memory (effective address). The memory address is formed from the stack pointer (SP) and an offset (number of bytes) that is a 19-bit unsigned constant (*ucst19*). If an offset is not given, the assembler assigns an offset of zero. Brackets, [ ], must surround the specified offset, if using the optional offset parameter.

The square brackets, [ ], indicate that the *ucst19* is scaled by a left-shift of 0 bits. After scaling, *ucst19* is added to SP. The result of the calculation is the effective address in memory that contains the content from *dst*.

Condition Codes

None

Pseudo Code

```
*(SP[ucst19]) = *dst
```

STB *Store Byte to Memory with a GDP-Relative 19-Bit Unsigned Constant Offset*

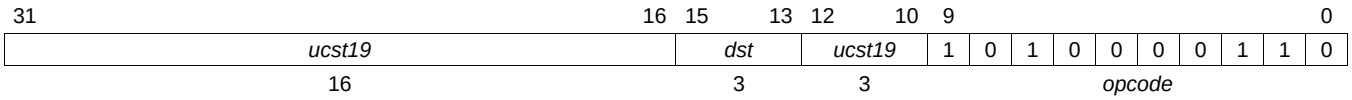
Syntax

STB *dst, **+**GDP***[ucst19]***

Functional unit = D

Opcode

32 bit



Description

The 8 LSBs (byte) of *dst* are stored to memory (effective address). The memory address is formed from the global data pointer register (GDP) and an offset (number of bytes) that is a 19-bit unsigned constant (*ucst19*). If an offset is not given, the assembler assigns an offset of zero. Brackets, [ ], must surround the specified offset, if using the optional offset parameter.

The square brackets, [ ], indicate that the *ucst19* is scaled by a left-shift of 0 bits. After scaling, *ucst19* is added to GDP. The result of the calculation is the effective address in memory that contains the content from *dst*.

Condition Codes

None

Pseudo Code

```
*(GDP[ucst19]) = *dst
```

STH *Store Halfword to Memory with a 3-Bit Unsigned Constant Offset*

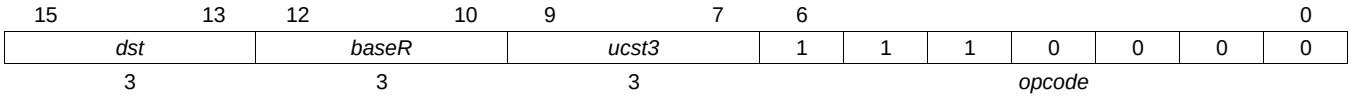
Syntax

STH *dst, *+baseR[ucst3]*

Functional unit = D

Opcode

16 bit



15–13	12–10	9–7	6	5	4	3	2	1	0
dst	*baseR*	*ucst3*	1	1	1	0	0	0	0
3	3	3	*opcode*						

Description

The 16 LSBs (halfword) of *dst* are stored to memory (effective address). The memory address is formed from a base address register (*baseR*) and an offset (number of halfwords) that is a 3-bit unsigned constant (*ucst3*). If an offset is not given, the assembler assigns an offset of zero. Brackets, [ ], must surround the specified offset, if using the optional offset parameter.

The square brackets, [ ], indicate that the *ucst3* is scaled by a left-shift of 1 bit. After scaling, *ucst3* is added to *baseR*. The result of the calculation is the effective address in memory that contains the content from *dst*.

Halfword addresses must be aligned on halfword (LSB is 0) boundaries.

Condition Codes

None

Pseudo Code

```
*(baseR[ucst3]) = *dst
```

STH *Store Halfword to Memory with a 16-Bit Unsigned Constant Offset*

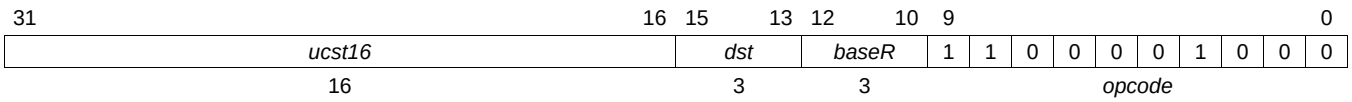
Syntax

STH *dst, *+baseR[ucst16]*

Functional unit = D

Opcode

32 bit



Description

The 16 LSBs (halfword) of *dst* are stored to memory (effective address). The memory address is formed from a base address register (*baseR*) and an offset (number of halfwords) that is a 16-bit unsigned constant (*ucst16*). If an offset is not given, the assembler assigns an offset of zero. Brackets, [ ], must surround the specified offset, if using the optional offset parameter.

The square brackets, [ ], indicate that the *ucst16* is scaled by a left-shift of 1 bit. After scaling, *ucst16* is added to *baseR*. The result of the calculation is the effective address in memory that contains the content from *dst*.

Halfword addresses must be aligned on halfword (LSB is 0) boundaries.

Condition Codes

None

Pseudo Code

```
*(baseR[ucst16]) = *dst
```

STH *Store Halfword to Memory with a Register Offset*

Syntax

STH *dst, *+baseR[src1]*

Functional unit = D

Opcode

16 bit

15 13	12 10	9 7	6						0
dst	*baseR*	*src1*	1	1	0	1	1	0	1
3	3	3	*opcode*						

Description

The 16 LSBs (halfword) of *dst* are stored to memory (effective address). The memory address is formed from a base address register (*baseR*) and an offset (number of halfwords) that is a register (*src1*). If an offset is not given, the assembler assigns an offset of zero. Brackets, [ ], must surround the specified offset, if using the optional offset parameter.

The square brackets, [ ], indicate that the *src1* is scaled by a left-shift of 1 bit. After scaling, *src1* is added to *baseR*. The result of the calculation is the effective address in memory that contains the content from *dst*.

Halfword addresses must be aligned on halfword (LSB is 0) boundaries.

Condition Codes

None

Pseudo Code

```
*(baseR[src1]) = *dst
```

STH *Store Halfword to Memory, Postincrement with a 3-Bit Unsigned Constant Offset*

Syntax

STH *dst, *baseR++[ucst3]*

Functional unit = D

Opcode

16 bit

15 13	12 10	9 7	6						0
dst	*baseR*	*ucst3*	1	1	1	0	1	1	0
3	3	3	*opcode*						

Description

The 16 LSBs (halfword) of *dst* are stored to memory (effective address). The memory address is the value of the base address register (*baseR*). After accessing, an offset (number of halfwords) that is a 3-bit unsigned constant (*ucst3*) is added to *baseR* at the end of the EXE phase. You must type the brackets, [ ], around the specified offset, if you use the optional offset parameter.

Halfword addresses must be aligned on halfword (LSB is 0) boundaries.

Condition Codes

None

Pseudo Code

```
*(baseR++[ucst3]) = *dst
```

STH *Store Halfword to Memory, Postincrement with a Register Offset*

Syntax

STH *dst, *baseR++[src1]*

Functional unit = D

Opcode

16 bit

15 13	12 10	9 7	6						0
dst	*baseR*	*src1*	1	1	1	0	0	1	1
3	3	3				*opcode*			

Description

The 16 LSBs (halfword) of *dst* are stored to memory (effective address). The memory address is the value of the base address register (*baseR*). After accessing, an offset (number of halfwords) that is a register (*src1*) is added to *baseR* at the end of the EXE phase. You must type the brackets, [ ], around the specified offset, if you use the optional offset parameter.

Halfword addresses must be aligned on halfword (LSB is 0) boundaries.

Condition Codes

None

Pseudo Code

```
*(baseR++[src1]) = *dst
```

STH *Store Halfword to Memory with a SP-Relative 6-Bit Unsigned Constant Offset*

Syntax

STH *dst, *+***SP***[ucst6]*

Functional unit = D

Opcode

16 bit

15 13	12 7	6						0
dst	*ucst6*	1	1	1	1	0	0	1
3	6	*opcode*						

Description

The 16 LSBs (byte) of *dst* are stored to memory (effective address). The memory address is formed from the stack pointer (SP) and an offset (number of halfwords) that is a 6-bit unsigned constant (*ucst6*). If an offset is not given, the assembler assigns an offset of zero. You must type the brackets, [ ], around the specified offset, if you use the optional offset parameter.

The square brackets, [ ], indicate that the *ucst6* is scaled by a left-shift of 1 bit. After scaling, *ucst6* is added to SP. The result of the calculation is the effective address in memory that contains the content from *dst*.

Halfword addresses must be aligned on halfword (LSB is 0) boundaries.

Condition Codes

None

Pseudo Code

```
*(SP[ucst6]) = *dst
```

STH *Store Halfword to Memory with a SP-Relative 19-Bit Unsigned Constant Offset*

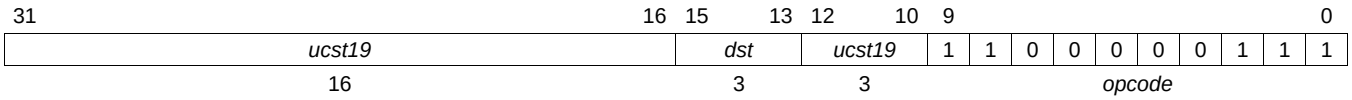
Syntax

STH *dst, *+***SP***[ucst19]*

Functional unit = D

Opcode

32 bit



31 – 16	15 – 13	12 – 10	9									0
ucst19	*dst*	*ucst19*	1	1	0	0	0	0	0	1	1	1
16	3	3	*opcode*									

Description

The 16 LSBs (byte) of *dst* are stored to memory (effective address). The memory address is formed from the stack pointer (SP) and an offset (number of halfwords) that is a 19-bit unsigned constant (*ucst19*). If an offset is not given, the assembler assigns an offset of zero. You must type the brackets, [ ], around the specified offset, if you use the optional offset parameter.

The square brackets, [ ], indicate that the *ucst19* is scaled by a left-shift of 1 bit. After scaling, *ucst19* is added to SP. The result of the calculation is the effective address in memory that contains the content from *dst*.

Halfword addresses must be aligned on halfword (LSB is 0) boundaries.

Condition Codes

None

Pseudo Code

```
*(SP[ucst19]) = *dst
```

STH *Store Halfword to Memory with a GDP-Relative 19-Bit Unsigned Constant Offset*

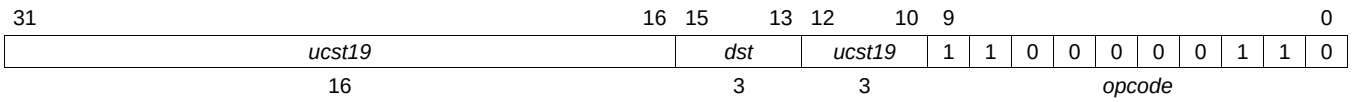
Syntax

STH *dst, **+**GDP***[ucst19]***

Functional unit = D

Opcode

32 bit



Description

The 16 LSBs (byte) of *dst* are stored to memory (effective address). The memory address is formed from the global data pointer register (GDP) and an offset (number of halfwords) that is a 19-bit unsigned constant (*ucst19*). If an offset is not given, the assembler assigns an offset of zero. You must type the brackets, [ ], around the specified offset, if you use the optional offset parameter.

The square brackets, [ ], indicate that the *ucst19* is scaled by a left-shift of 1 bit. After scaling, *ucst19* is added to GDP. The result of the calculation is the effective address in memory that contains the content from *dst*.

Halfword addresses must be aligned on halfword (LSB is 0) boundaries.

Condition Codes

None

Pseudo Code

```
*(GDP[ucst19]) = *dst
```

STW *Store Word to Memory with a 3-Bit Unsigned Constant Offset*

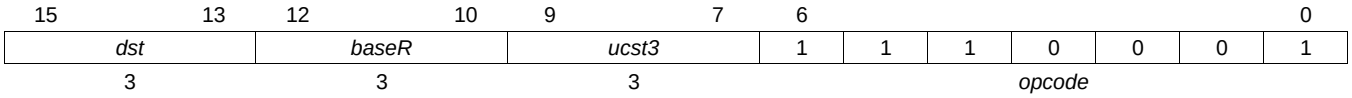
Syntax

STW *dst, *+baseR[ucst3]*

Functional unit = D

Opcode

16 bit



15–13	12–10	9–7	6			5–0			
dst	*baseR*	*ucst3*	1	1	1	0	0	0	1
3	3	3	*opcode*						

Description

The entire content (word) of *dst* is stored to memory (effective address). The memory address is formed from a base address register (*baseR*) and an offset (number of words) that is a 3-bit unsigned constant (*ucst3*). If an offset is not given, the assembler assigns an offset of zero. You must type the brackets, [ ], around the specified offset, if you use the optional offset parameter.

The square brackets, [ ], indicate that the *ucst3* is scaled by a left-shift of 2 bits. After scaling, *ucst3* is added to *baseR*. The result of the calculation is the effective address in memory that contains the content from *dst*.

Word addresses must be aligned on word (two LSBs are 0) boundaries.

Condition Codes

None

Pseudo Code

```
*(baseR[ucst3]) = *dst
```

STW *Store Word to Memory with a 16-Bit Unsigned Constant Offset*

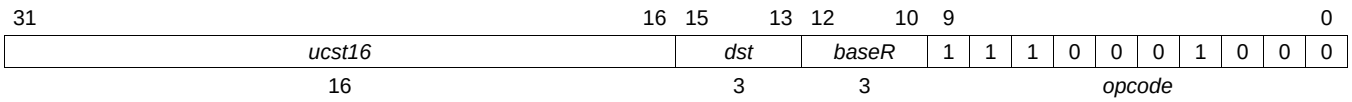
Syntax

STW *dst, *+baseR[ucst16]*

Functional unit = D

Opcode

32 bit



Description

The entire content (word) of *dst* is stored to memory (effective address). The memory address is formed from a base address register (*baseR*) and an offset (number of words) that is a 16-bit unsigned constant (*ucst16*). If an offset is not given, the assembler assigns an offset of zero. You must type the brackets, [ ], around the specified offset, if you use the optional offset parameter.

The square brackets, [ ], indicate that the *ucst16* is scaled by a left-shift of 2 bits. After scaling, *ucst16* is added to *baseR*. The result of the calculation is the effective address in memory that contains the content from *dst*.

Word addresses must be aligned on word (two LSBs are 0) boundaries.

Condition Codes

None

Pseudo Code

```
*(baseR[ucst16]) = *dst
```

STW *Store Word to Memory with a Register Offset*

Syntax

STW *dst, *+baseR[src1]*

Functional unit = D

Opcode

16 bit

15 13	12 10	9 7	6						0
dst	*baseR*	*src1*	1	1	0	1	1	1	0
3	3	3	*opcode*						

Description

The entire content (word) of *dst* is stored to memory (effective address). The memory address is formed from a base address register (*baseR*) and an offset (number of words) that is a register (*src1*). If an offset is not given, the assembler assigns an offset of zero. You must type the brackets, [ ], around the specified offset, if you use the optional offset parameter.

The square brackets, [ ], indicate that the *src1* is scaled by a left-shift of 2 bits. After scaling, *src1* is added to *baseR*. The result of the calculation is the effective address in memory that contains the content from *dst*.

Word addresses must be aligned on word (two LSBs are 0) boundaries.

Condition Codes

None

Pseudo Code

```
*(baseR[src1]) = *dst
```

STW *Store Word to Memory, Postincrement with a 3-Bit Unsigned Constant Offset*

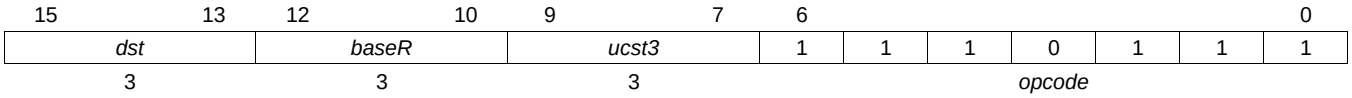
Syntax

STW *dst, *baseR++[ucst3]*

Functional unit = D

Opcode

16 bit



15–13	12–10	9–7	6						0
dst	*baseR*	*ucst3*	1	1	1	0	1	1	1
3	3	3	*opcode*						

Description

The entire content (word) of *dst* is stored to memory (effective address). The memory address is the value of the base address register (*baseR*). After accessing, an offset (number of halfwords) that is a 3-bit unsigned constant (*ucst3*) is added to *baseR* at the end of the EXE phase. You must type the brackets, [ ], around the specified offset, if you use the optional offset parameter.

Word addresses must be aligned on word (two LSBs are 0) boundaries.

Condition Codes

None

Pseudo Code

```
*(baseR++[ucst3]) = *dst
```

STW *Store Word to Memory, Postincrement with a Register Offset*

Syntax

STW *dst, *baseR++[src1]*

Functional unit = D

Opcode

16 bit

15 13	12 10	9 7	6						0
dst	*baseR*	*src1*	1	1	1	0	1	0	0
3	3	3	*opcode*						

Description

The entire content (word) of *dst* is stored to memory (effective address). The memory address is the value of the base address register (*baseR*). After accessing, an offset (number of halfwords) that is a register (*src1*) is added to *baseR* at the end of the EXE phase. You must type the brackets, [ ], around the specified offset, if you use the optional offset parameter.

Word addresses must be aligned on word (two LSBs are 0) boundaries.

Condition Codes

None

Pseudo Code

```
*(baseR++[src1]) = *dst
```

STW *Store Word to Memory with a SP-Relative 6-Bit Unsigned Constant Offset*

Syntax

STW *dst, *+***SP***[ucst6]*

Functional unit = D

Opcode

16 bit

15 13	12 7	6						0
dst	*ucst6*	1	1	1	1	0	1	0
3	6	*opcode*						

Description

The entire content (word) of *dst* is stored to memory (effective address). The memory address is formed from the stack pointer (SP) and an offset (number of words) that is a 6-bit unsigned constant (*ucst6*). If an offset is not given, the assembler assigns an offset of zero. You must type the brackets, [ ], around the specified offset, if you use the optional offset parameter.

The square brackets, [ ], indicate that the *ucst6* is scaled by a left-shift of 2 bits. After scaling, *ucst6* is added to SP. The result of the calculation is the effective address in memory that contains the content from *dst*.

Word addresses must be aligned on word (two LSBs are 0) boundaries.

Condition Codes

None

Pseudo Code

```
*(SP[ucst6]) = *dst
```

STW *Store Word to Memory with a SP-Relative 19-Bit Unsigned Constant Offset*

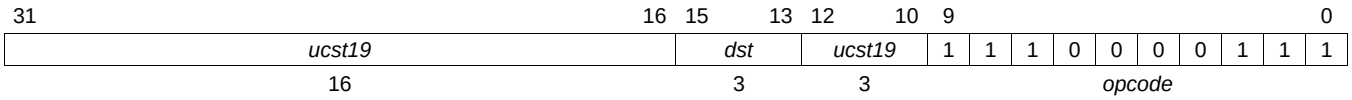
Syntax

STW *dst, *+***SP***[ucst19]*

Functional unit = D

Opcode

32 bit



Description

The entire content (word) of *dst* is stored to memory (effective address). The memory address is formed from the stack pointer (SP) and an offset (number of words) that is a 19-bit unsigned constant (*ucst19*). If an offset is not given, the assembler assigns an offset of zero. You must type the brackets, [ ], around the specified offset, if you use the optional offset parameter.

The square brackets, [ ], indicate that the *ucst19* is scaled by a left-shift of 2 bits. After scaling, *ucst19* is added to SP. The result of the calculation is the effective address in memory that contains the content from *dst*.

Word addresses must be aligned on word (two LSBs are 0) boundaries.

Condition Codes

None

Pseudo Code

```
*(SP[ucst19]) = *dst
```

STW *Store Word to Memory with a GDP-Relative 19-Bit Unsigned Constant Offset*

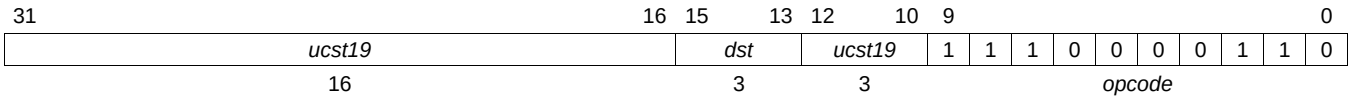
Syntax

STW *dst, **+**GDP[ucst19]*

Functional unit = D

Opcode

32 bit



Description

The entire content (word) of *dst* is stored to memory (effective address). The memory address is formed from the global data pointer register (GDP) and an offset (number of words) that is a 19-bit unsigned constant (*ucst19*). If an offset is not given, the assembler assigns an offset of zero. You must type the brackets, [ ], around the specified offset, if you use the optional offset parameter.

The square brackets, [ ], indicate that the *ucst19* is scaled by a left-shift of 2 bits. After scaling, *ucst19* is added to GDP. The result of the calculation is the effective address in memory that contains the content from *dst*.

Word addresses must be aligned on word (two LSBs are 0) boundaries.

Condition Codes

None

Pseudo Code

```
*(GDP[ucst19]) = *dst
```

STHI *Store 16-Bit Halfword to Memory with a 6-Bit Unsigned Constant Offset*

Syntax

STHI *ucst16, *+baseR[ucst6]*

Functional unit = S

Opcode

32 bit

31 16	15 13	12 10	9 7	6						0
ucst19	*ucst6*	*baseR*	*ucst6*	0	0	0	1	0	1	1
16	3	3	3						*opcode*	

Description

The 16-bit unsigned constant (*ucst16*) is stored to memory (effective address). The memory address is formed from a base address register (*baseR*) and an offset (number of halfwords) that is a 6-bit unsigned constant (*ucst6*). If an offset is not given, the assembler assigns an offset of zero. You must type the brackets, [ ], around the specified offset, if you use the optional offset parameter.

The square brackets, [ ], indicate that the *ucst6* is scaled by a left-shift of 1 bit. After scaling, *ucst6* is added to *baseR*. The result of the calculation is the effective address in memory that contains *ucst16*.

Halfword addresses must be aligned on halfword (LSB is 0) boundaries.

Condition Codes

None

Pseudo Code

```
*(baseR[ucst6]) = ucst16
```

STRF *Store Register File to Stack Pointer*

Syntax

STRF *op1, op2*

Functional unit = D

Opcode

16 bit

15 13	12 10	9	8	7	6						0
op2	*op1*	x	x	x	1	0	1	0	0	1	0
3	3							*opcode*			

Description

The current base register file is stored to the starting address specified by the postdecremented value of the stack pointer (SP). *op1* specifies the ending register (last register stored) and *op2* specifies the beginning register (first register stored). All registers between *op1* and *op2* are stored inclusive. The registers are stored in order from lowest register (*op2*) to highest register (*op1*).

Valid values of *op1* and *op2* are: *op1* ≥ *op2*

For *op1* < *op2*, results in unspecified behavior.

Condition Codes

None

Pseudo Code

```
for ( i = op2; i <= op1 ; i++) {
  *(SP) = Ri;
  SP -= 4;
}
```

SUB *Subtract 16-Bit Unsigned Constant from Register*

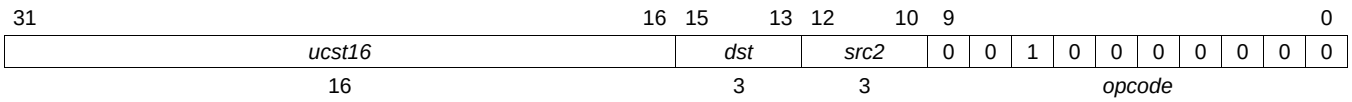
Syntax

SUB *ucst16, src2, dst*

Functional unit = D

Opcode

32 bit



Description

Subtract a 16-bit unsigned constant (*ucst16*) from *src2* and store result to *dst*.

Condition Codes

CSR[2]EQ = (dst == 0)

CSR[5]C = {not borrow} from (src2 - ucst16)

CSR[7]V = {overflow} from (src2 - ucst16)

Note that the carry (CSR[5]C) and overflow (CSR[7]V) bits are relevant to unsigned and signed operations, respectively. Appropriate bits are checked depending on if the operands are represented (or, interpreted) as unsigned or signed numbers.

Pseudo Code

```
dst = src2 - ucst16
```

SUB *Subtract 16-Bit Unsigned Constant from Stack Pointer, Result to Stack Pointer*

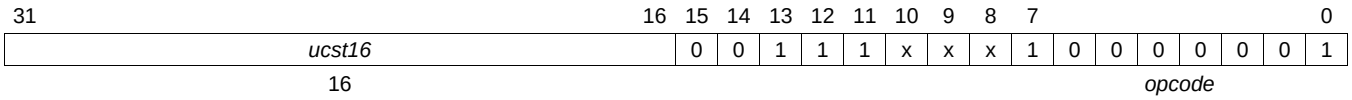
Syntax

SUB *ucst16,* **SP**

Functional unit = D

Opcode

32 bit



Description

Subtract a 16-bit unsigned constant (*ucst16*) from the stack pointer (SP) and store result to SP.

Condition Codes

None

Pseudo Code

```
SP = SP - ucst16
```

SUB *Subtract 16-Bit Unsigned Constant from Stack Pointer, Result to Register*

Syntax

SUB *ucst16,* **SP**, *dst*

Functional unit = D

Opcode

32 bit

31 16	15 13	12	11	10	9	8	7							0
ucst16	*dst*	0	0	1	1	1	1	0	0	0	0	0	1	0
16	3									*opcode*				

Description

Subtract a 16-bit unsigned constant (*ucst16*) from the stack pointer (SP) and store result to *dst*.

Condition Codes

CSR[2]EQ = (dst == 0)

CSR[5]C = {not borrow} from (SP - ucst16)

CSR[7]V = {overflow} from (SP - ucst16)

Note that the carry (CSR:C) and overflow (CSR:V) bits are relevant to unsigned and signed operations, respectively. Appropriate bits are checked depending on if the operands are represented (or, interpreted) as unsigned or signed numbers.

Pseudo Code

```
dst = SP - ucst16
```

SUB *Subtract 3-Bit Unsigned Constant from Register*

Syntax

SUB *ucst3, src2, dst*

Functional unit = D

Opcode

16 bit

15 13	12 10	9 7	6						0
dst	*src2*	*ucst3*	0	1	1	0	0	0	1
3	3	3	*opcode*						

Description

Subtract a 3-bit unsigned constant (*ucst3*) from *src2* and store result to *dst*.

Condition Codes

CSR[2]EQ = (dst == 0)

CSR[5]C = {not borrow} from (src2 - ucst3)

CSR[7]V = {overflow} from (src2 - ucst3)

Note that the carry (CSR:C) and overflow (CSR:V) bits are relevant to unsigned and signed operations, respectively. Appropriate bits are checked depending on if the operands are represented (or, interpreted) as unsigned or signed numbers.

Pseudo Code

```
dst = src2 - ucst3
```

SUB *Signed Subtraction of Two Register Values*

Syntax

SUB *src1, src2, dst*

Functional unit = D

Opcode

16 bit

15 13	12 10	9 7	6						0
dst	*src2*	*src1*	0	1	1	0	0	1	1
3	3	3	*opcode*						

Description

Subtract *src1* from *src2* and store result to *dst*.

Condition Codes

CSR[2]EQ = (dst == 0)

CSR[5]C = {not borrow} from (src2 - src1)

CSR[7]V = {overflow} from (src2 - src1)

Note that the carry (CSR[5]C) and overflow (CSR[7]V) bits are relevant to unsigned and signed operations, respectively. Appropriate bits are checked depending on if the operands are represented (or, interpreted) as unsigned or signed numbers.

The status of the CSR[5]C bit can be used to synthesize multiword (wider) subtraction. With the **SUB** instruction, the CSR[5]C bit is set if no borrow occurs and the CSR[5]C bit is cleared if a borrow occurs. In other words, for **SUB** instructions, the CSR[5]C bit represents a not borrow. To synthesize multiword subtractions, subsequent instructions can use the CSR[5]C bit as a NOT(borrow) operand, performing a normal subtraction if CSR[5]C == 1 and subtracting one more than usual if CSR[5]C == 0.

For example, if register pairs R0/R1 and R2/R3 hold 64-bit values (where R0 and R2 hold the least-significant words), the following instructions leave the 64-bit difference in the register pair R4/R5:

```
     SUB   R2, R0, R4       ; R4=R0-R2, sets CSR[C]
     MVC   CSR, R6          ; leaves CSR[C] untouched
     EXTU  5, 5, R6, R6     ; if (CSR[C]) R6=0x1 else R6=0x0
     BNEQ  L1
     SUB   R3, R1, R5       ; R5=R1-R3
     SUB   1, R5, R5        ; R5=R5-1 if CSR[C] was NOT set
L1:  NOP                    ; else R5=R5
```

Note that the above example needs appropriate modification in order to generate correct carry/overflow applicable for the total 64-bit result.

Pseudo Code

```
dst = src2 - src1
```

SWI *Software Interrupt*

Syntax

SWI

Functional unit = S

Opcode

16 bit

15	14	13	12	11	10	9									0
x	x	x	x	x	x	0	0	0	1	1	1	1	1	1	1
										opcode					

Description

Subtract *src1* from *src2* and store result to *dst*.

Condition Codes

SCSR = CSR; CSR[0] = 0

Pseudo Code

```
PC = *(IMEM[IST[SWI]]); IRP = PC + 1;
```

XOR — *Bitwise Exclusive-OR Unsigned 16-Bit Constant with Register*

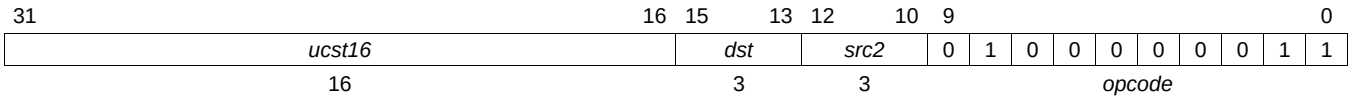
Syntax

XOR *ucst16, src2, dst*

Functional unit = L

Opcode

32 bit



Description

Bitwise exclusive-OR (XOR) of a zero-extended 16-bit unsigned constant (*ucst16*) with *src2* and store result to *dst*.

Condition Codes

CSR[2]EQ = (dst == 0)

Pseudo Code

```
dst = src2 XOR {zero extend}ucst16
```

XOR *Bitwise Exclusive-OR Two Registers*

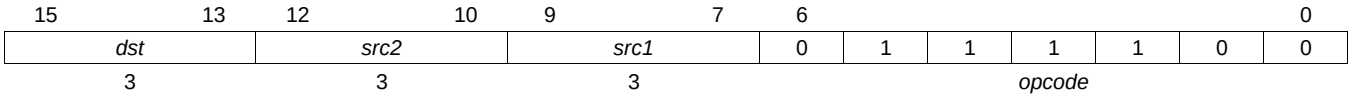
Syntax

XOR *src1, src2, dst*

Functional unit = L

Opcode

16 bit



15–13	12–10	9–7	6						0
dst	*src2*	*src1*	0	1	1	1	1	0	0
3	3	3	*opcode*						

Description

Bitwise exclusive-OR (XOR) of *src1* with *src2* and store result to *dst*.

Condition Codes

CSR[2]EQ = (dst == 0)

Pseudo Code

```
dst = src2 XOR src1
```

6.2.6 Clock, Reset, and Dynamic Power Management

6.2.6.1 Introduction

The ARP32 CPU has a single clock domain (cpu_fclk) and two reset domains:

- An asynchronous Power-On-Reset (cpu_porz_i)
- An asynchronous CPU Functional Reset (cpu_resetz_i)

Both of the reset signals are used asynchronously inside the design. Both the leading and trailing edges of these reset signals are synchronized with the CPU input clock by the external clock/reset management logic.

To reduce dynamic power consumption, the ARP32 CPU gates all of its internal clocks based on debugger connection status and CPU idle/standby status:

- If the ARP32 CPU is in the IDLE state and the CPU standby indication is asserted, all internal clocks of the ARP32 CPU remain gated.
- If debug connection is established, indicated via the input port cpu_dbgenable_i, the ARP32 CPU unconditionally enables all its internal clocks.

6.2.6.2 CPU Reset Modes

Table 6-349 summarizes the two reset signals of the ARP32 CPU and their effects. Table 6-350 summarizes the different application of the two resets.

Figure 6-52 illustrates the ARP32 CPU power-on-reset sequence.

If warm reset is applied without Power on reset, to avoid meta-stability issues within the CPU core, the functional clock to the CPU core must be stopped by external clock generation logic prior to Warm reset assertion and the functional clock must be restored prior to Warm reset de-assertion.

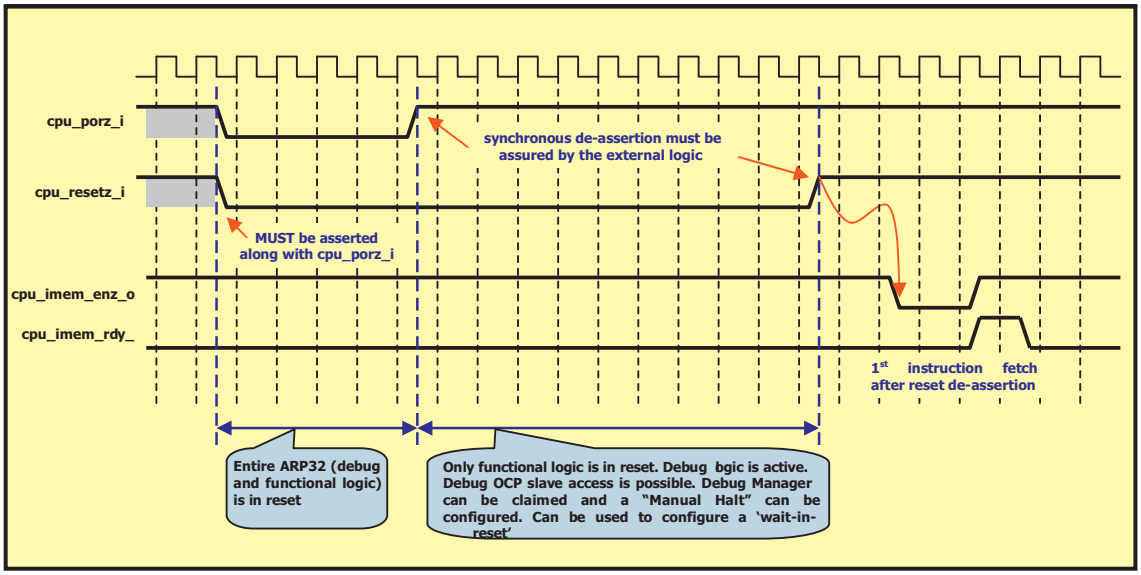
Table 6-349. CPU Reset Types

			Does CPU Reset?	
Reset Type	**Signal Name**	**Reset Type**	**Debug Logic**	**Functional Logic**
Power On Reset	cpu_porz_i	Async	Yes	No
Warm Reset	cpu_resetz_i	Async	No	Yes

Table 6-350. CPU Reset Modes

				Does CPU Reset?		Need Synchronization?	
cpu_ porz_i	**cpu_resetz_i**	**Reset Mode**	**Application**	**Debug Logic**	**Functional Logic**	**Assertion**	**De-assertion**
0	0	Power On Reset	Reset at power up, full system reset	Yes	No	NA	Yes
0	1	Not supported	NA	NA	NA	NA	NA
1	0	Warm Reset	Reset of CPU core only, for example, watchdog reset	No	Yes	Yes	Yes
1	1	Normal	No reset, normal running mode	No	No	NA	NA

Figure 6-52. Power-On-Reset



6.2.6.3 Dynamic Power Management

The ARP32 CPU supports instruction driven transition to a low-power state that is used to trigger a CPU system-wide low-power state. This is accomplished by the **IDLE** instruction. Upon execution of the **IDLE** instruction, the ARP32 CPU waits for any pending instruction or data memory transaction to complete and then goes to an endless wait state.

When in idle state, the CPU output signal cpu_standby_o is asserted high. During this window, the CPU gates clocks to all of its registers (except the interrupt flag register (IFR)) to achieve maximum saving of dynamic power. It is assured that the ARP32 CPU does not issue any new transactions on either instruction or data memory interfaces during this window. So, peripherals external to the CPU that don't need to be active while the CPU is in IDLE are also clock gated during this window.

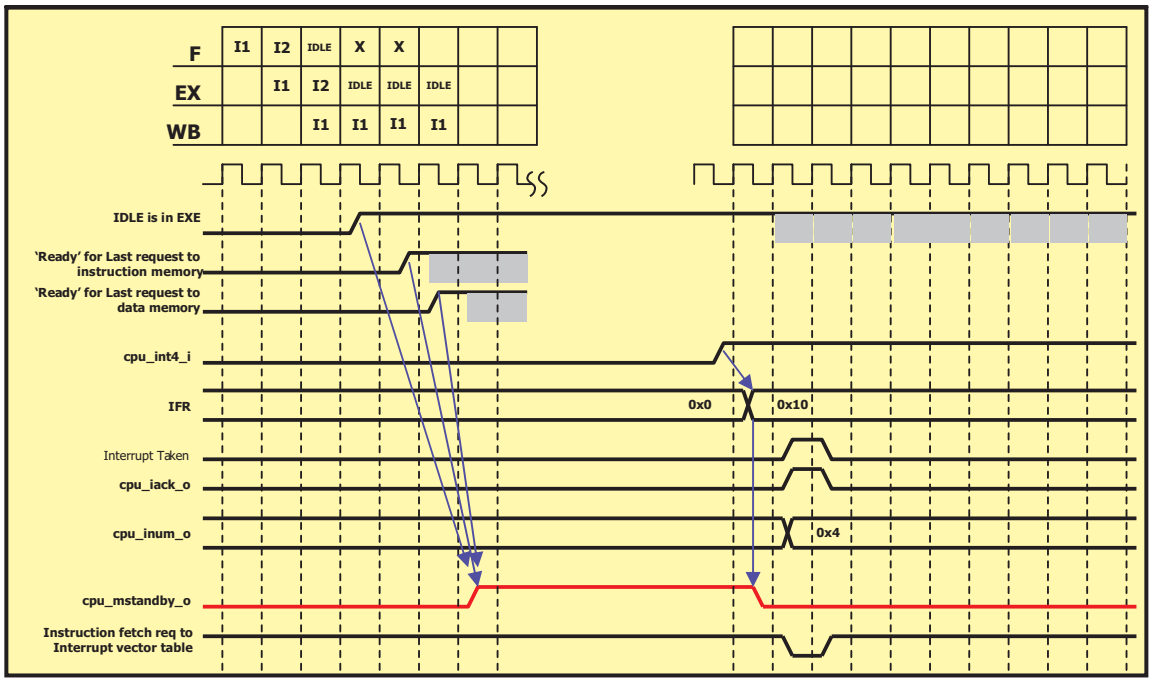
The CPU comes out of this wait state only via an enabled external interrupt (NMI, INT4-INT7) or reset. The cpu_standby_o signal is de-asserted as soon as one of the following events occurs:

- any of the interrupt flag register (IFR) bits are set
- a debug connection is established (cpu_dbgenable_i is asserted high)

Interrupt enable conditions (like GIE, IER, etc.) are not considered when de-asserting cpu_standby_o. Thus, a disabled interrupt being asserted at the CPU boundary causes the cpu_standby_o signal to be de-asserted but the CPU still remains in idle state until an enabled interrupt (or reset) occurs.

Figure 6-53 provides an illustrative waveform showing the CPU going into IDLE mode followed by a wakeup via INT4.

Figure 6-53. CPU Standby and Wakeup Procedure



6.2.7 Notes on Programming Model

6.2.7.1 Booting

The entry for reset in the ARP32 interrupt service table (IST) is setup such that the program flow after reset calls an initialization or boot code as soon as possible and performs at least the following functions:

- Initialize the stack pointer (SP) register.
- Initialize the global data pointer (GDP) register.
- Call 'main' function.

The following tasks are additionally performed if interrupt processing is needed from the very beginning; else, they are done as soon as the system is ready for (or expecting) interrupt processing:

- Initialize the interrupt enable register (IER) to enable the required maskable interrupts, as required.
- Initialize the interrupt enable register (IER) to enable the nonmaskable interrupt, if required.
- Initialize the control status register (CSR) to enable all the maskable interrupts globally.

6.2.7.2 Enabling and Disabling Interrupts

The ARP32 CPU provides a mechanism for easy and flexible interrupt control.

6.2.7.2.1 Globally Enabling or Disabling Maskable Interrupts

The global interrupt enable (GIE) bit in the control status register (CSR) allows programmers to enable or disable all maskable interrupts by controlling the value of a single bit. On the ARP32 CPU, programs must directly manipulate the GIE bit in CSR to disable and enable interrupts:

- CSR[0]GIE = 1 enables the maskable interrupts so that they are processed
- CSR[0]GIE = 0 disables the maskable interrupts so that they are not processed

For example, the following code sequence globally enables all maskable interrupts:

```
MVC   CSR, R0        ; Get CSR into R0
SET   0, 0, R0, R0   ; Set R0[0]
MVC   R0, CSR        ; Copy R0 back to CSR (sets GIE)
```

Similarly, the following code sequence globally disables all maskable interrupts:

```
MVC   CSR, R0        ; Get CSR into R0
CLR   0, 0, R0, R0   ; Clear R0[0]
MVC   R0, CSR        ; Copy R0 back to CSR (clears GIE)
```

As interrupt detection occurs in parallel with CPU execution, the CPU takes an interrupt in the cycle immediately following an **MVC** instruction that clears the GIE bit. However, CPU context save/restore behavior ensures that interrupts do not occur after subsequent instruction. Consider the following code example where the CPU takes an interrupt between instructions 1 and 2, between instructions 2 and 3, or between instructions 3 and 4. The CPU does not service an interrupt between instructions 4 and 5.

```
; assume GIE=1
MVC   CSR, R0        ;(1) Get CSR
AND   -2, R0, R0     ;(2) Get ready to clear GIE
MVC   R0, CSR        ;(3) Clear GIE
ADD   R0, R2, R3     ;(4)
ADD   R4, R4, R5     ;(5)
```

If the CPU services an interrupt between instructions 1 and 2 or between instructions 2 and 3, the SCSR[0]GIE bit holds the value 1 when arriving at the interrupt service routine. If the CPU services an interrupt between instructions 3 and 4, the SCSR[0]GIE bit holds the value 0. Thus, when the interrupt service routine resumes the interrupted code, it resumes with the GIE bit cleared (as a result of context restore of SCSR to CSR) as the interrupted code intended.

6.2.7.2.2 Enabling or Disabling Individual Interrupts

Software can enable and disable individual interrupts by setting and clearing the bits in the interrupt enable register (IER) that correspond to the individual interrupts. An interrupt can trigger interrupt processing only if the corresponding bit in the IER is set.

For example, the following code sequence enables INT4:

```
MVC   IER, R0        ; Get IER into R0
SET   4, 4, R0, R0   ; Set R0[4]
MVC   R0, IER        ; Copy R0 back to IER (sets IER[4])
```

Similarly, the following code sequence disables INT5:

```
MVC   IER, R0        ; Get IER into R0
CLR   5, 5, R0, R0   ; Clear R0[5]
MVC   R0, IER        ; Copy R0 back to IER (clears IER[5])
```

6.2.7.3 Stack Usage in Interrupt Service Routine

The ARP32 CPU has only one stack pointer and thus a unified stack is used for background and interrupt code. As a result, the programmer is responsible for maintaining a clean boundary of operations that modifies the stack pointer (for example, stack allocation and de-allocation for a function, context save onto stack via push/pop operations using the **LDRF/STRF** instructions).

Instructions/operations in the CPU that modify the stack pointer are designed so that an interrupt cannot disrupt the operations halfway through to completion. For example, a stack allocation and de-allocation for function can happen via the **ADD** *ucst16,* **SP** instruction that atomically modifies the stack. The return address save while taking a CALL or retrieving the return address while executing a RET is also atomic, and cannot be interrupted. The **LDRF/STRF** instructions used for single or multiple pop and push are also uninterruptable. The ARP32 C/++ compiler uses these instructions to maintain the C/C++ stack and thus the sanity of the stack is always maintained.

Under certain scenarios, to save additional user context onto the stack, it is done via the **LDRF/STRF** instruction. **LDRF/STRF** modifies the stack pointer and hence results in the stack layout of a function to be not valid anymore. Hence, such additional context save/restore to the stack must happen at function boundaries only.

These considerations are important for assembly level programmers doing additional context save/restore (beyond what the ARP32 compiler does) and also to new custom instruction design for the ARP32 CPU if they depend on stack pointer relative operations.

6.2.7.4 General Restrictions

The ARP32 CPU has the following restrictions on the programming model:

- **Long Long (as 64 bit) is not supported natively in the architecture.** 'long long' is supported via emulation in the Run Time Support (RTS) library. Since this is non-standard and sub-optimal, use of long may severely degrade performance. Hence, it is highly recommended not to use the C 'long long' datatype.
- **64-bit arithmetic.** Add/sub of more than 32-bit values (operand or results) requires **ADDC/SUBC/BC** (addition with carry, subtraction with borrow, branch on carry) instructions that are not supported in the ARP32 CPU. Again, it is possible to emulate this in RTS but it is a sub-optimal implementation and it is not recommended.
- **Multi-threading via mutex is not supported in the architecture.** In the ARP32 CPU, it does not implement an "EXCLUSIVE access instruction" to implement a 'proper' mutex in RTS. A polling based multi-threading is still possible.

6.3 VCOP CPU and Instruction Set

NOTE: Throughout this section, various sample assembly code pieces are used to illustrate VCOP architecture and functionality. Programmers are expected to write code in VCOP Kernel-C, instead of in assembly.

6.3.1 Module Overview

The Embedded Vision Engine (EVE) module is a programmable imaging and vision processing engine, intended to be used in devices that serves consumer electronics imaging and vision applications. Its programmability allows late-in-development or post-silicon processing requirements to be met, and allows third party or customers to add differentiating features in imaging and vision products.

The EVE Module consists of an ARP32 scalar core, a VCOP vector core, and a DMA controller.

6.3.2 Features

- Interfaces
  - CLK/2 MHz custom interface with the scalar core
- Functionality
  - CLK MHz functional clock
  - Based on VICP coprocessor
  - Dual-issue, N-way SIMD per instruction
  - Scalable in N ways of SIMD, N = {2 | 4 | 8 | 16 | 32}
  - For the first version of VCOP N = 8
  - 16 entries x N ways x 40-bit register file
  - Binary morphology acceleration
  - N-way table lookup and histogram acceleration (capable of N lookups per cycle or N histograms per 4 cycle)
  - Up to 20-bit vector data memory address space
- Debug interface with visibility of loop variables and load/store pointers (VCOP_LD_PTR_i/VCOP_ST_PTR_j).

6.3.3 Block Diagram

The EVE subsystem contains an ARP32 scalar core, VCOP vector core, EDMA, memory blocks, memory switch, local interconnect, reset/clock, interrupt controller, SCTM and SMSET instrumentation blocks. Figure 6-54 provides a block diagram.

Figure 6-54. EVE Block Diagram

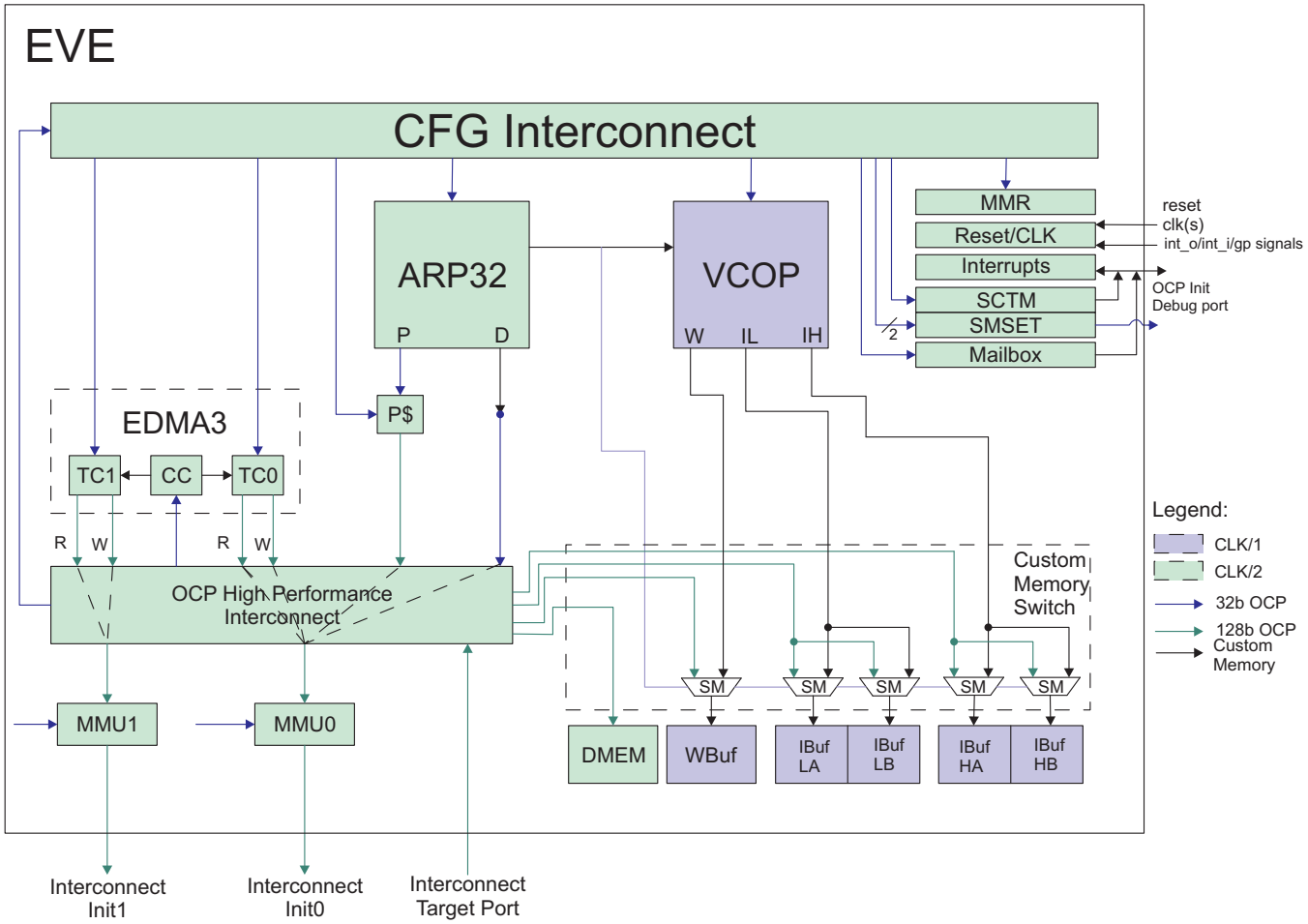
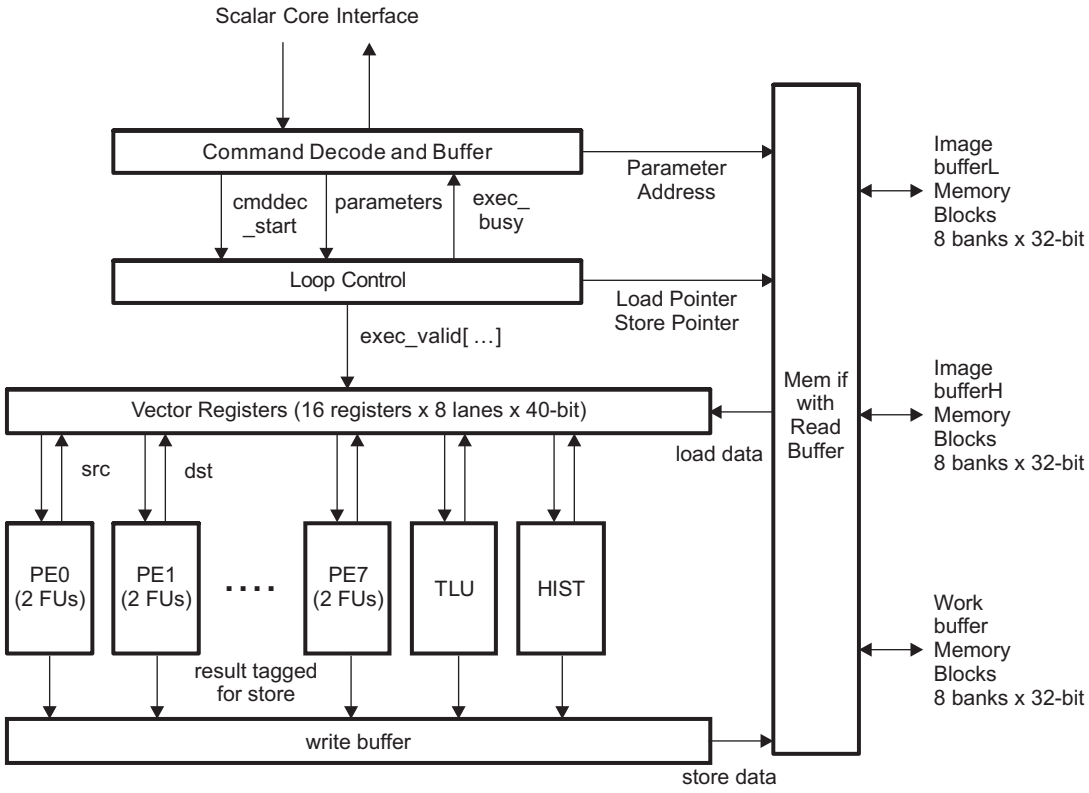


Figure 6-55 depicts the functional block diagram of EVE's Vector Coprocessor (VCOP).

Figure 6-55. VCOP Block Diagram



6.3.4 System Interfaces

6.3.4.1 Interrupts

The vec_err_intn interrupt is driven by the vector core and indicates that the vector core has encountered an error condition, see Section 6.3.7.2.1 VCOP Register Manual for details.

6.3.4.2 Configuration Bus Slave Port

VCOP has a configuration bus slave interface that follows the OCP 32-bit slave protocol. The interface enables module configuration, status, processor state readout during debug.

The interface operates in the same clock domain as clk_eve_sca, the scalar core and scalar/vector interface clock.

The memory map of the configuration bus is specified in Section 6.3.7.2.1.

6.3.4.3 Performance Counter Interface

On the EVE level there is an SCTM module and an SMSET module to collect performance statistics.

SCTM is for counting signal active time and rise and fall transitions. SMSET is for conveying events to include into the system trace. Table 6-351 lists the signals that are exported to SCTM or SMSET as indicated. All SCTM signals are level signals operating at full vector core speed (clk_eve_vec), and all SMSET signals are rising-edge triggered signals.

Table 6-351. Performance Counter Signal List

Signal	Destination	Description
vcop_busy	SCTM	Indicate VCOP busy, for measuring VCOP loading
vcop_idle_and_done	SCTM	Indicate VCOP idle and done, basically not(vcop_busy) and vec_done, for measuring slack in real-time schedule
vcop_wait_for_arp32	SCTM	Indicate VCOP idle and waiting for ARP32 to feed instruction, basically vec_rdy and not(vec_instr) and not(vcop_busy). This indicates VCOP under utilization that might be improved by reducing ARP32 interrupt services
vcop_arp32_awaits	SCTM	Indicate ARP32 has vector instruction ready and is waiting for VCOP to accept it, basically not(vec_rdy) and vec_valid. This indicates ARP32 idle time that might be utilized, however, not to the extent that vcop_wait_for_ARP32 time becomes significant.
vcop_overhead	SCTM	Indicate VCOP busy but not executing the loop, meaning exposed command decode (from decode-execute pipeline) and parameter access time. The former might be reduced by having more back-to-back loops, and the latter is unavoidable.
vcop_ld_stall_by_st	SCTM	Indicate LD stage is stalled by ST stage (forced write when write buffer is full). This means there is insufficient gaps in LD for ST to occur in the background, but this does not mean LD+ST is slowing down computation.
vcop_op_stall_by_ldst	SCTM	Indicate gaps in operation stage between iterations due to LD+ST, indicating extent of load/store bound (as opposed to compute-bound). This may be reduced by merging loops to reduce load/store.
vcop_op_stall_by_ dependency	SCTM	Idle cycle inserted in operation stage due to dependency. This may be minimized by scheduling.
vcop_rd_ibufl	SCTM	Indicate VCOP reading from IBUFL
vcop_rd_ibufh	SCTM	Indicate VCOP reading from IBUFH
vcop_rd_wbuf	SCTM	Indicate VCOP reading from WBUF
vcop_wr_ibufl	SCTM	Indicate VCOP writing to IBUFL
vcop_wr_ibufh	SCTM	Indicate VCOP writing to IBUFH
vcop_wr_wbuf	SCTM	Indicate VCOP writing to WBUF
vcop_loop_start	SMSET	Indicate VCOP getting a VLOOP instruction
vcop_done	SMSET	Indicate VCOP executing a VWDONE instruction, copy of vec_done

6.3.4.4 Data Memory Map

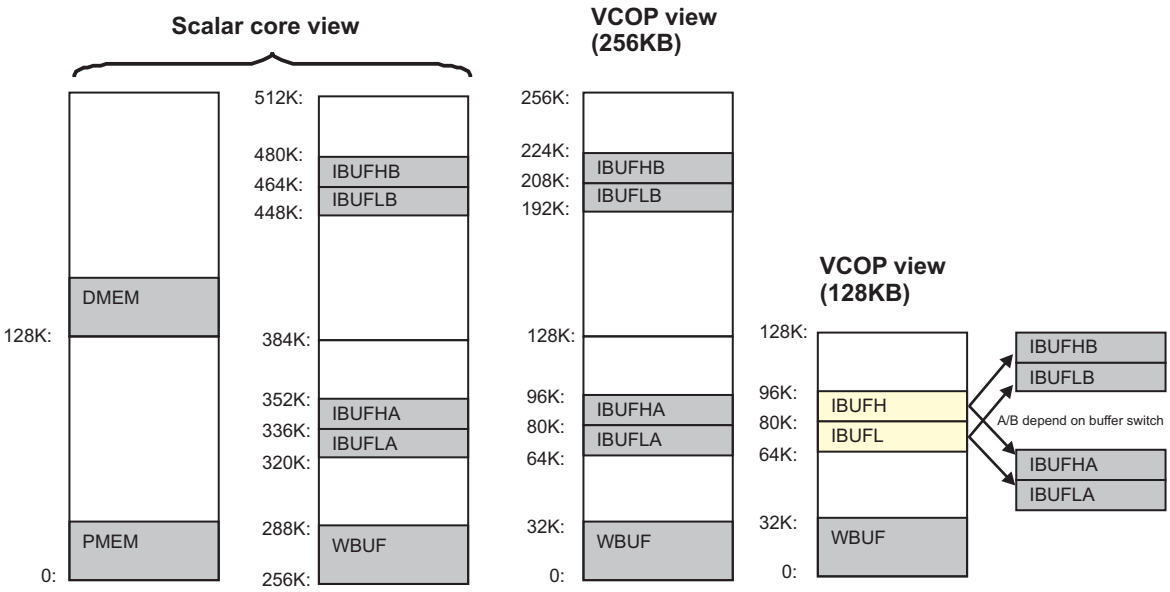
The first instance of EVE has a data memory configuration and a memory-map as listed in Table 6-352.

Memory maps for scalar and vector core are shown in Figure 6-56. An optional MMR field, VEC_MEM_128KB, wraps the VCOP memory map around 128KB, aliasing the A and B sets to the same address range.

Table 6-352. EVE Vector Data Memory Map

Byte Address	Name	Description
0x00000 ~ 0x07FFF	WBUF	Vector working buffer, 32Kbytes
0x08000 ~ 0x0FFFF	reserved	Reserved for future expansion
0x10000 ~ 0x13FFF	IBUFLA	Image buffer low copy A, 16 Kbytes
0x14000 ~ 0x17FFF	IBUFHA	Image buffer high copy A, 16 Kbytes
0x30000 ~ 0x33FFF	IBUFLB	Image buffer low copy B, 16 Kbytes
0x34000 ~ 0x37FFF	IBUFHB	Image buffer high copy B, 16 Kbytes

Figure 6-56. EVE Memory Map



6.3.5 Functional Description

6.3.5.1 Scalar-Vector Architecture

The EVE subsystem consists of a scalar core as the subsystem controller, a vector core as the high-throughput coprocessor, shared memories, DMA, clock/reset and interrupt controllers. See the *EVE Subsystem Reference Guide* for details.

6.3.5.1.1 Scalar Core

The scalar core executes conventional RISC instructions. The scalar core is based on the ARP32 processor.

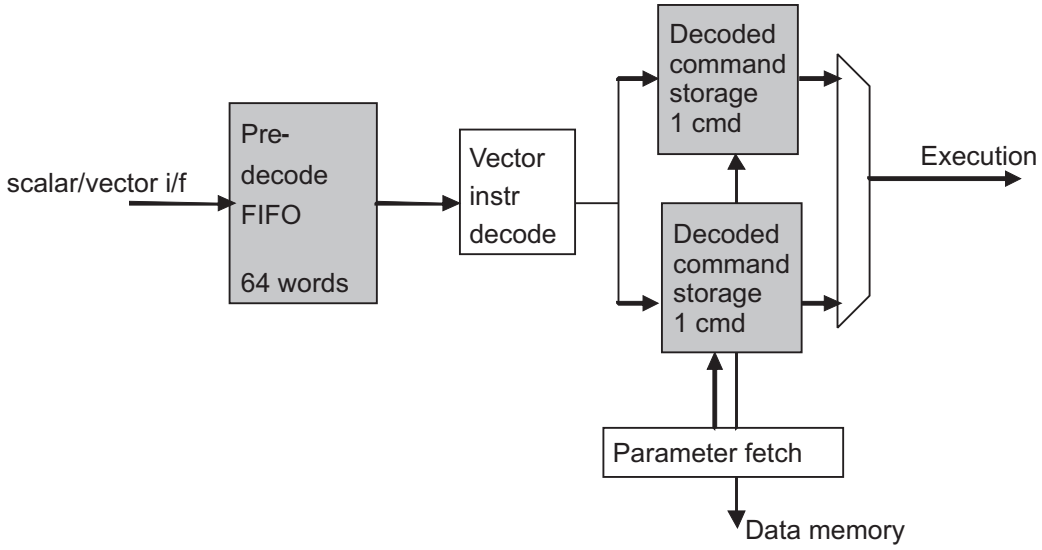
See the *ARP32 CPU and Instruction Set Reference Guide* for more details.

6.3.5.1.2 Scalar-Vector Interaction

The EVE scalar and vector cores share one program memory, but can execute in parallel on separate threads. The scalar core has sole control of the program memory, so all vector instructions are accessed by the scalar core, recognized being a vector instruction, and relayed via the scalar/vector interface to the vector core for execution. The vector core executes a sequence of instructions (called a vector command) repeatedly, so typically can be busy executing and does not need further vector instructions for thousands of clock cycles at a time.

The vector core has temporary instruction storage. There is a pre-decode buffer of 64 words, plus a decoded buffer for two maximal-sized commands (see Figure 6-57). The vector core attempts to decode and fill the two commands first, and if vector instructions keep coming, fill the pre-decode buffer. Together the buffering holds 4 or more typical vector commands, and keep the vector core busy for a while, allowing the scalar core to switch context to serve interrupts when interrupts occur.

Figure 6-57. VCOP Instruction Buffering



6.3.5.2 Vector Core Overview

The vector core is a SIMD machine with built-in loop control and address generation. It is programmed in array or 2D block processing level. The vector core has the following resources:

- 4 nested for loops, with loop variables i1, i2, i3, and i4, plus an optional outer loop i0
- 8 address generators, each capable of 4-dimensional addressing, that is, the address pattern is: base + i1 × const1 + i2 × const2 + i3 × const3 + i4 × const4
- 16-entry vector register file, each entry is N-way SIMD × 40-bit signed (sign-extended or zero-padded from 8/16/32-bit signed/unsigned memory data, or from operation upon register data)
- Two general-purpose functional units, each N-way SIMD. Functional spec supports N = {2 | 4 | 8 | 16 | 32}, first instance N = 8.
- Table lookup unit supporting up to N parallel lookups.
- Histogram unit supporting up to N parallel histogram operations.
- 8 load units
- 8 store units

The vector core supports the following functions:

- Generic compute
- Table lookup
- Histogram and weighted histogram

6.3.5.2.1 Nested for Loop Model

EVE vector operation is controlled by a four-level nested for loop. Inside the loop, the behavior can be represented in sequential stages: load, arithmetic operation, store, and pointer update.

In each stage, a number of load, operation, stores, or pointer updates are carried out, with respect to the 16 × N × 40-bit vector register file. There is no dependency among loads, stores, or pointer updates; multiple loads unto the same register file is not allowed. Dependency among operations assumes 2 operations are carried out in parallel, with no delay slot for most operations, and 1 cycle of delay slot for some specific operations.

The vector register file, loop variables, address generators are replicated as needed in the hardware to support pipelining among the various stages.

Example 6-8 shows a skeleton of the nested loop model. There are four loop variables, i1, i2, i3, and i4. A snapshots of the loop variables are contained in the follwoing registers:

- VCOP_I0_I1[15:0] I0 - I0 loop variable
- VCOP_I0_I1[31:16] I1 - I1 loop variable
- VCOP_I2_I3[15:0] I2 - I2 loop variable
- VCOP_I2_I3[31:16] I3 - I3 loop variable
- VCOP_I4[15:0]I4 - I4 loop variable

Example 6-8. Nested Loop Model Skeleton

```
EVE_compute(...)
{
   for (i1=0; i1<=lpend1; i1++) {
      for (i2=0; i2<=lpend2; i2++) {
         for (i3=0; i3<=lpend3; i3++) {
            for (i4=0; i4<=lpend4; i4++) {

               for (k=0; k<num_inits; k++)
                  initialize_vreg_from_parameters(...);

               for (k=0; k<num_loads; k++)
                  load_vreg_from_local_memory(...);

               for (k=0; k<num_ops; k++)
                  op(...);         // 2 functional units, executing 2 ops per cycle

               for (k=0; k<num_stores; k++)
                  store_vreg_to_local_memory(...);

               for (k=0; k<num_agens; k++)
                  update_agen(...);
            }
         }
      }
   }
}
```

Each stage is described in more details in the following sections.

Note that loads and stores are predicated on the loop variables matching specific conditions and are thus not always carried out at every i4 iteration.

Each i4 iteration takes a number of cycles equal to the maximal number of cycles spent in loads, arithmetic operatoins, and stores. Cycle count in the arthmetic operations is constant for each loop, but cycle count in load and store can change depending on pointer update, loop level, and read/write memory contention.

For example, with lpend1 = 3, lpend2 = 1, lpend3 = 2, and lpend4 = 4, the nested for loop executes in exactly 4 × 2 × 3 × 5 = 120 i4 iterations. For the first few iterations the loop variables progress as:

i1	i2	i3	i4
0	0	0	0
0	0	0	1
0	0	0	2
0	0	0	3
0	0	0	4
0	0	1	0
0	0	1	1
0	0	1	2
...			

6.3.5.2.2 Instruction Organization

Vector instructions cannot appear in the program stream in arbitrary order. Instructions are classified as:

- Control
  - Repeat end count
  - Parameter pointer
  - Write buffer switch
- Synchronization
  - Wait for ready
  - Wait for done
- Computation
  - VLOOP
  - Register initialization
  - Address generator
  - Load
  - Operations
  - Store

The sequencing rule is that computation instructions that form a loop are in one contiguous block, starting with VCOP_VLOOP_PTR[31:0]VLOOP_PTR. This block of computation instructions is called a vector command. Control, synchronization, and scalar instructions can appear in any order, except that they cannot appear inside a vector command.

6.3.5.3 Vector Control

The VCTRL instruction is used to convery control information to the vector core. The instruction syntax is:

VCTRL *scalar_register*, *control_register*

The control registers so far defined are 0: RPT_END and 1: PARAM_PTR (VCOP_PARAM.

6.3.5.3.1 Repeat End Count

A VCTRL instruction variation programs the repeat end count. The syntax is:

VCTRL *scalar_register*, **RPT_END**

Subsequent VLOOP is executed RPT_END + 1 times. Reset default is 0, or repeating just once, and is the normal mode of operation. At end of loop execution, RPT_END is automatically reset to 0, so if the subsequent is non-repeating, which is most of the cases, there is no need to issue another VCTRL RPT_END to reset it.

RPT_END is a 12-bit value; the largest value supported is 0:4095, allowing for up to 4096 iterations of the repeat loop.

Note that the VLOOP instruction that follows can be of any type (computer, table lookup, histogram).

The repeat mechanism can be viewed as enclosing VLOOP by an outer loop, i0. The difference between this i0 loop and the i1, i2, i3, i4 loops inside VLOOP is that parameter pointer is advanced at the end of VLOOP. The repeat feature allows the same program code to be executed with a different set of parameters each time.

The repeat feature is useful for performing the same processing steps on a list of data regions that are not spaced regularly and/or not having the same dimension. Instead of having a scalar loop around the repeated VLOOP, using the repeat feature reduces scalar/vector interaction and resulting overhead.

Software must allocate memory appropriately to allow results of these data blocks to be stored (sequentially or otherwise is under programmer's control).

6.3.5.3.2 Parameter Pointer

In addition to the vector instructions in PMEM, the vector core must read a parameter file in WBUF or IBUFL/IBUFH (depending on the pointer value) before it begins intended computation. The parameters include loop counts, address pointers to arrays, constants used in the computation, round/truncate shift count, saturation bounds, etc. The parameter pointer is located in VCOP_PARAM_PTR[31:0] PARAM_PTR.

Separation of configuration coded in the program memory versus configuration conveyed in data memory is essential, as this allows library functions to be constructed with reasonable code size and can serve a variety of need in applications. Loop counts and address pointers being conveyed via parameters is needed, so that, for example, one filter function works for various data/coefficient/output arrays and block and coefficient kernel sizes. Round/truncation shifts being convey via parameters means that, for example, the same filtering kernel program can work for filters with different radix point bit positions for filter coefficients.

Parameter registers are 16-bit per entry in the parameter file, up to 64 entries. The first 2 entries of the parameter file, P0 and P1, are hard-coded to constant values 0 and 1 respectively, as these are commonly used and it reduces code size by referring to these constants without taking up data memory space.

Parameters are referred to in the vector instructions either as single register, 16-bit parameter, or as a register pair, together make up 32-bit. Where a pair is used, the first parameter must be an even index, and contains the lower 16-bit of the 32-bit.

From an application development point of view, algorithm kernels are sometimes coded in one vector command per kernel, sometimes in several commands. Thus it is beneficial for the vector core to advance the parameter pointer automatically to reduce overhead. This requires a mechanism to specify or modify the parameter pointer when needed. This is addressed by the parameter pointer update feature through VCTRL.

A VCTRL instruction variation updates the parameter pointer for the subsequent VLOOP. The syntax is:

VCTRL *scalar_register*, **PARAM_PTR**

Unless updated again by VCTRL, the parameter pointer is advanced automatically at the end of VLOOP execution by the #PL field of VLOOP. Thus, parameter blocks for multiple consecutive VLOOPs can be placed consecutively in data memory, and no VCTRL VCOP_PARAM_PTR is needed except for the first VLOOP.

For efficiency of parameter access by VLOOP, the pointer value conveyed via VCTRL must be 32-bit aligned. The parameter length specified in VLOOP is also 32-bit aligned. Thus, at the beginning of any VLOOP, the pointer is 32-bit aligned.

Note that inside a VLOOP while parameters are pulled off to configure the loop, the parameter pointer is only 16-bit aligned.

For example, if 11 parameter registers are used in a vector command, P0..10, the encoded parameter length is ceiling((11 – 2)/2) = 5. The subsequent next vector command accesses parameter registers 5 32-bit words from the current vector command, leaving one 16-bit halfword in memory unused.

6.3.5.3.3 Switch Buffers

To minimize task-switching overhead, an instruction is added to write to buffer switch MMR. The instruction syntax is:

VSWITCHBUF *ucst20*

Where ucst20 is a 20-bit immediate value to be written to a dedicated MMR for buffer switches.

This instruction is carried out by the scalar core, and vector core just ignores it.

Change made to the buffer switch is visible in MMR as well, as both VSWITCHBUF and generic scalar core store to the buffer switch MMR are both valid ways to program the buffer switches, only that VSWITCHBUF achieves lower latency. Both mechanisms are kept consistent in the simulator/ debugger toolset.

6.3.5.4 Vector-Scalar Synchronization

6.3.5.4.1 Wait for Vector Core Done

The VWDONE instruction is used as a synchronization barrier between scalar and vector core, and affects the scalar/vector interface signal vec_done. Upon reset, vec_done is true. Upon execution of VWDONE in the scalar core, the scalar stalls and waits for vec_done. The vector core, upon exeucting this instruction, implying all previous submitted vector instructions are all done, asserts vec_done. Once asserted, vec_done stays true until receiving a subsequent VLOOP that turns the vector busy and vec_done false again.

The VWDONE instruction is typically used before the scalar core switches shared memory to itself or DMA. Before it changes the switch it makes sure that the vector core is idle and thus not accessing the memory.

VWDONE must appear between vector commands in the program, not inside a vector command.

6.3.5.4.2 Wait for Vector Core Ready

The VWRDY instruction is used to stall the scalar core until vector core is ready to accept additonal vector instructions. Vector core ignores this instruciton.

Normally the scalar core can simply issues vector instruction, and when the vector core is not ready, VCOP_STATUS[2]VEC_RDY = 0 the scalar core is automatically stalled. Having VWRDY allows inserting timer read before and after VWRDY to measure the time duration of vector core not being ready, for performance tuning purposes.

6.3.5.5 Vector Computation

6.3.5.5.1 Vector Loop

The VLOOP instruction marks the beginning of a vector command specifying a loop. Assembly format:

VLOOP *cmd_type* **,** *CL#* **:** *cmd_len***,** *PL#***:***param_len*

The *cmd_type* field specifies whether it is the main compute (COMP), table lookup (TLU), or histogram (HIST).

Command length (*cmd_len*) and parameter length (*param_len*) in 32-bit words are encoded to facilitate quick parsing of the vector commands. The ending loop counts, lpend1/2/3/4, are encoded at fixed entries in the parameter file, P2 = lpend1, P3 = lpend2, P4 = lpend3, P5 = lpend4. This allows easy adaptation of EVE function to different array dimensions.

The first two entries of the parameter file have implicit values and are thus not conveyed in the data memory. P0 = 0 and P1 = 1. **The param_len does not include these two entries.**

The binary code of VLOOP includes a *version* field, which allows a compiler/assembler to indicate which version of EVE hardware the vector command is compiled for. The first version has *version* = 0, the next version, *version* = 1, and so on. This allows future version of EVE to offer backward compatibility; that is, execute previous-version binary as is. For example, if a future version of EVE has 16-way SIMD, it can execute 8-way SIMD binary by disabling half of the datapath.

Each vector command can have:

- Up to 8 address generators
- Up to 16 vector register initialization
- Up to 8 loads
- Up to 40 operations
- Up to 8 stores

Maximal command length is 81 instructions (VLOOP + 16 VINITs + 8 VAGENs + 8 VLDs + 40 operations + 8 VSTs).

6.3.5.5.1.1 Retention of State Between VLOOPs

From one VLOOP to the next VLOOP, the following state information is retained:

- Parameter pointer, advanced by parameter length in the first VLOOP
- Vector register contents, whatever left over from the last iteration of the first VLOOP

The following state information is not retained, thus must be configured within each VLOOP:

- Parameter register file (must be reloaded from data memory)
- Agen configuration, such as association of the four counts to parameter registers
- Agen address counter (reset at the beginning of each VLOOP execution)

6.3.5.5.2 Vector Register Initialization

It is useful to initialize vector register with constants provided in the parameter file. As parameter file is scalar (not NWAY SIMD), each value is broadcast to all NWAY of the destination vector register. Assembly syntax:

VINIT *type_init_loop preg***,** *vreg*

The data *type* can be signed/unsigned halfword or word. In case of word, two parameter register entries (16-bit each) are used, first one being copied to the lower halfword, and the second one to the upper halfword.

For the word type initialization, *preg* specified is the lower halfword and must be even. For example, *VINITWU_ONCE P10, V2* initializes V2 using (P11 << 16) + P10.

The *init_loop* field can be {ONCE, I234_ZERO, I34_ZERO, I4_ZERO, ALWS }.

The VINIT instruction is equivalent to initializing a specific vector register in the nested i1/i2/i3/i4 loop shown in Section 6.3.5.3.2:

- ONCE is for initialization before the loop.
- I234_ZERO is for initialization inside the i1 loop, before entering the i2 loop.
- I34_ZERO is for initialization inside the i2 loop, before entering the i3 loop.
- I4_ZERO is for initialization inside the i3 loop, before entering the i4 loop.
- ALWS is for initialization inside the i4 loop, running every iteration.

The VINIT feature is used to initialize vector registers to certain initial values. For example, FIR filtering is implemented by iterating over filter taps with MADD instructions, and VINIT is used to initialize the accumulator to zero at the proper iteration, i4 for 1-D filter, and i3 for 2-D filter. Constants are brought into the computation loop with VINIT as well. For example, to scale an entire input array by a constant factor.

Note that VINIT broadcasts one value to all SIMD lanes. Initializing a vector register to multiple values is accomplished with a vector load (VLD) with zero address increment.

Sometimes it is useful to refer to loop variables in the computation. For example, to collect the maximal five values in an array and recording their indices, software can first do this in SIMD fashion to get to 5 sets of values and indices, then merge among SIMD lanes. Indices are readily copied from the loop variable i4, instead of spending cycles to compute with ALU operations.

VINIT provides this capability by extending the parameter count to cover loop variables:

- P64 = i0, the repeat outer loop enabled via VCTRL <scalar_reg>, RPT_END
- P65 = i1
- P66 = i2
- P67 = i3
- P68 = i4

When using these parameter counts, only the unsigned halfword type is supported.

6.3.5.5.3 Address Generator (agen)

In image, video, vision computation, there are often many operations performed on arrays of the same dimension; which can use the same addressing pattern (with different base addresses) in the memory read/write. The address generator resource is included to take advantage of this. One agen can be shared among multiple loads and/or multiple stores.

The VAGEN instruction has this assembly format:

VAGEN *agen*, [*pinc4*, *pinc3*, *pinc2*, *pinc1*]

Agen field is a label (A0..7) to identify one of 8 agen resources.

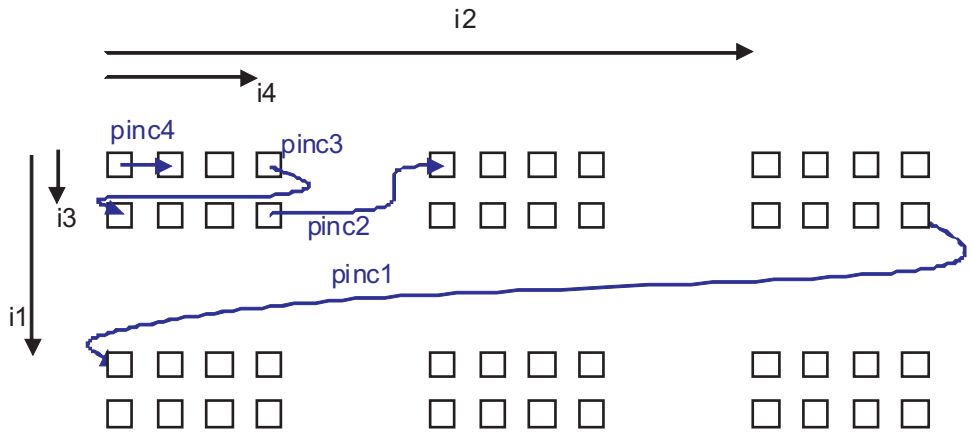
VAGEN is defined in strict linear order A0, A1, A2, till the last agen used in the command. The agens defined can be used (in loads and stores) in any order. It is allowed to share agen between load(s) and store(s).

The address increments represent:

- pinc4: inside the i4 loop, default
- pinc3: inside the i3 loop, when i4 matches the ending loop count
- pinc2: inside the i2 loop, when where i3 and i4 both match the ending loop counts
- pinc1: inside the i1 loop, when i2, i3, and i4 all match the ending loop counts

Figure 6-58 shows an example of 2 × 3 × 2 × 4 data object, addressed by the loop variables i1, i2, i3 and i4, with i1 being the outer-most loop. The address increments to access the object are shown in Figure 6-58.

Figure 6-58. Addressing a Four-Dimensional Data Object



In the VAGEN instruction, pointers to the parameter file are encoded, not the address increments themselves. Most data objects do not require 4-dimension addressing. For example, coefficient data for 2D FIR filter is normally stored consecutively, and advance by 1 data point most of the times, and wraps back at the end of i3 loop.Then pinc4 = pinc3 is specified to one parameter register (initialized to 2 for 16-bit coefficients), and pinc2 = pinc1 to another parameter register for the wrap-back. Thus, this scheme reduces command plus parameter data.

All address increments are in bytes. This allows misaligned halfword (16-bit) and word (32-bit) accesses that are some times needed in the processing.

Each address increment is encoded with one 16-bit parameter register, treated as signed value. The above example shows address increment moving forward, but in many cases, for example coefficients for filtering, negative increments are used.

Note that the address increments together specify addressing pattern of a single pointer, and the data distribution controls how consecutive memory data from that pointer, up to 32N-bit, are distributed among the SIMD dimension of the vector register entry.

In case multiple VAGEN instructions are issued for the same agen resource (for example, A0), the last one takes effect, and the preceding ones with the same agen resource is ignored.

6.3.5.5.4 Vector Load

The VLD instruction specifies the data distribution, data type, base address, agen, and the vector register to load the data to. Format of the VLD instruction in assembly is:

VLD *type_distribution base* [*agen*], *vreg*

The *base* field points to a pair of parameter registers (even/odd pair only), for example base = P8 to refer to the P8:P9 pair. The vector data memory space is 20-bit, with the lower 16-bit encoded in the even parameter register (P8 in this case), the upper 4-bit in the odd parameter register (P9 in this case).

The destination vector register specified in VLD must be an even register. VLD_DINTRLV loads into an even/odd pair of registers. VLD with any other distribution option loads into one register.There are up to eight VLD instructions in a loop to potentially load up all 16 registers. No register can be the destination of more than one VLD instruction in the same loop.

As EVE is an N-way SIMD machine, loading of N consecutive data points from local memory is supported, with each data point being a byte, short, or long (32-bit) word, either signed or unsigned.

In addition to loading N data points, EVE supports a few other distribution options. EVE supports the following:

- NPT: N data points, one to each way of SIMD
- 1PT: broadcasting one data point to all
- CIRC2: circuiting between two data points, useful for Bayer image processing
- DS2: downsample by 2; every other data point (**not available for word-size input**)
- US2: upsample by 2; repeat each input data point twice
- DINTRLV: deinterleave inputs; load 2N data points into two registers, deinterleave data items so that even items go to the first register, odd items go to the second register. Supported only for byte or halfword per data point, **not word data type**. Two registers must be an even/odd pair, but only the first register (even) is encoded in assembly and in instruction binary.
- CUST_P8: custom distribution as provided by {P8..} in the parameter file
- CUST_P16: custom distribution as provided by {P16..} in the parameter file
- CUST_P24: custom distribution as provided by {P24..} in the parameter file
- EXP: load with expansion using predicate register V0
- NBITS: N bits, one bit to each way of SIMD, regardless of the data type (for example, when N = 16, a halfword is read). Signed/unsigned data type affects how this bit is to be sign-extended. Unsigned means 0 -> 0, 1 -> 1, and signed means 0 -> 0, 1 -> -1. This is useful for expanding load and other cases where boolean arrays are packed.

For CUST_Pi options, number of bits required to specify the address offset for each way of SIMD depends on N, the number of ways of SIMD. Arbitrary distribution of N data items (each a byte, halfword, or word) is supported. For N = {2 | 4 | 8 | 16}, 4 bits is used to encode each destination. For N = 32, 5 bits is used; and each 16-bit parameter register supplies three fields (leaving 1 bit unused).

- N = 2, 2 × 4 = 8 bits, one parameter register required
- N = 4, 4× 4 = 16 bits, one parameter register required
- N = 8, 8 × 4 = 32 bits, two parameter registers required
- N = 16, 16 × 4 = 64 bits, four parameter registers required
- N = 32, ceiling(32/3) = 11 parameter registers required

The DINTRLV option is supported so software can fully utilize the two function units for simple operation sequences, by processing 2N data points in parallel.

VLD instruction normally loads into one destination vector register. VLD_DINTRLV is the exception in that it loads into a pair of vector registers. Only even registers, V0, V2, ..., V14, can be referenced in the destination register field. In case of VLD_DINTRLV, the referenced register and the next register shall be the destination, for example V0 and V1.

The following data types are supported:

- B: signed byte
- BU: unsigned byte
- H: signed halfword (16-bit)
- HU: unsigned halfword
- W: signed word (32-bit)
- WU: unsigned word

Address offsets for each way of SIMD, as the function of distribution option and data type are shown in Table 6-353 and Figure 6-59, Figure 6-60, and Figure 6-61.

Table 6-353. VLD Data Distribution Options[1][2]

Distribution	vreg[r][0] gets	vreg[r][1] gets	vreg[r][2] gets	vreg[r][3] gets	vreg[r][4] gets	vreg[r][5] gets	vreg[r][6] gets	vreg[r][7] gets
NPT	data[0]	data[1]	data[2]	data[3]	data[4]	data[5]	data[6]	data[7]
1PT	data[0]	data[0]	data[0]	data[0]	data[0]	data[0]	data[0]	data[0]
CIRC2	data[0]	data[1]	data[0]	data[1]	data[0]	data[1]	data[0]	data[1]
DS2	data[0]	data[2]	data[4]	data[6]	data[8]	data[10]	data[12]	data[14]
US2	data[0]	data[0]	data[1]	data[1]	data[2]	data[2]	data[3]	data[3]
DINTRLV	vreg[r][0] = data[0], vreg[r+1][0] = data[1]	vreg[r][1] = data[2], vreg[r+1][1] = data[3]	vreg[r][2] = data[4], vreg[r+1][2] = data[5]	vreg[r][3] = data[6], vreg[r+1][3] = data[7]	vreg[r][4] = data[8], vreg[r+1][4] = data[9]	vreg[r][5] = data[10], vreg[r+1][5] = data[11]	vreg[r][6] = data[12], vreg[r+1][6] = data[13]	vreg[r][7] = data[14], vreg[r+1][7] = data[15]
CUST_Pi[3]	data[pf[0]]	data[pf[1]]	data[pf[2]]	data[pf[3]]	data[pf[4]]	data[pf[5]]	data[pf[6]]	data[pf[7]]

(1) Table “data” is type-cast to the appropriate signed/unsigned byte/halfword/word so that correct offsets are applied.

(2) This is a table of what data element gets loaded into each SIMD lane of a vector register, in the order lane 0 to lane 7. This does not reflect a debugger memory window hex display.

(3) pf[0], pf[1], ... pf[7] refer to bit fields in the indicated parameter registers. For example, for the 8-way SIMD EVE, it takes two parameter registers for the distribution information, so for CUST_P8 use: distrib_info = (P9 << 16) | P8, and assign (distrib_info >> (4 × i)) & 0xF to pf[i].

Loads are automatically predicated. Loads are always performed at the very first iteration, and then subsequently upon any change in the associated agen address pointer.

See Section 6.3.5.5.10 for address alignment requirement for various data type and distribution options.

LD_EXP is load with expansion, and is for reading a compacted (collated) array and expanding the elements to the original locations. It is the opposite of ST_COLLAT (collating store). Per-item address increment is implied by the load data type (1/2/4 bytes depending on byte/halfword/word type). The agen field is don't care. Per iteration, load pointer (VCOP_LD_PTR_i/VCOP_ST_PTR_j) is increment by number of non-zeroes in predicate register V2 times the data size. Agen is not used, and so the agen field is not required in the assembly code:

VLD *type*_**EXP** *base*, *vreg*

For example with this instruction, where V2 = {0, 0, 1, 0, 1, 1, 0, 0}, load_ptr = 0x100:

```
LDBU_EXP Pbase[A0], V1
```

There shall be, after this instruction:

```
V1 = {0, 0, mem[0x100], 0, mem[0x101], mem[0x102], 0, 0}, and load_ptr = 0x103
```

Load with expansion is restricted to be used in table lookup VLOOP. This is because the dependency (where the expanding load depends on predicated register data from another load) is more similar to table lookup, and thus it's feasible to achieve N data points per clock cycle inside table lookup pipeline. As in normal table lookup, no operation instructions are allowed in table lookup VLOOP.

Figure 6-59. Load Word Distribution Options

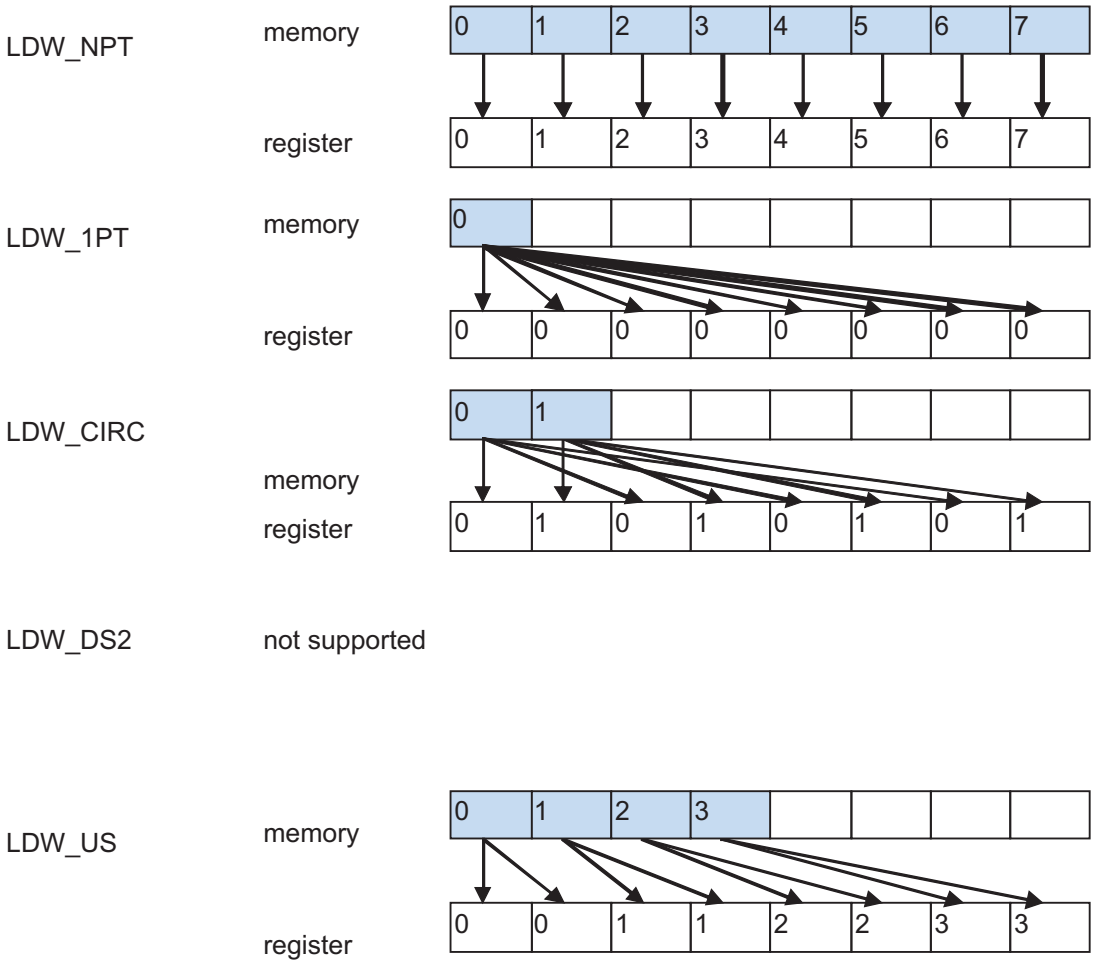


Figure 6-60. Load halfword Distribution Options

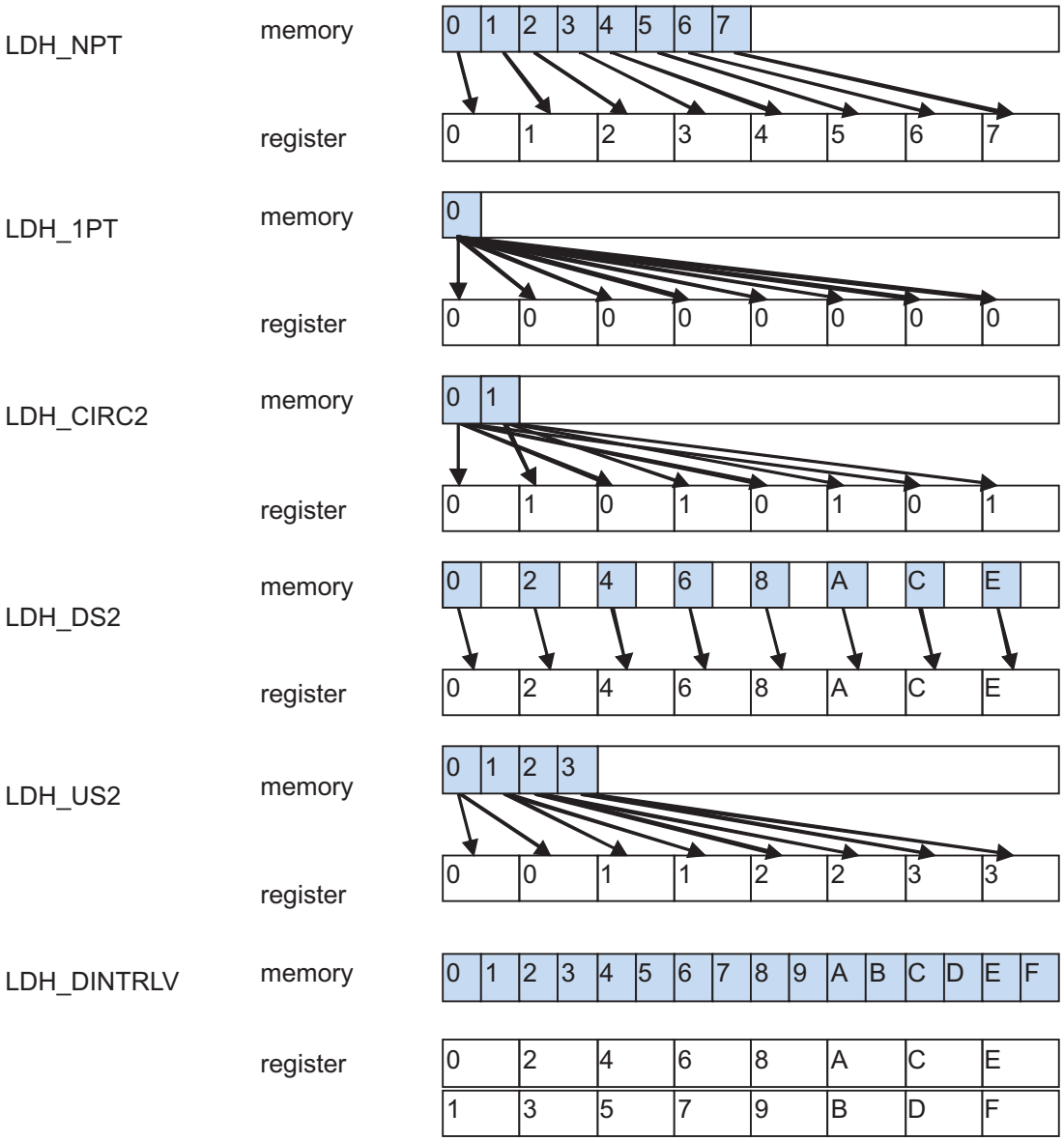
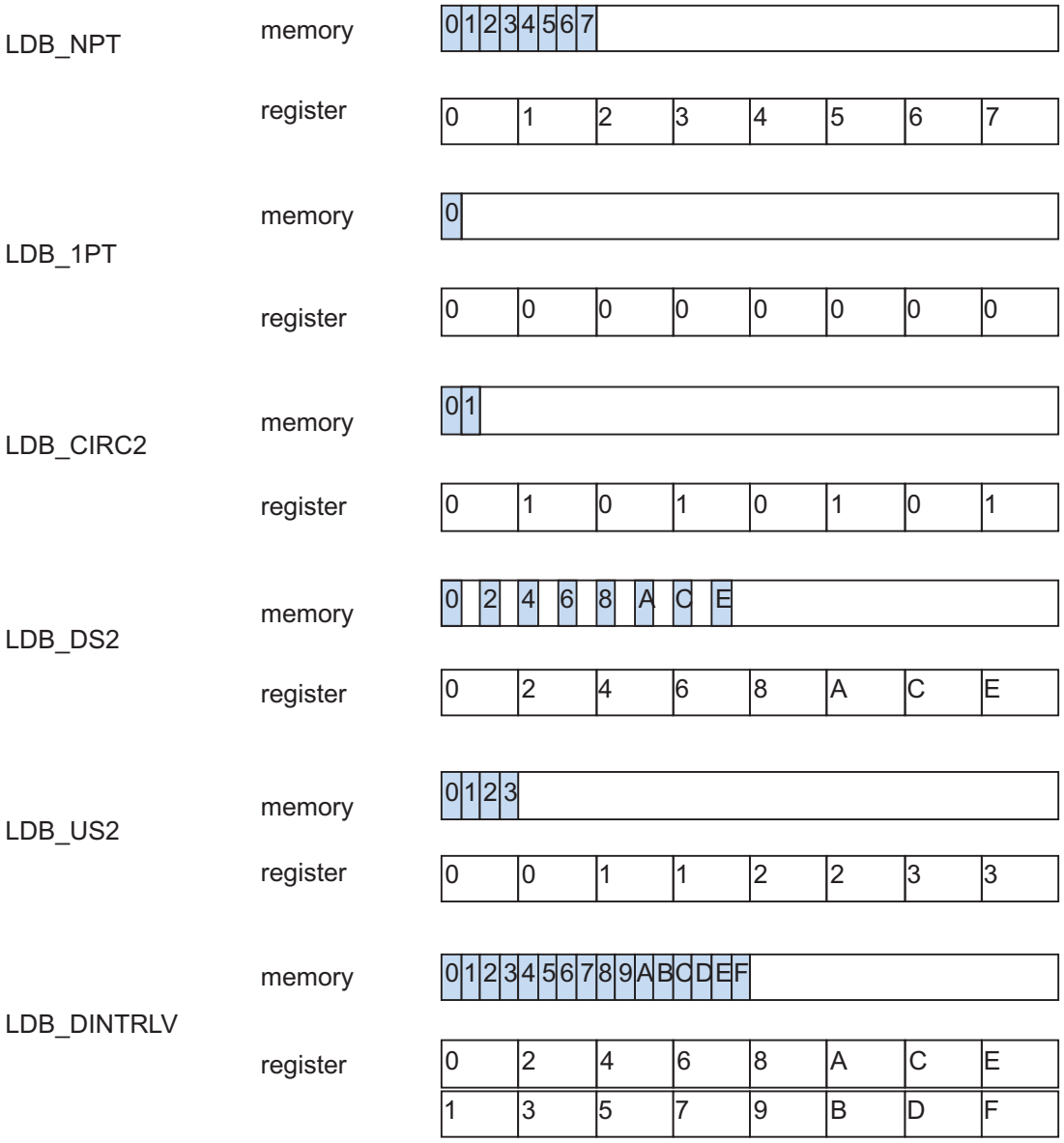


Figure 6-61. Load Byte Distribution Options



6.3.5.5.5 Vector Arithmetic/Logic Operations

Various arithmetic and logic operations are available in the compute command, which is indicated by starting the command with:

VLOOP COMP, *cmd_len*, *param_len*

Vector arithmetic/logic instructions have the following assembly formats, depending on whether it is 1-input-1-output, 2-input-1-output, 2-input-2-output, or 3-input-1-output, whether accumulator clearing is enabled, and whether rounding is enabled.

V<op_1i1o> src1, dst

V<op_2i1o> src1, src2, dst

V<op_2i1o> src1, src2, dst, RND: rnd_param

V<op_2i2o> src1/dst1, src2/dst2

V<op_2i2o> src1, src2/dst1, dst2

V<op_3i1o> src1, src2, src3, dst

V<op_3i1o> src1, src2, src3, dst, RND: rnd_param

All operations are 40-bit, except:

- Input to multiply (MPY, MADD, MSUB) is 17-bit data.
- BINLOG, BITC, BITDI, BITI, BITR are 32-bit instructions.
- BITPK, BITUNPK, SORT2, MIN, MAX are 33-bit instructions (to allow processing of up to 32-bit signed data)
- BITTR, bit transpose, bit width is limited to ways of SIMD, thus 8-bit data for 8-way SIMD.

The operations in Table 6-354 are supported. See Section 6.3.5.9 for details on each operation.

Table 6-354. VCOP Arithmetic/Logic Operations

Operation	#In- Out	#Bits	#Del	Syntax[1]	Note
VNOP		40		VNOP	
VADD	2-1	40		VADD src1, src2, dst	src1 + src2
VSUB	2-1	40		VSUB src1, src2, dst	src1 – src2
VABSDIF	2-1	40		VABSDIF src1, src2, dst	|src1 – src2|
VMPY	2-1	17/33	1	VMPY src1, src2, dst, RND:rnd_param	src1 * src2
VAND	2-1	40		VAND src1, src2, dst	src1 & src2
VOR	2-1	40		VOR src1, src2, dst	src1 | src2
VXOR	2-1	40		VXOR src1, src2, dst	src2 ^ src2
VMIN	2-1	33		VMIN src1, src2, dst	min(src1, src2)
VMAX	2-1	33		VMAX src1, src2, dst	max(src1, src2)
VANDN	2-1	40		VANDN src1, src2, dst	src1 & (~src2)
VSHF	2-1	40/6		VSHF src1, src2, dst	src1 << src2, or src1 >> (-src2)
VRND	2-1	40/5	1	VRND src1, src2, dst	(src1 + (1 << (src2-1)) >> src2
VCMPEQ	2-1	40		VCMPEQ src1, src2, dst	(src1 == src2) ? 1 : 0
VCMPGT	2-1	40		VCMPGT src1, src2, dst	(src1 > src2) ? 1 : 0
VCMPGE	2-1	40		VCMPGE src1, src2, dst	(src1 >= src2) ? 1 : 0
VBINLOG	1-1	32	1	VBINLOG src1, dst	approximate binary log
VBITC	1-1	32	1	VBITC src1, dst	count one bits
VNOT	1-1	40		VNOT src1, dst	~src1

(1) The parameters src1, src2, src3, dst, dst2 refer to the vector register entry for source or destination.

Table 6-354. VCOP Arithmetic/Logic Operations (continued)

Operation	#In- Out	#Bits	#Del	Syntax[1]	Note
VMADD	3-1	17/40	1 / 2	VMADD src1, src2, src3, dst, RND: rnd_param	src3 + src1 * src2
VMSUB	3-1	17/40	1 / 2	VMSUB src1, src2, src3, dst, RND: rnd_param	src3 – src1 * src2
VADD3	3-1	40	1	VADD3 src1, src2, src3, dst	src1 + src2 + src3
VSAD	3-1	40	1	VSAD src1, src2, src3, dst	src3 + abs(src1 – src2)
VSEL	3-1	40		VSEL src1, src2, src3, dst	src1 ? src2 : src3
VAND3	3-1	40	1	VAND3 src1, src2, src3, dst	src1 & src2 & src3
VOR3	3-1	40	1	VOR3 src1, src2, src3, dst	src1 | src2 | src3
VSHFOR	3-1	40/6	1	VSHFOR src1, src2, src3, dst	src3 | (src1 << src2) or src3 | (src1 >> -src2)
VSORT2	2-2	33		VSORT2 src1/dst1, src2/dst2	dst1 = min(src1, src2) dst2 = max(src1, src2)
VBITPK	2-1	33	1	VBITPK src1, src2, dst	compare, bit-pack, broadcast
VBITUNPK	2-1	40	1	VBITUNPK src1, src2, dst	bit unpack
VEXITNZ		40		VEXITNZ level, src1	exit loop at end of iteration when (src1 != 0)
VCMOV		40		VCMOV cond, src1, dst	conditional move
VBITR	1-1	32	1	VBITR src1, dst	bit reverse
VBITI	2-1	32	1	VBITI src1, src2, dst	bit interleave
VBITDI	1-2	32		VBITI src1, dst1, dst2	bit deinterleave
VABS	1-1	40		VABS src1, dst	abs(src1)
VADDH	2-1	40		VADDH src1, src2, dst	src1+ signext(src2[39:32])
VLMBD	3-1	40	1	VLMBD src1, src2, dst	left-most-bit-detect
VBITTR	1-1	NSIMD	1	VBITTR src1, dst	bit transpose
VSIGN	2-1	40		VSIGN src1, src2, dst	apply sign of src1 on src2
VADDSUB	2-2	40		VADDSUB src1/dst1, src2/dst2	dst1 = src1 + src2 dst2 = src1 – src2
VINTRLV	2-2	40		VINTRLV src1/dst1, src2/dst2	interleave
VDINTRLV	2-2	40		VDINTRLV src1/dst1, src2/dst2	deinterleave
VMINSETF	2-2	33		VMINSETF src1, src2/dst1, dst2	minimum and set flag
VMAXSETF	2-2	33		VMAXSETFsrc1, src2/dst1, dst2	maximum and set flag
VINTRLV2	2-2	40		VINTRLV2 src1/dst1, src2/dst2	interleave with 2-element frequency
VDINTRLV2	2-2	40		VDINTRLV2 src1/dst1, src2/dst2	deinterleave with 2-element frequency
VINTRLV4	2-2	40		VINTRLV2 src1/dst1, src2/dst2	interleave with 4-element frequency
VSHF16	1-2	33		VSHF16 src1, dst1, dst2	shift up 16 bits into 2 registers
VADIF3	3-1	40	1	VADIF3 src1, src2, src3, dst	add difference, dst = src1 – src2 + src3
VSWAP	3-2	40		VSWAP cond, src1/dst1, src2/dst2	conditional swap

Operations with two destinations must use two different destination registers, otherwise the outcome is undefined.

The accumulating register (being a source as well as the destination of an operation) must be src1 for 2-input-1-output operations, and must be src3 for 3-input-1-output operations.

The rounding parameter is needed for VMPY, VMADD and VMSUB instructions, and specifies an index to the parameter file, see Section 6.3.5.9.

EVE hardware executes up to 2 operations in parallel per clock cycle. Assembly program (by programmer or by compiler) contains the parallel bar notation to indicate if an instruction is to be executed by itself, or is to be executed in parallel with another instruction.

VMADD, VMSUB instruction have two delay slots for the multiplication input, and one delay slot for the addition/subtraction input.

VMPY, VRND, VBINLOG, VBITC, VADD3, VSAD, VAND3, VOR3, VSHFOR, VBITPK, VBITUNPK, VBITR, VBITI, VLMBD, VBITTR, VADIF3 instructions have one delay slot.

All other operations do not have delay slots.

The hardware detects and treats write/read dependency between any two sequential instructions, but not inside parallel executed instruction pairs. When necessary for correctness, hardware inserts idle cycles automatically. Still to achieve good performance, software must try to schedule the operations to avoid automatic idle cycles.

There is a forwarding path *within each* functional unit and *between* the two functional units to forward from destination to source 3 (accumulator input) of 3-input-1-output operations.

Register forwarding, dependency checking and automatic idle cycle insertion work across iterations as well, from end of one iteration to the beginning of the next iteration. For example, the FIR filtering kernel executes in one cycle per iteration with the destination-to-source3 dependency across iterations.

Table 6-355 shows delay slots and automatically inserted idle cycles.

Table 6-355. Example of Operation Delay Slots

Time	VMPY	VMADD	VSUB	VMSUB	VAND	VOR	VADD
0	V0*V1						
1	V0*V1 => V7	V2*V3	V2-V6 => V6				
2		V2*V3 =>p		V4*V5	V3 & V6 => V6		
3		V7 + p => V7		V4*V5 => p		wait for V7	
4				V7 - p => V7		wait for V7	
5						V1|V8 => V9	V6 + V7 => V6

Zero-delay slot operation executes in one cycle. VMPY, an one-delay slot opeartion takes two cycles to execute. VMADD and VMSUB, two-delay slot operations take 3 cycles for execution, but need its additional input only on the third cycle. The dependency between the two instructions (VMADD-VMSUB) on the additional operand does not introduce idle cycles, but VMSUB-VADD dependency adds 2 idle cycles to execution per iteration.

6.3.5.5.6 Vector Store

The VST instruction specifies the data distribution, data type, base address (index to parameter file), agen, the vector register to get the store from, loop level when stores are to occur, and rounding/saturation configuration. Format of VST is:

[*pred***] VST***type_ distribution_wr_loop vreg***, base[***agen***], RND_SAT:** *rnd_sat_param*

Storing of up to N data points from vector register to local memory is supported, with each data point being a byte, short, or long (32-bit) word, either signed or unsigned. Signed/unsigned information is used in performing saturation.

There are up to eight VST instructions in one loop, and each VST can be from the full vector register file, V0..V15. It is allowed to overlap the source registers, i.e., having more than one VST instructions storing out the same vector register. Also, in the extreme case of having all eight VST being interleaving stores, all 16 vector registers can be written out.

There is a further constraint on source registers for vector store. Only V0, V1, V2, and V3 can be stored without being the destination of any operation.

VCOP supports the following store patterns:

- NPT: For 8-way SIMD, 8 data points
- 1PT: write only the first SIMD unit
- DS2: downsample by 2; every-other SIMD units are written
- INTRLV: interleave outputs; store 2N data points from two registers interleaved. Two registers must be an {i, i+1} pair (i = 0..14), but only the first register (i) is encoded in assembly and in instruction binary. Byte and halfword types are supported, but not word type. It is NOT allowed to predicate interleaving stores. (If one must predicate interleaving store, use VINTRLV operation to interleave two sets of N data points, then use two predicated VST<type>_NPT.)
- OFFST_NP1: write out all N data points, in (N + 1) words = (4N + 4) bytes address increments (for transposition). Note that although address offset is word-aligned, storing to partial word, in case of byte and halfword types, are supported. In case of byte or halfword type, the 2 LSBs of address determines the byte or halfword within each word that VCOP writes data into.
- COLLAT: collating store, use predicate register to indicate store/not-store, up to N data items stored per cycle, agen is don't care, and hardware increments store address by the size implied by the data type. The agen field is not required in the assembly code:

[*pred***] VST***type_* **COLLAT***_wr_loop vreg***, base, RND_SAT:** *rnd_sat_param*

- SDDA: sequential data-driven addressing, taking M cycles to store M data points (predication can lead to less than N stores), store Vreg[i] to address base + agen + V0[i] Note that agen can be used to vary the pointer, in addition to V0. Only unsigned address offset in V0 is supported.
- PDDA: parallel data-driven addressing. Use this mode when address offsets in V0 is such that all lanes enabled for store go to separate memory banks. Hardware executes the store of up to N data points in one cycle. In case there is bank conflict, outcome is indefinite, and an error interrupt is generated.
- SKIP: store N data points into every other elements in memory. Allowed for byte and halfword types, not for word type.

The following data types are supported:

- B: signed byte
- BU: unsigned byte
- H: signed halfword (16-bit)
- HU: unsigned halfword
- W: signed word (32-bit)
- WU: unsigned word

Table 6-356. EVE ST Data Distribution Options for NWAY = 8[1][2]

Distribution	vreg[r][0] goes to	vreg[r][1] goes to	vreg[r][2] goes to	vreg[r][3] goes to	vreg[r][4] goes to	vreg[r][5] goes to	vreg[r][6] goes to	vreg[r][7] goes to
NPT	dptr[0]	dptr[1]	dptr[2]	dptr[3]	dptr[4]	dptr[5]	dptr[6]	dptr[7]
1PT	dptr[0]	n/a	n/a	n/a	n/a	n/a	n/a	n/a
DS2	dptr[0]	n/a	dptr[1]	n/a	dptr[2]	n/a	dptr[3]	n/a
INTRLV	dptr[0] = vreg[r][0], dptr[1] = vreg[r+1][0]	dptr[2] = vreg[r][1], dptr[3] = vreg[r+1][1]	dptr[4] = vreg[r][2], dptr[5] = vreg[r+1][2]	dptr[6] = vreg[r][3], dptr[7] = vreg[r+1][3]	dptr[8] = vreg[r][4], dptr [9] = vreg[r+1][4]	dptr[10] = vreg[r][5], dptr[11] = vreg[r+1][5]	dptr[12] = vreg[r][6], dptr[13] = vreg[r+1][6]	dptr[14] = vreg[r][7], dptr[15] = vreg[r+1][7]
OFFST_NP1	dptr[0]	dptr[9]	dptr[18]	dptr[27]	dptr[36]	dptr[45]	dptr[54]	dptr[63]
COLLAT	none or *dptr++	none or *dptr++	none or *dptr++	none or *dptr++	none or *dptr++	none or *dptr++	none or *dptr++	none or *dptr++
SDDA/PDDA	dptr[V0[0]]	dptr[V0[1]]	dptr[V0[2]]	dptr[V0[3]]	dptr[V0[4]]	dptr[V0[5]]	dptr[V0[6]]	dptr[V0[7]]
SKIP	dptr[0]	dptr[2]	dptr[4]	dptr[6]	dptr[8]	dptr[10]	dptr[12]	dptr[14]

(1) Table “dptr” is type-cast to the appropriate signed/unsigned byte/halfword/word so that correct offsets are applied. dptr = base + agen.
(2) Custom distribution is not supported for stores.

Stores are predicated to perform at the indicated loop level, with wr_loop selectable from:

- ALWS: write out on every i4 iteration
- LAST_I4: write out on the last i4 iteration, when i4 = lpend4
- LAST_I34: write out on the last i3 iteration, when i3 = lpend3 and i4 = lpend4
- LAST_I234: write out on the last i2 iteration, when i2 = lpend2, i3 = lpend3, and i4 = lpend4

Rounding and saturation arithmetic is available in the store pipeline to provide rounding and saturation without taking additional cycles.

Rounding and saturation parameters are provided in the parameter file to allow easy adaptation of EVE functions to different rounding and saturation configurations. The rnd_sat_param field in the instruction points to a parameter (instruction encoding limits this field to 5 bits, thus restricting it to P0..P31) that contains the following fields shown in Figure 6-62.

Figure 6-62. VST Rounding and Saturation Parameters

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
sat_mode: 0 = NO_SAT 1 = SYMM 2 = ASYMM 3 = 4PARAM 4 = SYMM32 5 = ASYMM32			sat_bound_param: index to param that specifies saturation bound						rnd_mode: 0 = no rounding 1 = round 2 = truncate		rnd_shift: number of bits to round/shift down				

The following saturation modes are supported:

- NO_SAT: no saturation performed
- SYMM: signed symmetrical saturation [-bound, bound] (for unsigned store, [0, bound])
- ASYMM: signed asymmetrical saturation [-bound-1, bound] (for unsigned store, [0, bound]), useful for fixed bit width. For example, when bound = 1023, saturate to [-1024, 1023]
- 4PARAM: use 4 parameter registers to specify sat_high_cmp, sat_high_set, sat_low_cmp, sat_low_set.
- SYMM32: same as SYMM, except using a pair of parameters to specify a 32-bit bound
- ASYMM32: same as ASYMM, except using a pair of parameters to specify a 32-bit bound

Signed/unsigned designation is on the VST type, and signed/unsigned designation affects whether the bound parameter(s) are interpreted as signed or unsigned. The comparison is carried out between signed 40-bit register and such interpreted bounds.

When no rounding and saturation feature is needed, specifying RND_SAT: P0 is working, since constant 0 in these fields represents doing nothing.

Conditional store is specified by the pred field. Omission means no predication. Pred = V1, V2, V3 indicates that register V1, V2, V3 is used to predicate the store of each way of SIMD; store is executed when the register is not zero.

With the COLLAT distribution option, all SIMD ways are examined for conditional store. Truth in the predication register leads to the data item being stored and incrementing of the data pointer by size of the store type (1/2/4 bytes). False in the predication register leads to doing nothing. Thus, stored data items are compacted without leaving any gaps for the blocked data items.

With other distribution options, conditional store behaves differently. Agen is incremented as in unconditional stores. Truth in the predication register leads to the data item being stored, and false in the predication register leads to the data item being skipped. Thus, it is possible for the stored data items to contain gaps among them from the blocked data items.

The store stage hardware has separate resource for each VCOP data memory region (IBUFL, IBUFH, WBUF). Sequential data-driven store is executed in X cycles, X being the number of data items enabled in predication (or when predication is not used, N). The other store modes are executed in a single cycle per store (note that collating store executes in one cycle as well). All memory regions are carried out in parallel, but there is also potential memory contention with the load stage. There is write buffering in the store stage to reduce load/store contention to some degree. See Section 6.3.5.6 for load/store buffering details.

6.3.5.5.7 Table Lookup Operation

Table lookup operation is indicated by starting the vector command with

VLOOP TLU, *CL#***:***cmd_len***,** *PL#***:** *param_len*

The table lookup operation uses a subset of the available resources and instructions:

- VAGEN (address generator): fixed A0 for data, A1 for table pointer, A2 for output.
- VLD (load): one for loading data into V2.
- VTLD (table load): for performing the lookup, using V2 as indices to load table entry into V0.
- VST (store): for storing outcome in V0.
- Full set of 4 levels for looping.

The table pointer is allowed to move via agen, to allow loop-variable-dependent manipulation of table base. For example, software might process three color components (RGB or YUV), each having a different table, in the same command, and use the outer loop variables to index among the color components and the corresponding tables.

Data is loaded with normal load into V2, for example:

```
VLDBU_1PT      data_base[A0], V2
```

The data load using VLD has access to a subset of load distribution options: {NPT, 1PT, DS2, US2}.

- 1PT works for single table configuration
- NPT works for all other cases of normal table lookup
- DS2 and US2 provides additional flexibility to downsample/upsample data

A special table load, VTLD, is used for the lookup:

VTLD*type_m* **TBL_** *n***PT** *tbl_base*[*tbl_agen*][**V2**]**, V0, RND_SAT:***rnd_sat*

Normally lookups only fetch one item per lookup. Multiple items are useful for bilinear and bicubic interpolations. The _*m*TBL field in VTLD specifies the number of parallel tables and the _*n*PT field in VTLD specifies the number of data items per lookup table.

There are constraints on num_par_tbl, num_data_per_lu, and the table data size:

- num_par_tbl × num_data_per_lu <= N (ways of SIMD)
- num_data_per_lu = {1 | 2 | 4} regardless of N (ways of SIMD)

Constraints for 8-way SIMD architecture are shown in Table 6-357.

Table 6-357. Lookup Constraints for 8-Way SIMD

Table type	Num items per lookup, num_data_per_lu	Number of parallel tables, num_par_tbl			
		1	2	4	8
Byte	1	√	√	√	√
	2	√	√	√	
	4	√	√		
	8	√			
Half word	1	√	√	√	√
	2	√	√	√	
	4	√	√		
	8	√			
Word	1	√	√	√	√
	2	√	√	√	
	4	√	√		
	8	√			

Results are stored back to memory with a normal data store as in compute loop with simplification:

VST*type_distribution*_**ALWS V0,** *base*[*agen*]**, RND_SAT:** *rnd_sat_param*

This VST has access to a subset of store distribution optons: {NPT | SKIP}.

Number of data points that are stored per i4 iteration is num_data_per_lu × num_par_tbl from the table load, and this is an implicit predication.

For example, when there is table load VTLDHU_1TBL_2PT, the store VSTHU_NPT is not storing N points, but just 2 point.

In forming the indices for lookup, data are read in, rounded and saturated. Rounding and saturation parameters are specified in VTLD's RND_SAT: rnd_sat field. This field is 5-bit (like VST), so only P0..P31 is allowed. The same set of round/saturate parameters as the normal store, VST, are used. VTLD's RND_SAT is for round/saturate the data to form table lookup index. VST also contains a RND_SAT field, which is used to round/saturate looked up table entries before storing back to data memory.

Various other table lookup parameters are extracted from the instructions:

- Input data type is from VLD*type*, and is limited to unsigned types.
- Table entry type is from VTLD*type*, and can be signed or unsigned although the signed/ unsigned information is not used.
- Output type is from VST*type*, can be signed or unsigned, and can be different from the table entry type. Signed versus unsigned information is used for saturation.
- Input data distribution is from VLD V2, with SIMD units 0..num_par_tbl-1 used for the lookup.
- Output distribution option is either NPT, storing consecutively (which may not be N points), or SKIP, storing to every other memory location.

- Table entries are read into the vector register SIMD lanes by iterating on num_par_tbl in the outer loop, and num_data_per_lu in the inner loop. For example, with num_par_tbl = 2 and num_data_per_lu = 4, with VSTBU_NPT we have:

```
V0[0] = tbl_base[data0],      V0[1] = tbl_base[data0 + 1],
V0[2] = tbl_base[data0+2],    V0[3] = tbl_base[data0 + 3],
V0[4] = tbl_base[data1],      V0[5] = tbl_base[data1 + 1],
V0[6] = tbl_base[data1+2],    V0[7] = tbl_base[data1 + 3]
```

Table organization in memory depends on table type and num_par_tbl. The N physical banks of 32-bit memory is partitioned into num_par_tbl logical banks, and each parallel table is organized inside its own bank.

For example, when NWAY = 8, you can have 1, 2, 4, or 8 parallel tables. Data organization for these and byte, halfword, and word type entries is shown in Figure 6-63.

A few examples of the table lookup operation follow. The NPT distribution for load and store works for normal lookup scenarios.

```
; single table, byte data, short table/output

   VLOOP            TLU, #cmd_len, #param_len
   VAGEN            A0, ...
   VAGEN            A1, ...
   VAGEN            A2, ...
   VLDBU_1PT        data_base[A0], V2                          ; load data to V0
   VTLDHU_1TBL_1PT  table_base[A1][V2], V0, RND_SAT: rnd_sat   ; look up
   VSTH_NPT_ALWS    V0, outp_base[A2], RND_SAT: rnd_sat_st     ; store outcome
; 4 parallel tables, 2 items per table, short data, byte table/output

   VLOOP            TLU, #cmd_len, #param_len
   VAGEN            A0, ...
   VAGEN            A1, ...
   VAGEN            A2, ...
   VLDHU_NPT        data_base[A0], V2                          ; load data to V0
   VTLDBU_4TBL_2PT  table_base[A1][V2], V0, RND_SAT: rnd_sat   ; look up
   VSTB_NPT_ALWS    V0, outp_base[A2], RND_SAT: rnd_sat_st     ; store outcome
```

Load with expansion can only be used in a table lookup loop. An example follows.

```
; load with expansion, short input data, unsigned byte flags, short expanded output

   VLOOP            TLU, #cmd_len, #param_len
   VAGEN            A0, ...
   VAGEN            A2, ...
   VLDBU_NPT        flag_base[A0], V2                          ; load flags to V2
   VLDH_EXP         input_base, V0                             ; load with expansion
   VSTH_NPT_ALWS    V0, outp_base[A2], RND_SAT: rnd_sat_st     ; store outcome
```

For example, in a particular iteration, suppose there are flags V2 = {0, 0, 1, 0, 1, 1, 0, 0}, and the expanding load pointer = 0x100 before the expanding load. After the expanding load:

```
V0 = {0, 0, mem[0x100], 0, mem[0x102], mem[0x104], 0, 0}
```

The pointer is advanced to 0x106, incremented by 3 (number of nonzero flags) times the size of data type (2 bytes for halfword).

Optionally, predicated store can be used while writing the expanded outcome array, leaving flag==0 data points unaltered.

```
   VLOOP            TLU, #cmd_len, #param_len
   VAGEN            A0, ...
   VAGEN            A2, ...
   VLDBU_NPT        flag_base[A0], V2                        ; load flags to V2
   VLDH_EXP         input_base, V0                           ; load with expansion
[V2] VSTH_NPT_ALWS  V0, outp_base[A2], RND_SAT: rnd_sat_st   ; store where V2 != 0
```

Performance of load with expansion is NWAY expanded data points per cycle when there is no memory contention, by pipelining the flag load, data load, and output store.

Figure 6-63. Lookup Table Organization for Various Entry Size and Parallel Tables (NWAY = 8)

Byte-size single table

a[0], a[1], ... a[31] a[32], a[33], ... a[63] ...

Byte-size 2 parallel tables

a[0], a[1], ... a[15] a[16], a[17], ... a[31] ...	b[0], b[1], ... b[15] b[16], b[17], ... b[31] ...

Byte-size 4 parallel tables

a[0], a[1], ... a[7] a[8], a[9], ... a[15] ...	b[0], a[1], ... b[7] b[8], b[9], ... b[15] ...	c[0], c[1], ... c[7] c[8], c[9], ... c[15] ...	d[0], d[1], ... d[7] d[8], d[9], ... d[15] ...

Byte-size 8 parallel tables

a[0]... a[3] a[4]... a[7] ...	b[0]... b[3] b[4]... b[7] ...	c[0]... c[3] c[4]... c[7] ...	d[0]... d[3] d[4]... d[7] ...	e[0]... e[3] e[4]... e[7] ...	f[0]... f[3] f[4]... f[7] ...	g[0]... g[3] g[4]... g[7] ...	h[0]... h[3] h[4]... h[7] ...

Half-word-size single table

a[0], a[1], ... a[15] a[16], a[17], ... a[31] ...

Half-word-size 2 parallel tables

a[0], a[1], ... a[7] a[8], a[9], ... a[15] ...	b[0], b[1], ... b[7] b[8], b[9], ... b[15] ...

Half-word-size 4 parallel tables

a[0], a[1], ... a[3] a[4], a[5], ... a[7]	b[0], b[1], ... b[3] b[4], b[5], ... b[7]	c[0], c[1], ... c[3] c[4], c[5], ... c[7]	d[0], d[1], ... d[3] d[4], d[5], ... d[7]

Half-word-size 8 parallel tables

a[0]... a[1] a[2]... a[3] ...	b[0]... b[1] b[2]... b[3] ...	c[0]... c[1] c[2]... c[3] ...	d[0]... d[1] d[2]... d[3] ...	e[0]... e[1] e[2]... e[3] ...	f[0]... f[1] f[2]... f[3] ...	g[0]... g[1] g[2]... g[3] ...	h[0]... h[1] h[2]... h[3] ...

Word-size single table

a[0], a[1], ... a[7] a[8], a[9], ... a[15] ...

Word-size 2 parallel tables

a[0], a[1], ... a[3] a[4], a[5], ... a[7] ...	b[0], b[1], ... b[3] b[4], b[5], ... b[7] ...

Word-size 4 parallel tables

a[0]... a[1] a[2]... a[3] ...	b[0]... b[1] b[2]... b[3] ...	c[0]... c[1] c[2]... c[3] ...	d[0]... d[1] d[2]... d[3] ...

Word-size 8 parallel tables

a[0] a[1] ...	b[0] b[1] ...	c[0] c[1] ...	d[0] d[1] ...	e[0] e[1] ...	f[0] f[1] ...	g[0] g[1] ...	h[0] h[1] ...

6.3.5.5.8 Histogram Operation

Histogram functionality includes normal histogram, in which the addressed bin entry is incremented by 1, and weighted histogram, in which the addressed bin entry is incremented by an element in the weight array input.

Histogram operation is indicated by starting the vector command with:

VLOOP HIST,*CL#***:***cmd_len* **,** *PL#***:** *param_len*

The histogram operation uses a subset of the available resources and instructions:

- VAGEN (address generator): fixed A0 for data, optional A1 for histogram input/output pointer, A2 for weights
- VINIT (register initialization) for normal histogram, providing a fixed increment value (normally 1)
- VLD (load): one load for data, and in the case of weighted histogram, one additional load for weights
- VHLD (histogram load): load histogram bins
- VADD (add) to increment histogram bins
- VHST (histogram store): store histogram bins back to memory
- Full set of 4 levels for looping

For normal histogram, the increment value is initialized via a VINIT instruction on V4:

```
VINITHU_ONCE   P1, V4
```

Data is loaded with normal load into V2, for example,

```
VLDBU_NPT      data_base[A0], V2
```

Distribution options of VLD are {1PT, NPT, DS2, US2}. 1PT works for single histogram configuration, NPT works for 2, 4, or 8 parallel histograms.

Optional weights (for weighted histogram) are loaded with another normal load into V4, for example,

```
VLDBU_NPT      weight_base[A1], V4
```

A special histogram load, VHLD, is used to read in the bins:

VHLD*type_m***HIST** *hist_base*[*hist_agen*][**V2**]**, V0, RND_SAT:** *rnd_sat*

The bins are then incremented with normal VADD.

```
VADD  V0, V4, V0
```

Results are stored back to the histogram memory with a special histogram store, VHST:

VHST*type_m***HIST V0,** *hist_base*[*hist_agen*][**V2**]

Base and agen are supposed to be consistent between VHLD and VHST; otherwise, the outcome is undetermined.

Both signed and unsigned types are supported in data load, weight load, histogram load/store, to support unsigned histogram as well as weighted histogram with positive and negative weights.

Parameters for histogram are extracted from the instructions:

- Input data type is from VLD V2
- Input data distribution is from VLD V2, with SIMD units 0..num_par_hist-1 used for indexing histogram. NPT, 1PT, DS2, US2 distribution options are supported.
- Optional weight type is from VLD V4.
- Weight distribution behaves like data distribution, i.e., SIMD units 0..num_par_hist-1 used for incrementing parallel histogram bins
- Histogram data type is from VHLD and VHST, and the two instructions must be consistent in the data type, otherwise the outcome is indeterminate.

- rnd_sat in VHLD specifies round/truncate mode and how many bits to round off data before indexing the histogram, and also saturation mode and bound. This field is 5-bit (like VST and VTLD), so only P0..P31 is allowed.

The histogram command basically takes the data array, and processes num_par_hist data points at a time by assuming that many sets of parallel histogram bin storage. Each data value is rounded, saturated to generate a bin index. The address bin storage elements (which can be byte, halfword, or word) is then incremented by 1 or the weight, saturated to min/max values of that type, signed/unsigned byte/halfword/word:

- For unsigned byte, bin data is saturated to [0, 255].
- For signed byte, bin data is saturated to [-128, 127].
- For unsigned halfword, bin data is saturated to [0, 65535].
- For signed halfword, bin data is saturated to [-32768, 32767].
- For unsigned word, bin data is saturated to [0, 0xFFFF FFFF].
- For signed word, bin data is saturated to [-0x8000 0000, 0x7FFF FFFF].

There are restrictions on the weight data type (as indicated on the VLD into V4):

- Its signed/unsigned designation must agree with the histogram bin (as indicated in VHLD, VHST).
- Its size is limited to byte and halfword, and must be no bigger size then the histogram bin. Thus, for byte sized bins, only byte-sized weights are possible. For halfword or word sized bins, byte or halfword sized weights are possible.

C representation of histogram processing for a single histogram is as follows.

```
for (i=0; i<num_data; i++)
{
    data = data_base[i];
    bin = saturate(round(data));
    hist[bin]++;
}
```

Weighted histogram has the following C representation:

```
for (i=0; i<num_data; i++)
{
    data = data_base[i];
    bin = saturate(round(data));
    hist[bin] += weight[i];
}
```

Histogram organization in memory depends on histogram type and num_par_hist. The N physical banks of 32-bit memory is partitioned into num_par_hist logical banks, and each parallel histogram is organized inside its own bank, as with parallel lookup tables (see Figure 6-63).

The histogram command does not pre-initialize the histogram bin storage. The histogram command also does not sum up the parallel sets of storage into a single set; this is performed as a post-processing step.

Histogram operation takes 2M/P cycles plus some per-command overhead to process M data items with P parallel histograms (P = 1, 2, 4, or 8). As P parallel histograms takes P times the storage of a single histogram, it is a tradeoff of memory size and performance.

An example of histogram operation is:

```
; single histogram, byte data, short histogram, round down 2 bits, 33 bins
; rounding mode, shift, saturation bound coded in P7 using scalar instructions


   VLOOP           HIST, CL#: cmd_len, PL#: param_len
   VINITHU_ONCE    P1, V4                          ; initialize V4 = 1
   VLDBU_NPT       data_base[A0], V2               ; load data to V2
   VHLDHU_1HIST    hist_base[A1][V2], V0, RND_SAT: P7 ; load hist entry into V0
   VADD            V0, V4, V0                      ; increment entry
   VHSTH_1HIST     V0, hist_base[A1][V2]           ; store histogram entry
```

An example of weighted histogram operation is:

```
; single weighted histogram, byte data, byte weight, short histogram,
; round down 2 bits, 33 bins
; rounding mode, shift, saturation bound coded in P7 using scalar instructions


   VLOOP            HIST, CL#: cmd_len, PL#: param_len
   VLDBU_NPT        data_base[A0], V2                ; load data to V2
   VLDBU_NPT        weight_base[A2], V4              ; load weights to V4
   VHLDHU_1HIST     hist_base[A1][V2], V0, RND_SAT: P7 ; load hist entry into V0
   VADD             V0, V4, V0                       ; increment entry by weights
   VHSTH_1HIST      V0, hist_base[A1][V2]            ; store histogram entry
```

Table lookup and histogram loops are limited to use designated vector register and address generator resources as shown in Table 6-358.

Table 6-358. Table Lookup and Histogram Hardware Resources

Loop type	Input/outputstream	Address generator	Vector register
table lookup	data	A0	V2
	table	A1	V0
	output	A2	V0
load with expansion	flag	A0	V2
	data	n/a	V0
	output	A2	V0
histogram	data	A0	V2
	histogram	A1	V0
	weight	A2	V4

6.3.5.5.9 Circular Buffer Addressing Support

In certain applications, it is desirable to maintain circular buffers for input data and intermediate results. Thus, circular buffer addressing is supported in VCOP.

- Support circular buffer in VLD and VST instructions in compute loops, table lookup loops, and histogram loops. No circular buffer in VTLD (table load) and VHLD/VHST (histogram bin load/store).
- Support is in pointer increment arithmetic, so no load or store in any iteration should straddle the circular buffer boundary.
- Support only 2's power buffer size from 1Kbytes, aligned to the size. For example, 1KB circular buffer can only start/end at 1KB alignment of the memory map.
- Support is in all memory regions.

Circular buffer addressing option is encoded in the circ_buf field taking up bits 23..20 of base address:

- 0: no circular buffer addressing
- 1: circular buffer size 1K bytes
- 2: circular buffer size 2K bytes
- 3: circular buffer size 4K bytes
- 4: circular buffer size 8K bytes
- 5: circular buffer size 16K bytes
- 6: circular buffer size 32K bytes
- 7~15: reserved

The base address does not need to be aligned to the circular buffer size, in the abive example's case, 0x400. To avoid any single load/store straddle the circular buffer boundary, the base address and the pointer increments must all be aligned to the access size of each iteration.

For example, when loading N data points of halfword data, where N = 8. Access size = 16, so the base address and any pointer increment must be aligned to multiples of 16.

Hardware adjusts the memory address for circular buffering by this process:

```
circ_buf = (base >> 20) & 0xF;            // extract circ_buf
mask = (0x200 << circ_buf) - 1;           // mask = size of buffer - 1
cbuf = (base & 0xFFFFF) & ~mask;          // find base addr of buffer
addr = cbuf + (base + agen) & mask;       // add offset from base of buffer
```

For example, base = 0x10420, agen = 0xFF0.

1. Extract circ_buf field by cir_buf = (base >> 20) & 0xF = 1, indicating 1KB circular buffer size.
2. Set mask = (0x200 << 1) – 1 = 0x3FF, forming the bit mask that defines the circular addressing bits versus linear, non-circular bits.
3. Get rid of the circ_buf field by ANDing base with 0xFFFFF, then zero out the circular addressing bits by ANDing with the complement of mask, to get cbuf = 0x400, so the circular buffer is 0x400 ~ 0x7FF.
4. Next, add the address generator value to the base address, and AND with mask to find offset from the cbuf base. (base + agen) & mask = (0x10420 + 0xFF0) & 0x3FF = 0x11410 & 0x3FF = 0x10.
5. Finally, add the offset 0x10 to the circular buffer base 0x400 to get the final address, 0x410.

The agen is allowed to go over the circular buffer many times over to wrap back inside, and that agen can even go backward to wrap from the beginning to the end of the buffer.

6.3.5.5.10 Load/Store Address Alignment Constraints

Load/store pointer alignment constraints are summarized in Table 6-359, for data load/store as well as table and histogram load/store, for various data types.

Table 6-359. Load and Store Address Alignment Constraints

Type of load/store and distribution	Byte type	halfword type	Word type
VLD_NPT	any	any	word-aligned
VLD_1PT	any	any	any
VLD_CIRC2	any	any	any
VLD_DS2	any	halfword-aligned	not supported
VLD_US2	any	any	any
VLD_DINTRLV	any	word-aligned	not supported
VLD_CUST_Pi	any	any	word-aligned
VLD_EXP	any	halfword-aligned	word-aligned
VST_NPT	any	halfword-aligned	word-aligned
VST_1PT	any	halfword-aligned	word-aligned
VST_DS2	any	halfword-aligned	word-aligned
VST_INTRLV	any	word-aligned	not supported
VST_OFFST_NP1	any	halfword-aligned	word-aligned
VST_COLLAT	any	halfword-aligned	word-aligned
VST_DDA	any	halfword-aligned	word-aligned
VST_SKIP	any	halfword-aligned	not supported
VTLD_1TBL_1PT	any	halfword-aligned	word-aligned
VTLD_mTBL_nPT (other than 1TBL_1PT)	32-byte-aligned	32-byte-aligned	32-byte-aligned
VHLD_mHIST	32-byte-aligned	32-byte-aligned	32-byte-aligned
VHST_mHIST	32-byte-aligned	32-byte-aligned	32-byte-aligned

The rationale is:

- For data load, in general any alignment is supported. Exception: When software loads 32 bytes, the pointer needs to be word-aligned.
- For data store, in general any alignment is supported for byte, halfword alignment for halfword, and word-alignment for word. Exception: When 32 or 64 bytes is stored, the value needs to be word-aligned. In case of LDH_DS2, the pointer must be 16-bit aligned.
- For table load with one table, 1-point lookup, the table base pointer can be of any alignment for byte, halfword alignment for halfword, and word-alignment for word.
- For table load with other configurations, only a 32-byte aligned table base is supported.
- Table store is really data store with 1, 2, 4, or 8 data points. The same alignment is supported as a data store, i.e., any alignment for byte, halfword alignment for halfword, and word-alignment for word.
- For histogram load/store, the histogram base pointer must be 32-byte aligned.

6.3.5.6 Load/Store Buffer and Scheduling

There is load data buffer associated with each VLD instruction in the loop to reduce memory reads and resulting contention and slow-down.

- Size of the buffer is 32 bytes, and is filled with 8 32-bit words on word alignment.
- When the requested data is not contained in the buffer (completely or partially), VCOP reads another 8 32-bit words from the first requested word.
- There is no sharing of data among separate load data buffers.

There is one common store data buffer, basically a shadow copy of all vector registers, for all the VST instructions in the loop to skew memory write timing to reduce memory contentions and resulting slow-down.

- The store data buffer is 16 × 8 × 40-bit in size.
- Vector registers are copied to the store data buffer at the end of i4 iteration, when there is any store to be performed for that iteration. In case stores are not all completed then, load and execution stages are stalled, until all stores are completed.
- Write data from multiple iterations of the same store instruction are not combined. For example, STB_NPT_ALWS causes a memory store of 8 bytes in each iteration.
- Write data from multiple stores of the loop, same iteration, are not combined.

Memory loads and stores are arbitrated for each memory port. There are three memory ports:

- WBUF: Dedicated to WBUF
- IBUFL: For IBULA and IBUFLB
- IBUFH: For IBUFHA and IBUFHB

Load store priorities are: force store > load > store. Force store is issued when the load and operation stages are stalled to free up the store stage so the hardware can advance to the next iteration. Other than force store, load has priority over normal store.

Example 6-9. 3-Tap Horizontal Filtering, Byte Type

```
      VLDB_DINTRLV  Pdata[a0], V0                ; data, starts from 0x10000 (IBUFLA)
      VLDB_1PT      Pcoef[a1], V2                ; coef, starts from 0x0 (WBUF)
      VMADD         V0, V2, V4, V4
||    VMADD         V1, V2, V5, V5
      VSTB_INTRLV   V4, Pout[a2], RND_SAT: rnd_sat_param ; output, start from 0x10400 (IBUFLA)
```

Table 6-360. Load and Store Buffering Example, Byte-Type Horizontal Filter

Cycle	LD stage	OP stage	ST stage
0	iter 0 LD0 need 0x10000..0x1000F, **read IBUFLA** LD0_buf = {0x10000..0x1001F} LD2 need 0x0, **read WBUF** LD2_buf = {0x0..0x1F}		
1	iter 1 LD0 need 0x10001..0x10010, from LD0_buf LD2 need 0x1, from LD2_buf	iter 0 MADD || MADD	
2	iter 2 LD0 need 0x10002..0x10011, from LD0_buf LD2 need 0x2, from LD2_buf	iter 1 MADD || MADD	iter 0
3	iter 3 LD0 need 0x10010..0x1001F, from LD0_buf LD2 need 0x0, from LD2_buf	iter 2 MADD || MADD	iter 1
4	iter 4 LD0 need 0x10011..0x10020, **read IBUFLA** LD0_buf = {0x10010..0x1002F} LD2 need 0x1, from LD2_buf	iter 3 MADD || MADD	iter 2 ST0 store 0x10400..0x1040F queue the store

Table 6-360. Load and Store Buffering Example, Byte-Type Horizontal Filter (continued)

Cycle	LD stage	OP stage	ST stage
5	iter 5 LD0 need 0x10012..0x10021, from LD0_buf LD2 need 0x2, from LD2_buf	iter 4 MADD || MADD	iter 3 store IBUFLA
6	iter 6 LD0 need 0x10020..0x1002F, from LD0_buf LD2 need 0x0, from LD2_buf	iter 5 MADD || MADD	iter 4
7	iter 7 LD0 need 0x10021..0x10030, **read IBUFLA** LD0_buf = {0x10020..0x1003F} LD2 need 0x1, from LD2_buf	iter 6 MADD || MADD	iter 5 ST0 store 0x10410..0x1041F queue the store

Here LDB_DINTRLV is used to read 16 data points, to operate on 16 outputs in parallel. The hardware issues memory reads for iteration 0, 4, 7, etc, is able to supply data for subsequent hold data read in for subsequent LD; on average there is 1 read every 3 cycles. For coefficient, since only 3 bytes is used, just one read in iteration 0 is sufficient to supply data for the whole loop. Output is produced in iteration 2, 5, etc, and the store buffer delays the write until there is no read traction, avoiding read/write contention and slowdown.

Due to memory read/write timing, delayed slot in operation, and round/saturate feature in the store stage, the skew between load, operation, and store stages is more than what is shown in Table 6-360, but for calculating steady-state performance, simplified analysis like this is sufficient.

Example 6-10. Horizontal Filtering, Short Type

```
      VLDH_NPT      Pdata0[a0], V0              ; data, start from 0x10000 (IBUFLA)
      VLDH_NPT      Pdata1[a0], V2              ; data next row, start from 0x10080 (IBUFLA)
      VLDH_1PT      Pcoef[a1], V4               ; coef, start from 0x0 (WBUF)
      VMADD         V0, V4, V6, V6
||    VMADD         V2, V4, V7, V7
      VSTH_NPT      V6, Pout0[a2], RND_SAT: rnd_sat_param ; start from 0x10400 (IBUFLA)
      VSTH_NPT      V7, Pout1[a2], RND_SAT: rnd_sat_param ; start from 0x10480 (IBUFLA)
```

Software cannot issue VLDH_DINTRLV on odd halfwords, so use two LDH_NPT.

Table 6-361. Load and Store Buffering Example, Short-Type Horizontal Filter

Cycle	LD stage	OP stage	ST stage
0	iter 0 LD0 need 0x10000..0x1000F, **read IBUFLA** LD0_buf = {0x10000..0x1001F} LD1 need 0x10080 .. 0x1008F, **stalled** LD2 need 0..1, **read WBUF** LD2_buf = {0x0..0x1F}		
1	iter 0 LD1 **read IBUFLA** LD1_buf = {0x10080..0x1009F}		
2	iter 1 LD0 need 0x10002..0x10011, from LD0_buf LD1 need 0x10082..0x10091, from LD1_buf LD2 need 2..3, from LD2_buf	iter 0 MADD || MADD	
3	iter 2 LD0 need 0x10004..0x10013, from LD0_buf LD1 need 0x10084..0x10093, from LD1_buf LD2 need 4..5, from LD2_buf	iter 1 MADD || MADD	iter 0
4	iter 3 LD0 need 0x10010..0x1001F, from LD0_buf LD1 need 0x10090 .. 0x1009F, from LD1_buf LD2 need 0..1, from LD2_buf	iter 2 MADD || MADD	iter 1

Table 6-361. Load and Store Buffering Example, Short-Type Horizontal Filter (continued)

Cycle	LD stage	OP stage	ST stage
5	iter 4 LD0 need 0x10012..0x10021, **read IBUFLA** LD0_buf = {0x10010..0x1002F} LD1 need 0x10092 .. 0x100A1, **stall** LD2 need 2..3, from LD2_buf	iter 3 MADD || MADD	iter 2 ST0 store 0x10400..0x1040F ST0 queued ST1 store 0x10480..0x1048F ST1 queued
6	iter 4 LD1 **read IBUFLA** LD1_buf = {0x10090..0x100AF}	stalled	stalled
7	iter 5 LD0 need 0x10014..0x10023, from LD0_buf LD1 need 0x10094..0x100A3, from LD1_buf LD2 need 4..5, from LD2_buf	iter 4 MADD || MADD	iter 3 ST0 **write IBUFLA**
8	iter 6 LD0 need 0x10020..0x1002F, from LD0_buf LD1 need 0x100A0..0x100AF, from LD1_buf LD2 need 0..1, from LD2_buf	iter 5 MADD || MADD	iter 4 ST1 **write IBUFLA**
9	iter 7 LD0 need 0x10022..0x10031, **read IBUFLA** LD0_buf = {0x20..x3F} LD1 need 0x100A2..0x100B1, **stall** LD2 need 2..3, from LD2_buf	iter 6	iter 5 ST0 store 0x10410..0x1041F ST0 queuedST1 store 0x10490..0x1049F ST1 queued
10	iter 7 LD1 **read IBUFLA** LD1_buf = {0x100A0..0x100BF}	stalled	stalled

In this loop, with 2 VLDH_NPT to sustain 16 multiply-accumulates per iteration, the load stage stalled due to memory contention in IBUFL from the 2 loads. The read buffer supplies data for 2 subsequent iterations in steady state, leaving 2 memory-read free cycles every 3 iterations. With output writing back to IBUFL, the store buffer effectively delays the memory writes to use these free memory slots. Each i4 loop of 3 iterations thus takes 4 cycles to complete.

6.3.5.7 VCOP Per-Loop Overhead

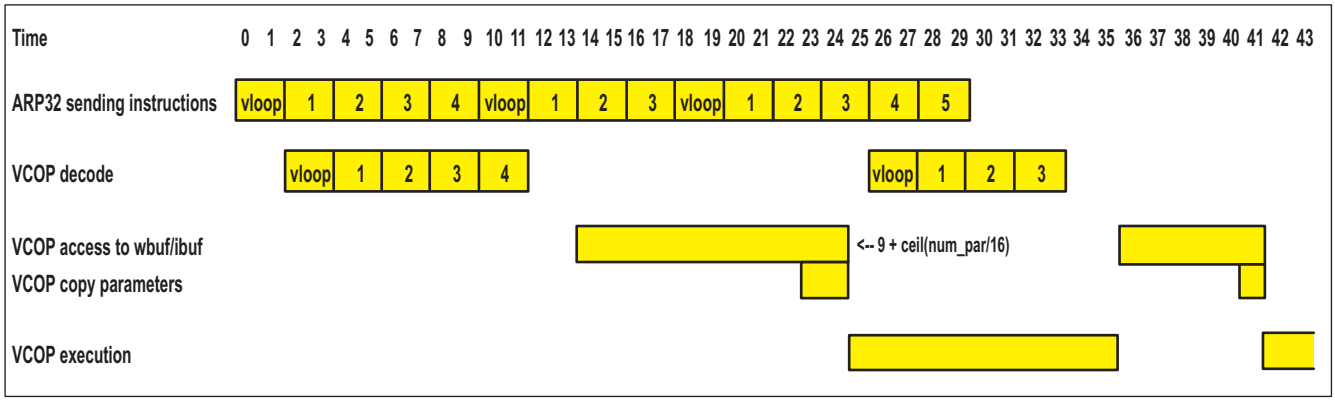
For each vector command specifying a loop, several overhead components are present. In general, developers are encouraged to run place into each loop as much processing as possible and run as many iterations as possible to reduce the percentage of time spent on overhead.

Overhead components are:

- Command decode time: it takes 2 cycles per instruction to decode, since scalar core is providing instructions at this rate. Command decode time can be hidden if ARP32 feeds these instructions while VCOP is executing a previous command.
- Parameter fetch time: it takes 9 + ceiling(num_param/16) cycles to fetch parameter from data memory (WBUF, IBUFL, or IBUFH).
- Execution pipeline ramp up/down time: it takes time to ramp up and down the long pipeline of VCOP. Minimal time for any loop is execution is about 17 cycles, due to the pipeline ramp up/down time.

Figure 6-64 shows a decode-parameter-fetch-execution time line.

Figure 6-64. Example of Operation Delay Slots



- t = 0, ARP32 sends VLOOP, followed by VCOP instructions in the loop, instructions go into FIFO.
- t = 2, VCOP pulls instruction out of FIFO, recognizes VLOOP.
- t = 10 (depend on command length), VCOP pulls last instruction of loop from FIFO.
- t = 10, ARP32 sends next command, starting from VLOOP.
- t = 14, VCOP starts access to parameter memory, 16 parameters/cycle.
- t = 18, ARP32 keeps sending vector commands, until program stream switches to scalar code, or VCOP pre-decode instruction FIFO becomes full (capacity is 64 words).
- t = 23 (14 + 9), VCOP starts to copy parameters into decoded command buffer.
- t = 25 (after parameter copy), VCOP starts executing the loop.
- t = 26 (sync to 250-MHz clock), VCOP pulls instruction out of FIFO, recognizes VLOOP.
- t = 32 (depend on command length), VCOP pulls last instruction of loop from FIFO.
- t = 36 (after prev loop finishes executing), VCOP starts to access parameter memory.

This is for showing the dependency among activities; execution time here is not realistic; it takes minimally 17 cycles to execute any loop, due to the pipeline depth.

6.3.5.8 VCOP Error Handling

Upon detection of erroneous instruction code, parameter, load/store address, or loop in execution exceeding programmed max iteration count, VCOP signals the error by generating an interrupt and registering the error source in the MMR VCOP_ERROR. Refer to register manual for details.

Upon detection of error, VCOP:

1. Terminates current loop, leaving vloop_ptr, param_ptr, loop variables i0, i1, i2, i3, i4 in MMR.
2. Parses and clears vector instructions until encountering VWDONE.
3. Issues interrupt, assert vec_done = 1, deassert vec_rdy = 0 to block subsequent vector instructions.
4. Upon VCOP_ERROR status being cleared (by writing 1s to clear bits 10:0), VCOP becomes idle (assert VCOP_STATUS[2]VEC_RDY = 1) and ready to resume processing.
5. Alternatively, VCOP can be reset to resume processing.

Depending on the EVE-level memory switch setting, VCOP's access of the data memory space may be deemed invalid by the memory switch. When this happens, the memory switch signals VCOP to enter error handling state, as described above. VCOP does not send an error interrupt, as the memory switch is doing that (for invalid access from VCOP or from other masters).

6.3.5.9 Vector Operation Details

6.3.5.9.1 VABS

Assembly syntax:	**VABS** *src1*, *dst*
Operation:	Absolute value
Classification:	1-input 1-output
Bit width:	40-bit
Delay slot:	no
C statement:	dst = abs(src1);

6.3.5.9.2 VABSDIF

Assembly syntax:	**VABSDIF** *src1*, *src2*, *dst*
Operation:	absolute-difference
Classification:	2-input 1-output
Bit width:	40-bit
Delay slot:	no
C statement:	dst = abs(src1 – src2);

6.3.5.9.3 VADD

Assembly syntax:	**VADD** *src1*, *src2*, *dst*
Operation:	addition
Classification:	2-input 1-output
Bit width:	40-bit
Delay slot:	no
C statement:	dst = src1 + src2;

6.3.5.9.4 VADDH

Assembly syntax:	**VADDH** *src1*, *src2*, *dst*
Operation:	Add high
Classification:	2-input 1-output
Bit width:	40-bit
Delay slot:	no
C statement:	dst = src1 + (src2 >> 32);

6.3.5.9.5 VADDSUB

Assembly syntax:	VADDSUB src1/dst1, src2/dst2
Operation:	Add-subtract
Classification:	2-input 2-output
Bit width:	40-bit
Delay slot:	no
C statement:	dst1 = src1 + src2; dst2 = src1 – src2;

6.3.5.9.6 VADD3

Assembly syntax:	**VADD3** *src1*, *src2*, *src3*, *dst* (dst same as src3)
Operation:	Add 3 items,
Classification:	3-input 1-output
Bit width:	40-bit
Delay slot:	one
C statement:	dst = src1 + src2 + src3;

6.3.5.9.7 VADIF3

Assembly syntax:	**VADIF3** *src1*, *src2*, *src3*, *dst* (dst same as src3)
Operation:	Add difference
Classification:	3-input 1-output
Bit width:	40-bit
Delay slot:	one
C statement:	dst = src1 – src2 + src3;

6.3.5.9.8 VAND

Assembly syntax:	**VAND** *src1*, *src2*, *dst*
Operation:	Bitwise and
Classification:	2-input 1-output
Bit width:	40-bit
Delay slot:	no
C statement:	dst = src1 & src2;

6.3.5.9.9 VANDN

Assembly syntax:	**VANDN** *src1*, *src2*, *dst*
Operation:	Bitwise and-not
Classification:	2-input 1-output
Bit width:	40-bit
Delay slot:	no
C statement:	dst = src1 & ~src2;

6.3.5.9.10 VAND3

Assembly syntax:	**VAND3** *src1*, *src2*, *src3*, *dst* (dst same as src3)
Operation:	Bitwise-and 3 items
Classification:	3-input 1-output
Bit width:	40-bit
Delay slot:	one
C statement:	dst3 = src1 & src2 & src3;

Using VAND3 accelerates a morphological erosion operation.

6.3.5.9.11 VBINLOG

Assembly syntax:	**VBINLOG** *src1*, *dst*
Operation:	Binary log
Classification:	1-input 1-output
Bit width:	32-bit
Delay slot:	one
C statement:	As follows

VBINLOG computes an approximate binary log, by detecting the left-most 1 bit in the input, encoding the bit position in bits 31..28, and left-justifying the following input bits in bits 27..0. The detected position of bit 31 is encoded as 15 (bit position minus 16), bit 30 as 14, and so on, and bit 17 as 1. If the left-most 1 bit is not found in bits 31..17, 0 is coded.

The VBINLOG operation is a non-standard floating point representation. Say a 32-bit integer x = 2^exp * (1 + frac/(2^28)), where exp and frac are integers. If (exp >= 16), binlog(x) = (exp – 16) * (2^28) + frac, concatenating the 4-bit adjusted exponent, exp – 16, with the 28-bit fraction. Otherwise, the representation is subnormal, leaving 0 in the exponent and bits 16..0 of input left-justified in the fraction part of output, bits 27..0.

For example, the number 0x0700_0000 has leading 1 in bit 26, so the bit position is coded as 26 – 16 = 10 = 0xA. The subsequent bits, two 1s immediately following the leading 1, left-justified in 28 bits or 7 hex digits, is 0xC00_0000. Thus the VBINLOG outcome of the number is 0xAC00_0000.

For a subnormal example, the number 0x8000 < 2^16 (is half of that), so its VBINLOG outcome is 0x0400_0000.

Figure 6-65. Binlog Function

input																		binlog output							
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	...	31	30	29	28	27	26	...	0
1	b30	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16	b15	...	1	1	1	1	b30	b29	...	b3
0	1	b29	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16	b15	...	1	1	1	0	b29	b28	...	b2
0	0	1	b28	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16	b15	...	1	1	0	1	b28	b27	...	b1
0	0	0	1	b27	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16	b15	...	1	1	0	0	b27	b26	...	b0
0	0	0	0	1	b26	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16	b15	...	1	0	1	1	b26	b25	...	0
0	0	0	0	0	1	b25	b24	b23	b22	b21	b20	b19	b18	b17	b16	b15	...	1	0	1	0	b25	b24	...	0
0	0	0	0	0	0	1	b24	b23	b22	b21	b20	b19	b18	b17	b16	b15	...	1	0	0	1	b24	b23	...	0
0	0	0	0	0	0	0	1	b23	b22	b21	b20	b19	b18	b17	b16	b15	...	1	0	0	0	b23	b22	...	0
0	0	0	0	0	0	0	0	1	b22	b21	b20	b19	b18	b17	b16	b15	...	0	1	1	1	b22	b21	...	0
0	0	0	0	0	0	0	0	0	1	b21	b20	b19	b18	b17	b16	b15	...	0	1	1	0	b21	b20	...	0
0	0	0	0	0	0	0	0	0	0	1	b20	b19	b18	b17	b16	b15	...	0	1	0	1	b20	b19	...	0
0	0	0	0	0	0	0	0	0	0	0	1	b19	b18	b17	b16	b15	...	0	1	0	0	b19	b18	...	0
0	0	0	0	0	0	0	0	0	0	0	0	1	b18	b17	b16	b15	...	0	0	1	1	b18	b17	...	0
0	0	0	0	0	0	0	0	0	0	0	0	0	1	b17	b16	b15	...	0	0	1	0	b17	b16	...	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	b16	b15	...	0	0	0	1	b16	b15	...	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	b16	b15	...	0	0	0	0	b16	b15	...	0

Input of VBINLOG is 32-bit, and so is the outcome, not 40-bit. VLMBD supports full 40-bit input range leading bit detection. VBINLOG is useful in compressing dynamic range and preserving precision. VBINLOG followed by table lookup can be used to produce the reciprocal of a number.

6.3.5.9.12 VBITC

Assembly syntax:	**VBIT** *src1*, *dst*
Operation:	Bit count
Classification:	1-input 1-output
Bit width:	32-bit
Delay slot:	one
C statement:	As follows

Count number of “1” bits in src1[31:0].

6.3.5.9.13 VBITDI

Assembly syntax:	**VBITDI** *src1*, *dst1*, *dst2*
Operation:	Bit deinterleave
Classification:	1-input 2-output
Bit width:	32-bit
Delay slot:	0
C statement:	As follows

Bit deinterleave is basically the inverse of bit interleave (VBITI).

For example,

```
src1 = 0x523(0000_0000_0000_0000_0000_1001_0010_0011)
```

Results in:

```
dst1 = 0x25 (0000_0000_0010_0101),
dst2 = 0x11(0000_0000_0001_0001)
```

6.3.5.9.14 VBITI

Assembly syntax:	**VBITI** *src1*, *src2*, *dst*
Operation:	Bit interleave
Classification:	2-input 1-output
Bit width:	32-bit
Delay slot:	one
C statement:	As follows

Bit interleave 16 LSBs of src1 and src2 (each), and write to dst.

For example,

```
src1 = 0x25 (0000_0000_0010_0101)
src2 = 0x11(0000_0000_0001_0001)
```

Results in:

```
dst = 0x523(0000_0000_0000_0000_0000_1001_0010_0011)
```

6.3.5.9.15 VBITPK

Assembly syntax:	**VBITPK** *src1*, *src2*, *dst*
Operation:	Bit-pack
Classification:	2-input 1-output
Bit width:	33-bit
Delay slot:	one
C statement:	As follows

Pack boolean (src1[i] >= src2[i]) into bits in dst[0], and broadcast dst[0] to all SIMD lanes so that dst[j] = dst[0], 1 <= j <= NSIMD.

VBITPK is useful in accessing bit-packed data (binary images).

6.3.5.9.16 VBITR

Assembly syntax:	**VBITR** *src1*, *dst*
Operation:	Bit reverse
Classification:	1-input 1-output
Bit width:	32-bit
Delay slot:	one
C statement:	As follows

Reverse bits in src1[31:0] and write to dst.

6.3.5.9.17 VBITTR

Assembly syntax:	**VBITTR** *src1*, *dst*
Operation:	Bit transpose
Classification:	1-input 1-output
Bit width:	NSIMD-bit
Delay slot:	one

VBITTR, bit transpose, also called corner turn, takes N × N bits in the source, from N LSBs of each of the N registers, and transpose them. Here is an example:

Input:			**Output:**		
		Bit position			**Bit position**
Lane	**Value**	**0 1 2 3**	**Lane**	**Value**	**0 1 2 3**
0	1	1 0 0 0	0	5	1 0 1 0
1	2	0 1 0 0	1	6	0 1 1 0
2	3	1 1 0 0	2	8	0 0 0 1
3	4	0 0 1 0	3	0	0 0 0 0

6.3.5.9.18 VBITUNPK

Assembly syntax:	**VBITUNPK** *src1*, *src2*, *dst*
Operation:	Bit-unpack
Classification:	2-input 1-output
Bit width:	33-bit
Delay slot:	one
C statement:	As follows

Unpack bits in src1[0]. Set dst[i] = src2[i] if the unpacked bit to lane i is 1, otherwise dst[i] = 0.

6.3.5.9.19 VCMOV

Assembly syntax:	**VCMOV** *cond*, *src1*[, *src2*|*dst*]
Operation:	Conditional move
Classification:	special
Bit width:	40-bit
Delay slot:	no
C statement:	As follows

cond = {I1234_ZERO, I234_ZERO, I34_ZERO, LAST_I4_ZERO, LAST_I34, LAST_I234, LAST_I1234}.

Move src1 to dst in the iterations designated by cond. This is parallel move within each SIMD lane; dst[i] = src1[i] when condition is true. Otherwise, dst[i] = src2[i]. By always writing to the dst register, dst register does not need to be {V0, V1, V2, V3} (see Section 6.3.5.5.6 for details on the store-without-being destination register constraint).

6.3.5.9.20 VCMPEQ

Assembly syntax:	**VCMPEQ***src1*, *src2*, *dst*
Operation:	Compare equal
Classification:	2-input 1-output
Bit width:	40-bit
Delay slot:	no
C statement:	dst = (src1 == src2) ? 1 : 0;

6.3.5.9.21 VCMPGE

Assembly syntax:	**VCMPGE** *src1*, *src2*, *dst*
Operation:	Compare greater-than-or-equal
Classification:	2-input 1-output
Bit width:	40-bit
Delay slot:	no
C statement:	dst = (src1 >= src2) ? 1 : 0;

6.3.5.9.22 VCMPGT

Assembly syntax:	**VCMPGT** *src1*, *src2*, *dst*
Operation:	Compare greater-than
Classification:	2-input 1-output
Bit width:	40-bit
Delay slot:	no
C statement:	dst = (src1 > src2) ? 1 : 0;

6.3.5.9.23 VDINTRLV

Assembly syntax:	**VDINTRLV** {*src1*|*dst1*}, {*src2*|*dst2*}
Operation:	Deinterleave
Classification:	2-input 2-output
Bit width:	40-bit
Delay slot:	no
C statement:	As follows

VDINTRLV is basically the inverse of VINTRLV.

For 8-way SIMD, this is used:

```
dst1[0] = src1[0];
dst2[0] = src1[1];
dst1[1] = src1[2];
dst2[1] = src1[3];
dst1[2] = src1[4];
dst2[2] = src1[5];
dst1[3] = src1[6];
dst2[3] = src1[7];
dst1[4] = src2[0];
dst2[4] = src2[1];
dst1[5] = src2[2];
dst2[5] = src2[3];
dst1[6] = src2[4];
dst2[6] = src2[5];
dst1[7] = src2[6];
dst2[7] = src2[7];
```

6.3.5.9.24 VDINTRLV2

Assembly syntax:	**VDINTRLV2** {*src1*|*dst1*}**,** {*src2*|*dst2*}
Operation:	Deinterleave with 2-element frequency
Classification:	2-input 2-output
Bit width:	40-bit
Delay slot:	no
C statement:	As follows

VDINTRLV2 is basically the inverse of VINTRLV2.

For 8-way SIMD, this is used:

```
dst1[0] = src1[0];
dst1[1] = src1[1];
dst2[0] = src1[2];
dst2[1] = src1[3];
dst1[2] = src1[4];
dst1[3] = src1[5];
dst2[2] = src1[6];
dst2[3] = src1[7];
dst1[4] = src2[0];
dst1[5] = src2[1];
dst2[4] = src2[2];
dst2[5] = src2[3];
dst1[6] = src2[4];
dst1[7] = src2[5];
dst2[6] = src2[6];
dst2[7] = src2[7];
```

6.3.5.9.25 VEXITNZ

Assembly syntax:	**VEXITNZ** *level***,** *src1*
Operation:	Exit upon non-zero
Classification:	special
Bit width:	40-bit
Delay slot:	no
C statement:	As follows

VEXITNZ is for early termination of VLOOP or in the repeat-loop level. The designated src1 register's 0th way is examined every time the instruction executes, and if it is not zero, the hardware terminates either at VLOOP level or at VRPT level, as indicated by the level parameter.

Parameter src1 is examined at the time the instruction executes, and an exit/no-exit decision is made. However, any exit/no-exit action is not taken immediately. For VEXITNZ VLOOP, subsequent instructions in the same iteration, remaining operations and all the stores, are executed, then the loop terminates. For VEXITNZ VRPT; all subsequent instructions in the same loop are executed, including the rest of the iteration, and all remaining iterations until i1, i2, i3, i4 all run to respective end counts, then the repeated loop terminates.

There can only be one VEXITNZ in each vector command; no mixing of VEXITNZ VLOOP and VEXITNZ VRPT. When the vector command is not repeated (RPT_END = 0), VEXITNZ VRPT behaves like VEXITNZ VLOOP.

6.3.5.9.26 VINTRLV

Assembly syntax:	**VINTRLV** {*src1*|*dst1*}**,** {*src2*|*dst2*}
Operation:	Interleave
Classification:	2-input 2-output
Bit width:	40-bit
Delay slot:	no
C statement:	As follows

VINTRLV is basically interleaving among SIMD lanes.

For 8-way SIMD, this is used:

```
dst1[0] = src1[0];
dst1[1] = src2[0];
dst1[2] = src1[1];
dst1[3] = src2[1];
dst1[4] = src1[2];
dst1[5] = src2[2];
dst1[6] = src1[3];
dst1[7] = src2[3];
dst2[0] = src1[4];
dst2[1] = src2[4];
dst2[2] = src1[5];
dst2[3] = src2[5];
dst2[4] = src1[6];
dst2[5] = src2[6];
dst2[6] = src1[7];
dst2[7] = src2[7];
```

6.3.5.9.27 VINTRLV2

Assembly syntax:	**VINTRLV2** {*src1*|*dst1*}**,** {*src2*|*dst2*}
Operation:	Interleave with 2-element frequency
Classification:	2-input 2-output
Bit width:	40-bit
Delay slot:	no
C statement:	As follows

VINTRLV is basically interleaving among SIMD lanes.

For 8-way SIMD, this is used:

```
dst1[0] = src1[0];
dst1[1] = src1[1];
dst1[2] = src2[0];
dst1[3] = src2[1];
dst1[4] = src1[2];
dst1[5] = src1[3];
dst1[6] = src2[2];
dst1[7] = src2[3];
dst2[0] = src1[4];
dst2[1] = src1[5];
dst2[2] = src2[4];
dst2[3] = src2[5];
dst2[4] = src1[6];
dst2[5] = src1[7];
dst2[6] = src2[6];
dst2[7] = src2[7];
```

6.3.5.9.28 VINTRLV4

Assembly syntax:	**VINTRLV4** {*src1*|*dst1*}**,** {*src2*|*dst2*}
Operation:	Interleave with 4-element frequency
Classification:	2-input 2-output
Bit width:	40-bit
Delay slot:	no
C statement:	As follows

VINTRLV is basically interleaving among SIMD lanes.

For 8-way SIMD, this is used:

```
dst1[0] = src1[0];
dst1[1] = src1[1];
dst1[2] = src1[2];
dst1[3] = src1[3];
dst1[4] = src2[0];
dst1[5] = src2[1];
dst1[6] = src2[2];
dst1[7] = src2[3];
dst2[0] = src1[4];
dst2[1] = src1[5];
dst2[2] = src1[6];
dst2[3] = src1[7];
dst2[4] = src2[4];
dst2[5] = src2[5];
dst2[6] = src2[6];
dst2[7] = src2[7];
```

6.3.5.9.29 VLMBD

Assembly syntax:	**VLMBD** *src1***,** *src2***,** *dst*
Operation:	Left-most bit detect
Classification:	3-input 1-output
Bit width:	40-bit
Delay slot:	one
C statement:	As follows

Depending on src2 being zero or not, VLMBD searches for 0 or 1 in src1, starting from bit 39. Bit position where the first 0 or 1 occurs is returned, and -1 is returned if it's not found.

6.3.5.9.30 VMADD

Assembly syntax:	**VMADD** *src1*, *src2*, *src3*, *dst*, **RND:** *rnd_param* (dst same as src3)
Operation:	Multiply-add
Classification:	3-input 1-output
Bit width:	17-bit src1/src2, 40-bit src3/dst
Delay slot:	2 delay from src1/src2, 1 delay from src3
C statement:	dst = src3 + round_or_shift(src1 * src2, rnd_param)

The RND field specifies a parameter, in which additional fields are read to support built-in round/shift feature in the operation:

Figure 6-66. VMPY, VMADD and VMSUB Rounding Parameters

Parameter Indexed by rnd_param

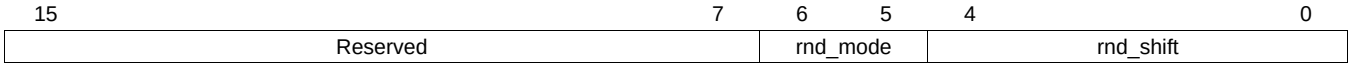


Table 6-362. Parameter Indexed by rnd_param Field Descriptions

Bits	Name	Description	
15-7	Reserved		
6-5	rnd_mode	Sets rounding mode:	
		0	No rounding
		1	Round
		2	Truncate
		3	Left-shift
4-0	rnd_shift	Number of bits to round/shift, limited to:	
		{8|15|16}	Round/truncate
		1	Left-shift

When rnd_mode = round or truncate, src1 * src2 is rounded down before the addition or subtraction. When rnd_mode = left-shift, src1 * src2 is shifted up before the addition or subtraction. Note the restriction on the number of bits to shift.

6.3.5.9.31 VMAX

Assembly syntax:	**VMAX** *src1*, *src2*, *dst*
Operation:	Maximum
Classification:	2-input 1-output
Bit width:	33-bit
Delay slot:	no
C statement:	dst = (src1 > src2) ? src1 : src2;

6.3.5.9.32 VMAXSETF

Assembly syntax:	**VMAXSETF** *src1*, {*src2*|*dst1*}, *dst2*
Operation:	Maximum and set flag
Classification:	2-input 2-output
Bit width:	33-bit
Delay slot:	no
C statement	dst1 = max(src1, src2); dst2 = (src1>src2) ? 1:0

6.3.5.9.33 VMIN

Assembly syntax:	**VMIN** *src1*, *src2*, *dst*
Operation:	Minimum
Classification:	2-input 1-output
Bit width:	33-bit
Delay slot:	no
C statement:	dst = (src1 < src2) ? src1 : src2;

6.3.5.9.34 VMINSETF

Assembly syntax:	**VMINSETF** *src1*, {*src2*|*dst1*}, *dst2*
Operation:	Minimum and set flag
Classification:	2-input 2-output
Bit width:	33-bit
Delay slot:	
C statement:	dst1 = min(src1, src2); dst2 = (src1 < src2) ? 1 : 0;

6.3.5.9.35 VMPY

Assembly syntax:	**VMPY** *src1*, *src2*, *dst*, **RND:** *rnd_param*
Operation:	multiply
Classification:	2-input 1-output
Bit width:	17-bit input, 40-bit output
Delay slot:	one
C statement:	dst = round_or_shift(src1 * src2, rnd_param);

See VMADD for the RND field.

6.3.5.9.36 VMSUB

Assembly syntax:	**VMSUB** *src1*, *src2*, *src3*, *dst*, **RND:** *rnd_param* (dst same as src3)
Operation:	Multiply-subtract
Classification:	3-input 1-output
Bit width:	17-bit src1/src2, 40-bit src3/dst
Delay slot:	2 delay from src1/src2, 1 delay from src3
C statement:	dst = src3 - round_or_shift(src1 * src2, rnd_param);

See VMADD for the RND field.

6.3.5.9.37 VNOP

Assembly syntax:	**VNOP**
Operation:	no operation
Classification:	n/a
Bit width:	n/a
Delay slot:	no
C statement:	n/a

NOP is used to pad the two operations per clock cycle operation pipeline, so the next operation is mapped to functional unit 1. This is used to align data dependency to within functional unit 0 or 1, to avoid idle cycles being inserted by hardware.

6.3.5.9.38 VNOT

Assembly syntax:	**VNOT** *src1*, *dst*
Operation:	Bitwise complement
Classification:	1-input 1-output
Bit width:	40-bit
Delay slot:	no
C statement:	dst = ~src1;

6.3.5.9.39 VOR

Assembly syntax:	**VOR** *src1*, *src2*, *dst*
Operation:	Bitwise or
Classification:	2-input 1-output
Bit width:	40-bit
Delay slot:	no
C statement:	dst = src1 | src2;

6.3.5.9.40 VOR3

Assembly syntax:	**VOR3** *src1*, *src2*, *src3*, *dst* (dst same as src3)
Operation:	Bitwise-or 3 items
Classification:	3-input 1-output
Bit width:	40-bit
Delay slot:	one
C statement:	dst3 = src1 | src2 | src3;

Using VOR3 accelerates the morphological dilation operation.

6.3.5.9.41 VRND

Assembly syntax:	**VRND** *src1*, *src2*, *dst*
Operation:	Round src1 by number of bits expressed in src2
Classification:	2-input 1-output
Bit width:	40-bit src1/dst, 5-bit src2, allowing rounding by 0 ~ 31 bits
Delay slot:	one
C statement:	rnd_add = (src2 > 0) ? (1 << (src2 – 1)) : 0; dst = (src1 + rnd_add) >> src2;

A half-unit is added before the right-shift. For example, if src2 = 8, dst = (src1 + 128) >> 8.

Only src2[4:0] are read, so for example src2 = 40 causes a rounding by 8 bits, since src2[4:0] = 0x08.

6.3.5.9.42 VSAD

Assembly syntax:	**VSAD** *src1*, *src2*, *src3*, *dst* (dst same as src3)
Operation:	Sum of absolute difference, dst = |src1 – src2| + src3
Classification:	3-input 1-output
Bit width:	40-bit
Delay slot:	one
C statement:	dst = src3 + abs(src1 – src2);

6.3.5.9.43 VSEL

Assembly syntax:	**VSEL** *src1*, *src2*, *src3*, *dst* (dst same as src3)
Operation:	Select
Classification:	3-input 1-output
Bit width:	40-bit
Delay slot:	no
C statement:	dst = src1 ? src2 : src3;

6.3.5.9.44 VSHF

Assembly syntax:	**VSHF** *src1*, *src2*, *dst*
Operation:	Arithmetic shift, shift left when src2 > 0, shift right by –src2 when src2 < 0
Classification:	2-input 1-output
Bit width:	40-bit src1/dst, 6-bit src2, allowing shifting by -32 ~ 31 bits
Delay slot:	no
C statement:	dst = (src2 >= 0) ? (src1 << src2) : (src1 >> –src2);

VSHF is an arithmetic shift, not logic shift, compared to conventional instruction sets that have both options. This is because VCOP performs sign extension in the load stage, and all signed/unsigned 8/16/32-bit operations are carried out with the 40-bit register file.

Only src2[5:0] are read, so for example src2 = 32 causes a right-shift by 32 bits, since src2[5:0] = 0x20.

6.3.5.9.45 VSHFOR

Assembly syntax:	**VSHFOR** *src1*, *src2*, *src3*, *dst* (dst same as src3)
Operation:	Shift-or
Classification:	3-input 1-output
Bit width:	40-bit src1/src3, 6-bit src2
Delay slot:	one
C statement:	dst3 = (src2 >= 0) ? (src3 | (src1 << src2)) : (src3 | (src1 >> –src2));

See VSHF for the supported range of src2.

6.3.5.9.46 VSHF16

Assembly syntax:	**VSHF16** *src1*, *dst2*, *dst2*
Operation:	Shift-up 16 bits
Classification:	1-input 2-output
Bit width:	33-bit src1
Delay slot:	no
C statement:	dst1 = (src1 & 0xffff) << 16;dst2 = src1 >> 16;

Shift up src1 by 16 bits, producing two destinations to hold the outcome, dst1 having the lower unsigned 32-bit, and dst2 having the signed upper portion, sign extended from src1[32] to full 40 bits in dst2.

VSHF16 is useful for 32 × 32 multiplication.

6.3.5.9.47 VSIGN

Assembly syntax:	**VSIGN** *src1*, *src2*, *dst*
Operation:	Sign
Classification:	2-input 1-output
Bit width:	40-bit
Delay slot:	no
C statement:	dst = (src1 == 0) ? 0 :((src1 > 0) ? src2 : –src2);

6.3.5.9.48 VSORT2

Assembly syntax:	**VSORT2** *src1*, *src2*, *dst*
Operation:	Sort two items
Classification:	2-input 2-output
Bit width:	33-bit
Delay slot:	no
C statement:	dst1 = min(src1, src2); dst2 = max(src1, src2);

6.3.5.9.49 VSUB

Assembly syntax:	**VSUB** *src1*, *src2*, *dst*
Operation:	Subtraction
Classification:	2-input 1-output
Bit width:	40-bit
Delay slot:	no
C statement:	dst = src1 – src2;

6.3.5.9.50 VSWAP

Assembly syntax:	**VSWAP** *cond*, {*src1*|*dst1*}, {*src2*|*dst2*}
Operation:	Conditional swap
Classification:	3-input 2-output
Bit width:	1-bit cond, 40-bit src1 & src2
Delay slot:	no
C statement:	if (cond & 1) swap(src1, src2)

Only the LSB of the vector register indicated by cond is checked. A swap/no-swap decision is made in each SIMD lane using the LSB of cond register in the lane.

6.3.5.9.51 VXOR

Assembly syntax:	**VXOR** *src1*, *src2*, *dst*
Operation:	Bitwise exclusive-or
Classification:	2-input 1-output
Bit width:	40-bit
Delay slot:	no
C statement:	dst = src1 ^ src2;

6.3.6 Debug Support

VCOP supports single stepping, one innermost (i4) iteration per step. Due to the deeply pipelined architecture, instruction-by-instruction single stepping support is not feasible.

Normally VCOP executes load in one stage of the pipeline, and operations and stores in the next stage of the pipeline. In single step mode, only one iteration is executed at a time, so there is no pipelining among iterations. The hardware executes all the loads, then operations and stores, then halts until instructed to continue on the next iteration. The signle step operation is controlled via the VCOP_CTRL register. The operation is enabled by setting VCOP_CTRL[0]STEP_EN to 1. Setting VCOP_CTRL[1]STEP_GO to 1 the engine starts executing i4 iteration.

As outlined in the previous section, step_en, step_go, step_rdy register fields are provided for single stepping control. In addition, VCOP internal state is exported as read-only view as memory-map registers as well.

Interaction of debug driver, scalar core, and VCOP is as follows:

1. Debug driver enables debug mode by setting VCOP_CTRL[0]STEP_EN = 1 and wait for VCOP_STATUS[0]STEP_RDY = 1.
2. Scalar core sends the vector command to be single-stepped the normal fashion.
3. Vector core executes the vector command and stalls at the end of the first i4 iteration and assert sVCOP_STATUS[0]STEP_RDY.
4. When VCOP_STATUS[0]STEP_RDY =1, debug driver accesses buffer switch and switch WBUF, IBUFL/H memories to system bus.
5. Debug driver read out the memory contents to refresh on the debugger user interface.
6. Debug driver read out VCOP internal state to refresh on the debugger user interface.
7. Debug driver accesses buffer switch and switch memories to VCOP.
8. Debug driver triggers VCOP to take the next step by writing ‘1’ to step_go, and wait for VCOP_STATUS[0]STEP_RDY = 1.
9. Repeat 3 through 8, until completion of the vector command or user input to stop single-stepping.
10. The debug driver disables the debug mode by setting VCOP_CTRL[0]STEP_EN = 0.
11. VCOP and scalar core continue to operate.

6.3.7 VCOP Register Manual

6.3.7.1 VCOP Instance Summary

Table 6-363. VCOP Instance Summary

Module Name	Base Address L3_MAIN Interconnect	Size

6.3.7.2 VCOP Registers

6.3.7.2.1 VCOP Registers Mapping Summary

6.3.7.2.2 VCOP Register Description

Table 6-364. VCOP_PID

Address Offset	0x0000 0000		
Physical Address		**Instance**	
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PID																															

Bits	Field Name	Description	Type	Reset
31:0	PID	IP Revision	R	0x0

Table 6-365. Register Call Summary for Register VCOP_PID

VCOP CPU and Instruction Set
- VCOP Registers Mapping Summary:

Table 6-366. VCOP_CTRL

Address Offset	0x0000 0004
Physical Address	**Instance**
Description	VCOP Control Register
Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														STEP_GO	STEP_EN

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1	STEP_GO	Starts executing a single i4 iteration 0: NOP 1: START	RW	0x0
0	STEP_EN	Enable Single Step mode 0: Disable 1: Enable	RW	0x0

Table 6-367. Register Call Summary for Register VCOP_CTRL

VCOP CPU and Instruction Set
- Debug Support: [0][1][2][3][4]
- VCOP Registers Mapping Summary:

Table 6-368. VCOP_STATUS

Address Offset	0x0000 0008
Physical Address	**Instance**
Description	VCOP status register.
Type	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																													VEC_RDY	VEC_DONE	STEP_RDY

Bits	Field Name	Description	Type	Reset
31:3	RESERVED	Reserved. Read returns 0s	R	0x0
2	VEC_RDY	Vector core ready to accept next vector instruction	R	0x0
1	VEC_DONE	Vector core has completed execution of submitted vector loops.	R	0x0
0	STEP_RDY	Ready for next step (single step) 0: Busy 1: Idle and ready for next step	R	0x0

Table 6-369. Register Call Summary for Register VCOP_STATUS

VCOP CPU and Instruction Set
- Wait for Vector Core Ready: [0]
- VCOP Error Handling: [1]
- Debug Support: [2][3][4][5]
- VCOP Registers Mapping Summary:

Table 6-370. VCOP_MAX_ITERS

Address Offset	0x0000 000C
Physical Address	**Instance**
Description	
Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																MAX_ITERS															

Bits	Field Name	Description	Type	Reset
31:16	RESESERVED	Reserved. Read returns 0s	R	0x0
15:0	MAX_ITERS	Maximum iteration count. Send interrupt when a loop in execution exceeds the programmed max iteration count. This is to guard against VCOP hangs due to run-away program. 0: Disable (default) 1: Enable	RW	0x0

Table 6-371. Register Call Summary for Register VCOP_MAX_ITERS

VCOP CPU and Instruction Set
- VCOP Registers Mapping Summary:

Table 6-372. VCOP_ERROR

Address Offset	0x0000 0010
Physical Address	**Instance**
Description	Error interrupt enalbe and status register. Writing 1 to the ERR_STi bits clears the interrupt status.
Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								ERR_DIS7	ERR_DIS6	ERR_DIS5	ERR_DIS4	ERR_DIS3	ERR_DIS2	ERR_DIS1	ERR_DIS0	RESERVED								ERR_ST7	ERR_ST6	ERR_ST5	ERR_ST4	ERR_ST3	ERR_ST2	ERR_ST1	ERR_ST0

Bits	Field Name	Description	Type	Reset
31:24	RESERVED	Reserved. Read returns 0s	RW	0x0
23	ERR_DIS7	Error Interrupt disable. 0:Enable 1: Disable ST_PDDA bank conflict	RW	0x0
22	ERR_DIS6	Error Interrupt disable. 0:Enable 1: Disable ST WBUF out-of-bound	RW	0x0
21	ERR_DIS5	Error Interrupt disable. 0:Enable 1: Disable ST IBUF out-of-bound	RW	0x0
20	ERR_DIS4	Error Interrupt disable. 0:Enable 1: Disable LD WBUF out-of-bound	RW	0x0
19	ERR_DIS3	Error Interrupt disable. 0:Enable 1: Disable LD IBUF out-of-bound	RW	0x0
18	ERR_DIS2	Error Interrupt disable. 0: Enable 1: Disable Illegal parameter (pointer not 32-bit aligned, pointer out-of-bound, exceed max repeat count)	RW	0x0
17	ERR_DIS1	Error Interrupt disable. 0:Enable 1:Disable Illegal instruction, all other causes than inside-loop instructions detected outside loop	RW	0x0
16	ERR_DIS0	Error Interrupt disable. 0: Enable 1: Disable Illegal instruction; inside-loop instructions (eg, VADD) detected outside loop.When this occurs, the decode value is indeterminate, since VCOP expects valid PC on vec_paddr bus, and ARP32 only sends PC with valid VLOOP instruction.	RW	0x0
15:8	RESERVED	Reserved. Read returns 0s	RW	0x0
7	ERR_ST7	ST_PDDA bank conflict error status: 0: No error 1: Error	RW	0x0
6	ERR_ST6	ST WBUF out-of-bound error status: 0: No error 1: Error	RW	0x0
5	ERR_ST5	ST IBUF out-of-bound error status: 0: No error 1: Error	RW	0x0
4	ERR_ST4	LD WBUF out-of-bound error status: 0: No error 1: Error	RW	0x0
3	ERR_ST3	LD IBUF out-of-bound error status: 0: No error 1: Error	RW	0x0
2	ERR_ST2	Illegal parameter error status. Effected when pointer is not 32-bit aligned, pointer is outof- bound, or exceed max repeat count. 0: No error 1: Error	RW	0x0

Bits	Field Name	Description	Type	Reset
1	ERR_ST1	Illegal instruction error status. Effected by all other causes than inside-loop instructions detected outside loop. 0: No error 1: Error	RW	0x0
0	ERR_ST0	Illegal instruction error status. Effected by inside-loop instructions (eg, VADD) detected outside loop. When this occurs, VCOP_VLOOP_PTR_DEC value is indeterminate, since VCOP expects valid PC on vec_paddr bus, and ARP32 only sends PC with valid VLOOP instruction. 0: No error 1: Error	RW	0x0

Table 6-373. Register Call Summary for Register VCOP_ERROR

VCOP CPU and Instruction Set
- VCOP Error Handling: [0][1]
- VCOP Registers Mapping Summary:

Table 6-374. VCOP_VLOOP_PTR

Address Offset	0x0000 0020		
Physical Address		**Instance**	
Description	The VLOOP pointer		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VLOOP_PTR																															

Bits	Field Name	Description	Type	Reset
31:0	VLOOP_PTR	VLOOP pointer.	R	0x0

Table 6-375. Register Call Summary for Register VCOP_VLOOP_PTR

VCOP CPU and Instruction Set
- Instruction Organization: [0]
- VCOP Registers Mapping Summary:

Table 6-376. VCOP_PARAM_PTR

Address Offset	0x0000 0024		
Physical Address		**Instance**	
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PARAM_PTR																															

Bits	Field Name	Description	Type	Reset
31:0	PARAM_PTR	Points to the beginning of parameter block for the loop in execution.	R	0x0

Table 6-377. Register Call Summary for Register VCOP_PARAM_PTR

VCOP CPU and Instruction Set

- Parameter Pointer: [0][1]
- VCOP Registers Mapping Summary:

Table 6-378. VCOP_I0_I1

Address Offset	0x0000 0030
Physical Address	**Instance**
Description	I0, I1 loop variables register provides a snapshot of i0 and i1
Type	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I1																I0															

Bits	Field Name	Description	Type	Reset
31:16	I1	Snapshot of I1 loop variable.	R	0x0
15:0	I0	Snapshot of I0 loop variable.	R	0x0

Table 6-379. Register Call Summary for Register VCOP_I0_I1

VCOP CPU and Instruction Set

- Nested for Loop Model: [0][1]
- VCOP Registers Mapping Summary:

Table 6-380. VCOP_I2_I3

Address Offset	0x0000 0034
Physical Address	**Instance**
Description	I2, I3 loop variables register provides a snapshot of i2 and i3
Type	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I3																I2															

Bits	Field Name	Description	Type	Reset
31:16	I3	Snapshot of I2 loop variable.	R	0x0
15:0	I2	Snapshot of I3 loop variable.	R	0x0

Table 6-381. Register Call Summary for Register VCOP_I2_I3

VCOP CPU and Instruction Set

- Nested for Loop Model: [0][1]
- VCOP Registers Mapping Summary:

Table 6-382. VCOP_I4

Address Offset	0x0000 0038
Physical Address	**Instance**
Description	I4 loop variables register provides a snapshot of i4
Type	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																I4															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Reserved Read returns 0s	R	0x0
15:0	I4	Snapshot of I4 loop variable.	R	0x0

Table 6-383. Register Call Summary for Register VCOP_I4

VCOP CPU and Instruction Set

- Nested for Loop Model: [0]
- VCOP Registers Mapping Summary:

Table 6-384. VCOP_LD_PTR_i

Address Offset	0x0000 0040 + (0x4*i)
Physical Address	**Instance**
Description	The LD pointer registers 0 to 7 or (VCOP_LD_PTR(0..7)) is a snapshot of the LD memory address. The LD unit is identified by the destination vector register V0..V7
Type	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LD_PTRi																															

Bits	Field Name	Description	Type	Reset
31:0	LD_PTRi	LD pointer i (i = 0 to 7).	R	0x0

Table 6-385. Register Call Summary for Register VCOP_LD_PTR_i

VCOP CPU and Instruction Set

- Features: [0]
- Vector Load: [1]
- VCOP Registers Mapping Summary:

Table 6-386. VCOP_ST_PTR_j

Address Offset	0x0000 0060 + (0x4*j)
Physical Address	**Instance**
Description	The ST pointer registers 0 to 7 (VCOP_VLOOP_ST_PTR(0..7)) is a snapshot of the ST memory address. The ST unit is identified by the order ST appears in program.
Type	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ST_PTR0																															

Bits	Field Name	Description	Type	Reset
31:0	ST_PTR0	ST pointer j (j=0 to 7).	R	0x0

Table 6-387. Register Call Summary for Register VCOP_ST_PTR_j

VCOP CPU and Instruction Set

- Features: [0]
- Vector Load: [1]
- VCOP Registers Mapping Summary:

Video Input Port

This chapter describes the Video Input Port (VIP) module for the device.

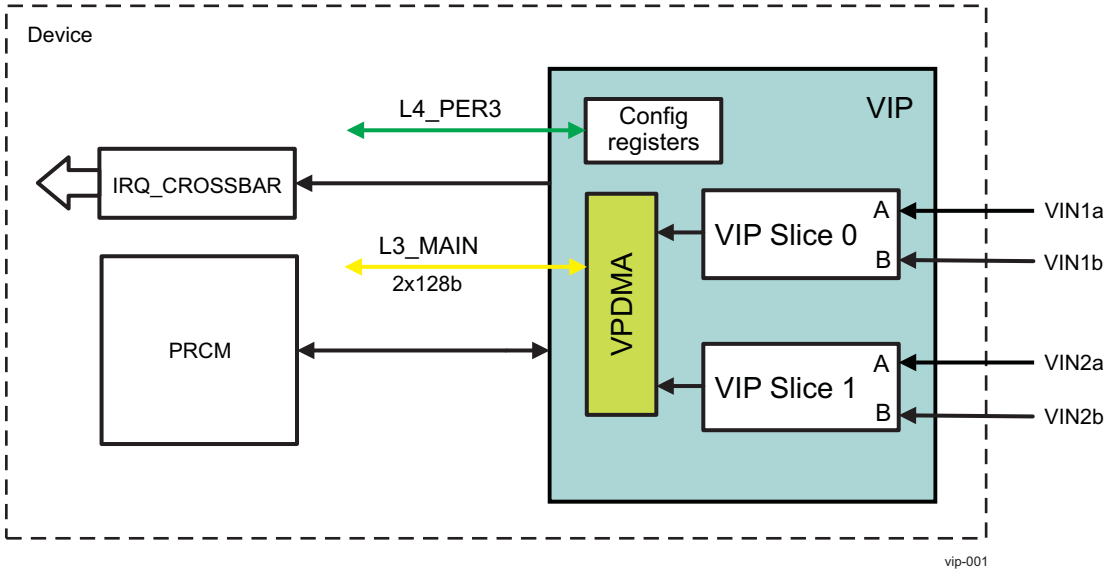
Topic **Page**

7.1 VIP Overview

The VIP module provides video capture functions for the device. VIP incorporates a multi-channel raw video parser, various video processing blocks, and a flexible Video Port Direct Memory Access (VPDMA) engine to store incoming video in various formats. The device integrates a single instantiation of the VIP module giving the ability of capturing up to four video streams.

Figure 7-1 shows a block diagram with the VIP module within the device.

Figure 7-1. VIP Overview



A VIP module includes the following main features:

- Two independently configurable external video input capture slices (Slice 0 and Slice 1) each providing up to two video input ports, Port A and Port B:
  - Port A can be configured as a 16/8-bit port.
  - Port B is a fixed 8-bit port.
  - At device level, the same device pads may be shared between Port A and Port B of each slice. For more information, see the multiplexing characteristics in device Data Manual.
- Each video Port A can be operated as a port with clock independent input channels (with interleaved or separated Y/C data input). Embedded sync and external sync modes are supported for all input configurations.
- Support for a single external asynchronous pixel clock, up to 165MHz per port.
- Pixel Clock Input Domain Port A supports up to one 16-bit input data bus, including BT.1120 style embedded sync for 16-bit data.
- Embedded Sync data interface mode supports single or multiplexed sources
- Discrete Sync data interface mode supports only single source input
- 16-bit data input plus discrete syncs can be configured to include:
  - 8-bit YUV422 (Y and U/V time interleaved)
  - 16-bit YUV422 (CbY and CrY time interleaved)
  - 16-bit RGB565
  - 16-bit RAW Capture
- Discrete sync modes include:
  - VSYNC + HSYNC (FID determined by FID signal pin or HSYNC/VSYNC skew)
  - VSYNC + ACTVID + FID
  - VBLANK + ACTVID (ACTVID toggles in VBLANK) + FID

  - VBLANK + ACTVID (no ACTVID toggles in VBLANK) + FID
- Multichannel parser (embedded syncs only):
  - Embedded syncs only
  - Pixel (2x or 4x) or Line multiplexed modes supported
  - Performs demultiplexing and basic error checking
  - Supports maximum of 9 channels in Line Mux (8 normal + 1 split line)
- Ancillary data capture support:
  - For 16-bit input, ancillary data may be extracted from any single channel
  - For 8-bit time interleaved input, ancillary data can be chosen from the Luma channel, the Chroma channel, or both channels
  - Horizontal blanking interval data capture only supported when using discrete syncs (VSYNC + HSYNC or VSYNC + HBLANK)
  - Ancillary data extraction supported on multichannel capture as well as single source streams
- Format conversion and scaling:
  - Programmable color space conversion
  - YUV422 to YUV444 conversion
  - YUV444 to YUV422 conversion
  - YUV422 to YUV420 conversion
  - YUV422 Source: YUV422 to YUV422, YUV422 to YUV420, YUV422 to YUV444, YUV422 to RGB888
  - Supports RAW to RAW (no processing)
  - Scaling and format conversions do not work for multiplexed input
- Supports up to 2047 pixels wide input - when scaling is engaged
- Supports up to 3840 pixels wide input - when only chroma up/down sampling is engaged, without scaling
- Supports up to 4095 pixels wide input - without scaling and chroma up/down sampling
- The maximum supported input resolution is further limited by pixel clock and feature-dependent constraints

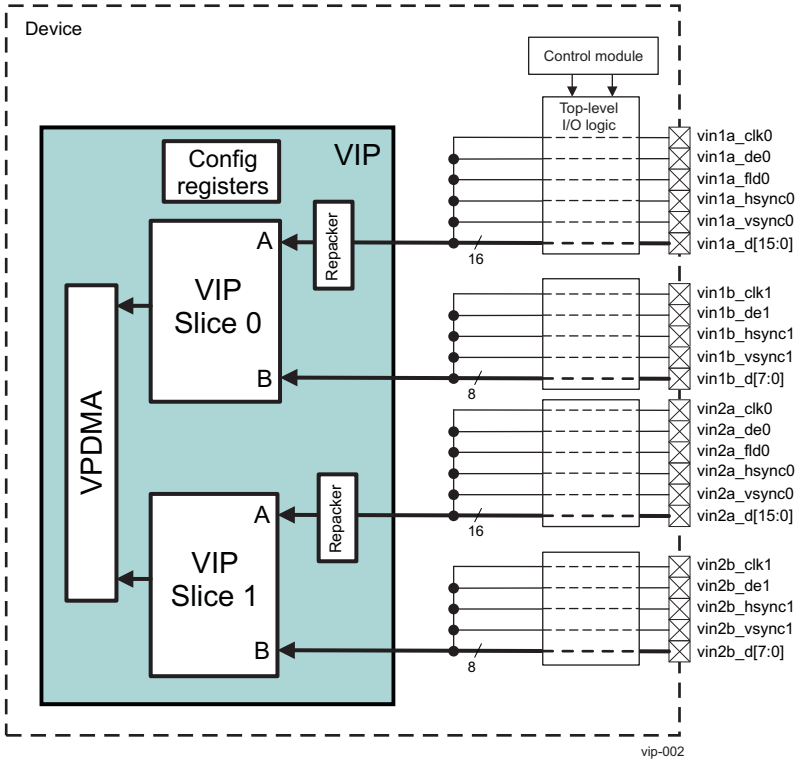
A VPDMA module includes the following main features:

- VPDMA output buffer size restriction feature, which ensures that writes do not exceed allocated memory buffer size
- Support for Tiled (2D) and raster addressing without bandwidth penalty
- Dual clients per channel allows for capture of scaled and nonscaled versions of the data stream (non-multiplexed mode only)
- Start on new frame capability
- Interrupt every X number of frames
- Interrupt every X lines (synced to frame start)
- Dynamic MFLAG generation

7.2 VIP Environment

This section describes the VIP module from an environment point of view (external connections). It describes the VIP connectivity options and lists all possible interfaces. Figure 7-2 is a block diagram of the VIP environment.

Figure 7-2. VIP Environment



NOTE: The path from a module pin to device pad(s) is defined at the device I/O logic level. The I/O logic maps the module signals to different pads of the device and is programmable in the device Control Module registers and/or dedicated module registers. For more information, see Section 13.4.6.1, *Pad Configuration Registers* in Chapter 13, *Control Module*.

Table 7-1. VIP Interface Signals

Sub-module Name	Signal Name	Type[1]	Description
Slice 0	vin1a_d[15:0]	I	Pixel data.
Port A	vin1a_clk0	I	Pixel clock.
	vin1a_vsync0	I	Vertical synchronization.
	vin1a_hsync0	I	Horizontal synchronization.
	vin1a_de0	I	The DE signal acts as an Input-enable signal to indicate when data must be latched using the input clock. DE is also referred to as ACTVID throughout the VIP chapter. Both of these terms, ACTVID and DE, are the same and used interchangeably.
	vin1a_fld0	I	The FID signal indicates the field identifier for the video input field: • 0 means even. • 1 means odd.
Slice 0	vin1b_d[7:0]	I	Pixel data.
Port B	vin1b_clk1	I	Pixel clock.
	vin1b_vsync1	I	Vertical synchronization.
	vin1b_hsync1	I	Horizontal synchronization.

(1) I = Input, O = Output, I/O = Input/Output

Table 7-1. VIP Interface Signals (continued)

Sub-module Name	Signal Name	Type[1]	Description
	vin1b_de1	I	The DE signal acts as an Input-enable signal to indicate when data must be latched using the input clock. DE is also referred to as ACTVID throughout the VIP chapter. Both of these terms, ACTVID and DE, are the same and used interchangeably.
Slice 1	vin2a_d[15:0]	I	Pixel data.
Port A	vin2a_clk0	I	Pixel clock.
	vin2a_vsync0	I	Vertical synchronization.
	vin2a_hsync0	I	Horizontal synchronization.
	vin2a_de0	I	The DE signal acts as an Input-enable signal to indicate when data must be latched using the input clock. DE is also referred to as ACTVID throughout the VIP chapter. Both of these terms, ACTVID and DE, are the same and used interchangeably.
	vin2a_fld0	I	The FID signal indicates the field identifier for the video input field: • 0 means even. • 1 means odd.
Slice 1	vin2b_d[7:0]	I	Pixel data.
Port B	vin2b_clk1	I	Pixel clock.
	vin2b_vsync1	I	Vertical synchronization.
	vin2b_hsync1	I	Horizontal synchronization.
	vin2b_de1	I	The DE signal acts as an Input-enable signal to indicate when data must be latched using the input clock. DE is also referred to as ACTVID throughout the VIP chapter. Both of these terms, ACTVID and DE, are the same and used interchangeably.

NOTE: At device level, the same device pads may be shared between Port A and Port B of each slice. For more information, see the multiplexing characteristics in device Data Manual.

Table 7-2 summarizes the mapping of YUV color components to VIP input data signals, with corresponding settings of VIP_MAIN[1:0] DATA_INTERFACE_MODE register bit-field.

Table 7-2. VIP Input Data Signals to YUV Color Components Mapping

VIP Port A Data Signals	VIP Port B Data Signals	16-bit YUV422 Input Mode [1]	8-bit YUV422 Input Mode [2]
X = 1 to 2	X = 1 to 2	DATA_INTERFACE_MODE = 01b	DATA_INTERFACE_MODE = 10b
vinXa_d15	-	Y7 (MS bit)	-
vinXa_d14	-	Y6	-
vinXa_d13	-	Y5	-
vinXa_d12	-	Y4	-
vinXa_d11	-	Y3	-
vinXa_d10	-	Y2	-
vinXa_d9	-	Y1	-
vinXa_d8	-	Y0	-
vinXa_d7	vinXb_d7	Cb7/Cr7/...	Cb7/Y7/Cr7/... (MS bit)

(1) Chroma is time division multiplexed (interleaved). For more details, see Section 7.4.5.6.2, *16b Interface Mode*.
(2) Luma and Chroma are time division multiplexed (interleaved). For more details, see Section 7.4.5.6.1, *8b Interface Mode*.

Table 7-2. VIP Input Data Signals to YUV Color Components Mapping (continued)

VIP Port A Data Signals	VIP Port B Data Signals	16-bit YUV422 Input Mode (1)	8-bit YUV422 Input Mode (2)
vinXa_d6	vinXb_d6	Cb6/Cr6/...	Cb6/Y6/Cr6/...
vinXa_d5	vinXb_d5	Cb5/Cr5/...	Cb5/Y5/Cr5/...
vinXa_d4	vinXb_d4	Cb4/Cr4/...	Cb4/Y4/Cr4/...
vinXa_d3	vinXb_d3	Cb3/Cr3/...	Cb3/Y3/Cr3/...
vinXa_d2	vinXb_d2	Cb2/Cr2/...	Cb2/Y2/Cr2/...
vinXa_d1	vinXb_d1	Cb1/Cr1/...	Cb1/Y1/Cr1/...
vinXa_d0	vinXb_d0	Cb0/Cr0/... (LS bit)	Cb0/Y0/Cr0/... (LS bit)

NOTE: 16-bit RGB data can also be captured on the vinXa_d[15:0] input data bus of Port A. The 16-bit RGB data captured by the VIP_PARSER will be passed to VPDMA, as if it is a 16-bit YUV data. The VIP_MAIN[1:0] DATA_INTERFACE_MODE register bit-field must be configured for a 16-bit input mode. The VPDMA will then directly store this data in memory as a 16-bit data, provided that any 16-bit data type in the VPDMA outbound descriptor is set.

7.3 VIP Integration

This section describes the integration of the module in the device, including information about clocks, resets, and hardware requests. Figure 7-3summarizes the integration of the module in the device.

Figure 7-3. VIP Integration

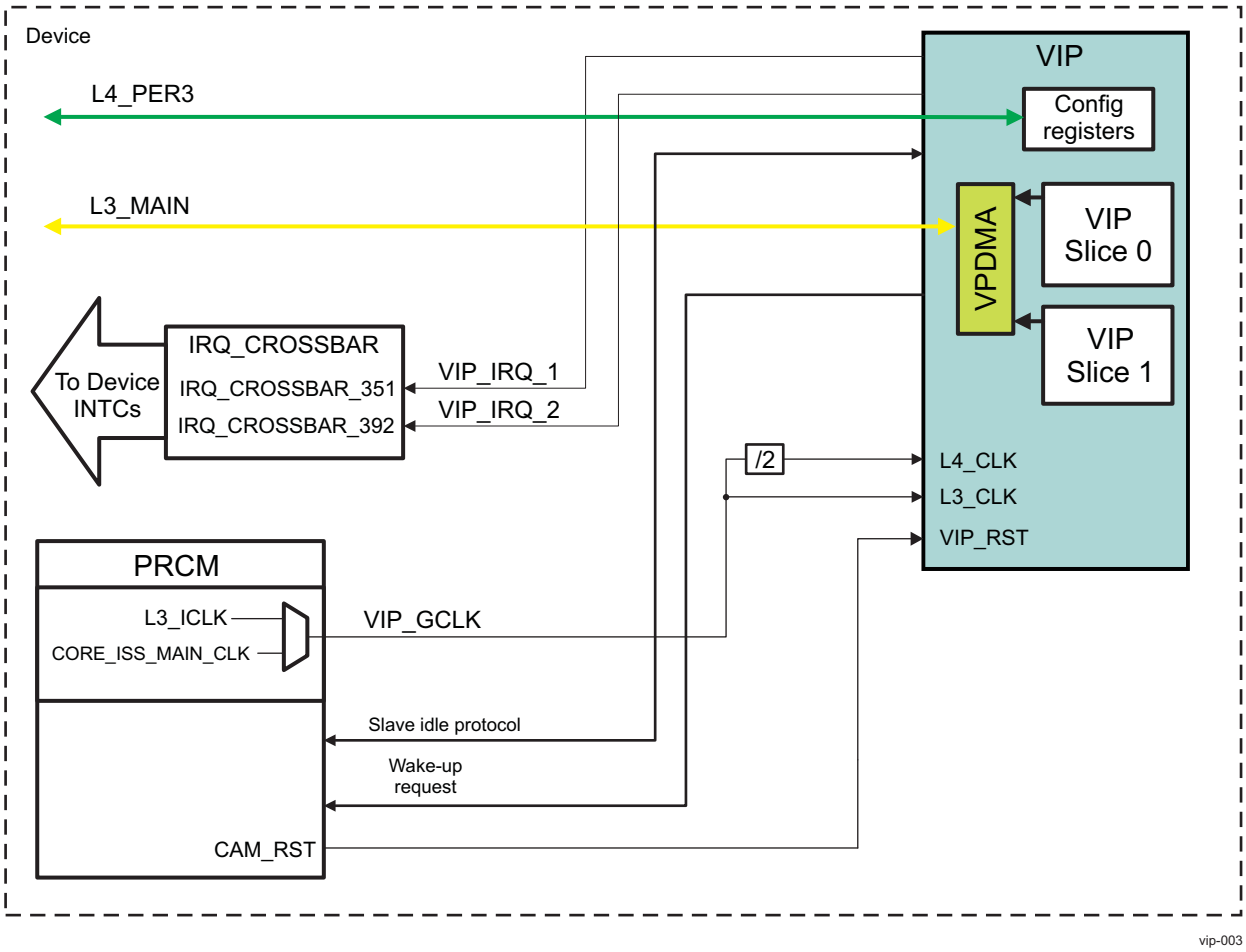


Table 7-3 and Table 7-4 list the integration attributes and clock and resets, respectively.

Table 7-3. VIP Integration Attributes

Module Instance	Attributes	
	Power Domain	Interconnect
VIP	PD_CAM	L4_PER3 for configuration L3_MAIN for data (through VPDMA module)

Table 7-4. VIP Clocks and Resets

Clocks				
Module Instance	**Destination Signal Name**	**Source Signal Name**	**Source**	**Description**
VIP	L3_CLK PROC_CLK	VIP_GCLK	PRCM	L3_CLK is the clock used to drive and receive data over the bus to L3_MAIN. The VIP subsystem uses this clock to fetch external data and transfer this data to internal processing PROC_CLK is the clock used to drive data processing within the VIP subsystem.
	L4_CLK	VIP_GCLK/2	PRCM	L4_CLK is the interface clock for Memory Mapped Registers (MMR) configuration bus
Resets				
Module Instance	**Destination Signal Name**	**Source Signal Name**	**Source**	**Description**
VIP	VIP_RST	CAM_RST	PRCM	VIP1 Reset

Table 7-5. VIP Hardware Requests

Interrupt Requests				
Module Instance	**Source Signal Name**	**Destination IRQ_CROSSBAR Input**	**Default Mapping**	**Description**
VIP	VIP_IRQ1	IRQ_CROSSBAR_351	N/A	VIP interrupt requests. These IRQ source signals are not mapped by default to any device INTC.
	VIP_IRQ2	IRQ_CROSSBAR_392	N/A	

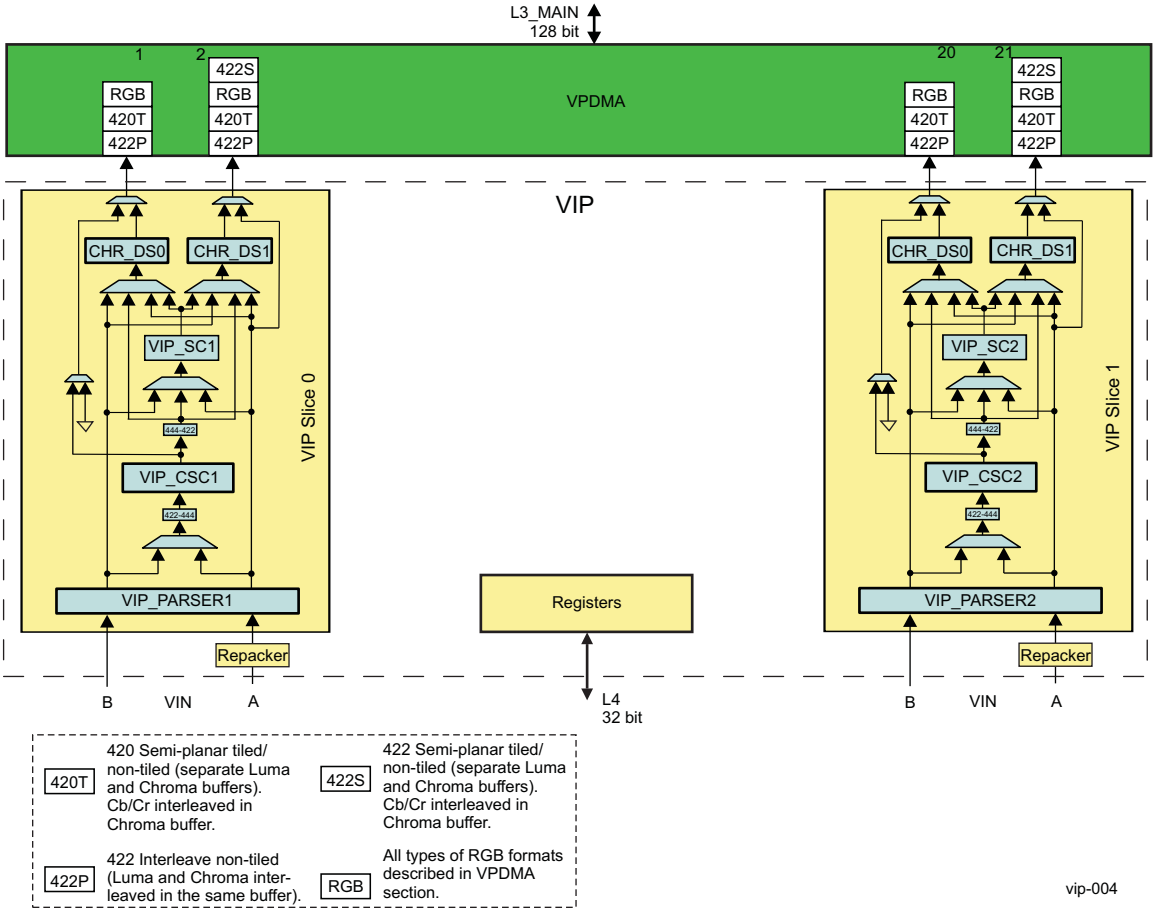
NOTE: The "**Default Mapping**" column in Table 7-5 *VIP Hardware Requests* shows the default mapping of module IRQ source signals. These module IRQ source signals can also be mapped to other lines of each device Interrupt controller through the IRQ_CROSSBAR module, respectively.
For more information about the IRQ_CROSSBAR module, see Section 13.4.6.4, *IRQ_CROSSBAR Module Functional Description*, in Chapter 13, *Control Module*.
For more information about the device interrupt controllers, see Chapter 12, *Interrupt Controllers*.

7.4 VIP Functional Description

7.4.1 VIP Block Diagram

Figure 7-4 shows the internal structure of a single VIP module in the device.

Figure 7-4. VIP Block Diagram



7.4.2 VIP Software Reset

Software reset in the VIP module can be done by setting the VIP_CLKC_RST[16] VIP1_DP_RST for Slice 0, VIP_CLKC_RST[17] VIP2_DP_RST for Slice 1, VIP_CLKC_RST[0] VPDMA_RST for VIP VPDMA to 0x1. By setting VIP_CLKC_RST[31] MAIN_RST reset is performed for all modules within the instance. Software must ensure that the software reset completes before performing operations within the VIP module.

7.4.3 VIP Power and Clocks Management

The VIP modules support the MStandby/Wait and IdleReq/SidleAck protocols as defined in Chapter 3, *Power, Reset, and Clock Management.*

Power Management within the VIP module can be accomplished in several ways:

- L4 MConnect/SConnect can disable the internal L4 clock network
- L3 MConnect/Sconnect can disable the internal L3 clock network

These items are accomplished using the standard slave idle (for L4) and master standby (for L3) protocols. When these modules are instructed to disable clocks for the internal L3 or L4 (MMR) clock domains, the internal clock networks will be shut down. This shut down applies to the clock signals - L3_CLK and L4_CLK.

7.4.3.1 VIP Clocks

The VIP internal clock domains can only be shut down by writing the appropriate register bit within the Clock Enable register - VIP_CLKC_CLKEN[16] VIP1_DP_EN for slice0, VIP_CLKC_CLKEN[17] VIP2_DP_EN for slice1 and VIP_CLKC_CLKEN[0] VPDMA_EN for the VPDMA engine

7.4.3.2 VIP Idle Mode

The VIP supports no-idle mode, force-idle mode, and smart-idle mode. The mode can be selected byprogramming the appropriate value in the VIP_SYSCONFIG[3:2] IDLEMODE bit field.

Configuration of the local target state management mode. By definition, target can handle read/write transaction as long as it is out of IDLE state.

- Force-idle mode: local target's idle state follows (acknowledges) the system's idle requests unconditionally, i.e. regardless of the IP module's internal requirements.
- No-idle mode: local target never enters idle state.
- Smart-idle mode: local target's idle state eventually follows (acknowledges) the system's idle requests, depending on the IP module's internal requirements. IP module shall not generate (IRQ- or DMA related requests) wakeup events

7.4.3.3 VIP StandBy Mode

The VIP supports no-standby mode and force-standby mode. The mode is set in the VIP_SYSCONFIG[5:4] STANDBYMODE bit field.

Configuration of the local initiator state management mode:

- Force-standby mode: local initiator is unconditionally placed in standby state.
- No-standby mode: local initiator is unconditionally placed out of standby state.

7.4.4 VIP Slice

7.4.4.1 VIP Slice Processing Path Overview

Figure 7-5 shows in details the internal processing path and output signals to VPDMA for a single VIP Slice. External video source drives the input side of the VIP Slice. Port A[x:y] can be in YUV422 format (A[15:0] in the diagram). Port B[x:y] can be in YUV422 format (B[15:0] in the diagram). Multiplexer selections and controls shown in Figure 7-5 are described in register VIP_CLKC_VIP0DPS for Slice 0, and register VIP_CLKC_VIP1DPS for Slice 1. The outputs of each VIP Slice drive the VPDMA module, which sends the resulting data to DDR memory.

NOTE: VIP_MAIN[1:0] DATA_INTERFACE_MODE bit-field must be properly configured for the expected data format on Port A.

By default, DATA_INTERFACE_MODE bit-field is set for 24-bit input data interface on Port A. It is SW responsibility to select 16-bit or 8-bit data interface, depending on the use case.

Port B input data interface is always 8-bit.

Figure 7-5. VIP Slice Processing Path Block Diagram

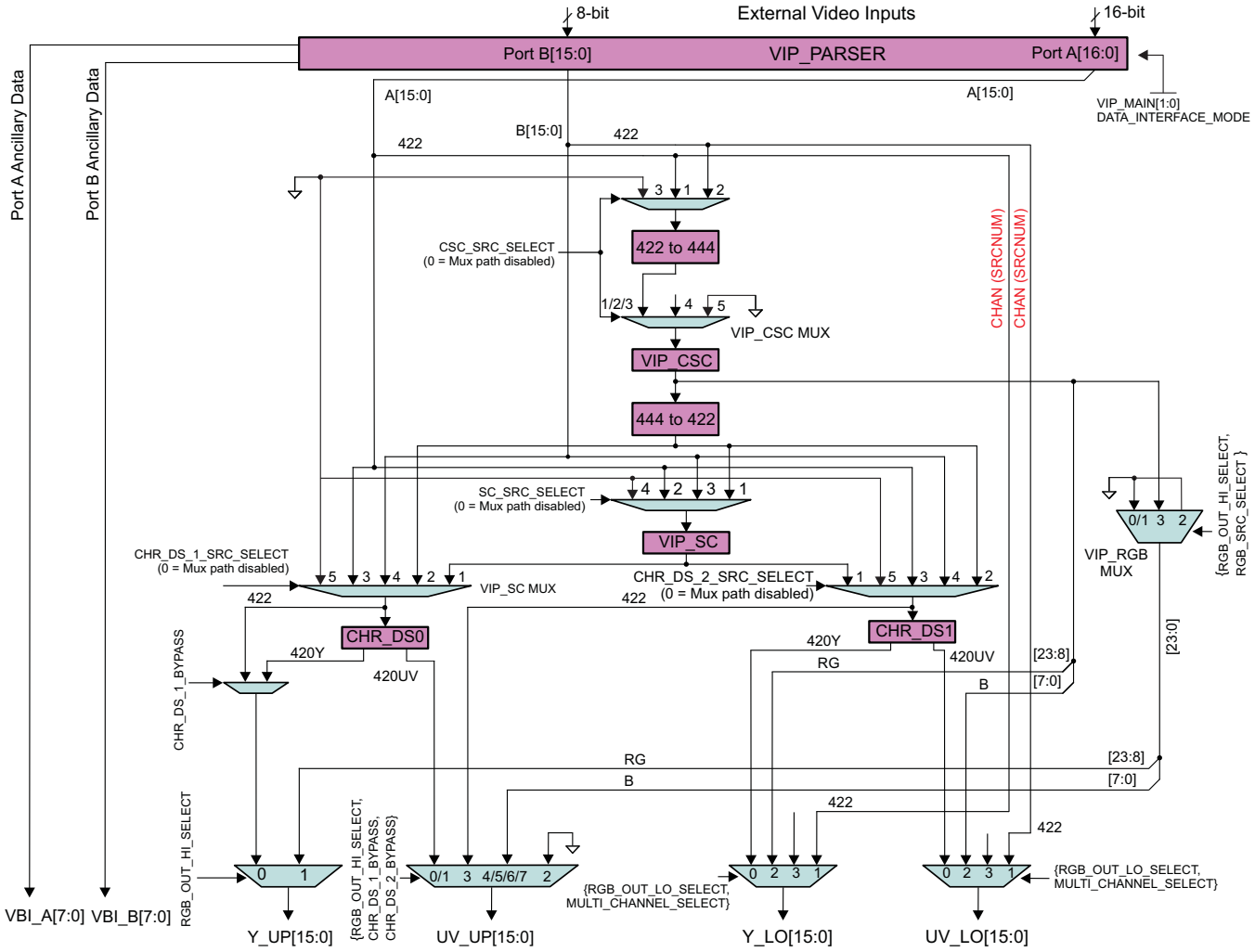


Table 7-6 provides summary of the registers controlling the multiplexers within VIP slice processing path.

Table 7-6. VIP Slice Processing Path Control

Multiplexer Control	Register Bit-fields for Slice 0	Register Bit-fields for Slice 1	Description
CSC_SRC_SELECT	VIP_CLKC_VIP0DPS[2:0] VIP1_CSC_SRC_SELECT	VIP_CLKC_VIP1DPS[2:0] VIP2_CSC_SRC_SELECT	VIP CSC Source Selection MUX
SC_SRC_SELECT	VIP_CLKC_VIP0DPS [5:3] VIP1_SC_SRC_SELECT	VIP_CLKC_VIP1DPS [5:3] VIP2_SC_SRC_SELECT	VIP SC_M Source Selection MUX
CHR_DS_1_SRC_SELECT	VIP_CLKC_VIP0DPS[11:9] VIP1_CHR_DS_1_SRC_SELECT	VIP_CLKC_VIP1DPS[11:9] VIP2_CHR_DS_1_SRC_SELECT	VIP Chroma Downsampler 1 Source Selection MUX
CHR_DS_1_BYPASS	VIP_CLKC_VIP0DPS[16] VIP1_CHR_DS_1_BYPASS	VIP_CLKC_VIP1DPS[16] VIP2_CHR_DS_1_BYPASS	VIP Chroma Downsampler 1 Bypass MUX
CHR_DS_2_SRC_SELECT	VIP_CLKC_VIP0DPS[14:12] VIP1_CHR_DS_2_SRC_SELECT	VIP_CLKC_VIP1DPS[14:12] VIP2_CHR_DS_2_SRC_SELECT	VIP Chroma Downsampler 2 Source Selection MUX
CHR_DS_2_BYPASS	VIP_CLKC_VIP0DPS[17] VIP1_CHR_DS_2_BYPASS	VIP_CLKC_VIP1DPS[17] VIP2_CHR_DS_2_BYPASS	VIP Chroma Downsampler 1 Bypass MUX

Table 7-6. VIP Slice Processing Path Control (continued)

Multiplexer Control	Register Bit-fields for Slice 0	Register Bit-fields for Slice 1	Description
RGB_OUT_HI_SELECT	VIP_CLKC_VIP0DPS[8] VIP1_RGB_OUT_HI_SELECT	VIP_CLKC_VIP1DPS[8] VIP2_RGB_OUT_HI_SELECT	VIP HI RGB Output Selection MUX
RGB_OUT_LO_SELECT	VIP_CLKC_VIP0DPS[7] VIP1_RGB_OUT_LO_SELECT	VIP_CLKC_VIP1DPS[7] VIP2_RGB_OUT_LO_SELECT	VIP LO RGB Output Selection MUX
MULTI_CHANNEL_SELECT	VIP_CLKC_VIP0DPS[15] VIP1_MULTI_CHANNEL_SELECT	VIP_CLKC_VIP1DPS[15] VIP2_MULTI_CHANNEL_SELECT	VIP Multi Channel Selection MUX
RGB_SRC_SELECT	VIP_CLKC_VIP0DPS[6] VIP1_RGB_SRC_SELECT	VIP_CLKC_VIP1DPS[6] VIP2_RGB_SRC_SELECT	VIP RGB Output Path Selection MUX

7.4.4.2 VIP Slice Processing Path Multiplexers

7.4.4.2.1 VIP_CSC Multiplexers

The following registers are controling the VIP Color Space Converter (CSC) multiplexers: VIP_CLKC_VIP0DPS[2:0] VIP1_CSC_SRC_SELECT for Slice 0 and VIP_CLKC_VIP1DPS[2:0] VIP2_CSC_SRC_SELECT for Slice 1.

The VIP_CSC block (for each slice within the VIP subsystem) receives data from one the following sources:

- VIP_PARSER Port A Output (422)
- VIP_PARSER Port B Output (422)

The default state for this multiplexer is disabled, so there is no VIP_CSC input.

7.4.4.2.2 VIP_SC Multiplexer

The multiplexer for Slice 0 and Slice 1 is controleld by VIP_CLKC_VIP0DPS[5:3] VIP1_SC_SRC_SELECT and VIP_CLKC_VIP1DPS[5:3] VIP2_SC_SRC_SELECT, respectively.

The scaler module (VIP_SC) within the VIP subsystem (for a single slice) receives data from one of the following sources:

- VIP_CSC
- VIP_PARSER Port A Output
- VIP_PARSER Port B Output

The default state for this multiplexer is disabled, so there is no VIP_SC input.

7.4.4.2.3 Output to VPDMA Multiplexers

This section is under development and is included as a placeholder for future updates.

7.4.4.3 VIP Slice Processing Path Examples

The following sections provide VIP Slice data path examples for different types of input data, and describe the corresponding multiplexer configurations. Refer to Table 7-6, *VIP Slice Processing Path Control*, for mapping of the multiplexer controls to registers.

In the block diagrams of the following sections:

- Output A refers to the result of processing Input A data.
- Output B refers to the result of processing Input B data.

7.4.4.3.1 Input: B=YUV422; Output: B=RGB

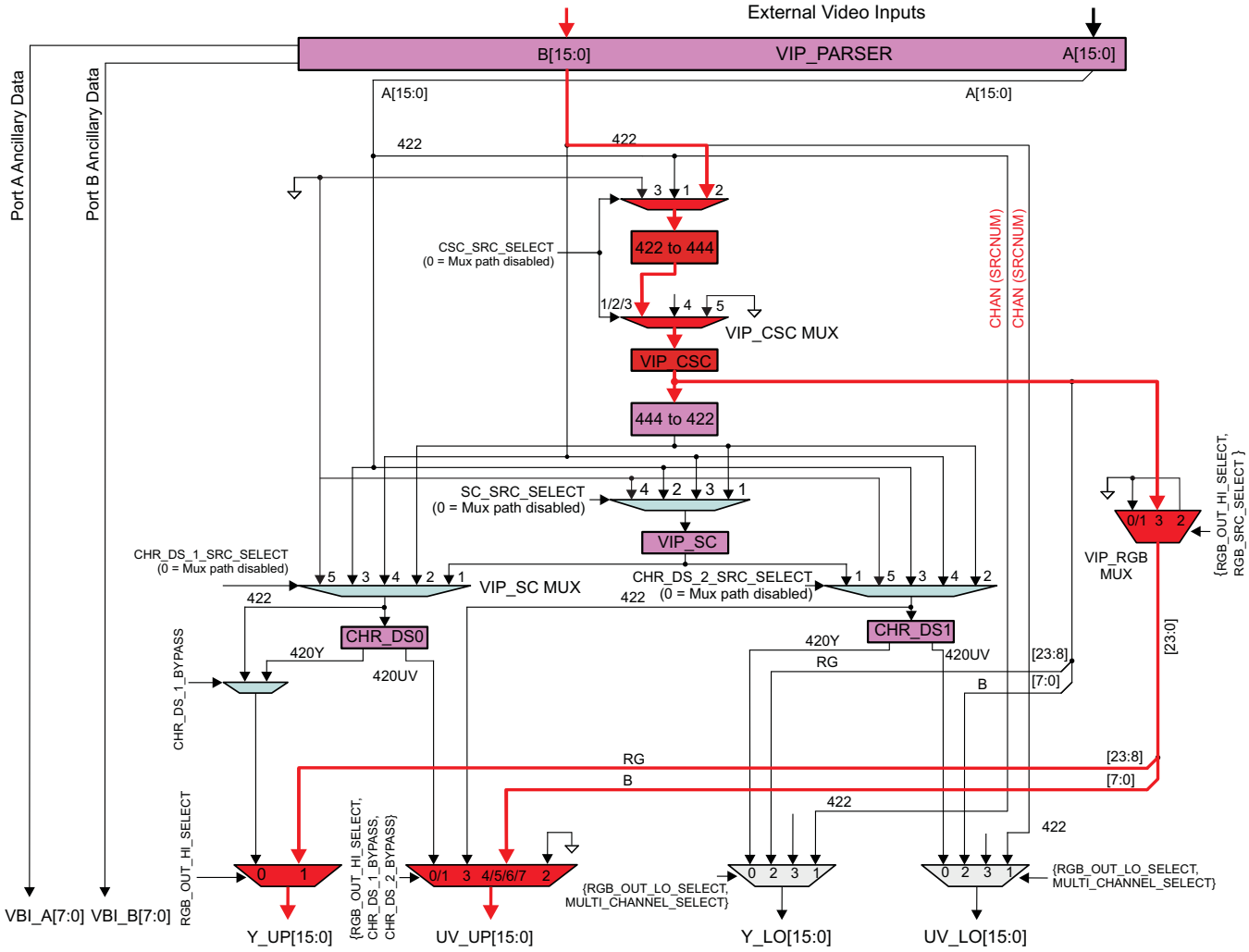
Tested in single channel embedded and discrete mode.

Input: B=YUV422; Output: B=RGB

Multiplexers settings.

- VIPx_CSC_SRC_SELECT = 2
- VIPx_SC_SRC_SELECT = 0
- VIPx_CHR_DS_1_SRC_SELECT = 0
- VIPx_CHR_DS_1_BYPASS = 0
- VIPx_CHR_DS_2_SRC_SELECT = 0
- VIPx_CHR_DS_2_BYPASS = 0
- VIPx_RGB_SRC_SELECT = 1
- VIPx_RGB_OUT_HI_SELECT = 1

Figure 7-6. Input: B=YUV422; Output: B=RGB



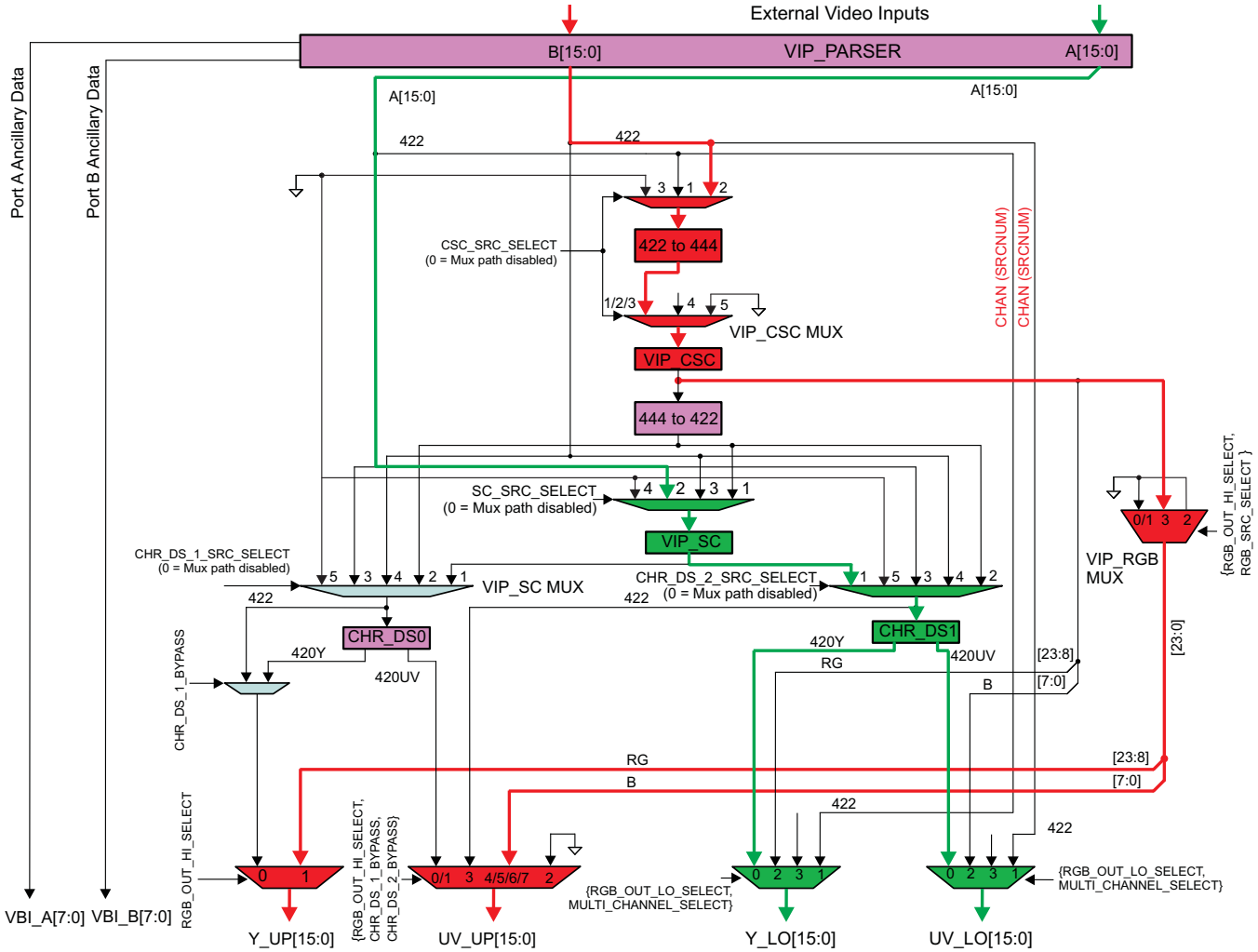
7.4.4.3.2 Input: A=YUV422 8/16, B=YUV422; Output: A=Scaled YUV420, B=RGB

Tested in single channel embedded and discrete mode.

Multiplexers settings:

- VIPx_CSC_SRC_SELECT = 2
- VIPx_SC_SRC_SELECT = 2
- VIPx_CHR_DS_1_SRC_SELECT = 0
- VIPx_CHR_DS_1_BYPASS = 0
- VIPx_CHR_DS_2_SRC_SELECT = 1
- VIPx_CHR_DS_2_BYPASS = 0
- VIPx_RGB_SRC_SELECT = 1
- VIPx_RGB_OUT_HI_SELECT = 1
- VIPx_RGB_OUT_LO_SELECT = 0
- VIPx_MULTI_CHANNEL_SELECT = 0

Figure 7-7. Input: A=YUV422 8/16, B=YUV422; Output: A=Scaled YUV420, B=RGB



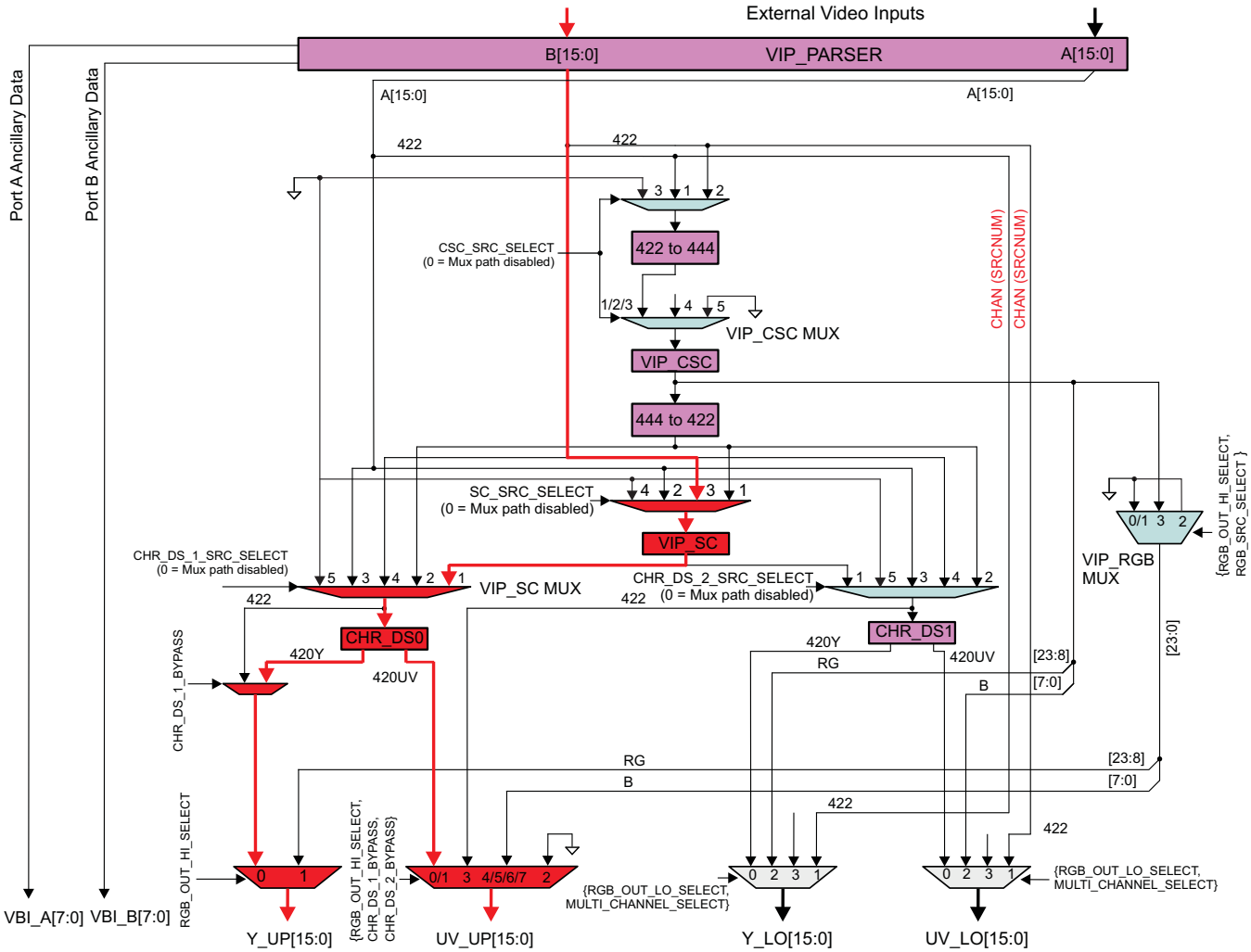
7.4.4.3.3 Input: B=YUV422; Output: B=Scaled YUV420

Tested in single channel embedded and discrete mode.

Multiplexers settings:

- VIPx_CSC_SRC_SELECT = 4
- VIPx_SC_SRC_SELECT = 3
- VIPx_CHR_DS_1_SRC_SELECT = 1
- VIPx_CHR_DS_1_BYPASS = 0
- VIPx_CHR_DS_2_SRC_SELECT = 1
- VIPx_CHR_DS_2_BYPASS = 1
- VIPx_RGB_SRC_SELECT = 0
- VIPx_RGB_OUT_HI_SELECT = 0

Figure 7-8. Input: B=YUV422; Output: B=Scaled YUV420



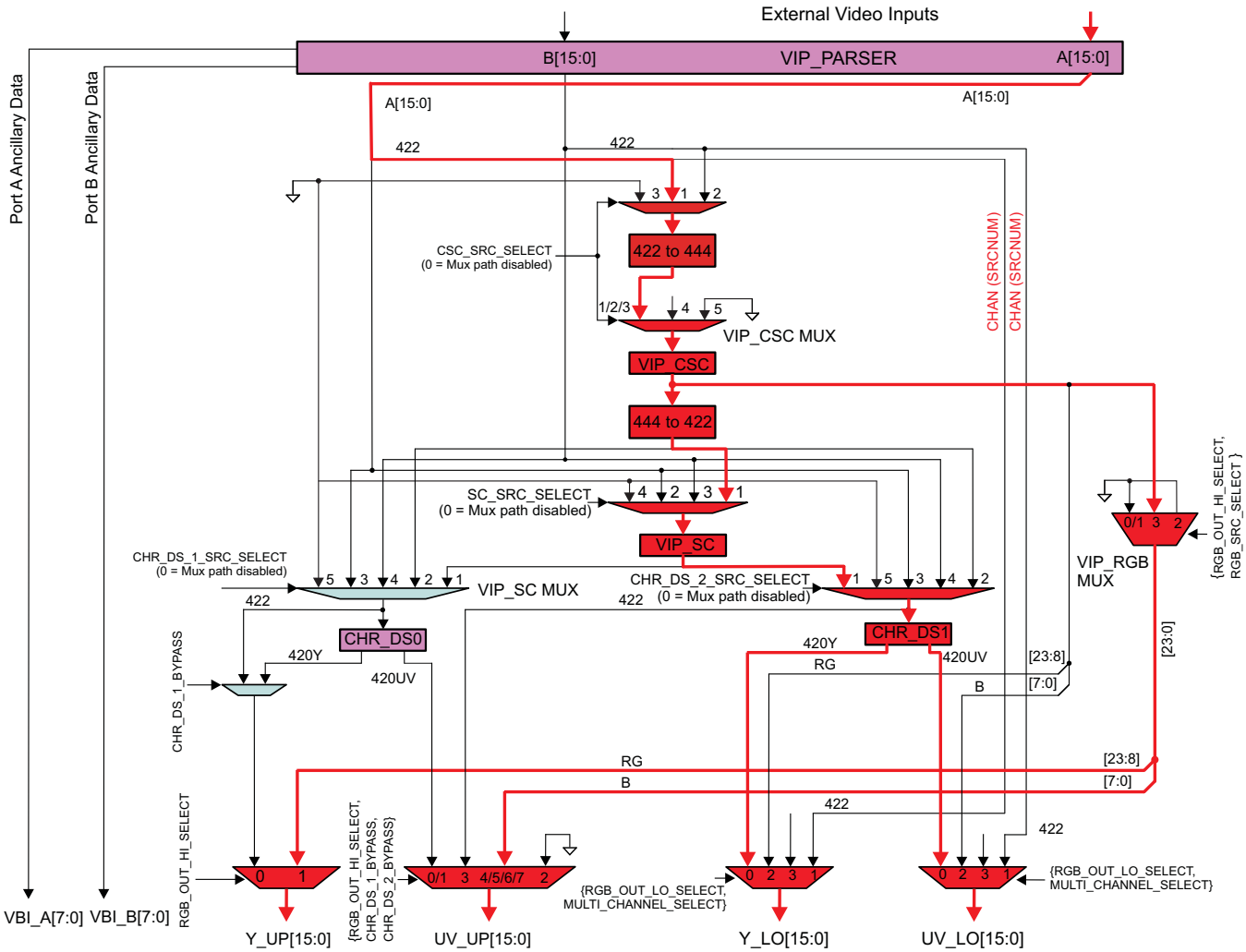
7.4.4.3.4 Input: A=YUV422 8/16; Output: A=Scaled YUV420, A=YUV444

Tested in single channel embedded and discrete mode.

Multiplexer settings:

- VIPx_CSC_SRC_SELECT = 1
- VIPx_SC_SRC_SELECT = 1
- VIPx_CHR_DS_1_SRC_SELECT = 0
- VIPx_CHR_DS_1_BYPASS = 0
- VIPx_CHR_DS_2_SRC_SELECT = 1
- VIPx_CHR_DS_2_BYPASS = 0
- VIPx_RGB_SRC_SELECT = 1
- VIPx_RGB_OUT_HI_SELECT = 1
- VIPx_RGB_OUT_LO_SELECT = 0
- VIPx_MULTI_CHANNEL_SELECT = 0

Figure 7-9. Input: A=YUV422 8/16; Output: A=Scaled YUV420, A=YUV444



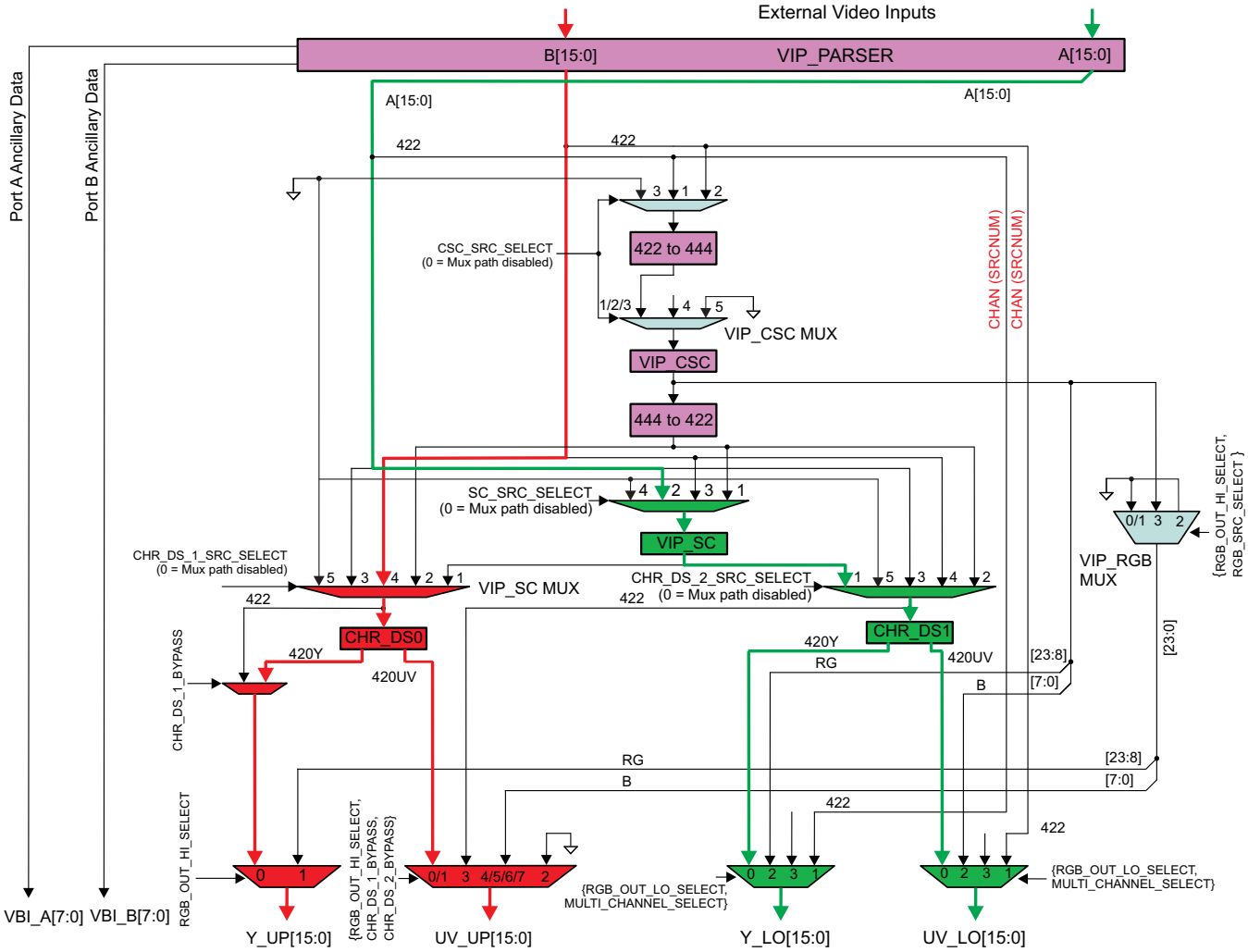
7.4.4.3.5 Input: A=YUV422 8/16, B=YUV422; Output: A=Scaled YUV420, B=YUV420

Tested in single channel embedded and discrete mode.

Multiplexer settings:

- VIPx_CSC_SRC_SELECT = 0
- VIPx_SC_SRC_SELECT = 2
- VIPx_CHR_DS_1_SRC_SELECT = 4
- VIPx_CHR_DS_1_BYPASS = 0
- VIPx_CHR_DS_2_SRC_SELECT = 1
- VIPx_CHR_DS_2_BYPASS = 0
- VIPx_RGB_SRC_SELECT = 0
- VIPx_RGB_OUT_HI_SELECT = 0
- VIPx_RGB_OUT_LO_SELECT = 0
- VIPx_MULTI_CHANNEL_SELECT = 0

Figure 7-10. Input: A=YUV422 8/16, B=YUV422; Output: A=Scaled YUV420, B=YUV420



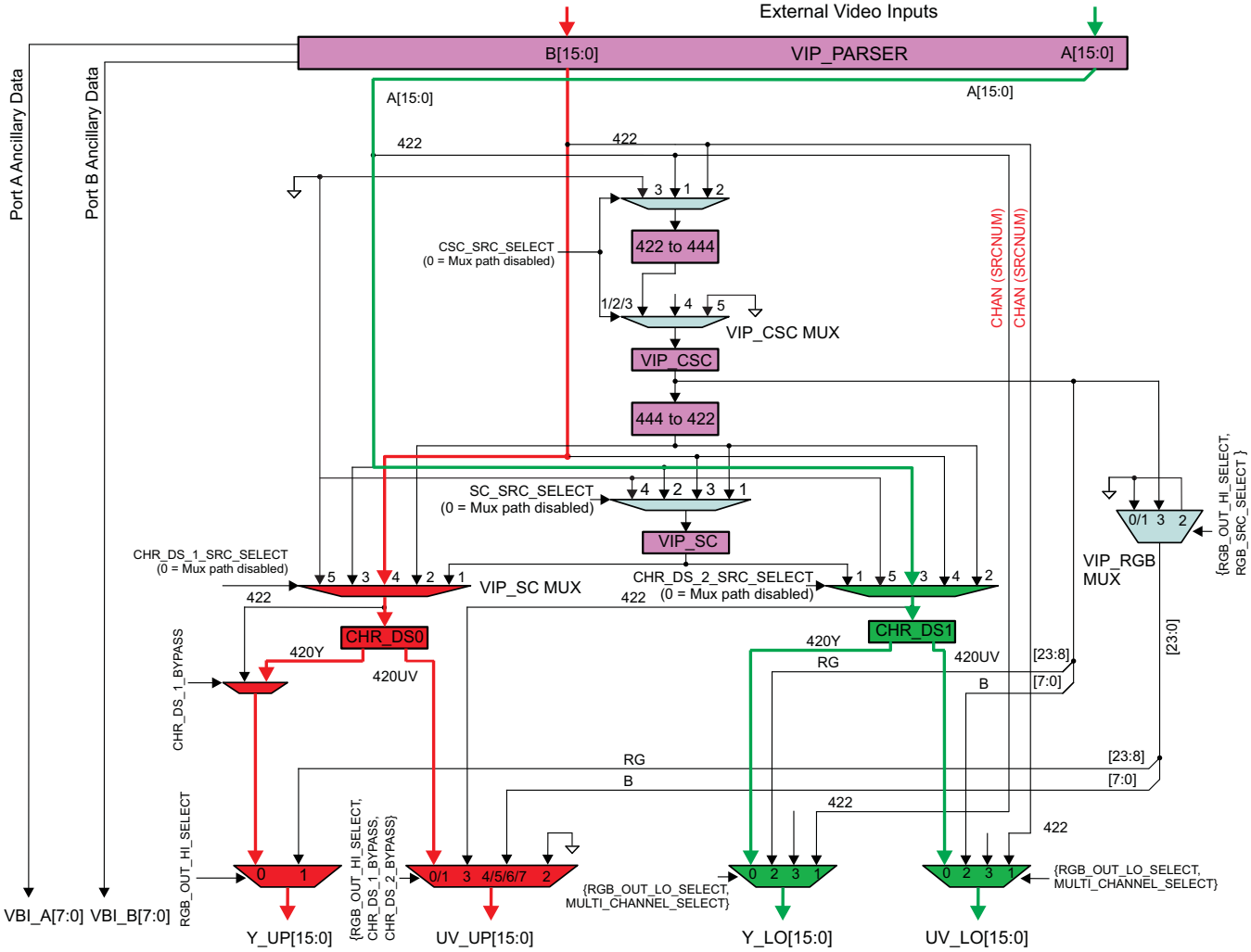
7.4.4.3.6 Input: A=YUV422 8/16, B=YUV422; Output: A=YUV420, B=YUV420

Tested in single channel embedded and discrete mode.

Multiplexer settings:

- VIPx_CSC_SRC_SELECT = 0
- VIPx_SC_SRC_SELECT = 0
- VIPx_CHR_DS_1_SRC_SELECT = 4
- VIPx_CHR_DS_1_BYPASS = 0
- VIPx_CHR_DS_2_SRC_SELECT = 3
- VIPx_CHR_DS_2_BYPASS = 0
- VIPx_RGB_SRC_SELECT = 0
- VIPx_RGB_OUT_HI_SELECT = 0
- VIPx_RGB_OUT_LO_SELECT = 0
- VIPx_MULTI_CHANNEL_SELECT = 0

Figure 7-11. Input: A=YUV422 8/16, B=YUV422; Output: A=YUV420, B=YUV420



7.4.5 VIP Parser

The VIP Parser (VIP_PARSER) module is used to capture the external video data into the VIP module.

7.4.5.1 Features

Each VIP module contains two VIP_PARSER modules (one VIP_PARSER per slice).

For a single VIP_PARSER, the video capture functions include:

- Two Pixel Clock Input Domains are supported (Port A and Port B):
  - Each Pixel Clock Input Domain has separate clock and framing signals.
  - Each Pixel Clock Input Domain can support embedded (BT.656/1120 style in 16-bit, or BT.656 in 8-bit) or discrete (BT.601 style) sync.
  - Pixel Clock Input Domain Port B supports one 8-bit input data bus. Port A supports one 16-bit input data bus. At device level, the same device pads may be shared between Port A and Port B. For more information, see the multiplexing characteristics in device Data Manual.
- Embedded Sync data interface mode supports single or multiplexed sources;
- Discrete Sync data interface mode supports only single source input;
- The two Pixel Clock Input Domains can be individually configured in any combination of Embedded or Discrete Sync;
- Vertical Ancillary Data capture is supported for each input source;
- A maximum of 8 + 1 (8 normal line sources + 1 split-line source) multiplexed sources are supported for a single Pixel Clock Input Domain using TI Line Mux Mode;
- Multiplexed data can only appear in embedded sync mode;
- Where possible, blanking pixels that may contain embedded vertical ancillary data will be stored in a dedicated buffer per each video source;
- Optional selection of channel (Luma or Chroma, or both) from which Vertical Ancillary data is extracted for YUV422 source;
- Ancillary Data can appear in the Horizontal Blanking as well as the Vertical Blanking. Typically, only Vertical Blanking Ancillary Data is captured. However, Horizontal Blanking Ancillary Data can be captured as well using HSYNC style discrete sync capture mode;
- Video up to WUXGA (1920 × 1200) can be supported using Port A in 16-bit mode.

7.4.5.2 Repacker

The Repacker module rearranges the input bit ordering of the 16-bit data bus on Port A of each VIP slice. This module allows external input data to be presented to the datapath such that various data packing formats can be achieved in memory. As shown in Figure 7-12, a Repacker exists for each 16-bit input port.

The Repacker module is a simple multiplexer that serves to move input bits to different locations on the its output bus to VIP Parser. Figure 7-12 shows the supported bytelane swapping modes corresponding to different VIP_XTRA_PORT_A[30:28] REPACK_SEL settings.

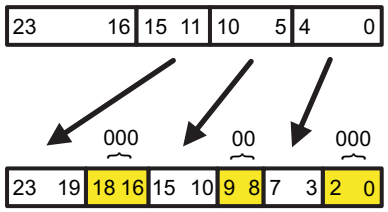
Figure 7-12. Bytelane Swapping Modes

REPACK_SEL = 0, Straight Through Mode

REPACK_SEL = 3, Center/Right Swap Mode

16-bit RAW data entering the VIP subsystem is packed as a contiguous input bus from bits 15 to 0. This 16-bit RAW input must be remapped to the RGB565 format, so that it can be saved to DDR memory properly, because the VPDMA does not support the RAW16 input format natively. Instead, the RAW 16 format is first remapped as RGB565 data and then given to the VPDMA. Figure 7-13 describes the VIP_XTRA_PORT_A[30:28] REPACK_SEL = 6 option to remap a contiguous [15:0] RAW input data bus to a RGB565 compliant output bus. This RAW16 data will use RGB565 data types in the VPDMA Data Descriptors.

Figure 7-13. RAW16 to RGB565 Mapping



REPACK_SEL = 6, RAW16 Mapping Mode

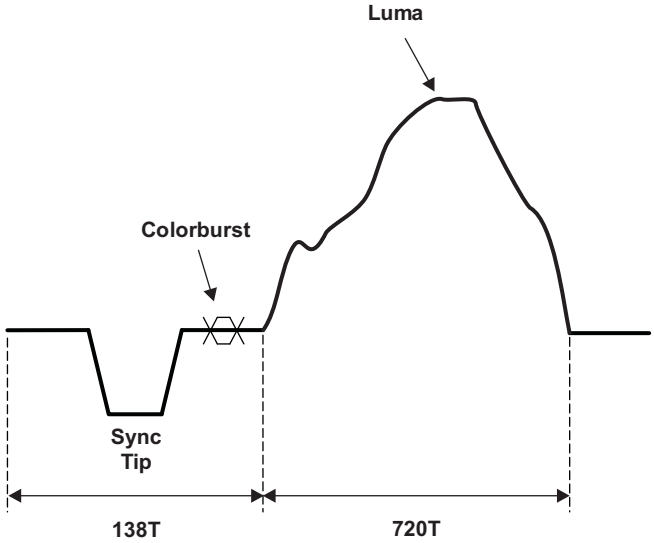
NOTE: There is no repacker for the 8-bit input port (Port B of each VIP slice).

The RAW16 mapping mode does not work for embedded sync streams.

7.4.5.3 Analog Video

A digital interface stream is based on analog video. The waveform for a line of NTSC analog video is shown in Figure 7-14.

Figure 7-14. NTSC Analog Video Waveform for One Horizontal Line



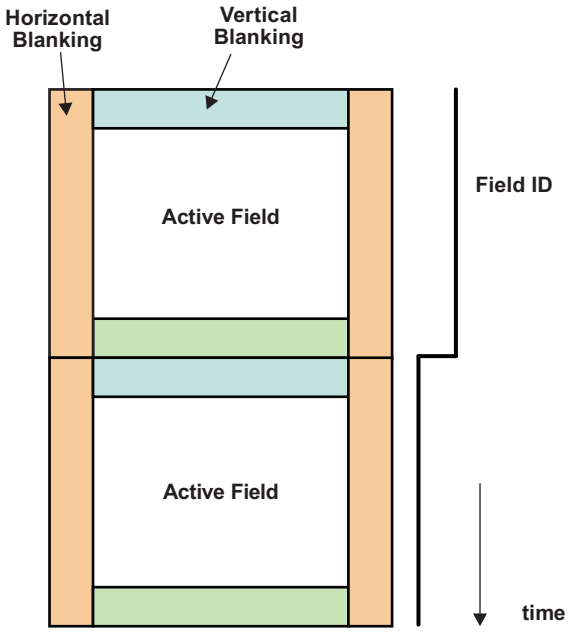
T is a time constant. For NTSC, T=1/13.5 MHz = 74ns.

7.4.5.4 Digitized Video

Digitized video is based on scan lines in found in analog video. BT.601 uses various sync signals to specify when a new field and a new line starts. BT.656 and BT.1120 uses sync words embedded in the data stream to specify start of field and start of line.

An image can be digitized into regions shown in Figure 7-15.

Figure 7-15. Digitized Video



With the capability to encode sync words inside the data stream, there is more flexibility for adding non-video related data, called Ancillary Data. Also, code words embedded in the digital stream can be used as a type of identifier for multiplexing several sources of video into one data stream.

Figure 7-16 shows End-of-Active-Video (EAV) and Start-of-Active-Video (SAV) code words added to a video transmission. The period between the EAV and SAV is equivalent to Horizontal Blanking. The period between the SAV to the next EAV is active video or vertical blanking.

In the BT.656 or BT.1120 embedded code word scheme, three bits of the EAV/SAV code word are important: F (field), H (horizontal blanking), and V (vertical blanking).

Figure 7-16. Code Word Embedded Video Format

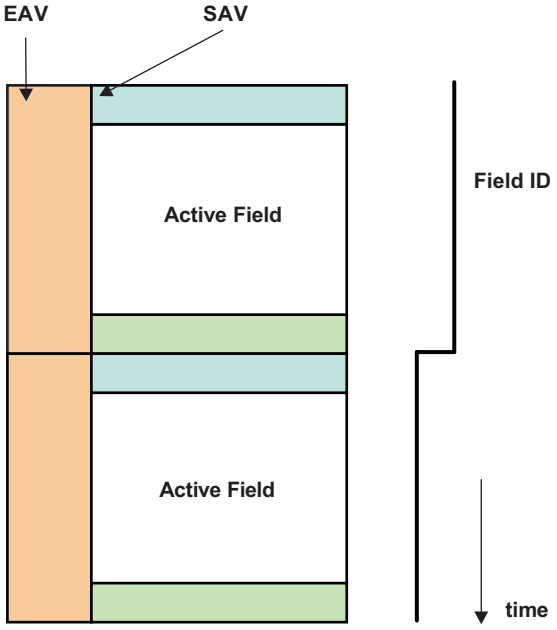
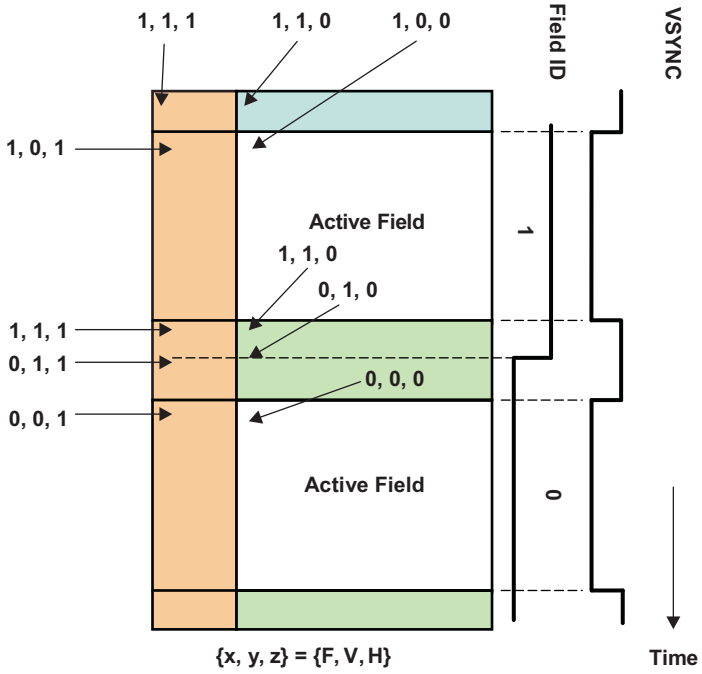


Figure 7-17 shows the values of F, V, and H flags at different locations in the picture. The Field flag represents the state of the Field ID for the picture. For progressive frames, F is always '0.' The V flag specifies vertical blanking areas. The H flag specifies horizontal blanking portions of the picture.

Figure 7-17. Digitized Video with F, V, and H Flags in EAV/SAV



7.4.5.5 Frame Buffers

The VIP/VPDMA support Frame Buffers in DDR memory for Active Video and Ancillary Data.

4:2:2 data is always saved in packed pixel buffers.

4:2:0 data is saved in Planar Luma buffers and Planar CbCr pair buffers.

A Luma Frame Buffer is a Planar storage area. Each line is the width in pixels (1Byte/pixel) of the output picture size format. The frame buffer contains the number of active video lines in the output picture size format.

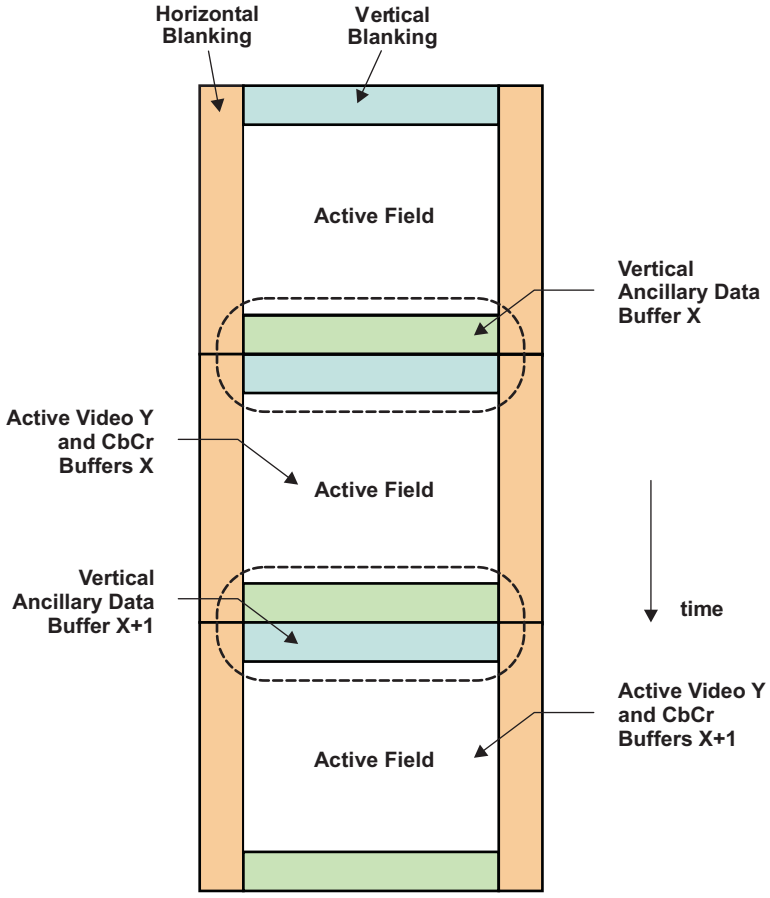
A Chroma Pair Frame Buffer is Planar storage of CbCr pixel pairs, with each pixel being a byte. For 4:2:0 storage, N lines in the output picture active video results in N/2 lines of CbCr pairs being stored.

The Ancillary Data buffer is different than the Active Video Frame Buffers. The Ancillary Data buffer only stores Vertical Blanking Ancillary Data. The number of lines in the Ancillary Data buffer is the same as the number of Vertical Blanking lines. Typically, only one channel is extracted from the Vertical Blanking data, so the width of the Ancillary Data buffer is the same as the width of the Luma Buffer.

In 8-bit input mode, it is possible for both Luma and Chroma sites to be extracted for Vertical Ancillary data. Each color component is strobed on separate input clock cycles. In this case, the line width of the Ancillary data is twice the Luma line width of the picture. Both Luma and Chroma sites cannot be extracted for 16-bit input mode because both Luma and Chroma are sent on the same input clock cycle and the Ancillary port to the VPDMA VPI is only 8 bits wide.

Figure 7-18 shows how the planar data regions are stored in DDR memory. The vertical blanking data is stored in a set of Planar Buffers. Note that the bottom of the Vertical Ancillary Data from the previous field or frame is stored in the same buffer as the top of the Vertical Ancillary Data from the current field or frame.

Figure 7-18. Planar Buffer Storage Description



The Luma representing Active Video is stored in a set of Planar Buffers. The CbCr Chroma Pairs are stored in a set of Planar Buffers.

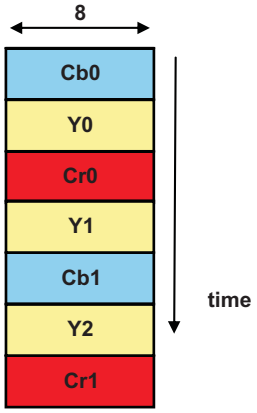
7.4.5.6 Input Data Interface

This section describes how the data (luma and chroma data for YUV422 format capture) is muxed for the various interface modes.

7.4.5.6.1 8b Interface Mode

In 8-bit data interface mode, the input pixels are multiplexed according to Figure 7-19. The Chroma Format is 4:2:2. Sites with Cb/Cr pixels are known as Chroma sites. Those sites with Y pixels are known as Luma sites. The 8-bit data interface mode is selected by setting VIP_MAIN[1:0] DATA_INTERFACE_MODE bit-field to 0x2.

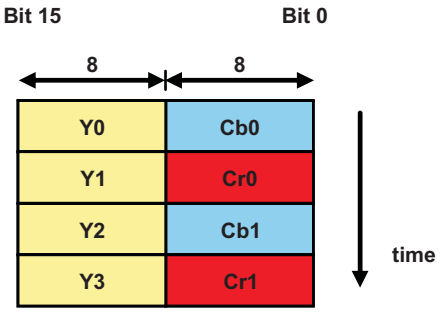
Figure 7-19. 8-bit Interface Discrete Sync Pixel Multiplexing



7.4.5.6.2 16b Interface Mode

In 16-bit interface mode, Luma is on 8 MSB bits of the data bus and Cb/Cr chroma pixels alternate on the other 8 bits of the data bus as shown in Figure 7-20. The 16-bit data interface mode is selected by setting VIP_MAIN[1:0] DATA_INTERFACE_MODE bit-field to 0x1.

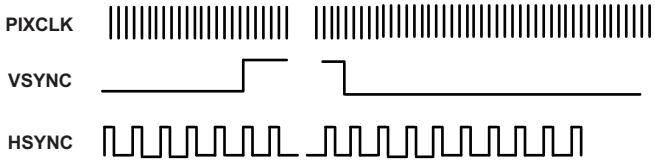
Figure 7-20. 16-bit Interface Discrete Sync Pixel Multiplexing



7.4.5.6.3 Signal Relationships

A digital representation of video can be realized by using HSYNC and VSYNC signals to identify frame start and line start. Suppose HSYNC and VSYNC are active high, Figure 7-21 shows the general relationship of these signals.

Figure 7-21. Discrete Sync Signals



Every PIXCLK cycle carries either an active pixel or a blanking pixel. VSYNC pulses between two fields (or frames, in the case of progressive video). HSYNC pulses to signify the beginning of every line. An ACTVID signal can be used as a data valid to specify active video.

Discrete Sync cannot be used with any multi-camera multiplexed stream inputs. In the device, if Port A is configured for 24-bit discrete sync, then Port B must be disabled since there are no more data input pins left over for Port B.

If Port A is not 24 bits, then the 8-bit Port B can be configured and enabled for either discrete or embedded sync.

7.4.5.6.4 General 5 Pin Interfaces

Discrete Sync signal handling varies among different sending devices. The information that must be conveyed includes the pixel data value, field ID, horizontal blanking, and vertical blanking. Many devices can be configured to adjust the timing of the signals relative to each other.

In this section, DATA will be depicted as 8 bits. However, discrete sync does optionally support 16-bit data input. Type 1 is named after a generic five pin interface between the sending and receiving devices.

In Figure 7-22, P0 represents the first pixel in the horizontal blanking interval following the last vertical blanking line of the previous field or frame. HSYNC specifies the horizontal blanking region and VSYNC specifies that the P0 pixel is in the vertical sync area. HSYNC can be a strobe that is active one or more cycles and can deassert before the actual end of horizontal blanking or HSYNC may be active for the full duration of horizontal blanking.

Likewise, VSYNC can be a strobe that is active one or more cycles and can deassert before the actual end of vertical blanking or VSYNC may be active for the full duration of vertical blanking.

FID can change at this pixel or it may change later. For interlaced source, though, the FID will be inverted for this pixel at the same time point in the next field. So, it does not really matter when FID is captured. Many sending devices allow the location of FID changes to be programmable.

In this diagram and all others in this document, the active polarities of the interface signals can be either high or low. For the sake of uniformity in this document, all polarities are drawn active high. Also, different vendors have different datasheet names for the interface signals.

Figure 7-22. Type 1, First Horizontal Blanking Pixel

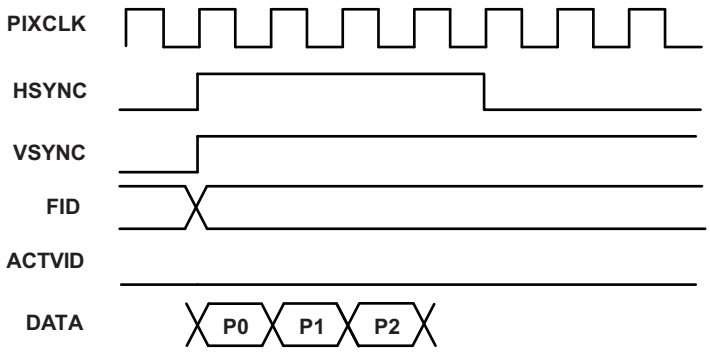
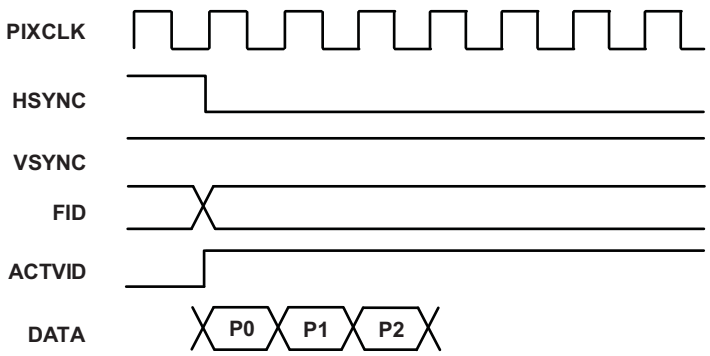


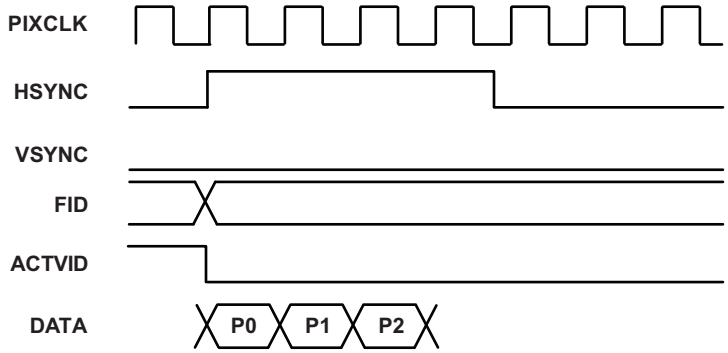
Figure 7-23 shows the P0 pixel being the first Chroma Channel data value in the Vertical Ancillary Data region. HSYNC is definitely de-asserted by now since P0 is no longer in horizontal blanking. ACTVID may or may not be active for Vertical Ancillary Data. Some devices consider these pixels to be Active (as in non-horizontal blanking). Other devices consider only video to be ACTIVE Video.

Figure 7-23. Type 1, First Vertical Ancillary Data Pixel



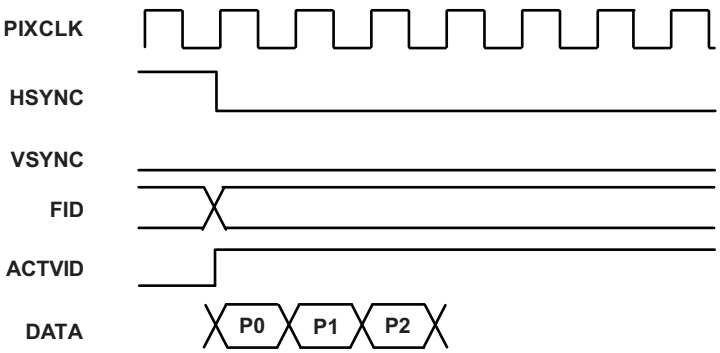
Following the vertical blanking region, the video portion of the field or frame starts. Figure 7-24 shows the horizontal blanking area in this video portion of the field or frame. P0 is the first pixel in the horizontal blanking. HSYNC is active for one or more pixel clocks. VSYNC is inactive in this video area. FID can change here. ACTVID is low since P0 is horizontal blanking.

Figure 7-24. Type 1, Horizontal Blanking in Video Region



In Figure 7-25, P0 represents the first Chroma pixel in the Active video line. HSYNC is inactive, since P0 is in the active video region. Likewise, VSYNC is inactive. FID may or may not change here. ACTVID is high to signal capturing of video pixels.

Figure 7-25. Type 1, First Video Pixel



7.4.5.6.5 Signal Subsets—4 Pin VSYNC, ACTVID, and FID

A sending device may use only a subset of the signals described in Section 7.4.5.6.3. The sending device just needs to convey important signals required to capture the field or frame. It can be shown that various selections of four pins can be used to satisfy all Type 1 conditions.

Three pins, VSYNC, ACTVID, and FID, plus a pixel clock can be used to support discrete sync. VSYNC would bump the capture buffer. An inactive to active level of ACTVID specifies a line of data to capture. FID determines the field ID polarity. The scenario in which the sending device wants the receiving end to capture Vertical Ancillary Data using 4-pin signaling is shown in Figure 7-26.

Figure 7-26. 4-Pin Reduced ACTVID Signaling with Vertical Ancillary Data

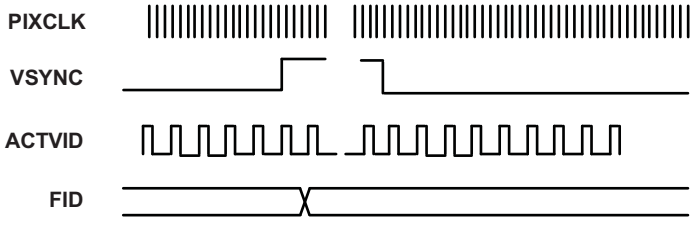
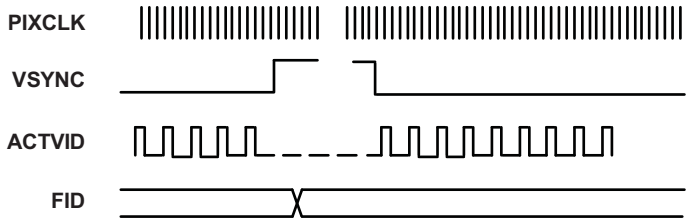


Figure 7-27 describes the case using the 4-pin interface in which the sending device does not send Vertical Ancillary Data.

Figure 7-27. 4-Pin Reduced ACTVID Signaling with No Vertical Ancillary Data

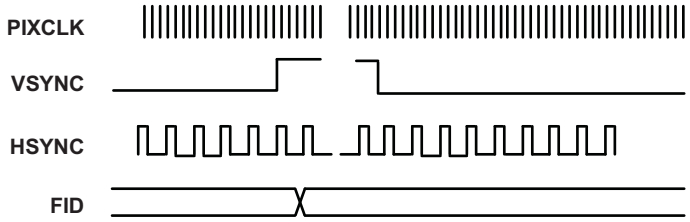


7.4.5.6.6 Signal Subsets—4 Pin VSYNC, HSYNC, and FID

In this style of Discrete Sync, as shown in Figure 7-28, four pins are used including the Pixel Clock. HSYNC signals the beginning of the line. All data in the line is captured, including Horizontal Blanking Data. In fact, this signaling mode is the only one which allows Horizontal Blanking Data to be captured.

Of course, by capturing the horizontal blanking pixels in the frame buffers, there is no way to be certain exactly where the blanking ends and the active video starts. One would have to rely solely on video format specs to find the active video inside the frame buffer.

Figure 7-28. 4-Pin Reduced HSYNC Signaling with Vertical Ancillary Data



7.4.5.6.7 Vertical Sync

Vertical Sync is used to indicate lines that are in the vertical blanking interval. The VSYNC also separates fields or frames. To be spec compliant, VSYNC should be active for a few lines at the bottom of the picture. The exact number of vertical blanking lines at the bottom depends on the specification for the picture format (480i, 480p, 720p, 1080i, 1080p).

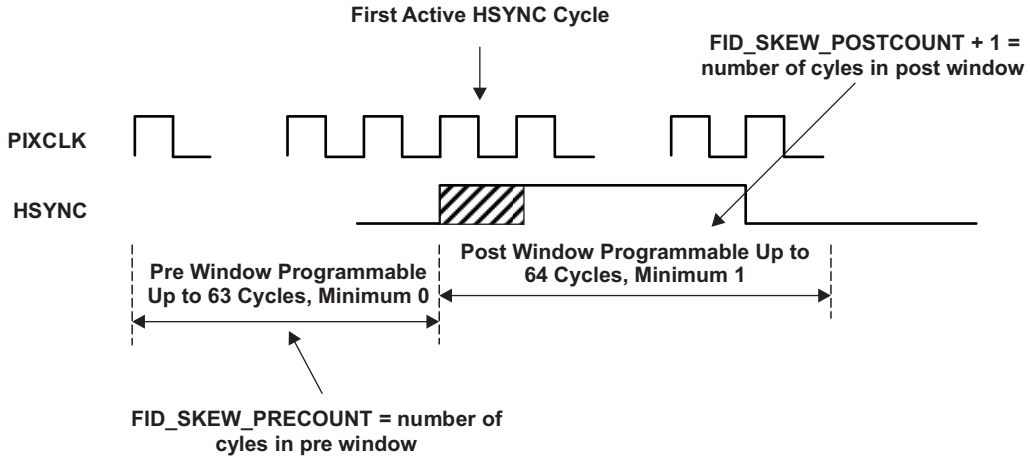
Likewise, the number of vertical blanking lines at the top of the picture depends on the video format specification corresponding to the incoming picture.

In the VIP_PARSER, lines associated with an Active VSYNC are stored in the vertical ancillary data buffers using the ANC VPI port to the VPDMA. Lines without an Active VSYNC are stored in the active video Luma and Chroma-pair buffers using the Y and UV VPI ports, respectively, to the VPDMA.

When using HSYNC signaling instead of ACTVID, the VSYNC signal may be derived from an analog source such as an NTSC/PAL decoder. In this case, VSYNC may not transition on the exact cycle as HSYNC. Thus, the VIP_PARSER supports a window region around HSYNC in which VSYNC transitions will be detected. A VSYNC transition occurring within the window is identified the same way as if VSYNC transitioned on the same cycle as HSYNC going active.

The window is defined by a pre-window, which is determined by the VIP_PORT_A[21:16] FID_SKEW_PRECOUNT and VIP_PORT_B[21:16] FID_SKEW_PRECOUNT registers. There is also a post-window that is defined by VIP_PORT_A[29:24] FID_SKEW_POSTCOUNT and VIP_PORT_B[29:24] FID_SKEW_POSTCOUNT for port B. Note that although the configuration registers are named FID_SKEW, they are also used for defining the VSYNC transition window. The window region definition is shown in Figure 7-29.

Figure 7-29. VSYNC Pre and Post Window



The results of VSYNC behavior in the transition window are shown in Figure 7-30. A low to high transition in the window is equivalent to VSYNC going low to high on the active HSYNC cycle.

Likewise, a high to low transition in the window is equivalent to VSYNC going high to low on the active HSYNC cycle.

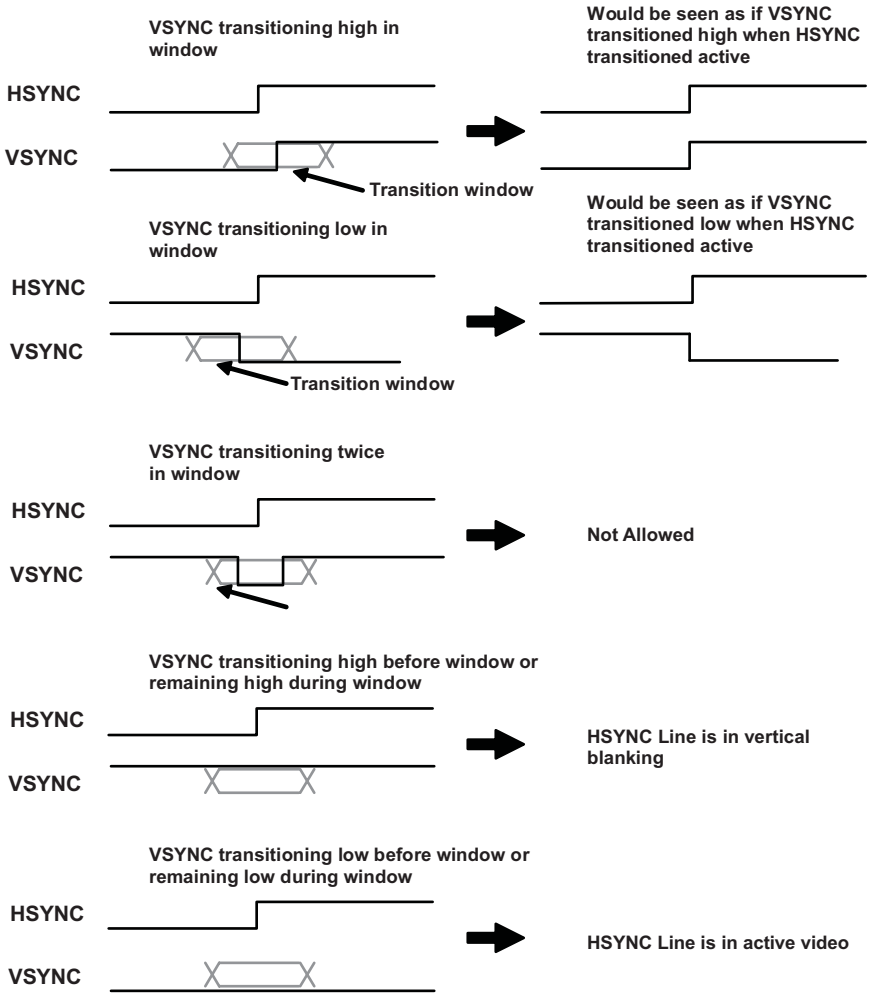
Two transitions of VSYNC within the VSYNC window is not allow and is undefined behavior.

If VSYNC is high throughout the transition window, then the HSYNC line is in vertical blanking.

If VSYNC is low throughout the transition window, then the HSYNC line is in active video.

Note that VSYNC skew generally only applies to input signals that have been sampled from an analog source, as in a NTSC/PAL decoder. If the VSYNC is a VBLANK-type signal or if the sending device is another all-digital IC, then the VSYNC signal does not have a skew since VSYNC will be aligned to HSYNC. In this case, setting FID_SKEW_PRECOUNT = ‘0’ and FID_SKEW_POSTCOUNT = ‘0’ (within PORT_A and PORT_B registers) defines a minimum size window which will capture the value of VSYNC on the same cycle that HSYNC goes active.

Figure 7-30. VSYNC Equivalence When Using Transition Window



7.4.5.6.8 Field ID Determination Using Dedicated Signal

For Progressive Source, FIELD ID is always '0.'

For Interlaced Source, FIELD ID needs to be extracted consistently.

In some cases, vertical sync is active on the first pixel of a line in the vertical blanking period and it stays active until the last line in the vertical blanking period.

However, the pixel where the FIELD ID signal transitions can be quite variable and depends on the external chip driving the VIP_PARSER. Many parts that generate digitized raw video have a programmable feature to specify when FIELD ID changes. FIELD ID is valid at the same point for every field. That is, if FIELD ID is read at one particular place in a field, the polarity of the signal will be reversed at the same location in the next field. So, FIELD ID can be corrected with a programmable polarity configuration bit FID_POLARITY (within VIP_PORT_A and VIP_PORT_B registers) that is XOR'ed with the captured value.

For discrete sync mode, FIELD ID will be registered on the first active pixel capture cycle of each line in both styles of HSYNC and ACTVID usage as specified in Figure 7-31 and Figure 7-32.

Figure 7-31. FID Registering When Using HSYNC

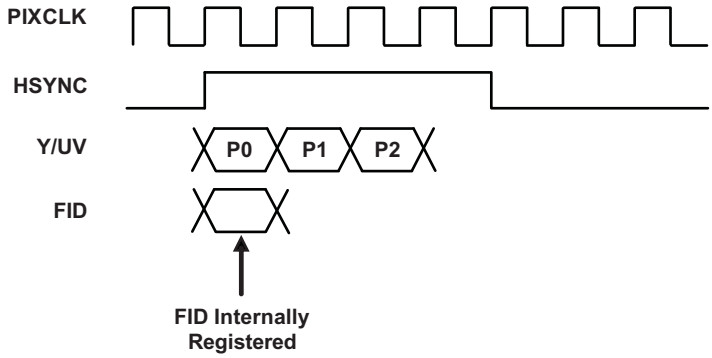
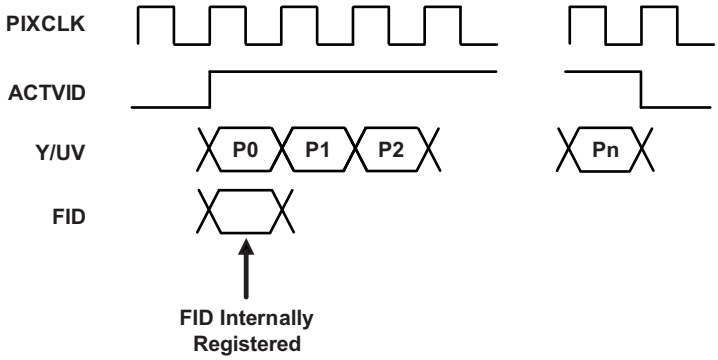


Figure 7-32. FID Registering When Using ACTVID

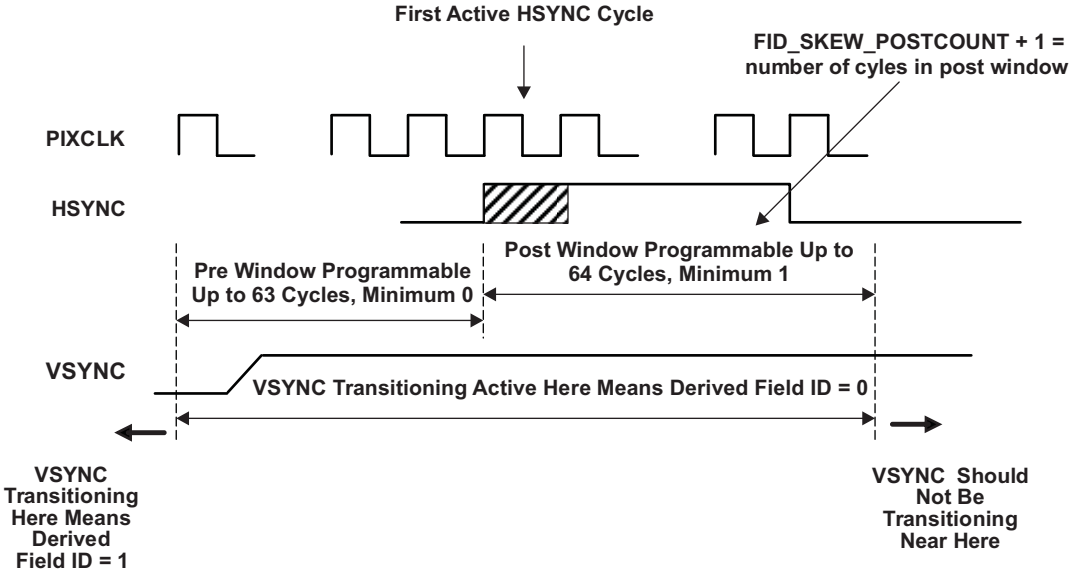


7.4.5.6.9 Field ID Determination Using VSYNC Skew

In order to save a device pin, there is a case where a skew may be inserted into VSYNC (with respect to HSYNC) when HSYNC is used as a start of line indicator as described in Figure 7-28. In this case, no FIELD ID signal is sent by the source chip. A description of Field ID determination by VSYNC skew is shown in Figure 7-33.

The active polarity of VSYNC falling within n pixel clock cycles of the first active cycle of HSYNC indicates the field id. If VSYNC is active before this time window, then the FIELD_ID = ‘1’ for the next picture. If VSYNC becomes active within this window, then FIELD_ID = ‘0’ for the next picture.

Figure 7-33. Field ID Determination By VSYNC Skew



When using FID determination by VSYNC skew, the value for VSYNC is also determined by transitions in the window as shown in Figure 7-30.

The VIP_PARSER supports a configuration FID_POLARITY bit within VIP_PORT_A and VIP_PORT_B registers. For FID determination by VSYNC Skew, the fid determination functions are described in Table 7-7.

Table 7-7. Polarity Table for FID Determination By VSYNC Skew

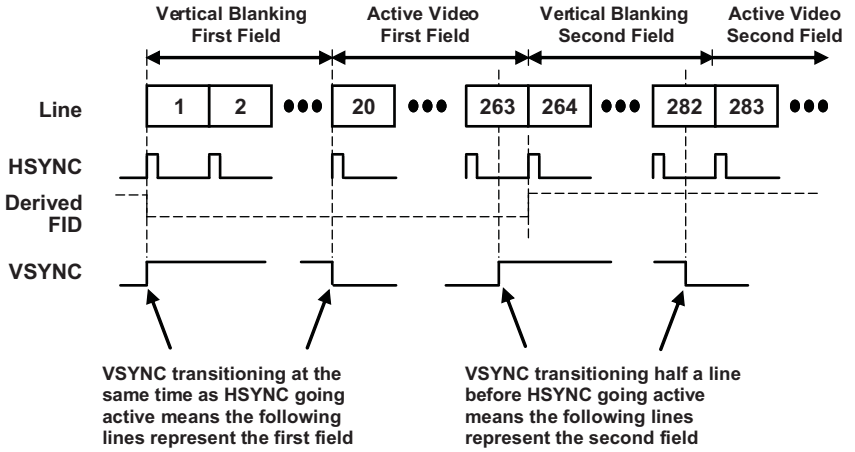
FID_POLARITY	Transition in Pre/Post Range	FID Determination
0	No	1
0	Yes	0
1	No	0
1	Yes	1

7.4.5.6.10 Rationale for FID Determination By VSYNC Skew

FID determination by VSYNC skew is a method for field ID determination derived from the analog NTSC and PAL interlaced specifications. Under this method, the sending device will not be providing a FID signal. NTSC has 525 total lines split between two fields. PAL has 625 total lines split between two fields. Each of these interlaced standards support an odd number of lines.

Let's consider just the 525 active line NTSC signal. For the sake of consistency, let's call Line 1 the first line of the 2-field pair and Line 525 the last line of the 2-field pair.

Figure 7-34. Example of 525-line FID Determination By VSYNC Skew



A waveform is shown in Figure 7-34. VSYNC is defined to go active at the same time as HSYNC for the first line of the first field in a two-field picture pair. For this first field, VSYNC will go inactive after Line 20.

For the second field, VSYNC will go active in the middle of Line 263 to signal that Line 264 is the start of a vertical blanking interval. When HSYNC for Line 264 arrives, coinciding with the vertical blanking interval for the beginning of the second field, VSYNC has already been active for half of Line 263. For the second field, VSYNC will go inactive midway through Line 282 to indicate that Line 283 is active video. When HSYNC for Line 283 appears, VSYNC has already been inactive for half a line.

By seeing whether VSYNC transitions at the beginning of a line or whether it transitions at the midway point of a line, one can determine whether the upcoming group of lines represents the first field or the second field. The derived FID is shown in dashed lines.

The analog NTSC specification defines the field ID changing part way into the vertical blanking. That is, the first few lines of vertical blanking belong to the previous field and the next several lines of vertical blanking belong to the upcoming field. The VIP_PARSER saves one channel of the entire vertical blanking interval between two active video fields into a single buffer. The hardware does not discriminate between whether the vertical blanking lines belong to the bottom of the previous field or those belonging to the start of the next field. This usage model is consistent with vertical ancillary data capture for embedded sync mode of operation.

Obviously, FID Determination by VSYNC skew cannot be used when framing does not use the VSYNC signal but rather relies on the ACTVID signal instead.

7.4.5.6.11 ACTVID Framing

Instead of an HSYNC signal, the VIP_PARSER can use ACTVID framing as described in Figure 7-27. Under ACTVID Framing, VSYNC is used to separate vertical blanking lines from active video lines.

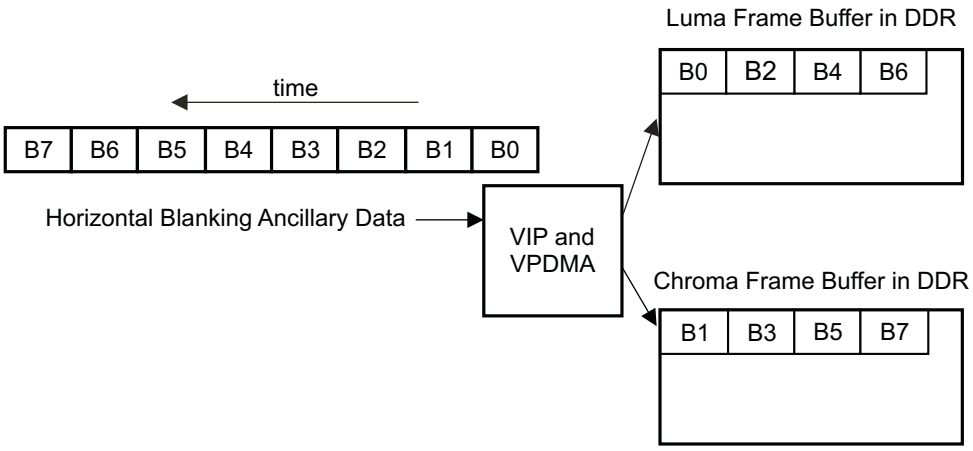
FID determination by VSYNC Skew is not allowed for ACTVID framing because there is no HSYNC input signal in this mode. Also, the VSYNC transition window is not employed. VSYNC is captured at the first pixel of each ACTVID grouping of pixels. Lines are separated by ACTVID transitioning inactive.

7.4.5.6.12 Ancillary Data Storage in Descrete Sync Mode

Ancillary data appearing in horizontal blanking is called Horizontal Blanking Ancillary Data. Ancillary Data in vertical blanking is called Vertical Blanking Ancillary Data.

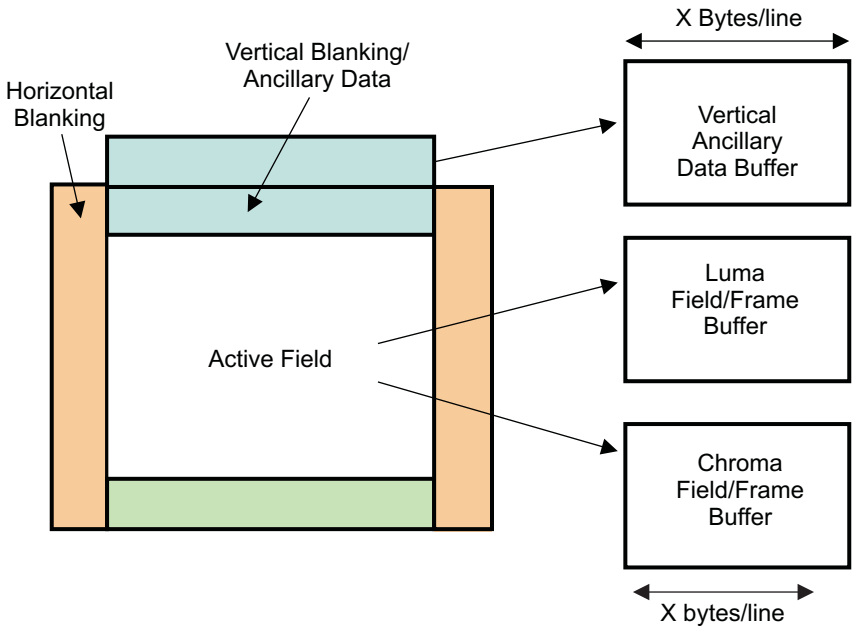
Horizontal Blanking Ancillary Data is not commonly used. For the ACTVID data valid mode described in Figure 7-26, there is no way to capture Horizontal Blanking Ancillary Data. Using the HSYNC mode in Figure 7-28, all blanking pixels are captured. However, the horizontal ancillary data is byte-by-byte distributed between the Luma and Chroma frame buffers. Chroma sited bytes are saved in the Chroma frame buffer and Luma sited bytes are saved in the Luma frtame buffer. Some CPU effort would be needed in order to extract ancillary data from the desired Luma or Chroma channel frame buffers. Also, the video on each line starts after the horizontal blanking period. This situation is shown in Figure 7-35.

Figure 7-35. Horizontal Ancillary Data Packing When HSYNC Used as Sync Signal



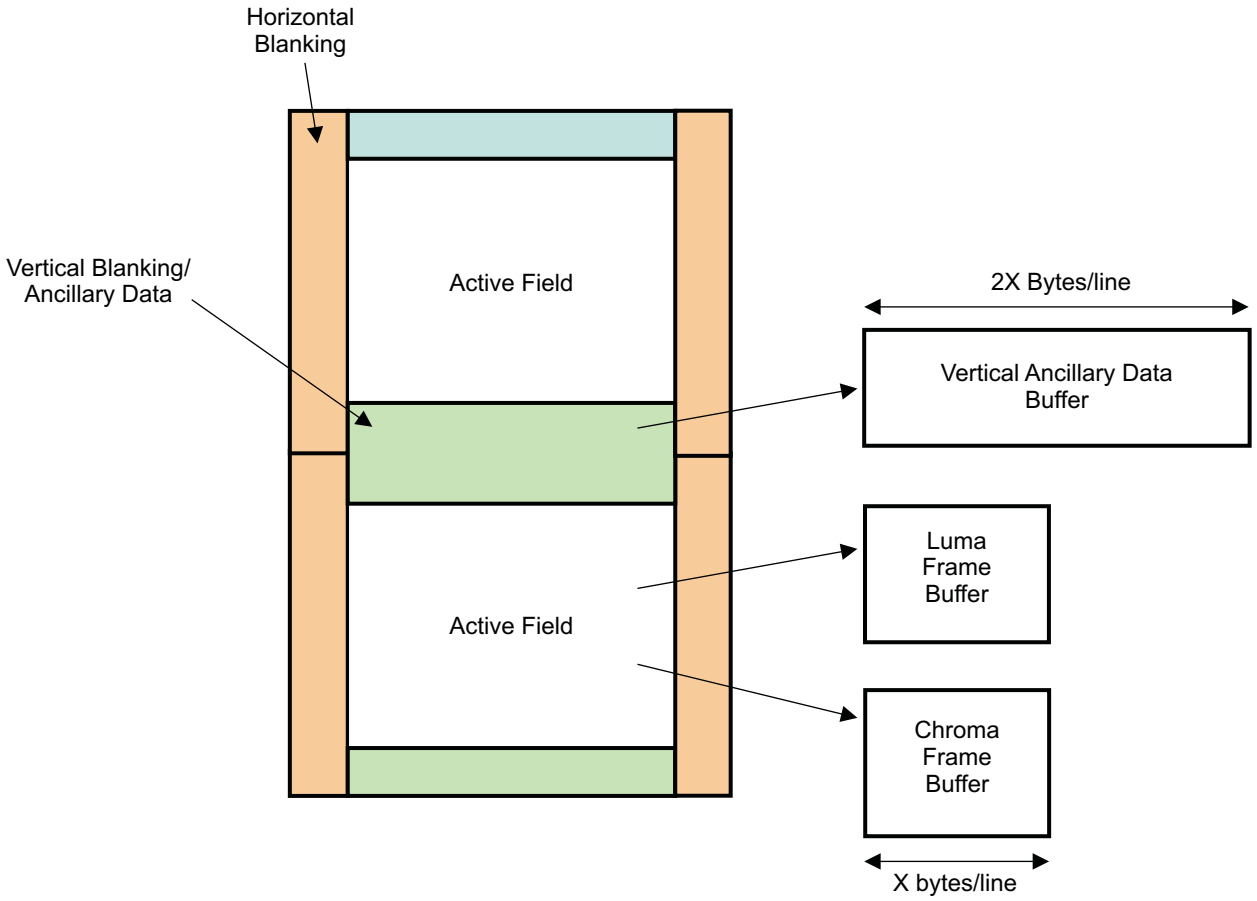
With Interlaced source material, Vertical Blanking Ancillary data will be stored in a separate Vertical Ancillary Data Buffer as shown in Figure 7-36. The Channel from which vertical ancillary data is extracted is a configuration option. For an input image of x active pixels per line, each line of Vertical Blanking Ancillary Data will have x bytes. Unlike the horizontal case, the CPU parsing this Ancillary Data will see a contiguous section of Vertical Ancillary Data that is not intermixed with Video data.

Figure 7-36. Interlaced Field Vertical Blanking Ancillary Data Storage



For Progressive source video, the FIELD ID does not change. So, the Vertical Ancillary Data Buffer will contain all the information beginning from the vertical blanking of the previous frame. This situation is shown in Figure 7-37.

Figure 7-37. Progressive Frame Vertical Blanking Ancillary Data Storage

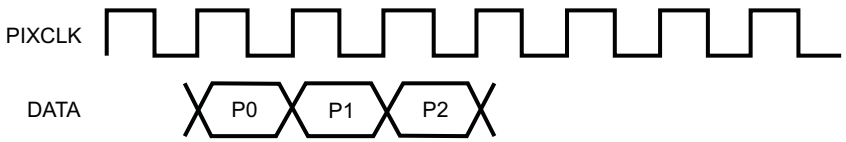


7.4.5.7 BT.656 Style Embedded Sync

7.4.5.7.1 Data Input

Like Discrete Sync Input, Embedded Sync mode takes data from the input bus. Input data can be 8, 16 bits wide. A sample is retrieved each and every Pixel Clock cycle. There is no valid signal gating data entry. Figure 7-38 shows a valid data sample each Pixel Clock period.

Figure 7-38. Embedded Sync Data Entry



7.4.5.7.2 Sync Words

In embedded sync mode, code words are inserted into the stream at pixel clock rates. For external devices that send out 10 bits (single pixel interface) or 20 bits (parallel Y-Cb/Cr interface) of data, only the 8 (single pixel interface) or 16 (parallel 8bY-8bCb/Cr interface) most significant bits of each pixel are used.

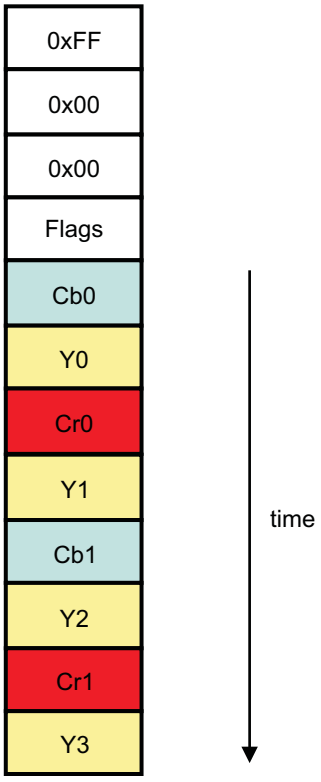
The key code words are Start of Active Video (SAV) and End of Active Video (EAV). Three flags are found in these code words: F (field), V (vertical sync), and H (horizontal sync). These flags signify the position in the frame corresponding to the data immediately following the codeword. The flags determine whether the code is EAV or SAV and where they lie in the picture. The first byte of the code word is 0xFF. The second and third bytes are 0x00. The bit ordering of the fourth byte is detailed in Table 7-8.

Table 7-8. Fourth Byte of EAV/SAV Code Word

7 (fixed)	6 (F)	5 (V)	4 (H)	3 (P3=V^H)	2 (P2=F^H)	1 (P1=F^V)	0 (P0=F^V^H)	Description
1	0	0	0	0	0	0	0	SAV, Field 0, Active Video
1	0	0	1	1	1	0	1	EAV. Field 0, Horizontal Blanking
1	0	1	0	1	0	1	1	SAV, Field 0, Vertical Blanking
1	0	1	1	0	1	1	0	EAV, Field 0, Horizontal Blanking in Vertical Blanking Region
1	1	0	0	0	1	1	1	SAV, Field 1, Active Video
1	1	0	1	1	0	1	0	EAV, Field 1, Horizontal Blanking
1	1	1	0	1	1	0	0	SAV, Field 1, Vertical Blanking
1	1	1	1	0	0	0	1	EAV, Field 1, Horizontal Blanking in Vertical Blanking Region

An example of the input ordering of the embedded code word, followed by active video, is shown in Figure 7-39. The input data mode is 8 bits for the example.

Figure 7-39. Code Word Format Example Followed by Video Data



7.4.5.7.3 Error Correction

The FVH flags are sent with four protection bits to support double error detection, single error correction. A non-correctable detected error is simply ignored. An option exists for the protection bits to correct a single bit error in the FVH flags. The correction table is shown in Table 7-9. n/c means that the error condition is detected, but it is non-correctable.

Table 7-9. Error Correction Matrix

	F, V, and H Flags							
P3, P2, P1, P0	**000**	**001**	**010**	**011**	**100**	**101**	**110**	**111**
0000	000	000	000	n/c	000	n/c	n/c	111
0001	000	n/c	n/c	111	n/c	111	111	111
0010	000	n/c	n/c	011	n/c	101	n/c	n/c
0011	n/c	n/c	010	n/c	100	n/c	n/c	111
0100	000	n/c	n/c	011	n/c	n/c	110	n/c
0101	n/c	001	n/c	n/c	100	n/c	n/c	111
0110	n/c	011	011	011	100	n/c	n/c	011
0111	100	n/c	n/c	011	100	100	100	n/c
1000	000	n/c	n/c	n/c	n/c	101	110	n/c
1001	n/c	001	010	n/c	n/c	n/c	n/c	111
1010	n/c	101	010	n/c	101	101	n/c	101
1011	010	n/c	010	010	n/c	101	010	n/c
1100	n/c	001	110	n/c	110	n/c	110	110
1101	001	001	n/c	001	n/c	001	110	n/c
1110	n/c	n/c	n/c	011	n/c	101	110	n/c
1111	n/c	001	010	n/c	100	n/c	n/c	n/c

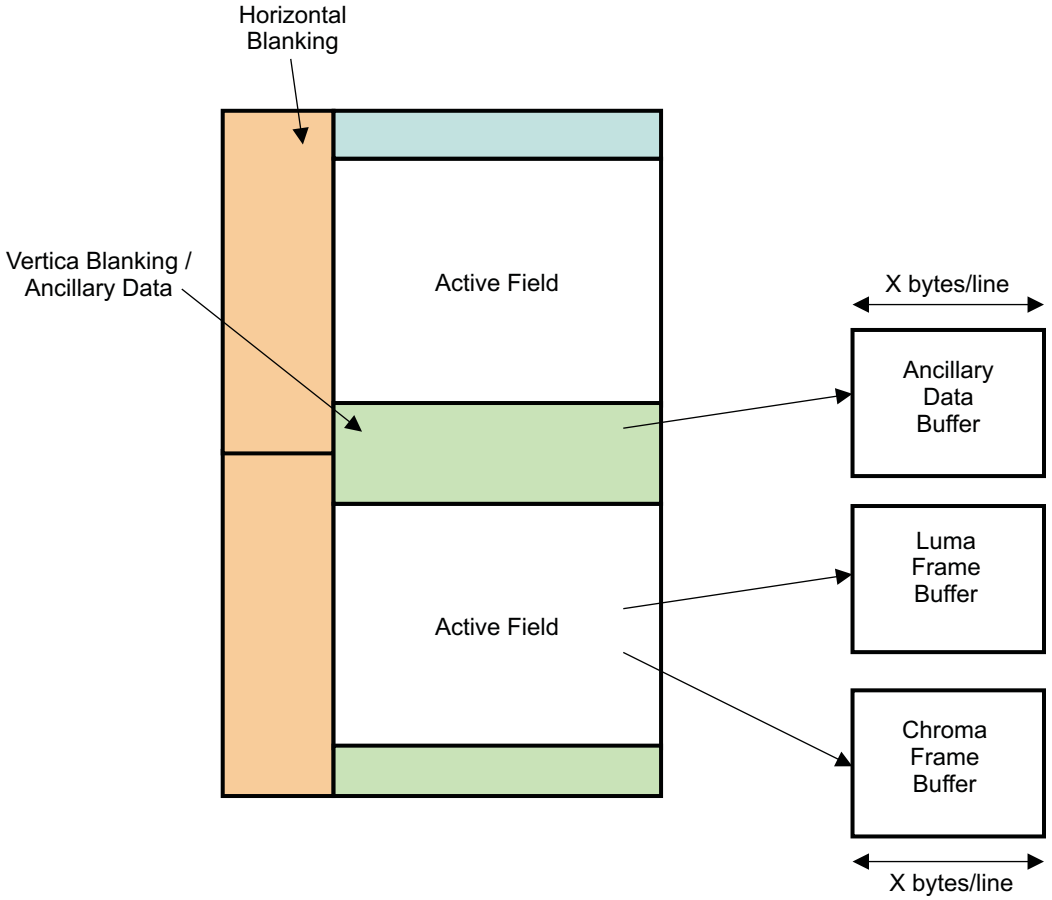
7.4.5.7.4 Embedded Sync Ancillary Data

With Embedded Sync streams, only Vertical Ancillary Data can be extracted. The Vertical Ancillary Data buffer is the same width as the corresponding Luma and Chroma buffers. The channel from which Vertical Ancillary Data is extracted is a configuration option.

Horizontal Ancillary data cannot be extracted using embedded sync mode.

The Vertical Ancillary Data is captured starting from the end of the previous active video. See Figure 7-40 for a more detailed description of embedded sync packing.

Figure 7-40. Embedded Sync Packing



7.4.5.8 Source Multiplexing

7.4.5.8.1 Multiplexing Scenarios

Some applications require multiple camera sources to be used at the same time. For this type of device, one solution would be to support N-number of 8-bit or 16-bit data interfaces for each of N cameras. However, this solution does not efficiently minimize pin count. One set of 8-bit or 16-bit interfaces has the bandwidth to support more than one video source, depending on the resolution of the video. Table 7-10 is explanatory only and shows the number of sources that can be multiplexed in one VIP for 8-bit and 16-bit interface modes. Note that it does not reflect the capabilities of the VIP_PARSER. In addition, the interface pixel clock rates are shown. The VPDMA limits 16 camera sources to be saved to DDR memory per Pixel Clock Input Domain.

Table 7-10. Multiplexing Configurations and Pixel Clock Rates

	Maximum Channels in Single 16-bit Data Interface Mode	Maximum Channels in Dual 8-bit Data Interface Mode - Interleaved Channels per Single 8-bit Port. One 16-bit VIP can be configured to support two such 8-bit ports.	Interface Clock Rate (MHz)
HD Interlaced	2	1	148.5
D1 Interlaced	8	4	108.1
CIF Interlaced	n/a	n/a	n/a
HD Progressive	1	n/a	148.5

Table 7-10. Multiplexing Configurations and Pixel Clock Rates (continued)

D1 Progressive	4	2	108.1
CIF Progressive (1)	32	16	162.2

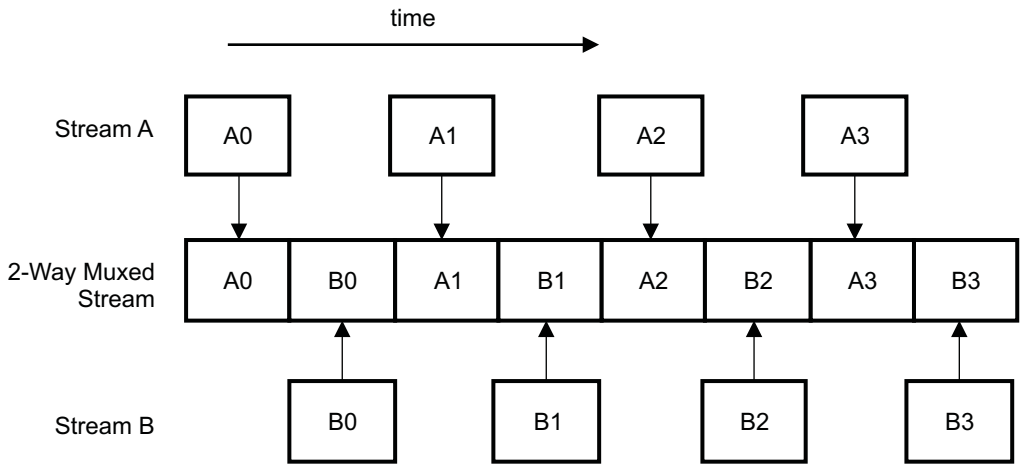
(1) Blanking pixels are not used in the CIF clock rate calculations. Addition of blanking pixels would require a slightly higher clock rate.

NOTE: These Channel Density values reflect one VIP subsystem.

7.4.5.8.2 2-Way Multiplexing

For 2-Way Multiplexing, two embedded sync streams are interleaved a pixel at a time as shown in Figure 7-41.

Figure 7-41. 2-Way Multiplexing



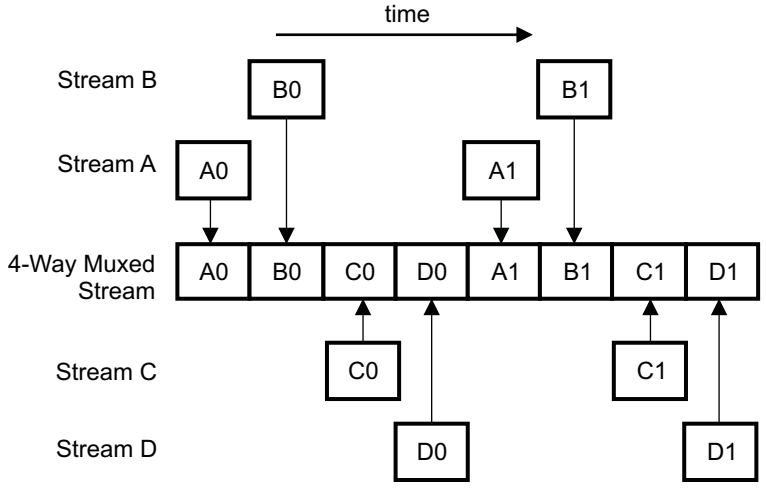
The sync codeword, FF-00-00-XY, is replicated in both source streams. In 2-Way Multiplexing, the sizes of both camera sources must be the same. Likewise, the Vertical Ancillary Data size for both sources must be identical. However, the two streams are not necessarily sending the same pixel site in adjacent clock cycles.

7.4.5.8.3 4-Way Multiplexing

For 4-Way Multiplexing, four embedded sync streams are multiplexed into one as seen in Figure 7-42.

Again, the sync codeword is in all four sources. Like 2-Way Multiplexing, the sizes of the four camera sources are the same and the sizes of the Vertical Ancillary Data regions are the same. The four streams are not necessarily sending the same pixel site in adjacent clock cycles.

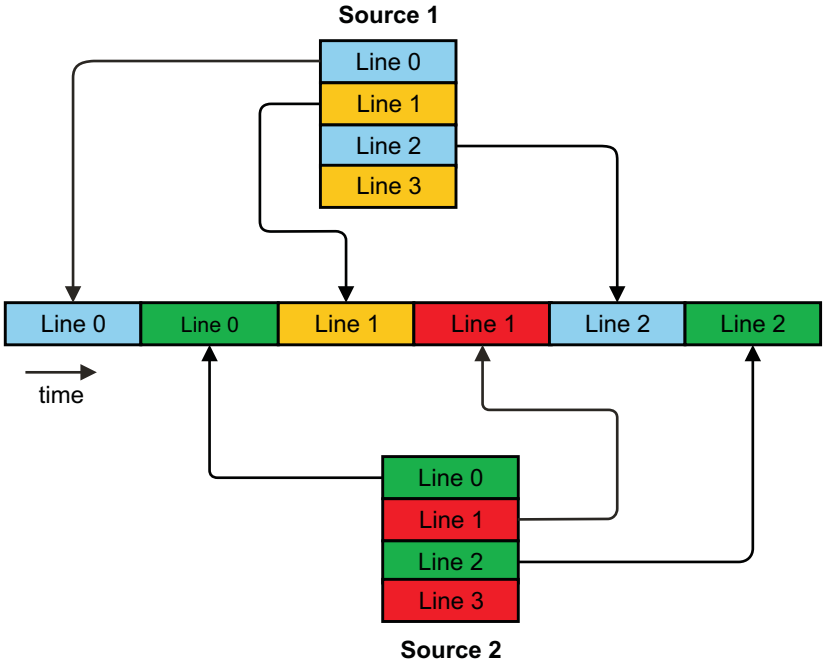
Figure 7-42. Example of 4-Way Multiplexing



7.4.5.8.4 Line Multiplexing

In Line Multiplexing, n-different sources are sent into the VIP a complete line at a time using a modified version of embedded sync. An example of Line Multiplexing for two sources is shown in Figure 7-43.

Figure 7-43. Example of Line Multiplexing



The width and height of each source in Line Multiplexed data can be different. For instance, one source can be PAL while another one can be NTSC. A line is comprised of YUV422 pixels in repeating patterns of CbYCrY.

7.4.5.8.5 Super Frame Concept in Line Multiplexing

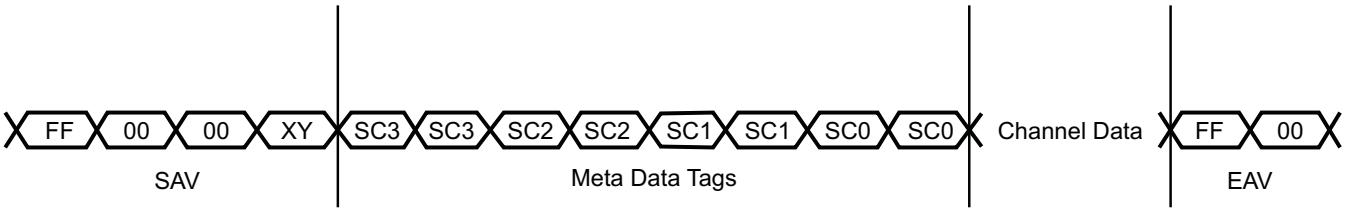
Different camera sources are interleaved on a line-by-line basis. The beginning of each line carries a Metadata tag that provides key information about that line. This Metadata tag is preceded by a SAV codeword in which F=0, V=0, and H=0.

At the end of the line, an EAV code is inserted with an appropriate number of padding pixels following it. This EAV code has F=0, V=0, and H=1. The startcodes used for the Metadata wrapped lines and the dummy lines form the Super Frame. The VIP_PARSER module has logic to parse out the super frame, analyze the Metadata tags, and frame buffer the line contents appropriately.

7.4.5.8.6 8-bit Data Interface in Line Multiplexing

Figure 7-44 is an example of an 8-bit line multiplexing interface. Channel Data is the CbYCrY sequence representing a line. Preceding Channel Data is the four byte Meta Data tags. Note that the Meta Data bytes are replicated in both the Luma and the Chroma sites. The entire structure is bounded by a traditional SAV/EAV code in which the V flag is 0.

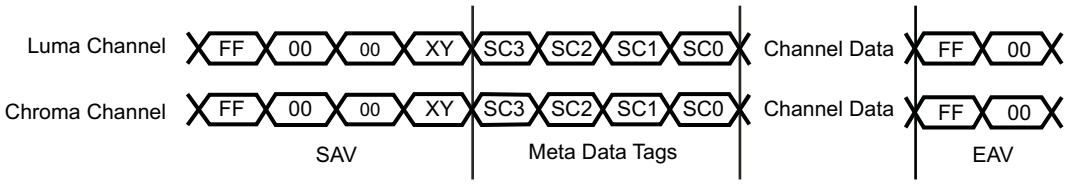
Figure 7-44. 8-bit Line Mux Interface



7.4.5.8.7 16-bit Data Interface in Line Multiplexing

Figure 7-45 describes the 16-bit line multiplexing interface. Channel Data is the active line. All the Y pixels are in the Luma Channel. The CbCr pixels are in the Chroma Channel. Each cycle, a 16-bit value representing one Luma sample and one Chroma sample enters the VIP_PARSER. The Meta Data tags are replicated in both channels. Likewise, the SAV/EAV startcodes are found in both channels. The V flags for the SAV/EAV startcodes are always 0.

Figure 7-45. 16-bit Line Mux Interface



7.4.5.8.8 Split Lines in Line Multiplex Mode

Suppose an external device is sending two dissimilar sources in Line Multiplex mode. One narrower source has X pixels per line. The wider source has 2X pixels per line.

The Meta Data has provisions for the external device to split a line. The Beginning of Line (BOL) and End of Line (EOL) flags tag a split line as described in Table 7-11.

Table 7-11. Split Line Table

BOL	EOL	Function
0	0	Undefined
0	1	Line Segment is the second half of a line
1	0	Line Segment is the first half of a line
1	1	Line has not been segmented into two.

7.4.5.8.9 Meta Data

Table 7-12 shows the bitfields in the Meta Data start codes.

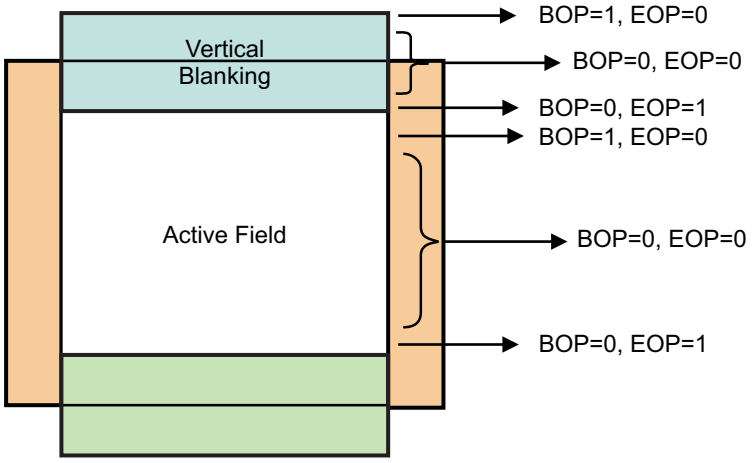
Table 7-12. Meta Data Layout

Byte	7	6	5	4	3	2	1	0
SC3	1	BOP	EOP	RSVD	CH_ID[3:0]			
SC3	1	BOP	EOP	RSVD	CH_ID[3:0]			
SC2	0	BOL	EOL	VDET	LINE_ID[10:7]			
SC1	~LINE_ID[6]	LINE_ID[6:0]						
SC0	PAD	F	V	H	P3	P2	P1	P0

BOP tags the line as a startline in a period. A period is defined as the contiguous lines in a vertical blanking period or the contiguous lines in a field or frame. For the vertical blanking period case, the vertical blanking at the bottom of the previous field or frame combined with the vertical blanking at the top of the current field or frame is combined to create one period.

EOP tags the line as an endline in a period. Figure 7-46 shows the definition of the two types of Periods as outlined by BOP and EOP.

Figure 7-46. BOP/EOP Definition of a Period



For the Split Line at the top of a Period, the BOP bit is set for both halves of the split line. Likewise, for the Split Line at the bottom of a Period, the EOP bit is set for both halves of the split line.

CH_ID is the channel ID, which tags the camera source that generated the incoming line. A maximum of 16 camera sources are support per Pixel Input Clock Domain.

LINE_ID is the line number, starting from 0 and incrementing by one for each subsequent line from the same source.

PAD is a flag which tags the line as an artificially inserted padding line. When PAD is ‘0’, the line should be discarded.

F, V, H, P3, P2, P1, and P0 are the bits representing the normal XY code. F is the Field ID associated with line, V signals when the line is in the vertical blanking, H specifies that the line is in the Horizontal Blanking, and P3:P0 are the protection bits.

Since only active video and vertical ancillary data lines are encapsulated in the Meta Data, the H bit in the SC0 byte should never be ‘1’.

7.4.5.8.10 TI Line Mux Mode, Split Lines, and Channel ID Remapping

The VIP_PARSER supports a maximum of 8 different Channel IDs per Port. The Channel IDs must be in the range {0:7} (3 bits). In the source multiplex, only one source can be a split line source and have the same Channel ID as one of the non-split line sources.

This scenario involves an external NTSC decoder which supports 8 D1 cameras. The external NTSC decoder will downscale the 8 sources to SIF format. However, one camera source will be sent in the multiplex as both the downscaled version and the original D1 sized version. They will both have the same Channel ID. However, the D1 version will be sent as a split-line. The source multiplex will thus have 9 maximum streams.

The VIP_PARSER (for TI Line Mux Mode only), will left shift the Channel ID by one. Bit 0 is used as an indicator whether the Source is a split-line source or a normal non-split line source. Only one of the nine inputs can be a split line source.

Table 7-13. TI Line Mux Mode Channel ID Remapping

	Channel ID Sent to VPDMA	
Source Input Channel ID	**Non-split Line**	**Split Line**
0x0	0x0	0x1
0x1	0x2	0x3
0x2	0x4	0x5
0x3	0x6	0x7
0x4	0x8	0x9
0x5	0xA	0xB
0x6	0xC	0xD
0x7	0xE	0xF

All subsequent references to the Camera Source, such as in a VPDMA return descriptor, will reference the remapped Channel ID.

The VIP_OUTPUT_PORT_A_SRC0_SIZE through VIP_OUTPUT_PORT_A_SRC15_SIZE registers for Port A, and VIP_OUTPUT_PORT_B_SRC0_SIZE through VIP_OUTPUT_PORT_B_SRC15_SIZE registers for Port B and the VIP_OUTPUT_PORT_A_SRC_FID and VIP_OUTPUT_PORT_B_SRC_FID status registers reflect the remapped Channel ID when the port is in TI Line Mux mode.

7.4.5.9 Channel ID Extraction for 2x/4x Multiplexed Source

7.4.5.9.1 Channel ID Extraction Overview

For 2-way and 4-way multiplexed source, the Channel ID is either embedded in the four protection bits inside the EAV/SAV code words or in the horizontal blanking pixel data. A configuration setting determines where the VIP_PARSER would search for the Channel ID.

7.4.5.9.2 Channel ID Embedded in Protection Bits for 2- and 4-Way Multiplexing

The four-bit channel ID is an identifier corresponding with the source number (camera) of the incoming video. As shown in Table 7-14, the Channel ID is placed in the code fourth byte of the EAV/SAV code words normally used for protection bits. With 4 bits, the maximum number of sources that can be defined in this range is 16. However, only Channel IDs in the range {0:7} are supported. Obviously, error correction cannot be performed on the FVH flags since the protection bits are no longer there.

Table 7-14. Channel ID Embedded in EAV/SAV

7 (fixed)	6 (F)	5 (V)	4 (H)	3 (ch_id[3])	2 (ch_id[2])	1 (ch_id[1])	0 (ch_id[0])	Description
1	0	0	0	Ch_id = {0:15}				SAV, Field 0, Active Video
1	0	0	1	Ch_id = {0:15}				EAV, Field 0, Horizontal Blanking
1	0	1	0	Ch_id = {0:15}				SAV, Field 0, Vertical Blanking

Table 7-14. Channel ID Embedded in EAV/SAV (continued)

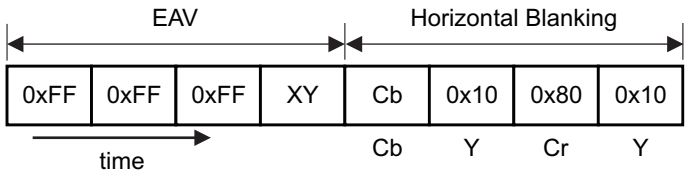
7 (fixed)	6 (F)	5 (V)	4 (H)	3 (ch_id[3])	2 (ch_id[2])	1 (ch_id[1])	0 (ch_id[0])	Description
1	0	1	1	Ch_id = {0:15}				EAV, Field 0, Horizontal Blanking in Vertical Blanking Region
1	1	0	0	Ch_id = {0:15}				SAV, Field 1, Active Video
1	1	0	1	Ch_id = {0:15}				EAV, Field 1, Horizontal Blanking
1	1	1	0	Ch_id = {0:15}				SAV, Field 1, Vertical Blanking
1	1	1	1	Ch_id = {0:15}				EAV, Field 1, Horizontal Blanking in Vertical Blanking Region

7.4.5.9.3 Channel ID Embedded in Horizontal Blanking Pixel Data for 2- and 4-Way Multiplexing

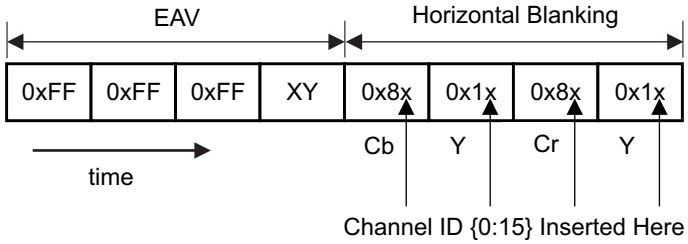
In Horizontal Blanking and Vertical Blanking, non-ancillary data pixels should be Y=0x10 and Cb=Cr=0x80. When the Channel ID is embedded in the Horizontal Blanking for 2 and 4-way multiplexing, the lower nibbles of all Luma and Chroma pixels are replaced by the 4 bit Channel ID. This scenario is shown in Figure 7-47. The maximum number of values defined by this 4-bit range is 2^4 = 16. However, only Channel IDs in the range {0:7} are supported.

Figure 7-47. Channel ID Inserted Into Horizontal Blanking

Regular Horizontal Blanking Following EAV



Channel ID Inserted Into Horizontal Blanking Following EAV



7.4.5.10 Embedded Sync Mux Modes and Data Bus Widths

Legal combinations of Embedded Sync Mux Modes and Data Bus Widths are described in Table 7-15.

Table 7-15. Valid Embedded Sync Mux Mode and Data Bus Width Combinations

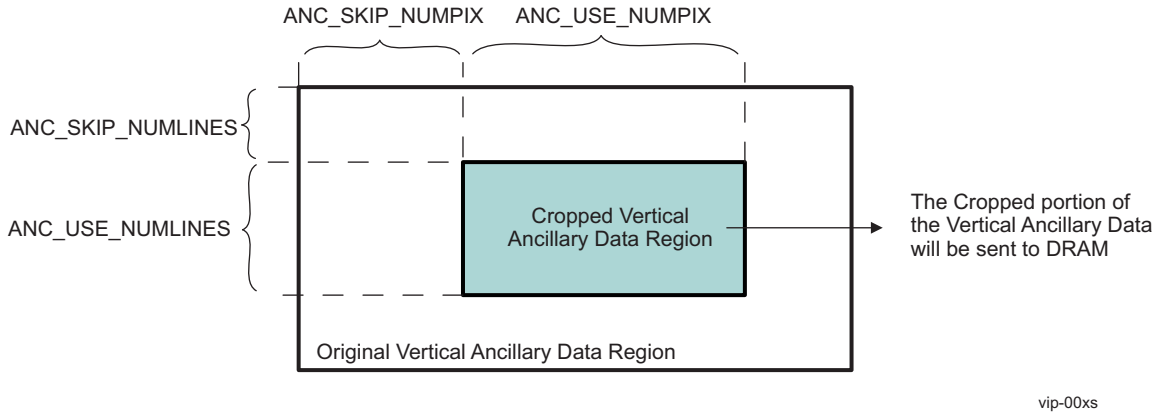
	1x Mux	2x Mux	4x Mux	Line Mux
8 Bit	v	v	v	v
16 Bit	v	n/a	n/a	v

7.4.5.11 Ancillary and Active Video Cropping

One Source Number for each Port can be cropped. Cropping is available for both Ancillary Data and Active Video.

For the Vertical Ancillary Data from Port A, cropping is enabled by setting the VIP_ANC_CROP_HORZ_PORT_A[15] ANC_BYPASS_N bit. The Source Number from Port A that gets cropped is defined by the VIP_ANC_CROP_HORZ_PORT_A[31:28] ANC_TARGET_SRCNUM register. VIP_ANC_CROP_HORZ_PORT_A[11:0] ANC_SKIP_NUMPIX, VIP_ANC_CROP_HORZ_PORT_A[27:16] ANC_USE_NUMPIX, VIP_ANC_CROP_VERT_PORT_A[11:0] ANC_SKIP_NUMLINES, and VIP_ANC_CROP_VERT_PORT_A[27:16] ANC_USE_NUMLINES define the region of the selected Source Number that is cropped and sent to DRAM. The Vertical Ancillary Data Cropping region is described in Figure 7-48.

Figure 7-48. Vertical Ancillary Data Cropping

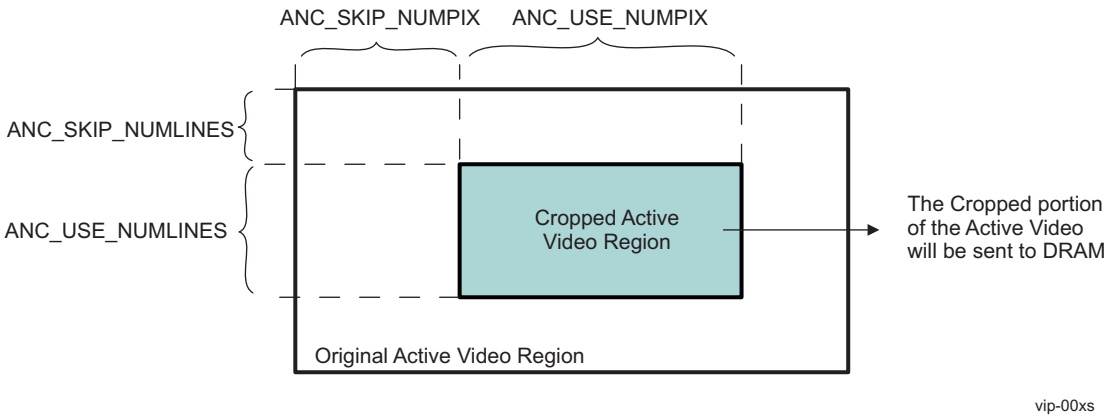


Note that for 8-bit input mode only, a setting exists that allow Vertical Ancillary data from both the Luma and Chroma channels to be captured. Both channels of Vertical Ancillary Data are captured when VIP_XTRA_PORT_A[14:13] ANC_CHAN_SEL_8B is set to “1x”. Thus, the number of data elements per line in this case is twice the equivalent number of Luma pixels per line. In other words, for this particular dual channel capture example, if there are 720 Luma pixels per line, then the total number of Vertical Ancillary Data Pixels in the source picture can be 2x720=1440 pixels.

For Active Video from Por tA, cropping is enabled by setting the VIP_CROP_HORZ_PORT_A[15] ACT_BYPASS_N bit. The Source Number from Port A that gets cropped is defined by the VIP_CROP_HORZ_PORT_A[31:28] ACT_TARGET_SRCNUM register.

VIP_CROP_HORZ_PORT_A[11:0] ACT_SKIP_NUMPIX, VIP_CROP_HORZ_PORT_A[27:16] ACT_USE_NUMPIX, VIP_CROP_VERT_PORT_A[11:0] ACT_SKIP_NUMLINES, and VIP_CROP_VERT_PORT_A[27:16] ACT_USE_NUMLINES define the region of the selected Source Number that is cropped and sent to DRAM. The Vertical Ancillary Data Cropping region is described in Figure 7-49

Figure 7-49. Active Video Cropping

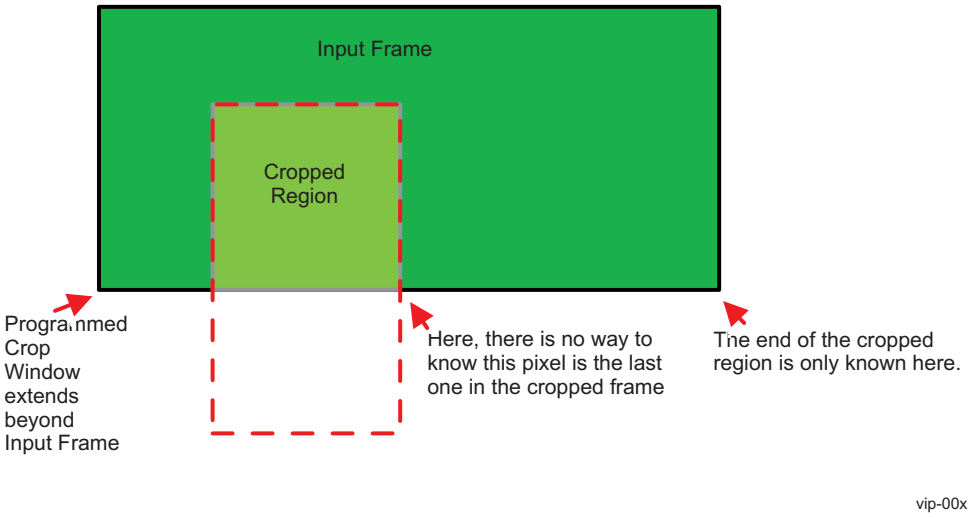


Cropping for Port B works in a similar way to Port A. Since picture data is in 4:2:2 format, ANC_SKIP_NUMPIX(ACT_SKIP_NUMPIX) and ANC_USE_NUMPIX (ACT_USE_NUMPIX) must be evenly divisible by 2. If the output of VIP_PARSER is sent to a 4:2:2 to 4:2:0 converter, then ANC_USE_NUMLINES(ACT_USE_NUMLINES) must also be evenly divisible by 2.

Error cases in cropping occur when the crop window programmed is larger than the incoming video frame. Crop window errors normally result in the return of the crop region where the crop window overlays the incoming video frame. However, there is one problematic error cropping case, as illustrated in Figure 7-50.

The programmed crop window extends below the input picture and the last pixel of the input picture is not a part of the selected crop region. In this case, at the last pixel of the last line in the green crop output, there is no way to determine that this pixel is the last pixel of the cropped output.

Figure 7-50. Problematic Error Cropping Case



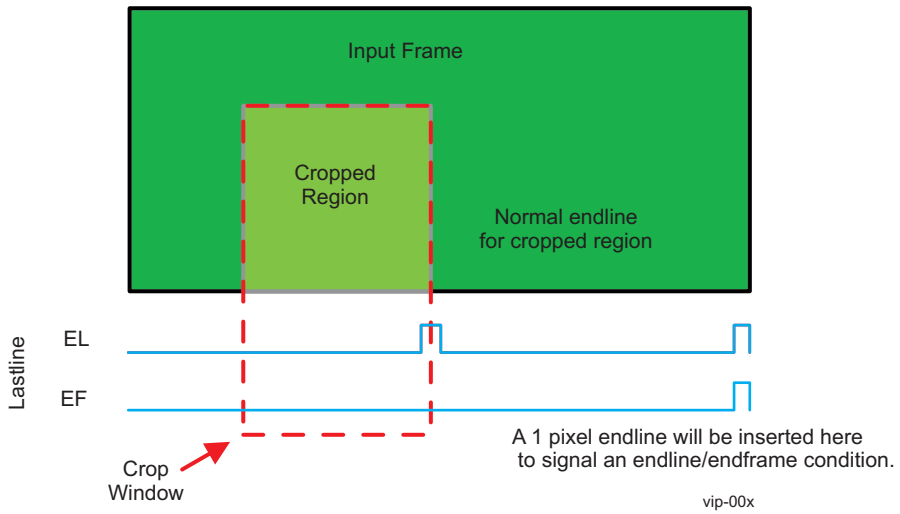
In this case, the crop tool sends outs a single pixel with an endline and endframe when it reaches the last pixel of the input picture. If the green cropped region has n lines, the actual output from the crop tool will have n+1 lines and the width of the last line will be 1 pixel.

Figure 7-51 shows the endline (EL) and endframe (EF) signals corresponding to the last line of the cropped region. At the last line and last pixel of the cropped region, the crop tool will only output an endline. It cannot output an endframe at this point because the crop tool might get another line from the streaming input.

Later, on the same input line, the last pixel of the input frame appears. Here, the crop tool knows that the cropped region has ended. In this case, a single endline/endframe pixel is sent out to signal that the frame has ended.

NOTE: There is no interrupt to notify that application level that a crop error has occurred.

Figure 7-51. Endline/Endframe Behavior for Error Cropping Case



7.4.5.12 Interrupts

The VIP_PARSER module has 19 interrupts out of which one can be mapped to VIP top level.

When an interrupt occurs and is determined to be from the VIP_PARSER module, the VIP_PARSER level of masks, clears, and status registers must be checked and updated first.

Table 7-16 describes each of the interrupts events supported by the VIP_PARSER, together with associated Interrupt Mask (VIP_FIQ_MASK), Interrupt Clear (VIP_FIQ_CLEAR), and Interrupt Status (VIP_FIQ_STATUS) registers.

Table 7-16. VIP_PARSER Interrupt Events

Event Flag	Event Mask	Map to	Description
VIP_FIQ_STATUS[21] PORT_A_YUV_PROTOCOL_VIOLATION	VIP_FIQ_MASK[21] PORT_A_YUV_PROTOCOL_VIOLATION_MASK	PrtBDisableComplete	When a port is running and VIP_PORT_B[8] ENABLE bit is turned off, logic exists to ensure that a complete frame is sent out of the Ancillary and Active Video VPI ports. That is, a frame is not stopped in the middle. This interrupt is activated when all active frames have been sent out Port B following a disable.
VIP_FIQ_STATUS[20] PORT_A_ANC_PROTOCOL_VIOLATION	VIP_FIQ_MASK[20] PORT_A_ANC_PROTOCOL_VIOLATION_MASK	PrtADisableComplete	When a port is running and VIP_PORT_A[8] ENABLE bit is turned off, logic exists to ensure that a complete frame is sent out of the Ancillary and Active Video VPI ports. That is, a frame is not stopped in the middle. This interrupt is activated when all active frames have been sent out Port A following a disable.
VIP_FIQ_STATUS[19] PORT_B_YUV_PROTOCOL_VIOLATION	VIP_FIQ_MASK[19] PORT_B_YUV_PROTOCOL_VIOLATION_MASK	PrtBANCProtocolVio	This interrupt is enabled when the protocol checker on the output of the VIP_PARSER encounters a violation on the Ancillary VPI of Port B.
VIP_FIQ_STATUS[18] PORT_B_ANC_PROTOCOL_VIOLATION	VIP_FIQ_MASK[18] PORT_B_ANC_PROTOCOL_VIOLATION_MASK	PrtBYUVProtocolVio	This interrupt is enabled when the protocol checker on the output of the VIP_PARSER encounters a violation on the Active Video VPI of Port B.
VIP_FIQ_STATUS[17] PORT_A_CFG_DISABLE_COMPLETE	VIP_FIQ_MASK[17] PORT_A_CFG_DISABLE_COMPLETE_MASK	PrtAANCProtocolVio	This interrupt is enabled when the protocol checker on the output of the VIP_PARSER encounters a violation on the Ancillary VPI of Port A.
VIP_FIQ_STATUS[16] PORT_B_CFG_DISABLE_COMPLETE_CLR	VIP_FIQ_MASK[16] PORT_B_CFG_DISABLE_COMPLETE_MASK	PrtAYUVProtocolVio	This interrupt is enabled when the protocol checker on the output of the VIP_PARSER encounters a violation on the Active Video VPI of Port A.
VIP_FIQ_STATUS[15] PORT_B_SRC0_SIZE_STATUS	VIP_FIQ_MASK[15] PORT_B_SRC0_SIZE	PrtBSrc0Size	The output size for Srcnum=0 on Port B differs from the XTRA_PORT_B[11:0] SRC0_NUMLINES and [27:16] SRC0_NUMPIX register settings
VIP_FIQ_STATUS[14] PORT_A_SRC0_SIZE_STATUS	VIP_FIQ_MASK[14] PORT_A_SRC0_SIZE	PrtASrc0Size	The output size for Srcnum=0 on Port A differs from the XTRA_PORT_B[11:0] SRC0_NUMLINES and [27:16] SRC0_NUMPIX register settings
VIP_FIQ_STATUS[13] PORT_B_DISCONN_STATUS	VIP_FIQ_MASK[13] PORT_B_DISCONN	PrtBDisconn	Port B Link Disconnect for Srcnum 0
VIP_FIQ_STATUS[12] PORT_B_CONN_STATUS	VIP_FIQ_MASK[12] PORT_B_CONN	PrtBConn	Port B Link Connect for Srcnum 0
VIP_FIQ_STATUS[11] PORT_A_DISCONN_STATUS	VIP_FIQ_MASK[11] PORT_A_DISCONN	PrtADisConn	Port A Link Disconnect for Srcnum 0
VIP_FIQ_STATUS[10] PORT_A_CONN_STATUS	VIP_FIQ_MASK[10] PORT_A_CONN	PrtAConn	Port A Link Connect for Srcnum 0
VIP_FIQ_STATUS[9] OUTPUT_FIFO_PRTB_ANC_STATUS	VIP_FIQ_MASK[9] OUTPUT_FIFO_PRTB_ANC_OF	OpPrtBAnc	Overflow at Ancillary Data VPDMA interface for the Port B
VIP_FIQ_STATUS[7] OUTPUT_FIFO_PRTB_LUMA_STATUS	VIP_FIQ_MASK[7] OUTPUT_FIFO_PRTB_YUV_OF	OpPrtBYUV	Overflow at Luma VPDMA interface for Port B

Table 7-16. VIP_PARSER Interrupt Events (continued)

Event Flag	Event Mask	Map to	Description
VIP_FIQ_STATUS[6] OUTPUT_FIFO_PRTA_ANC_STATUS	VIP_FIQ_MASK[6] OUTPUT_FIFO_PRTA_ANC_OF	OpPrtAAnc	Overflow at Ancillary Data VPDMA interface for the Port A
VIP_FIQ_STATUS[4] OUTPUT_FIFO_PRTA_LUMA_STATUS	VIP_FIQ_MASK[4] OUTPUT_FIFO_PRTA_YUV_OF	OpPrtAYUV	Overflow at Luma VPDMA interface for Port A
VIP_FIQ_STATUS[3] ASYNC_FIFO_PRTB_STATUS	VIP_FIQ_MASK[3] ASYNC_FIFO_PRTB_OF	InPrtB	Overflow at Input Async FIFO for Port B
VIP_FIQ_STATUS[2] ASYNC_FIFO_PRTA_STATUS	VIP_FIQ_MASK[2] ASYNC_FIFO_PRTA_OF	InPrtA	Overflow at Input Async FIFO for Port A
VIP_FIQ_STATUS[1] PRTB_VDET_STATUS	VIP_FIQ_MASK[1] PRTB_VDET_MASK	PrtBVdet	Video Detect Interrupt for Port B
VIP_FIQ_STATUS[0] PRTA_VDET_STATUS	VIP_FIQ_MASK[0] PRTA_VDET_MASK	PrtAVdet	Video Detect Interrupt for Port A

A ‘1’ in the Status register associated with an Interrupt source shows that the interrupt source is pending. The Status register is read-only. To clear a bit in the Status register, the associated bit in the Clear register must be written with a ‘1.’

A ’1’ in the bit position of the Mask register associated with an Interrupt source ensures that the hardware interrupt will never be passed on to the VIP top level. A ‘0’ in the bit position of the Mask register associated with an Interrupt source will cause the interrupt controller to see a VIP_PARSER interrupt in the event the hardware in the parser triggers it.

A ‘1’ in the bit position of the Clear register associated with an Interrupt source clears the hardware interrupt status register until the next time the hardware triggers it. After a Clear, the CPU should set the bit back to a ‘0.’ Otherwise, the hardware would not be able to set any subsequent interrupts of the same type.

7.4.5.13 VDET Interrupt

For Line Multiplexing Embedded Sync mode only, the Meta Data Header includes a Video Detect (VDET) flag. The device sets this VDET flag whenever NTSC or PAL sync is found. Some other external devices using Line Multiplexing mode may not use VDET. However, when VDET changes, a VDET interrupt is issued (see Table 7-16, *Interrupts*, for more details on the interrupt). Each Pixel Clock Input Domain (Port A and Port B) has a separate VDET interrupt.

The VDET status register is comprised of 32 bits, each bit representing the value of the VDET flag found in the meta data of the Channel ID. Bit 0 is the VDET value from Channel ID 0, Bit 1is is VDET value from Channel ID 1, and so on. There is a separate status register for each Pixel Input Clock Domain (VIP_PORT_A_VDET_VEC and VIP_PORT_B_VDET_VEC registers).

In Line Mux mode, the meta-data field defining the srcnum is 5-bits wide. Only the last three bits of this field and the upper two bits are reserved. This bit should always be set to 1 in TI Line Mux mode.

7.4.5.14 Source Video Size

For each Pixel Input Clock Domain, status registers are available to log the last active video height and width found from 16 camera sources. There is no interrupt activated on the change in the source size in any of the input sources. These readonly registers only inform the application of the width and the height of the last active field or frame associated with each channel ID.

In 2x/4x pixel multiplexing, the four bit nibble carrying the Srcnum defines a maximum of 16 sources.

7.4.5.15 Clipping

In ITU-656/BT.1120 embedded sync streams, the values 0x00 and 0xFF are reserved for sync detection. These values are illegal in the rest of the stream. Only when the ITU-1364 standard came out to for digitally inserted vertical blanking data structures did the 0x00 and 0xFF codes get re-used in the packet synchronization structure.

The VIP_PARSER supports one configuration bit that, when enabled, changes all 0x00 to 0x01 and 0xFF to 0xFE in the vertical ancillary data. Another configuration bit, when enabled, changes all 0x00 to 0x01 and 0xFF to 0xFE in the active video portion of the input picture.

Generally, clipping is only desired for discrete sync input data captured from a NTSC/PAL decoder type of device which does not follow pixel range rules. For Discrete Sync input, the possibility to clip inputs to legal values exists. Clipping is desired if the picture sent to DRAM will be streamed out of the IC again usin an ITU-656/BT.1120 style output port. If the picture is processed inside the SOC before it is streamed out, then the processing procedure or the output streaming hardware needs to ensure that illegal values are not in the stream.

For Embedded Sync input, illegal values should not exist except for the ITU-1364 data sync sequence 00-FF-FF. Otherwise, the VIP Parser cannot determine good EAV/SAV. In the hardware, clipping is allowed for Embedded Sync streams even though it is not particularly a useful feature.

Note that if the clipping is enabled for ancillary data, the post processing software will never be able to find a data packet sync header, since the 00-FF-FF sequence will be changed to 01-FE-FE.

For 24-bit YUV, clipping is done on each 8 bit channel. If data[23:16]==0xFF, the clipped value will be 0xFE. If data[23:16]==0x00, the clipped value will be 0x01. Likewise, clipping is done for data bit ranges 15:8 and 7:0

From a software point of view, clipping should never be enabled for 24-bit RGB. RGB should use the full 8-bit quantization range for each color component. The hardware, however, will clip RGB in active video, if VIP_MAIN[5] CLIP_ACTIVE='1'. The clipping will be done on each 8-bit channel as described for 24-bit YUV.

7.4.5.16 Current and Last FID Value

The FID values for the current field or frame are reported in the Status Registers. When a new field or frame enters, the current FID values are saved into the previous FID status registers and the new FID value is loaded into the current FID register.

Following a reset, the previous and current FID status registers are set to '1.' The first two fields or frames are ignored. On the third input field or frame after a reset, the previous FID is loaded with the current FID ('1'), and the current FID is loaded with the actual FID. By the fourth field or frame after a reset, both the previous and current FID values should represent the values found in the input stream.

The FID values are reported for each camera source in both Pixel Input Clock Domains.

7.4.5.17 Disable Handling

A feature was defined for the case of single stream (either discrete sync or embedded sync) input handling where VIP_PORT_A[8] ENABLE for Port A and VIP_PORT_B[8] ENABLE for Port B is taken inactive. The single stream case was deemed more important than the multi-stream one since the output of the VIP_PARSER may be used to drive the Scaler module. The Scaler needs to work on a frame boundary (startframe to endframe) or it may lock up without a reset. The goal of the disable handling for the single stream case is to complete a field or frame of output data to the downstream module. Then, upon enabling of the port again, the system should start up properly without a need to reset the individual modules within the VIP instance.

In this scenario, suppose the VIP_PARSER has been processing a single input stream. Then, ENABLE is brought inactive. The VIP_PARSER will continue to output data downstream until it sends out an endframe pixel and the downstream module accepts the endframe pixel.

7.4.5.18 Picture Size Interrupt

Each VIP port can be set up to trigger an interrupt if the picture size varies from a pre-programmed expected picture size. This interrupt is supported only for the Active Video portion of the input video and not for the Vertical Ancillary portion. Also this interrupt is only support for source number 0 in multi channel capture.

The interrupts are named PrtASrc0Size and PrtBSrc0Size. They are described in Table 7-16, *VIP_PARSER Interrupt Events*

For Port A, the expected active video picture size values are programmed in VIP_XTRA_PORT_A[11:0] SRC0_NUMLINES and VIP_XTRA_PORT_A[27:16] SRC0_NUMPIX. For PortB, the expected active video picture size values are programmed in VIP_XTRA_PORT_B[11:0] SRC0_NUMLINES and VIP_XTRA_PORT_B[27:16] SRC0_NUMPIX.

NOTE: Picture Size Interrupt reflects the Active Video going into the DRAM. If cropping is enabled for Srcnum=0, the Picture Size is the post-cropped size.

7.4.5.19 Discrete Sync Signals

External ICs generally produce discrete sync interface signals seen in Figure 7-52.

VBLNKrepresents the vertical blanking interval. Generally, vertical blanking is at the top of a NTSC/PAL field. In certain standards, the last few lines of a field or frame are in vertical blanking in addition to the beginning few lines in the following field or frame.

VSYNC is the vertical sync indicator. VSYNC is active during a portion of the vertical blanking. For NTSC and PAL, since these standards define an odd number of lines for a field pair, VSYNC can be use in conjunction with HSYNC to determine FID polarity. Generally, VSYNC is defined to transition inactive to active sometime during vertical blanking. This signal then transitions to an inactive state before the end of vertical blanking. Certain standards define the line numbers where VSYNC transitions.

HBLNK is the horizontal blanking interval for each line. The horizontal blanking is the same number of pixels whether the line is in the active video region or in the vertical blanking region of the scan.

ACTVID is the region of a line that is active video. It is the inverse of the HBLNK signal. The number of pixels in the ACTVID region is the same for a line in vertical blanking as a line in active video. ACTVID(1) is a situation where the signal toggles in vertical blanking as well as active video. ACTVID(2) shows the signal toggling only in non-vertical blanking regions. Once ACTVID transitions active, it stays active for every PIXCLK until the end of the line.

HSYNC transitions from inactive to active for the first pixel of each line, which is a horizontal blanking pixel. HSYNC will transition to the inactive state before the end of the line. HSYNC is similar to HBLNK in that they both transition active on the same PIXCLK cycle. However, HBLNK transitions inactive at the end of the horizontal blanking period. HSYNC can transition inactive either before or after the horizontal blanking period.

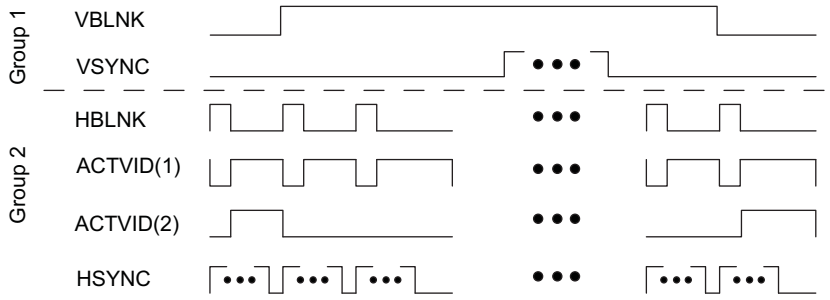
Group 1 signals define the vertical separation between fields or frames. Group 2 signals define the separation between lines. One of the Group 1 signals can be tied to the VSYNC input. The ACTVID(1) and ACTVID(2) signals from Group 2 are tied to the ACTVID input. One of the other two Group 2 signals, HBLNK or HSYNC, can be tied to the HSYNC input.

VIP_PORT_A[15] USE_ACTVID_HSYNC_N for Port A and VIP_PORT_B[15] USE_ACTVID_HSYNC_N for Port B defines whether the line separation method uses the signal from the ACTVID or the HSYNC input of the VIP_PARSER module.

VIP_PORT_A[22] DISCRETE_BASIC_MODE for Port A VIP_PORT_B[22] DISCRETE_BASIC_MODE for Port B determines whether discrete sync works as described in Section 7.4.5.6 *Input Data Interface* or whether a “basic mode” input handler is invoked.

By choosing one signal from Group 1 and one signal from Group 2, there should be a way to capture the external data.

Figure 7-52. Generic External Sync Signals



NOTE: 1. VIP_PARSER module defines three discrete sync control signals: ACTVID, HSYNC, and VSYNC.

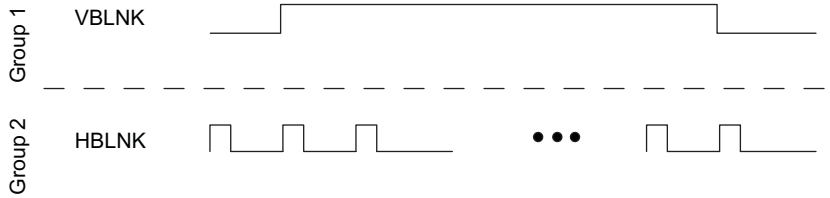
2. In order to capture external data PIXCLK must never stop, for either horizontal or vertical blanking.

7.4.5.19.1 VBLNK and HBLNK

Figure 7-53 shows VBLNK from Group 1 and HBLNK from Group 2 being used. In this case, set USE_ACTVID_HSYNC_N=’0’ and DISCRETE_BASIC_MODE=’0’. Vertical Ancillary lines will be sent to the Vertical Ancillary output at Active Video will be sent out the Active Video output.

If DISCRETE_BASIC_MODE=’1’ is chosen, all lines including vertical blanking ones will be sent to the Active Video buffer. Since a line is delineated by HBLNK and HBLNK toggles in vertical blanking as well as active video, every incoming pixel will be save to the Active Video buffer.

Figure 7-53. vblnk and hblnk

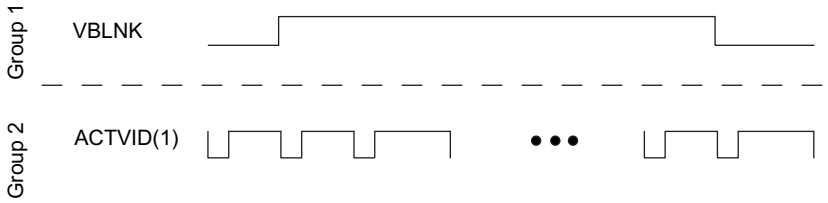


7.4.5.19.2 BLNK and ACTVID (1)

Figure 7-54 shows VBLNK from Group 1 and ACTVID(1) from Group 2 being used. ACTVID is toggling during Vertical Blanking. Set USE_ACTVID_HSYNC_N='1' and DISCRETE_BASIC_MODE='0'. Vertical Ancillary lines will be sent to the Vertical Ancillary output and Active Video will be sent out the Active Video output.

If DISCRETE_BASIC_MODE='1' is chosen, all lines will be sent to the Active Video output.

Figure 7-54. VBLNK and ACTID (1)

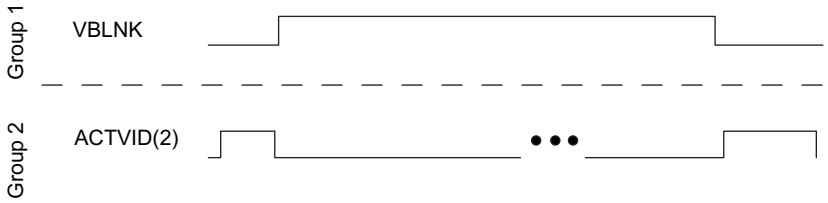


7.4.5.19.3 VBLNK and ACTVID(2)

Figure 7-55 shows VBLNK from Group 1 and ACTVID(2) from Group 2 being used. ACTVID is not toggling during Vertical Blanking. Set USE_ACTVID_HSYNC_N='1' and DISCRETE_BASIC_MODE='1'. Since there are no line sync/clocking signals in the Vertical Blanking period, only Active Video lines will be sent to the Active Video output.

If DISCRETE_BASIC_MODE='0' is set, then the hardware will lock up as there is no way for it to determine a frame boundary.

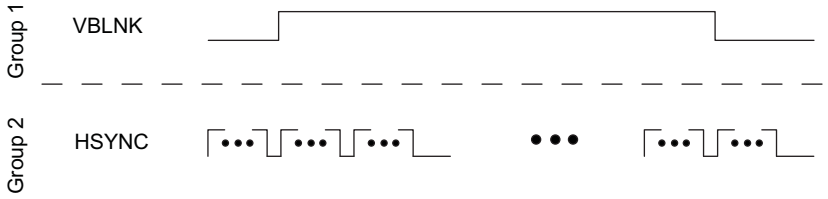
Figure 7-55. VBLNK and ACTVID(2)



7.4.5.19.4 VBLNK and HSYNC

Figure 7-56 shows VBLNK from Group 1 and HSYNC from Group 2 being used. In this scenario, set USE_ACTVID_HSYNC_N='0' and DISCRETE_BASIC_MODEN='0'. Vertical Ancillary lines will be sent to the Vertical Ancillary output at Active Video will be sent out the Active Video output.

Figure 7-56. VBLNK and HSYNC



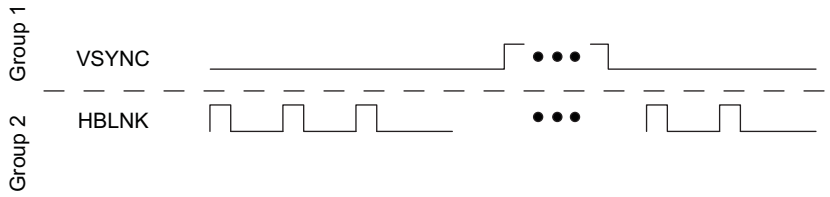
7.4.5.19.5 VSYNC and HBLNK

Figure 7-57 shows VSYNC from Group 1 and HBLINK from Group 2 being used. Set USE_ACTVID_HSYNC_N='0' and DISCRETE_BASIC_MODE='1'.

Also, no automatic parsing of vertical ancillary data will be performed so the Ancillary VPI port to the VPDMA should be disabled. All lines, including both vertical ancillary and active video, will appear in the Video DRAM buffer. Lines starting after an inactive to active transition on VSYNC will delineate a start of frame. Every data element strobed on the Pixel clock's active edge will be stored in the Active Video Buffer.

In Figure 7-57, it is likely the HBLINK will toggle when VSYNC is active. In this case, setting USE_ACTVID_HSYNC_N='0' and DISCRETE_BASIC_MODE='0' mean that those lines appearing under the active VSYNC will be sent to the Ancillary Data Buffer. All other captured lines will be sent to the Active Video Buffer.

Figure 7-57. VSYNC and HBLNK

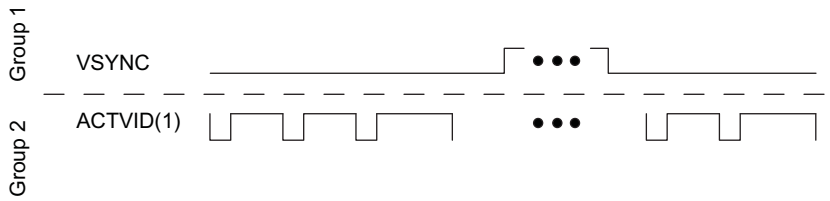


7.4.5.19.6 VSYNC and ACTIVID(1)

Figure 7-58 shows VSYNC from Group 1 and ACTIVID(1) from Group 2 being used. ACTIVID is toggling during the VBLNK interval. ACTIVID does not necessarily toggle during VSYNC. Set USE_ACTVID_HSYNC_N='1' and DISCRETE_BASIC_MODE='1'.

Also, no automatic parsing of vertical ancillary data will be performed so there is no activity on the Ancillary VPI port to the VPDMA. All lines, including both vertical ancillary and active video, will appear in the Video DRAM buffer. Lines starting after an inactive to active transition on VSYNC will delineate a start of frame. Lines are denoted by an inactive to active transition of ACTIVID. Only those pixels gated by an active ACTIVID will be saved.

Figure 7-58. VSYNC and ACTIVID(1)

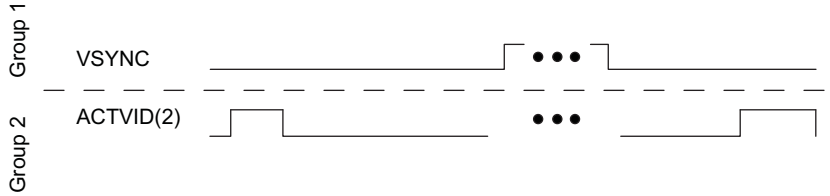


7.4.5.19.7 VSYNC and ACTIVID(2)

Figure 7-59 shows VSYNC from Group 1 and ACTIVID(2) from Group 2 being used. ACTIVID is not toggling during the entire VBLNK interval. Set USE_ACTVID_HSYNC_N='1' and DISCRETE_BASIC_MODE='1'.

Also, no automatic parsing of vertical ancillary data will be performed so the the Ancillary VPI port to the VPDMA should be turned off. All active video lines, since there are no vertical ancillary data lines, will appear in the Video DRAM buffer. Lines starting after an inactive to active transition on VSYNC will delineate a start of frame. Lines are denoted by an inactive to active transition of ACTIVID. Once a line starts, ACTIVID stays active for every pixel clock until the end of the line. Only those pixels gated by an active ACTIVID will be saved.

Figure 7-59. VSYNC and ACTIVID(2)

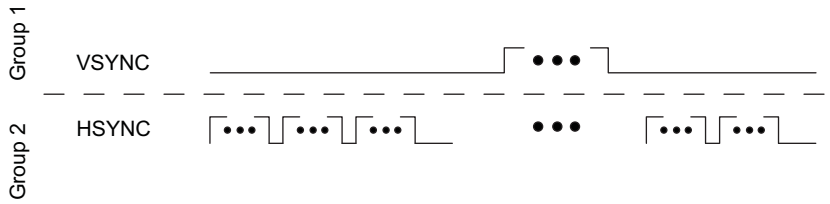


7.4.5.19.8 VSYNC and HSYNC

Figure 7-60 shows VSYNC from Group 1 and HSYNC from Group 2 being used. Set USE_ACTVID_HSYNC_N='0' and DISCRETE_BASIC_MODE='1'.

In the event that the machine is set to DISCRETE_BASIC_MODE='0', lockup will not occur as long as there is an HSYNC active transition during the time that VSYNC is active. This scenario is likely. However, if there is not at least one such transition on HSYNC, then the machine experiences a lock up. It cannot distinguish the end of one frame from the start of the next frame.

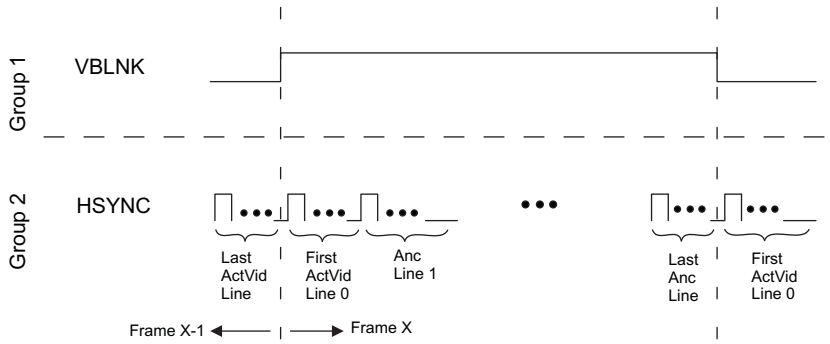
Figure 7-60. VSYNC and HSYNC



7.4.5.19.9 Line and Pixel Capture Examples

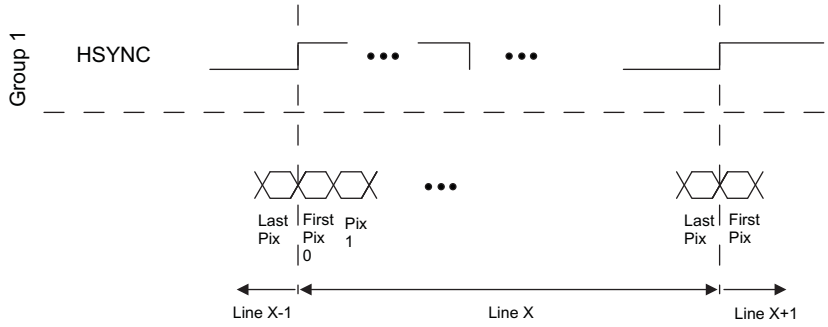
When DISCRETE_BASIC_MODE='0', VBLNK is generally used. All the lines where the start of line is under an active VBLNK are sent to the Ancillary Data buffer. All the lines where the start of line is not under an active VBLNK are sent to the Active Video framebuffer. This situation is shown in Figure 7-61.

Figure 7-61. Ancillary and Active Video Line Determination



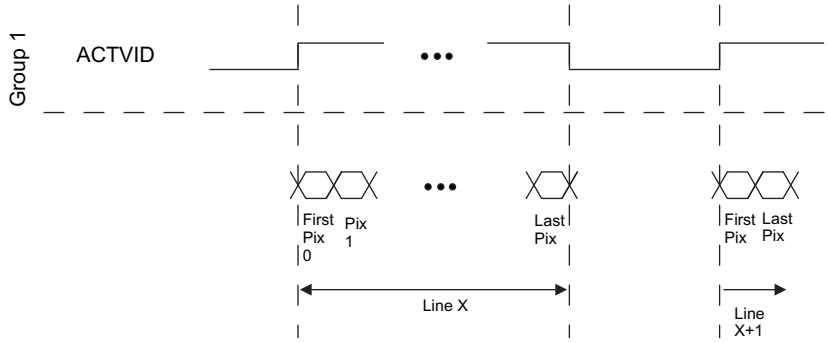
The start of line is the pixel represented by the inactive to active transition on HSYNC when USE_ACTVID_HSYNC_N = '1'. Figure 7-62 illustrates the delineation of a line when using USE_ACTVID_HSYNC_N = '1.'

Figure 7-62. HSYNC Pixel Capture



The start of line is the pixel represented by the inactive to active transition on ACTVID when USE_ACTVID_HSYNC_N = '0.' Note that ACTVID stays active for the entire duration of active video portion of the line. This scenario is shown in Figure 7-63

Figure 7-63. ACTVID Pixel Capture



In 8-bit mode, the 4:2:2 YUV input color component order is Cb, Y followed by Cr and Y. For 16-bit input mode, all the components are sent in the same cycle.

7.4.5.20 VIP Overflow Detection and Recovery

It is possible that an overflow can occur in the VIP_PARSER. Overflow detection is determined by reading the VIP_FIQ_STATUS register and checking for bits 8, 7, 5, 4, 3 and 2. If video is being captured, and any of these bits are set, it indicates that not all of the incoming video data was sent to DDR meory. VIP overflow can be caused by one of the following:

1. External pixel clock is faster than processing clock

2. DDR bandwidth is temporarily over-consumed

3. VIP scaler is being used inline with external video input, and is upscaling.

- VIP scaler in this use case can only be used for downscaling

4. VIP scaler is being used inline with external video input, but has not been configured with scaler coefficients

- VIP scaler will not accept video input if it is not first configured with scaler coefficients. This will cause overflow

5. VIP scaler is being used inline, but has not been enabled

6. External cables are connected or disconnected while the system is running, resulting in corrupted video streams going into the VIP

7. Bad external video cable, which causes corrupted video streams going into the VIP

Items 6 and 7 above are typically seen as noise events, where it is likely that multiple horizontal syncs per line and/or multiple vertical syncs per frame will be observed. These result in high peak throughput requirements, leading to DDR bandwidth being temporarily over-consumed, and thus VIP overflow.

The high level recovery method for VIP overflow on Port A is outlined in the steps below. Port B is similar.

1. Set VIP_XTRA6_PORT_A[31:16] YUV_SRCNUM_STOP_IMMEDIATELY = 0xFFFF_FFFF
2. Set VIP_XTRA6_PORT_A[15:0] ANC_SRCNUM_STOP_IMMEDIATELY = 0xFFFF_FFFF
3. Set VIP_PORT_A[8] ENABLE = 0
4. Set VIP_PORT_A[7] CLR_ASYNC_FIFO_RD and VIP_PORT_A[6] CLR_ASYNC_FIFO_WR to 1
5. Set VIP_PORT_A[23] SW_RESET to 1
6. Reset other VIP modules
   - For each module used downstream of VIP_PARSER, write 1 to the bit location of the VIP_CLKC_RST register which is connected to VIP_PARSER
7. Abort VPDMA channels
   - Write to list attribute to stop list 0
   - Write to list address register location of abort list
   - Write to list attribute register list 0 and size of abort list
8. Set VIP_PORT_A[23] SW_RESET to 0
9. Un-reset other VIP modules
   - For each module used downstream of VIP_PARSER, write 0 to the bit location of the VIP_CLKC_RST register which is connected to VIP_PARSER
10. (Delay)
11. SC coeff downloaded (if VIP_SCALER is being used)
12. (Delay)
13. Set VIP_XTRA6_PORT_A[31:16] YUV_SRCNUM_STOP_IMMEDIATELY = 0x0000_0000
14. Set VIP_XTRA6_PORT_A[15:0] ANC_SRCNUM_STOP_IMMEDIATELY = 0x0000_0000
15. Set VIP_PORT_A[8] ENABLE = 1
16. Set VIP_PORT_A[7] CLR_ASYNC_FIFO_RD and VIP_PORT_A[6] CLR_ASYNC_FIFO_WR to 0

7.4.6 VIP Color Space Converter (CSC)

The Color Space Converter (CSC) module is used to convert video data from one color space to another with nine programmable integer multipliers.

7.4.6.1 CSC Features

- All parameters are programmable
- Each parameter is configurable in signed 13-bits
- Support for Bypass Mode

7.4.6.2 CSC Functional Description

The conversion between the different color spaces requires addition and multiplication operations on color and intensity components. The mathematical expression of the conversion can be written as:

```
Y   = A0*R + B0*G + C0*B + D0
Cb  = A1*R + B1*G + C1*B + D1
Cr  = A2*R + B2*G + C2*B + D2
```

Color space coeficients are set through the following registers:

- For luma component:
  - VIP_CSC00[12:0] A0
  - VIP_CSC00[28:16] B0

  - VIP_CSC01[28:16] C0
  - VIP_CSC04[27:16] D0
- For Cb component:
  - VIP_CSC01[28:16] A1
  - VIP_CSC02[12:0] B1
  - VIP_CSC02[27:16] C1
  - VIP_CSC05[11:0] D1
- For Cr component :
  - VIP_CSC03[12:0] A2
  - VIP_CSC03[27:16] B2
  - VIP_CSC04[12:0] C2
  - VIP_CSC05[27:16] D2

Using YUV to RGB conversion as an example: YUV represents one color space and RGB represents another color space. The conversion can be written in the matrix format shown in Figure 7-64.

Figure 7-64. Matrix Format

$$\begin{bmatrix} Y \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} A0 & B0 & C0 \\ A1 & B1 & C1 \\ A2 & B2 & C2 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} + \begin{bmatrix} D0 \\ D1 \\ D2 \end{bmatrix}$$

Since HDTV and SDTV have different conversion requirements, both conversions of RGB-to-YCbCr and YCbCr-to-RGB are described. The details of derivations of these matrixes will be given in the following subsections.

7.4.6.2.1 HDTV Application

7.4.6.2.1.1 HDTV Application with Video Data Range

The two equations presented in this section are for the HDTV application. The chromaticity parameters are defined by ITU-R709 standard.

The input video data for these equations should be within the range that is defined for video application.

In an 8-bit system:

- Rd, Gd, Bd, and Yd will be in the range [16-235]
- Cb and Cr will be the range [16-240]
- D = 1

In a 10-bit system:

- Rd, Gd, Bd, and Yd will be in the range [64-940]
- Cb and Cr will be in the range [64-960]
- D = 4

Conversion from RGB to YCbCr:

Figure 7-65. Conversion from RGB to YCbCr

$$\begin{bmatrix} Yd \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 0.2126 & 0.7152 & 0.0722 \\ -0.1172 & -0.3942 & 0.5114 \\ 0.5114 & -0.4646 & -0.0468 \end{bmatrix} \begin{bmatrix} Rd \\ Gd \\ Bd \end{bmatrix} + \begin{bmatrix} 0 \\ 128 \\ 128 \end{bmatrix} D$$

Conversion from YCbCr to RGB:

Figure 7-66. Conversion from YCbCr to RGB

$$\begin{bmatrix} Rd \\ Gd \\ Bd \end{bmatrix} = \begin{bmatrix} 1 & 0 & 1.5396 \\ 1 & -0.1831 & -0.4577 \\ 1 & 1.8142 & 0 \end{bmatrix} \begin{bmatrix} Yd \\ Cb \\ Cr \end{bmatrix} + \begin{bmatrix} -197 \\ 82 \\ -232 \end{bmatrix} D$$

7.4.6.2.1.2 HDTV Application with Graphics Data Range

The two equations presented in this section are for the HDTV application with graphics range data input. The main application is for computer graphics display. The chromaticity parameters are defined by ITU-R709 standard.

The input data ranges for these equations are as follows.

In an 8-bit system:

- Rd, Gd, Bd, Yd, Cb and Cr will be the range [0-255]
- D = 1

In a 10-bit system:

- Rd, Gd, Bd, Yd, Cb and Cr will be in the range [0-1023]
- D = 4

Conversion from RGB to YCbCr:

Figure 7-67. Conversion from RGB to YCbCr

$$\begin{bmatrix} Yd \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 0.1826 & 0.6142 & 0.0620 \\ -0.1006 & -0.3385 & 0.4392 \\ 0.4392 & -0.3990 & -0.0402 \end{bmatrix} \begin{bmatrix} Rd \\ Gd \\ Bd \end{bmatrix} + \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} D$$

Conversion from YCbCr to RGB:

Figure 7-68. Conversion from YCbCr to RGB

$$\begin{bmatrix} Rd \\ Gd \\ Bd \end{bmatrix} = \begin{bmatrix} 1.1644 & -0.0003 & 1.7927 \\ 1.1644 & -0.2132 & -0.5329 \\ 1.1642 & 2.1125 & -0.0001 \end{bmatrix} \begin{bmatrix} Yd \\ Cb \\ Cr \end{bmatrix} + \begin{bmatrix} -248 \\ 77 \\ -289 \end{bmatrix} D$$

7.4.6.2.1.3 Quantized Coefficients for Color Space Converter in HDTV

This section quantizes all the coefficients of color space conversion based on a 10-bit system for HDTV application. These coefficients can be input to the registers of programmable color space converter. Refer to register section for the corresponding register setting.

Table 7-17. Quantized Coefficients of HDTV Application with Video Data Range

Conversion from RGB to YCbCr					Conversion from YCbCr to RGB			
Coefficient Names	Registers	Real Number Format	Quantized Format	Hex Format	Coefficient Names	Real Number Format	Quantized Format	Hex Format
A0(13-bit)	VIP_CSC00[12:0] A0	0.2126	218	0x00DA	A0(13-bit)	1	1024	0x0400
B0(13-bit)	VIP_CSC00[28:16] B0	0.7152	732	0x02DC	B0(13-bit)	0	0	0x0000
C0(13-bit)	VIP_CSC01[28:16] C0	0.0722	74	0x004A	C0(13-bit)	1.5396	1577	0x0629
A1(13-bit)	VIP_CSC01[28:16] A1	-0.1172	-120	0x1F88	A1(13-bit)	1	1024	0x0400
B1(13-bit)	VIP_CSC02[12:0] B1	-0.3942	-404	0x1E6C	B1(13-bit)	-0.1831	-187	0x1F45
C1(13-bit)	VIP_CSC02[27:16] C1	0.5114	524	0x020C	C1(13-bit)	-0.4577	-469	0x1E2B
A2(13-bit)	VIP_CSC03[12:0] A2	0.5114	524	0x020C	A2(13-bit)	1	1024	0x0400
B2(13-bit)	VIP_CSC03[27:16] B2	-0.4646	-476	0x1E24	B2(13-bit)	1.8142	1858	0x0742
C2(13-bit)	VIP_CSC04[12:0] C2	-0.0468	-48	0x1FD0	C2(13-bit)	0	0	0x0000
D0(12-bit)	VIP_CSC04[27:16] D0	0	0	0x000	D0(12-bit)	-197	-788	0xCEC
D1(12-bit)	VIP_CSC05[11:0] D1	128	512	0x200	D1(12-bit)	82	328	0x148
D2(12-bit)	VIP_CSC05[27:16] D2	128	512	0x200	D2(12-bit)	-232	-928	0xC60

Table 7-18. Quantized Coefficients of HDTV Application with Graphics Data Range

Conversion from RGB to YCbCr					Conversion from YCbCr to RGB			
Coefficient Names	Registers	Real Number Format	Quantized Format	Hex Format	Coefficient Names	Real Number Format	Quantized Format	Hex Format
A0(13-bit)	VIP_CSC00[12:0] A0	0.1826	187	0x00BB	A0(13-bit)	1.1644	1192	0x04A8
B0(13-bit)	VIP_CSC00[28:16] B0	0.6142	629	0x0275	B0(13-bit)	-0.0003	0	0x0000
C0(13-bit)	VIP_CSC01[28:16] C0	0.062	63	0x003F	C0(13-bit)	1.7927	1836	0x072C
A1(13-bit)	VIP_CSC01[28:16] A1	-0.1006	-103	0x1F99	A1(13-bit)	1.1644	1192	0x04A8
B1(13-bit)	VIP_CSC02[12:0] B1	-0.3385	-347	0x1EA5	B1(13-bit)	-0.2132	-218	0x1F26
C1(13-bit)	VIP_CSC02[27:16] C1	0.4392	450	0x01C2	C1(13-bit)	-0.5329	-546	0x1DDE
A2(13-bit)	VIP_CSC03[12:0] A2	0.4392	450	0x01C2	A2(13-bit)	1.1642	1192	0x04A8
B2(13-bit)	VIP_CSC03[27:16] B2	-0.399	-409	0x1E67	B2(13-bit)	2.1125	2163	0x0873
C2(13-bit)	VIP_CSC04[12:0] C2	-0.0402	-41	0x1FD7	C2(13-bit)	-0.0001	0	0x0000
D0(12-bit)	VIP_CSC04[27:16] D0	16	64	0x040	D0(12-bit)	-248	-992	0xC20
D1(12-bit)	VIP_CSC05[11:0] D1	128	512	0x200	D1(12-bit)	77	308	0x134
D2(12-bit)	VIP_CSC05[27:16] D2	128	512	0x200	D2(12-bit)	-289	-1156	0xB7C

7.4.6.2.2 SDTV Application

7.4.6.2.2.1 SDTV Application with Video Data Range

The two equations presented in this section are for the SDTV application. The chromaticity parameters are defined by ITU-R601 standard.

The input video data for these equations should be within the range that is defined for video application.

In an 8-bit system:

- Rd, Gd, Bd, and Yd will be in the range [16-235]
- Cb and Cr will be the range [16-240]
- D = 1

In a 10-bit system:

- Rd, Gd, Bd, and Yd will be in the range [64-940]
- Cb and Cr will be in the range [64-960]
- D = 4

Conversion from RGB to YCbCr:

Figure 7-69. Conversion from RGB to YCbCr

$$\begin{bmatrix} Yd \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ -0.172 & -0.339 & 0.511 \\ 0.511 & -0.428 & -0.083 \end{bmatrix} \begin{bmatrix} Rd \\ Gd \\ Bd \end{bmatrix} + \begin{bmatrix} 0 \\ 128 \\ 128 \end{bmatrix} D$$

Conversion from YCbCr to RGB:

Figure 7-70. Conversion from YCbCr to RGB

$$\begin{bmatrix} Rd \\ Gd \\ Bd \end{bmatrix} = \begin{bmatrix} 1 & -0.0003 & 1.3717 \\ 1 & -0.3365 & -0.6984 \\ 1 & 1.7336 & -0.0016 \end{bmatrix} \begin{bmatrix} Yd \\ Cb \\ Cr \end{bmatrix} + \begin{bmatrix} -176 \\ 132 \\ -222 \end{bmatrix} D$$

7.4.6.2.2.2 SDTV Application with Graphics Data Range

The two equations presented in this section are for the SDTV application with graphics range data input. The main application is for computer graphics display. The chromaticity parameters are defined by ITU-R601 standard.

The input data ranges for these equations are as following.

In an 8-bit system:

- Rd, Gd, Bd, Yd, Cb and Cr will be the range [0-255]
- D = 1

In a 10-bit system:

- Rd, Gd, Bd, Yd, Cb and Cr will be in the range [0-1023]
- D = 4

Conversion from RGB to YCbCr:

Figure 7-71. Conversion from RGB to YCbCr

$$\begin{bmatrix} Yd \\ Cb \\ Cr \end{bmatrix} = \begin{bmatrix} 0.257 & 0.504 & 0.098 \\ -0.148 & -0.291 & 0.439 \\ 0.439 & -0.368 & -0.071 \end{bmatrix} \begin{bmatrix} Rd \\ Gd \\ Bd \end{bmatrix} + \begin{bmatrix} 16 \\ 128 \\ 128 \end{bmatrix} D$$

Conversion from YCbCr to RGB:

Figure 7-72. Conversion from YCbCr to RGB

$$\begin{bmatrix} Rd \\ Gd \\ Bd \end{bmatrix} = \begin{bmatrix} 1.1641 & -0.0018 & 1.5958 \\ 1.1641 & -0.3914 & -0.8135 \\ 1.1641 & 2.0178 & -0.0012 \end{bmatrix} \begin{bmatrix} Yd \\ Cb \\ Cr \end{bmatrix} + \begin{bmatrix} -223 \\ 136 \\ -277 \end{bmatrix} D$$

7.4.6.2.2.3 Quantized Coefficients for Color Space Converter in SDTV

This section quantizes all the coefficients of color space conversion based on a 10-bit system for SDTV application. These coefficients can be input to the registers of programmable color space converter. Refer to register section for the corresponding register setting.

Table 7-19. Quantized Coefficients of SDTV Application with Video Data Range

Conversion from RGB to YCbCr					Conversion from YCbCr to RGB			
Coefficient Names	**Registers**	**Real Number Format**	**Quantized Format**	**Hex Format**	**Coefficient Names**	**Real Number Format**	**Quantized Format**	**Hex Format**
A0(13-bit)	VIP_CSC00[12:0] A0	0.299	306	0x0132	A0(13-bit)	1	1024	0x0400
B0(13-bit)	VIP_CSC00[28:16] B0	0.587	601	0x0259	B0(13-bit)	-0.0003	0	0x0000
C0(13-bit)	VIP_CSC01[28:16] C0	0.114	117	0x0075	C0(13-bit)	1.3717	1405	0x057D
A1(13-bit)	VIP_CSC01[28:16] A1	-0.172	-176	0x1F50	A1(13-bit)	1	1024	0x0400
B1(13-bit)	VIP_CSC02[12:0] B1	-0.339	-347	0x1EA5	B1(13-bit)	-0.3365	-345	0x1EA7
C1(13-bit)	VIP_CSC02[27:16] C1	0.511	523	0x020B	C1(13-bit)	-0.6984	-715	0x1D35
A2(13-bit)	VIP_CSC03[12:0] A2	0.511	523	0x020B	A2(13-bit)	1	1024	0x0400
B2(13-bit)	VIP_CSC03[27:16] B2	-0.428	-438	0x1E4A	B2(13-bit)	1.7336	1775	0x06EF
C2(13-bit)	VIP_CSC04[12:0] C2	-0.083	-85	0x1FAB	C2(13-bit)	-0.0016	-2	0x1FFE
D0(12-bit)	VIP_CSC04[27:16] D0	0	0	0x000	D0(12-bit)	-176	-704	0xD40
D1(12-bit)	VIP_CSC05[11:0] D1	128	512	0x200	D1(12-bit)	132	528	0x210
D2(12-bit)	VIP_CSC05[27:16] D2	128	512	0x200	D2(12-bit)	-222	-888	0xC88

Table 7-20. Quantized Coefficients of SDTV Application with Graphics Data Range

Conversion from RGB to YCbCr					Conversion from YCbCr to RGB			
Coefficient Names	Registers	Real Number Format	Quantized Format	Hex Format	Coefficient Names	Real Number Format	Quantized Format	Hex Format
A0(13-bit)	VIP_CSC00[12:0] A0	0.257	263	0x0107	A0(13-bit)	1.1641	1192	0x04A8
B0(13-bit)	VIP_CSC00[28:16] B0	0.504	516	0x0204	B0(13-bit)	-0.0018	-2	0x1FFE
C0(13-bit)	VIP_CSC01[28:16] C0	0.098	100	0x0064	C0(13-bit)	1.5958	1634	0x0662
A1(13-bit)	VIP_CSC01[28:16] A1	-0.148	-152	0x1F68	A1(13-bit)	1.1641	1192	0x04A8
B1(13-bit)	VIP_CSC02[12:0] B1	-0.291	-298	0x1ED6	B1(13-bit)	-0.3914	-401	0x1E6F
C1(13-bit)	VIP_CSC02[27:16] C1	0.439	450	0x01C2	C1(13-bit)	-0.8135	-833	0x1CBF
A2(13-bit)	VIP_CSC03[12:0] A2	0.439	450	0x01C2	A2(13-bit)	1.1641	1192	0x04A8
B2(13-bit)	VIP_CSC03[27:16] B2	-0.368	-377	0x1E87	B2(13-bit)	2.0178	2066	0x0812
C2(13-bit)	VIP_CSC04[12:0] C2	-0.071	-73	0x1FB7	C2(13-bit)	-0.0012	-1	0x1FFF
D0(12-bit)	VIP_CSC04[27:16] D0	16	64	0x040	D0(12-bit)	-223	-892	0xC84
D1(12-bit)	VIP_CSC05[11:0] D1	128	512	0x200	D1(12-bit)	136	544	0x220
D2(12-bit)	VIP_CSC05[27:16] D2	128	512	0x200	D2(12-bit)	-277	-1108	0xBAC

7.4.6.3 CSC Bypass Mode

CSC module can be bypassed by setting VIP_CSC05[28] BYPASS bit-field to 1.

7.4.7 VIP Scaler (SC)

This section describes the highly optimized video resizers, SC (scalers), in the VIP modules.

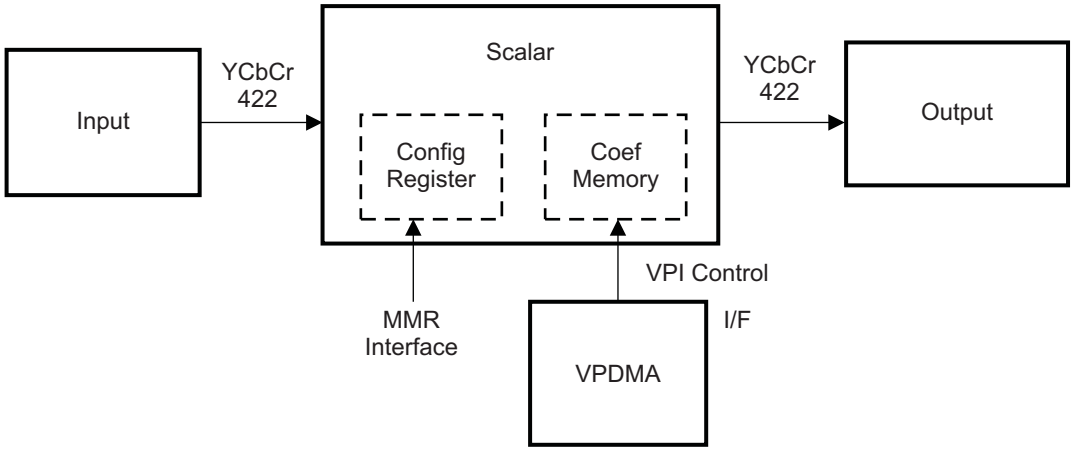
7.4.7.1 SC Features

- Independent vertical and horizontal up and down scaling
- Running average vertical down scaling for memory optimization
- Decimation and polyphase filtering for horizontal scaling
- Non-linear scaling for stretched/compressed left and right sides
- Input image trimmer for pan/scan support
- Pre-scaling peaking filter for enhanced sharpness
- Scale field as frame
- Interlacing of scaled output
- Full 1080p input and output support
- YCbCr422 input and output
- Maximum horizontal scaling ratio only limited by output line buffer (2047 pixels)
- Scaling filter Coefficient memory download via VPI (Video Port Interface) Control interface

7.4.7.2 SC Functional Description

Scaler takes in a 10-bit YCbCr 422 video frame from an upstream module, performs vertical/horizontal scaling and outputs a YCbCr422 scaled image to a next downstream module. All configurations are done via the MMR interface except for the scaler coefficient memory configuration that is done via the common VPI control interface bus by VPDMA. Figure 7-73 shows the high-level block diagram of the scaler module.

Figure 7-73. High Level Block Diagram

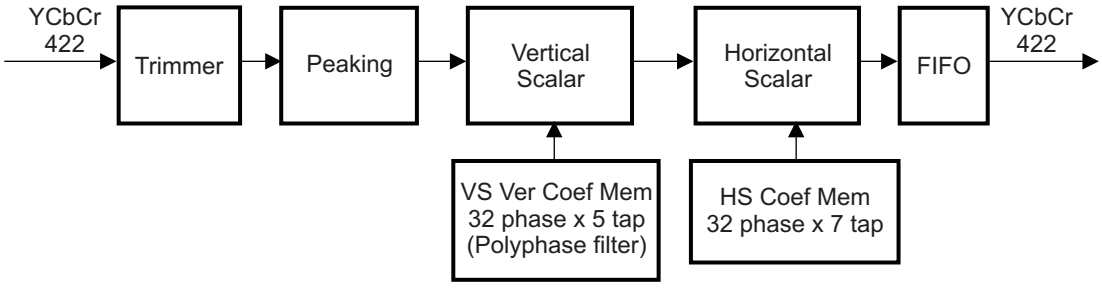


The SC is used in the video path and in all other video write-back data paths in the VIP module.

Scaling is performed in following three steps:

1. Trimming and Pre-peaking filtering
2. Vertical Scaling (Polyphase/Running Average Filter)
3. Horizontal polyphase scaling

Figure 7-74. SC Block Diagram



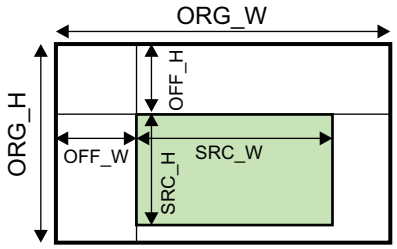
7.4.7.2.1 Trimmer

The trimmer can be programmed to re-define a new source image within the input video frame sent from an upstream module before it is sent to the scaling sub-modules. This feature enables small area zoom out, source pan/scan, or removal of unwanted area in the video (such as black box / curtains / noisy line-21 video) without modifying the VPDMA parameters.

Horizontal and vertical offset is set through VIP_CFG_SC25[26:16] CFG_OFF_W and VIP_CFG_SC25[10:0] CFG_OFF_H registers.

Width and height are set through VIP_CFG_SC24[26:16] CFG_ORG_W and VIP_CFG_SC24[10:0] CFG_ORG_H registers.

Figure 7-75. Input Image Trimming



NOTE: Width and height of the source image are global parameters and are set with VIP_CFG_SC5[22:12] CFG_SRC_W and VIP_CFG_SC5[10:0] CFG_SRC_H registers.

It is required that the input image frame (CFG_SRC_W x CFG_SRC_H) to be at least 32 × 32 (after trimming, if the trimming is enabled) to properly fill the input filter stage pipelines.

7.4.7.2.2 Peaking

The peaking block increases the amplitude of high frequency luminance information in horizontal direction to increase the sharpness of a video image before it is scaled. As shown in Figure 7-76, the high-frequency luminance is increased using an 11-tap High-Pass filter with adjustable gain. The non-linear coring function removes low-level noise and the modified luminance is then added to the original luminance signal. The implementation details are shown in Figure 7-76.

Figure 7-76. Filter Implementation and Parameter Description

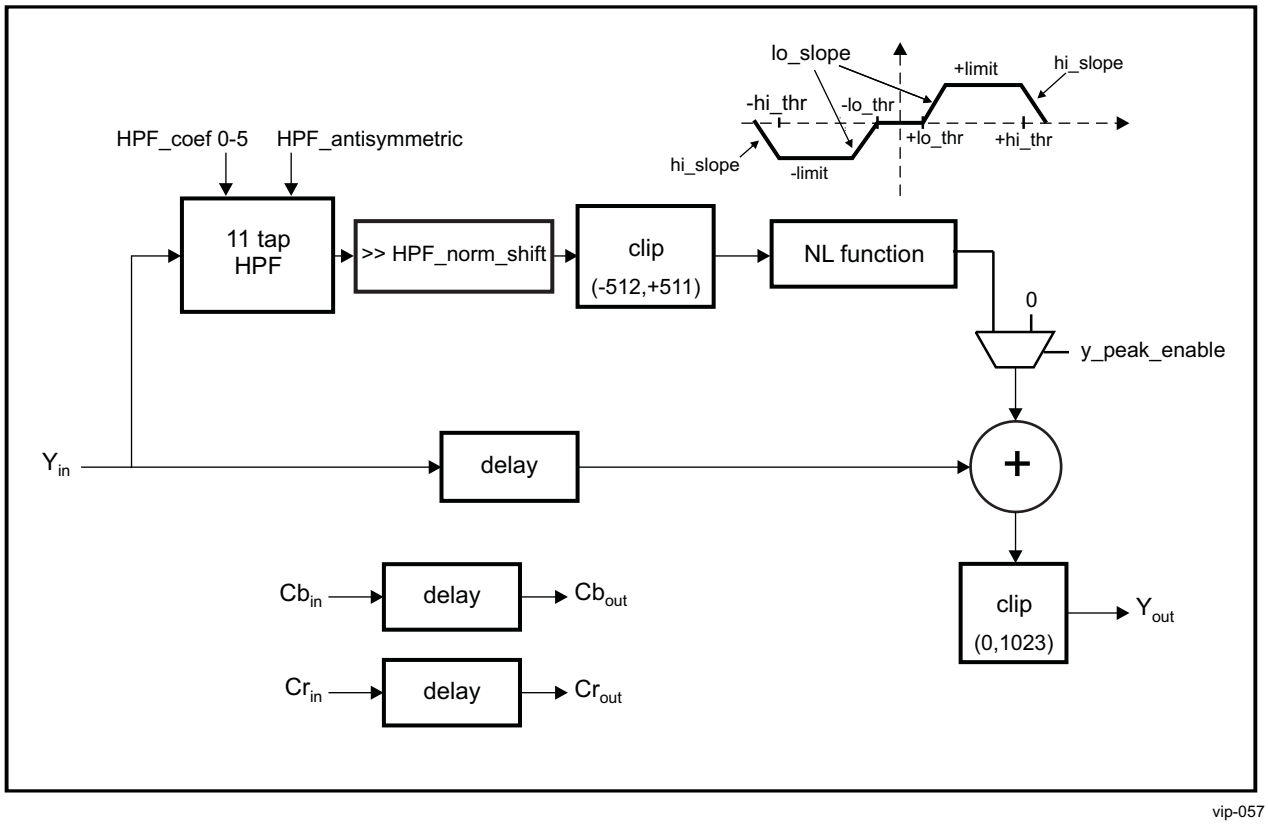
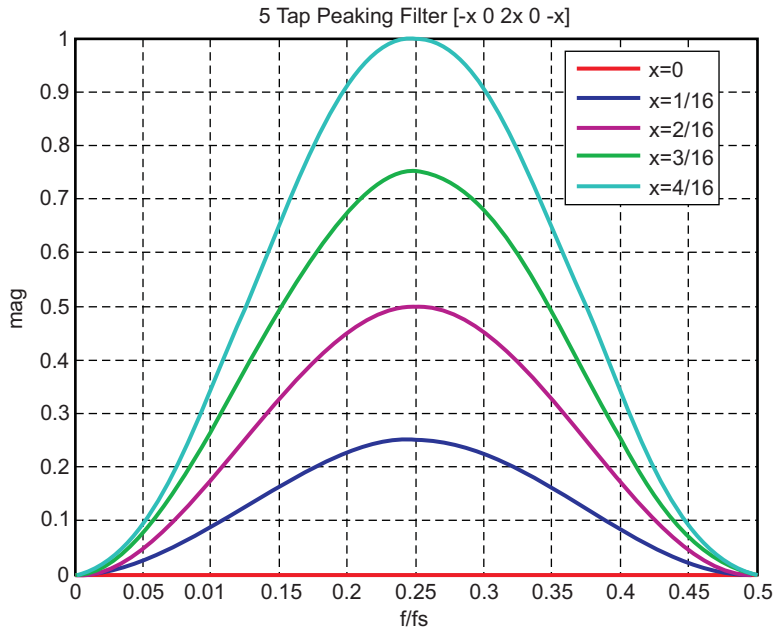


Table 7-21. Parameter Description

Parameter	Description	Bits	Default
VIP_CFG_SC19[7:0] CFG_HPF_COEF0 to VIP_CFG_SC20[15:8] CFG_HPF_COEF5	FIR coefficients	8	[0 0 0-4 0 8]
VIP_CFG_SC20[18:16] CFG_HPF_NORM_SHIFT	Right shift	3	4
VIP_CFG_SC21[8:0] CFG_NL_LO_THR	Coring threshold	9	16
VIP_CFG_SC22[8:0] CFG_NL_HI_THR	High threshold	9	400
VIP_CFG_SC21[23:16] CFG_NL_LO_SLOPE	Lo slope= - CFG_NL_LO_SLOPE/16	8	16
VIP_CFG_SC22[18:16] CFG_NL_HI_SLOPE_SHIFT	Hi slope = $2^{(CFG\_NL\_HI\_SLOPE\_SHIFT-3)}$	3	4
VIP_CFG_SC20[28:20] CFG_NL_LIMIT	Clipping limit	9	200
VIP_CFG_SC0[14] CFG_Y_PK_EN	Control	1	0

Parameters for the Peaking filters are defined in VIP_CFG_SC19 through VIP_CFG_SC22 registers. The frequency responses of the peaking-filter with different sets of coefficients are shown in Figure 7-77. If the source of the input video is NTSC or PAL format, the peaking filter can be configured to reject the color subcarrier frequency.

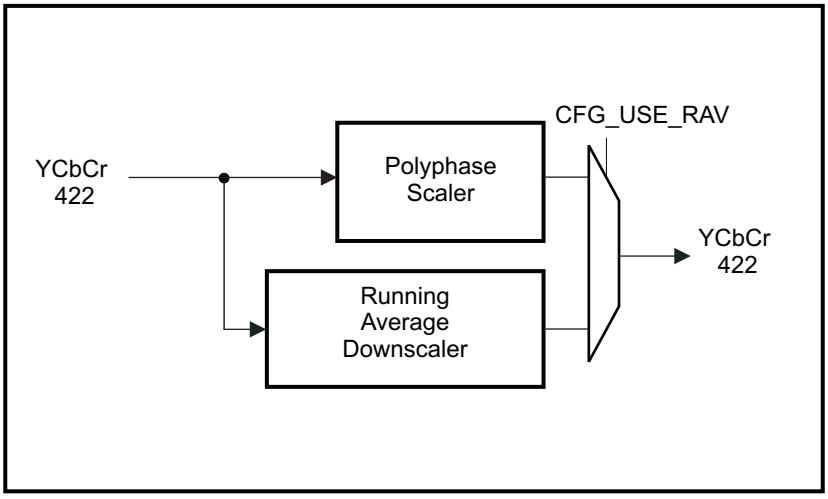
Figure 7-77. Peaking Filter at fs/4



7.4.7.2.3 Vertical Scaler

The vertical scaler has a polyphase (32-phase × 5-tap) filter and a running average filter as shown in Figure 7-78. While the polyphase filter can be used for any up-scaling and preferably downscaling to 3/16 scale factor, the running average filter is used only for downscaling to a ½ or less size. Selection between these two scalers is based on the user setting of VIP_CFG_SC0[4] CFG_USE_RAV parameter (CFG_USE_RAV= '0' for poliphase filter, and CFG_USE_RAV= '1' for running average filter), according to the user preference of the tradeoff between sharpness preserving and introduced artifacts.

Figure 7-78. Vertical Scaler Block Diagram



7.4.7.2.3.1 Running Average Filter

When a poly-phase filter is used, usually it has to have many taps in order to achieve acceptable quality for very small downscaling ratio, which requires the use of many line buffers. In VIP, there is a weighted Running Average filter for downscaling when the scaling factor is small (for example, when the scaling ratio is less than 0.5). This highly optimized design requires only one line buffer for luma and one for chroma, which still achieving acceptable quality. The output of the running average filter is based on weighted average of pixels in the current and previous rows in vertical direction. Initializations of accumulators affect the weights.

7.4.7.2.3.2 Vertical Scaler Configuration Parameters

Table 7-22. Vertical Scaler Configuration Parameters

Parameter	Typical Value	Controls	Description
VIP_CFG_SC0[10] CFG_INTERLACE_I		Frame or Field	0 = progressive, 1 = interlace
VIP_CFG_SC0[0] CFG_INTERLACE_O			0 = progressive 1 = interlace
VIP_CFG_SC0[3] CFG_INVT_FID			Invert field ID input
VIP_CFG_SC0[4] CFG_USE_RAV		Scaler Mode	0 = use polyphase scaler 1 = use running average scaler
VIP_CFG_SC1[26:0] CFG_ROW_ACC_INC		Bilinear & Polyphase Scalers	For progressive in/progressive out: round(2^{16}*(srcH-1)/(tarH - 1)) For progressive_in/interlace_out: round(2^{16}*2*(srcH-1)/(2*tarH - 1)) For interlace_in/progressive_out: round(2^{16}*(2*srcH-1)/(2*(tarH - 1))) For interlace_in/interlace_out: round(2^{16}*(2*srcH - 1)/(2*tarH - 1)) For interlace in/out, srcH/tarH are number of field lines as specified in VIP_CFG_SC4/VIP_CFG_SC5 descriptions.
VIP_CFG_SC2[27:0] CFG_ROW_ACC_OFFSET	0		Initial row accumulator value for progressive frame and top field
VIP_CFG_SC3[27:0] CFG_ROW_ACC_OFFSET_B	0		Initial row accumulator value for bottom field
VIP_CFG_SC13[27:24] CFG_DELTA_CHROMA_THR	4	Bilinear Scaler	Range for chroma soft switch based on pixel differences (max limit = 8)

Table 7-22. Vertical Scaler Configuration Parameters (continued)

Parameter	Typical Value	Controls	Description
VIP_CFG_SC13[21:12] CFG_CHROMA_INTP_THR	64		Threshold used in chroma soft switch based on pixel differences
VIP_CFG_SC13[9:0] CFG_SC_FACTOR_RAV		Running Average Scaler	Scale factor = round(1024 × tarH/srcH)
VIP_CFG_SC6[9:0] CFG_ROW_ACC_INIT_RAV			Initial row accumulator value for progressive frame and top field
VIP_CFG_SC6[19:10] CFG_ROW_ACC_INIT_RAV_B			Initial row accumulator value for bottom field

NOTE: Bi-linear scaler is not present in this device

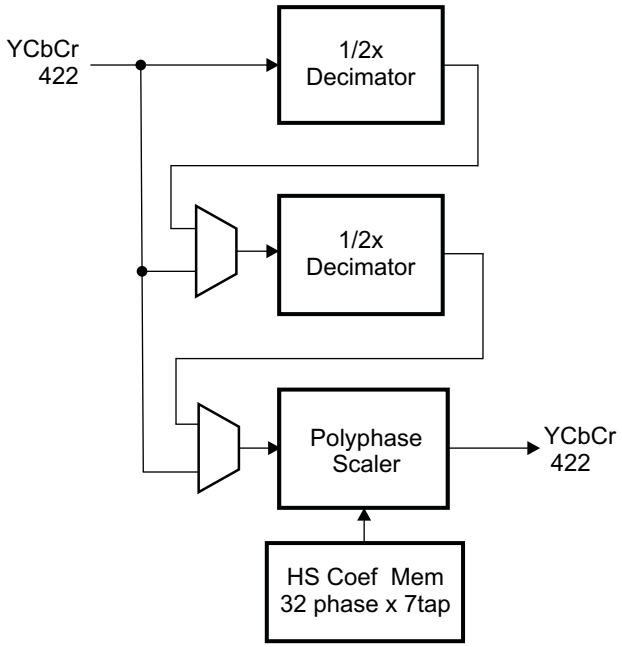
7.4.7.2.4 Horizontal Scaler

The Horizontal scaler is implemented using a 32-phase × 7-tap polyphase filter preceded by two sets of 1/2x decimators. The general configuration of the Horizontal scaler is performed as follows:

- For up scaling, the input video is interpolated using the polyphase scaler.
- For downscaling, it is recommended that input video is decimated by 2 until the modified scale factor falls between 1/2 and 1. Then, a polyphase filter is configured with coefficients selected based on the mod_scale_factor calculated as shown below from one of nine different sets of coefficients: (8,9,10,11,12,13,14,15,16)/16.

```
if  (scale_factor>=1/2)  {
   mux=0;  mod_scale_factor=scale factor;
} else if  (scale_factor>=1/4) {
   mux=1; mod_scale_factor=2*scale_factor;
} else {
   mux=2; mod_scale_factor=4*scale_factor;
}
```

Figure 7-79. Horizontal Scaler Block Diagram



In auto mode (CFG_AUTO_HS == 1), scaler will operate as per above recommendation. In addition to this, for (CFG_AUTO_HS==1), polyphase filtering will be bypassed when (scale_factor == 1) or (scale_factor == ½) or (scale_factor == ¼). If CFG_AUTO_HS==0) is used, user must provide proper values for dcm_2x, dcm_4x, proper values for all inputs to polyphase filter in the registers as well as appropriate coefficient sets. There is no constraint on polyphase filtering in terms of scaling ratio. However, there are constraints on input and output image width for scaler. Frame dimensions are limited from 64x64 to 2047x2047.

7.4.7.2.4.1 Half Decimation Filter

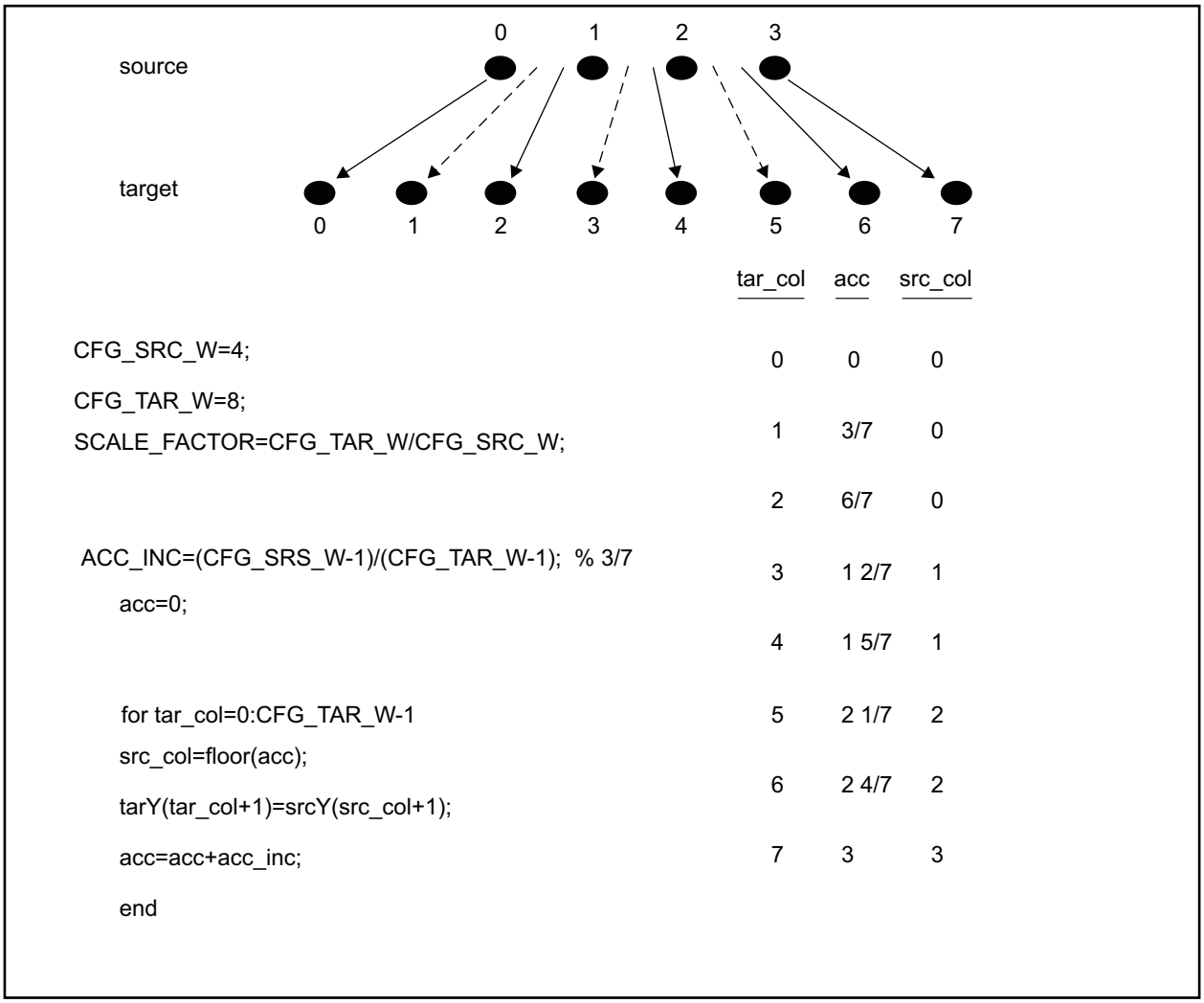
The half-decimation filter is an 11-tap filter with following coefficients: (14, 0, -29, 0, 79, 128, 79, 0, -29, 0, 14). For processing left and right edge pixels, the first and last data are repeated to pre-fill and extend the filter data pipeline respectively. These coefficients are hard-coded into scaler design and user cannot modify these.

7.4.7.2.4.2 Polyphase Filter

The horizontal scaling is done by stepping across the target row and interpolating target pixels based on source pixels. Accumulator points to the source pixel that corresponds to the target pixel.

Figure 7-80 shows an up-scaling example.

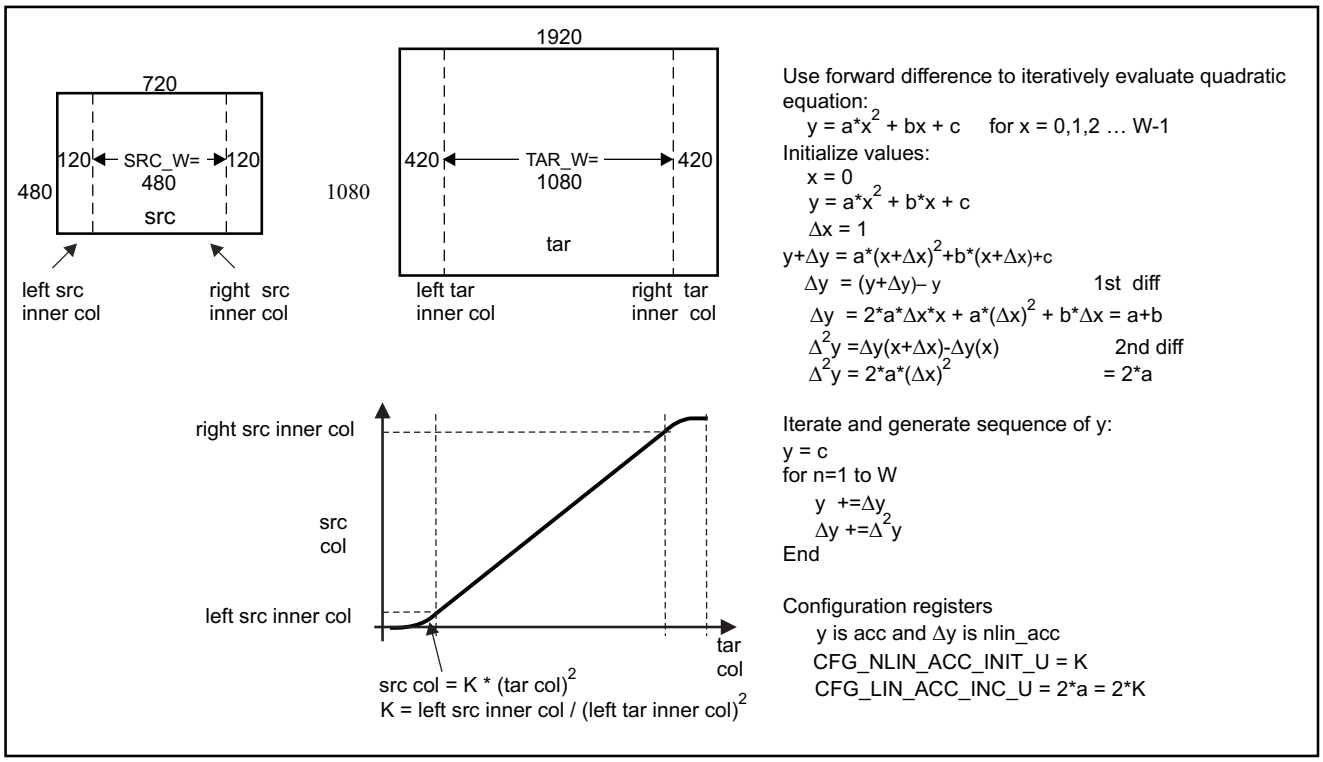
Figure 7-80. Polyphase Filtering Example



7.4.7.2.4.3 Nonlinear Horizontal Scaling

The horizontal scaler supports non-linear scaling to maintain the same aspect ratio for inner picture while stretching picture edges to fill the required resolution when capturing a 4 × 3 picture and fetching it as a 16 × 9 to memory. Non-linear scaling parameters are set with VIP_CFG_SC4[30:28] CFG_NLIN_ACC_INIT_U and VIP_CFG_SC4[26:24] CFG_LIN_ACC_INC_U registers. When setting up a non-linear scaling application, the inner picture is defined as the center square portion of the image with width and height equal to the height of the source image. Remaining side regions are then scaled non-linearly. Figure 7-81 shows a non-linear scaling case.

Figure 7-81. Non-linear Scaling Example



7.4.7.2.4.4 Horizontal Scaler Configuration Registers

Table 7-23. Register Group 1

Parameter	Controls	Description			
VIP_CFG_SC4[10:0] CFG_TAR_H	Image Dimension	Source Width			
VIP_CFG_SC4[22:12] CFG_TAR_W		Target Width			
VIP_CFG_SC0[1] CFG_LINEAR	Scaler Mode	If (linear == 1) SRC_Wi = SRC_W and TAR_Wi = TAR_W Else SRC_W= SRC_H and TAR_W = TAR_H			
VIP_CFG_SC0[2] CFG_SC_BYPASS		0 = enable scaler, 1 = bypass scaler			
VIP_CFG_SC0[6] CFG_AUTO_HS		CFG_AUTO_HS	CFG_DCM_2X	CFG_DCM_4X	Definition
		0	0	0	Polyphase scaling
VIP_CFG_SC0[7] CFG_DCM_2X		0	0	1	Horizontal decimation by 4 and polyphase scaling
		0	1	0	Horizontal decimation by 2 and polyphase scaling
VIP_CFG_SC0[8] CFG_DCM_4X		1	-	-	Automatic (selection of decimation filter is automatic)

Table 7-24. Register Group 2

Scale Factor	Decimation Usage	Control Register Bit
< 1/4	Decimation by 4	VIP_CFG_SC0[8] CFG_DCM_4X (set to 1 to enable decimation; disabled by default)
== 1/4	Decimation by 4	
1/4 < and < 1/2	Decimation by 2	VIP_CFG_SC0[7] CFG_DCM_2X (set to 1 to enable decimation; disabled by default)
== 1/2	Decimation by 2	
1/2 < and < 1	Bypassed	VIP_CFG_SC0[7] CFG_DCM_2X and CFG_DCM_4X (set to 0 to disable decimation; default value)
1	Bypassed	
> 1	Bypassed	

Table 7-25. Register Group 3

Parameter	Controls	Description
VIP_CFG_SC9[26:24] CFG_LIN_ACC_INC	Polyphase Scaler	if upscaling then CFG_LIN_ACC_INC = round(2^{24}*(srcWi-1)/(tarWi-1)) elseif downscaling CFG_LIN_ACC_INC= round(2^{24}*(srcWi/n-1)/(tarWi-1)) where n=2 or 4
VIP_CFG_SC8[10:0] CFG_NLIN_LEFT		if linear==1 CFG_NLIN_LEFT = 0 else CFG_NLIN_LEFT = (tarW - tarWi)/2
VIP_CFG_SC8[22:12] CFG_NLIN_RIGHT		if linear==1 CFG_NLIN_RIGHT = tarW-1 else CFG_NLIN_RIGHT = Ltar + tarWi – 1

VIP_CFG_SC5[26:24] CFG_NLIN_ACC_INC_U		if tarW/srcW >= 1 then
		d = 0
		if Ltar!=0 K = round[2^{24}*Lsrc/(Ltar*Ltar)] where Lsrc = (srcW-srcWi)/2
		else K = 0
		else
		d = (tarW-1)/2
		if Ltar!=0 K = round[2^{24} * Lsrc / (Ltar*(Ltar-2d))]
		where Lsrc= (srcW-srcWi)/(2n) and n=1,2 or 4
		else K = 0
		CFG_LIN_ACC_INC = 2*K (negative for downscaling)

VIP_CFG_SC4[30:28] CFG_NLIN_ACC_INIT_U		CFG_LIN_ACC_INC = K*(1-2*d)

NOTE: Source width and height variables for the polyphase filter are internally set with trimmer and decimation filter adjusted values.

7.4.7.2.5 Basic Configurations

Table 7-26 shows how the scaler should be configured based on the scale factor and the input/output mode.

Table 7-26. Scaler Configuration

Interlace		Mode[1]	VIP_CFG_SC5[10:0] CFG_SRC_H mod_srcH	VIP_CFG_SC4[10:0] CFG_TAR_H mod_tarH	Scale Factor
In	Out				
0	0	p->p	CFG_SRC_H	CFG_TAR_H	CFG_TAR_H/CFG_SRC_H
0	1	p->i	CFG_SRC_H	CFG_TAR_H/2	CFG_TAR_H/CFG_SRC_H
1	0	i->p	CFG_SRC_H/2	CFG_TAR_H	CFG_TAR_H/(CFG_SRC_H/2)
1	1	i->i	CFG_SRC_H/2	CFG_TAR_H/2	(CFG_TAR_H/2)/(CFG_SRC_H/2)

(1) p = progressive; i = interlaced

Table 7-27 shows how the vertical scaler should be configured based on the scale factor and the input/output mode.

Table 7-27. Vertical Scaler Configuration

Interlace		Mode[1]	VIP_CFG_SC9[26:24] CFG_ROW_ACC_INC/216	VIP_CFG_SC6[9:0] CFG_ROW_ACC_INIT_RAV/216	VIP_CFG_SC6[19:10] CFG_ROW_ACC_INIT_RAV_B/216
In	Out			Top	Bot
0	0	p->p	(CFG_SRC_H-1)/(CFG_TAR_H-1)	0	0
0	1	p->i	2*(CFG_SRC_H-1)/(CFG_TAR_H-1)	0	(CFG_SRC_H-1)/(CFG_TAR_H-1)
1	0	i->p	1/2*(CFG_SRC_H-1)/(CFG_TAR_H-1)	0	-0.5

(1) p = progressive; i = interlaced

Table 7-27. Vertical Scaler Configuration (continued)

1	1	i->i	(CFG_SRC_H-1)/(CFG_TAR_H-1)	0	[(CFG_SRC_H-1)/(CFG_TAR_H-1)-1]/2

7.4.7.2.6 Coefficient Memory

7.4.7.2.6.1 Overview

The scaler requires initialization of eight coefficient SRAMS prior to processing a video frame. These coefficients are stored in the external DDR memories and downloaded by the VPDMA. The VPDMA writes the coefficient data through the VPI Control Interface bus.

The eight coefficient SRAMs are:

- HS horizontal polyphase scaler, Luma and Chroma (7tap)
- VS vertical polyphase scaler, Luma and Chroma (5tap or 3tap)

7.4.7.2.6.2 Physical Coefficient SRAM Layout

Each of the six legacy coefficient SRAMs is 32 phases × 91 bits. A single coefficient value is 13 bits. Therefore, one word of the SRAM contains 7 coefficient values as shown in Figure 7-82, and 224 coefficient values are stored in each SRAM.

Figure 7-82. SRAM Layout for 7tap Coefficient

Phase 0	C6	C5	C4	C3	C2	C1	C0
Phase 31	C223	C222	C221	C220	C219	C218	C217

The two vertical polyphase SRAMs are 32 phases × 65 bits. A single coefficient is 13 bits. Therefore, one word of SRAM contains 5 coefficient values as shown in Figure 7-83.

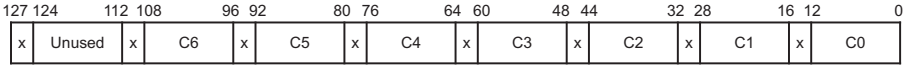
Figure 7-83. SRAM Layout for 5tap Coefficient

Phase 0	C4	C3	C2	C1	C0
Phase 31	C221	C220	C219	C218	C217

7.4.7.2.6.3 Scaler Coefficients Packing on 128-bit VPI Control I/F

A coefficient value is 13 bits wide and takes a single half word. Thus, one VPI Control write carries one phase's worth of coefficients. The last half-word in a quad-word is not used for 7tap coefficients.

Figure 7-84. VPI Control I/F Coef Data Format (7tap)



The Vertical Polyphase scaler coefficients have only five or three values per phase, so the last three (or five) entries are unused. The Vertical Polyphase coefficient packing is shown in Figure 7-85 and Figure 7-86.

Figure 7-85. VPI Control I/F Coef Data Format (5tap)

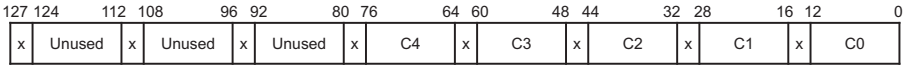
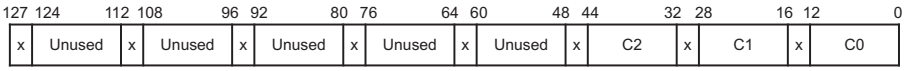


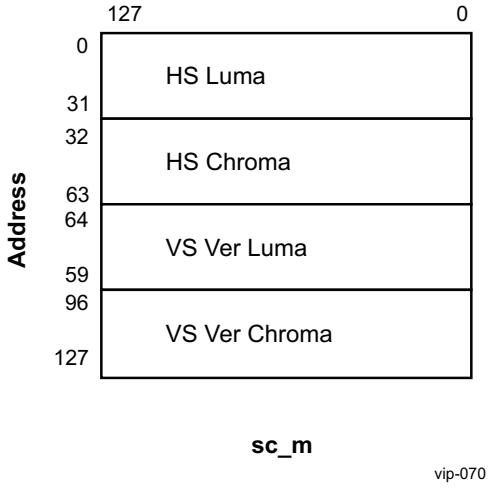
Figure 7-86. VPI Control I/F Coef Data Format (3tap)



7.4.7.2.6.4 VPI Control I/F Memory Map for Scaler Coefficients

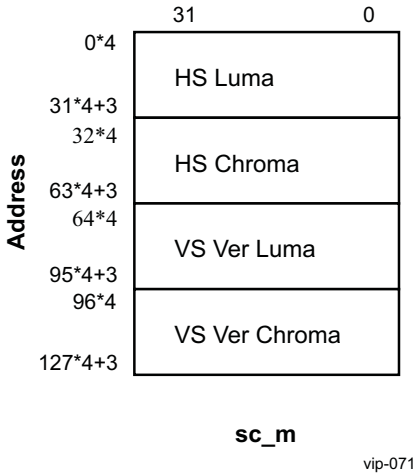
The memory map of the VPI Control I/F for the Scaler coefficients is shown in Figure 7-87. All coefficients can be filled up by a single VPI Control write command. The update address feature of VPI Control I/F enables individual access to a certain location of memories.

Figure 7-87. VPI Control I/F Memory Map (Write)



The module supports the VPI Control Read to read back the contents in the coefficient memories for debug purpose. Figure 7-88 shows the memory map of the VPI Control Read. As the VPI Control Read has only 32 bit data bus, it requires the word addressing while the write does the quad-word addressing.

Figure 7-88. VPI Control I/F Memory Map (Read)



7.4.7.2.6.5 VPI Control Interface

VPDMA is used to configure the coefficient memories of scaler through VPI control interface. Since the coefficient memories are not shadowed (unlike the memory mapped registers), VPI control write access needs to be done only during the gap between video frame processing times. If a write request is made while the Scaler is active, the access will be held off by the hardware until the last data of the currently processed frame is sent out. Care must be given to the order of the DMA descriptors so that blocking of VPI control bus does not occur.

7.4.7.2.6.6 Coefficient Table Selection Guide

The scaler filter coefficient tables are pre-generated using a MATLAB® program for various scaling factor ranges. Table 7-28 provides a general selection guide table for coefficient data files.

The mentioned .dat files are available in Section 7.4.7.4.

Table 7-28. Coefficient Data Files

Scaler	Scale Factor	Coeff table
HS Polyphase Filter	All upscaling	Section 7.4.7.4.1.1, *ppfcoef_scael_eq_1_32_phases_flip.dat*
	½ or ¼ down scaling	Section 7.4.7.4.1.1, ppfcoef_scale_eq_1_32_phases_flip.dat
	> 15/16	Section 7.4.7.4.1.9, *ppfcoef_scale_eq_15div16_32_phases_flip.dat*
	> 14/16	Section 7.4.7.4.1.8, *ppfcoef_scale_eq_14div16_32_phases_flip.dat*
	> 13/16	Section 7.4.7.4.1.7, *ppfcoef_scale_eq_13div16_32_phases_flip.dat*
	> 12/16	Section 7.4.7.4.1.6, *ppfcoef_scale_eq_12div16_32_phases_flip.dat*
	> 11/16	Section 7.4.7.4.1.5, *ppfcoef_scale_eq_11div16_32_phases_flip.dat*
	> 10/16	Section 7.4.7.4.1.4, *ppfcoef_scale_eq_10div16_32_phases_flip.dat*
	> 9/16	Section 7.4.7.4.1.3, *ppfcoef_scale_eq_9div16_32_phases_flip.dat*
	> 8/16	Section 7.4.7.4.1.2, *ppfcoef_scale_eq_8div16_32_phases_flip.dat*[(1)]

(1) HS Scaler has two sets of ½ decimator to perform downscaling ratios below ½ and ¼.

Table 7-28. Coefficient Data Files (continued)

Scaler	Scale Factor	Coeff table
VS Polyphase Filter	Upscaling	Section 7.4.7.4.2.1, *ppfcoef_scale_eq_1_32_phases_ver_5tap_flip.dat*
	> 15/16	Section 7.4.7.4.2.6.8, *ppfcoef_scale_eq_15div16_32_phases_ver_5tap_flip.dat*
	> 14/16	Section 7.4.7.4.2.6.7, *ppfcoef_scale_eq_14div16_32_phases_ver_5tap_flip.dat*
	> 13/16	Section 7.4.7.4.2.6.6, *ppfcoef_scale_eq_13div16_32_phases_ver_5tap_flip.dat*
	> 12/16	Section 7.4.7.4.2.6.5, *ppfcoef_scale_eq_12div16_32_phases_ver_5tap_flip.dat*
	> 11/16	Section 7.4.7.4.2.6.4, *ppfcoef_scale_eq_11div16_32_phases_ver_5tap_flip.dat*
	> 10/16	Section 7.4.7.4.2.6.3, *ppfcoef_scale_eq_10div16_32_phases_ver_5tap_flip.dat*
	> 9/16	Section 7.4.7.4.2.6.2, *ppfcoef_scale_eq_9div16_32_phases_ver_5tap_flip.dat*
	> 8/16	Section 7.4.7.4.2.6.1, *ppfcoef_scale_eq_8div16_32_phases_ver_5tap_flip.dat*
	else	For down-scaling <8/16, RAV filter is recommended. In this case, coeffcients for vertical scaling need to be loaded.

7.4.7.3 SC Code

7.4.7.3.1 Generate Coefficient Memory Image

The following Perl script is used to generate a coefficient memory image for a given set of scaling factors.

```
#!/usr/local/bin/perl

#$dir="coef/";                    # directory which contains the coef files
#$cfg_file="sc_config1.cfg";      # configuration file name
#$spl_file="sc_config_supl.cfg";  # supplemental configuration file name
$cfg_file=$ARGV[0];               # configuration file name
$spl_file=$ARGV[1];               # supplemental configuration file name
$dir=$ARGV[2];                    # directory which contains the coef files

$coef_width=13;  # coef bit width
$coef_ntap=7;    # coef tap
$coef_nphase=32; # coef phase
$coef_norm=11;   # coef norm

#-----------------------------------------------------------
# read config file to get srcH/tarH/interlace_i/interlace_o
#-----------------------------------------------------------
open(INFILE, "<$cfg_file") or die "### ERROR: Cannot open $cfg_file";
while(<INFILE>){
  if (m/([-0-9]+) +\/\/ +srcW/) {
    $srcW = $1;
  } elsif (m/([-0-9]+) +\/\/ +srcH/) {
    $srcH = $1;
  } elsif (m/([-0-9]+) +\/\/ +tarW/) {
    $tarW = $1;
  } elsif (m/([-0-9]+) +\/\/ +tarH/) {
    $tarH = $1;
  } elsif (m/([-0-9]+) +\/\/ +interlace_in/) {
    $interlace_i = $1;
  } elsif (m/([-0-9]+) +\/\/ +interlace_out/) {
    $interlace_o = $1;
  }
```

```
}
close(INFILE);

#-----------------------------------------------------------
# read supplemental config file to get srcWi/tarWi from
#-----------------------------------------------------------
open(INFILE,"<$spl_file") or die "### ERROR: Cannot open $spl_file";
while(<INFILE>){
  if (m/([-0-9]+) +\/\/ +srcWi/) {
    $srcWi = $1;
  } elsif (m/([-0-9]+) +\/\/ +tarWi/) {
    $tarWi = $1;
  } elsif (m/([-0-9]+) +\/\/ +profile/) {
    $profile = $1; # 0:HIGH,1:MEDIUM,2:LOW
  }
}
close(INFILE);
#-----------------------------------------------------------
# determine coef file based on the width/height
#-----------------------------------------------------------
#VS
#$vsc_file0 = "mod_ppfcoef_scale_eq_1_32_phases_flip.dat";
$vsc_file0  = "ppfcoef_scale_eq_1_32_phases_flip_PPF3_peak5_gain_eq_1_25.dat";
#VS VER
$mod_tarH = ($interlace_i == 0 && $interlace_o == 1)   $tarH<<1 : $tarH; if ($profile==2) {
  # LOW profile
  if ($mod_tarH >= $srcH) {
    $vsc_ver_file0 = "ppfcoef_scale_eq_1_32_phases_ver_3tap_flip.dat";
} else {
    if ($mod_tarH >=($srcH>>1)) {
      $n = int(16.0*$mod_tarH/$srcH);
      $vsc_ver_file0 = sprintf("ppfcoef_scale_eq_%ddiv16_32_phases_ver_3tap_flip.dat",$n);
    } else {
      $n = 0;
      $vsc_ver_file0 = "ppfcoef_scale_eq_1_32_phases_ver_3tap_flip.dat";
    }
  }
} else {
  if ($mod_tarH >= $srcH) {
    $vsc_ver_file0 = "ppfcoef_scale_eq_1_32_phases_ver_5tap_flip.dat";
  } else {
      $n = int(16.0*$mod_tarH/$srcH);
      $vsc_ver_file0 = sprintf("ppfcoef_scale_eq_%ddiv16_32_phases_ver_5tap_flip.dat",$n);
  }
}

# HS
if ($tarWi >= $srcWi) {
  $hsc_file0 = "ppfcoef_scale_eq_1_32_phases_flip.dat";
} elsif ( ($tarWi == ($srcWi>>1)) || ($tarWi == ($srcWi>>2)) ) {
  $hsc_file0 = "ppfcoef_scale_eq_1_32_phases_flip.dat";
} else {
if ($tarWi > ($srcWi>>1)) {
    $n = int(16.0*$tarWi/$srcWi);
  } elsif ($tarWi > ($srcWi>>2)) {
    $n = int(16.0*$tarWi/($srcWi>>1));
  } elsif ($tarWi >=($srcWi>>3)) {
    $n = int(16.0*$tarWi/($srcWi>>2));
  } else {
    $n = 0;
  }
  $hsc_file0 = sprintf("ppfcoef_scale_eq_%ddiv16_32_phases_flip.dat",$n);
}
#-----------------------------------------------------------
# write out the coef hex file
#-----------------------------------------------------------
```

```
&write_coef($hsc_file0);
&write_coef($hsc_file0);
&write_coef($vsc_ver_file0);
&write_coef($vsc_ver_file0);
&write_coef($vsc_file0);
&write_coef($vsc_file0);
sub write_coef {

  my ($filename) = @_;

  open(INFILE, "<$dir/$filename") or die "### ERROR: Cannot open $dir/$filename";

  $line=<INFILE>;
@val=split(' ',$line);
  $ntap=$val[0];
  $nphase=$val[1];
  $norm=$val[2];
for ($p=0;$p<$nphase;$p++) {
    $line=<INFILE>;@val=split(' ',$line);
    for($i=0;$i<$ntap;$i++) {
      if ($val[$i]<0) {
        $val[$i]+=(1<<$coef_width);
      }
    }
    undef(@coef);
    unshift(@coef, sprintf("%04x",$val[0]));
    unshift(@coef, sprintf("%04x",$val[1]));
    unshift(@coef, sprintf("%04x",$val[2]));
    unshift(@coef, sprintf("%04x",$val[3]));
    unshift(@coef, sprintf("%04x",$val[4]));
    unshift(@coef, sprintf("%04x",$val[5]));
    unshift(@coef, sprintf("%04x",$val[6]));
    unshift(@coef, sprintf("%04x",0));
    $coef=join("",@coef);
    print "$coef\n";
  }

  close(INFILE);
}
```

7.4.7.3.2 Scaler Configuration Calculation

The following C-code shows how configuration parameters are calculated:

```
// ======================================
// Required Input Parameter
// ======================================
//   srcW, srcH, tarW, tarH, srcWi, tarWi
//   input/output scan modes
//   Note:  srcH and tarH refer to number of lines in the frame even for interlace in/out
//          scaling.  Based on scaling scan mode input/output scan mode option,
//          heights are adjusted during internal calculations see mod_srcH and mod_tarH.


pixel_scale_factor=4;  // 10 bit pixel
hor_pixel_offset =0.0

  // ================================================================================================
  // Peaking Filter Configuration
  // ================================================================================================
  // ----------------------------------------------------------------
  // HPF Coef
  // ----------------------------------------------------------------
    y_peak_enable     =   0;

    peak_select=0; // 0=peak at fs/4  1=NTSC  2=PAL
```

```
        switch(peak_select) {
        case 0: {// peak at fs/4 and gain = 1
                 HPF_coef0         =   0;
                 HPF_coef1         =   0;
                 HPF_coef2         =   0;
                 HPF_coef3         =  -4;
                 HPF_coef4         =   0;
                 HPF_coef5         =   8; // mid tap
                 HPF_norm_shift    =   4;
             break;
    }
        case 1: {// NTSC: peak at 0.133*fs and gain=1
                 HPF_coef0         =  -2;
                 HPF_coef1         =  -8;
                 HPF_coef2         =  -8;
                 HPF_coef3         =  -2;
                 HPF_coef4         =  12;
                 HPF_coef5         =  16; // mid tap
                 HPF_norm_shift    =   6;
                 break;
        }
        case 2: {// PAL: peak at 0.163*fs and gain=1
                 HPF_coef0         =   2;
                 HPF_coef1         =  -4;
                 HPF_coef2         = -11;
                 HPF_coef3         =  -7;
                 HPF_coef4         =   9;
                 HPF_coef5         =  22; // mid tap
                 HPF_norm_shift    =   6;
                 break;
        }
        }

      // ---------------------------------------------------------------
      // NonLinear Coring Function typical values
      // ---------------------------------------------------------------
        NL_coring_thr     =  16;
        NL_limit          = 200;
        NL_lo_slope       =  16;
        NL_hi_thr         = 400;
        NL_hi_slope_shift =   4;



      // ================================================================================================
      // Edge Detection Configuration
      // ================================================================================================
        // edge detection
        confidence_default = 0; // 0  =use 5 tap polyphase filter for SC with ev_enable =0

        min_Gy_thr         =  64; // 64
        min_Gy_thr_range   =   3; // 3 power of 2
        gradient_thr       = 200; // 200
    gradient_thr_range =   6; // 6 power of 2

        ev_thr = int(4.0*3.111+0.5); // edge vector soft switch threshold (3.2)


      // ================================================================================================
      // vertical scaler configuration
      // ================================================================================================
      // ---------------------------------------------------------------
      // vertical scaler typical parameters
      // ---------------------------------------------------------------
        invert_field_ID           =  0; // invert field ID input
        delta_ev_thr              =  1; // edge vector soft switch range
```

```
    ver_pixel_offset          =  0.0;
    uv_intp_thr               =  pixel_scale_factor*16;
    delta_y_thr               =  4; // luma soft switch range
    delta_uv_thr              =  4; // chroma soft switch range
  //
  //
  // ------------------------------------------------------------------
  // Vertical Scaler Mode Determination
  // ------------------------------------------------------------------
  //
  //  interlace
  //   in   out   mode mod_srcH mod_tarH         scale
  // ----- ------ ---- -------- --------   -----------------
  //   0      0   p->p   srcH     tarH          tarH/srcH
  //   0      1   p->i   srcH     tarH>>1       tarH/srcH
  //   1      0   i->p   srcH>>1  tarH          tarH/(srcH/2)
  //   1      1   i->i   srcH>>1  tarH>>1    (tarH/2)/(srcH/2)

  if (interlace_in)  mod_srcH=srcH>>1; // interlace
  else               mod_srcH=srcH;    // progressive
  if (interlace_out) mod_tarH=tarH>>1; // interlace
  else               mod_tarH=tarH;    // progressive

  // determine vertical scaler
  if ((interlace_in==0)&&(interlace_out==1)) {
    if (tarH>((1+srcH)>>1)) use_rav = 0;   // 1=use RAV scaler 0=use polyphase scaler
    else                    use_rav = 1;
  } else {
    if (mod_tarH>((1+mod_srcH)>>1)) use_rav = 0; // 1=use RAV scaler  0=use polyphase scaler
    else                            use_rav = 1;
}



  // ------------------------------------------------------------------
  // RAV or Polyphase parameters
  // ------------------------------------------------------------------
  if (use_rav) {  // downscale
    // ------------
    // --- RAV ----
    // ------------
    if (use_internal_defaults) enable_edge_detection =   0;

    if ((interlace_in==0)&&(interlace_out==1)) scale = double(tarH)/double(srcH);
    else                                      scale = double(mod_tarH)/double(mod_srcH);
    sc_factor_rav = int(1024.0*scale+0.5);
// Peter's method
    delta     = (1.0/scale-1.0)/2.0;
    int_part  = floor(delta);
    frac_part = delta-int_part;

    row_acc_init_rav   = int(1024*(scale+(1.0-
scale)/2.0)+0.5);                                    // top field
    row_acc_init_b_rav = int(1024*(scale+(1.0-2.0*frac_part)*(1.0-
(1.0+2.0*int_part)*scale)/2.0)+0.5); // bottom field

    row_acc_inc        = 0; // polyphase scaler
    row_acc_offset     = 0; // polyphase scaler
    row_acc_offset_b   = 0; // polyphase scaler

  } else { // upscale using polyphase scaler
    // ------------
    // --- PPF ----
    // ------------
    if (use_internal_defaults) enable_edge_detection =   1;
```

```
    sc_factor_rav      = 0;
    delta_rav          = 0;
    row_acc_init_rav   = 0;
    row_acc_init_b_rav = 0;

    // upscaler
    // interlace                              row acc init value
        //  in   out mode      row acc inc      top          bottom
    // ---- ---- ---- -------------------- ---  ---------------------
    //   0    0  p->p     (srcH-1)/(tarH-1)  0             0
    //   0    1  p->i   2*(srcH-1)/(tarH-1)  0    (srcH-1)/(tarH-1)
    //   1    0  i->p 1/2*(srcH-1)/(tarH-1)  0           -0.5
    //   1    1  i->i     (srcH-1)/(tarH-1)  0   [(srcH-1)/(tarH-1)-1]/2

    row_acc_offset   = int(65536.0*ver_pixel_offset +0.5);  // progressive or top field
    if (interlace_in) {
if (interlace_out) {
         row_acc_inc      =  int(65536.0*double(srcH-1)/double(tarH-1)+0.5);
        row_acc_offset_b =  (int(65536.0/2.0*(double(srcH-1)/(double(tarH-1))-
                              1.0)+0.5))+row_acc_offset;
       } else { // progressive out
          row_acc_inc      = int(65536.0*double(srcH-1)/(2.0*double(tarH-1))+0.5);
            if ((-0.5+row_acc_offset)<0.0) round_factor=-0.5;
            else                           round_factor= 0.5;
            row_acc_offset_b = int(65536.0*(-0.5)+round_factor)+row_acc_offset;
       }
    } else {   // progressive in
       if (interlace_out) {
        row_acc_inc      = int(65536.0*2.0*double(srcH-1)/double(tarH-1)+0.5);
        row_acc_offset_b =  int(65536.0*double(srcH-1)/double(tarH-1)+0.5)+row_acc_offset;
       } else { // progressive out
          row_acc_inc      =  int(65536.0*double(srcH-1)/double(tarH-1)+0.5);
           row_acc_offset_b = row_acc_offset;
       }
    }
  }
// ===============================================================================================
  // Horizontal Scaler configuration
  // ===============================================================================================
  // ---------------------------------------------------------------
  // horizontal scaler mode determination
  // ---------------------------------------------------------------
  auto_hs                = 1;
  dcm_2x                 = 0;
  dcm_4x                 = 0;
  hp_bypass              = 0;
  if (srcWi==srcW) linear = 1;
  else             linear = 0;

  // hor scaler parameters
  if (tarW>srcW) {              // upscale
    mod_srcW  = srcW;
    mod_srcWi = srcWi;
} else if (tarW<=(srcW>>2)) {  // downscale by <=1/4
    mod_srcW  = srcW>>2;
    mod_srcWi = srcWi>>2;
  } else if (tarW<=(srcW>>1)) {  // downscale by <=1/2
    mod_srcW  = srcW>>1;
    mod_srcWi = srcWi>>1;
  } else {                      // downscale by <=1
    mod_srcW  = srcW;
    mod_srcWi = srcWi;
  }

  // Not used any more:
  // hs_factor       = int(16.0*double(tarWi)/double(mod_srcWi)+0.5);  // hor scale factor (6.4)
```

```
  // ---------------------------------------------------------------
  // Horizontal PolyPhase Settings  --
  // ---------------------------------------------------------------
  lin_acc_inc     = int(16777216.0*double(mod_srcWi-1)/double(tarWi-1)+0.5);
  col_acc_offset  = int(16777216.0*hor_pixel_offset +0.5);
  nlin_left       = (tarW-tarWi)>>1;
  nlin_right      = nlin_left+tarWi-1;
  if (linear) {
    nlin_acc_inc    = 0;
    nlin_acc_init   = 0;
  } else {
     // ---------------------------------
     // Non-linear scaling configuration
     // ---------------------------------
    nlin_left_src   = (mod_srcW-mod_srcWi)>>1;

        if (tarWi>=srcWi) {    // upscale
      d             =  0.0;
      round_factor  =  0.5;
        } else {                 // downscale
      d             = (double(tarW)-1.0)/2.0;
      round_factor  = -0.5;
    }

      K             = 16777216.0*double(nlin_left_src)/(double(nlin_left)*double(nlin_left-
2.0*d));
    nlin_acc_inc  = int(2.0*K+round_factor);
    nlin_acc_init = int(K*(1.0-2.0*d)+0.5);
  }
nlin_left_tar  = nlin_left;
  nlin_right_tar = nlin_right;

  // ================================================================================================
  // Bypass Determination
  // ================================================================================================
  // bypass
  if ((srcW==tarW)&&(srcWi==tarWi)&&(mod_srcH==mod_tarH)) sc_bypass = 1;
  else                                                    sc_bypass = 0;
  //

}
```

7.4.7.3.3 Typical Configuration Values

The following is the list of all scaler register fields that are set to constant values, representing typical settings:

VIP_CFG_SC0[3] CFG_INVT_FID = 0 (Field ID will be used without inversion)

VIP_CFG_SC0[5] CFG_ENABLE_EV = 1 (Field ID will be used without inversion)

VIP_CFG_SC0[6] CFG_AUTO_HS = 1 (The hardware will automatically decide, if current operation is up or down scaling. In down-scaling, it will also decide, if 2X or 4X decimation filter is needed)

VIP_CFG_SC0[7] CFG_DCM_2X = 0 (The 2X decimation filter is disabled)

VIP_CFG_SC0[8] CFG_DCM_4X = 0 (The 4X decimation filter is disabled)

VIP_CFG_SC0[11] CFG_ENABLE_SIN2_VER_INTP = 1 (Modified bilinear interpolation is used)

VIP_CFG_SC0[14] CFG_Y_PK_EN = 0 (Luma peaking is disabled)

VIP_CFG_SC0[15] CFG_TRIM= 1 (Trimming is enabled)

VIP_CFG_SC12[24:0] CFG_COL_ACC_OFFSET = 0 (No horizontal offset is involved)

VIP_CFG_SC13[21:12] CFG_CHROMA_INTP_THR = 64 (If the difference is less than this threshold, the interpolation of chroma should be done along edge direction. Otherwise, the interpolation of chroma should be done vertically)

VIP_CFG_SC13[27:24] CFG_DELTA_CHROMA_THR = 4 (max limit=8)

VIP_CFG_SC18[24:16] CFG_CONF_DEFAULT = 0x100 (Defines confidence factor when edge detection is disabled (VIP_CFG_SC0[5] CFG_ENABLE_EV bit = 0))
VIP_CFG_SC19 = 0xFC000000
VIP_CFG_SC20 = 0x0C840800
VIP_CFG_SC21 = 0x00100010
VIP_CFG_SC22 = 0x00040190

7.4.7.4 SC Coefficient Data Files

7.4.7.4.1 HS Polyphase Filter Coefficients

7.4.7.4.1.1 ppfcoef_scale_eq_1_32_phases_flip.dat

```
 7  32 11
31 -112  210 1790  210 -112   31
28  -98  159 1787  264 -126   34
25  -84  111 1779  320 -140   37
22  -71   65 1767  379 -154   40
19  -58   23 1750  439 -168   43
16  -45  -17 1728  502 -181   45
14  -33  -53 1701  565 -193   47
11  -22  -86 1670  631 -205   49
 9  -11 -116 1635  696 -216   51
 7   -1 -142 1594  763 -225   52
 5    8 -166 1551  830 -233   53
 3   16 -186 1504  898 -240   53
 2   23 -204 1455  965 -245   52
 1   30 -218 1401 1031 -248   51
 0   35 -230 1345 1097 -249   50
-1   40 -238 1286 1162 -248   47
44 -244 1224 1224 -244   44    0
47 -248 1162 1286 -238   40   -1
50 -249 1097 1345 -230   35    0
51 -248 1031 1401 -218   30    1
52 -245  965 1455 -204   23    2
53 -240  898 1504 -186   16    3
53 -233  830 1551 -166    8    5
52 -225  763 1594 -142   -1    7
51 -216  696 1635 -116  -11    9
49 -205  631 1670  -86  -22   11
47 -193  565 1701  -53  -33   14
45 -181  502 1728  -17  -45   16
43 -168  439 1750   23  -58   19
40 -154  379 1767   65  -71   22
37 -140  320 1779  111  -84   25
34 -126  264 1787  159  -98   28
```

7.4.7.4.1.2 ppfcoef_scale_eq_8div16_32_phases_flip.dat

```
 7  32 11
  -28   61  542  898  542   61  -28
  -27   52  523  899  560   70  -29
  -26   44  505  898  578   79  -30
  -25   37  487  895  595   89  -30
  -24   30  468  892  613  100  -31
  -22   23  450  887  630  111  -31
  -21   17  432  883  647  122  -32
  -20   11  414  877  664  134  -32
  -19    6  396  871  680  146  -32
  -18    1  378  864  695  159  -31
  -16   -4  360  856  711  172  -31
  -15   -8  343  847  726  185  -30
  -14  -12  325  838  740  200  -29
```

```
  -13  -15  308  828  754  214  -28
  -12  -18  292  816  768  229  -27
  -10  -21  275  805  780  244  -25
  -23  258  789  789  258  -23    0
  -25  244  780  805  275  -21  -10
  -27  229  768  816  292  -18  -12
  -28  214  754  828  308  -15  -13
  -29  200  740  838  325  -12  -14
  -30  185  726  847  343   -8  -15
  -31  172  711  856  360   -4  -16
  -31  159  695  864  378    1  -18
  -32  146  680  871  396    6  -19
  -32  134  664  877  414   11  -20
  -32  122  647  883  432   17  -21
  -31  111  630  887  450   23  -22
  -31  100  613  892  468   30  -24
  -30   89  595  895  487   37  -25
  -30   79  578  898  505   44  -26
  -29   70  560  899  523   52  -27
```

7.4.7.4.1.3 ppfcoef_scale_eq_9div16_32_phases_flip.dat

```
7  32 11
  -33    8  547 1004  547    8  -33
  -31    0  525 1003  570   16  -35
  -29   -7  503 1001  592   25  -37
  -27  -13  481  998  614   34  -39
  -26  -19  459  995  636   44  -41
  -24  -25  437  990  658   55  -43
  -22  -29  414  983  679   67  -44
  -20  -34  393  976  700   79  -46
  -18  -38  371  968  721   91  -47
  -17  -41  350  959  742  104  -49
  -15  -44  330  948  761  118  -50
  -13  -46  309  936  780  133  -51
  -12  -48  289  924  799  148  -52
  -11  -50  270  911  817  163  -52
   -9  -51  250  897  833  180  -52
   -8  -52  232  882  850  196  -52
  -52  213  863  863  213  -52    0
  -52  196  850  882  232  -52   -8
  -52  180  833  897  250  -51   -9
  -52  163  817  911  270  -50  -11
  -52  148  799  924  289  -48  -12
  -51  133  780  936  309  -46  -13
  -50  118  761  948  330  -44  -15
  -49  104  742  959  350  -41  -17
  -47   91  721  968  371  -38  -18
  -46   79  700  976  393  -34  -20
  -44   67  679  983  414  -29  -22
  -43   55  658  990  437  -25  -24
  -41   44  636  995  459  -19  -26
  -39   34  614  998  481  -13  -27
  -37   25  592 1001  503   -7  -29
  -35   16  570 1003  525    0  -31
```

7.4.7.4.1.4 ppfcoef_scale_eq_10div16_32_phases_flip.dat

```
7  32 11
  -30  -46  542 1116  542  -46  -30
  -28  -52  515 1115  570  -39  -33
  -25  -57  488 1113  597  -32  -36
  -23  -62  462 1109  624  -24  -38
  -20  -65  435 1104  650  -15  -41
  -18  -69  409 1097  678   -5  -44
  -16  -71  383 1089  704    6  -47
```

```
-14  -74  358 1081  730   17  -50
-12  -75  333 1070  756   29  -53
-11  -76  309 1058  782   42  -56
 -9  -77  285 1045  806   56  -58
 -8  -77  262 1030  831   71  -61
 -6  -77  239 1015  855   86  -64
 -5  -76  218  997  877  103  -66
 -4  -75  196  980  899  120  -68
 -3  -74  176  961  920  138  -70
-72  156  940  940  156  -72    0
-70  138  920  961  176  -74   -3
-68  120  899  980  196  -75   -4
-66  103  877  997  218  -76   -5
-64   86  855 1015  239  -77   -6
-61   71  831 1030  262  -77   -8
-58   56  806 1045  285  -77   -9
-56   42  782 1058  309  -76  -11
-53   29  756 1070  333  -75  -12
-50   17  730 1081  358  -74  -14
-47    6  704 1089  383  -71  -16
-44   -5  678 1097  409  -69  -18
-41  -15  650 1104  435  -65  -20
-38  -24  624 1109  462  -62  -23
-36  -32  597 1113  488  -57  -25
-33  -39  570 1115  515  -52  -28
```

7.4.7.4.1.5 ppfcoef_scale_eq_11div16_32_phases_flip.dat

```
7  32 11
-19  -94  522 1230  522  -94  -19
-17  -98  490 1230  555  -90  -22
-14 -100  458 1227  587  -85  -25
-12 -102  427 1223  620  -79  -29
-10 -103  397 1217  652  -73  -32
 -8 -104  367 1209  685  -65  -36
 -6 -104  337 1199  717  -56  -39
 -4 -103  309 1187  749  -47  -43
 -3 -102  281 1174  781  -36  -47
 -1 -100  253 1159  812  -24  -51
  0  -98  227 1142  843  -11  -55
  1  -96  201 1124  874    3  -59
  1  -93  177 1105  903   18  -63
  2  -90  153 1084  932   34  -67
  2  -87  131 1062  961   51  -72
  3  -83  109 1038  987   69  -75
-79   89 1014 1014   89  -79    0
-75   69  987 1038  109  -83    3
-72   51  961 1062  131  -87    2
-67   34  932 1084  153  -90    2
-63   18  903 1105  177  -93    1
-59    3  874 1124  201  -96    1
-55  -11  843 1142  227  -98    0
-51  -24  812 1159  253 -100   -1
-47  -36  781 1174  281 -102   -3
-43  -47  749 1187  309 -103   -4
-39  -56  717 1199  337 -104   -6
-36  -65  685 1209  367 -104   -8
-32  -73  652 1217  397 -103  -10
-29  -79  620 1223  427 -102  -12
-25  -85  587 1227  458 -100  -14
-22  -90  555 1230  490  -98  -17
```

7.4.7.4.1.6 ppfcoef_scale_eq_12div16_32_phases_flip.dat

```
7  32 11
   -3 -132  486 1346  486 -132   -3
```

```
  -1 -132  449 1345  524 -131   -6
   1 -131  413 1342  562 -130   -9
   3 -130  378 1336  600 -127  -12
   4 -128  343 1328  639 -123  -15
   5 -125  309 1319  677 -119  -18
   6 -122  277 1306  716 -113  -22
   7 -118  245 1292  754 -106  -26
   8 -114  214 1276  793  -98  -31
   8 -109  185 1257  831  -89  -35
   9 -105  156 1237  869  -78  -40
   9 -100  130 1214  906  -66  -45
   9  -94  104 1190  942  -53  -50
   9  -89   79 1165  978  -38  -56
   8  -83   56 1138 1012  -22  -61
   8  -78   35 1108 1046   -4  -67
 -72   15 1081 1081   15  -72    0
 -67   -4 1046 1108   35  -78    8
 -61  -22 1012 1138   56  -83    8
 -56  -38  978 1165   79  -89    9
 -50  -53  942 1190  104  -94    9
 -45  -66  906 1214  130 -100    9
 -40  -78  869 1237  156 -105    9
 -35  -89  831 1257  185 -109    8
 -31  -98  793 1276  214 -114    8
 -26 -106  754 1292  245 -118    7
 -22 -113  716 1306  277 -122    6
 -18 -119  677 1319  309 -125    5
 -15 -123  639 1328  343 -128    4
 -12 -127  600 1336  378 -130    3
  -9 -130  562 1342  413 -131    1
  -6 -131  524 1345  449 -132   -1
```

7.4.7.4.1.7 ppfcoef_scale_eq_13div16_32_phases_flip.dat

```
7  32 11
   14 -154  435 1458  435 -154   14
   15 -150  393 1458  477 -157   12
   16 -146  353 1454  521 -160   10
   16 -141  314 1447  565 -161    8
   17 -135  276 1436  609 -161    6
   17 -129  239 1425  654 -161    3
   17 -123  204 1410  699 -159    0
   16 -116  170 1393  745 -156   -4
   16 -109  137 1373  790 -151   -8
   16 -102  107 1350  835 -146  -12
   15  -94   77 1325  879 -138  -16
   14  -87   50 1298  924 -130  -21
   13  -80   24 1269  968 -119  -27
   12  -72    0 1238 1010 -107  -33
   11  -65  -22 1204 1053  -94  -39
   10  -58  -43 1169 1093  -78  -45
  -52  -62 1138 1138  -62  -52    0
  -45  -78 1093 1169  -43  -58   10
  -39  -94 1053 1204  -22  -65   11
  -33 -107 1010 1238    0  -72   12
  -27 -119  968 1269   24  -80   13
  -21 -130  924 1298   50  -87   14
  -16 -138  879 1325   77  -94   15
  -12 -146  835 1350  107 -102   16
   -8 -151  790 1373  137 -109   16
   -4 -156  745 1393  170 -116   16
    0 -159  699 1410  204 -123   17
    3 -161  654 1425  239 -129   17
    6 -161  609 1436  276 -135   17
    8 -161  565 1447  314 -141   16
   10 -160  521 1454  353 -146   16
```

```
   12 -157  477 1458  393 -150   15
```

7.4.7.4.1.8 ppfcoef_scale_eq_14div16_32_phases_flip.dat

```
 7  32 11
   27 -158  370 1570  370 -158   27
   27 -150  324 1568  417 -165   27
   26 -142  281 1563  465 -172   27
   25 -133  238 1555  515 -178   26
   24 -124  198 1543  565 -183   25
   23 -115  159 1527  616 -186   24
   22 -106  122 1510  667 -189   22
   21  -97   87 1489  719 -191   20
   19  -87   54 1464  772 -191   17
   18  -78   23 1437  824 -190   14
   16  -69   -6 1407  876 -187   11
   15  -60  -32 1373  927 -182    7
   13  -52  -57 1339  979 -176    2
   12  -44  -79 1300 1030 -168   -3
   11  -36  -99 1261 1079 -159   -9
    9  -28 -117 1218 1128 -147  -15
  -21 -134 1179 1179 -134  -21    0
  -15 -147 1128 1218 -117  -28    9
   -9 -159 1079 1261  -99  -36   11
   -3 -168 1030 1300  -79  -44   12
    2 -176  979 1339  -57  -52   13
    7 -182  927 1373  -32  -60   15
   11 -187  876 1407   -6  -69   16
   14 -190  824 1437   23  -78   18
   17 -191  772 1464   54  -87   19
   20 -191  719 1489   87  -97   21
   22 -189  667 1510  122 -106   22
   24 -186  616 1527  159 -115   23
   25 -183  565 1543  198 -124   24
   26 -178  515 1555  238 -133   25
   27 -172  465 1563  281 -142   26
   27 -165  417 1568  324 -150   27
```

7.4.7.4.1.9 ppfcoef_scale_eq_15div16_32_phases_flip.dat

```
 7  32 11
   33 -143  294 1680  294 -143   33
   31 -132  246 1678  345 -155   35
   30 -121  199 1671  398 -165   36
   27 -109  154 1661  452 -175   38
   25  -97  112 1647  508 -185   38
   23  -86   72 1629  564 -193   39
   21  -75   35 1607  622 -201   39
   19  -64    0 1580  681 -207   39
   17  -53  -32 1551  740 -213   38
   15  -43  -61 1518  799 -217   37
   13  -33  -88 1481  859 -219   35
   11  -24 -113 1442  919 -220   33
    9  -15 -134 1399  978 -219   30
    8   -7 -153 1354 1036 -217   27
    6    0 -170 1307 1094 -212   23
    5    7 -184 1257 1150 -205   18
   13 -196 1207 1207 -196   13    0
   18 -205 1150 1257 -184    7    5
   23 -212 1094 1307 -170    0    6
   27 -217 1036 1354 -153   -7    8
   30 -219  978 1399 -134  -15    9
   33 -220  919 1442 -113  -24   11
   35 -219  859 1481  -88  -33   13
   37 -217  799 1518  -61  -43   15
   38 -213  740 1551  -32  -53   17
```

```
 39 -207  681 1580    0  -64   19
 39 -201  622 1607   35  -75   21
 39 -193  564 1629   72  -86   23
 38 -185  508 1647  112  -97   25
 38 -175  452 1661  154 -109   27
 36 -165  398 1671  199 -121   30
 35 -155  345 1678  246 -132   31
```

7.4.7.4.2 VS Polyphase Filter Coefficients

7.4.7.4.2.1 ppfcoef_scale_eq_1_32_phases_ver_5tap_flip.dat

```
5  32 11
  -47  177 1788  177  -47
  -40  133 1785  225  -55
  -33   91 1778  276  -64
  -27   53 1765  330  -73
  -21   18 1747  386  -82
  -15  -13 1722  445  -91
  -11  -41 1693  507 -100
   -7  -66 1660  570 -109
   -3  -88 1622  635 -118
    0 -107 1579  703 -127
    2 -122 1532  771 -135
    4 -135 1482  839 -142
    5 -145 1428  909 -149
    6 -153 1371  978 -154
    7 -158 1310 1047 -158
    7 -161 1247 1116 -161
 -162 1186 1186 -162    0
 -161 1116 1247 -161    7
 -158 1047 1310 -158    7
 -154  978 1371 -153    6
 -149  909 1428 -145    5
 -142  839 1482 -135    4
 -135  771 1532 -122    2
 -127  703 1579 -107    0
 -118  635 1622  -88   -3
 -109  570 1660  -66   -7
 -100  507 1693  -41  -11
  -91  445 1722  -13  -15
  -82  386 1747   18  -21
  -73  330 1765   53  -27
  -64  276 1778   91  -33
  -55  225 1785  133  -40
```

7.4.7.4.2.2 ppfcoef_scale_eq_3_32_phases_flip.dat

```
5,   32,   11,
  130,  515,  758,  515,  130,
  121,  503,  757,  528,  139,
  113,  490,  756,  541,  148,
  105,  477,  755,  553,  158,
   97,  464,  753,  566,  168,
   90,  451,  751,  578,  178,
   83,  437,  749,  590,  189,
   76,  424,  746,  602,  200,
   69,  411,  743,  614,  211,
   63,  398,  739,  626,  222,
   57,  386,  734,  637,  234,
   52,  373,  729,  648,  246,
   46,  360,  725,  659,  258,
   41,  347,  719,  670,  271,
   37,  335,  713,  680,  283,
   32,  322,  707,  690,  297,
  314,  710,  710,  314,    0,
```

```
  297,  690,  707,  322,   32,
  283,  680,  713,  335,   37,
  271,  670,  719,  347,   41,
  258,  659,  725,  360,   46,
  246,  648,  729,  373,   52,
  234,  637,  734,  386,   57,
  222,  626,  739,  398,   63,
  211,  614,  743,  411,   69,
  200,  602,  746,  424,   76,
  189,  590,  749,  437,   83,
  178,  578,  751,  451,   90,
  168,  566,  753,  464,   97,
  158,  553,  755,  477,  105,
  148,  541,  756,  490,  113,
  139,  528,  757,  503,  121};
```

7.4.7.4.2.3 ppfcoef_scale_eq_4_32_phases_flip.dat

```
5,   32,   11,
  116,  515,  786,  515,  116,
  107,  502,  785,  530,  124,
   99,  488,  784,  544,  133,
   92,  473,  783,  557,  143,
   85,  459,  781,  571,  152,
   78,  445,  778,  585,  162,
   71,  431,  775,  598,  173,
   65,  417,  772,  611,  183,
   59,  403,  767,  624,  195,
   53,  389,  763,  637,  206,
   48,  375,  758,  649,  218,
   43,  362,  752,  661,  230,
   38,  348,  747,  673,  242,
   34,  334,  740,  685,  255,
   30,  321,  733,  696,  268,
   26,  308,  726,  707,  281,
  298,  726,  726,  298,    0,
  281,  707,  726,  308,   26,
  268,  696,  733,  321,   30,
  255,  685,  740,  334,   34,
  242,  673,  747,  348,   38,
  230,  661,  752,  362,   43,
  218,  649,  758,  375,   48,
  206,  637,  763,  389,   53,
  195,  624,  767,  403,   59,
  183,  611,  772,  417,   65,
  173,  598,  775,  431,   71,
  162,  585,  778,  445,   78,
  152,  571,  781,  459,   85,
  143,  557,  783,  473,   92,
  133,  544,  784,  488,   99,
  124,  530,  785,  502,  107};
```

7.4.7.4.2.4 ppfcoef_scale_eq_5_32_phases_flip.dat

```
5,   32,   11,
   98,  515,  822,  515,   98,
   90,  500,  821,  531,  106,
   83,  484,  820,  547,  114,
   75,  469,  819,  562,  123,
   69,  453,  816,  577,  133,
   63,  438,  813,  592,  142,
   57,  422,  809,  607,  153,
   51,  407,  805,  622,  163,
   46,  391,  801,  636,  174,
   41,  376,  795,  650,  186,
   37,  361,  789,  664,  197,
```

```
 32,  347,  782,  678,  209,
 28,  332,  775,  691,  222,
 25,  317,  767,  704,  235,
 22,  303,  759,  716,  248,
 18,  289,  750,  729,  262,
278,  746,  746,  278,    0,
262,  729,  750,  289,   18,
248,  716,  759,  303,   22,
235,  704,  767,  317,   25,
222,  691,  775,  332,   28,
209,  678,  782,  347,   32,
197,  664,  789,  361,   37,
186,  650,  795,  376,   41,
174,  636,  801,  391,   46,
163,  622,  805,  407,   51,
153,  607,  809,  422,   57,
142,  592,  813,  438,   63,
133,  577,  816,  453,   69,
123,  562,  819,  469,   75,
114,  547,  820,  484,   83,
106,  531,  821,  500,   90};
```

7.4.7.4.2.5 ppfcoef_scale_eq_6_32_phases_flip.dat

```
5,   32,   11,
   77,  513,  868,  513,   77,
   70,  496,  867,  531,   84,
   63,  479,  866,  548,   92,
   57,  461,  864,  566,  100,
   51,  444,  861,  583,  109,
   46,  427,  857,  600,  118,
   41,  409,  853,  617,  128,
   36,  393,  847,  633,  139,
   32,  376,  841,  650,  149,
   28,  359,  835,  666,  160,
   24,  343,  827,  682,  172,
   21,  327,  819,  697,  184,
   18,  311,  810,  712,  197,
   15,  296,  800,  727,  210,
   13,  281,  790,  741,  223,
   11,  266,  779,  755,  237,
  253,  771,  771,  253,    0,
  237,  755,  779,  266,   11,
  223,  741,  790,  281,   13,
  210,  727,  800,  296,   15,
  197,  712,  810,  311,   18,
  184,  697,  819,  327,   21,
  172,  682,  827,  343,   24,
  160,  666,  835,  359,   28,
  149,  650,  841,  376,   32,
  139,  633,  847,  393,   36,
  128,  617,  853,  409,   41,
  118,  600,  857,  427,   46,
  109,  583,  861,  444,   51,
  100,  566,  864,  461,   57,
   92,  548,  866,  479,   63,
   84,  531,  867,  496,   70};
```

7.4.7.4.2.6 ppfcoef_scale_eq_7_32_phases_flip.dat

```
5,   32,   11,
   53,  510,  922,  510,   53,
   47,  490,  922,  529,   60,
   41,  470,  921,  549,   67,
   36,  451,  918,  569,   74,
   32,  431,  915,  588,   82,
```

```
  27,  412,  910,  608,   91,
  23,  393,  905,  627,  100,
  20,  374,  898,  646,  110,
  17,  356,  890,  665,  120,
  14,  337,  882,  684,  131,
  11,  320,  873,  702,  142,
   9,  302,  863,  720,  154,
   7,  285,  852,  737,  167,
   6,  269,  840,  753,  180,
   4,  253,  827,  770,  194,
   3,  237,  815,  785,  208,
 223,  801,  801,  223,    0,
 208,  785,  815,  237,    3,
 194,  770,  827,  253,    4,
 180,  753,  840,  269,    6,
 167,  737,  852,  285,    7,
 154,  720,  863,  302,    9,
 142,  702,  873,  320,   11,
 131,  684,  882,  337,   14,
 120,  665,  890,  356,   17,
 110,  646,  898,  374,   20,
 100,  627,  905,  393,   23,
  91,  608,  910,  412,   27,
  82,  588,  915,  431,   32,
  74,  569,  918,  451,   36,
  67,  549,  921,  470,   41,
  60,  529,  922,  490,   47};
```

7.4.7.4.2.6.1 ppfcoef_scale_eq_8div16_32_phases_ver_5tap_flip.dat

```
5  32 11
   28  502  988  502   28
   24  479  987  524   34
   19  457  985  547   40
   15  435  982  570   46
   12  413  978  592   53
    9  392  972  614   61
    6  371  965  637   69
    4  350  957  659   78
    2  330  948  680   88
    0  310  938  702   98
   -1  291  926  723  109
   -2  272  914  744  120
   -3  254  900  764  133
   -3  237  886  783  145
   -4  220  871  802  159
   -4  204  855  820  173
  188  836  836  188    0
  173  820  855  204   -4
  159  802  871  220   -4
  145  783  886  237   -3
  133  764  900  254   -3
  120  744  914  272   -2
  109  723  926  291   -1
   98  702  938  310    0
   88  680  948  330    2
   78  659  957  350    4
   69  637  965  371    6
   61  614  972  392    9
   53  592  978  413   12
   46  570  982  435   15
   40  547  985  457   19
   34  524  987  479   24
```

7.4.7.4.2.6.2 ppfcoef_scale_eq_9div16_32_phases_ver_5tap_flip.dat

```
5  32 11
    3  489 1064  489    3
    0  464 1062  515    7
   -3  439 1060  540   12
   -5  414 1056  566   17
   -7  390 1050  592   23
   -9  366 1044  618   29
  -10  343 1035  644   36
  -11  320 1025  670   44
  -12  298 1014  695   53
  -12  277 1001  720   62
  -12  256  987  745   72
  -12  236  972  769   83
  -12  217  956  792   95
  -11  199  938  815  107
  -10  181  920  837  120
  -10  165  900  859  134
  148  876  876  148    0
  134  859  900  165  -10
  120  837  920  181  -10
  107  815  938  199  -11
   95  792  956  217  -12
   83  769  972  236  -12
   72  745  987  256  -12
   62  720 1001  277  -12
   53  695 1014  298  -12
   44  670 1025  320  -11
   36  644 1035  343  -10
   29  618 1044  366   -9
   23  592 1050  390   -7
   17  566 1056  414   -5
   12  540 1060  439   -3
    7  515 1062  464    0
```

7.4.7.4.2.6.3 ppfcoef_scale_eq_10div16_32_phases_ver_5tap_flip.dat

```
5  32 11
  -20  470 1148  470  -20
  -22  442 1147  499  -18
  -23  413 1144  529  -15
  -24  386 1139  558  -11
  -24  359 1132  588   -7
  -24  333 1124  618   -3
  -24  308 1113  648    3
  -23  283 1101  678    9
  -23  260 1088  707   16
  -22  237 1072  737   24
  -21  215 1056  765   33
  -19  194 1037  793   43
  -18  174 1017  822   53
  -16  156  995  848   65
  -15  138  973  875   77
  -13  121  949  900   91
  105  919  919  105    0
   91  900  949  121  -13
   77  875  973  138  -15
   65  848  995  156  -16
   53  822 1017  174  -18
   43  793 1037  194  -19
   33  765 1056  215  -21
   24  737 1072  237  -22
   16  707 1088  260  -23
    9  678 1101  283  -23
    3  648 1113  308  -24
   -3  618 1124  333  -24
```

```
 -7  588 1132  359  -24
-11  558 1139  386  -24
-15  529 1144  413  -23
-18  499 1147  442  -22
```

7.4.7.4.2.6.4 ppfcoef_scale_eq_11div16_32_phases_ver_5tap_flip.dat

```
5  32 11
-40  444 1240  444  -40
-40  412 1240  476  -40
-40  381 1236  510  -39
-39  350 1231  544  -38
-37  321 1223  577  -36
-36  293 1212  612  -33
-34  265 1200  646  -29
-32  239 1185  681  -25
-30  214 1169  715  -20
-28  190 1150  750  -14
-26  167 1130  783   -6
-23  146 1107  816    2
-21  126 1083  849   11
-19  107 1057  882   21
-17   90 1030  913   32
-15   73 1002  943   45
 58  966  966   58    0
 45  943 1002   73  -15
 32  913 1030   90  -17
 21  882 1057  107  -19
 11  849 1083  126  -21
  2  816 1107  146  -23
 -6  783 1130  167  -26
-14  750 1150  190  -28
-20  715 1169  214  -30
-25  681 1185  239  -32
-29  646 1200  265  -34
-33  612 1212  293  -36
-36  577 1223  321  -37
-38  544 1231  350  -39
-39  510 1236  381  -40
-40  476 1240  412  -40
```

7.4.7.4.2.6.5 ppfcoef_scale_eq_12div16_32_phases_ver_5tap_flip.dat

```
5  32 11
-56  409 1342  409  -56
-54  373 1342  445  -58
-51  339 1337  482  -59
-49  306 1330  521  -60
-46  274 1321  559  -60
-43  244 1308  598  -59
-40  215 1293  638  -58
-36  187 1275  678  -56
-33  161 1255  718  -53
-30  137 1233  757  -49
-27  114 1208  797  -44
-24   93 1182  836  -39
-21   73 1152  875  -31
-18   55 1122  912  -23
-16   38 1090  950  -14
-14   23 1056  986   -3
  9 1015 1015    9    0
 -3  986 1056   23  -14
-14  950 1090   38  -16
-23  912 1122   55  -18
-31  875 1152   73  -21
-39  836 1182   93  -24
```

```
  -44  797 1208  114  -27
  -49  757 1233  137  -30
  -53  718 1255  161  -33
  -56  678 1275  187  -36
  -58  638 1293  215  -40
  -59  598 1308  244  -43
  -60  559 1321  274  -46
  -60  521 1330  306  -49
  -59  482 1337  339  -51
  -58  445 1342  373  -54
```

7.4.7.4.2.6.6 ppfcoef_scale_eq_13div16_32_phases_ver_5tap_flip.dat

```
5  32 11
  -65  364 1450  364  -65
  -61  326 1448  404  -69
  -57  289 1443  445  -72
  -53  253 1435  488  -75
  -48  220 1423  531  -78
  -44  188 1408  576  -80
  -40  158 1390  621  -81
  -36  130 1370  666  -82
  -32  103 1346  713  -82
  -28   79 1320  758  -81
  -24   56 1290  805  -79
  -21   36 1259  850  -76
  -18   17 1224  896  -71
  -15    0 1188  940  -65
  -12  -15 1149  984  -58
  -10  -28 1109 1027  -50
  -40 1064 1064  -40    0
  -50 1027 1109  -28  -10
  -58  984 1149  -15  -12
  -65  940 1188    0  -15
  -71  896 1224   17  -18
  -76  850 1259   36  -21
  -79  805 1290   56  -24
  -81  758 1320   79  -28
  -82  713 1346  103  -32
  -82  666 1370  130  -36
  -81  621 1390  158  -40
  -80  576 1408  188  -44
  -78  531 1423  220  -48
  -75  488 1435  253  -53
  -72  445 1443  289  -57
  -69  404 1448  326  -61
```

7.4.7.4.2.6.7 ppfcoef_scale_eq_14div16_32_phases_ver_5tap_flip.dat

```
5  32 11
  -67  310 1562  310  -67
  -61  269 1559  353  -72
  -55  230 1553  398  -78
  -50  193 1543  445  -83
  -44  158 1529  493  -88
  -39  125 1512  543  -93
  -34   94 1491  594  -97
  -30   66 1468  645 -101
  -25   41 1439  697 -104
  -22   17 1408  751 -106
  -18   -4 1373  804 -107
  -15  -23 1336  857 -107
  -12  -40 1296  910 -106
   -9  -55 1253  962 -103
   -7  -67 1208 1013  -99
   -5  -78 1161 1064  -94
```

```
  -86 1110 1110  -86    0
  -94 1064 1161  -78   -5
  -99 1013 1208  -67   -7
 -103  962 1253  -55   -9
 -106  910 1296  -40  -12
 -107  857 1336  -23  -15
 -107  804 1373   -4  -18
 -106  751 1408   17  -22
 -104  697 1439   41  -25
 -101  645 1468   66  -30
  -97  594 1491   94  -34
  -93  543 1512  125  -39
  -88  493 1529  158  -44
  -83  445 1543  193  -50
  -78  398 1553  230  -55
  -72  353 1559  269  -61
```

7.4.7.4.2.6.8 ppfcoef_scale_eq_15div16_32_phases_ver_5tap_flip.dat

```
5  32 11
  -61  248 1674  248  -61
  -54  204 1673  293  -68
  -47  163 1665  342  -75
  -41  125 1654  392  -82
  -35   90 1638  445  -90
  -29   57 1618  499  -97
  -24   27 1593  556 -104
  -20    0 1565  613 -110
  -16  -24 1532  672 -116
  -12  -46 1495  732 -121
   -9  -65 1455  793 -126
   -6  -81 1411  854 -130
   -4  -95 1364  915 -132
   -2 -107 1315  975 -133
    0 -116 1262 1035 -133
    1 -123 1208 1094 -132
 -128 1152 1152 -128    0
 -132 1094 1208 -123    1
 -133 1035 1262 -116    0
 -133  975 1315 -107   -2
 -132  915 1364  -95   -4
 -130  854 1411  -81   -6
 -126  793 1455  -65   -9
 -121  732 1495  -46  -12
 -116  672 1532  -24  -16
 -110  613 1565    0  -20
 -104  556 1593   27  -24
  -97  499 1618   57  -29
  -90  445 1638   90  -35
  -82  392 1654  125  -41
  -75  342 1665  163  -47
  -68  293 1673  204  -54
```

7.4.7.4.3 VS (Bilinear Filter Coefficients)

7.4.7.4.3.1 ppfcoef_scale_eq_1_32_phases_flip_PPF3_peak5_gain_eq_1_25.dat

This is not applicable for this device

```
7  32 11
  -11 -106  190 1902  190 -106  -11
   -9 -105  153 1897  230 -105  -13
   -8 -105  121 1887  274 -105  -16
   -6 -104   91 1869  320 -104  -18
   -5 -102   65 1843  370 -102  -21
   -4 -101   42 1812  424 -101  -24
```

```
 -3  -99   20 1776  480  -99  -27
 -2  -96    3 1730  539  -96  -30
 -1  -93  -12 1679  602  -93  -34
 -1  -90  -26 1627  665  -90  -37
  0  -87  -37 1568  732  -87  -41
  0  -84  -46 1506  801  -84  -45
  0  -80  -54 1439  871  -80  -48
  0  -76  -60 1371  941  -76  -52
  1  -72  -65 1299 1013  -72  -56
  1  -68  -69 1227 1085  -68  -60
-64  -64 1152 1152  -64  -64    0
-60  -68 1085 1227  -69  -68    1
-56  -72 1013 1299  -65  -72    1
-52  -76  941 1371  -60  -76    0
-48  -80  871 1439  -54  -80    0
-45  -84  801 1506  -46  -84    0
-41  -87  732 1568  -37  -87    0
-37  -90  665 1627  -26  -90   -1
-34  -93  602 1679  -12  -93   -1
-30  -96  539 1730    3  -96   -2
-27  -99  480 1776   20  -99   -3
-24 -101  424 1812   42 -101   -4
-21 -102  370 1843   65 -102   -5
-18 -104  320 1869   91 -104   -6
-16 -105  274 1887  121 -105   -8
-13 -105  230 1897  153 -105   -9
```

7.4.8 VIP Video Port Direct Memory Access (VPDMA)

7.4.8.1 VPDMA Introduction

The VPDMA primary function is to move data between external memory and internal processing modules that source or sink data. VPDMA is capable buffering this data and then delivering the data as demanded to the modules as programmed. The modules that source or sink data are referred to as clients. A channel is setup inside the VPDMA to connect a specific memory buffer to a specific client. The VPDMA centralizes the DMA control functions and buffering required to allow all the clients to minimize the effect of long latency times. The VPDMA also supports a descriptor based mode where lists of descriptors can be setup to configure all the channels as they become available.

Additionally, in a third-party configuration, the VPDMA is capable of performing DMA transfers as requested by the pulsing of an event strobe. The VPDMA is capable of generation of an address which may reach any location in the range for which it is configured. It is capable of moving this data either to or from a Shared Buffer and ultimately to or from a Client Buffer. For the two type of Client Buffers that are used to drive data into a subsystem (Streaming Buffer and Random Access Buffer) data is either pushed into the buffer or pulled out of the buffer dependent upon the direction of the data transfer. For third-party DMA operations the data is transferred into a Client buffer, and then transferred out of the Client Buffer to the transfer destination. These transfers are triggered by an Event pulse to each of the Client Buffers which are Routing Buffer types.

7.4.8.2 VPDMA Basic Definitions

7.4.8.2.1 Client

The modules that source or sink data are referred to as clients. The clients of the VPDMA are the physical between the processing modules (VIP) and external memory. A channel is the mechanism inside the VPDMA that connects a specific memory buffer or transfer to a specific client.

The start event can also be selected by a channel attribute or to be controlled by an internal frame signal controlled by the List Manager.

7.4.8.2.2 Channel

The VPDMA requires a channel to be setup for each group of transfers. All the channels are described through a Data Transfer Descriptor that has a common format. The client that the channel is mapped to interprets the information in the descriptor to perform the requested data transfer.

Each of the channels has a type of data that it can support based upon the client that it services. The VPDMA supports three types of channels:

- **YUV Channel** - Clients taking data YUV data
- **RGB Channel** - Clients taking RGB data
- **Miscellaneous Channel** - The Miscellaneous channel type is for any data type that is not a normal video type. The Miscellaneous channel type makes no assumptions on data type and just passes the data to the client and supports a single buffer for the client.

7.4.8.2.3 List

A list is a group of descriptors that makes up a set of DMA transfers that need to be completed. The VPDMA supports one kind of list only:

- The **Regular List** is a single list that the VPDMA will execute each descriptor once and initiate an interrupt when the list has completed. A regular list can contain any kind of descriptor without limitation and be of any size.

The VPDMA Controller works on lists of descriptors. In this mode the processor writes the lists of descriptors in the order it wants them executed. It then writes the location of the list to the VIP_LIST_ADDR register, followed by writing the size (bit LIST_SIZE) and type (bit LIST_TYPE) of the list, and list number (bit LIST_NUM) to the VIP_LIST_ATTR register. The List Manager module then schedules a DMA transfer to pull in the portion of the list that it can store in internal VPDMA memory. The List Manager will sequentially process the active list of descriptors until either the descriptor requires the use of a client that is currently active, or if it is waiting for the next portion of the list to be transferred from a DMA request. If there is no active list to process the list manager will go into an IDLE mode waiting for any client that is blocking a list or a list DMA transfer to complete.

All the DMA transfers are controlled by List Manager module inside VPDMA. List Manager needs to be loaded with FIRMWARE, before any DMA transfer from memory, after the VPDMA reset. The first MMR write to VIP_LIST_ADDR register after VPDMA reset should be the address of the memory buffer(128-bit aligned, that is, last four bits of the buffer address should be zero) where the firmware is stored. List Manager then schedules a DMA transaction to fetch the firmware and sets the VIP_LIST_ATTR[19] RDY bit after the firmware loading is complete.

7.4.8.2.4 Data Formats Supported

Following list summarizes the data formats supported in the VPDMA. For more inforamtion see Section 7.4.8.9, *VPDMA Data Formats*.

- RGB Data Types:
  - RGB16-565
  - ARGB-1555
  - ARGB-4444
  - RGBA-5551
  - RGBA-4444
  - ARGB24-6666
  - RGB24-888
  - ARGB32-8888
  - RGBA24-6666
  - RGBA32-8888
  - BGR16-565
  - ABGR-1555

  - ABGR-4444
  - BGRA-5551
  - BGRA-4444
  - ABGR24-6666
  - BGR24-888
  - ABGR32-8888
  - BGRA24-6666
  - BGRA32-8888
- YUV Data Types:
  - Y 4:4:4
  - Y 4:2:2
  - Y 4:2:0
  - C 4:4:4
  - C 4:2:2
  - C 4:2:0
  - CY 4:2:2
  - YCbC 4:4:4
  - YC 4:2:2

NOTE: VPDMA supports swapping formats (RGB/BGR and Cb/Cr)

7.4.8.3 VPDMA Client Buffering and Functionality

Table 7-29 lists for each client:

- The channels used, amount of buffering allocated for it, and the shared buffer used for its memory
- The line sizes it handles for tiled and non-tiled memory spaces, as well as any additional features it supports

Table 7-29. VPDMA Client Buffering and Functionality

Client	Channel(s)	Tiled Memory Max Line Size	Non-Tiled Memory Max Line Size	Additional Features
vip1_lo_y	vip1_mult_porta_src0 vip1_mult_porta_src1 vip1_mult_porta_src2 vip1_mult_porta_src3 vip1_mult_porta_src4 vip1_mult_porta_src5 vip1_mult_porta_src6 vip1_mult_porta_src7 vip1_mult_porta_src8 vip1_mult_porta_src9 vip1_mult_porta_src10 vip1_mult_porta_src11 vip1_mult_porta_src12 vip1_mult_porta_src13 vip1_mult_porta_src14 vip1_mult_porta_src15 vip1_portb_luma vip1_portb_rgb	1920 (color seperate) 960 (interleaved)	4096	TILED

Table 7-29. VPDMA Client Buffering and Functionality (continued)

vip1_lo_uv	vip1_mult_portb_src0 vip1_mult_portb_src1 vip1_mult_portb_src2 vip1_mult_portb_src3 vip1_mult_portb_src4 vip1_mult_portb_src5 vip1_mult_portb_src6 vip1_mult_portb_src7 vip1_mult_portb_src8 vip1_mult_portb_src9 vip1_mult_portb_src10 vip1_mult_portb_src11 vip1_mult_portb_src12 vip1_mult_portb_src13 vip1_mult_portb_src14 vip1_mult_portb_src15 vip1_portb_chroma	1920 (color seperate) 960 (interleaved)	4096	TILED
vip1_up_y	vip1_porta_luma vip1_porta_rgb	1920 (color seperate) 960 (interleaved)	4096	TILED
vip1_up_uv	vip1_porta_chroma	1920 (color seperate) 960 (interleaved)	4096	TILED
vip2_lo_y	vip2_mult_porta_src0 vip2_mult_porta_src1 vip2_mult_porta_src2 vip2_mult_porta_src3 vip2_mult_porta_src4 vip2_mult_porta_src5 vip2_mult_porta_src6 vip2_mult_porta_src7 vip2_mult_porta_src8 vip2_mult_porta_src9 vip2_mult_porta_src10 vip2_mult_porta_src11 vip2_mult_porta_src12 vip2_mult_porta_src13 vip2_mult_porta_src14 vip2_mult_porta_src15 vip2_portb_luma vip2_portb_rgb	1920 (color seperate) 960 (interleaved)	4096	TILED
vip2_lo_uv	vip2_mult_portb_src0 vip2_mult_portb_src1 vip2_mult_portb_src2 vip2_mult_portb_src3 vip2_mult_portb_src4 vip2_mult_portb_src5 vip2_mult_portb_src6 vip2_mult_portb_src7 vip2_mult_portb_src8 vip2_mult_portb_src9 vip2_mult_portb_src10 vip2_mult_portb_src11 vip2_mult_portb_src12 vip2_mult_portb_src13 vip2_mult_portb_src14 vip2_mult_portb_src15 vip2_portb_chroma	1920 (color seperate) 960 (interleaved)	4096	TILED
vip2_up_y	vip2_porta_luma vip2_porta_rgb	1920 (color seperate) 960 (interleaved)	4096	TILED
vip2_up_uv	vip2_porta_chroma	1920 (color seperate) 960 (interleaved)	4096	TILED
vpi_ctl		Tiled Data Not Supported	4096	

Table 7-29. VPDMA Client Buffering and Functionality (continued)

vip1_anc_a	vip1_mult_anca_src0 vip1_mult_anca_src1 vip1_mult_anca_src2 vip1_mult_anca_src3 vip1_mult_anca_src4 vip1_mult_anca_src5 vip1_mult_anca_src6 vip1_mult_anca_src7 vip1_mult_anca_src8 vip1_mult_anca_src9 vip1_mult_anca_src10 vip1_mult_anca_src11 vip1_mult_anca_src12 vip1_mult_anca_src13 vip1_mult_anca_src14 vip1_mult_anca_src15	Tiled Data Not Supported	4096	
vip1_anc_b	vip1_mult_ancb_src0 vip1_mult_ancb_src1 vip1_mult_ancb_src2 vip1_mult_ancb_src3 vip1_mult_ancb_src4 vip1_mult_ancb_src5 vip1_mult_ancb_src6 vip1_mult_ancb_src7 vip1_mult_ancb_src8 vip1_mult_ancb_src9 vip1_mult_ancb_src10 vip1_mult_ancb_src11 vip1_mult_ancb_src12 vip1_mult_ancb_src13 vip1_mult_ancb_src14 vip1_mult_ancb_src15	Tiled Data Not Supported	4096	
vip2_anc_a	vip2_mult_anca_src0 vip2_mult_anca_src1 vip2_mult_anca_src2 vip2_mult_anca_src3 vip2_mult_anca_src4 vip2_mult_anca_src5 vip2_mult_anca_src6 vip2_mult_anca_src7 vip2_mult_anca_src8 vip2_mult_anca_src9 vip2_mult_anca_src10 vip2_mult_anca_src11 vip2_mult_anca_src12 vip2_mult_anca_src13 vip2_mult_anca_src14 vip2_mult_anca_src15	Tiled Data Not Supported	4096	

7.4.8.4 VPDMA Channels Assignment

Table 7-30 lists all of the channels in VPDMA and its base attributes. The Data Type column states what type of data YUV, RGB or OTHER the channel handles and in parentheses are the legal data type values that can be entered into a data transfer decriptor. The Client field states the name of the Client and in parentheses it states the reference number in Figure 7-4, *VIP Block Diagram*.

Table 7-30. VPDMA Channels Assignment

Channel	Description	Channel Number	Data Type	Client
vip1_mult_porta_src0	Video Input 1 Port A Channel 0	38	YUV (0x7)	vip1_lo_y (2)
vip1_mult_porta_src1	Video Input 1 Port A Channel 1	39	YUV (0x7)	vip1_lo_y (2)
vip1_mult_porta_src2	Video Input 1 Port A Channel 2	40	YUV (0x7)	vip1_lo_y (2)
vip1_mult_porta_src3	Video Input 1 Port A Channel 3	41	YUV (0x7)	vip1_lo_y (2)
vip1_mult_porta_src4	Video Input 1 Port A Channel 4	42	YUV (0x7)	vip1_lo_y (2)

Table 7-30. VPDMA Channels Assignment (continued)

Channel	Description	Channel Number	Data Type	Client
vip1_mult_porta_src5	Video Input 1 Port A Channel 5	43	YUV (0x7)	vip1_lo_y (2)
vip1_mult_porta_src6	Video Input 1 Port A Channel 6	44	YUV (0x7)	vip1_lo_y (2)
vip1_mult_porta_src7	Video Input 1 Port A Channel 7	45	YUV (0x7)	vip1_lo_y (2)
vip1_mult_porta_src8	Video Input 1 Port A Channel 8	46	YUV (0x7)	vip1_lo_y (2)
vip1_mult_porta_src9	Video Input 1 Port A Channel 9	47	YUV (0x7)	vip1_lo_y (2)
vip1_mult_porta_src10	Video Input 1 Port A Channel 10	48	YUV (0x7)	vip1_lo_y (2)
vip1_mult_porta_src11	Video Input 1 Port A Channel 11	49	YUV (0x7)	vip1_lo_y (2)
vip1_mult_porta_src12	Video Input 1 Port A Channel 12	50	YUV (0x7)	vip1_lo_y (2)
vip1_mult_porta_src13	Video Input 1 Port A Channel 13	51	YUV (0x7)	vip1_lo_y (2)
vip1_mult_porta_src14	Video Input 1 Port A Channel 14	52	YUV (0x7)	vip1_lo_y (2)
vip1_mult_porta_src15	Video Input 1 Port A Channel 15	53	YUV (0x7)	vip1_lo_y (2)
vip1_mult_portb_src0	Video Input 1 Port B Channel 0	54	YUV (0x7)	vip1_lo_uv(1)
vip1_mult_portb_src1	Video Input 1 Port B Channel 1	55	YUV (0x7)	vip1_lo_uv(1)
vip1_mult_portb_src2	Video Input 1 Port B Channel 2	56	YUV (0x7)	vip1_lo_uv(1)
vip1_mult_portb_src3	Video Input 1 Port B Channel 3	57	YUV (0x7)	vip1_lo_uv(1)
vip1_mult_portb_src4	Video Input 1 Port B Channel 4	58	YUV (0x7)	vip1_lo_uv(1)
vip1_mult_portb_src5	Video Input 1 Port B Channel 5	59	YUV (0x7)	vip1_lo_uv(1)
vip1_mult_portb_src6	Video Input 1 Port B Channel 6	60	YUV (0x7)	vip1_lo_uv(1)
vip1_mult_portb_src7	Video Input 1 Port B Channel 7	61	YUV (0x7)	vip1_lo_uv (1)
vip1_mult_portb_src8	Video Input 1 Port B Channel 8	62	YUV (0x7)	vip1_lo_uv (1)
vip1_mult_portb_src9	Video Input 1 Port B Channel 9	63	YUV (0x7)	vip1_lo_uv (1)
vip1_mult_portb_src10	Video Input 1 Port B Channel 10	64	YUV (0x7)	vip1_lo_uv (1)
vip1_mult_portb_src11	Video Input 1 Port B Channel 11	65	YUV (0x7)	vip1_lo_uv (1)
vip1_mult_portb_src12	Video Input 1 Port B Channel 12	66	YUV (0x7)	vip1_lo_uv (1)
vip1_mult_portb_src13	Video Input 1 Port B Channel 13	67	YUV (0x7)	vip1_lo_uv (1)
vip1_mult_portb_src14	Video Input 1 Port B Channel 14	68	YUV (0x7)	vip1_lo_uv (1)
vip1_mult_portb_src15	Video Input 1 Port B Channel 15	69	YUV (0x7)	vip1_lo_uv (1)
vip1_mult_anca_src0	Video Input 1 Port A Ancillary Data Channel 0	70	OTHER (8)	vip1_anc_a(2)
vip1_mult_anca_src1	Video Input 1 Port A Ancillary Data Channel 1	71	OTHER (8)	vip1_anc_a(2)
vip1_mult_anca_src2	Video Input 1 Port A Ancillary Data Channel 2	72	OTHER (8)	vip1_anc_a(2)
vip1_mult_anca_src3	Video Input 1 Port A Ancillary Data Channel 3	73	OTHER (8)	vip1_anc_a(2)
vip1_mult_anca_src4	Video Input 1 Port A Ancillary Data Channel 4	74	OTHER (8)	vip1_anc_a(2)
vip1_mult_anca_src5	Video Input 1 Port A Ancillary Data Channel 5	75	OTHER (8)	vip1_anc_a(2)
vip1_mult_anca_src6	Video Input 1 Port A Ancillary Data Channel 6	76	OTHER (8)	vip1_anc_a(2)
vip1_mult_anca_src7	Video Input 1 Port A Ancillary Data Channel 7	77	OTHER (8)	vip1_anc_a(2)
vip1_mult_anca_src8	Video Input 1 Port A Ancillary Data Channel 8	78	OTHER (8)	vip1_anc_a(2)
vip1_mult_anca_src9	Video Input 1 Port A Ancillary Data Channel 9	79	OTHER (8)	vip1_anc_a(2)
vip1_mult_anca_src10	Video Input 1 Port A Ancillary Data Channel 10	80	OTHER (8)	vip1_anc_a(2)
vip1_mult_anca_src11	Video Input 1 Port A Ancillary Data Channel 11	81	OTHER (8)	vip1_anc_a(2)

Table 7-30. VPDMA Channels Assignment (continued)

Channel	Description	Channel Number	Data Type	Client
vip1_mult_anca_src12	Video Input 1 Port A Ancillary Data Channel 12	82	OTHER (8)	vip1_anc_a(2)
vip1_mult_anca_src13	Video Input 1 Port A Ancillary Data Channel 13	83	OTHER (8)	vip1_anc_a(2)
vip1_mult_anca_src14	Video Input 1 Port A Ancillary Data Channel 14	84	OTHER (8)	vip1_anc_a(2)
vip1_mult_anca_src15	Video Input 1 Port A Ancillary Data Channel 15	85	OTHER (8)	vip1_anc_a(2)
vip1_mult_ancb_src0	Video Input 1 Port B Ancillary Data Channel 0	86	OTHER (8)	vip1_anc_b(2)
vip1_mult_ancb_src1	Video Input 1 Port B Ancillary Data Channel 1	87	OTHER (8)	vip1_anc_b(2)
vip1_mult_ancb_src2	Video Input 1 Port B Ancillary Data Channel 2	88	OTHER (8)	vip1_anc_b(2)
vip1_mult_ancb_src3	Video Input 1 Port B Ancillary Data Channel 3	89	OTHER (8)	vip1_anc_b(2)
vip1_mult_ancb_src4	Video Input 1 Port B Ancillary Data Channel 4	90	OTHER (8)	vip1_anc_b(2)
vip1_mult_ancb_src5	Video Input 1 Port B Ancillary Data Channel 5	91	OTHER (8)	vip1_anc_b(2)
vip1_mult_ancb_src6	Video Input 1 Port B Ancillary Data Channel 6	92	OTHER (8)	vip1_anc_b(2)
vip1_mult_ancb_src7	Video Input 1 Port B Ancillary Data Channel 7	93	OTHER (8)	vip1_anc_b(2)
vip1_mult_ancb_src8	Video Input 1 Port B Ancillary Data Channel 8	94	OTHER (8)	vip1_anc_b(2)
vip1_mult_ancb_src9	Video Input 1 Port B Ancillary Data Channel 9	95	OTHER (8)	vip1_anc_b(2)
vip1_mult_ancb_src10	Video Input 1 Port B Ancillary Data Channel 10	96	OTHER (8)	vip1_anc_b(2)
vip1_mult_ancb_src11	Video Input 1 Port B Ancillary Data Channel 11	97	OTHER (8)	vip1_anc_b(2)
vip1_mult_ancb_src12	Video Input 1 Port B Ancillary Data Channel 12	98	OTHER (8)	vip1_anc_b(2)
vip1_mult_ancb_src13	Video Input 1 Port B Ancillary Data Channel 13	99	OTHER (8)	vip1_anc_b(2)
vip1_mult_ancb_src14	Video Input 1 Port B Ancillary Data Channel 14	100	OTHER (8)	vip1_anc_b(2)
vip1_mult_ancb_src15	Video Input 1 Port B Ancillary Data Channel 15	101	OTHER (8)	vip1_anc_b(2)
vip1_porta_luma	Video Input 1 Port A 420 Data Luma	102	YUV (0x1, 0x2, 0x7)	vip1_up_y (1)
vip1_porta_chroma	Video Input 1 Port A 420 Data Chroma	103	YUV (0x5, 0x6, 0x7)	vip1_up_uv(1)
vip1_portb_luma	Video Input 1 Port B 420 Data Luma	104	YUV (0x1, 0x2, 0x7)	vip1_lo_y (2)
vip1_portb_chroma	Video Input 1 Port B 420 Data Chroma	105	YUV (0x5, 0x6, 0x7)	vip1_lo_uv (2)
vip1_porta_rgb	Video Input 1 Port A RGB Data	106	RGB (0x0 - 0x8)	vip1_up_y (1)
vip1_portb_rgb	Video Input 1 Port B RGB Data	107	RGB (0x0 - 0x8)	vip1_lo_y (2)
vip2_mult_porta_src0	Video Input 2 Port A Channel 0	108	YUV (0x7)	vip2_lo_y (21)
vip2_mult_porta_src1	Video Input 2 Port A Channel 1	109	YUV (0x7)	vip2_lo_y (21)
vip2_mult_porta_src2	Video Input 2 Port A Channel 2	110	YUV (0x7)	vip2_lo_y (21)
vip2_mult_porta_src3	Video Input 2 Port A Channel 3	111	YUV (0x7)	vip2_lo_y (21)
vip2_mult_porta_src4	Video Input 2 Port A Channel 4	112	YUV (0x7)	vip2_lo_y (21)
vip2_mult_porta_src5	Video Input 2 Port A Channel 5	113	YUV (0x7)	vip2_lo_y (21)
vip2_mult_porta_src6	Video Input 2 Port A Channel 6	114	YUV (0x7)	vip2_lo_y (21)
vip2_mult_porta_src7	Video Input 2 Port A Channel 7	115	YUV (0x7)	vip2_lo_y (21)

Table 7-30. VPDMA Channels Assignment (continued)

Channel	Description	Channel Number	Data Type	Client
vip2_mult_porta_src8	Video Input 2 Port A Channel 8	116	YUV (0x7)	vip2_lo_y (21)
vip2_mult_porta_src9	Video Input 2 Port A Channel 9	117	YUV (0x7)	vip2_lo_y (21)
vip2_mult_porta_src10	Video Input 2 Port A Channel 10	118	YUV (0x7)	vip2_lo_y (21)
vip2_mult_porta_src11	Video Input 2 Port A Channel 11	119	YUV (0x7)	vip2_lo_y (21)
vip2_mult_porta_src12	Video Input 2 Port A Channel 12	120	YUV (0x7)	vip2_lo_y (21)
vip2_mult_porta_src13	Video Input 2 Port A Channel 13	121	YUV (0x7)	vip2_lo_y (21)
vip2_mult_porta_src14	Video Input 2 Port A Channel 14	122	YUV (0x7)	vip2_lo_y (21)
vip2_mult_porta_src15	Video Input 2 Port A Channel 15	123	YUV (0x7)	vip2_lo_y (21)
vip2_mult_portb_src0	Video Input 2 Port B Channel 0	124	YUV (0x7)	vip2_lo_uv (20)
vip2_mult_portb_src1	Video Input 2 Port B Channel 1	125	YUV (0x7)	vip2_lo_uv (20)
vip2_mult_portb_src2	Video Input 2 Port B Channel 2	126	YUV (0x7)	vip2_lo_uv (20)
vip2_mult_portb_src3	Video Input 2 Port B Channel 3	127	YUV (0x7)	vip2_lo_uv (20)
vip2_mult_portb_src4	Video Input 2 Port B Channel 4	128	YUV (0x7)	vip2_lo_uv (20)
vip2_mult_portb_src5	Video Input 2 Port B Channel 5	129	YUV (0x7)	vip2_lo_uv (20)
vip2_mult_portb_src6	Video Input 2 Port B Channel 6	130	YUV (0x7)	vip2_lo_uv (20)
vip2_mult_portb_src7	Video Input 2 Port B Channel 7	131	YUV (0x7)	vip2_lo_uv (20)
vip2_mult_portb_src8	Video Input 2 Port B Channel 8	132	YUV (0x7)	vip2_lo_uv (20)
vip2_mult_portb_src9	Video Input 2 Port B Channel 9	133	YUV (0x7)	vip2_lo_uv (20)
vip2_mult_portb_src10	Video Input 2 Port B Channel 10	134	YUV (0x7)	vip2_lo_uv (20)
vip2_mult_portb_src11	Video Input 2 Port B Channel 11	135	YUV (0x7)	vip2_lo_uv (20)
vip2_mult_portb_src12	Video Input 2 Port B Channel 12	136	YUV (0x7)	vip2_lo_uv (20)
vip2_mult_portb_src13	Video Input 2 Port B Channel 13	137	YUV (0x7)	vip2_lo_uv (20)
vip2_mult_portb_src14	Video Input 2 Port B Channel 14	138	YUV (0x7)	vip2_lo_uv (20)
vip2_mult_portb_src15	Video Input 2 Port B Channel 15	139	YUV (0x7)	vip2_lo_uv (20)
vip2_mult_anca_src0	Video Input 2 Port A Ancillary Data Channel 0	140	OTHER (8)	vip2_anc_a(21)
vip2_mult_anca_src1	Video Input 2 Port A Ancillary Data Channel 1	141	OTHER (8)	vip2_anc_a(21)
vip2_mult_anca_src2	Video Input 2 Port A Ancillary Data Channel 2	142	OTHER (8)	vip2_anc_a(21)
vip2_mult_anca_src3	Video Input 2 Port A Ancillary Data Channel 3	143	OTHER (8)	vip2_anc_a(21)
vip2_mult_anca_src4	Video Input 2 Port A Ancillary Data Channel 4	144	OTHER (8)	vip2_anc_a(21)
vip2_mult_anca_src5	Video Input 2 Port A Ancillary Data Channel 5	145	OTHER (8)	vip2_anc_a(21)
vip2_mult_anca_src6	Video Input 2 Port A Ancillary Data Channel 6	146	OTHER (8)	vip2_anc_a(21)
vip2_mult_anca_src7	Video Input 2 Port A Ancillary Data Channel 7	147	OTHER (8)	vip2_anc_a(21)
vip2_mult_anca_src8	Video Input 2 Port A Ancillary Data Channel 8	148	OTHER (8)	vip2_anc_a(21)
vip2_mult_anca_src9	Video Input 2 Port A Ancillary Data Channel 9	149	OTHER (8)	vip2_anc_a(21)
vip2_mult_anca_src10	Video Input 2 Port A Ancillary Data Channel 10	150	OTHER (8)	vip2_anc_a(21)
vip2_mult_anca_src11	Video Input 2 Port A Ancillary Data Channel 11	151	OTHER (8)	vip2_anc_a(21)
vip2_mult_anca_src12	Video Input 2 Port A Ancillary Data Channel 12	152	OTHER (8)	vip2_anc_a(21)
vip2_mult_anca_src13	Video Input 2 Port A Ancillary Data Channel 13	153	OTHER (8)	vip2_anc_a(21)

Table 7-30. VPDMA Channels Assignment (continued)

Channel	Description	Channel Number	Data Type	Client
vip2_mult_anca_src14	Video Input 2 Port A Ancillary Data Channel 14	154	OTHER (8)	vip2_anc_a(21)
vip2_mult_anca_src15	Video Input 2 Port A Ancillary Data Channel 15	155	OTHER (8)	vip2_anc_a(21)
vip2_mult_ancb_src0	Video Input 2 Port B Ancillary Data Channel 0	156	OTHER (8)	vip2_anc_b(21)
vip2_mult_ancb_src1	Video Input 2 Port B Ancillary Data Channel 1	157	OTHER (8)	vip2_anc_b(21)
vip2_mult_ancb_src2	Video Input 2 Port B Ancillary Data Channel 2	158	OTHER (8)	vip2_anc_b(21)
vip2_mult_ancb_src3	Video Input 2 Port B Ancillary Data Channel 3	159	OTHER (8)	vip2_anc_b(21)
vip2_mult_ancb_src4	Video Input 2 Port B Ancillary Data Channel 4	160	OTHER (8)	vip2_anc_b(21)
vip2_mult_ancb_src5	Video Input 2 Port B Ancillary Data Channel 5	161	OTHER (8)	vip2_anc_b(21)
vip2_mult_ancb_src6	Video Input 2 Port B Ancillary Data Channel 6	162	OTHER (8)	vip2_anc_b(21)
vip2_mult_ancb_src7	Video Input 2 Port B Ancillary Data Channel 7	163	OTHER (8)	vip2_anc_b(21)
vip2_mult_ancb_src8	Video Input 2 Port B Ancillary Data Channel 8	164	OTHER (8)	vip2_anc_b(21)
vip2_mult_ancb_src9	Video Input 2 Port B Ancillary Data Channel 9	165	OTHER (8)	vip2_anc_b(21)
vip2_mult_ancb_src10	Video Input 2 Port B Ancillary Data Channel 10	166	OTHER (8)	vip2_anc_b(21)
vip2_mult_ancb_src11	Video Input 2 Port B Ancillary Data Channel 11	167	OTHER (8)	vip2_anc_b(21)
vip2_mult_ancb_src12	Video Input 2 Port B Ancillary Data Channel 12	168	OTHER (8)	vip2_anc_b(21)
vip2_mult_ancb_src13	Video Input 2 Port B Ancillary Data Channel 13	169	OTHER (8)	vip2_anc_b(21)
vip2_mult_ancb_src14	Video Input 2 Port B Ancillary Data Channel 14	170	OTHER (8)	vip2_anc_b(21)
vip2_mult_ancb_src15	Video Input 2 Port B Ancillary Data Channel 15	171	OTHER (8)	vip2_anc_b(21)
vip2_porta_luma	Video Input 2 Port A 420 Data Luma	172	YUV (0x1, 0x2, 0x7)	vip2_up_y (20)
vip2_porta_chroma	Video Input 2 Port A 420 Data Chroma	173	YUV (0x5, 0x6, 0x7)	vip2_up_uv(20)
vip2_portb_luma	Video Input 2 Port B 420 Data Luma	174	YUV (0x1, 0x2, 0x7)	vip2_lo_y (21)
vip2_portb_chroma	Video Input 2 Port B 420 Data Chroma	175	YUV (0x5, 0x6, 0x7)	vip2_lo_uv (21)
vip2_porta_rgb	Video Input 2 Port A RGB Data	176	RGB (0x0 - 0x8)	vip2_up_y(20)
vip2_portb_rgb	Video Input 2 Port B RGB Data	177	RGB (0x0 - 0x8)	vip2_lo_y (21)

7.4.8.5 VPDMA MFLAG Mechanism

The device L3_MAIN interconnect accepts MFLAG signals from certain initiators that can influence the internal L3_MAIN arbitration mechanisms. As a result, a higher priority is given to the data traffic initiated by these initiators. The VIP VPDMA can directly drive such MFLAG signals dynamically. The MFLAG generation for VIP VPDMA is enabled by default, and there is no register control over it.

The VPDMA arbitrates between multiple DMA sources within the VIP based on FIFO levels of DMA channels connected to VPDMA. Priority escalation mechanism implemented within VIP subsystem is based on overflow threshold and FIFO margin.

The following is a summary of priority and MFLAG levels provided by the VIP:

- High priority (MFLAG = 3) when FIFO margin is below 25%
- Medium priority (MFLAG = 1) when FIFO margin is between 25% and 50%
- Low priority (MFLAG = 0) when FIFO margin is above 50%

Additionally, the VIP subsystem also generates MReqPriority based upon a programmed descriptor configuration. The MReqPriority configuration influences the arbitration mechanism in the Memory Subsystem only and has no influence on the arbitration that takes place within L3_MAIN interconnect. For more information see Section 7.4.8.7.1.4, *Data Packet Descriptor Word 3*.

7.4.8.6 VPDMA Interrupts

The VPDMA has 4 interrupt group(s). Each group has an interrupt for all the client interrupts, an interrupt for every 32 channels, a interrupt for each list complete, an interrupt for each list notify and an interrupt for for all of the descriptor interrupts. Each of these groups can be individually masked so that only the interrupts specified will trigger the higher level interrupt.

Each interrupt source can be individually masked independently for each seperate interrupt group. A status register bit exists for each interrupt source for for each interrupt group, that is set whenever the interrupt event occurs even when if the interrupt is masked. The status register bit will remain set until cleared by software by writing a one to the status bit.

Table 7-31 shows all interrupt events from VPDMA that go to VIP top level. The interrupt events are mapped to two interrupt lines, INT0 and INT1, that go to VIP top level.

Table 7-31. VPDMA Interrupt Events

Interrupt	Event Flag Registers	Event Mask Registers	Description
vpdma_int_channel_group0	VIP_INT0_CHANNEL0_INT_STAT VIP_INT1_CHANNEL0_INT_STAT	VIP_INT0_CHANNEL0_INT_MASK VIP_INT1_CHANNEL0_INT_MASK	An unmasked channel interrupt for interrupt group 0 in channel register 0 or 1 has fired.
vpdma_int_channel_group1	VIP_INT0_CHANNEL1_INT_STAT VIP_INT1_CHANNEL1_INT_STAT	VIP_INT0_CHANNEL1_INT_MASK VIP_INT1_CHANNEL1_INT_MASK	An unmasked channel interrupt for interrupt group 1 in channel register 0 or 1 has fired.
vpdma_int_channel_group2	VIP_INT0_CHANNEL2_INT_STAT VIP_INT1_CHANNEL2_INT_STAT	VIP_INT0_CHANNEL2_INT_MASK VIP_INT1_CHANNEL2_INT_MASK	An unmasked channel interrupt for interrupt group 2 in channel register 0 or 1 has fired.
vpdma_int_channel_group3	VIP_INT0_CHANNEL3_INT_STAT VIP_INT1_CHANNEL3_INT_STAT	VIP_INT0_CHANNEL3_INT_MASK VIP_INT1_CHANNEL3_INT_MASK	An unmasked channel interrupt for interrupt group 3 in channel register 0 or 1 has fired.
vpdma_int_channel_group4	VIP_INT0_CHANNEL4_INT_STAT VIP_INT1_CHANNEL4_INT_STAT	VIP_INT0_CHANNEL4_INT_MASK VIP_INT1_CHANNEL4_INT_MASK	An unmasked channel interrupt for interrupt group 4 in channel register 0 or 1 has fired.
vpdma_int_channel_group5	VIP_INT0_CHANNEL5_INT_STAT VIP_INT1_CHANNEL5_INT_STAT	VIP_INT0_CHANNEL5_INT_MASK VIP_INT1_CHANNEL5_INT_MASK	An unmasked channel interrupt for interrupt group 5 in channel register 0 or 1 has fired.
vpdma_int_list0_complete			List 0 has completed
vpdma_int_list0_notify			The data transfer in list 0 with the Notify Field set in the descriptor has completed
vpdma_int_list1_complete			List 1 has completed

Table 7-31. VPDMA Interrupt Events (continued)

Interrupt	Event Flag Registers	Event Mask Registers	Description
vpdma_int_list1_notify			The data transfer in list 1 with the Notify Field set in the descriptor has completed
vpdma_int_list2_complete			List 2 has completed
vpdma_int_list2_notify			The data transfer in list 2 with the Notify Field set in the descriptor has completed
vpdma_int_list3_complete			List 3 has completed
vpdma_int_list3_notify			The data transfer in list 3 with the Notify Field set in the descriptor has completed
vpdma_int_list4_complete	VIP_INT0_LIST0_INT_STAT VIP_INT1_LIST0_INT_STAT	VIP_INT0_LIST0_INT_MASK VIP_INT1_LIST0_INT_MASK	List 4 has completed
vpdma_int_list4_notify			The data transfer in list 4 with the Notify Field set in the descriptor has completed
vpdma_int_list5_complete			List 5 has completed
vpdma_int_list5_notify			The data transfer in list 5 with the Notify Field set in the descriptor has completed
vpdma_int_list6_complete			List 6 has completed
vpdma_int_list6_notify			The data transfer in list 6 with the Notify Field set in the descriptor has completed
vpdma_int_list7_complete			List 7 has completed
vpdma_int_list7_notify			The data transfer in list 7 with the Notify Field set in the descriptor has completed
vpdma_int_client	VIP_INT0_CLIENT0_INT_STAT VIP_INT0_CLIENT1_INT_STAT VIP_INT1_CLIENT0_INT_STAT VIP_INT1_CLIENT1_INT_STAT	VIP_INT0_CLIENT0_INT_MASK VIP_INT0_CLIENT1_INT_MASK VIP_INT1_CLIENT0_INT_MASK VIP_INT1_CLIENT1_INT_MASK	Client Interrupt
vpdma_int_descriptor	VIP_INT0_LIST0_INT_STAT VIP_INT1_LIST0_INT_STAT	VIP_INT0_LIST0_INT_STAT VIP_INT1_LIST0_INT_STAT	Descriptor Interrupt

In Table 7-31 above, the “channel_group”, “client” and “descriptor” interrupts are actually a set of additional interrupts. When software receives an interrupt from a “channel_group,” “client,” or “descriptor” it must read the appropriate register within the VPDMA (refer to Table 7-32 to determine what the actual interrupt was).

Table 7-32. VIP Interrupt Sources

Interrupt	Interrupt Group	Description
channel_vip1_mult_anca_src0	channel_group2	The last write DMA transaction has completed for channel vip1_mult_anca_src0. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point.
channel_vip1_mult_anca_src1	channel_group2	The last write DMA transaction has completed for channel vip1_mult_anca_src1. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point.
channel_vip1_mult_anca_src10	channel_group2	The last write DMA transaction has completed for channel vip1_mult_anca_src10. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point.
channel_vip1_mult_anca_src11	channel_group2	The last write DMA transaction has completed for channel vip1_mult_anca_src11. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point.
channel_vip1_mult_anca_src12	channel_group2	The last write DMA transaction has completed for channel vip1_mult_anca_src12. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point.
channel_vip1_mult_anca_src13	channel_group2	The last write DMA transaction has completed for channel vip1_mult_anca_src13. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point.
channel_vip1_mult_anca_src14	channel_group2	The last write DMA transaction has completed for channel vip1_mult_anca_src14. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point.
channel_vip1_mult_anca_src15	channel_group2	The last write DMA transaction has completed for channel vip1_mult_anca_src15. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point.
channel_vip1_mult_anca_src2	channel_group2	The last write DMA transaction has completed for channel vip1_mult_anca_src2. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point.
channel_vip1_mult_anca_src3	channel_group2	The last write DMA transaction has completed for channel vip1_mult_anca_src3. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point.
channel_vip1_mult_anca_src4	channel_group2	The last write DMA transaction has completed for channel vip1_mult_anca_src4. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point.
channel_vip1_mult_anca_src5	channel_group2	The last write DMA transaction has completed for channel vip1_mult_anca_src5. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point.

Table 7-32. VIP Interrupt Sources (continued)

Interrupt	Interrupt Group	Description
channel_vip1_mult_anca_src6	channel_group2	The last write DMA transaction has completed for channel vip1_mult_anca_src6. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point.
channel_vip1_mult_anca_src7	channel_group2	The last write DMA transaction has completed for channel vip1_mult_anca_src7. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point.
channel_vip1_mult_anca_src8	channel_group2	The last write DMA transaction has completed for channel vip1_mult_anca_src8. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point.
channel_vip1_mult_anca_src9	channel_group2	The last write DMA transaction has completed for channel vip1_mult_anca_src9. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point.
channel_vip1_mult_ancb_src0	channel_group2	The last write DMA transaction has completed for channel vip1_mult_ancb_src0. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point.
channel_vip1_mult_ancb_src1	channel_group2	The last write DMA transaction has completed for channel vip1_mult_ancb_src1. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point.
channel_vip1_mult_ancb_src10	channel_group3	The last write DMA transaction has completed for channel vip1_mult_ancb_src10. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point.
channel_vip1_mult_ancb_src11	channel_group3	The last write DMA transaction has completed for channel vip1_mult_ancb_src11. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point.
channel_vip1_mult_ancb_src12	channel_group3	The last write DMA transaction has completed for channel vip1_mult_ancb_src12. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point.
channel_vip1_mult_ancb_src13	channel_group3	The last write DMA transaction has completed for channel vip1_mult_ancb_src13. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point.
channel_vip1_mult_ancb_src14	channel_group3	The last write DMA transaction has completed for channel vip1_mult_ancb_src14. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point.
channel_vip1_mult_ancb_src15	channel_group3	The last write DMA transaction has completed for channel vip1_mult_ancb_src15. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point.

Table 7-32. VIP Interrupt Sources (continued)

Interrupt	Interrupt Group	Description
channel_vip1_mult_ancb_src2	channel_group2	The last write DMA transaction has completed for channel vip1_mult_ancb_src2. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point.
channel_vip1_mult_ancb_src3	channel_group2	The last write DMA transaction has completed for channel vip1_mult_ancb_src3. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point.
channel_vip1_mult_ancb_src4	channel_group2	The last write DMA transaction has completed for channel vip1_mult_ancb_src4. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point.
channel_vip1_mult_ancb_src5	channel_group2	The last write DMA transaction has completed for channel vip1_mult_ancb_src5. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point.
channel_vip1_mult_ancb_src6	channel_group2	The last write DMA transaction has completed for channel vip1_mult_ancb_src6. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point.
channel_vip1_mult_ancb_src7	channel_group2	The last write DMA transaction has completed for channel vip1_mult_ancb_src7. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point.
channel_vip1_mult_ancb_src8	channel_group2	The last write DMA transaction has completed for channel vip1_mult_ancb_src8. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point.
channel_vip1_mult_ancb_src9	channel_group2	The last write DMA transaction has completed for channel vip1_mult_ancb_src9. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point.
channel_vip1_mult_porta_src0	channel_group1	The last write DMA transaction has completed for channel vip1_mult_porta_src0. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point.
channel_vip1_mult_porta_src1	channel_group1	The last write DMA transaction has completed for channel vip1_mult_porta_src1. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point.
channel_vip1_mult_porta_src10	channel_group1	The last write DMA transaction has completed for channel vip1_mult_porta_src10. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point.
channel_vip1_mult_porta_src11	channel_group1	The last write DMA transaction has completed for channel vip1_mult_porta_src11. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point.

Table 7-32. VIP Interrupt Sources (continued)

Interrupt	Interrupt Group	Description
channel_vip1_mult_porta_src12	channel_group1	The last write DMA transaction has completed for channel vip1_mult_porta_src12. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point.
channel_vip1_mult_porta_src13	channel_group1	The last write DMA transaction has completed for channel vip1_mult_porta_src13. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point.
channel_vip1_mult_porta_src14	channel_group1	The last write DMA transaction has completed for channel vip1_mult_porta_src14. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point.
channel_vip1_mult_porta_src15	channel_group1	The last write DMA transaction has completed for channel vip1_mult_porta_src15. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point.
channel_vip1_mult_porta_src2	channel_group1	The last write DMA transaction has completed for channel vip1_mult_porta_src2. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point.
channel_vip1_mult_porta_src3	channel_group1	The last write DMA transaction has completed for channel vip1_mult_porta_src3. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point.
channel_vip1_mult_porta_src4	channel_group1	The last write DMA transaction has completed for channel vip1_mult_porta_src4. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point.
channel_vip1_mult_porta_src5	channel_group1	The last write DMA transaction has completed for channel vip1_mult_porta_src5. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point.
channel_vip1_mult_porta_src6	channel_group1	The last write DMA transaction has completed for channel vip1_mult_porta_src6. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point.
channel_vip1_mult_porta_src7	channel_group1	The last write DMA transaction has completed for channel vip1_mult_porta_src7. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point.
channel_vip1_mult_porta_src8	channel_group1	The last write DMA transaction has completed for channel vip1_mult_porta_src8. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point.
channel_vip1_mult_porta_src9	channel_group1	The last write DMA transaction has completed for channel vip1_mult_porta_src9. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point.

Table 7-32. VIP Interrupt Sources (continued)

Interrupt	Interrupt Group	Description
channel_vip1_mult_portb_src0	channel_group1	The last write DMA transaction has completed for channel vip1_mult_portb_src0. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point.
channel_vip1_mult_portb_src1	channel_group1	The last write DMA transaction has completed for channel vip1_mult_portb_src1. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point.
channel_vip1_mult_portb_src10	channel_group2	The last write DMA transaction has completed for channel vip1_mult_portb_src10. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point.
channel_vip1_mult_portb_src11	channel_group2	The last write DMA transaction has completed for channel vip1_mult_portb_src11. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point.
channel_vip1_mult_portb_src12	channel_group2	The last write DMA transaction has completed for channel vip1_mult_portb_src12. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point.
channel_vip1_mult_portb_src13	channel_group2	The last write DMA transaction has completed for channel vip1_mult_portb_src13. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point.
channel_vip1_mult_portb_src14	channel_group2	The last write DMA transaction has completed for channel vip1_mult_portb_src14. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point.
channel_vip1_mult_portb_src15	channel_group2	The last write DMA transaction has completed for channel vip1_mult_portb_src15. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point.
channel_vip1_mult_portb_src2	channel_group1	The last write DMA transaction has completed for channel vip1_mult_portb_src2. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point.
channel_vip1_mult_portb_src3	channel_group1	The last write DMA transaction has completed for channel vip1_mult_portb_src3. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point.
channel_vip1_mult_portb_src4	channel_group1	The last write DMA transaction has completed for channel vip1_mult_portb_src4. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point.
channel_vip1_mult_portb_src5	channel_group1	The last write DMA transaction has completed for channel vip1_mult_portb_src5. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point.

Table 7-32. VIP Interrupt Sources (continued)

Interrupt	Interrupt Group	Description
channel_vip1_mult_portb_src6	channel_group1	The last write DMA transaction has completed for channel vip1_mult_portb_src6. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point.
channel_vip1_mult_portb_src7	channel_group1	The last write DMA transaction has completed for channel vip1_mult_portb_src7. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point.
channel_vip1_mult_portb_src8	channel_group1	The last write DMA transaction has completed for channel vip1_mult_portb_src8. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point.
channel_vip1_mult_portb_src9	channel_group1	The last write DMA transaction has completed for channel vip1_mult_portb_src9. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point.
channel_vip1_porta_chroma	channel_group3	The last write DMA transaction has completed for channel vip1_porta_chroma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point.
channel_vip1_porta_luma	channel_group3	The last write DMA transaction has completed for channel vip1_porta_luma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point.
channel_vip1_porta_rgb	channel_group3	The last write DMA transaction has completed for channel vip1_porta_rgb. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_up_y then the client will be fully empty at this point.
channel_vip1_portb_chroma	channel_group3	The last write DMA transaction has completed for channel vip1_portb_chroma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point.
channel_vip1_portb_luma	channel_group3	The last write DMA transaction has completed for channel vip1_portb_luma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point.
channel_vip1_portb_rgb	channel_group3	The last write DMA transaction has completed for channel vip1_portb_rgb. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point.
channel_vip2_mult_anca_src0	channel_group4	The last write DMA transaction has completed for channel vip2_mult_anca_src0. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point.
channel_vip2_mult_anca_src1	channel_group4	The last write DMA transaction has completed for channel vip2_mult_anca_src1. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point.

Table 7-32. VIP Interrupt Sources (continued)

Interrupt	Interrupt Group	Description
channel_vip2_mult_anca_src10	channel_group4	The last write DMA transaction has completed for channel vip2_mult_anca_src10. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point.
channel_vip2_mult_anca_src11	channel_group4	The last write DMA transaction has completed for channel vip2_mult_anca_src11. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point.
channel_vip2_mult_anca_src12	channel_group4	The last write DMA transaction has completed for channel vip2_mult_anca_src12. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point.
channel_vip2_mult_anca_src13	channel_group4	The last write DMA transaction has completed for channel vip2_mult_anca_src13. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point.
channel_vip2_mult_anca_src14	channel_group4	The last write DMA transaction has completed for channel vip2_mult_anca_src14. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point.
channel_vip2_mult_anca_src15	channel_group4	The last write DMA transaction has completed for channel vip2_mult_anca_src15. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point.
channel_vip2_mult_anca_src2	channel_group4	The last write DMA transaction has completed for channel vip2_mult_anca_src2. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point.
channel_vip2_mult_anca_src3	channel_group4	The last write DMA transaction has completed for channel vip2_mult_anca_src3. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point.
channel_vip2_mult_anca_src4	channel_group4	The last write DMA transaction has completed for channel vip2_mult_anca_src4. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point.
channel_vip2_mult_anca_src5	channel_group4	The last write DMA transaction has completed for channel vip2_mult_anca_src5. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point.
channel_vip2_mult_anca_src6	channel_group4	The last write DMA transaction has completed for channel vip2_mult_anca_src6. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point.
channel_vip2_mult_anca_src7	channel_group4	The last write DMA transaction has completed for channel vip2_mult_anca_src7. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point.

Table 7-32. VIP Interrupt Sources (continued)

Interrupt	Interrupt Group	Description
channel_vip2_mult_anca_src8	channel_group4	The last write DMA transaction has completed for channel vip2_mult_anca_src8. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point.
channel_vip2_mult_anca_src9	channel_group4	The last write DMA transaction has completed for channel vip2_mult_anca_src9. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point.
channel_vip2_mult_ancb_src0	channel_group4	The last write DMA transaction has completed for channel vip2_mult_ancb_src0. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point.
channel_vip2_mult_ancb_src1	channel_group4	The last write DMA transaction has completed for channel vip2_mult_ancb_src1. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point.
channel_vip2_mult_ancb_src10	channel_group5	The last write DMA transaction has completed for channel vip2_mult_ancb_src10. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point.
channel_vip2_mult_ancb_src11	channel_group5	The last write DMA transaction has completed for channel vip2_mult_ancb_src11. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point.
channel_vip2_mult_ancb_src12	channel_group5	The last write DMA transaction has completed for channel vip2_mult_ancb_src12. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point.
channel_vip2_mult_ancb_src13	channel_group5	The last write DMA transaction has completed for channel vip2_mult_ancb_src13. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point.
channel_vip2_mult_ancb_src14	channel_group5	The last write DMA transaction has completed for channel vip2_mult_ancb_src14. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point.
channel_vip2_mult_ancb_src15	channel_group5	The last write DMA transaction has completed for channel vip2_mult_ancb_src15. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point.
channel_vip2_mult_ancb_src2	channel_group4	The last write DMA transaction has completed for channel vip2_mult_ancb_src2. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point.
channel_vip2_mult_ancb_src3	channel_group4	The last write DMA transaction has completed for channel vip2_mult_ancb_src3. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point.

Table 7-32. VIP Interrupt Sources (continued)

Interrupt	Interrupt Group	Description
channel_vip2_mult_ancb_src4	channel_group5	The last write DMA transaction has completed for channel vip2_mult_ancb_src4. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point.
channel_vip2_mult_ancb_src5	channel_group5	The last write DMA transaction has completed for channel vip2_mult_ancb_src5. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point.
channel_vip2_mult_ancb_src6	channel_group5	The last write DMA transaction has completed for channel vip2_mult_ancb_src6. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point.
channel_vip2_mult_ancb_src7	channel_group5	The last write DMA transaction has completed for channel vip2_mult_ancb_src7. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point.
channel_vip2_mult_ancb_src8	channel_group5	The last write DMA transaction has completed for channel vip2_mult_ancb_src8. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point.
channel_vip2_mult_ancb_src9	channel_group5	The last write DMA transaction has completed for channel vip2_mult_ancb_src9. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point.
channel_vip2_mult_porta_src0	channel_group3	The last write DMA transaction has completed for channel vip2_mult_porta_src0. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point.
channel_vip2_mult_porta_src1	channel_group3	The last write DMA transaction has completed for channel vip2_mult_porta_src1. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point.
channel_vip2_mult_porta_src10	channel_group3	The last write DMA transaction has completed for channel vip2_mult_porta_src10. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point.
channel_vip2_mult_porta_src11	channel_group3	The last write DMA transaction has completed for channel vip2_mult_porta_src11. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point.
channel_vip2_mult_porta_src12	channel_group3	The last write DMA transaction has completed for channel vip2_mult_porta_src12. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point.
channel_vip2_mult_porta_src13	channel_group3	The last write DMA transaction has completed for channel vip2_mult_porta_src13. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point.

Table 7-32. VIP Interrupt Sources (continued)

Interrupt	Interrupt Group	Description
channel_vip2_mult_porta_src14	channel_group3	The last write DMA transaction has completed for channel vip2_mult_porta_src14. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point.
channel_vip2_mult_porta_src15	channel_group3	The last write DMA transaction has completed for channel vip2_mult_porta_src15. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point.
channel_vip2_mult_porta_src2	channel_group3	The last write DMA transaction has completed for channel vip2_mult_porta_src2. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point.
channel_vip2_mult_porta_src3	channel_group3	The last write DMA transaction has completed for channel vip2_mult_porta_src3. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point.
channel_vip2_mult_porta_src4	channel_group3	The last write DMA transaction has completed for channel vip2_mult_porta_src4. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point.
channel_vip2_mult_porta_src5	channel_group3	The last write DMA transaction has completed for channel vip2_mult_porta_src5. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point.
channel_vip2_mult_porta_src6	channel_group3	The last write DMA transaction has completed for channel vip2_mult_porta_src6. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point.
channel_vip2_mult_porta_src7	channel_group3	The last write DMA transaction has completed for channel vip2_mult_porta_src7. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point.
channel_vip2_mult_porta_src8	channel_group3	The last write DMA transaction has completed for channel vip2_mult_porta_src8. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point.
channel_vip2_mult_porta_src9	channel_group3	The last write DMA transaction has completed for channel vip2_mult_porta_src9. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point.
channel_vip2_mult_portb_src0	channel_group3	The last write DMA transaction has completed for channel vip2_mult_portb_src0. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point.
channel_vip2_mult_portb_src1	channel_group3	The last write DMA transaction has completed for channel vip2_mult_portb_src1. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point.

Table 7-32. VIP Interrupt Sources (continued)

Interrupt	Interrupt Group	Description
channel_vip2_mult_portb_src10	channel_group4	The last write DMA transaction has completed for channel vip2_mult_portb_src10. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point.
channel_vip2_mult_portb_src11	channel_group4	The last write DMA transaction has completed for channel vip2_mult_portb_src11. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point.
channel_vip2_mult_portb_src12	channel_group4	The last write DMA transaction has completed for channel vip2_mult_portb_src12. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point.
channel_vip2_mult_portb_src13	channel_group4	The last write DMA transaction has completed for channel vip2_mult_portb_src13. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point.
channel_vip2_mult_portb_src14	channel_group4	The last write DMA transaction has completed for channel vip2_mult_portb_src14. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point.
channel_vip2_mult_portb_src15	channel_group4	The last write DMA transaction has completed for channel vip2_mult_portb_src15. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point.
channel_vip2_mult_portb_src2	channel_group3	The last write DMA transaction has completed for channel vip2_mult_portb_src2. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point.
channel_vip2_mult_portb_src3	channel_group3	The last write DMA transaction has completed for channel vip2_mult_portb_src3. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point.
channel_vip2_mult_portb_src4	channel_group4	The last write DMA transaction has completed for channel vip2_mult_portb_src4. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point.
channel_vip2_mult_portb_src5	channel_group4	The last write DMA transaction has completed for channel vip2_mult_portb_src5. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point.
channel_vip2_mult_portb_src6	channel_group4	The last write DMA transaction has completed for channel vip2_mult_portb_src6. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point.
channel_vip2_mult_portb_src7	channel_group4	The last write DMA transaction has completed for channel vip2_mult_portb_src7. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point.

Table 7-32. VIP Interrupt Sources (continued)

Interrupt	Interrupt Group	Description
channel_vip2_mult_portb_src8	channel_group4	The last write DMA transaction has completed for channel vip2_mult_portb_src8. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point.
channel_vip2_mult_portb_src9	channel_group4	The last write DMA transaction has completed for channel vip2_mult_portb_src9. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point.
channel_vip2_porta_chroma	channel_group5	The last write DMA transaction has completed for channel vip2_porta_chroma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point.
channel_vip2_porta_luma	channel_group5	The last write DMA transaction has completed for channel vip2_porta_luma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point.
channel_vip2_porta_rgb	channel_group5	The last write DMA transaction has completed for channel vip2_porta_rgb. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_up_y then the client will be fully empty at this point.
channel_vip2_portb_chroma	channel_group5	The last write DMA transaction has completed for channel vip2_portb_chroma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point.
channel_vip2_portb_luma	channel_group5	The last write DMA transaction has completed for channel vip2_portb_luma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point.
channel_vip2_portb_rgb	channel_group5	The last write DMA transaction has completed for channel vip2_portb_rgb. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point.
client_vip1_anc_a	client	The client interface vip1_anc_a has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module.
client_vip1_anc_b	client	The client interface vip1_anc_b has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module.
client_vip1_lo_uv	client	The client interface vip1_lo_uv has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module.
client_vip1_lo_y	client	The client interface vip1_lo_y has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module.

Table 7-32. VIP Interrupt Sources (continued)

Interrupt	Interrupt Group	Description
client_vip1_up_uv	client	The client interface vip1_up_uv has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module.
client_vip1_up_y	client	The client interface vip1_up_y has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module.
client_vip2_anc_a	client	The client interface vip2_anc_a has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module.
client_vip2_anc_b	client	The client interface vip2_anc_b has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module.
client_vip2_lo_uv	client	The client interface vip2_lo_uv has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module.
client_vip2_lo_y	client	The client interface vip2_lo_y has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module.
client_vip2_up_uv	client	The client interface vip2_up_uv has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module.
client_vip2_up_y	client	The client interface vip2_up_y has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module.
client_vpi_ctl	client	The client interface vpi_ctl has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module.
control_descriptor_int0	descriptor	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 0.
control_descriptor_int1	descriptor	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 1.
control_descriptor_int10	descriptor	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 10.
control_descriptor_int11	descriptor	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 11.
control_descriptor_int12	descriptor	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 12.
control_descriptor_int13	descriptor	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 13.
control_descriptor_int14	descriptor	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 14.
control_descriptor_int15	descriptor	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 15.

Table 7-32. VIP Interrupt Sources (continued)

Interrupt	Interrupt Group	Description
control_descriptor_int2	descriptor	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 2.
control_descriptor_int3	descriptor	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 3.
control_descriptor_int4	descriptor	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 4.
control_descriptor_int5	descriptor	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 5.
control_descriptor_int6	descriptor	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 6.
control_descriptor_int7	descriptor	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 7.
control_descriptor_int8	descriptor	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 8.
control_descriptor_int9	descriptor	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 9.
list0_complete	list0_complete	List 0 has completed
list0_notify	list0_notify	The data transfer in list 0 with the Notify Field set in the descriptor has completed
list1_complete	list1_complete	List 1 has completed
list1_notify	list1_notify	The data transfer in list 1 with the Notify Field set in the descriptor has completed
list2_complete	list2_complete	List 2 has completed
list2_notify	list2_notify	The data transfer in list 2 with the Notify Field set in the descriptor has completed
list3_complete	list3_complete	List 3 has completed
list3_notify	list3_notify	The data transfer in list 3 with the Notify Field set in the descriptor has completed
list4_complete	list4_complete	List 4 has completed
list4_notify	list4_notify	The data transfer in list 4 with the Notify Field set in the descriptor has completed
list5_complete	list5_complete	List 5 has completed
list5_notify	list5_notify	The data transfer in list 5 with the Notify Field set in the descriptor has completed
list6_complete	list6_complete	List 6 has completed
list6_notify	list6_notify	The data transfer in list 6 with the Notify Field set in the descriptor has completed
list7_complete	list7_complete	List 7 has completed
list7_notify	list7_notify	The data transfer in list 7 with the Notify Field set in the descriptor has completed

7.4.8.7 VPDMA Descriptors

The VPDMA needs to be programmed through descriptors (a pre-defined structure of eight or four 32-bit words depending on type of descriptors) other than VPDMA Memory Mapped Registers (MMR). Descriptors are of three types:

i. **Data Transfer Descriptors** - A memory structure used to describe a desired memory transaction to or from a client.

ii. **Control Descriptors** - A memory structure used to perform a control operation inside the DMA controller

iii. **Configuration Descriptors** - A memory structure used to described a setup that should be applied to an processing modules like MMR write, scalar coefficient write etc.

7.4.8.7.1 Data Transfer Descriptors

In order to set up data transfers from the VPDMA, a data transfer descriptor is added into a list. The fields used for an Inbound and Outbound descriptor vary slightly. An outbound transfer can have two different outbound transfers. The two transfers are the main data transfer and a write of an Inbound Descriptor. The created inbound descriptor is formatted so it can be read back in directly to the next channel specified in the original descriptor.

Figure 7-89. Inbound Data Transfer Descriptor Format

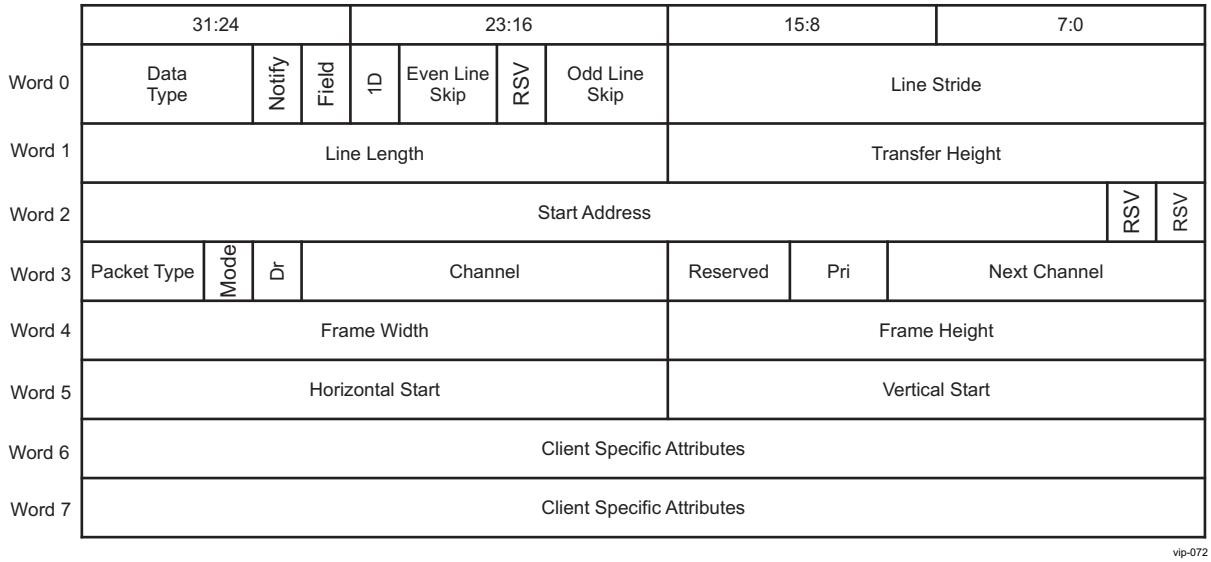
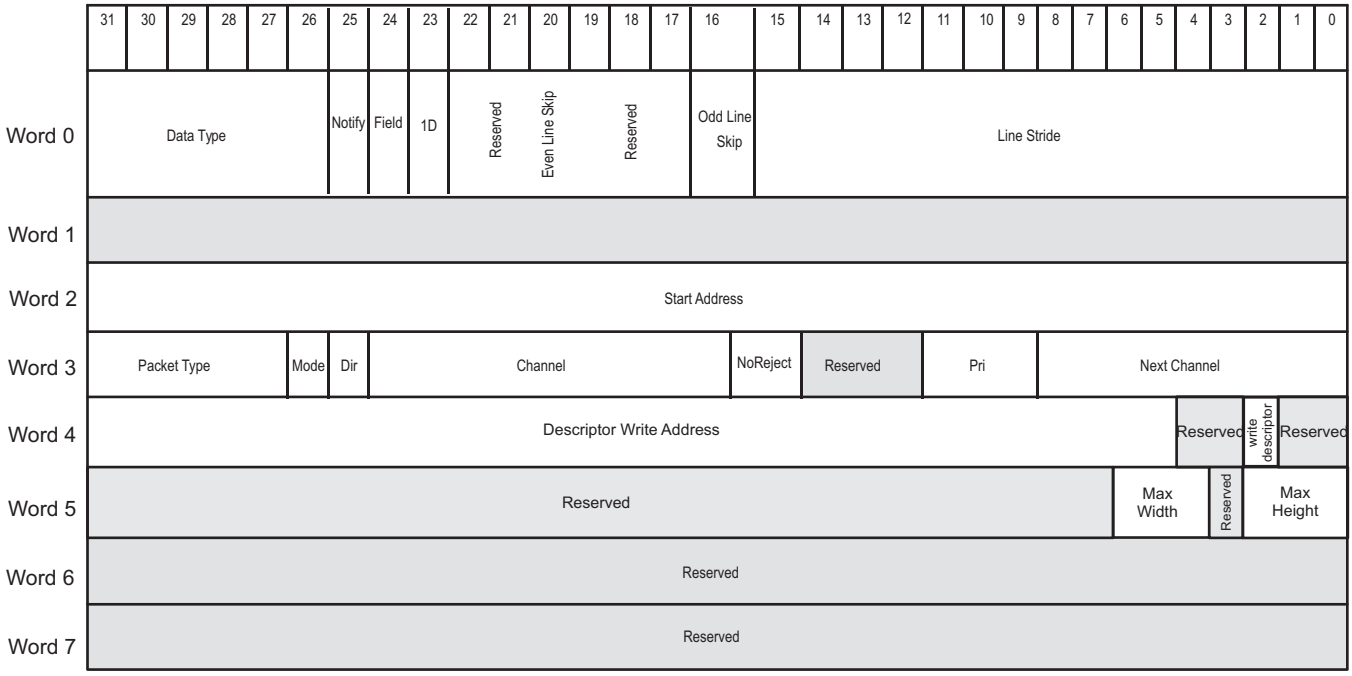


Figure 7-90. Outbound Data Transfer Descriptor Format



The general data transfer Descriptor formats can be seen in the above figures. The descriptor consists of 8 × 32bit words. A Data Transfer Descriptor will be removed from the list when the resource specified by the Channel field is free. If the Channel is not free when the list reaches a data transfer descriptor then the list will stall until the current transfer on the channel has completed.

7.4.8.7.1.1 Data Packet Descriptor Word 0 (Data)

Table 7-33. Data Packet Descriptor Word 0 Field Descriptions

Bit	Field	Value	Description
31:26	Data Type		**Miscellaneous Channel**
			Sets the pixel size in bits plus 1
			RGB Channel
		0	RGB16-565
		1h	ARGB-1555
		2h	ARGB-4444
		3h	RGBA-5551
		4h	RGBA-4444
		5h	ARGB24-6666
		6h	RGB24-888
		7h	ARGB32-8888
		8h	RGBA24-6666
		9h	RGBA32-8888
		10h	BGR16-565
		11h	ABGR-1555
		12h	ABGR-4444
		13h	BGRA-5551
		14h	BGRA-4444
		15h	ABGR24-6666
		16h	BGR24-888
		17h	ABGR32-8888
		18h	BGRA24-6666
		19h	BGRA32-8888
			YUV Channel
		0	Y 4:4:4
		1	Y 4:2:2
		2	Y 4:2:0
		4	C 4:4:4
		5	C 4:2:2
		6	C 4:2:0
		7	CY 4:2:2
		8	YCbC 4:4:4
		14h	Cb 4:4:4
		15h	Cb 4:2:2
		16h	Cb 4:2:0
		17h	CbY 4:2:2
		27h	YC 4:2:2
		37h	YCb 4:2:2
25	Notify	0-1	Send List Notification Interrupt upon last transfer of this channel
24	Field	0-1	Field Value
23	1D	0	The transfer is one dimensional. The transfer and frame sizes are combined to make a single 32 bit transfer size. For writes this value is passed to the generated descriptor. This feature is not supported by all clients. Only clients that support the feature will recognize this bit. **Note:** 1D mode is not supported by VIP modules

Table 7-33. Data Packet Descriptor Word 0 Field Descriptions (continued)

Bit	Field	Value	Description
22:20	Even Line Skip		Field Value
		0	+1 line
		1h	+2 lines
		2h-7h	Reserved
19	Reserved		Reserved for future use
18:16	Odd Line Skip		Field Value
		0	+1 line
		1h	+2 lines
		2h-7h	Reserved
15:0	Line Stride	0-FFFFh	Address stride between lines in bytes

7.4.8.7.1.1.1 Data Type

Bits 31-26 indicate the type of data that is to be transferred. This value is used to compute the number of bytes per pixel so that transactions may be made for the appropriate amount of data. The types of data are dependent on the type of channel. The VPDMA descriptor data types RGB or YUV are defined associated with the VPDMA channel assignment. This helps the engine to distinguish between the overlapping values of the descriptor data types for RGB and YUV data.

- For the Miscellaneous channel, the Data Type selects the size in bits of the data. The range is from 0 to 63 to represent the sizes from 1 to 64 bits.
- For a YUV channel, the Data Type determines, if the data channel is interleaved or color space separate. If color spaced separate, it is still assumed that the two chroma pixels are interleaved.

CAUTION

VPDMA defines the component ordering for its RGB data types in the opposite direction of what commonly used image identifiers expect. To avoid color component swapping in the display and/or in the video/image data written out to the memory, the proper Data Type settings for both RGB and YUV data types must be made. The following paragraphs provide more details on how to set Data Type correctly, in order to match the data stored or expected in the memory.

Setting RGB Data Types

The commonly used RGB format identifiers require the color components to be stored in a little-endian style, where the left most component is the LSB component.

- For an ARGB data type, the A component is the LSB location, as shown in Table 7-34 and Table 7-35;
- For a BGRA data type, the B component would be in the LSB location;

Table 7-34. Common ARGB in Memory (Byte Order)

Addr (LSB)	Addr + 1	Addr + 2	Addr + 3 (MSB)
A	R	G	B

Table 7-35. Common ARGB in 32-bit Memory/CPU Register

Bit 31	Bit 23	Bit 15	Bit 7
B	G	R	A

VPDMA specifies its component ordering in the big-endian style, which requires the data to be stored in the reversed order. Example with ARGB data type is shown in Table 7-36 and Table 7-37. The VPDMA ordering for ARGB data type matches the common BGRA data format.

Table 7-36. VPDMA ARGB in Memory (Byte Order)

Addr (LSB)	Addr + 1	Addr + 2	Addr + 3 (MSB)
B	G	R	A

Table 7-37. VPDMA ARGB in 32-bit Memory/CPU Register

Bit 31	Bit 23	Bit 15	Bit 7
A	R	G	B

In order color components to be mapped correctly and to avoid swapping, the reversal must be taken into consideration when configuring the Data Type in the VPDMA transfer descriptor.

Table 7-38 shows the proper settings required for RGB data types for both storage schemes.

Table 7-38. VPDMA Descriptor RGB Data Type Mapping

Source/Destination Image		VPDMA Data Type Mapping Value	
RGB Component order	Common Image Format Names	**Column A** Source data stored in the VPDMA defined order	**Column B** Source data stored in the opposite of VPDMA defined order
RGB	RGB16-565	0x0	0x10
	ARGB-1555	0x1	0x13
	ARGB-4444	0x2	0x14
	RGBA-5551	0x3	0x11
	RGBA-4444	0x4	0x12
	ARGB24-6666	0x5	0x18
	RGB24-888	0x6	0x16
	ARGB32-8888	0x7	0x19
	RGBA24-6666	0x8	0x15
	RGBA32-8888	0x9	0x17
BGR	BGR16-565	0x10	0x0
	ABGR-1555	0x11	0x3
	ABGR-4444	0x12	0x4
	BGRA-5551	0x13	0x1
	BGRA-4444	0x14	0x2
	ABGR24-6666	0x15	0x8
	BGR24-888	0x16	0x6
	ABGR32-8888	0x17	0x7
	BGRA24-6666	0x18	0x5
	BGRA32-8888	0x19	0x9

In Table 7-38, if the application uses the same data type definition as the VPDMA (that is, RGB24 refers to the B in the LSB), the data types in Column A should be used. But, if the application expects the common data type component order for RGB data type names, the VPDMA data types in Column B should be used.

For example:

- To display an ARGB32-8888 source image data with A in the LSB, the data type in the descriptor should be set to 0x19. But, to display an ARGB32-888 source image data with B in the LSB, the data type in the descriptor should be set to 0x7.
- To capture and write out a RGB24-888 image in the memory with R in the LSB, the data type in the

descriptor should be set to 0x16 (this case assumes the VIP VIN d[23:0] data input is mapped to RGB bus with B component in the LSB).

Setting YUV Data Types

There is no component order reversal for YUV data types. The VPDMA uses generic data type names to specify the memory storage format and the application simply needs to follow the VPDMA defined ordering.

Table 7-39 shows how common YUV data types map to the VPDMA YUV data types in order to clarify the YUV data type configuration.

Table 7-39. VPDMA Descriptor YUV Data Type Mapping

Source YUV Image Types			VPDMA Data Type Mapping (Value)		
Chroma Sub-sample	Common YUV Image Format Type Names	Memory Packed Order [MSB - LSB]	Luma/Chroma Interleaved Channel	Luma-only Channel	Chroma-only Channel
444	YUV	V U Y	YC 4:4:4 (0x8)		
	UVY	Y V U	Cb 4:4:4 (0x14)		
422	NV16 (YUV422SP_UV)	V U		Y 4:2:2 (0x1)	C 4:2:2 (0x5)
	NV16 (YUV422SP_VU)	U V		Y 4:2:2 (0x1)	Cb 4:2:2 (0x15)
	YUV2/YUYV/V422 (YUV422I_YUYV)	V Y U Y	YC 4:2:2 (0x7)		
	YUV422I_YVYU	U Y V Y	CbY 4:2:2 (0x17)		
	Y422/UYVY (YUV422I_UYVY)	Y V Y U	YC 4:2:2 (0x27)		
	YUV422I_VYUY	Y U Y V	YCb 4:2:2 (0x37)		
420	NV12 (YUV420SP_UV)	V U		Y 4:2:0 (0x2) YC 4:2:2 (0x7) (see (1))	C 4:2:0 (0x6) YC 4:2:2 (0x7) (see (1))
	NV21 (YUV420SP_VU)	U V		Y 4:2:0 (0x2) YC 4:2:2 (0x7) (see (1))	Cb 4:2:0 (0x16) YC 4:2:2 (0x7) (see (1))

(1) If 422 source data is used, unused component data fetched (either Luma or Chroma) will be discarded.

For further details on the data formats, refer to Section 7.4.8.9, *VPDMA Data Formats*.

7.4.8.7.1.1.2 Notify

The Notify bit is used in conjunction with the Notify interrupts and when the channel has completed the DMA transfer the Notify Interrupt for the list that contains the descriptor will fire. The last Notify bit set for a specific list will be used so only a single Notify should be set in a list.

7.4.8.7.1.1.3 Field

The Field bit is the field value of the data that will be passed down to the clients. If the data is interlaced, then this value should be cleared to 0.

7.4.8.7.1.1.4 Even Line Skip

The Even Line skip is used with Line Stride to generate the next line address on an even line. All frames start on line 0. This value allows for the DMA controller to skip lines for interlaced data in a progressive frame buffer.

7.4.8.7.1.1.5 Odd Line Skip

The Odd Line skip is used with Line Stride to generate the next line address on an odd line. All frames start on line 0, so this will apply starting with the second line. This value allows for the DMA controller to skip lines for interlaced data in a progressive frame buffer.

7.4.8.7.1.1.6 Line Stride

Bits 15:0 are the stride between lines in bytes at the external address. This value is added or subtracted based upon an adjustment using the current skip value. Operation of the external address pointer shall load the Source Address upon start of the transfer, then at the end of each line increment or decrement by the value computed using the Line Stride and Skip value for the line. The line stride must be aligned to an L3 data bus width. The lower bits of the stride will always be treated as zero to force the alignment.

7.4.8.7.1.2 Data Packet Descriptor Word 1

Table 7-40. Data Packet Descriptor Word 1 Field Description

Bits	Name	Description
31:16	Line Length	Line Length in Pixels
15:0	Transfer Height	Number of rows in transfer.

7.4.8.7.1.2.1 Line Length

Bits 31-16 are the line length in pixels of the current channel. This is ignored for the outbound transfer as the client will provide the line length with the end of line signal on the client interface. The maximum supported line length is currently 4096.

7.4.8.7.1.2.2 Transfer Height

Bits 15-0 are the number of lines to be transferred in the current channel. This is ignored for outbound transfers as the client will provide the line length with the end of frame signal on the client interface. The maximum supported transfer size is currently 2048.

7.4.8.7.1.3 Data Packet Descriptor Word 2

Table 7-41. Data Packet Descriptor Word 2 Field Descriptions

Bit	Field	Value	Description
31:0	Start Address		32-bit data source address [31:0]
			If Mode is TILED, then TILER specific ADDRESS Map is used:
			Bits 31-29:
		0	0-degree view
		1h	180-degree view + mirroring
		2h	0-degree view + mirroring
		3h	180-degree view
		4h	270-degree view + mirroring
		5h	270-degree view
		6h	90-degree view
		7h	90-degree view + mirroring
			Bits 28-27:
		0	8-bit container
		1h	16-bit container
		2h	32-bit container
		3h	Page Mode
			If Mode is NORMAL, then bits 31-26 are the upper bits of the address.

7.4.8.7.1.3.1 Start Address

This is the byte aligned address for the first data transfer. The address on the OCP bus will always be word aligned.

7.4.8.7.1.4 Data Packet Descriptor Word 3

Table 7-42. Data Packet Descriptor Word 3 Field Descriptions

Bits	Name	Description
31:27	Packet Type	Host Packet Descriptor Type = 0xa
26	Mode	0= Normal, 1=TILED
25	Direction	Inbound = 0, Outbound = 1
24:16	Channel	Channel for which this descriptor describes
15	Reserved	Reserved for future use
11:9	Priority	Only Bit 9 and Bit 11 are used to set the priority. Bit 10 is ignored. Highest = 0, Lowest = 3 By default, hardware assigns priority = 3. This priority level is used in the arbitration between the masters (for DDR access). See Section 7.4.8.7.1.4.5, *Priority*, for more details.
8:0	Next Channel	Next Channel to execute on a line or the channel to use in the generated write descriptor.

7.4.8.7.1.4.1 Packet Type

Bits 31:27 are a unique code which indicates that this Descriptor is a VPDMA descriptor.

7.4.8.7.1.4.2 Mode

Bit 26 is used to indicate if the transfer is to regular memory space or to Tiled memory space. The VPDMA will use this to determine if it does standard raster based addressing or if it assumes that the data is stored in TILER format. If data is stored in TILER format then the buffer is turned into 2 or 4 line buffers depending on the container type. For shared clients that use the memory, such as the ancillary data and the VIP port, only one can be active, if the mode field is set. Only clients that support Tiling will properly pack the data for tiling on the output interface. This must only be set for channels going to clients that support the TILING feature in the client configuration.

7.4.8.7.1.4.3 Direction

Bit 25 is used to indicate the direction of transfer. This bit indicates that the data flow is from an external source to an internal buffer (inbound) or data transfers form an internal buffer to an external location (outbound).

7.4.8.7.1.4.4 Channel

Bits 24:16 are the Channels which is supported by the descriptor. This is the identification of the specific Channel which is controlled by the contents of the descriptor. The channel assignments can be found in the channel table.

7.4.8.7.1.4.5 Priority

Bits 11:9 are set to indicate priority of the transfer, these are directly mapped to the OCP reqinfo bits.

7.4.8.7.1.4.6 Next Channel

Bits 8:0 give the next channel to use to create a composite frame. The next channel must be to a free channel. The last channel of a row should point back to the initial channel which must be a channel tied directly to a client. The Descriptor for the Next Channel must be of the same type as the current descriptor.

7.4.8.7.1.5 Data Packet Descriptor Word 4

7.4.8.7.1.5.1 Inbound data

Table 7-43. Data Packet Descriptor Word 4 Inbound Data Field Descriptions

Bits	Name	Description
31:16	Frame Width	Width of the client frame.
15:0	Frame Height	Height of the client frame

7.4.8.7.1.5.1.1 Frame Width

Bits 31:16 indicate the width in pixels of the frame. This is the width of the entire frame and not just the width of the data represented by the current channel unless this channel fills the entire frame. This allows for the VPDMA to present a larger frame of data to the client while only fetching the required data. The currently supported maximum width is 4096.

7.4.8.7.1.5.1.2 Frame Height

Bits 15:0 indicate the height in pixels of the entire frame. This is the height of the entire frame and not just the height of the data represented by the current channel unless this channel fills the entire frame. This allows for the VPDMA to present a larger frame of data to the client while only fetching the required data. The currently supported maximum height is 2048.

7.4.8.7.1.5.2 Outbound data

Table 7-44. Data Packet Descriptor Word 4 Outbound Data Field Descriptions

Bits	Name	Description
31:5	Descriptor Write Address	The 32 byte aligned location to write an inbound descriptor
2	Write Descriptor	If set to 1, a descriptor will be generated when the client completes a frame.
1	Drop Data	If set to 1, the data will not be written out. Also, if this is set, the write descriptor bit must be set. This allows for descriptors to only be written out so that software can determine the size of the required buffer before data is written out.
0	Use Descriptor Register	If set to 1, the CURRENT_DESCRIPTOR register will be used for the write address location. If set to 0, the Descriptor Write Address field in this word will be used for the Descriptor Write Address.

7.4.8.7.1.5.2.1 Descriptor Write Address

Bits 31:5 set the 32 byte address to write the generated descriptor. This address is only used if the Write Descriptor bit is set to 1 and the Use Descriptor Register bit is set to 0. If this case is met when the channel is complete a descriptor that meets the inbound descriptor format will be written to the location specified by this field. This allows for software to have a specific channel write its descriptor to a specific address no matter what order the channel completes compared to other outbound channels.

7.4.8.7.1.5.2.2 Write Descriptor

Bit 2 determines if a descriptor should be written out when the client is completed. If this bit is set the descriptor will be written. The format of the descriptor will be of an Inbound Data Transfer descriptor with the LINE LENGTH and TRANSFER HEIGHT fields determined by the counters in the clients. This means that the direction bit will be the opposite of the inbound descriptor. The FRAME WIDTH and FRAME HEIGHT values will match the LINE LENGTH and TRANSFER HEIGHT fields. The CHANNEL and NEXT CHANNEL fields will match the NEXT CHANNEL field of the original descriptor. The FIELD bit will match the source field captured by the client. The NOTIFY bit will always be 0. All other fields will match the original outbound descriptor. If the Outbound descriptor MODE is set to TILED data then the descriptor address used will be in TILED data space and software must ensure that the address is in page mode for the address of the descriptor.

7.4.8.7.1.5.2.3 Drop Data

Bit 1 determines if the data should be written out or not. In some cases software might only want to write the descriptor out to determine the size of the buffer that will be required to store incoming data. By setting this bit,no data will be written out. If this bit is set then the Write Descriptor bit MUST be set. Bit 0 determines where the descriptor should be written.

7.4.8.7.1.6 Data Packet Descriptor Word 5

7.4.8.7.1.6.1 Outbound data

Width and Height are set in the following register bit-fields:

- For Max_Size1: VIP_MAX_SIZE1[31:16] MAX_WIDTH and VIP_MAX_SIZE1[15:0] MAX_HEIGHT registers
- For Max_Size2: VIP_MAX_SIZE2[31:16] MAX_WIDTH and VIP_MAX_SIZE2[15:0] MAX_HEIGHT registers
- For Max_Size3: VIP_MAX_SIZE3[31:16] MAX_WIDTH and VIP_MAX_SIZE3[15:0] MAX_HEIGHT registers

Table 7-45. Data Packet Descriptor Word 5 Outbound Data Field Descriptions

Bits	Name	Description
6:4	Max Width	The maximum allowable pixels per line. 0: Unlimited Line Size
		1: Use Max_Size1 Max Width field
		2: Use Max_Size2 Max Width field
		3: Use Max_Size3 Max Width field
		4: 352 pixels
		5: 768 pixels
		6: 1280 pixels
		7: 1920 pixels
		Others: Reserved
2:0	Max Height	The maximum allowable lines per frame. 0: Unlimited Frame Size
		1: Use Max_Size1 Max Height field
		2: Use Max_Size2 Max Height field
		3: Use Max_Size3 Max Height field
		4: 288 lines
		5: 576 lines
		6: 720 lines
		7: 1080 lines
		Others: Reserved

7.4.8.7.1.6.1.1 Max Width

Bits 6:4 are encoded to set the maximum transferred line size. If the field is not set to unlimited if the client sending data into the VPDMA exceeds the maximum allowed pixels the data will continue to be received from the client but will not be sent to external memory until an end of line is received from the client sending data. The outbound descriptor if created will still have the transmitted size of the last line of the frame which will match the max width. If the frame does not exceed the max width then the actual transmitted size will be placed in the descriptor. Tiled clients such as the noise filter should always set this to 0. If Max Width is 0 and the line size is larger then 4096 pixels then the address counters will overflow and the data at the start of the line will be overwritten.

7.4.8.7.1.6.1.2 Max Height

Bits 2:0 are encoded to set the maximum transferred frame size. If the field is not set to unlimited if the client sending data into the VPDMA exceeds the maximum allowed lines the data will continue to be received from the client but will not be sent to external memory until an end of frame is received from the client sending data. The outbound descriptor if created will still have the transmitted number of lines received which would match the max height. If the frame does not exceed the max height then the actual transmitted size will be placed in the descriptor. Tiled clients such as the noise filter should always set this to 0.

7.4.8.7.2 Configuration Descriptor

The Configuration Descriptor is used in a List to setup a client configuration. The configuration descriptor consists of a header and a payload portion. The payload can either be part of the list that the descriptor is contained in or it can be at a separate location. The VPDMA will pass the payload of the configuration descriptor down to the client over the VPI Control port interface or through the MMR configuration port depending on the destination field. A Configuration Descriptor to any single destination except Destination 0 must be on a single list. Configuration Descriptors to different destinations may be on different lists..

The Configuration Descriptor Header is 4 × 32 bit words. A configuration descriptor is not consumed until the destination specified has received the entire configuration payload. Therefore the list is expected to stall while the configuration takes place. This ensures that all descriptors after a configuration descriptor in the list will occur after the desired configuration.

7.4.8.7.2.1 Configuration Descriptor Header Word0

Table 7-46. Configuration Descriptor Header Word0 Field Descriptions

Bits	Name	Description
31:0	Address Offset of the Destination	This is the address offset location in the destination that the block of data in the payload should be written. This field is only used if the descriptor Class is a block type. Otherwise this field is reserved.

7.4.8.7.2.2 Configuration Descriptor Header Word1

Table 7-47. Configuration Descriptor Header Word1 Field Descriptions

Bits	Name	Description
15:0	Number of Data Words	Length of First Data Packet for Class 1(block).

7.4.8.7.2.2.1 Number of Data Words

Bits 15:0 indicate the length of the first data block if the class is 1 as specified in Configuration Descriptor Header Word3. If the class is not 1 then this field should be set to 0.

7.4.8.7.2.3 Configuration Descriptor Header Word2

Table 7-48. Configuration Descriptor Header Word2 Field Descriptions

Bits	Name	Description
31:0	Payload Location	Pointer to the data payload

7.4.8.7.2.3.1 Payload Location

Bits 31:0 contain the pointer to the data payload if the command packet is an indirect type. This value along with the Payload Length will be combined to issue a DMA transaction that must complete before the List Manager can finish processing this descriptor. The list will be made inactive pending this DMA completion. This address should be on a 16 byte boundary so the lower 4 bits should always be 0.

7.4.8.7.2.4 Configuration Descriptor Header Word3

Table 7-49. Configuration Descriptor Header Word3 Field Descriptions

Bits	Name	Description
31:27	Packet Type	Host Packet Descriptor Type = 0xb
26	Direct	Direct Command = 1 Indirect Command = 0
25:24	Class	0 = Address, Data Set 1 = Block 2,3 Reserved for Future Use
23:16	Destination	Destination of the configuration payload
15:0	Payload Length	Length of Payload in Words.

7.4.8.7.2.4.1 Packet Type

Bits 31:27, this field indicates the type of descriptor and should be set 0xB for configuration descriptor.

7.4.8.7.2.4.2 Direct

Bit 26, this field indicates that the descriptor is a direct command or indirect command. A direct command means that the payload is contiguous with the descriptor and will be pulled in by the list manager descriptor control as it processes the list. A direct payload must end on or before a 256 byte boundary in the list. An indirect command is when the Address is located in Word2 is a pointer to the data payload. A DMA transaction will be scheduled to bring the payload into the List Manager to allow for the payload to be sent to the destination.

7.4.8.7.2.4.3 Class

Bits 25:24, this field indicates the type of payload is associated with command descriptor, and thus how to handle the payload.

A 0 indicates that the payload consists of blocks of data with each block having an address and length. The first word after a sub-block contains the next address and length in bytes. This allows for writing to multiple configuration regions in a client. The Configuration Descriptor word1 is the first address and length pair. All addresses used must be larger then the previous address for all destinations except for the MMR destination. If a sub block length is not evenly divisible into 16 then zeroes should be padded in the payload so that the Next Client Address and Length field start on a 16 byte boundary.

7.4.8.7.2.4.3.1 Address Data Block Format

Table 7-50. Address Data Block Format Field Descriptions

Bits	Name	Description
31-0		Next Client Address
31-0		Configuration for Next Client Address
31-0		Configuration for Next Client Address + 4
31-0		Configuration for Next Client Address + 8
31-0		Configuration for Next Client Address + 12
31-0		Configuration for Next Client Address + 16
31-0		Next Client Address 2
15-0		Sub Block Length

A 1 indicates the payload is simply block data. The data is contiguous and starts at the offset of the destination as specified in word1.

7.4.8.7.2.4.4 Destination

The Destination field is used to determine where the configuration payload should be sent. The values for the destination field can be seen in the table below.

Table 7-51. Destination Field Description

Destination Value	Actual Destination	Description
0	mmr_client	Write to MMR registers to setup other modules
7	VIP Slice 0	VIP Slice 0 Scaler Coefficient Tables
8	VIP Slice 1	VIP Slice 1 Scaler Coefficient Tables

7.4.8.7.2.4.5 Descriptor Length

Bits 15:0, this field indicates the size of the payload in words for the command. This is 128 bit words. The maximum number of words is 1023 words in a single payload

7.4.8.7.3 Control Descriptor

7.4.8.7.3.1 Generic Control Descriptor Format

A control descriptor is the mechanism that is used in a list to give commands to the List Manager. These descriptors are not considered consumed until the List Manager has done the instruction required by the descriptor. All control descriptors have a common Header located at Word 3 but the remaining words are based on the specific control descriptor.

7.4.8.7.3.2 Control Descriptor Header Description

Table 7-52. Control Descriptor Header Description

Bits	Name	Description
31:27	Packet Type	Host Packet Descriptor Type = 0xc
26:25	Reserved	Reserved
24:16	Source	Specifies the source used for the control event.
15:4	Reserved	Reserved for future use
3:0	Control	The type of control descriptor that should be run by the List Manager

7.4.8.7.3.2.1 Packet Type

This field indicates a VPDMA control descriptor.

7.4.8.7.3.2.2 Source

The source is combined with the control field to determine what the control descriptor should use as the source for its synchronization event.

7.4.8.7.3.2.3 Control

The Control field defines the specific function of the descriptor. Table 7-53 lists the different control descriptors.

7.4.8.7.3.3 Control Descriptor Types

Table 7-53. Control Descriptor Types Summary

Control Descriptor	Control Field Value	Description
Sync on Client	0	Wait for the client attached to the channel specified to reach a certain event before proceeding.
Sync on List	1h	Wait for another list(s) to reach its Sync on List Descriptor
Sync on External Event	2h	Wait for a Register write to the LIST_STAT_SYNC bit specified or for an external event.
Sync on Channel	4h	Wait for the channel specified to complete
Change Client Interrupt	5h	Change the interrupt event for a client interrupt but do not wait for the event to occur.
Send Interrupt	6h	Generate an interrupt event on one of the Control Descriptor Interrupts
Reload List	7h	Reload the list from the location and size specified.
Abort Channel	8h	Abort the transfer in the channel specified.

7.4.8.7.3.3.1 Sync on Client

A Sync on Client Control descriptor uses the source field to select a channel to modify the attached clients interrupt generation event. For a client that supports multiple channels then only an event on the portion of the client that supports that client will cause the interrupt to be generation. After configuring the interrupt generation event the list will then stall until that event has occurred.

Table 7-54. Sync on Client Field Descriptions (Word - 1)

Bits	Name	Description
31:16	PIXEL_COUNT	Specify the pixel position on a line specified by LINE_COUNT where a pixel based event would occur.
15:0	LINE_COUNT	Specify the line where a line based event would trigger

Table 7-55. Sync on Client Field Descriptions (Word - 3)

Bits	Name	Description
31:27	Packet Type	Host Packet Descriptor Type = Ch
26:24	Reserved	Reserved
23:16	Source	Specifies the source used for the control event.
15:4	Reserved	Reserved for future use
3:0	Control	Control type = 0

7.4.8.7.3.3.2 Sync on List

The Sync on List Control descriptor is a mechanism to ensure that multiple lists have all reached a common point. The Sync on List descriptor uses the Source field as a mask for each list that should be synchronized which must include the list where the control descriptor resides. Once all lists specified in the source field have reached their Sync on List descriptor all lists will resume with the lowest list number resuming first. All Sync on List Control descriptors in each of the lists must have the same source field so that all lists will synchronize upon reaching that point.

If it is desired to synchronize list 0 and list 1 then the source field would be 0x3. If it is desired to synchronize list 1, list 3, and list 4 then the source field would be 0x1a. Word 0, Word 1 and Word 2 are reserved.

Table 7-56. Sync on List Field Descriptions (Word - 3)

Bits	Name	Description
31:27	Packet Type	Host Packet Descriptor Type = Ch
26:24	Reserved	Reserved
23:16	Source	Specifies the source used for the control event.
15:4	Reserved	Reserved for future use
3:0	Control	Control type = 1h

7.4.8.7.3.3.3 Sync on External Event

A Sync on External Event descriptor ensures an external event has occurred before proceeding. The external event can be a write to a bit of the LIST_STAT_SYNC register or other external events that are determined at VPDMA elaboration to have occurred. This descriptor can be used to allow for external software to control progression of the list. The descriptor can also be used to select external signals that are brought into the VPDMA. The current implementation just synchronize on the LIST_STAT_SYNC bit for the list number that called the descriptor.

Table 7-57. Sync on External Event Field Descriptions (Word - 3)

Bits	Name	Description
31:27	Packet Type	Host Packet Descriptor Type = Ch
26:4	Reserved	Reserved
3:0	Control	Control type = 2h

7.4.8.7.3.3.4 Sync on Channel

A Sync on Channel descriptor stalls the list until the channel specified is free. If the channel is already free then the descriptor will not cause the list to stall. Word 0, Word 1 and Word 2 are reserved.

Table 7-58. Sync on Channel Field Descriptions (Word - 3)

Bits	Name	Description
31:27	Packet Type	Host Packet Descriptor Type = Ch
26:24	Reserved	Reserved
23:16	Source	Specifies the source used for the control event.
15:4	Reserved	Reserved for future use
3:0	Control	Control type = 4h

7.4.8.7.3.3.5 Sync on LM Timer

A Sync on LM Timer descriptor sets a value from the current timer position to wait. The LM timer is a free running counter at the LM processing clock. The Timer Value in the descriptor is added to the value of the timer at the time the descriptor is received and the list will stall for this many cycles before it becomes active again.

7.4.8.7.3.3.6 Change Client Interrupt

A Change Channel Interrupt Source descriptor uses the source field to select a channel to modify the attached clients interrupt generation event. The list will not stall on this descriptor. The format of the fields is identical to the Sync on Client Descriptor. Word 0 is reserved.

Table 7-59. Change Client Interrupt Field Descriptions (Word - 1)

Bits	Name	Description
31:16	PIXEL_COUNT	Specify the pixel position on a line specified by LINE_COUNT where a pixel based event would occur.
15:0	LINE_COUNT	Specify the line where a line based event would trigger.

Table 7-60. Change Client Interrupt Field Descriptions (Word - 2)

Bits	Name	Description
31:4	Reserved	Reserved
3:0	Event	Specify the event which should trigger the client interrupt.

Table 7-61. Change Client Interrupt Field Descriptions (Word - 3)

Bits	Name	Description
31:27	Packet Type	Host Packet Descriptor Type = Ch
26:24	Reserved	Reserved

Table 7-61. Change Client Interrupt Field Descriptions (Word - 3) (continued)

Bits	Name	Description
23:16	Source	Specifies the source used for the control event.
15:4	Reserved	Reserved for future use
3:0	Control	Control type = 5h

7.4.8.7.3.3.7 Send Interrupt

A Send Interrupt descriptor will cause the VPDMA to generate an interrupt on the list manager controlled interrupts as specified by the Source Field. The list will not stall on this descriptor. For example, if source is 0 then control_descriptor_int0 will fire. If source is 12, then control_descriptor_int12 will fire. For more information ot VPDMA interrupt events, see Section 7.4.8.6, *VPDMA Interrupts*.

Table 7-62. Send Interrupt Field Descriptions (Word - 3)

Bits	Name	Description
31:27	Packet Type	Host Packet Descriptor Type = Ch
26:24	Reserved	Reserved
23:16	Source	Specifies the source used for the control event.
15:4	Reserved	Reserved for future use
3:0	Control	Control type = 6h

7.4.8.7.3.3.8 Reload List

A Reload List descriptor causes ending descriptors after this descriptor in the original list to be dropped and a new list at the location and of the size specified in the descriptor. This descriptor can be used to allow for linked lists in the VPDMA as the list will continue from this point and fetch the list specified and it does not have to be continuous with the current list.

Table 7-63. Reload List Field Descriptions (Word - 0)

Bits	Name	Description
31:0	LIST_ADDRESS	31 most-significant Bits of the memory address where the descriptors to be loaded are stored. Address must be 16 byte aligned.

Table 7-64. Reload List Field Descriptions (Word - 1)

Bits	Name	Description
31:16	Reserved	Reserved
15:0	LIST_SIZE	Size of the list to load

Table 7-65. Reload List Field Descriptions (Word - 3)

Bits	Name	Description
31:27	Packet Type	Host Packet Descriptor Type = Ch
26:4	Reserved	Reserved
3:0	Control	Control type = 7h

7.4.8.7.3.3.9 Abort Channel

An Abort Channel descriptor is used to clear a channel. This clears the channel from issuing any more requests. Any outstanding requests for that channel will complete as originally scheduled. All data inside the client will be flushed. For Tiled Clients such as the noise filter it is required to issue two consecutive abort channel descriptors to ensure the channel is aborted for both the current tile and next tile. Word 0, Word 1 and Word 2 are reserved.

Table 7-66. Abort Channel Field Descriptions (Word - 3)

Bit	Name	Description
31:27	Packet Type	Host Packet Descriptor type = 0xC
26:24	Reserved	Reserved
23:16	Source	VPDMA Channel Number whose transfers are to be aborted
15:4	Reserved	Reserved
3:0	Control	Control type = 9h

7.4.8.8 VPDMA Configuration

The following section describes the different ways of configuring VPDMA for data transfers.

7.4.8.8.1 Regular List

A regular list executes each descriptor in order until the end of the list is reached. When the end of the list is reached an interrupt is sent and the list can be reused by software. A regular list can contain any descriptor types. Software creates the list at some location in external memory. After completing writing the list software then writes the location of the list to the VIP_LIST_ADDR[31:0] VIP_LIST_ADDR register and then writes the VIP_LIST_ATTR register. If the NUMBER in the VIP_LIST_ATTR[26:24] LIST_NUM is not an active list then the List will be loaded and begin to execute the next time the List Manager gets to IDLE after processing previous loaded lists. If the NUMBER in the VIP_LIST_ATTR[26:24] LIST_NUM is busy then the VIP_LIST_ADDR and VIP_LIST_ATTR registers will be locked until the active list specified by NUMBER completes.

The different ports inside VPDMA requires different list setup, as explained in the following sections.

7.4.8.8.2 Video Input Ports

The Video Input Ports can be used in multiple ways. Depending on how the input ports are configured will determine how the lists to manage them need to be setup. The ways in which the ports can work are multiplexed data stream, single YUV color separate stream, dual YUV interleaved or single RGB stream. Each port also has an ancillary data port that can run either multiplexed or single data stream and shares the buffering with the main data stream. For all cases the descriptor must be loaded into the client before the vertical sync is received on the VIP port or the entire frame will be dropped. As with all write clients the VIP channels can be shadowed so the next frame/field descriptor can be loaded while the previous frame is running without stalling the list.

7.4.8.8.2.1 Multiplexed Data Streams

In the case of a multiplexed data stream input the channels that should be used are VIP(X)_MULT_PORT(Y)_SRC(Z). Where X is the specific VIP slice of the instance that wants to be used and port Y is the port A or port B that is receiving the data. Finally Z is the channel number. For Split line mux mode the LSB of the channel will determine, if the line is a split line or a complete line. This is required so that the data streams do not get mixed when a channel ends without completing a line. In this mode the data will always be sent out as 422 Interleaved data to the destination specified in the descriptor.

7.4.8.8.2.2 Single YUV Color Separate

If the port is configured to be used as sending out color separated YUV data then the channels that must be used are VIPX_PORTY_LUMA and VIPX_PORTY_CHROMA for the luma and chroma components respectively. The data type can be either 420 or 422 color separate in this case.

7.4.8.8.2.3 Dual YUV Interleaved

If the port is configured to be used to send out 422 interleaved data from a non-multiplexed source then the channels that must be used are VIPX_PORTY_LUMA or VIPX_PORTY_CHROMA depending on if the data stream is connected to the luma or chroma port. The data type must be 422 interleaved in this case.

7.4.8.9 VPDMA Data Formats

Each of the channels has a type of data that it can support based upon the client that it services. Also for each channel, the data type will assume that data is packed in memory a certain way and will be prevented to the client in the same manner to the client no matter what the format of the data in memory.

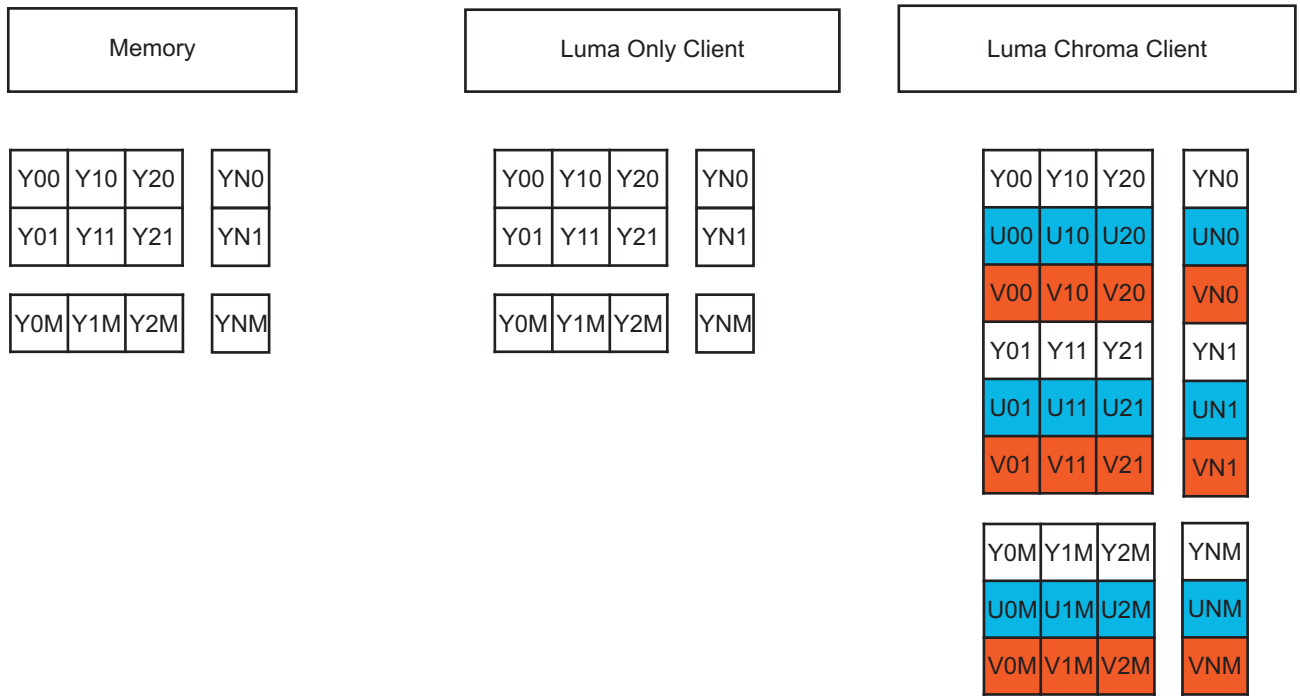
7.4.8.9.1 YUV Data Formats

The YUV formatted channels expect video data that is in YUV color space. The YUV data can be type is for YUV data and it can support both interleaved data where Luma and Chroma are in the same data buffer or it can support co-planar data where the Luma and Chroma are in separate data buffers. The YUV channel also can be given a position to give a different start position for the client instead of the default upper left hand corner of the frame. The storage format for YUV data depends on the data type field. The data type must be set to a data type that the channel can support. All the clients that data are provided too will either accept Luma Only, Chroma Only or Luma and Chroma in a parallel data bus.

7.4.8.9.1.1 Y 4:4:4 (Data Type 0)

The Y 4:4:4 data type is used for a co-planar 4:4:4 data. In this mode the data is assumed to be a single byte with each pixel is packed in a line. If a 2D transfer of the data is used the VPDMA will assume this data type is stored in an 8 bit container. This data block should have the width and height set to the desired frame size expected by the receiving client.

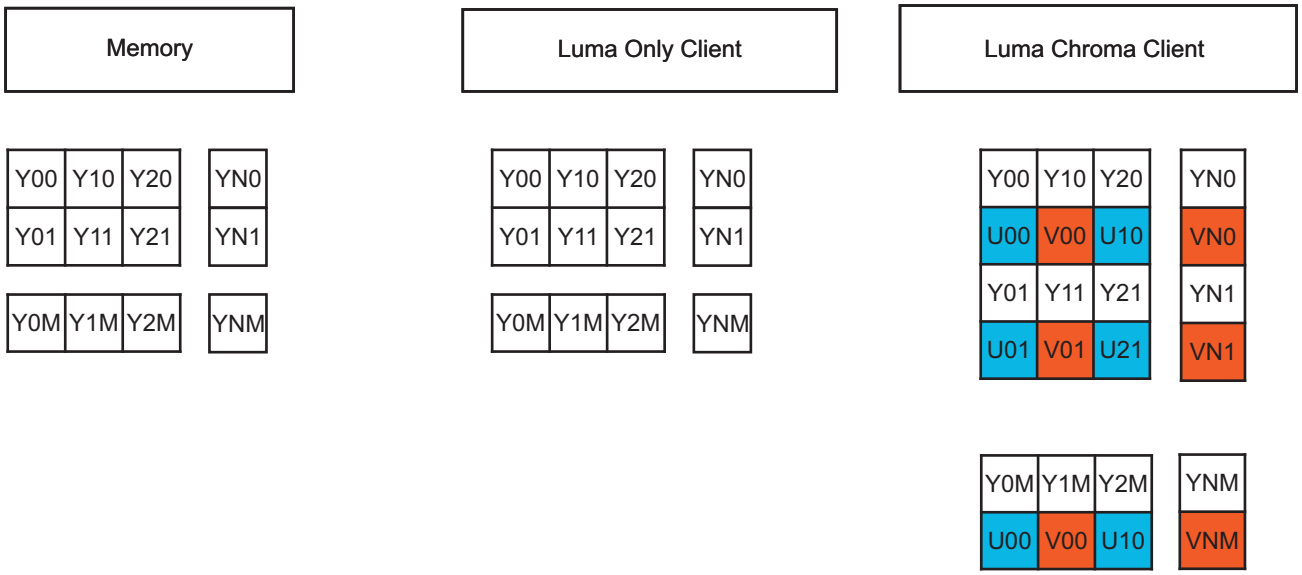
Figure 7-91. Y 4:4:4 (Data Type 0)



7.4.8.9.1.2 Y 4:2:2 (Data Type 1)

The Y 4:2:2 data type is used for a co-planar 4:2:2 data. In this mode the data is assumed to be a single byte with each pixel is packed in a line. If a 2D transfer of the data is used the VPDMA will assume this data type is stored in an 8 bit container. This data block should have the width and the height of the desired frame sent by the VPDMA to the receive client.

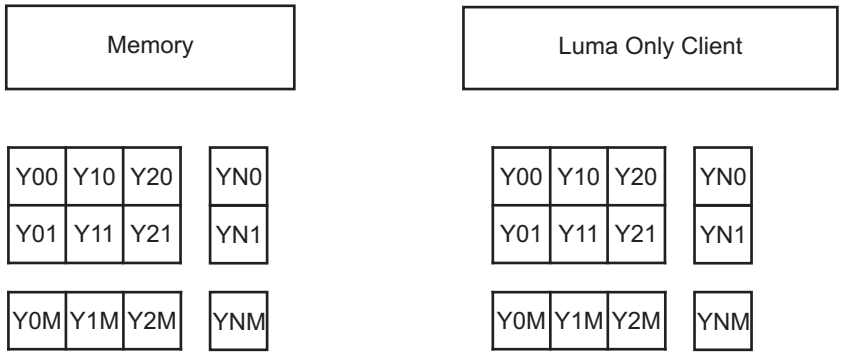
Figure 7-92. Y 4:2:2 (Data Type 1)



7.4.8.9.1.3 Y 4:2:0 (Data Type 2)

The Y 4:2:0 data type is used for a co-planar 4:2:0 data type. In this mode the data is assumed to be a single byte with each pixel is packed in a line. If a 2D transfer of the data is used the VPDMA will assume this data type is stored in an 8 bit container. This data block should have the width and the height of the expected frame for the client.

Figure 7-93. Y 4:2:0 (Data Type 2)



7.4.8.9.1.4 C 4:4:4 (Data Type 4)

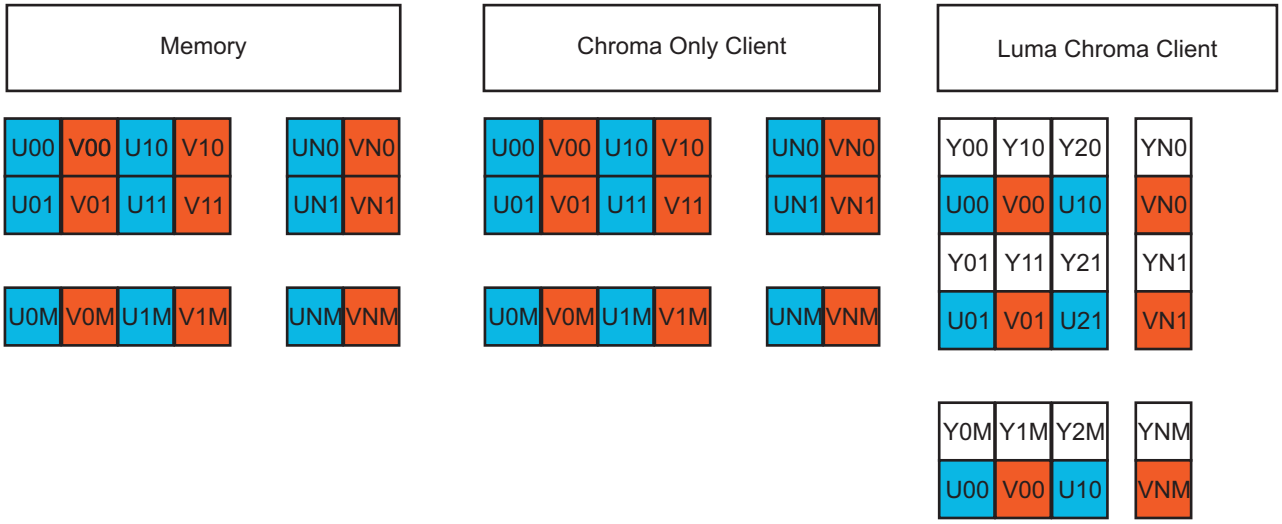
The C 4:4:4 data type is used for a co-planar 4:4:4 data. It is expected to be packed Cb in the lower byte and Cr in the upper byte in 16 bit words for each pixel. If a 2D transfer of the data is used the VPDMA will assume this data type is stored in an 16 bit container. This data block should have the width and height of the expected client frame.

Figure 7-94. C 4:4:4 (Data Type 4)

7.4.8.9.1.5 C 4:2:2 (Data Type 5)

The C 4:2:2 data type is used for co-planar 4:2:2 data.It is expected to be packed Cb in the lower byte and Cr in the upper byte in 16 bit words for each pixel. If a 2D transfer of the data is used the VPDMA will assume this data type is stored in an 16 bit container. This data block should have the width and the height of the expected client frame.

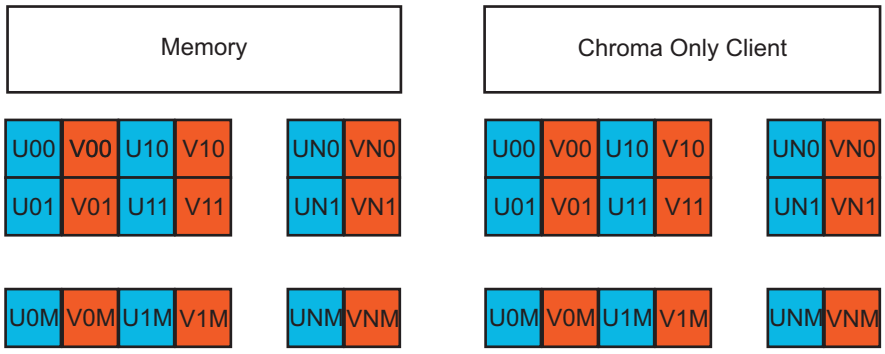
Figure 7-95. C 4:2:2 (Data Type 5)



7.4.8.9.1.6 C 4:2:0 (Data Type 6)

The C 4:2:0 data type is used for a co-planar 4:2:0 data. It is expected to be packed Cb in the lower byte and Cr in the upper byte in 16 bit words for each pixel. If a 2D transfer of the data is used the VPDMA will assume this data type is stored in a 16 bit container. This data block should have the width and half the height of the expected clients frame.

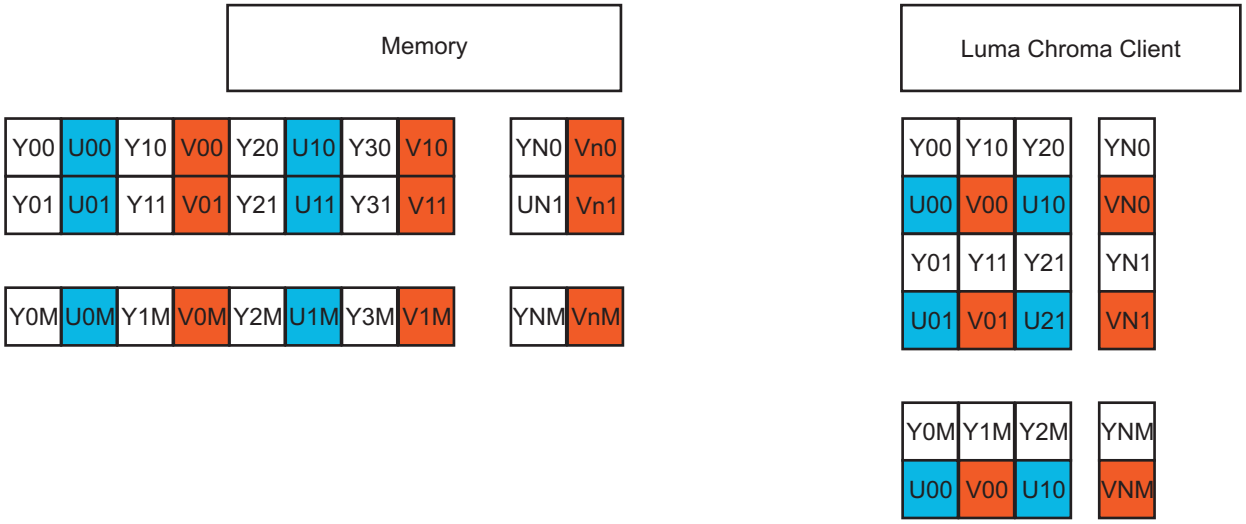
Figure 7-96. C 4:2:0 (Data Type 6)



7.4.8.9.1.7 YC 4:2:2 (Data Type 7)

The YC 4:2:2 data type is used for interleaved 4:2:2 data. It is expected to be packed with Y0 in the lowest byte followed by Cb followed by Y1 finally Cr in the upper most byte. The transfer counts each YC pair in a 16 bit word as a pixel but the number of pixels should be even. If a 2D transfer of the data is used the VPDMA will assume this data type is stored in an 32 bit container. The data block width and height should be set the same as the expected client.

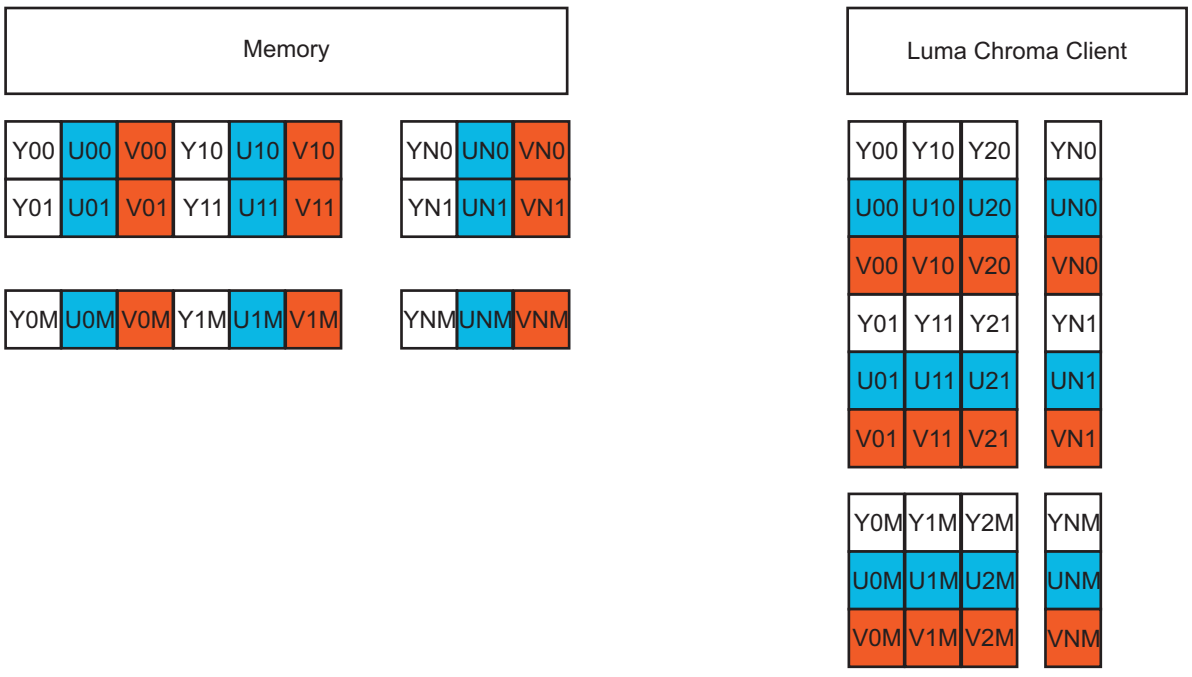
Figure 7-97. YC 4:2:2 (Data Type 7)



7.4.8.9.1.8 YC 4:4:4 (Data Type 8)

The YC 4:4:4 data type is used for interleaved 4:4:4 data. It is expected to be packed with Y0 in the lowest byte followed by Cb followed by Cr. The transfer counts each YCbCr triplet in a 24 bit word as a pixel. A 2D transfer of the data is NOT supported in the VPDMA. The data block width and height should be set to the number of pixels and lines that are expected by the client.

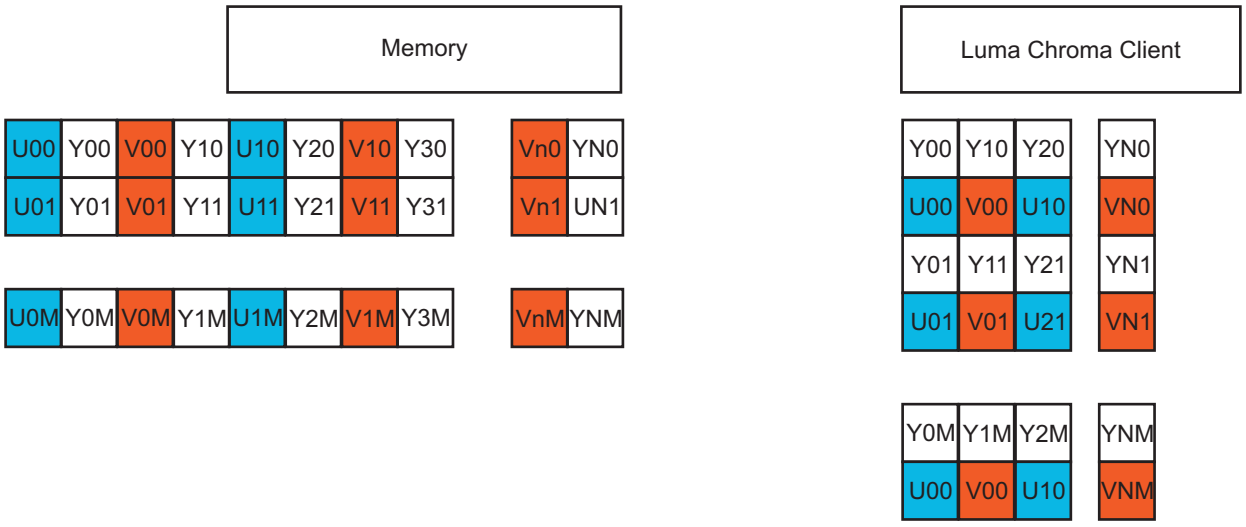
Figure 7-98. YC 4:4:4 (Data Type 8)



7.4.8.9.1.9 CY 4:2:2 (Data Type 23)

The CY 4:2:2 data type is used for interleaved 4:2:2 data. It is expected to be packed with Cb in the lowest byte followed by Y0 followed by Cr finally Y1 in the upper most byte. The transfer counts each YC pair in a 16 bit word as a pixel but the number of pixels should be even. If a 2D transfer of the data is used the VPDMA will assume this data type is stored in an 32 bit container. The data block width and height should be set the same as the expected client.

Figure 7-99. CY 4:2:2 (Data Type 23h)



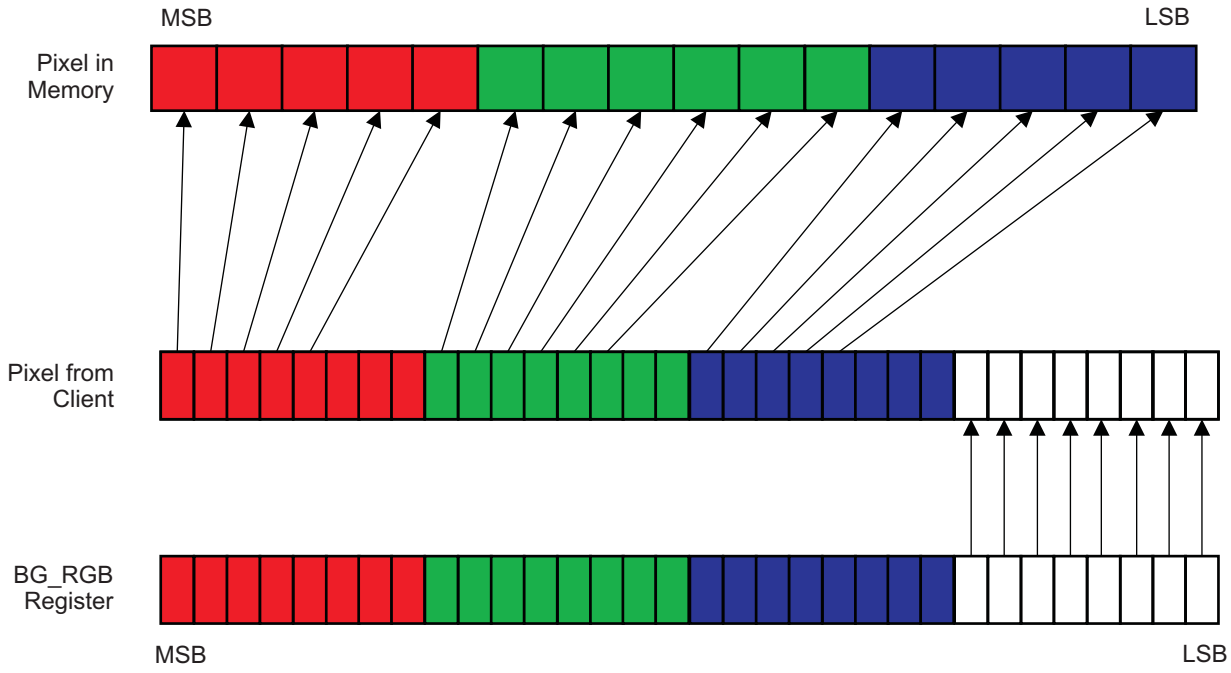
7.4.8.9.2 RGB Data Formats

The RGB channel type is used to provide data for a client that expects to transmit RGB data. In all modes the client is always RGBA 8888 data. The lower bits, if not provided by the data stream, are a replication of the lower bit of the data. For outbound data the input is assumed to be RGB 888 and the Alpha value is taken from the Background Color register to make a full ARGB 8888. The lower bits are dropped from this data, if a data type specifies less then the full 8 bits per color. The client has individual data buses for each component so they have no order dependency in the data bus.

7.4.8.9.2.1 RGB16-565 (Data Type 0)

In RGB16-565 mode, each pixel is a single RGB pixel with 16 bits of data for each pixel. The 16 bits of data uses the lower 5 bits for blue data, the middle 6 bits for green data and the upper 5 bits for red data.

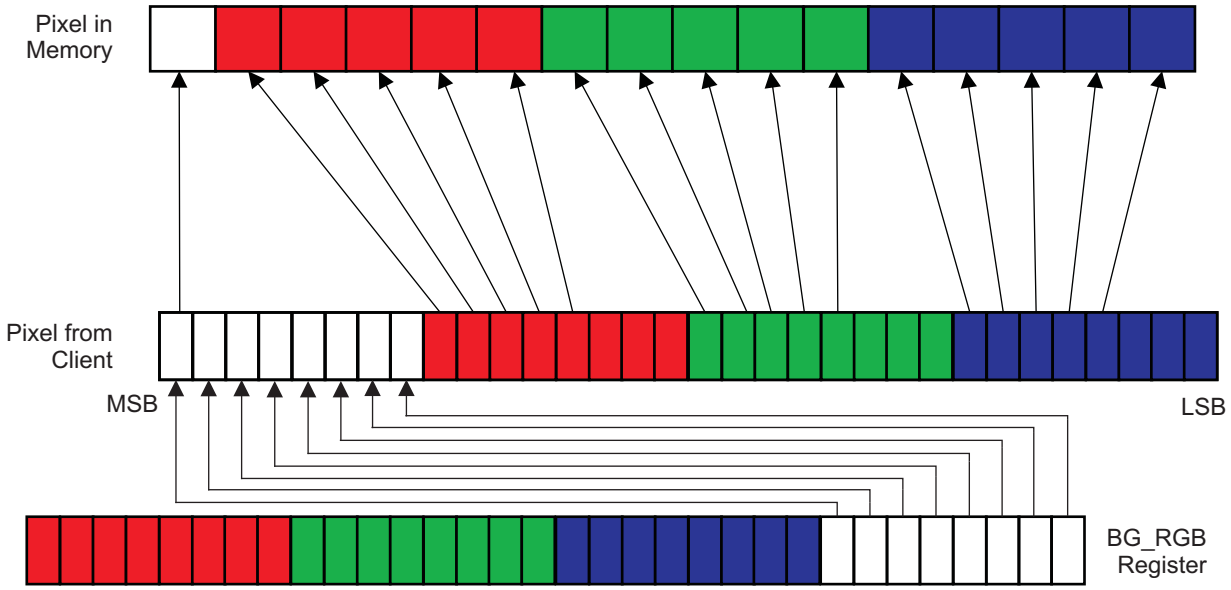
Figure 7-100. RGB16-565 (Data Type 0)



7.4.8.9.2.2 ARGB-1555 (Data Type 1)

In ARGB-1555 mode, each pixel is a single ARGB pixel with 16 bits of data for each pixel. The 16 bits of data uses the lower 5 bits for blue data, the next 5 bits for green data the next 5 bits for red data and the upper most bit for the blend value to use 0 or 0xFF.

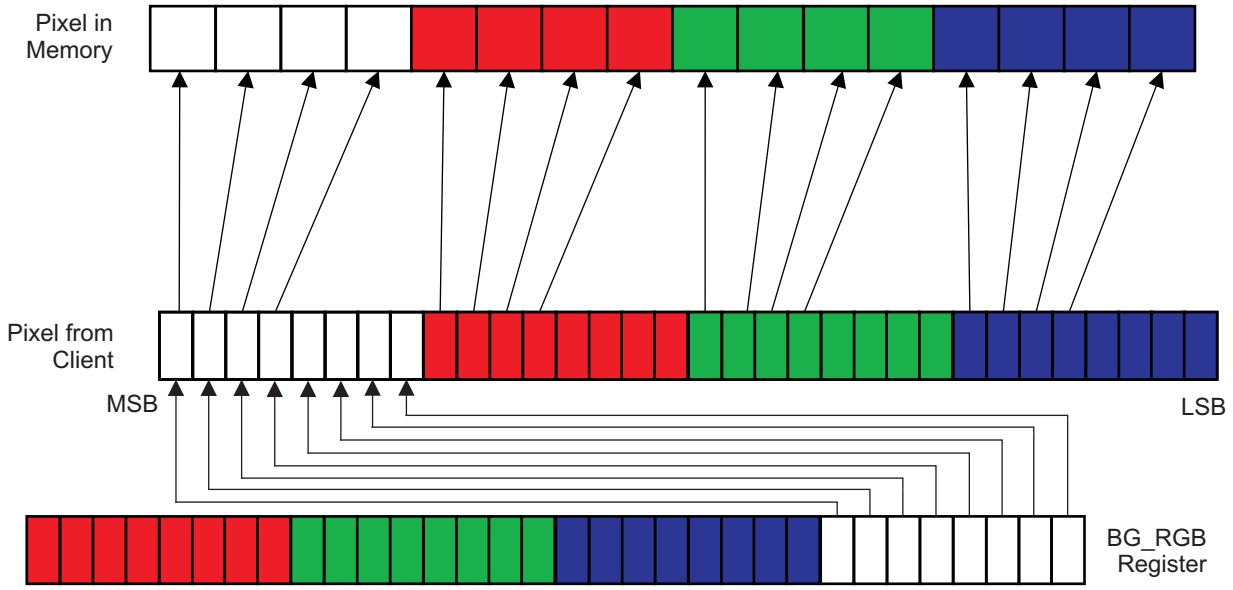
Figure 7-101. ARGB-1555 (Data Type 1)



7.4.8.9.2.3 ARGB-4444 (Data Type 2)

In ARGB16-4444 mode, each pixel is a single RGBA pixel with 16 bits of data for each pixel. The 16 bits of data uses the lowest 4 bits for blue data, the next 4 bits for green data the next 4 bits for red data and the upper most 4 bits for the blend value.

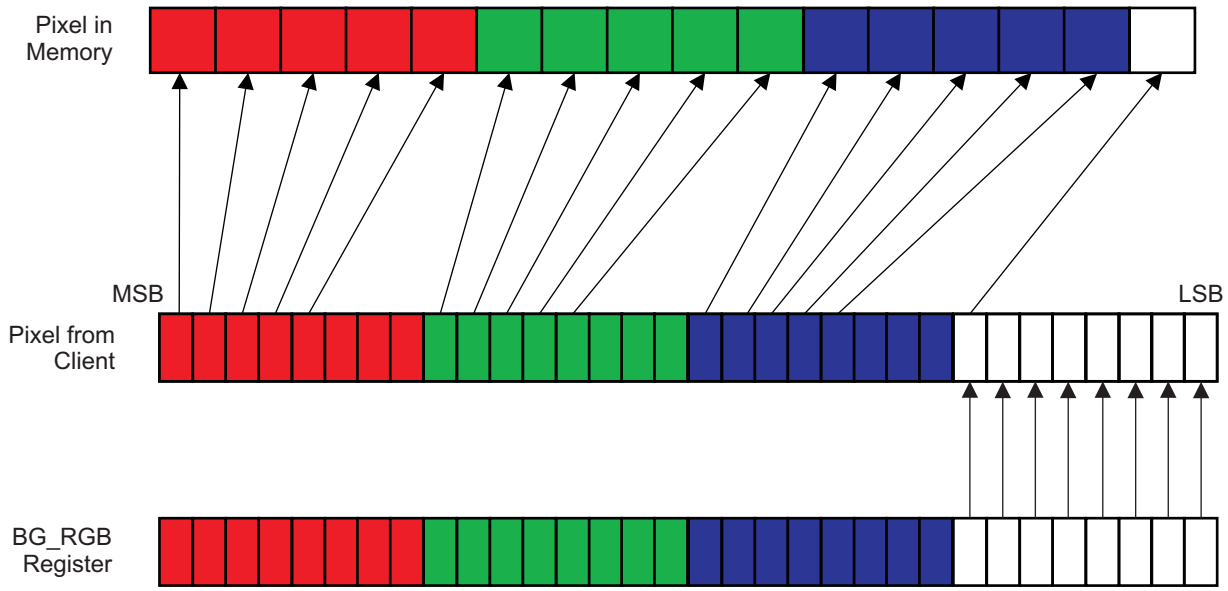
Figure 7-102. ARGB-4444 (Data Type 2)



7.4.8.9.2.4 RGBA-5551 (Data Type 3)

In RGBA-5551 mode, each pixel is a single ARGB pixel with 16 bits of data for each pixel. The 16 bits of data uses the lowest most bit for the blend value to use 0 or 0xff the next 5 bits for blue data, the next 5 bits for green data the next 5 bits for red data.

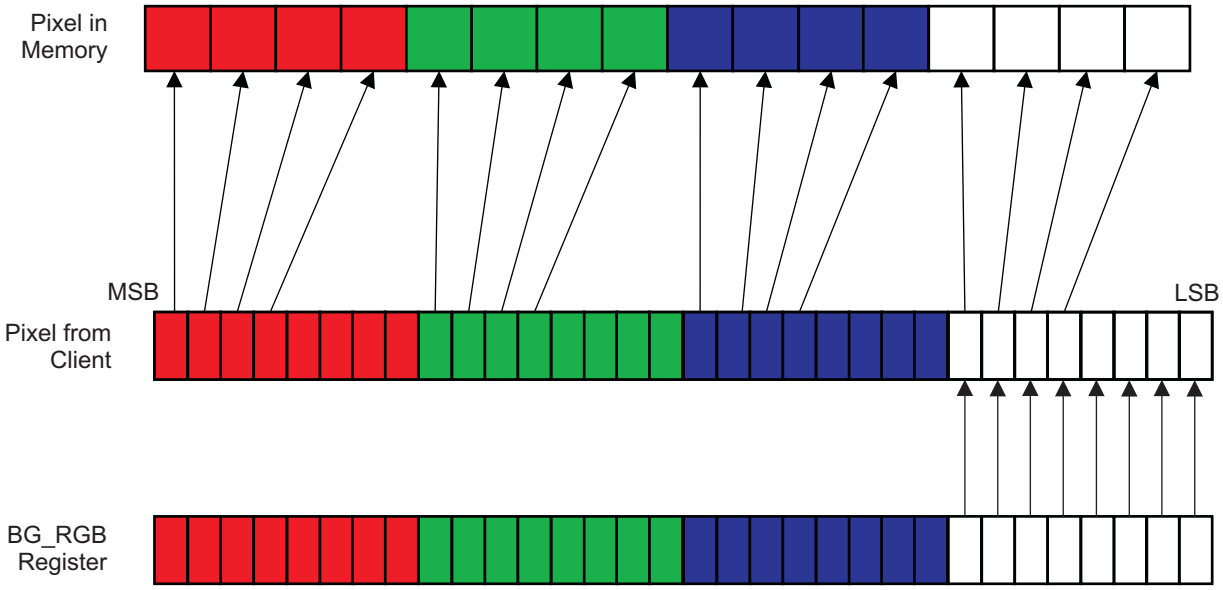
Figure 7-103. RGBA-5551 (Data Type 3)



7.4.8.9.2.5 RGBA-4444 (Data Type 4)

In RGBA-4444 mode, each pixel is a single RGBA pixel with 16 bits of data for each pixel. The 16 bits of data uses the lowest 4 bits for the blend value, the next 4 bits for blue data, the next 4 bits for green data and the upper most 4 bits for red data.

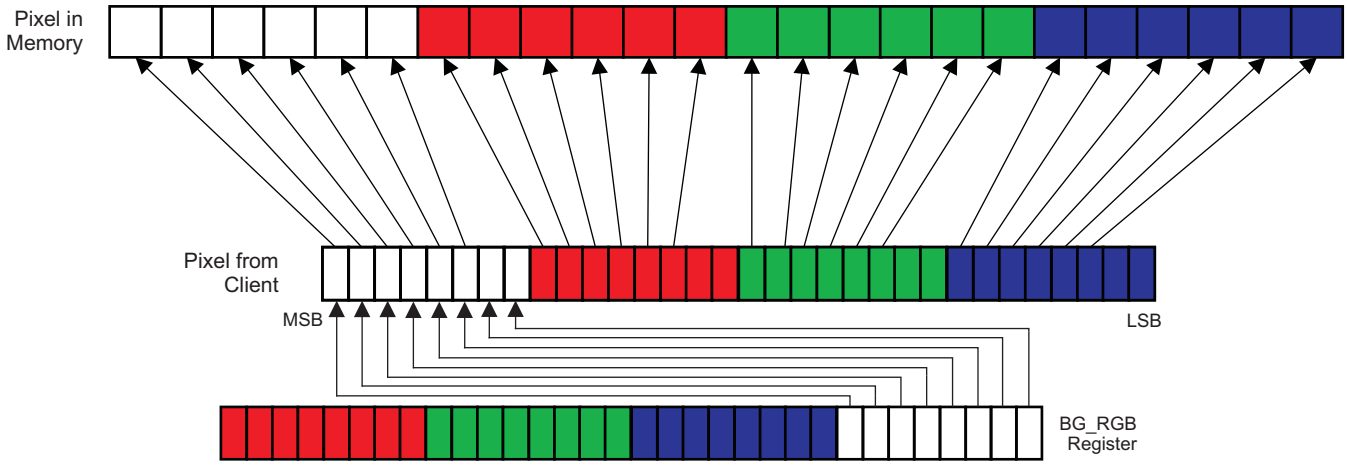
Figure 7-104. RGBA-4444 (Data Type 4)



7.4.8.9.2.6 ARGB24-6666 (Data Type 5)

In ARGB24-6666 mode, each pixel is a single ARGB pixel with 24 bits of data for each pixel. The 24 bits of data uses the lowest 6 bits for the blend value, the next 6 bits for blue data, the next 6 bits for green data and the upper most 6 bits for red data.

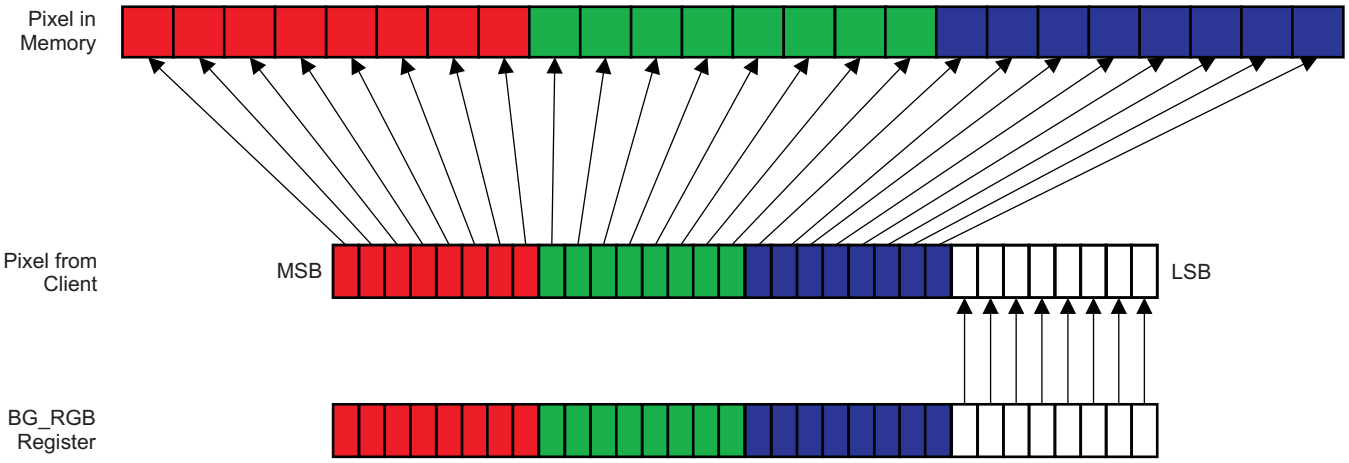
Figure 7-105. ARGB24-6666 (Data Type 5)



7.4.8.9.2.7 RGB24-888 (Data Type 6)

In RGB24-888 mode, each pixel is a single RGB pixel with 24 bits of data for each pixel. The 24 bits of data uses the lowest 8 bits for blue data, the next 8 bits for green data and the upper most 8 bits for red data it uses the BG_RGB Blend value for the Blend value.

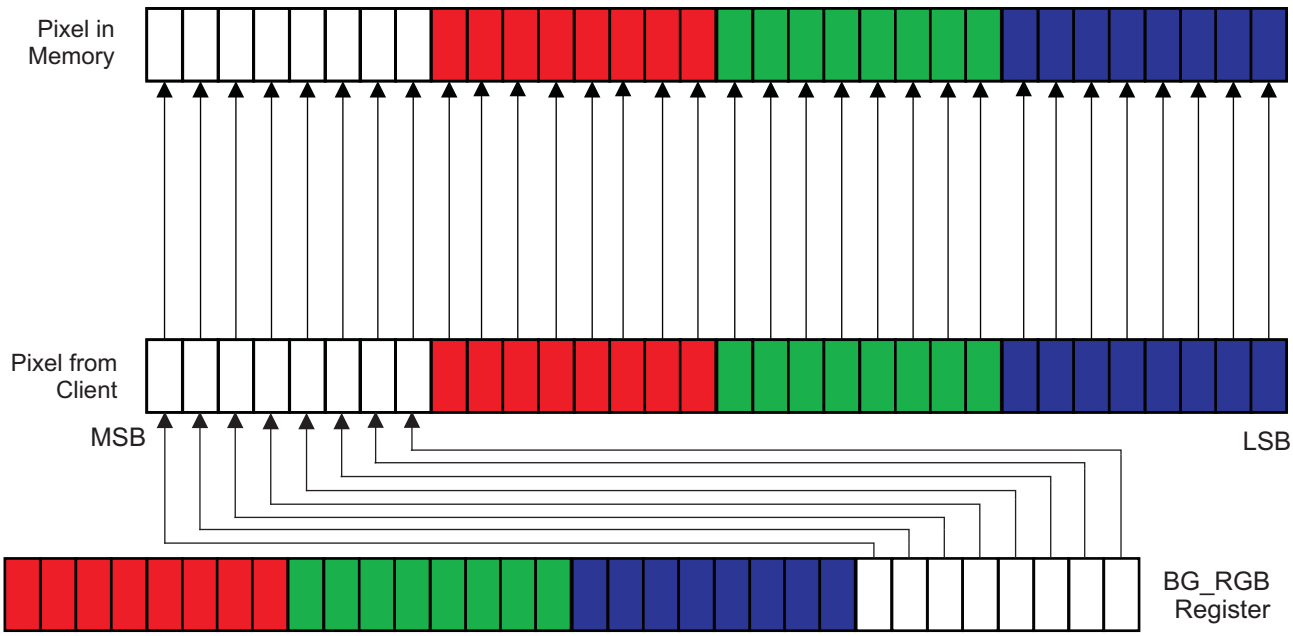
Figure 7-106. RGB24-888 (Data Type 6)



7.4.8.9.2.8 ARGB32-8888 (Data Type 7)

In ARGB32-8888 mode, each pixel is a single ARGB pixel with 32 bits of data for each pixel. The 32 bitsof data uses the lowest 8 bits for blue data, the next 8 bits for green data and the next 8 bits for red data it uses the upper most bits for the blend value.

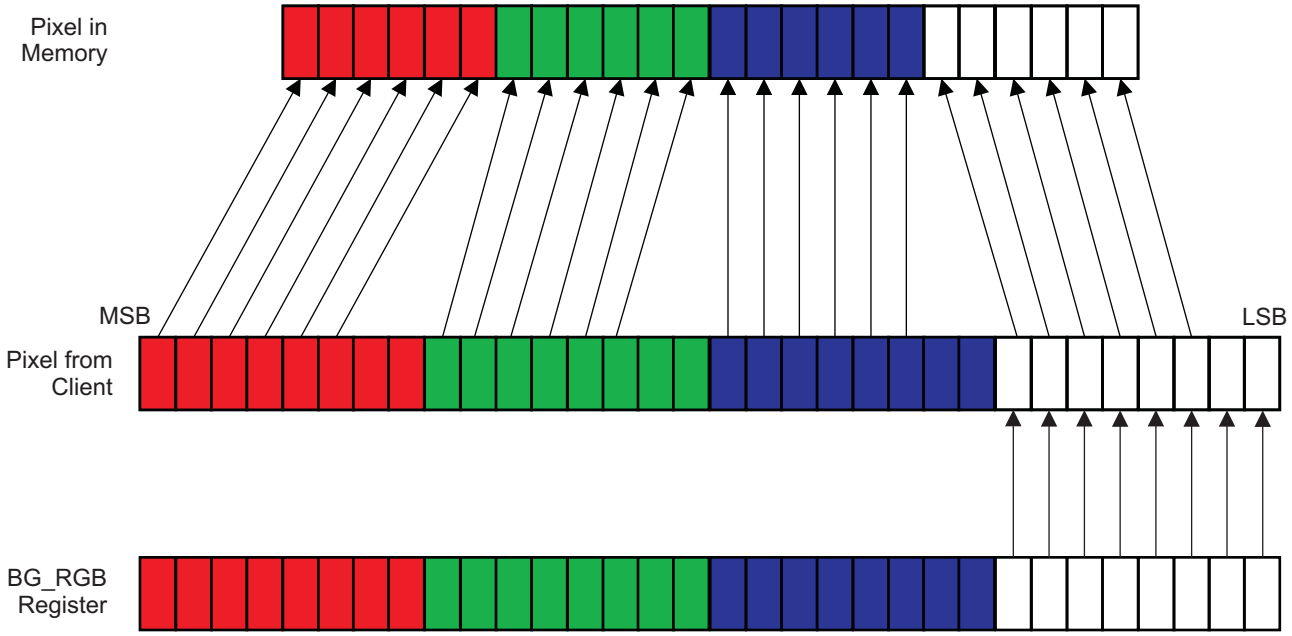
Figure 7-107. ARGB32-8888 (Data Type 7)



7.4.8.9.2.9 RGBA24-6666 (Data Type 8)

In RGBA24-6666 mode, each pixel is a single RGBA pixel with 24 bits of data for each pixel. The 24 bits of data uses the lowest 6 bits for the blend value, the next 6 bits for blue data, the next 6 bits for green data and the upper 6 bits for red data.

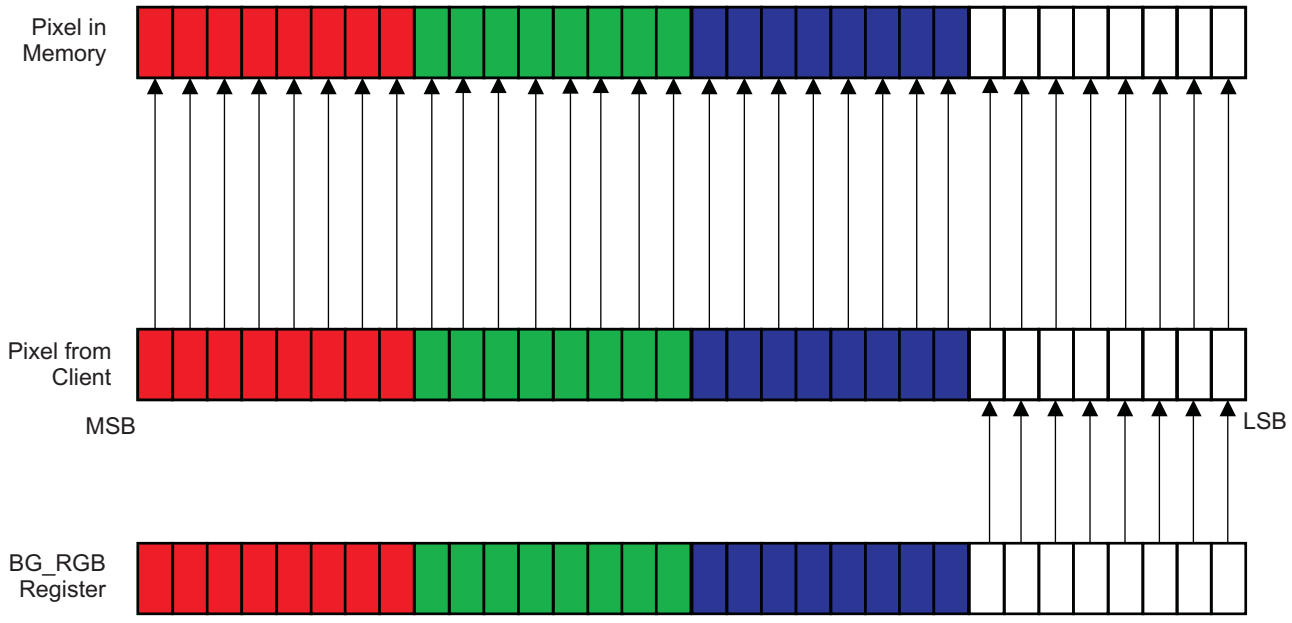
Figure 7-108. RGBA24-6666 (Data Type 8)



7.4.8.9.2.10 RGBA32-8888 (Data Type 9)

In RGBA32-8888 mode, each pixel is a single ARGB pixel with 32 bits of data for each pixel. The 32 bits of data uses the lowest 8 bits for the blend value the next 8 bits for blue data, the next 8 bits for green data and the upper most 8 bits for red data.

Figure 7-109. RGBA32-8888 (Data Type 9)



7.4.8.9.3 Miscellaneous Data Type

The Miscellaneous channel type is for any data type that is not a normal video type. The Miscellaneous channel type makes no assumptions on data type and just passes the data to the client and supports a single buffer for the client. The size of the miscellaneous data type is always determined by the Data Type field of the descriptor which specifies the number of bits in the descriptor.

A memory structure used to describe a desired memory transaction to or from a client. The descriptor at a minimum gives an address location for the memory portion of the transfer, the channel to use for this transaction and the size of the transaction. The data descriptor can also contain attributes to be passed down to the client or be linked to another data descriptor to form a larger frame from many smaller frames.

7.5 VIP Register Manual

7.5.1 VIP Instance Summary

Table 7-67. VIP Instance Summary

Module Name	Module Base Address	Size
VIP_top_level	0x4897 0000	276 Bytes
VIP_Slice0_parser	0x4897 5500	216 Bytes
VIP_Slice0_csc	0x4897 5700	24 Bytes
VIP_Slice0_sc	0x4897 5800	128 Bytes
VIP_Slice1_parser	0x4897 5A00	216 Bytes
VIP_Slice1_csc	0x4897 5C00	24 Bytes
VIP_Slice1_sc	0x4897 5D00	128 Bytes
VIP_VPDMA	0x4897 D000	1016 Bytes

7.5.2 VIP Top Level Registers

7.5.2.1 VIP Top Level Register Summary

Table 7-68. VIP Top Level Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	VIP_top_level Base Address
VIP_CLKC_PID	RW	32	0x0000 0000	0x4897 0000
VIP_SYSCONFIG	RW	32	0x0000 0010	0x4897 0010
VIP_INTC_INTR0_STATUS_RAW0	RW	32	0x0000 0020	0x4897 0020
VIP_INTC_INTR0_STATUS_RAW1	RW	32	0x0000 0024	0x4897 0024
VIP_INTC_INTR0_STATUS_ENA0	RW	32	0x0000 0028	0x4897 0028
VIP_INTC_INTR0_STATUS_ENA1	RW	32	0x0000 002C	0x4897 002C
VIP_INTC_INTR0_ENA_SET0	RW	32	0x0000 0030	0x4897 0030
VIP_INTC_INTR0_ENA_SET1	RW	32	0x0000 0034	0x4897 0034
VIP_INTC_INTR0_ENA_CLR0	RW	32	0x0000 0038	0x4897 0038
VIP_INTC_INTR0_ENA_CLR1	RW	32	0x0000 003C	0x4897 003C
VIP_INTC_INTR1_STATUS_RAW0	RW	32	0x0000 0040	0x4897 0040

Table 7-68. VIP Top Level Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	VIP_top_level Base Address
VIP_INTC_INTR1_STATUS_RAW1	RW	32	0x0000 0044	0x4897 0044
VIP_INTC_INTR1_STATUS_ENA0	RW	32	0x0000 0048	0x4897 0048
VIP_INTC_INTR1_STATUS_ENA1	RW	32	0x0000 004C	0x4897 004C
VIP_INTC_INTR1_ENA_SET0	RW	32	0x0000 0050	0x4897 0050
VIP_INTC_INTR1_ENA_SET1	RW	32	0x0000 0054	0x4897 0054
VIP_INTC_INTR1_ENA_CLR0	RW	32	0x0000 0058	0x4897 0058
VIP_INTC_INTR1_ENA_CLR1	RW	32	0x0000 005C	0x4897 005C
VIP_INTC_EOI	RW	32	0x0000 00A0	0x4897 00A0
VIP_CLKC_CLKEN	RW	32	0x0000 0100	0x4897 0100
VIP_CLKC_RST	RW	32	0x0000 0104	0x4897 0104
VIP_CLKC_DPS	RW	32	0x0000 0108	0x4897 0108
VIP_CLKC_VIP0DPS	RW	32	0x0000 010C	0x4897 010C
VIP_CLKC_VIP1DPS	RW	32	0x0000 0110	0x4897 0110

7.5.2.2 VIP Top Level Register Description

Table 7-69. VIP_CLKC_PID

Address Offset	0x0000 0000		
Physical Address	0x4897 0000	**Instance**	VIP_top_level
Description	This register follows the format described in PDR3.5		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCHEME		RESERVED		FUNC												RTL					MAJOR			CUSTOM				MINOR			

Bits	Field Name	Description	Type	Reset
31:30	SCHEME	The scheme of the register used. This indicates the PDR3.5 Method	R	0x0
29:28	RESERVED		R	0x0
27:16	FUNC	The function of the module being used	R	0x0
15:11	RTL	RTL Release Version The PDR release number of this IP	R	0x0
10:8	MAJOR	ajor Release Number	R	0x0
7:6	CUSTOM	Custom IP	R	0x0
5:0	MINOR	inor Release Number	R	0x0

Table 7-70. Register Call Summary for Register VIP_CLKC_PID

VIP Register Manual

- VIP Top Level Register Summary: [2]

Table 7-71. VIP_SYSCONFIG

Address Offset	0x0000 0010		
Physical Address	0x4897 0010	**Instance**	VIP_top_level
Description	VIP_SYSCONFIG		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								STANDBYMODE				IDLEMODE		RESERVED	

Bits	Field Name	Description	Type	Reset
31:6	RESERVED		R	0x0
5:4	STANDBYMODE	Configuration of the local initiator state management mode. By definition, initiator may generate read/write transaction as long as it is out of STANDBY state 0x0: Force-standby mode: local initiator is unconditionally placed in standby state. Backup mode, for debug only 0x1: No-standby mode: local initiator is unconditionally placed out of standby state. Backup mode, for debug only 0x2: Same behavior as bit-field value of 0x1. 0x3: Reserved	RW	0x2
3:2	IDLEMODE	Configuration of the local target state management mode. By definition, target can handle read/write transaction as long as it is out of IDLE state 0x0 : Force-idle mode: local target's idle state follows (acknowledges) the system's idle requests unconditionally, i.e. regardless of the IP module's internal requirements. Backup mode, for debug only 0x1 : No-idle mode: local target never enters idle state. Backup mode, for debug only 0x2 : Smart-idle mode: local target's idle state eventually follows (acknowledges) the system's idle requests, depending on the IP module's internal requirements. IP module shall not generate (IRQ- or DMA-request-related) wakeup events 0x3 : Smart-idle wakeup-capable mode: local target's idle state eventually follows (acknowledges) the system's idle requests, depending on the IP module's internal requirements. IP module may generate (IRQ- or DMA-request-related) wakeup events when in idle state. Mode is only relevant if the appropriate IP module 'swakeup' output(s) is (are) implemented	RW	0x2
1:0	RESERVED		R	0x0

Table 7-72. Register Call Summary for Register VIP_SYSCONFIG

VIP Functional Description

- VIP Idle Mode: [0]
- VIP StandBy Mode: [1]

VIP Register Manual

- VIP Top Level Register Summary: [4]
- VIP Top Level Register Description: [5]

Table 7-73. VIP_INTC_INTR0_STATUS_RAW0

Address Offset	0x0000 0020		
Physical Address	0x4897 0020	**Instance**	VIP_top_level
Description	INTC INTR0 Interrupt Status Raw/Set Register 0. This register contains the raw interrupt status as defined in HL0.8		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED										VIP2_PARSER_INT_RAW	VIP1_PARSER_INT_RAW	RESERVED			VPDMA_INT0_DESCRIPTOR_RAW	VPDMA_INT0_LIST7_NOTIFY_RAW	VPDMA_INT0_LIST7_COMPLETE_RAW	VPDMA_INT0_LIST6_NOTIFY_RAW	VPDMA_INT0_LIST6_COMPLETE_RAW	VPDMA_INT0_LIST5_NOTIFY_RAW	VPDMA_INT0_LIST5_COMPLETE_RAW	VPDMA_INT0_LIST4_NOTIFY_RAW	VPDMA_INT0_LIST4_COMPLETE_RAW	VPDMA_INT0_LIST3_NOTIFY_RAW	VPDMA_INT0_LIST3_COMPLETE_RAW	VPDMA_INT0_LIST2_NOTIFY_RAW	VPDMA_INT0_LIST2_COMPLETE_RAW	VPDMA_INT0_LIST1_NOTIFY_RAW	VPDMA_INT0_LIST1_COMPLETE_RAW	VPDMA_INT0_LIST0_NOTIFY_RAW	VPDMA_INT0_LIST0_COMPLETE_RAW

Bits	Field Name	Description	Type	Reset
31:22	RESERVED		R	0x0
21	VIP2_PARSER_INT_RAW	VIP2 Parser Interrupt Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
20	VIP1_PARSER_INT_RAW	VIP1 Parser Interrupt Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
19:17	RESERVED		R	0x0
16	VPDMA_INT0_DESCRIPTOR_RAW	VPDMA INT0 Descriptor Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
15	VPDMA_INT0_LIST7_NOTIFY_RAW	VPDMA INT0 List7 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
14	VPDMA_INT0_LIST7_COMPLETE_RAW	VPDMA INT0 List7 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
13	VPDMA_INT0_LIST6_NOTIFY_RAW	VPDMA INT0 List6 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
12	VPDMA_INT0_LIST6_COMPLETE_RAW	VPDMA INT0 List6 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
11	VPDMA_INT0_LIST5_NOTIFY_RAW	VPDMA INT0 List5 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
10	VPDMA_INT0_LIST5_COMPLETE_RAW	VPDMA INT0 List5 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
9	VPDMA_INT0_LIST4_NOTIFY_RAW	VPDMA INT0 List4 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
8	VPDMA_INT0_LIST4_COMPLETE_RAW	VPDMA INT0 List4 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0

Bits	Field Name	Description	Type	Reset
7	VPDMA_INT0_LIST3_NOTIFY_RAW	VPDMA INT0 List3 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
6	VPDMA_INT0_LIST3_COMPLETE_RAW	VPDMA INT0 List3 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
5	VPDMA_INT0_LIST2_NOTIFY_RAW	VPDMA INT0 List2 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
4	VPDMA_INT0_LIST2_COMPLETE_RAW	VPDMA INT0 List2 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
3	VPDMA_INT0_LIST1_NOTIFY_RAW	VPDMA INT0 List1 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
2	VPDMA_INT0_LIST1_COMPLETE_RAW	VPDMA INT0 List1 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
1	VPDMA_INT0_LIST0_NOTIFY_RAW	VPDMA INT0 List0 Notify Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
0	VPDMA_INT0_LIST0_COMPLETE_RAW	VPDMA INT0 List0 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0

Table 7-74. Register Call Summary for Register VIP_INTC_INTR0_STATUS_RAW0

VIP Register Manual

- VIP Top Level Register Summary: [2]

Table 7-75. VIP_INTC_INTR0_STATUS_RAW1

Address Offset	0x0000 0024		
Physical Address	0x4897 0024	**Instance**	VIP_top_level
Description	INTC INTR0 Interrupt Status Raw/Set Register 1. This register contains the raw interrupt status as defined in HL0.8		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED						VIP2_CHR_DS_2_UV_ERR_INT_RAW	VIP2_CHR_DS_1_UV_ERR_INT_RAW	VIP1_CHR_DS_2_UV_ERR_INT_RAW	VIP1_CHR_DS_1_UV_ERR_INT_RAW	RESERVED														VPDMA_INT0_CLIENT_RAW	RESERVED	VPDMA_INT0_CHANNEL_GROUP5_RAW	VPDMA_INT0_CHANNEL_GROUP4_RAW	VPDMA_INT0_CHANNEL_GROUP3_RAW	VPDMA_INT0_CHANNEL_GROUP2_RAW	VPDMA_INT0_CHANNEL_GROUP1_RAW	VPDMA_INT0_CHANNEL_GROUP0_RAW

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	VIP2_CHR_DS_2_UV_ERR_INT_RAW	VIP2 Chroma Downsampler 2 UV Error Interrupt Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
24	VIP2_CHR_DS_1_UV_ERR_INT_RAW	VIP2 Chroma Downsampler 1 UV Error Interrupt Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
23	VIP1_CHR_DS_2_UV_ERR_INT_RAW	VIP1 Chroma Downsampler 2 UV Error Interrupt Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
22	VIP1_CHR_DS_1_UV_ERR_INT_RAW	VIP1 Chroma Downsampler 1 UV Error Interrupt Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
21:8	RESERVED		R	0x0
7	VPDMA_INT0_CLIENT_RAW	VPDMA INT0 Client Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
6	RESERVED		R	0x0
5	VPDMA_INT0_CHANNEL_GROUP5_RAW	VPDMA INT0 Channel Group5 Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
4	VPDMA_INT0_CHANNEL_GROUP4_RAW	VPDMA INT0 Channel Group4 Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
3	VPDMA_INT0_CHANNEL_GROUP3_RAW	VPDMA INT0 Channel Group3 Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
2	VPDMA_INT0_CHANNEL_GROUP2_RAW	VPDMA INT0 Channel Group2 Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
1	VPDMA_INT0_CHANNEL_GROUP1_RAW	VPDMA INT0 Channel Group1 Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
0	VPDMA_INT0_CHANNEL_GROUP0_RAW	VPDMA INT0 Channel Group0 Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0

Table 7-76. Register Call Summary for Register VIP_INTC_INTR0_STATUS_RAW1

VIP Register Manual

- VIP Top Level Register Summary: [2]

Table 7-77. VIP_INTC_INTR0_STATUS_ENA0

Address Offset	0x0000 0028		
Physical Address	0x4897 0028	**Instance**	VIP_top_level
Description	INTC INTR0 Interrupt Status Enabled/Clear Register 0. This register contains the enabled interrupt status as defined in HL0.8		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED										VIP2_PARSER_INT_ENA	VIP1_PARSER_INT_ENA	RESERVED			VPDMA_INT0_DESCRIPTOR_ENA	VPDMA_INT0_LIST7_NOTIFY_ENA	VPDMA_INT0_LIST7_COMPLETE_ENA	VPDMA_INT0_LIST6_NOTIFY_ENA	VPDMA_INT0_LIST6_COMPLETE_ENA	VPDMA_INT0_LIST5_NOTIFY_ENA	VPDMA_INT0_LIST5_COMPLETE_ENA	VPDMA_INT0_LIST4_NOTIFY_ENA	VPDMA_INT0_LIST4_COMPLETE_ENA	VPDMA_INT0_LIST3_NOTIFY_ENA	VPDMA_INT0_LIST3_COMPLETE_ENA	VPDMA_INT0_LIST2_NOTIFY_ENA	VPDMA_INT0_LIST2_COMPLETE_ENA	VPDMA_INT0_LIST1_NOTIFY_ENA	VPDMA_INT0_LIST1_COMPLETE_ENA	VPDMA_INT0_LIST0_NOTIFY_ENA	VPDMA_INT0_LIST0_COMPLETE_ENA

Bits	Field Name	Description	Type	Reset
31:22	RESERVED		R	0x0
21	VIP2_PARSER_INT_ENA	VIP2 Parser Enabled Interrupt Status Read indicates enabled status 0 = inactive 1 = active Writing 1 will clear interrupt Writing 0 has no effect	RW	0x0
20	VIP1_PARSER_INT_ENA	VIP1 Parser Enabled Interrupt Status Read indicates enabled status 0 = inactive 1 = active Writing 1 will clear interrupt Writing 0 has no effect	RW	0x0
19:17	RESERVED		R	0x0
16	VPDMA_INT0_DESCRIPTOR_ENA	VPDMA INT0 Descriptor Enabled Status Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
15	VPDMA_INT0_LIST7_NOTIFY_ENA	VPDMA INT0 List7 Notify Enabled Status Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
14	VPDMA_INT0_LIST7_COMPLETE_ENA	VPDMA INT0 List7 Complete Enabled Status Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
13	VPDMA_INT0_LIST6_NOTIFY_ENA	VPDMA INT0 List6 Notify Enabled Status Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
12	VPDMA_INT0_LIST6_COMPLETE_ENA	VPDMA INT0 List6 Complete Enabled Status Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
11	VPDMA_INT0_LIST5_NOTIFY_ENA	VPDMA INT0 List5 Notify Enabled Status Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
10	VPDMA_INT0_LIST5_COMPLETE_ENA	VPDMA INT0 List5 Complete Enabled Statust Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
9	VPDMA_INT0_LIST4_NOTIFY_ENA	VPDMA INT0 List4 Notify Enabled Statust Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
8	VPDMA_INT0_LIST4_COMPLETE_ENA	VPDMA INT0 List4 Complete Enabled Status Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
7	VPDMA_INT0_LIST3_NOTIFY_ENA	VPDMA INT0 List3 Notify Enabled Status Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
6	VPDMA_INT0_LIST3_COMPLETE_ENA	VPDMA INT0 List3 Complete Enabled Status Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
5	VPDMA_INT0_LIST2_NOTIFY_ENA	VPDMA INT0 List2 Notify Enabled Statust Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0

Bits	Field Name	Description	Type	Reset
4	VPDMA_INT0_LIST2_COMPLETE_ENA	VPDMA INT0 List2 Complete Enabled Statust Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
3	VPDMA_INT0_LIST1_NOTIFY_ENA	VPDMA INT0 List1 Notify Enabled Status Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
2	VPDMA_INT0_LIST1_COMPLETE_ENA	VPDMA INT0 List1 Complete Enabled Status Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
1	VPDMA_INT0_LIST0_NOTIFY_ENA	VPDMA INT0 List0 Notify Enabled Status Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
0	VPDMA_INT0_LIST0_COMPLETE_ENA	VPDMA INT0 List0 Complete Enabled Status Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0

Table 7-78. Register Call Summary for Register VIP_INTC_INTR0_STATUS_ENA0

VIP Register Manual

- VIP Top Level Register Summary: [2]

Table 7-79. VIP_INTC_INTR0_STATUS_ENA1

Address Offset	0x0000 002C		
Physical Address	0x4897 002C	**Instance**	VIP_top_level
Description	INTC INTR0 Interrupt Status Enabled/Clear Register 1. This register contains the enabled interrupt status as defined in HL0.8		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED						VIP2_CHR_DS_2_UV_ERR_INT_ENA	VIP2_CHR_DS_1_UV_ERR_INT_ENA	VIP1_CHR_DS_2_UV_ERR_INT_ENA	VIP1_CHR_DS_1_UV_ERR_INT_ENA	RESERVED														VPDMA_INT0_CLIENT_ENA	RESERVED	VPDMA_INT0_CHANNEL_GROUP5_ENA	VPDMA_INT0_CHANNEL_GROUP4_ENA	VPDMA_INT0_CHANNEL_GROUP3_ENA	VPDMA_INT0_CHANNEL_GROUP2_ENA	VPDMA_INT0_CHANNEL_GROUP1_ENA	VPDMA_INT0_CHANNEL_GROUP0_ENA

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	VIP2_CHR_DS_2_UV_ERR_INT_ENA	VIP2 Chroma Downsampler 2 UV Error Enabled Interrupt Status Read indicates enabled status 0 = inactive 1 = active Writing 1 will clear interrupt Writing 0 has no effect	RW	0x0
24	VIP2_CHR_DS_1_UV_ERR_INT_ENA	VIP2 Chroma Downsampler 1 UV Error Enabled Interrupt Status Read indicates enabled status 0 = inactive 1 = active Writing 1 will clear interrupt Writing 0 has no effect	RW	0x0
23	VIP1_CHR_DS_2_UV_ERR_INT_ENA	VIP1 Chroma Downsampler 2 UV Error Enabled Interrupt Status Read indicates enabled status 0 = inactive 1 = active Writing 1 will clear interrupt Writing 0 has no effect	RW	0x0

Bits	Field Name	Description	Type	Reset
22	VIP1_CHR_DS_1_UV_ERR_INT_ENA	VIP1 Chroma Downsampler 1 UV Error Enabled Interrupt Status Read indicates enabled status 0 = inactive 1 = active Writing 1 will clear interrupt Writing 0 has no effect	RW	0x0
21:8	RESERVED		R	0x0
7	VPDMA_INT0_CLIENT_ENA	VPDMA INT0 Client Enabled Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
6	RESERVED		R	0x0
5	VPDMA_INT0_CHANNEL_GROUP5_ENA	VPDMA INT0 Channel Group5 Enabled Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
4	VPDMA_INT0_CHANNEL_GROUP4_ENA	VPDMA INT0 Channel Group4 Enabled Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
3	VPDMA_INT0_CHANNEL_GROUP3_ENA	VPDMA INT0 Channel Group3 Enabled Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
2	VPDMA_INT0_CHANNEL_GROUP2_ENA	VPDMA INT0 Channel Group3 Enabled Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
1	VPDMA_INT0_CHANNEL_GROUP1_ENA	VPDMA INT0 Channel Group1 Enabled Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
0	VPDMA_INT0_CHANNEL_GROUP0_ENA	VPDMA INT0 Channel Group0 Enabled Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0

Table 7-80. Register Call Summary for Register VIP_INTC_INTR0_STATUS_ENA1

VIP Register Manual

- VIP Top Level Register Summary: [2]

Table 7-81. VIP_INTC_INTR0_ENA_SET0

Address Offset	0x0000 0030		
Physical Address	0x4897 0030	**Instance**	VIP_top_level
Description	INTC INTR0 Interrupt Enable/Set Register 0. This register contains the interrupt enable status/set as defined in HL0.8		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED										VIP2_PARSER_INT_ENA_SET	VIP1_PARSER_INT_ENA_SET	RESERVED			VPDMA_INT0_DESCRIPTOR_ENA_SET	VPDMA_INT0_LIST7_NOTIFY_ENA_SET	VPDMA_INT0_LIST7_COMPLETE_ENA_SET	VPDMA_INT0_LIST6_NOTIFY_ENA_SET	VPDMA_INT0_LIST6_COMPLETE_ENA_SET	VPDMA_INT0_LIST5_NOTIFY_ENA_SET	VPDMA_INT0_LIST5_COMPLETE_ENA_SET	VPDMA_INT0_LIST4_NOTIFY_ENA_SET	VPDMA_INT0_LIST4_COMPLETE_ENA_SET	VPDMA_INT0_LIST3_NOTIFY_ENA_SET	VPDMA_INT0_LIST3_COMPLETE_ENA_SET	VPDMA_INT0_LIST2_NOTIFY_ENA_SET	VPDMA_INT0_LIST2_COMPLETE_ENA_SET	VPDMA_INT0_LIST1_NOTIFY_ENA_SET	VPDMA_INT0_LIST1_COMPLETE_ENA_SET	VPDMA_INT0_LIST0_NOTIFY_ENA_SET	VPDMA_INT0_LIST0_COMPLETE_ENA_SET

Bits	Field Name	Description	Type	Reset
31:22	RESERVED		R	0x0
21	VIP2_PARSER_INT_ENA_SET	VIP2 Parser Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
20	VIP1_PARSER_INT_ENA_SET	VIP1 Parser Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
19:17	RESERVED		R	0x0
16	VPDMA_INT0_DESCRIPTOR_ENA_SET	VPDMA INT0 Descriptor Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
15	VPDMA_INT0_LIST7_NOTIFY_ENA_SET	VPDMA INT0 List7 Notify Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
14	VPDMA_INT0_LIST7_COMPLETE_ENA_SET	VPDMA INT0 List7 Complete Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
13	VPDMA_INT0_LIST6_NOTIFY_ENA_SET	VPDMA INT0 List6 Notify Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
12	VPDMA_INT0_LIST6_COMPLETE_ENA_SET	VPDMA INT0 List6 Complete Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
11	VPDMA_INT0_LIST5_NOTIFY_ENA_SET	VPDMA INT0 List5 Notify Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
10	VPDMA_INT0_LIST5_COMPLETE_ENA_SET	VPDMA INT0 List5 Complete Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
9	VPDMA_INT0_LIST4_NOTIFY_ENA_SET	VPDMA INT0 List4 Notify Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
8	VPDMA_INT0_LIST4_COMPLETE_ENA_SET	VPDMA INT0 List4 Complete Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
7	VPDMA_INT0_LIST3_NOTIFY_ENA_SET	VPDMA INT0 List3 Notify Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
6	VPDMA_INT0_LIST3_COMPLETE_ENA_SET	VPDMA INT0 List3 Complete Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
5	VPDMA_INT0_LIST2_NOTIFY_ENA_SET	VPDMA INT0 List2 Notify Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
4	VPDMA_INT0_LIST2_COMPLETE_ENA_SET	VPDMA INT0 List2 Complete Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
3	VPDMA_INT0_LIST1_NOTIFY_ENA_SET	VPDMA INT0 List1 Notify Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0

Bits	Field Name	Description	Type	Reset
2	VPDMA_INT0_LIST1_COMPLETE_ENA_SET	VPDMA INT0 List1 Complete Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
1	VPDMA_INT0_LIST0_NOTIFY_ENA_SET	VPDMA INT0 List0 Notify Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
0	VPDMA_INT0_LIST0_COMPLETE_ENA_SET	VPDMA INT0 List0 Complete Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0

Table 7-82. Register Call Summary for Register VIP_INTC_INTR0_ENA_SET0

VIP Register Manual

- VIP Top Level Register Summary: [2]

Table 7-83. VIP_INTC_INTR0_ENA_SET1

Address Offset	0x0000 0034		
Physical Address	0x4897 0034	**Instance**	VIP_top_level
Description	INTC INTR0 Interrupt Enable/Set Register 1. This register contains the interrupt enable status/set as defined in HL0.8		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED						VIP2_CHR_DS_2_UV_ERR_INT_ENA_SET	VIP2_CHR_DS_1_UV_ERR_INT_ENA_SET	VIP1_CHR_DS_2_UV_ERR_INT_ENA_SET	VIP1_CHR_DS_1_UV_ERR_INT_ENA_SET	RESERVED														VPDMA_INT0_CLIENT_ENA_SET	VPDMA_INT0_CHANNEL_GROUP6_ENA_SET	VPDMA_INT0_CHANNEL_GROUP5_ENA_SET	VPDMA_INT0_CHANNEL_GROUP4_ENA_SET	VPDMA_INT0_CHANNEL_GROUP3_ENA_SET	VPDMA_INT0_CHANNEL_GROUP2_ENA_SET	VPDMA_INT0_CHANNEL_GROUP1_ENA_SET	VPDMA_INT0_CHANNEL_GROUP0_ENA_SET

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	VIP2_CHR_DS_2_UV_ERR_INT_ENA_SET	VIP2 Chroma Downsampler 2 UV Error Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
24	VIP2_CHR_DS_1_UV_ERR_INT_ENA_SET	VIP2 Chroma Downsampler 1 UV Error Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
23	VIP1_CHR_DS_2_UV_ERR_INT_ENA_SET	VIP1 Chroma Downsampler 2 UV Error Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0

Bits	Field Name	Description	Type	Reset
22	VIP1_CHR_DS_1_UV_ERR_INT_ENA_SET	VIP1 Chroma Downsampler 1 UV Error Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
21:8	RESERVED		R	0x0
7	VPDMA_INT0_CLIENT_ENA_SET	VPDMA INT0 Client Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
6	VPDMA_INT0_CHANNEL_GROUP6_ENA_SET	VPDMA INT0 Channel Group6 Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
5	VPDMA_INT0_CHANNEL_GROUP5_ENA_SET	VPDMA INT0 Channel Group5 Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
4	VPDMA_INT0_CHANNEL_GROUP4_ENA_SET	VPDMA INT0 Channel Group4 Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
3	VPDMA_INT0_CHANNEL_GROUP3_ENA_SET	VPDMA INT0 Channel Group3 Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
2	VPDMA_INT0_CHANNEL_GROUP2_ENA_SET	VPDMA INT0 Channel Group2 Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
1	VPDMA_INT0_CHANNEL_GROUP1_ENA_SET	VPDMA INT0 Channel Group1 Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
0	VPDMA_INT0_CHANNEL_GROUP0_ENA_SET	VPDMA INT0 Channel Group0 Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0

Table 7-84. Register Call Summary for Register VIP_INTC_INTR0_ENA_SET1

VIP Register Manual

- VIP Top Level Register Summary: [2]

Table 7-85. VIP_INTC_INTR0_ENA_CLR0

Address Offset	0x0000 0038		
Physical Address	0x4897 0038	**Instance**	VIP_top_level
Description	INTC INTR0 Interrupt Enable/Clear Register 0. This register contains the interrupt enable status/clear as defined in HL0.8		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED										VIP2_PARSER_INT_ENA_CLR	VIP1_PARSER_INT_ENA_CLR	RESERVED			VPDMA_INT0_DESCRIPTOR_ENA_CLR	VPDMA_INT0_LIST7_NOTIFY_ENA_CLR	VPDMA_INT0_LIST7_COMPLETE_ENA_CLR	VPDMA_INT0_LIST6_NOTIFY_ENA_CLR	VPDMA_INT0_LIST6_COMPLETE_ENA_CLR	VPDMA_INT0_LIST5_NOTIFY_ENA_CLR	VPDMA_INT0_LIST5_COMPLETE_ENA_CLR	VPDMA_INT0_LIST4_NOTIFY_ENA_CLR	VPDMA_INT0_LIST4_COMPLETE_ENA_CLR	VPDMA_INT0_LIST3_NOTIFY_ENA_CLR	VPDMA_INT0_LIST3_COMPLETE_ENA_CLR	VPDMA_INT0_LIST2_NOTIFY_ENA_CLR	VPDMA_INT0_LIST2_COMPLETE_ENA_CLR	VPDMA_INT0_LIST1_NOTIFY_ENA_CLR	VPDMA_INT0_LIST1_COMPLETE_ENA_CLR	VPDMA_INT0_LIST0_NOTIFY_ENA_CLR	VPDMA_INT0_LIST0_COMPLETE_ENA_CLR

Bits	Field Name	Description	Type	Reset
31:22	RESERVED		R	0x0
21	VIP2_PARSER_INT_ENA_CLR	VIP2 Parser Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
20	VIP1_PARSER_INT_ENA_CLR	VIP1 Parser Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
19:17	RESERVED		R	0x0
16	VPDMA_INT0_DESCRIPTOR_ENA_CLR	VPDMA INT0 Descriptor Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
15	VPDMA_INT0_LIST7_NOTIFY_ENA_CLR	VPDMA INT0 List7 Notify Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
14	VPDMA_INT0_LIST7_COMPLETE_ENA_CLR	VPDMA INT0 List7 Complete Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
13	VPDMA_INT0_LIST6_NOTIFY_ENA_CLR	VPDMA INT0 List6 Notify Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
12	VPDMA_INT0_LIST6_COMPLETE_ENA_CLR	VPDMA INT0 List6 Complete Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
11	VPDMA_INT0_LIST5_NOTIFY_ENA_CLR	VPDMA INT0 List5 Notify Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
10	VPDMA_INT0_LIST5_COMPLETE_ENA_CLR	VPDMA INT0 List5 Complete Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
9	VPDMA_INT0_LIST4_NOTIFY_ENA_CLR	VPDMA INT0 List4 Notify Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
8	VPDMA_INT0_LIST4_COMPLETE_ENA_CLR	VPDMA INT0 List4 Complete Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0

Bits	Field Name	Description	Type	Reset
7	VPDMA_INT0_LIST3_NOTIFY_ENA_CLR	VPDMA INT0 List3 Notify Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
6	VPDMA_INT0_LIST3_COMPLETE_ENA_CLR	VPDMA INT0 List3 Complete Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
5	VPDMA_INT0_LIST2_NOTIFY_ENA_CLR	VPDMA INT0 List2 Notify Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
4	VPDMA_INT0_LIST2_COMPLETE_ENA_CLR	VPDMA INT0 List2 Complete Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
3	VPDMA_INT0_LIST1_NOTIFY_ENA_CLR	VPDMA INT0 List1 Notify Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
2	VPDMA_INT0_LIST1_COMPLETE_ENA_CLR	VPDMA INT0 List1 Complete Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
1	VPDMA_INT0_LIST0_NOTIFY_ENA_CLR	VPDMA INT0 List0 Notify Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
0	VPDMA_INT0_LIST0_COMPLETE_ENA_CLR	VPDMA INT0 List0 Complete Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0

Table 7-86. Register Call Summary for Register VIP_INTC_INTR0_ENA_CLR0

VIP Register Manual

- VIP Top Level Register Summary: [2]

Table 7-87. VIP_INTC_INTR0_ENA_CLR1

Address Offset	0x0000 003C		
Physical Address	0x4897 003C	**Instance**	VIP_top_level
Description	INTC INTR0 Interrupt Enable/Clear Register 1. This register contains the interrupt enable status/clear as defined in HL0.8		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED						VIP2_CHR_DS_2_UV_ERR_INT_ENA_CLR	VIP2_CHR_DS_1_UV_ERR_INT_ENA_CLR	VIP1_CHR_DS_2_UV_ERR_INT_ENA_CLR	VIP1_CHR_DS_1_UV_ERR_INT_ENA_CLR	RESERVED														VPDMA_INT0_CLIENT_ENA_CLR	VPDMA_INT0_CHANNEL_GROUP6_ENA_CLR	VPDMA_INT0_CHANNEL_GROUP5_ENA_CLR	VPDMA_INT0_CHANNEL_GROUP4_ENA_CLR	VPDMA_INT0_CHANNEL_GROUP3_ENA_CLR	VPDMA_INT0_CHANNEL_GROUP2_ENA_CLR	VPDMA_INT0_CHANNEL_GROUP1_ENA_CLR	VPDMA_INT0_CHANNEL_GROUP0_ENA_CLR

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	VIP2_CHR_DS_2_UV_ERR_INT_ENA_CLR	VIP2 Chroma Downsampler 2 UV Error Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
24	VIP2_CHR_DS_1_UV_ERR_INT_ENA_CLR	VIP2 Chroma Downsampler 1 UV Error Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
23	VIP1_CHR_DS_2_UV_ERR_INT_ENA_CLR	VIP1 Chroma Downsampler 2 UV Error Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
22	VIP1_CHR_DS_1_UV_ERR_INT_ENA_CLR	VIP1 Chroma Downsampler 1 UV Error Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
21:8	RESERVED		R	0x0
7	VPDMA_INT0_CLIENT_ENA_CLR	VPDMA INT0 Client Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
6	VPDMA_INT0_CHANNEL_GROUP6_ENA_CLR	VPDMA INT0 Channel Group6 Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
5	VPDMA_INT0_CHANNEL_GROUP5_ENA_CLR	VPDMA INT0 Channel Group5 Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
4	VPDMA_INT0_CHANNEL_GROUP4_ENA_CLR	VPDMA INT0 Channel Group4 Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
3	VPDMA_INT0_CHANNEL_GROUP3_ENA_CLR	VPDMA INT0 Channel Group3 Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
2	VPDMA_INT0_CHANNEL_GROUP2_ENA_CLR	VPDMA INT0 Channel Group2 Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0

Bits	Field Name	Description	Type	Reset
1	VPDMA_INT0_CHANNEL_GROUP1_ENA_CLR	VPDMA INT0 Channel Group1 Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
0	VPDMA_INT0_CHANNEL_GROUP0_ENA_CLR	VPDMA INT0 Channel Group0 Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0

Table 7-88. Register Call Summary for Register VIP_INTC_INTR0_ENA_CLR1

VIP Register Manual
- VIP Top Level Register Summary: [2]

Table 7-89. VIP_INTC_INTR1_STATUS_RAW0

Address Offset	0x0000 0040		
Physical Address	0x4897 0040	**Instance**	VIP_top_level
Description	INTC intr1 Interrupt Status Raw/Set Register 0. This register contains the raw interrupt status as defined in HL0.8		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED										VIP2_PARSER_INT_RAW	VIP1_PARSER_INT_RAW	RESERVED			VPDMA_INT1_DESCRIPTOR_RAW	VPDMA_INT1_LIST7_NOTIFY_RAW	VPDMA_INT1_LIST7_COMPLETE_RAW	VPDMA_INT1_LIST6_NOTIFY_RAW	VPDMA_INT1_LIST6_COMPLETE_RAW	VPDMA_INT1_LIST5_NOTIFY_RAW	VPDMA_INT1_LIST5_COMPLETE_RAW	VPDMA_INT1_LIST4_NOTIFY_RAW	VPDMA_INT1_LIST4_COMPLETE_RAW	VPDMA_INT1_LIST3_NOTIFY_RAW	VPDMA_INT1_LIST3_COMPLETE_RAW	VPDMA_INT1_LIST2_NOTIFY_RAW	VPDMA_INT1_LIST2_COMPLETE_RAW	VPDMA_INT1_LIST1_NOTIFY_RAW	VPDMA_INT1_LIST1_COMPLETE_RAW	VPDMA_INT1_LIST0_NOTIFY_RAW	VPDMA_INT1_LIST0_COMPLETE_RAW

Bits	Field Name	Description	Type	Reset
31:22	RESERVED		R	0x0
21	VIP2_PARSER_INT_RAW	VIP2 Parser Interrupt Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
20	VIP1_PARSER_INT_RAW	VIP1 Parser Interrupt Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
19:17	RESERVED		R	0x0
16	VPDMA_INT1_DESCRIPTOR_RAW	VPDMA INT1 Descriptor Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
15	VPDMA_INT1_LIST7_NOTIFY_RAW	VPDMA INT1 List7 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
14	VPDMA_INT1_LIST7_COMPLETE_RAW	VPDMA INT1 List7 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0

Bits	Field Name	Description	Type	Reset
13	VPDMA_INT1_LIST6_NOTIFY_RAW	VPDMA INT1 List6 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
12	VPDMA_INT1_LIST6_COMPLETE_RAW	VPDMA INT1 List6 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
11	VPDMA_INT1_LIST5_NOTIFY_RAW	VPDMA INT1 List5 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
10	VPDMA_INT1_LIST5_COMPLETE_RAW	VPDMA INT1 List5 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
9	VPDMA_INT1_LIST4_NOTIFY_RAW	VPDMA INT1 List4 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
8	VPDMA_INT1_LIST4_COMPLETE_RAW	VPDMA INT1 List4 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
7	VPDMA_INT1_LIST3_NOTIFY_RAW	VPDMA INT1 List3 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
6	VPDMA_INT1_LIST3_COMPLETE_RAW	VPDMA INT1 List3 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
5	VPDMA_INT1_LIST2_NOTIFY_RAW	VPDMA INT1 List2 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
4	VPDMA_INT1_LIST2_COMPLETE_RAW	VPDMA INT1 List2 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
3	VPDMA_INT1_LIST1_NOTIFY_RAW	VPDMA INT1 List1 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
2	VPDMA_INT1_LIST1_COMPLETE_RAW	VPDMA INT1 List1 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
1	VPDMA_INT1_LIST0_NOTIFY_RAW	VPDMA INT1 List0 Notify Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
0	VPDMA_INT1_LIST0_COMPLETE_RAW	VPDMA INT1 List0 Complete Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0

Table 7-90. Register Call Summary for Register VIP_INTC_INTR1_STATUS_RAW0

VIP Register Manual

- VIP Top Level Register Summary: [2]

Table 7-91. VIP_INTC_INTR1_STATUS_RAW1

Address Offset	0x0000 0044		
Physical Address	0x4897 0044	**Instance**	VIP_top_level
Description	INTC intr1 Interrupt Status Raw/Set Register 1. This register contains the raw interrupt status as defined in HL0.8		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED						VIP2_CHR_DS_2_UV_ERR_INT_RAW	VIP2_CHR_DS_1_UV_ERR_INT_RAW	VIP1_CHR_DS_2_UV_ERR_INT_RAW	VIP1_CHR_DS_1_UV_ERR_INT_RAW	RESERVED														VPDMA_INT1_CLIENT_RAW	RESERVED	VPDMA_INT1_CHANNEL_GROUP5_RAW	VPDMA_INT1_CHANNEL_GROUP4_RAW	VPDMA_INT1_CHANNEL_GROUP3_RAW	VPDMA_INT1_CHANNEL_GROUP2_RAW	VPDMA_INT1_CHANNEL_GROUP1_RAW	VPDMA_INT1_CHANNEL_GROUP0_RAW

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	VIP2_CHR_DS_2_UV_ERR_INT_RAW	VIP2 Chroma Downsampler 2 UV Error Interrupt Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
24	VIP2_CHR_DS_1_UV_ERR_INT_RAW	VIP2 Chroma Downsampler 1 UV Error Interrupt Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
23	VIP1_CHR_DS_2_UV_ERR_INT_RAW	VIP1 Chroma Downsampler 2 UV Error Interrupt Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
22	VIP1_CHR_DS_1_UV_ERR_INT_RAW	VIP1 Chroma Downsampler 1 UV Error Interrupt Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
21:8	RESERVED		R	0x0
7	VPDMA_INT1_CLIENT_RAW	VPDMA INT1 Client Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
6	RESERVED		R	0x0
5	VPDMA_INT1_CHANNEL_GROUP5_RAW	VPDMA INT1 Channel Group5 Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
4	VPDMA_INT1_CHANNEL_GROUP4_RAW	VPDMA INT1 Channel Group4 Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
3	VPDMA_INT1_CHANNEL_GROUP3_RAW	VPDMA INT1 Channel Group3 Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
2	VPDMA_INT1_CHANNEL_GROUP2_RAW	VPDMA INT1 Channel Group2 Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
1	VPDMA_INT1_CHANNEL_GROUP1_RAW	VPDMA INT1 Channel Group1 Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
0	VPDMA_INT1_CHANNEL_GROUP0_RAW	VPDMA INT1 Channel Group0 Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0

Table 7-92. Register Call Summary for Register VIP_INTC_INTR1_STATUS_RAW1

VIP Register Manual

- VIP Top Level Register Summary: [2]

Table 7-93. VIP_INTC_INTR1_STATUS_ENA0

Address Offset	0x0000 0048		
Physical Address	0x4897 0048	**Instance**	VIP_top_level
Description	INTC intr1 Interrupt Status Enabled/Clear Register 0. This register contains the enabled interrupt status as defined in HL0.8		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED										VIP2_PARSER_INT_ENA	VIP1_PARSER_INT_ENA	RESERVED			VPDMA_INT1_DESCRIPTOR_ENA	VPDMA_INT1_LIST7_NOTIFY_ENA	VPDMA_INT1_LIST7_COMPLETE_ENA	VPDMA_INT1_LIST6_NOTIFY_ENA	VPDMA_INT1_LIST6_COMPLETE_ENA	VPDMA_INT1_LIST5_NOTIFY_ENA	VPDMA_INT1_LIST5_COMPLETE_ENA	VPDMA_INT1_LIST4_NOTIFY_ENA	VPDMA_INT1_LIST4_COMPLETE_ENA	VPDMA_INT1_LIST3_NOTIFY_ENA	VPDMA_INT1_LIST3_COMPLETE_ENA	VPDMA_INT1_LIST2_NOTIFY_ENA	VPDMA_INT1_LIST2_COMPLETE_ENA	VPDMA_INT1_LIST1_NOTIFY_ENA	VPDMA_INT1_LIST1_COMPLETE_ENA	VPDMA_INT1_LIST0_NOTIFY_ENA	VPDMA_INT1_LIST0_COMPLETE_ENA

Bits	Field Name	Description	Type	Reset
31:22	RESERVED		R	0x0
21	VIP2_PARSER_INT_ENA	VIP2 Parser Enabled Interrupt Status Read indicates enabled status 0 = inactive 1 = active Writing 1 will clear interrupt Writing 0 has no effect	RW	0x0
20	VIP1_PARSER_INT_ENA	VIP1 Parser Enabled Interrupt Status Read indicates enabled status 0 = inactive 1 = active Writing 1 will clear interrupt Writing 0 has no effect	RW	0x0
19:17	RESERVED		R	0x0
16	VPDMA_INT1_DESCRIPTOR_ENA	VPDMA INT1 Descriptor Enabled Status Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
15	VPDMA_INT1_LIST7_NOTIFY_ENA	VPDMA INT1 List7 Notify Enabled Status Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
14	VPDMA_INT1_LIST7_COMPLETE_ENA	VPDMA INT1 List7 Complete Enabled Status Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
13	VPDMA_INT1_LIST6_NOTIFY_ENA	VPDMA INT1 List6 Notify Enabled Status Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
12	VPDMA_INT1_LIST6_COMPLETE_ENA	VPDMA INT1 List6 Complete Enabled Status Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
11	VPDMA_INT1_LIST5_NOTIFY_ENA	VPDMA INT1 List5 Notify Enabled Status Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
10	VPDMA_INT1_LIST5_COMPLETE_ENA	VPDMA INT1 List5 Complete Enabled Statust Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
9	VPDMA_INT1_LIST4_NOTIFY_ENA	VPDMA INT1 List4 Notify Enabled Statust Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
8	VPDMA_INT1_LIST4_COMPLETE_ENA	VPDMA INT1 List4 Complete Enabled Status Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0

Bits	Field Name	Description	Type	Reset
7	VPDMA_INT1_LIST3_NOTIFY_ENA	VPDMA INT1 List3 Notify Enabled Status Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
6	VPDMA_INT1_LIST3_COMPLETE_ENA	VPDMA INT1 List3 Complete Enabled Status Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
5	VPDMA_INT1_LIST2_NOTIFY_ENA	VPDMA INT1 List2 Notify Enabled Statust Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
4	VPDMA_INT1_LIST2_COMPLETE_ENA	VPDMA INT1 List2 Complete Enabled Statust Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
3	VPDMA_INT1_LIST1_NOTIFY_ENA	VPDMA INT1 List1 Notify Enabled Status Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
2	VPDMA_INT1_LIST1_COMPLETE_ENA	VPDMA INT1 List1 Complete Enabled Status Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
1	VPDMA_INT1_LIST0_NOTIFY_ENA	VPDMA INT1 List0 Notify Enabled Status Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
0	VPDMA_INT1_LIST0_COMPLETE_ENA	VPDMA INT1 List0 Complete Enabled Status Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0

Table 7-94. Register Call Summary for Register VIP_INTC_INTR1_STATUS_ENA0

VIP Register Manual
- VIP Top Level Register Summary: [2]

Table 7-95. VIP_INTC_INTR1_STATUS_ENA1

Address Offset	0x0000 004C		
Physical Address	0x4897 004C	**Instance**	VIP_top_level
Description	INTC intr1 Interrupt Status Enabled/Clear Register 1. This register contains the enabled interrupt status as defined in HL0.8		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED						VIP2_CHR_DS_2_UV_ERR_INT_ENA	VIP2_CHR_DS_1_UV_ERR_INT_ENA	VIP1_CHR_DS_2_UV_ERR_INT_ENA	VIP1_CHR_DS_1_UV_ERR_INT_ENA	RESERVED														VPDMA_INT1_CLIENT_ENA	RESERVED	VPDMA_INT1_CHANNEL_GROUP5_ENA	VPDMA_INT1_CHANNEL_GROUP4_ENA	VPDMA_INT1_CHANNEL_GROUP3_ENA	VPDMA_INT1_CHANNEL_GROUP2_ENA	VPDMA_INT1_CHANNEL_GROUP1_ENA	VPDMA_INT1_CHANNEL_GROUP0_ENA

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	VIP2_CHR_DS_2_UV_ERR_INT_ENA	VIP2 Chroma Downsampler 2 UV Error Enabled Interrupt Status Read indicates enabled status 0 = inactive 1 = active Writing 1 will clear interrupt Writing 0 has no effect	RW	0x0
24	VIP2_CHR_DS_1_UV_ERR_INT_ENA	VIP2 Chroma Downsampler 1 UV Error Enabled Interrupt Status Read indicates enabled status 0 = inactive 1 = active Writing 1 will clear interrupt Writing 0 has no effect	RW	0x0
23	VIP1_CHR_DS_2_UV_ERR_INT_ENA	VIP1 Chroma Downsampler 2 UV Error Enabled Interrupt Status Read indicates enabled status 0 = inactive 1 = active Writing 1 will clear interrupt Writing 0 has no effect	RW	0x0
22	VIP1_CHR_DS_1_UV_ERR_INT_ENA	VIP1 Chroma Downsampler 1 UV Error Enabled Interrupt Status Read indicates enabled status 0 = inactive 1 = active Writing 1 will clear interrupt Writing 0 has no effect	RW	0x0
21:8	RESERVED		R	0x0
7	VPDMA_INT1_CLIENT_ENA	VPDMA INT1 Client Enabled Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
6	RESERVED		R	0x0
5	VPDMA_INT1_CHANNEL_GROUP5_ENA	VPDMA INT1 Channel Group5 Enabled Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
4	VPDMA_INT1_CHANNEL_GROUP4_ENA	VPDMA INT1 Channel Group4 Enabled Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
3	VPDMA_INT1_CHANNEL_GROUP3_ENA	VPDMA INT1 Channel Group3 Enabled Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
2	VPDMA_INT1_CHANNEL_GROUP2_ENA	VPDMA INT1 Channel Group3 Enabled Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
1	VPDMA_INT1_CHANNEL_GROUP1_ENA	VPDMA INT1 Channel Group1 Enabled Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0
0	VPDMA_INT1_CHANNEL_GROUP0_ENA	VPDMA INT1 Channel Group0 Enabled Status Read indicates raw status 0 = inactive 1 = active Writing 1 will set status Writing 0 has no effect	RW	0x0

Table 7-96. Register Call Summary for Register VIP_INTC_INTR1_STATUS_ENA1

VIP Register Manual

- VIP Top Level Register Summary: [2]

Table 7-97. VIP_INTC_INTR1_ENA_SET0

Address Offset	0x0000 0050		
Physical Address	0x4897 0050	**Instance**	VIP_top_level
Description	INTC intr1 Interrupt Enable/Set Register 0. This register contains the interrupt enable status/set as defined in HL0.8		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED										VIP2_PARSER_INT_ENA_SET	VIP1_PARSER_INT_ENA_SET	RESERVED			VPDMA_INT1_DESCRIPTOR_ENA_SET	VPDMA_INT1_LIST7_NOTIFY_ENA_SET	VPDMA_INT1_LIST7_COMPLETE_ENA_SET	VPDMA_INT1_LIST6_NOTIFY_ENA_SET	VPDMA_INT1_LIST6_COMPLETE_ENA_SET	VPDMA_INT1_LIST5_NOTIFY_ENA_SET	VPDMA_INT1_LIST5_COMPLETE_ENA_SET	VPDMA_INT1_LIST4_NOTIFY_ENA_SET	VPDMA_INT1_LIST4_COMPLETE_ENA_SET	VPDMA_INT1_LIST3_NOTIFY_ENA_SET	VPDMA_INT1_LIST3_COMPLETE_ENA_SET	VPDMA_INT1_LIST2_NOTIFY_ENA_SET	VPDMA_INT1_LIST2_COMPLETE_ENA_SET	VPDMA_INT1_LIST1_NOTIFY_ENA_SET	VPDMA_INT1_LIST1_COMPLETE_ENA_SET	VPDMA_INT1_LIST0_NOTIFY_ENA_SET	VPDMA_INT1_LIST0_COMPLETE_ENA_SET

Bits	Field Name	Description	Type	Reset
31:22	RESERVED		R	0x0
21	VIP2_PARSER_INT_ENA_SET	VIP2 Parser Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
20	VIP1_PARSER_INT_ENA_SET	VIP1 Parser Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
19:17	RESERVED		R	0x0
16	VPDMA_INT1_DESCRIPTOR_ENA_SET	VPDMA INT1 Descriptor Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
15	VPDMA_INT1_LIST7_NOTIFY_ENA_SET	VPDMA INT1 List7 Notify Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
14	VPDMA_INT1_LIST7_COMPLETE_ENA_SET	VPDMA INT1 List7 Complete Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
13	VPDMA_INT1_LIST6_NOTIFY_ENA_SET	VPDMA INT1 List6 Notify Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
12	VPDMA_INT1_LIST6_COMPLETE_ENA_SET	VPDMA INT1 List6 Complete Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
11	VPDMA_INT1_LIST5_NOTIFY_ENA_SET	VPDMA INT1 List5 Notify Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
10	VPDMA_INT1_LIST5_COMPLETE_ENA_SET	VPDMA INT1 List5 Complete Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
9	VPDMA_INT1_LIST4_NOTIFY_ENA_SET	VPDMA INT1 List4 Notify Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
8	VPDMA_INT1_LIST4_COMPLETE_ENA_SET	VPDMA INT1 List4 Complete Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0

Bits	Field Name	Description	Type	Reset
7	VPDMA_INT1_LIST3_NOTIFY_ENA_SET	VPDMA INT1 List3 Notify Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
6	VPDMA_INT1_LIST3_COMPLETE_ENA_SET	VPDMA INT1 List3 Complete Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
5	VPDMA_INT1_LIST2_NOTIFY_ENA_SET	VPDMA INT1 List2 Notify Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
4	VPDMA_INT1_LIST2_COMPLETE_ENA_SET	VPDMA INT1 List2 Complete Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
3	VPDMA_INT1_LIST1_NOTIFY_ENA_SET	VPDMA INT1 List1 Notify Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
2	VPDMA_INT1_LIST1_COMPLETE_ENA_SET	VPDMA INT1 List1 Complete Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
1	VPDMA_INT1_LIST0_NOTIFY_ENA_SET	VPDMA INT1 List0 Notify Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
0	VPDMA_INT1_LIST0_COMPLETE_ENA_SET	VPDMA INT1 List0 Complete Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0

Table 7-98. Register Call Summary for Register VIP_INTC_INTR1_ENA_SET0

VIP Register Manual

- VIP Top Level Register Summary: [2]

Table 7-99. VIP_INTC_INTR1_ENA_SET1

Address Offset	0x0000 0054		
Physical Address	0x4897 0054	**Instance**	VIP_top_level
Description	INTC intr1 Interrupt Enable/Set Register 1. This register contains the interrupt enable status/set as defined in HL0.8		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED						VIP2_CHR_DS_2_UV_ERR_INT_ENA_SET	VIP2_CHR_DS_1_UV_ERR_INT_ENA_SET	VIP1_CHR_DS_2_UV_ERR_INT_ENA_SET	VIP1_CHR_DS_1_UV_ERR_INT_ENA_SET	RESERVED														VPDMA_INT1_CLIENT_ENA_SET	VPDMA_INT1_CHANNEL_GROUP6_ENA_SET	VPDMA_INT1_CHANNEL_GROUP5_ENA_SET	VPDMA_INT1_CHANNEL_GROUP4_ENA_SET	VPDMA_INT1_CHANNEL_GROUP3_ENA_SET	VPDMA_INT1_CHANNEL_GROUP2_ENA_SET	VPDMA_INT1_CHANNEL_GROUP1_ENA_SET	VPDMA_INT1_CHANNEL_GROUP0_ENA_SET

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	VIP2_CHR_DS_2_UV_ERR_INT_ENA_SET	VIP2 Chroma Downsampler 2 UV Error Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
24	VIP2_CHR_DS_1_UV_ERR_INT_ENA_SET	VIP2 Chroma Downsampler 1 UV Error Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
23	VIP1_CHR_DS_2_UV_ERR_INT_ENA_SET	VIP1 Chroma Downsampler 2 UV Error Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
22	VIP1_CHR_DS_1_UV_ERR_INT_ENA_SET	VIP1 Chroma Downsampler 1 UV Error Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
21:8	RESERVED		R	0x0
7	VPDMA_INT1_CLIENT_ENA_SET	VPDMA INT1 Client Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
6	VPDMA_INT1_CHANNEL_GROUP6_ENA_SET	VPDMA INT1 Channel Group6 Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
5	VPDMA_INT1_CHANNEL_GROUP5_ENA_SET	VPDMA INT1 Channel Group5 Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
4	VPDMA_INT1_CHANNEL_GROUP4_ENA_SET	VPDMA INT1 Channel Group4 Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
3	VPDMA_INT1_CHANNEL_GROUP3_ENA_SET	VPDMA INT1 Channel Group3 Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
2	VPDMA_INT1_CHANNEL_GROUP2_ENA_SET	VPDMA INT1 Channel Group2 Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0

Bits	Field Name	Description	Type	Reset
1	VPDMA_INT1_CHANNEL_GROUP1_ENA_SET	VPDMA INT1 Channel Group1 Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0
0	VPDMA_INT1_CHANNEL_GROUP0_ENA_SET	VPDMA INT1 Channel Group0 Enable/Set Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will set interrupt enabled Writing 0 has no effect	RW	0x0

Table 7-100. Register Call Summary for Register VIP_INTC_INTR1_ENA_SET1

VIP Register Manual
- VIP Top Level Register Summary: [2]

Table 7-101. VIP_INTC_INTR1_ENA_CLR0

Address Offset	0x0000 0058		
Physical Address	0x4897 0058	**Instance**	VIP_top_level
Description	INTC intr1 Interrupt Enable/Clear Register 0. This register contains the interrupt enable status/clear as defined in HL0.8		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED										VIP2_PARSER_INT_ENA_CLR	VIP1_PARSER_INT_ENA_CLR	RESERVED			VPDMA_INT1_DESCRIPTOR_ENA_CLR	VPDMA_INT1_LIST7_NOTIFY_ENA_CLR	VPDMA_INT1_LIST7_COMPLETE_ENA_CLR	VPDMA_INT1_LIST6_NOTIFY_ENA_CLR	VPDMA_INT1_LIST6_COMPLETE_ENA_CLR	VPDMA_INT1_LIST5_NOTIFY_ENA_CLR	VPDMA_INT1_LIST5_COMPLETE_ENA_CLR	VPDMA_INT1_LIST4_NOTIFY_ENA_CLR	VPDMA_INT1_LIST4_COMPLETE_ENA_CLR	VPDMA_INT1_LIST3_NOTIFY_ENA_CLR	VPDMA_INT1_LIST3_COMPLETE_ENA_CLR	VPDMA_INT1_LIST2_NOTIFY_ENA_CLR	VPDMA_INT1_LIST2_COMPLETE_ENA_CLR	VPDMA_INT1_LIST1_NOTIFY_ENA_CLR	VPDMA_INT1_LIST1_COMPLETE_ENA_CLR	VPDMA_INT1_LIST0_NOTIFY_ENA_CLR	VPDMA_INT1_LIST0_COMPLETE_ENA_CLR

Bits	Field Name	Description	Type	Reset
31:22	RESERVED		R	0x0
21	VIP2_PARSER_INT_ENA_CLR	VIP2 Parser Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
20	VIP1_PARSER_INT_ENA_CLR	VIP1 Parser Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
19:17	RESERVED		R	0x0
16	VPDMA_INT1_DESCRIPTOR_ENA_CLR	VPDMA INT1 Descriptor Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
15	VPDMA_INT1_LIST7_NOTIFY_ENA_CLR	VPDMA INT1 List7 Notify Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0

Bits	Field Name	Description	Type	Reset
14	VPDMA_INT1_LIST7_COMPLETE_ENA_CLR	VPDMA INT1 List7 Complete Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
13	VPDMA_INT1_LIST6_NOTIFY_ENA_CLR	VPDMA INT1 List6 Notify Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
12	VPDMA_INT1_LIST6_COMPLETE_ENA_CLR	VPDMA INT1 List6 Complete Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
11	VPDMA_INT1_LIST5_NOTIFY_ENA_CLR	VPDMA INT1 List5 Notify Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
10	VPDMA_INT1_LIST5_COMPLETE_ENA_CLR	VPDMA INT1 List5 Complete Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
9	VPDMA_INT1_LIST4_NOTIFY_ENA_CLR	VPDMA INT1 List4 Notify Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
8	VPDMA_INT1_LIST4_COMPLETE_ENA_CLR	VPDMA INT1 List4 Complete Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
7	VPDMA_INT1_LIST3_NOTIFY_ENA_CLR	VPDMA INT1 List3 Notify Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
6	VPDMA_INT1_LIST3_COMPLETE_ENA_CLR	VPDMA INT1 List3 Complete Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
5	VPDMA_INT1_LIST2_NOTIFY_ENA_CLR	VPDMA INT1 List2 Notify Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
4	VPDMA_INT1_LIST2_COMPLETE_ENA_CLR	VPDMA INT1 List2 Complete Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
3	VPDMA_INT1_LIST1_NOTIFY_ENA_CLR	VPDMA INT1 List1 Notify Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
2	VPDMA_INT1_LIST1_COMPLETE_ENA_CLR	VPDMA INT1 List1 Complete Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
1	VPDMA_INT1_LIST0_NOTIFY_ENA_CLR	VPDMA INT1 List0 Notify Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
0	VPDMA_INT1_LIST0_COMPLETE_ENA_CLR	VPDMA INT1 List0 Complete Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0

Table 7-102. Register Call Summary for Register VIP_INTC_INTR1_ENA_CLR0

VIP Register Manual

- VIP Top Level Register Summary: [2]

Table 7-103. VIP_INTC_INTR1_ENA_CLR1

Address Offset	0x0000 005C		
Physical Address	0x4897 005C	**Instance**	VIP_top_level
Description	INTC intr1 Interrupt Enable/Clear Register 1. This register contains the interrupt enable status/clear as defined in HL0.8		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED						VIP2_CHR_DS_2_UV_ERR_INT_ENA_CLR	VIP2_CHR_DS_1_UV_ERR_INT_ENA_CLR	VIP1_CHR_DS_2_UV_ERR_INT_ENA_CLR	VIP1_CHR_DS_1_UV_ERR_INT_ENA_CLR	RESERVED														VPDMA_INT1_CLIENT_ENA_CLR	VPDMA_INT1_CHANNEL_GROUP6_ENA_CLR	VPDMA_INT1_CHANNEL_GROUP5_ENA_CLR	VPDMA_INT1_CHANNEL_GROUP4_ENA_CLR	VPDMA_INT1_CHANNEL_GROUP3_ENA_CLR	VPDMA_INT1_CHANNEL_GROUP2_ENA_CLR	VPDMA_INT1_CHANNEL_GROUP1_ENA_CLR	VPDMA_INT1_CHANNEL_GROUP0_ENA_CLR

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	VIP2_CHR_DS_2_UV_ERR_INT_ENA_CLR	VIP2 Chroma Downsampler 2 UV Error Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
24	VIP2_CHR_DS_1_UV_ERR_INT_ENA_CLR	VIP2 Chroma Downsampler 1 UV Error Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
23	VIP1_CHR_DS_2_UV_ERR_INT_ENA_CLR	VIP1 Chroma Downsampler 2 UV Error Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
22	VIP1_CHR_DS_1_UV_ERR_INT_ENA_CLR	VIP1 Chroma Downsampler 1 UV Error Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
21:8	RESERVED		R	0x0
7	VPDMA_INT1_CLIENT_ENA_CLR	VPDMA INT1 Client Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
6	VPDMA_INT1_CHANNEL_GROUP6_ENA_CLR	VPDMA INT1 Channel Group6 Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0

Bits	Field Name	Description	Type	Reset
5	VPDMA_INT1_CHANNEL_GROUP5_ENA_CLR	VPDMA INT1 Channel Group5 Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
4	VPDMA_INT1_CHANNEL_GROUP4_ENA_CLR	VPDMA INT1 Channel Group4 Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
3	VPDMA_INT1_CHANNEL_GROUP3_ENA_CLR	VPDMA INT1 Channel Group3 Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
2	VPDMA_INT1_CHANNEL_GROUP2_ENA_CLR	VPDMA INT1 Channel Group2 Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
1	VPDMA_INT1_CHANNEL_GROUP1_ENA_CLR	VPDMA INT1 Channel Group1 Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0
0	VPDMA_INT1_CHANNEL_GROUP0_ENA_CLR	VPDMA INT1 Channel Group0 Enable/Clear Read indicates interrupt enable 0 = disabled 1 = enabled Writing 1 will clear interrupt enabled Writing 0 has no effect	RW	0x0

Table 7-104. Register Call Summary for Register VIP_INTC_INTR1_ENA_CLR1

VIP Register Manual
- VIP Top Level Register Summary: [2]

Table 7-105. VIP_INTC_EOI

Address Offset	0x0000 00A0		
Physical Address	0x4897 00A0	**Instance**	VIP_top_level
Description	INTC EOI Register. This register contains the EOI vector register contents as defined by HL0.8		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EOI_VECTOR																															

Bits	Field Name	Description	Type	Reset
31:0	EOI_VECTOR	Number associated with the ipgenericirq for intr output. There are 4 interrupt outputs Write 0x0 : Write to intr0 IP Generic Write 0x1 : Write to intr1 IP Generic Write 0x2 : Write to intr2 IP Generic Write 0x3 : Write to intr3 IP Generic Any other write value is ignored.	RW	0x0

Table 7-106. Register Call Summary for Register VIP_INTC_EOI

VIP Register Manual
- VIP Top Level Register Summary: [2]

Table 7-107. VIP_CLKC_CLKEN

Address Offset	0x0000 0100		
Physical Address	0x4897 0100	**Instance**	VIP_top_level
Description	CLKC Module Clock Enable Register. This register contains clock enables for the processing paths in the VIP module.		

Table 7-107. VIP_CLKC_CLKEN (continued)

Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														VIP2_DP_EN	VIP1_DP_EN	RESERVED															VPDMA_EN

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17	VIP2_DP_EN	VIP Slice1 Data Path Clock Enable, 1 = Clock Enabled, 0 = Clock Disabled	RW	0x0
16	VIP1_DP_EN	VIP Slice0 Data Path Clock Enable, 1 = Clock Enabled, 0 = Clock Disabled	RW	0x0
15:1	RESERVED		R	0x0
0	VPDMA_EN	VPDMA Clock Enable, 1 = Clock Enabled, 0 = Clock Disabled	RW	0x0

Table 7-108. Register Call Summary for Register VIP_CLKC_CLKEN

VIP Functional Description
- VIP Clocks: [0][1][2]

VIP Register Manual
- VIP Top Level Register Summary: [5]

Table 7-109. VIP_CLKC_RST

Address Offset	0x0000 0104		
Physical Address	0x4897 0104	**Instance**	VIP_top_level
Description	CLKC Module Reset Register. This register contains resets for the processing paths in the VIP module.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MAIN_RST	RESERVED		S1_CHR_DS_1_RST	S0_CHR_DS_1_RST	S1_CHR_DS_0_RST	S0_CHR_DS_0_RST	RESERVED	S1_SC_RST	S0_SC_RST	S1_CSC_RST	S0_CSC_RST	S1_PARSER_RST	S0_PARSER_RST	VIP2_DP_RST	VIP1_DP_RST	RESERVED															VPDMA_RST

Bits	Field Name	Description	Type	Reset
31	MAIN_RST	Reset for all modules in VIP Main Data Path	RW	0x0
30:29	RESERVED	Reserved	R	0x0
28	S1_CHR_DS_1_RST	VIP Slice1 CHRDS1 reset	RW	0x0
27	S0_CHR_DS_1_RST	VIP Slice0 CHRDS1 reset	RW	0x0
26	S1_CHR_DS_0_RST	VIP Slice1 CHRDS0 reset	RW	0x0
25	S0_CHR_DS_0_RST	VIP Slice0 CHRDS0 reset	RW	0x0
24	RESERVED	Reserved	RW	0x0

Bits	Field Name	Description	Type	Reset
23	S1_SC_RST	VIP Slice1 SC reset	RW	0x0
22	S0_SC_RST	VIP Slice0 SC reset	RW	0x0
21	S1_CSC_RST	VIP Slice1 CSC reset	RW	0x0
20	S0_CSC_RST	VIP Slice0 CSC reset	RW	0x0
19	S1_PARSER_RST	VIP Slice1 parser reset	RW	0x0
18	S0_PARSER_RST	VIP Slice0 parser reset	RW	0x0
17	VIP2_DP_RST	VIP Slice1 Data Path Reset	RW	0x0
16	VIP1_DP_RST	VIP Slice0 Data Path Reset	RW	0x0
15:1	RESERVED	Reserved	R	0x0000
0	VPDMA_RST	VPDMA Reset	RW	0x0

Table 7-110. Register Call Summary for Register VIP_CLKC_RST

VIP Functional Description
- VIP Software Reset: [0][1][2][3]
- VIP Overflow Detection and Recovery: [4][5]

VIP Register Manual
- VIP Top Level Register Summary: [8]

Table 7-111. VIP_CLKC_DPS

Address Offset	0x0000 0108		
Physical Address	0x4897 0108	**Instance**	VIP_top_level
Description	CLKC Main Data Path Select Register. This register selects the various data paths within main portion (non-VIP) of the subsystem		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MAIN_RST	RESERVED													VIP2_DP_RST	VIP1_DP_RST	RESERVED															VPDMA_RST

Bits	Field Name	Description	Type	Reset
31	MAIN_RST	Reset for all modules in DSS Main Data Path	RW	0x0
30:18	RESERVED		R	0x0
17	VIP2_DP_RST	Video Input Port 2 Data Path Reset	RW	0x0
16	VIP1_DP_RST	Video Input Port 1 Data Path Reset	RW	0x0
15:1	RESERVED		R	0x0
0	VPDMA_RST	VPDMA Reset	RW	0x0

Table 7-112. Register Call Summary for Register VIP_CLKC_DPS

VIP Register Manual
- VIP Top Level Register Summary: [2]

Table 7-113. VIP_CLKC_VIP0DPS

Address Offset	0x0000 010C		
Physical Address	0x4897 010C	**Instance**	VIP_top_level
Description	CLKC Video Input Port 1 Data Path Select Register. This register selects the various data paths within the Video Input Port portion of the subsystem		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VIP1_DATAPATH_SELECT				VIP1_TESTPORT_A_SELECT	VIP1_TESTPORT_B_SELECT	RESERVED								VIP1_CHR_DS_2_BYPASS	VIP1_CHR_DS_1_BYPASS	VIP1_MULTI_CHANNEL_SELECT				VIP1_CHR_DS_2_SRC_SELECT				VIP1_CHR_DS_1_SRC_SELECT				VIP1_RGB_OUT_HI_SELECT				VIP1_RGB_OUT_LO_SELECT	VIP1_RGB_SRC_SELECT	VIP1_SC_SRC_SELECT				VIP1_CSC_SRC_SELECT	

Bits	Field Name	Description	Type	Reset
31:28	VIP1_DATAPATH_SELECT	VIP1 Datapath Register Field Enable 0000 : All fields written 0001 : Only vip1_csc_src_select written 0010 : Only vip1_sc_src_select written 0011 : Only vip1_rgb_src_select written 0100 : Only vip1_rgb_out_lo_select written 0101 : Only vip1_rgb_out_hi_select written 0110 : Only vip1_chr_ds_1_src_select written 0111 : Only vip1_chr_ds_2_src_select written 1000 : Only vip1_multi_channel_select written 1001 : Only vip1_chr_ds_1_bypass written 1010 : Only vip1_chr_ds_2_bypass written 1011 : Reserved 1100 : Reserved 1101 : Reserved 1110 : Reserved 1111 : Reserved	RW	0x0
27	VIP1_TESTPORT_A_SELECT	0 : Normal mode 1: Test Mode	RW	0x0
26	VIP1_TESTPORT_B_SELECT	0 : Normal mode 1: Test Mode	RW	0x0
25:18	RESERVED		R	0x0
17	VIP1_CHR_DS_2_BYPASS	Video Input Port 1 Chroma Downsampler 2 Bypass 0 : VIP Chroma Downsampler 1 selected 1 : VIP Chroma Downsampler 1 Bypassed Chroma Downsampler Bypassed means the output format from the VIP will be 422 data. Selected means the output format will be 420	RW	0x0
16	VIP1_CHR_DS_1_BYPASS	Video Input Port 1 Chroma Downsampler 1 Bypass 0 : VIP Chroma Downsampler 1 selected 1 : VIP Chroma Downsampler 1 Bypassed Chroma Downsampler Bypassed means the output format from the VIP will be 422 data. Selected means the output format will be 420	RW	0x0

Bits	Field Name	Description	Type	Reset
15	VIP1_MULTI_CHANNEL_SELECT	Video Input Port 1 Multi Channel Select 0 : VIP_PARSER A and B channels operate in single channel mode 1 : VIP_PARSER A and B channels directly drive VPDMA (multi-channel case) Multi-Channel means that the A and B sources are from multiple channels and used in a multiplexed stream mode. The VIP Parser extracts the channel ID from each source and outputs this information to VPDMA, and thus to memory. When operating in a multiplexed stream mode, this bit must be set to 1 to enable the channel information to be passed to memory. If this is not set, the channel number (or source number) will be 0 for all streams. If vip1_rgb_out_select = 1, then VIP_PARSER A port is connected to VPDMA	RW	0x0
14:12	VIP1_CHR_DS_2_SRC_SELECT	Video Input Port 1 Chroma Downsampler 2 Source Select 0 : Path Disabled (no input to CHR_DS) 1 : Source from Scaler (SC_M) 2 : Source from Color Space Converter (CSC) 3 : Source from VIP_PARSER A port 4 : Source from VIP_PARSER B port 5 : Source from Transcode (422) 6 : Reserved 7 : Reserved	RW	0x0
11:9	VIP1_CHR_DS_1_SRC_SELECT	Video Input Port 1 Chroma Downsampler 1 Source Select 000 : Path Disabled (no input to CHR_DS) 001 : Source from Scaler (SC_M) 010 : Source from Color Space Converter (CSC) 011 : Source from VIP_PARSER A port 100 : Source from VIP_PARSER B port 101 : Source from Transcode (422) 110 : Reserved 111 : Reserved	RW	0x0
8	VIP1_RGB_OUT_HI_SELECT	Video Input Port 1 HI RGB Output Select 0 : Output Type is 420/422 1 : Output Type is RGB	RW	0x0
7	VIP1_RGB_OUT_LO_SELECT	Video Input Port 1 LO RGB Output Select 0 : Output Type is 420/422 1 : Output Type is RGB	RW	0x0
6	VIP1_RGB_SRC_SELECT	Video Input Port 1 RGB Output Path Select 0 : Source from Compositor RGB input 1 : Source from CSC	RW	0x0
5:3	VIP1_SC_SRC_SELECT	Video Input Port 1 SC_M Source Select 000 : Path Disabled 001 : Source from Color Space Converter (CSC) 010 : Source from VIP_PARSER A port 011 : Source from VIP_PARSER B port 100 : Source from Transcode (422) 101 : Reserved 110 : Reserved 111 : Reserved	RW	0x0
2:0	VIP1_CSC_SRC_SELECT	Video Input Port 1 CSC Source Select 000 : Path Disabled 001 : Source from VIP_PARSER A (422) port 010 : Source from VIP_PARSER B port 011 : Source from Transcode (422) 100 : Source from VIP_PARSER A (RGB) port 101 : Source from Compositor (RGB) 110 : Reserved 111 : Reserved	RW	0x0

Table 7-114. Register Call Summary for Register VIP_CLKC_VIP0DPS

VIP Functional Description

- VIP Slice Processing Path Overview: [0][1][2][3][4][5][6][7][8][9][10]
- VIP Slice Processing Path Multiplexers: [11][12]

VIP Register Manual

- VIP Top Level Register Summary: [15]

Table 7-115. VIP_CLKC_VIP1DPS

Address Offset	0x0000 0110		
Physical Address	0x4897 0110	**Instance**	VIP_top_level
Description	CLKC Video Input Port 2 Data Path Select Register. This register selects the various data paths within the Video Input Port portion of the subsystem		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VIP2_DATAPATH_SELECT				VIP2_TESTPORT_A_SELECT	VIP2_TESTPORT_B_SELECT	RESERVED								VIP2_CHR_DS_2_BYPASS	VIP2_CHR_DS_1_BYPASS	VIP2_MULTI_CHANNEL_SELECT	VIP2_CHR_DS_2_SRC_SELECT			VIP2_CHR_DS_1_SRC_SELECT			VIP2_RGB_OUT_HI_SELECT	VIP2_RGB_OUT_LO_SELECT	VIP2_RGB_SRC_SELECT	VIP2_SC_SRC_SELECT				VIP2_CSC_SRC_SELECT		

Bits	Field Name	Description	Type	Reset
31:28	VIP2_DATAPATH_SELECT	VIP2 Datapath Register Field Enable 0000 : All fields written 0001 : Only vip2_csc_src_select written 0010 : Only vip2_sc_src_select written 0011 : Only vip2_rgb_src_select written 0100 : Only vip2_rgb_out_lo_select written 0101 : Only vip2_rgb_out_hi_select written 0110 : Only vip2_chr_ds_1_src_select written 0111 : Only vip2_chr_ds_2_src_select written 1000 : Only vip2_multi_channel_select written 1001 : Only vip2_chr_ds_1_bypass written 1010 : Only vip2_chr_ds_2_bypass written 1011 : Reserved 1100 : Reserved 1101 : Reserved 1110 : Reserved 1111 : Reserved	RW	0x0
27	VIP2_TESTPORT_A_SELECT	0 : Normal mode 1: Test Mode	RW	0x0
26	VIP2_TESTPORT_B_SELECT	0 : Normal mode 1: Test Mode	RW	0x0
25:18	RESERVED		R	0x0
17	VIP2_CHR_DS_2_BYPASS	Video Input Port 2 Chroma Downsampler 2 Bypass 0 : VIP Chroma Downsampler 1 selected 1 : VIP Chroma Downsampler 1 Bypassed Chroma Downsampler Bypassed means the output format from the VIP will be 422 data. Selected means the output format will be 420	RW	0x0
16	VIP2_CHR_DS_1_BYPASS	Video Input Port 2 Chroma Downsampler 1 Bypass 0 : VIP Chroma Downsampler 1 selected 1 : VIP Chroma Downsampler 1 Bypassed Chroma Downsampler Bypassed means the output format from the VIP will be 422 data. Selected means the output format will be 420	RW	0x0
15	VIP2_MULTI_CHANNEL_SELECT	Video Input Port 2 Multi Channel Select 0 : VIP_PARSER A and B channels operate in single channel mode 1 : VIP_PARSER A and B channels directly drive VPDMA (multi-channel case) Multi-Channel means that the A and B sources are from multiple channels and used in a multiplexed stream mode. The VIP Parser extracts the channel ID from each source and outputs this information to VPDMA, and thus to memory. When operating in a multiplexed stream mode, this bit must be set to 1 to enable the channel information to be passed to memory. If this is not set, the channel number (or source number) will be 0 for all streams. If vip2_rgb_out_select = 1, then VIP_PARSER A port is connected to VPDMA	RW	0x0

Bits	Field Name	Description	Type	Reset
14:12	VIP2_CHR_DS_2_SRC_SELECT	Video Input Port 2 Chroma Downsampler 2 Source Select 0 : Path Disabled (no input to CHR_DS) 1 : Source from Scaler (SC_M) 2 : Source from Color Space Converter (CSC) 3 : Source from VIP_PARSER A port 4 : Source from VIP_PARSER B port 5 : Source from Transcode (422) 6 : Reserved 7 : Reserved	RW	0x0
11:9	VIP2_CHR_DS_1_SRC_SELECT	Video Input Port 2 Chroma Downsampler 1 Source Select 000 : Path Disabled (no input to CHR_DS) 001 : Source from Scaler (SC_M) 010 : Source from Color Space Converter (CSC) 011 : Source from VIP_PARSER A port 100 : Source from VIP_PARSER B port 101 : Source from Transcode (422) 110 : Reserved 111 : Reserved	RW	0x0
8	VIP2_RGB_OUT_HI_SELECT	Video Input Port 2 HI RGB Output Select 0 : Output Type is 420/422 1 : Output Type is RGB	RW	0x0
7	VIP2_RGB_OUT_LO_SELECT	Video Input Port 2 LO RGB Output Select 0 : Output Type is 420/422 1 : Output Type is RGB	RW	0x0
6	VIP2_RGB_SRC_SELECT	Video Input Port 2 RGB Output Path Select 0 : Source from Compositor RGB input 1 : Source from CSC	RW	0x0
5:3	VIP2_SC_SRC_SELECT	Video Input Port 2 SC_M Source Select 000 : Path Disabled 001 : Source from Color Space Converter (CSC) 010 : Source from VIP_PARSER A port 011 : Source from VIP_PARSER B port 100 : Source from Transcode (422) 101 : Reserved 110 : Reserved 111 : Reserved	RW	0x0
2:0	VIP2_CSC_SRC_SELECT	Video Input Port 2 CSC Source Select 000 : Path Disabled 001 : Source from VIP_PARSER A (422) port 010 : Source from VIP_PARSER B port 011 : Source from Transcode (422) 100 : Source from VIP_PARSER A (RGB) port 101 : Source from Compositor (RGB) 110 : Reserved 111 : Reserved	RW	0x0

Table 7-116. Register Call Summary for Register VIP_CLKC_VIP1DPS

VIP Functional Description
- VIP Slice Processing Path Overview: [0][1][2][3][4][5][6][7][8][9][10]
- VIP Slice Processing Path Multiplexers: [11][12]

VIP Register Manual
- VIP Top Level Register Summary: [15]

7.5.3 VIP Parser Registers

7.5.3.1 VIP Parser Register Summary

Table 7-117. VIP Parser Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	VIP_Slice0_parser Base Address	VIP_Slice1_parser Base Address
VIP_MAIN	RW	32	0x0000 0000	0x4897 5500	0x4897 5A00
VIP_PORT_A	RW	32	0x0000 0004	0x4897 5504	0x4897 5A04
VIP_XTRA_PORT_A	RW	32	0x0000 0008	0x4897 5508	0x4897 5A08
VIP_PORT_B	RW	32	0x0000 000C	0x4897 550C	0x4897 5A0C
VIP_XTRA_PORT_B	RW	32	0x0000 0010	0x4897 5510	0x4897 5A10
VIP_FIQ_MASK	RW	32	0x0000 0014	0x4897 5514	0x4897 5A14
VIP_FIQ_CLEAR	RW	32	0x0000 0018	0x4897 5518	0x4897 5A18
VIP_FIQ_STATUS	R	32	0x0000 001C	0x4897 551C	0x4897 5A1C
VIP_OUTPUT_PORT_A_SRC_FID	R	32	0x0000 0020	0x4897 5520	0x4897 5A20

Table 7-117. VIP Parser Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	VIP_Slice0_parser Base Address	VIP_Slice1_parser Base Address
VIP_OUTPUT_PORT_A_ENC_FID	R	32	0x0000 0024	0x4897 5524	0x4897 5A24
VIP_OUTPUT_PORT_B_SRC_FID	R	32	0x0000 0028	0x4897 5528	0x4897 5A28
VIP_OUTPUT_PORT_B_ENC_FID	R	32	0x0000 002C	0x4897 552C	0x4897 5A2C
VIP_OUTPUT_PORT_A_SRC0_SIZE	R	32	0x0000 0030	0x4897 5530	0x4897 5A30
VIP_OUTPUT_PORT_A_SRC1_SIZE	R	32	0x0000 0034	0x4897 5534	0x4897 5A34
VIP_OUTPUT_PORT_A_SRC2_SIZE	R	32	0x0000 0038	0x4897 5538	0x4897 5A38
VIP_OUTPUT_PORT_A_SRC3_SIZE	R	32	0x0000 003C	0x4897 553C	0x4897 5A3C
VIP_OUTPUT_PORT_A_SRC4_SIZE	R	32	0x0000 0040	0x4897 5540	0x4897 5A40
VIP_OUTPUT_PORT_A_SRC5_SIZE	R	32	0x0000 0044	0x4897 5544	0x4897 5A44
VIP_OUTPUT_PORT_A_SRC6_SIZE	R	32	0x0000 0048	0x4897 5548	0x4897 5A48
VIP_OUTPUT_PORT_A_SRC7_SIZE	R	32	0x0000 004C	0x4897 554C	0x4897 5A4C
VIP_OUTPUT_PORT_A_SRC8_SIZE	R	32	0x0000 0050	0x4897 5550	0x4897 5A50
VIP_OUTPUT_PORT_A_SRC9_SIZE	R	32	0x0000 0054	0x4897 5554	0x4897 5A54
VIP_OUTPUT_PORT_A_SRC10_SIZE	R	32	0x0000 0058	0x4897 5558	0x4897 5A58
VIP_OUTPUT_PORT_A_SRC11_SIZE	R	32	0x0000 005C	0x4897 555C	0x4897 5A5C
VIP_OUTPUT_PORT_A_SRC12_SIZE	R	32	0x0000 0060	0x4897 5560	0x4897 5A60
VIP_OUTPUT_PORT_A_SRC13_SIZE	R	32	0x0000 0064	0x4897 5564	0x4897 5A64
VIP_OUTPUT_PORT_A_SRC14_SIZE	R	32	0x0000 0068	0x4897 5568	0x4897 5A68
VIP_OUTPUT_PORT_A_SRC15_SIZE	R	32	0x0000 006C	0x4897 556C	0x4897 5A6C
VIP_OUTPUT_PORT_B_SRC0_SIZE	R	32	0x0000 0070	0x4897 5570	0x4897 5A70
VIP_OUTPUT_PORT_B_SRC1_SIZE	R	32	0x0000 0074	0x4897 5574	0x4897 5A74
VIP_OUTPUT_PORT_B_SRC2_SIZE	R	32	0x0000 0078	0x4897 5578	0x4897 5A78
VIP_OUTPUT_PORT_B_SRC3_SIZE	R	32	0x0000 007C	0x4897 557C	0x4897 5A7C
VIP_OUTPUT_PORT_B_SRC4_SIZE	R	32	0x0000 0080	0x4897 5580	0x4897 5A80
VIP_OUTPUT_PORT_B_SRC5_SIZE	R	32	0x0000 0084	0x4897 5584	0x4897 5A84
VIP_OUTPUT_PORT_B_SRC6_SIZE	R	32	0x0000 0088	0x4897 5588	0x4897 5A88
VIP_OUTPUT_PORT_B_SRC7_SIZE	R	32	0x0000 008C	0x4897 558C	0x4897 5A8C

Table 7-117. VIP Parser Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	VIP_Slice0_parser Base Address	VIP_Slice1_parser Base Address
VIP_OUTPUT_PORT_B_SRC8_SIZE	R	32	0x0000 0090	0x4897 5590	0x4897 5A90
VIP_OUTPUT_PORT_B_SRC9_SIZE	R	32	0x0000 0094	0x4897 5594	0x4897 5A94
VIP_OUTPUT_PORT_B_SRC10_SIZE	R	32	0x0000 0098	0x4897 5598	0x4897 5A98
VIP_OUTPUT_PORT_B_SRC11_SIZE	R	32	0x0000 009C	0x4897 559C	0x4897 5A9C
VIP_OUTPUT_PORT_B_SRC12_SIZE	R	32	0x0000 00A0	0x4897 55A0	0x4897 5AA0
VIP_OUTPUT_PORT_B_SRC13_SIZE	R	32	0x0000 00A4	0x4897 55A4	0x4897 5AA4
VIP_OUTPUT_PORT_B_SRC14_SIZE	R	32	0x0000 00A8	0x4897 55A8	0x4897 5AA8
VIP_OUTPUT_PORT_B_SRC15_SIZE	R	32	0x0000 00AC	0x4897 55AC	0x4897 5AAC
VIP_PORT_A_VDET_VEC	R	32	0x0000 00B0	0x4897 55B0	0x4897 5AB0
VIP_PORT_B_VDET_VEC	R	32	0x0000 00B4	0x4897 55B4	0x4897 5AB4
VIP_ANC_CROP_HORZ_PORT_A	RW	32	0x0000 00B8	0x4897 55B8	0x4897 5AB8
VIP_ANC_CROP_VERT_PORT_A	RW	32	0x0000 00BC	0x4897 55BC	0x4897 5ABC
VIP_CROP_HORZ_PORT_A	RW	32	0x0000 00C0	0x4897 55C0	0x4897 5AC0
VIP_CROP_VERT_PORT_A	RW	32	0x0000 00C4	0x4897 55C4	0x4897 5AC4
VIP_ANC_VIP_CROP_HORZ_PORT_B	RW	32	0x0000 00C8	0x4897 55C8	0x4897 5AC8
VIP_ANC_VIP_CROP_VERT_PORT_B	RW	32	0x0000 00CC	0x4897 55CC	0x4897 5ACC
VIP_CROP_HORZ_PORT_B	RW	32	0x0000 00D0	0x4897 55D0	0x4897 5AD0
VIP_CROP_VERT_PORT_B	RW	32	0x0000 00D4	0x4897 55D4	0x4897 5AD4
VIP_XTRA6_PORT_A	RW	32	0x0000 00D8	0x4897 55D8	0x4897 5AD8
VIP_XTRA7_PORT_B	RW	32	0x0000 00DC	0x4897 55DC	0x4897 5ADC
VIP_XTRA8_PORT_A	RW	32	0x0000 00E0	0x4897 55E0	0x4897 5AE0
VIP_XTRA9_PORT_B	RW	32	0x0000 00E4	0x4897 55E4	0x4897 5AE4

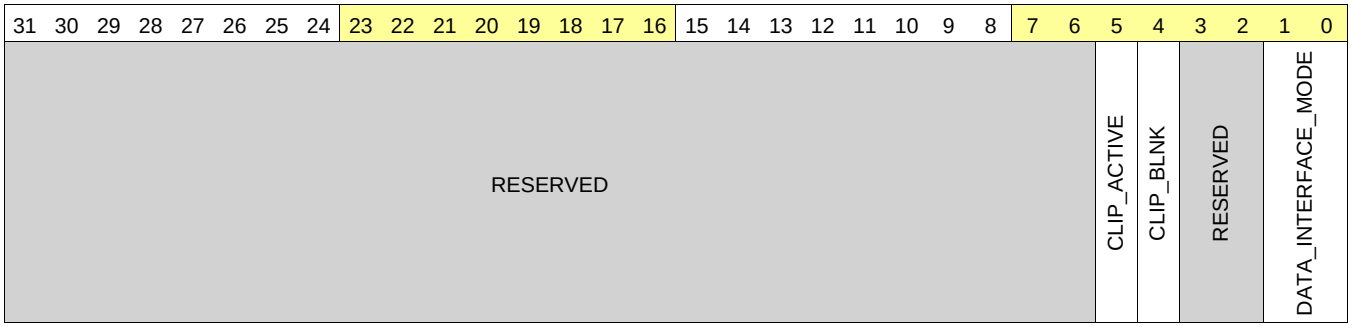
7.5.3.2 VIP Parser Register Description

Table 7-118. VIP_MAIN

Address Offset	0x0000 0000		
Physical Address	0x4897 5500 0x4897 5A00	**Instance**	VIP_Slice0_parser VIP_Slice1_parser
Description	Main Configuration for VIP Parser		

Table 7-118. VIP_MAIN (continued)

Type	RW



Bits	Field Name	Description	Type	Reset
31:6	RESERVED		R	0x0
5	CLIP_ACTIVE	Discrete Sync Only; 0 = Do not clip active pixels; 1 = Clip Active Pixels as follows: 0xFF -> 0xFE, 0x00 -> 0x01	RW	0x0
4	CLIP_BLNK	Discrete Sync Only; 0 = Do not clip Blanking Data; 1 = Clip Blanking Data as follows: 0xFF -> 0xFE, 0x00 -> 0x01	RW	0x0
3:2	RESERVED		R	0x0
1:0	DATA_INTERFACE_MODE	00 = 24b Port A data interface. 01 = 16b Port A data interface. 10 = 8b Port A data interfaces. 11 = Undefined. Port B is always an 8b data interface.	RW	0x0

Table 7-119. Register Call Summary for Register VIP_MAIN

VIP Environment • VIP Environment: [4][5]
VIP Functional Description • VIP Slice Processing Path Overview: [6] • Input Data Interface: [7][8] • Clipping: [9]
VIP Register Manual • VIP Parser Register Summary: [12]

Table 7-120. VIP_PORT_A

Address Offset	0x0000 0004		
Physical Address	0x4897 5504 0x4897 5A04	**Instance**	VIP_Slice0_parser VIP_Slice1_parser
Description	Configuration for Input Port A		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ANALYZER_FVH_ERR_CORRECTION_ENABLE	ANALYZER_2X4X_SRCNUM_POS	FID_SKEW_POSTCOUNT						SW_RESET	DISCRETE_BASIC_MODE	FID_SKEW_PRECOUNT						USE_ACTVID_HSYNC_N	FID_DETECT_MODE	ACTVID_POLARITY	VSYNC_POLARITY	HSYNC_POLARITY	PIXCLK_EDGE_POLARITY	FID_POLARITY	ENABLE	CLR_ASYNC_FIFO_RD	CLR_ASYNC_FIFO_WR	CTRL_CHAN_SEL	SYNC_TYPE				

Bits	Field Name	Description	Type	Reset
31	ANALYZER_FVH_ERR_CORRECTION_ENABLE	Embedded Sync Only 0 = Ignore the protection bits in the XV (fvh) codeword header. This setting is typically desired. 1 = Use the protection bits in an attempt to do error correction for the fvh control bits.	RW	0x0
30	ANALYZER_2X4X_SRCNUM_POS	Embedded Sync Only 0 = For 2x/4x mux mode, srcnum is in the least significant nibble of the XV/fvh codeword (srcnum replaces the protection bits) 1 = For 2x/4x mux mode, srcnum is in the least significant nibble of a horizontal blanking pixel value	RW	0x0
29:24	FID_SKEW_POSTCOUNT	Discrete Sync Only post count value when using vsync skew in FID determination	RW	0x0
23	SW_RESET	0 = Normal 1 = Reset Port A logic. Must be set to ?0? again by the software for the module to function.	RW	0x0
22	DISCRETE_BASIC_MODE	This register is valid for Discrete Sync mode only. 0 = Normal Discrete Mode. Hsync Style Capture operates as follows: - Captures line starting from HSYNC inactive to active condition. - VSYNC determined by sync window. - FID can be determined by VSYNC skew or captured from pin at first pixel in first line. ACTVID style capture works as follows: - Captures line during contiguous ACTVID envelope. - VSYNC is captured at the first pixel in each line. - FID is captured on first pixel of ACTVID window. 1 = Basic Discrete Mode. When using hsync with Hsync Style Capture operates as follows: - The last line of active video ends on the pixel clock cycle where VSYNC transitions from inactive to active. - FID pin value is captured on this cycle and is used for the next field. - FID detection by VSYNC skew is not allowed. ACTVID style capture works as follows: - VSYNC is expected to transition from inactive to active between ACTVID window. - This VSYNC transition allows the next line in an ACTVID envelope to be sent to a new VPDMA buffer. - FID value is determined by the FID pin value on the cycle where VSYNC transitions from inactive to active. In basic discrete mode, there is no Vertical Ancillary Data. Therefore, VPDMA descriptors should not use Ancillary Data channels.	RW	0x0
21:16	FID_SKEW_PRECOUNT	Discrete Sync Only pre count value when using vsync skew in FID determination	RW	0x0
15	USE_ACTVID_HSYNC_N	Discrete Sync Only 0 = Use HSYNC style line capture 1 = Use ACTVID style line capture	RW	0x0
14	FID_DETECT_MODE	Discrete Sync Only 0 = Take FID from pin 1 = FID is determined by VSYNC skew	RW	0x0

Bits	Field Name	Description	Type	Reset
13	ACTVID_POLARITY	Discrete Sync Only 0 = ACTVID is active low 1 = ACTVID is active high	RW	0x0
12	VSYNC_POLARITY	Discrete Sync Only 0 = VSYNC is active low 1 = VSYNC is active high	RW	0x0
11	HSYNC_POLARITY	Discrete Sync Only 0 = HSYNC is active low 1 = HSYNC is active high	RW	0x0
10	PIXCLK_EDGE_POLARITY	0 = Rising Edge is active PIXCLK edge 1 = Falling Edge is active PIXCLK edge	RW	0x0
9	FID_POLARITY	0 = Keep FID as found 1 = Invert Determined Value of FID	RW	0x0
8	ENABLE	0 = Disable Port 1 = Enable Port	RW	0x0
7	CLR_ASYNC_FIFO_RD	0 = Normal 1 = Clear Async FIFO Read Logic	RW	0x0
6	CLR_ASYNC_FIFO_WR	0 = Normal 1 = Clear Async FIFO Write Logic	RW	0x0
5:4	CTRL_CHAN_SEL	Embedded Sync Only In 8b mode.. there is only one channel on data[7:0]. In 16b mode.. there are two channels. The Luma Channel is on data[15:8]. The Chroma Channel is on data[7:0]. In 24b mode.. there are three channels. The R channel is on data[23:16].. the G channel is on [15:8]. and the B channel is on data[7:0]. 00 = Use data[7:0] to extract control codes. 01 = Use data[15:8] to extract control codes. 10 = Use data[23:16] to extract control codes. 11 = Undefined In 16b and 24b modes.. this register is also used to select the channel from which Ancillary Data is extracted. The Ancillary Data channel must be the same as the control code channel. For 8b mode.. the anc_chan_sel_8b register is used to select the Luma or Chroma channel from which Ancillary Data is taken.	RW	0x0
3:0	SYNC_TYPE	0000 = embedded sync single 4:2:2 YUV stream 0001 = embedded sync 2x multiplexed 4:2:2 YUV stream 0010 = embedded sync 4x multiplexed 4:2:2 YUV stream 0011 = embedded sync line multiplexed 4:2:2 YUV stream 0100 = discrete sync single 4:2:2 YUV stream 0101 = embedded sync single RGB stream or single 444 YUV stream 0110 = reserved 0111 = reserved 1000 = reserved 1001 = reserved 1010 = discrete sync single 24b RGB stream	RW	0x0

Table 7-121. Register Call Summary for Register VIP_PORT_A

VIP Functional Description

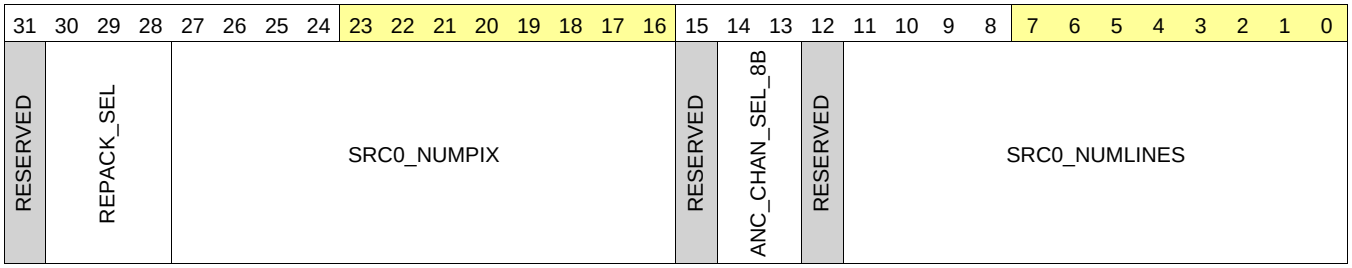
- Input Data Interface: [1][2][3][4]
- Interrupts: [5]
- Disable Handling: [6]
- Discrete Sync Signals: [7][8]
- VIP Overflow Detection and Recovery: [9][10][11][12][13][14][15][16]

VIP Register Manual

- VIP Parser Register Summary: [19]

Table 7-122. VIP_XTRA_PORT_A

Address Offset	0x0000 0008		
Physical Address	0x4897 5508 0x4897 5A08	**Instance**	VIP_Slice0_parser VIP_Slice1_parser
Description	ore Configuration for Input Port A		
Type	RW		



Bits	Field Name	Description	Type	Reset
31	RESERVED		R	0x0
30:28	REPACK_SEL	000 = Straight Through 001 = Cross Swap 010 = Left Center Swap 011 = Center Right Swap 100 = Right Rotate 101 = Left Rotate 110 = RAW16 to RGB565 Mapping 111 = RAW12 Swap	RW	0x0
27:16	SRC0_NUMPIX	Number of expected pixels on Source Number 0. The Port_a_src0_size interrupt will trigger if a line is encountered that differs from this pixelcount.	RW	0x0
15	RESERVED		R	0x0
14:13	ANC_CHAN_SEL_8B	In 8b mode, Vertically Ancillary Data typically resides in the Luma sites. This bit allows vertical ancillary data to be extracted from the chroma sites instead . 00 = Extract 8b Mode Vertical Ancillary Data from Luma Sites 01 = Extract 8b Mode Vertical Ancillary Data from Chroma Sites 10, 11 = Extract every single sample of vertical ancillary data. The output line is twice as wide as the other modes. For 16b and 24b inputs, ctrl_chan_sel is used to select which channel is used as a source for vertical ancillary data.	RW	0x0
12	RESERVED		R	0x0
11:0	SRC0_NUMLINES	Number of expected lines on Source Number 0. The Port_a_src0_size interrupt will trigger if a field/frame is encountered that differs from this linecount.	RW	0x0

Table 7-123. Register Call Summary for Register VIP_XTRA_PORT_A

VIP Functional Description
- Repacker: [0][1]
- Ancillary and Active Video Cropping: [2]
- Picture Size Interrupt: [3][4]

VIP Register Manual
- VIP Parser Register Summary: [7]

Table 7-124. VIP_PORT_B

Address Offset	0x0000 000C		
Physical Address	0x4897 550C 0x4897 5A0C	**Instance**	VIP_Slice0_parser VIP_Slice1_parser
Description	Configuration for Input Port B		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ANALYZER_FVH_ERR_CORRECTION_ENABLE	ANALYZER_2X4X_SRCNUM_POS	FID_SKEW_POSTCOUNT						SW_RESET	DISCRETE_BASIC_MODE	FID_SKEW_PRECOUNT						USE_ACTVID_HSYNC_N	FID_DETECT_MODE	ACTVID_POLARITY	VSYNC_POLARITY	HSYNC_POLARITY	PIXCLK_EDGE_POLARITY	FID_POLARITY	ENABLE	CLR_ASYNC_FIFO_RD	CLR_ASYNC_FIFO_WR	CTRL_CHAN_SEL	SYNC_TYPE				

Bits	Field Name	Description	Type	Reset
31	ANALYZER_FVH_ERR_CORRECTION_ENABLE	Embedded Sync Only 0 = Ignore the protection bits in the XV (fvh) codeword header. This setting is typically desired. 1 = Use the protection bits in an attempt to do error correction for the fvh control bits.	RW	0x0
30	ANALYZER_2X4X_SRCNUM_POS	Embedded Sync Only 0 = For 2x/4x mux mode, srcnum is in the least significant nibble of the XV/fvh codeword (srcnum replaces the protection bits) 1 = For 2x/4x mux mode, srcnum is in the least significant nibble of a horizontal blanking pixel value	RW	0x0
29:24	FID_SKEW_POSTCOUNT	Discrete Sync Only post count value when using vsync skew in FID determination	RW	0x0
23	SW_RESET	0 = Normal 1 = Reset Port B logic. Must be set to ?0? again by the software for the module to function.	RW	0x0
22	DISCRETE_BASIC_MODE	This register is valid for Discrete Sync mode only. 0 = Normal Discrete Mode. Hsync Style Capture operates as follows: - Captures line starting from HSYNC inactive to active condition. - VSYNC determined by sync window. - FID can be determined by VSYNC skew or captured from pin at first pixel in first line. ACTVID style capture works as follows: - Captures line during contiguous ACTVID envelope. - VSYNC is captured at the first pixel in each line. - FID is captured on first pixel of ACTVID window. 1 = Basic Discrete Mode. When using hsync with Hsync Style Capture operates as follows: - The last line of active video ends on the pixel clock cycle where VSYNC transitions from inactive to active. - FID pin value is captured on this cycle and is used for the next field. - FID detection by VSYNC skew is not allowed. ACTVID style capture works as follows: - VSYNC is expected to transition from inactive to active between ACTVID window. - This VSYNC transition allows the next line in an ACTVID envelope to be sent to a new VPDMA buffer. - FID value is determined by the FID pin value on the cycle where VSYNC transitions from inactive to active. In basic discrete mode, there is no Vertical Ancillary Data. Therefore, VPDMA descriptors should not use Ancillary Data channels.	RW	0x0
21:16	FID_SKEW_PRECOUNT	Discrete Sync Only pre count value when using vsync skew in FID determination	RW	0x0
15	USE_ACTVID_HSYNC_N	Discrete Sync Only 0 = Use HSYNC style line capture 1 = Use ACTVID style line capture	RW	0x0
14	FID_DETECT_MODE	Discrete Sync Only 0 = Take FID from pin 1 = FID is determined by VSYNC skew	RW	0x0

Bits	Field Name	Description	Type	Reset
13	ACTVID_POLARITY	Discrete Sync Only 0 = ACTVID is active low 1 = ACTVID is active high	RW	0x0
12	VSYNC_POLARITY	Discrete Sync Only 0 = VSYNC is active low 1 = VSYNC is active high	RW	0x0
11	HSYNC_POLARITY	Discrete Sync Only 0 = HSYNC is active low 1 = HSYNC is active high	RW	0x0
10	PIXCLK_EDGE_POLARITY	0 = Rising Edge is active PIXCLK edge 1 = Falling Edge is active PIXCLK edge	RW	0x0
9	FID_POLARITY	0 = Keep FID as found 1 = Invert Determined Value of FID	RW	0x0
8	ENABLE	0 = Disable 1 = Enable	RW	0x0
7	CLR_ASYNC_FIFO_RD	0 = Normal 1 = Clear Async FIFO Read Logic	RW	0x0
6	CLR_ASYNC_FIFO_WR	0 = Normal 1 = Clear Async FIFO Write Logic	RW	0x0
5:4	CTRL_CHAN_SEL	PORT B supports on 8b mode. Always write 0 to this field. The anc_chan_sel_8b register is used to select the Luma or Chroma channel from which Ancillary Data is taken.	RW	0x0
3:0	SYNC_TYPE	0000 = embedded sync single YUV stream 0001 = embedded sync 2x multiplexed YUV stream 0010 = embedded sync 4x multiplexed YUV stream 0011 = embedded sync line multiplexed YUV stream 0100 = discrete sync single YUV stream 0101 = embedded sync single RGB stream 0110 = reserved 0111 = reserved 1000 = reserved 1001 = reserved 1010 = discrete sync single 24b RGB stream	RW	0x0

Table 7-125. Register Call Summary for Register VIP_PORT_B

VIP Functional Description

- Input Data Interface: [1][2][3][4]
- Interrupts: [5]
- Disable Handling: [6]
- Discrete Sync Signals: [7][8]

VIP Register Manual

- VIP Parser Register Summary: [11]

Table 7-126. VIP_XTRA_PORT_B

Address Offset	0x0000 0010		
Physical Address	0x4897 5510 0x4897 5A10	**Instance**	VIP_Slice0_parser VIP_Slice1_parser
Description	ore Configuration for Input Port B		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				SRC0_NUMPIX												RESERVED	ANC_CHAN_SEL_8B		RESERVED	SRC0_NUMLINES												

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:16	SRC0_NUMPIX	Number of expected pixels on Source Number 0. The Port_b_src0_size interrupt will trigger if a line is encountered that differs from this pixelcount.	RW	0x0
15	RESERVED		R	0x0
14:13	ANC_CHAN_SEL_8B	In 8b mode, Vertically Ancillary Data typically resides in the Luma sites. This bit allows vertical ancillary data to be extracted from the chroma sites instead . 00 = Extract 8b Mode Vertical Ancillary Data from Luma Sites 01 = Extract 8b Mode Vertical Ancillary Data from Chroma Sites 10, 11 = Extract every single sample of vertical ancillary data. The output line is twice as wide as the other modes. For 16b and 24b inputs, ctrl_chan_sel is used to select which channel is used as a source for vertical ancillary data.	RW	0x0
12	RESERVED		R	0x0
11:0	SRC0_NUMLINES	Number of expected lines on Source Number 0. The Port_b_src0_size interrupt will trigger if a field/frame is encountered that differs from this linecount.	RW	0x0

Table 7-127. Register Call Summary for Register VIP_XTRA_PORT_B

VIP Functional Description
- Picture Size Interrupt: [0][1]

VIP Register Manual
- VIP Parser Register Summary: [4]

Table 7-128. VIP_FIQ_MASK

Address Offset	0x0000 0014		
Physical Address	0x4897 5514 0x4897 5A14	**Instance**	VIP_Slice0_parser VIP_Slice1_parser
Description	ask Bits for ARM FIQs		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED										PORT_B_CFG_DISABLE_COMPLETE_MASK	PORT_A_CFG_DISABLE_COMPLETE_MASK	PORT_B_ANC_PROTOCOL_VIOLATION_MASK	PORT_B_YUV_PROTOCOL_VIOLATION_MASK	PORT_A_ANC_PROTOCOL_VIOLATION_MASK	PORT_A_YUV_PROTOCOL_VIOLATION_MASK	PORT_B_SRC0_SIZE	PORT_A_SRC0_SIZE	PORT_B_DISCONN	PORT_B_CONN	PORT_A_DISCONN	PORT_A_CONN	OUTPUT_FIFO_PRTB_ANC_OF	RESERVED	OUTPUT_FIFO_PRTB_YUV_OF	OUTPUT_FIFO_PRTA_ANC_OF	RESERVED	OUTPUT_FIFO_PRTA_YUV_OF	ASYNC_FIFO_PRTB_OF	ASYNC_FIFO_PRTA_OF	PRTB_VDET_MASK	PRTA_VDET_MASK

Bits	Field Name	Description	Type	Reset
31:22	RESERVED		R	0x0
21	PORT_B_CFG_DISABLE_COMPLETE_MASK	Port B Cfg Disable Complete Mask	RW	0x0
20	PORT_A_CFG_DISABLE_COMPLETE_MASK	Port A Cfg Disable Complete Mask	RW	0x0
19	PORT_B_ANC_PROTOCOL_VIOLATION_MASK	Port B ANC VPI Protocol Violation Mask	RW	0x0
18	PORT_B_YUV_PROTOCOL_VIOLATION_MASK	Port B YUV VPI Protocol Violation Mask	RW	0x0
17	PORT_A_ANC_PROTOCOL_VIOLATION_MASK	Port A ANC VPI Protocol Violation Mask	RW	0x0
16	PORT_A_YUV_PROTOCOL_VIOLATION_MASK	Port A YUV VPI Protocol Violation Mask	RW	0x0
15	PORT_B_SRC0_SIZE	Video size detected on Port B does not match size programmed in xtra_port_b register	RW	0x0
14	PORT_A_SRC0_SIZE	Video size detected on Port A does not match size programmed in xtra_port_a register	RW	0x0
13	PORT_B_DISCONN	Port B Link Disconnect Srcnum 0 Mask	RW	0x0
12	PORT_B_CONN	Port B Link Connect Srcnum 0 Mask	RW	0x0
11	PORT_A_DISCONN	Port A Link Disconnect Scrnum 0 Mask	RW	0x0
10	PORT_A_CONN	Port A Link Connect Srcnum 0 Mask	RW	0x0
9	OUTPUT_FIFO_PRTB_ANC_OF	Output FIFO Port B Ancillary Overflow Mask	RW	0x0
8	RESERVED		R	0x0
7	OUTPUT_FIFO_PRTB_YUV_OF	Output FIFO Port B Luma Overflow Mask	RW	0x0
6	OUTPUT_FIFO_PRTA_ANC_OF	Output FIFO Port A Ancillary Overflow Mask	RW	0x0
5	RESERVED		R	0x0
4	OUTPUT_FIFO_PRTA_YUV_OF	Output FIFO Port A Luma Overflow Mask	RW	0x0
3	ASYNC_FIFO_PRTB_OF	Port B Async FIFO Overflow FIQ Mask	RW	0x0
2	ASYNC_FIFO_PRTA_OF	Port A Async FIFO Overflow FIQ Mask	RW	0x0
1	PRTB_VDET_MASK	Port B Video Detect FIQ Mask	RW	0x0
0	PRTA_VDET_MASK	Port A Video Detect FIQ Mask	RW	0x0

Table 7-129. Register Call Summary for Register VIP_FIQ_MASK

VIP Functional Description
- Interrupts: [0][1][2][3][4][5][6][7][8][9][10][11][12][13][15][16][18][19][20][21][22]

VIP Register Manual
- VIP Parser Register Summary: [25]

Table 7-130. VIP_FIQ_CLEAR

Address Offset	0x0000 0018		
Physical Address	0x4897 5518 0x4897 5A18	**Instance**	VIP_Slice0_parser VIP_Slice1_parser
Description	Clears bits in the FIQ Status		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED										PORT_A_YUV_PROTOCOL_VIOLATION_CLR	PORT_A_ANC_PROTOCOL_VIOLATION_CLR	PORT_B_YUV_PROTOCOL_VIOLATION_CLR	PORT_B_ANC_PROTOCOL_VIOLATION_CLR	PORT_A_CFG_DISABLE_COMPLETE_CLR	PORT_B_CFG_DISABLE_COMPLETE_CLR	PORT_B_SRC0_SIZE_CLR	PORT_A_SRC0_SIZE_CLR	PORT_B_DISCONN_CLR	PORT_B_CONN_CLR	PORT_A_DISCONN_CLR	PORT_A_CONN_CLR	OUTPUT_FIFO_PRTB_ANC_CLR	RESERVED	OUTPUT_FIFO_PRTB_YUV_CLR	OUTPUT_FIFO_PRTA_ANC_CLR	RESERVED	OUTPUT_FIFO_PRTA_YUV_CLR	ASYNC_FIFO_PRTB_CLR	ASYNC_FIFO_PRTA_CLR	PRTB_VDET_CLR	PRTA_VDET_CLR

Bits	Field Name	Description	Type	Reset
31:22	RESERVED		R	0x0
21	PORT_A_YUV_PROTOCOL_VIOLATION_CLR	Write 1 followed by 0 to Clear Port B Cfg Disable Complete FIQ	RW	0x0
20	PORT_A_ANC_PROTOCOL_VIOLATION_CLR	Write 1 followed by 0 to Clear Port A Cfg Disable Complete FIQ	RW	0x0
19	PORT_B_YUV_PROTOCOL_VIOLATION_CLR	Write 1 followed by 0 to Clear Port B ANC VPI Protocol Violation FIQ	RW	0x0
18	PORT_B_ANC_PROTOCOL_VIOLATION_CLR	Write 1 followed by 0 to Clear Port B YUV VPI Protocol Violation FIQ	RW	0x0
17	PORT_A_CFG_DISABLE_COMPLETE_CLR	Write 1 followed by 0 to Clear Port A ANC VPI Protocol Violation FIQ	RW	0x0
16	PORT_B_CFG_DISABLE_COMPLETE_CLR	Write 1 followed by 0 to Clear Port A YUV VPI Protocol Violation FIQ	RW	0x0
15	PORT_B_SRC0_SIZE_CLR	Write 1 followed by 0 to Clear Port B Src0 Size FIQ	RW	0x0
14	PORT_A_SRC0_SIZE_CLR	Write 1 followed by 0 to Clear Port A Src0 Size FIQ	RW	0x0
13	PORT_B_DISCONN_CLR	Write 1 followed by 0 to Clear Port B Link Disconnect FIQ	RW	0x0
12	PORT_B_CONN_CLR	Write 1 followed by 0 to Clear Port B Link Connect FIQ	RW	0x0
11	PORT_A_DISCONN_CLR	Write 1 followed by 0 to Clear Port A Link Disconnect FIQ	RW	0x0
10	PORT_A_CONN_CLR	Write 1 followed by 0 to Clear Port A Link Connect FIQ	RW	0x0
9	OUTPUT_FIFO_PRTB_ANC_CLR	Write 1 followed by 0 to Clear Output FIFO Port B Ancillary Overflow FIQ	RW	0x0
8	RESERVED		R	0x0
7	OUTPUT_FIFO_PRTB_YUV_CLR	Write 1 followed by 0 to Clear Output FIFO Port B Luma Overflow FIQ	RW	0x0
6	OUTPUT_FIFO_PRTA_ANC_CLR	Write 1 followed by 0 to Clear Output FIFO Port A Ancillary Overflow FIQ	RW	0x0
5	RESERVED		R	0x0
4	OUTPUT_FIFO_PRTA_YUV_CLR	Write 1 followed by 0 to Clear Output FIFO Port A Luma Overflow FIQ	RW	0x0
3	ASYNC_FIFO_PRTB_CLR	Write 1 followed by 0 to Clear Async FIFO Port B Overflow FIQ	RW	0x0
2	ASYNC_FIFO_PRTA_CLR	Write 1 followed by 0 to Clear Async FIFO Port A Overflow FIQ	RW	0x0

Bits	Field Name	Description	Type	Reset
1	PRTB_VDET_CLR	Write 1 followed by 0 to Clear Video Detect FIQ for Port B	RW	0x0
0	PRTA_VDET_CLR	Write 1 followed by 0 to Clear Video Detect FIQ for Port A	RW	0x0

Table 7-131. Register Call Summary for Register VIP_FIQ_CLEAR

VIP Functional Description
- Interrupts: [0]

VIP Register Manual
- VIP Parser Register Summary: [3]

Table 7-132. VIP_FIQ_STATUS

Address Offset	0x0000 001C		
Physical Address	0x4897 551C 0x4897 5A1C	**Instance**	VIP_Slice0_parser VIP_Slice1_parser
Description	FIQ Status values		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED										PORT_B_CFG_DISABLE_COMPLETE_CLR	PORT_A_CFG_DISABLE_COMPLETE	PORT_B_ANC_PROTOCOL_VIOLATION	PORT_B_YUV_PROTOCOL_VIOLATION	PORT_A_ANC_PROTOCOL_VIOLATION	PORT_A_YUV_PROTOCOL_VIOLATION	PORT_B_SRC0_SIZE_STATUS	PORT_A_SRC0_SIZE_STATUS	PORT_B_DISCONN_STATUS	PORT_B_CONN_STATUS	PORT_A_DISCONN_STATUS	PORT_A_CONN_STATUS	OUTPUT_FIFO_PRTB_ANC_STATUS	OUTPUT_FIFO_PRTB_CHROMA_STATUS	OUTPUT_FIFO_PRTB_LUMA_STATUS	OUTPUT_FIFO_PRTA_ANC_STATUS	OUTPUT_FIFO_PRTA_CHROMA_STATUS	OUTPUT_FIFO_PRTA_LUMA_STATUS	ASYNC_FIFO_PRTB_STATUS	ASYNC_FIFO_PRTA_STATUS	PRTB_VDET_STATUS	PRTA_VDET_STATUS

Bits	Field Name	Description	Type	Reset
31:22	RESERVED		R	0x0
21	PORT_B_CFG_DISABLE_COMPLETE_CLR	Port B Cfg Disable Complete FIQ	R	0x0
20	PORT_A_CFG_DISABLE_COMPLETE	Port A Cfg Disable Complete FIQ	R	0x0
19	PORT_B_ANC_PROTOCOL_VIOLATION	Port B ANC VPI Protocol Violation FIQ	R	0x0
18	PORT_B_YUV_PROTOCOL_VIOLATION	Port B YUV VPI Protocol Violation FIQ	R	0x0
17	PORT_A_ANC_PROTOCOL_VIOLATION	Port A ANC VPI Protocol Violation FIQ	R	0x0
16	PORT_A_YUV_PROTOCOL_VIOLATION	Port A YUV VPI Protocol Violation FIQ	R	0x0
15	PORT_B_SRC0_SIZE_STATUS	Port B Source 0 Size FIQ	R	0x0
14	PORT_A_SRC0_SIZE_STATUS	Port A Source 0 Size FIQ	R	0x0
13	PORT_B_DISCONN_STATUS	Port B Disconnect FIQ	R	0x0
12	PORT_B_CONN_STATUS	Port B Connect FIQ	R	0x0
11	PORT_A_DISCONN_STATUS	Port A Disconnect FIQ	R	0x0
10	PORT_A_CONN_STATUS	Port A Connect FIQ	R	0x0
9	OUTPUT_FIFO_PRTB_ANC_STATUS	Output FIFO Port B Ancillary Overflow Status	R	0x0
8	OUTPUT_FIFO_PRTB_CHROMA_STATUS	Output FIFO Port B Chroma Overflow Status	R	0x0

Bits	Field Name	Description	Type	Reset
7	OUTPUT_FIFO_PRTB_LUMA_STATUS	Output FIFO Port B Luma Overflow Status	R	0x0
6	OUTPUT_FIFO_PRTA_ANC_STATUS	Output FIFO Port A Ancillary Overflow Status	R	0x0
5	OUTPUT_FIFO_PRTA_CHROMA_STATUS	Output FIFO Port A Chroma Overflow Status	R	0x0
4	OUTPUT_FIFO_PRTA_LUMA_STATUS	Output FIFO Port A Luma Overflow Status	R	0x0
3	ASYNC_FIFO_PRTB_STATUS	Async FIFO Port B Overflow Status	R	0x0
2	ASYNC_FIFO_PRTA_STATUS	Async FIFO Port A Overflow Status	R	0x0
1	PRTB_VDET_STATUS	VDET Status for Port B	R	0x0
0	PRTA_VDET_STATUS	VDET Status for Port A	R	0x0

Table 7-133. Register Call Summary for Register VIP_FIQ_STATUS

VIP Functional Description
- Interrupts: [0][1][2][3][4][5][6][7][8][9][10][11][12][13][15][16][18][19][20][21][22]
- VIP Overflow Detection and Recovery: [23]

VIP Register Manual
- VIP Parser Register Summary: [26]

Table 7-134. VIP_OUTPUT_PORT_A_SRC_FID

Address Offset	0x0000 0020		
Physical Address	0x4897 5520 0x4897 5A20	**Instance**	VIP_Slice0_parser VIP_Slice1_parser
Description	Current and Previous Output Port A Source FID values		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PRTA_SRC15_CURR_SOURCE_FID	PRTA_SRC15_PREV_SOURCE_FID	PRTA_SRC14_CURR_SOURCE_FID	PRTA_SRC14_PREV_SOURCE_FID	PRTA_SRC13_CURR_SOURCE_FID	PRTA_SRC13_PREV_SOURCE_FID	PRTA_SRC12_CURR_SOURCE_FID	PRTA_SRC12_PREV_SOURCE_FID	PRTA_SRC11_CURR_SOURCE_FID	PRTA_SRC11_PREV_SOURCE_FID	PRTA_SRC10_CURR_SOURCE_FID	PRTA_SRC10_PREV_SOURCE_FID	PRTA_SRC9_CURR_SOURCE_FID	PRTA_SRC9_PREV_SOURCE_FID	PRTA_SRC8_CURR_SOURCE_FID	PRTA_SRC8_PREV_SOURCE_FID	PRTA_SRC7_CURR_SOURCE_FID	PRTA_SRC7_PREV_SOURCE_FID	PRTA_SRC6_CURR_SOURCE_FID	PRTA_SRC6_PREV_SOURCE_FID	PRTA_SRC5_CURR_SOURCE_FID	PRTA_SRC5_PREV_SOURCE_FID	PRTA_SRC4_CURR_SOURCE_FID	PRTA_SRC4_PREV_SOURCE_FID	PRTA_SRC3_CURR_SOURCE_FID	PRTA_SRC3_PREV_SOURCE_FID	PRTA_SRC2_CURR_SOURCE_FID	PRTA_SRC2_PREV_SOURCE_FID	PRTA_SRC1_CURR_SOURCE_FID	PRTA_SRC1_PREV_SOURCE_FID	PRTA_SRC0_CURR_SOURCE_FID	PRTA_SRC0_PREV_SOURCE_FID

Bits	Field Name	Description	Type	Reset
31	PRTA_SRC15_CURR_SOURCE_FID	For Source ID 15 from Port A. Source Field ID for Current Field	R	0x0
30	PRTA_SRC15_PREV_SOURCE_FID	For Source ID 15 from Port A. Source Field ID for Previous Field	R	0x0
29	PRTA_SRC14_CURR_SOURCE_FID	For Source ID 14 from Port A. Source Field ID for Current Field	R	0x0
28	PRTA_SRC14_PREV_SOURCE_FID	For Source ID 14 from Port A. Source Field ID for Previous Field	R	0x0
27	PRTA_SRC13_CURR_SOURCE_FID	For Source ID 13 from Port A. Source Field ID for Current Field	R	0x0
26	PRTA_SRC13_PREV_SOURCE_FID	For Source ID 13 from Port A. Source Field ID for Previous Field	R	0x0
25	PRTA_SRC12_CURR_SOURCE_FID	For Source ID 12 from Port A. Source Field ID for Current Field	R	0x0

Bits	Field Name	Description	Type	Reset
24	PRTA_SRC12_PREV_SOURCE_FID	For Source ID 12 from Port A. Source Field ID for Previous Field	R	0x0
23	PRTA_SRC11_CURR_SOURCE_FID	For Source ID 11 from Port A. Source Field ID for Current Field	R	0x0
22	PRTA_SRC11_PREV_SOURCE_FID	For Source ID 11 from Port A. Source Field ID for Previous Field	R	0x0
21	PRTA_SRC10_CURR_SOURCE_FID	For Source ID 10 from Port A. Source Field ID for Current Field	R	0x0
20	PRTA_SRC10_PREV_SOURCE_FID	For Source ID 10 from Port A. Source Field ID for Previous Field	R	0x0
19	PRTA_SRC9_CURR_SOURCE_FID	For Source ID 9 from Port A. Source Field ID for Current Field	R	0x0
18	PRTA_SRC9_PREV_SOURCE_FID	For Source ID 9 from Port A. Source Field ID for Previous Field	R	0x0
17	PRTA_SRC8_CURR_SOURCE_FID	For Source ID 8 from Port A. Source Field ID for Current Field	R	0x0
16	PRTA_SRC8_PREV_SOURCE_FID	For Source ID 8 from Port A. Source Field ID for Previous Field	R	0x0
15	PRTA_SRC7_CURR_SOURCE_FID	For Source ID 7 from Port A. Source Field ID for Current Field	R	0x0
14	PRTA_SRC7_PREV_SOURCE_FID	For Source ID 7 from Port A. Source Field ID for Previous Field	R	0x0
13	PRTA_SRC6_CURR_SOURCE_FID	For Source ID 6 from Port A. Source Field ID for Current Field	R	0x0
12	PRTA_SRC6_PREV_SOURCE_FID	For Source ID 6 from Port A. Source Field ID for Previous Field	R	0x0
11	PRTA_SRC5_CURR_SOURCE_FID	For Source ID 5 from Port A. Source Field ID for Current Field	R	0x0
10	PRTA_SRC5_PREV_SOURCE_FID	For Source ID 5 from Port A. Source Field ID for Previous Field	R	0x0
9	PRTA_SRC4_CURR_SOURCE_FID	For Source ID 4 from Port A. Source Field ID for Current Field	R	0x0
8	PRTA_SRC4_PREV_SOURCE_FID	For Source ID 4 from Port A. Source Field ID for Previous Field	R	0x0
7	PRTA_SRC3_CURR_SOURCE_FID	For Source ID 3 from Port A. Source Field ID for Current Field	R	0x0
6	PRTA_SRC3_PREV_SOURCE_FID	For Source ID 3 from Port A. Source Field ID for Previous Field	R	0x0
5	PRTA_SRC2_CURR_SOURCE_FID	For Source ID 2 from Port A. Source Field ID for Current Field	R	0x0
4	PRTA_SRC2_PREV_SOURCE_FID	For Source ID 2 from Port A. Source Field ID for Previous Field	R	0x0
3	PRTA_SRC1_CURR_SOURCE_FID	For Source ID 1 from Port A. Source Field ID for Current Field	R	0x0
2	PRTA_SRC1_PREV_SOURCE_FID	For Source ID 1 from Port A. Source Field ID for Previous Field	R	0x0
1	PRTA_SRC0_CURR_SOURCE_FID	For Source ID 0 from Port A. Source Field ID for Current Field	R	0x0
0	PRTA_SRC0_PREV_SOURCE_FID	For Source ID 0 from Port A. Source Field ID for Previous Field	R	0x0

Table 7-135. Register Call Summary for Register VIP_OUTPUT_PORT_A_SRC_FID

VIP Functional Description • Source Multiplexing: [0]
VIP Register Manual • VIP Parser Register Summary: [3]

Table 7-136. VIP_OUTPUT_PORT_A_ENC_FID

Address Offset	0x0000 0024		
Physical Address	0x4897 5524 0x4897 5A24	**Instance**	VIP_Slice0_parser VIP_Slice1_parser
Description	Current and Previous Output Port A Encoder FID values		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PRTA_SRC15_CURR_ENC_FID	PRTA_SRC15_PREV_ENC_FID	PRTA_SRC14_CURR_ENC_FID	PRTA_SRC14_PREV_ENC_FID	PRTA_SRC13_CURR_ENC_FID	PRTA_SRC13_PREV_ENC_FID	PRTA_SRC12_CURR_ENC_FID	PRTA_SRC12_PREV_ENC_FID	PRTA_SRC11_CURR_ENC_FID	PRTA_SRC11_PREV_ENC_FID	PRTA_SRC10_CURR_ENC_FID	PRTA_SRC10_PREV_ENC_FID	PRTA_SRC9_CURR_ENC_FID	PRTA_SRC9_PREV_ENC_FID	PRTA_SRC8_CURR_ENC_FID	PRTA_SRC8_PREV_ENC_FID	PRTA_SRC7_CURR_ENC_FID	PRTA_SRC7_PREV_ENC_FID	PRTA_SRC6_CURR_ENC_FID	PRTA_SRC6_PREV_ENC_FID	PRTA_SRC5_CURR_ENC_FID	PRTA_SRC5_PREV_ENC_FID	PRTA_SRC4_CURR_ENC_FID	PRTA_SRC4_PREV_ENC_FID	PRTA_SRC3_CURR_ENC_FID	PRTA_SRC3_PREV_ENC_FID	PRTA_SRC2_CURR_ENC_FID	PRTA_SRC2_PREV_ENC_FID	PRTA_SRC1_CURR_ENC_FID	PRTA_SRC1_PREV_ENC_FID	PRTA_SRC0_CURR_ENC_FID	PRTA_SRC0_PREV_ENC_FID

Bits	Field Name	Description	Type	Reset
31	PRTA_SRC15_CURR_ENC_FID	For Source ID 15 from Port A. Encoder Field ID for Current Field	R	0x0
30	PRTA_SRC15_PREV_ENC_FID	For Source ID 15 from Port A. Encoder Field ID for Previous Field	R	0x0
29	PRTA_SRC14_CURR_ENC_FID	For Source ID 14 from Port A. Encoder Field ID for Current Field	R	0x0
28	PRTA_SRC14_PREV_ENC_FID	For Source ID 14 from Port A. Encoder Field ID for Previous Field	R	0x0
27	PRTA_SRC13_CURR_ENC_FID	For Source ID 13 from Port A. Encoder Field ID for Current Field	R	0x0
26	PRTA_SRC13_PREV_ENC_FID	For Source ID 13 from Port A. Encoder Field ID for Previous Field	R	0x0
25	PRTA_SRC12_CURR_ENC_FID	For Source ID 12 from Port A. Encoder Field ID for Current Field	R	0x0
24	PRTA_SRC12_PREV_ENC_FID	For Source ID 12 from Port A. Encoder Field ID for Previous Field	R	0x0
23	PRTA_SRC11_CURR_ENC_FID	For Source ID 11 from Port A. Encoder Field ID for Current Field	R	0x0
22	PRTA_SRC11_PREV_ENC_FID	For Source ID 11 from Port A. Encoder Field ID for Previous Field	R	0x0
21	PRTA_SRC10_CURR_ENC_FID	For Source ID 10 from Port A. Encoder Field ID for Current Field	R	0x0
20	PRTA_SRC10_PREV_ENC_FID	For Source ID 10 from Port A. Encoder Field ID for Previous Field	R	0x0
19	PRTA_SRC9_CURR_ENC_FID	For Source ID 9 from Port A. Encoder Field ID for Current Field	R	0x0
18	PRTA_SRC9_PREV_ENC_FID	For Source ID 9 from Port A. Encoder Field ID for Previous Field	R	0x0
17	PRTA_SRC8_CURR_ENC_FID	For Source ID 8 from Port A. Encoder Field ID for Current Field	R	0x0
16	PRTA_SRC8_PREV_ENC_FID	For Source ID 8 from Port A. Encoder Field ID for Previous Field	R	0x0
15	PRTA_SRC7_CURR_ENC_FID	For Source ID 7 from Port A. Encoder Field ID for Current Field	R	0x0
14	PRTA_SRC7_PREV_ENC_FID	For Source ID 7 from Port A. Encoder Field ID for Previous Field	R	0x0

Bits	Field Name	Description	Type	Reset
13	PRTA_SRC6_CURR_ENC_FID	For Source ID 6 from Port A. Encoder Field ID for Current Field	R	0x0
12	PRTA_SRC6_PREV_ENC_FID	For Source ID 6 from Port A. Encoder Field ID for Previous Field	R	0x0
11	PRTA_SRC5_CURR_ENC_FID	For Source ID 5 from Port A. Encoder Field ID for Current Field	R	0x0
10	PRTA_SRC5_PREV_ENC_FID	For Source ID 5 from Port A. Encoder Field ID for Previous Field	R	0x0
9	PRTA_SRC4_CURR_ENC_FID	For Source ID 4 from Port A. Encoder Field ID for Current Field	R	0x0
8	PRTA_SRC4_PREV_ENC_FID	For Source ID 4 from Port A. Encoder Field ID for Previous Field	R	0x0
7	PRTA_SRC3_CURR_ENC_FID	For Source ID 3 from Port A. Encoder Field ID for Current Field	R	0x0
6	PRTA_SRC3_PREV_ENC_FID	For Source ID 3 from Port A. Encoder Field ID for Previous Field	R	0x0
5	PRTA_SRC2_CURR_ENC_FID	For Source ID 2 from Port A. Encoder Field ID for Current Field	R	0x0
4	PRTA_SRC2_PREV_ENC_FID	For Source ID 2 from Port A. Encoder Field ID for Previous Field	R	0x0
3	PRTA_SRC1_CURR_ENC_FID	For Source ID 1 from Port A. Encoder Field ID for Current Field	R	0x0
2	PRTA_SRC1_PREV_ENC_FID	For Source ID 1 from Port A. Encoder Field ID for Previous Field	R	0x0
1	PRTA_SRC0_CURR_ENC_FID	For Source ID 0 from Port A. Encoder Field ID for Current Field	R	0x0
0	PRTA_SRC0_PREV_ENC_FID	For Source ID 0 from Port A. Encoder Field ID for Previous Field	R	0x0

Table 7-137. Register Call Summary for Register VIP_OUTPUT_PORT_A_ENC_FID

VIP Register Manual
- VIP Parser Register Summary: [2]

Table 7-138. VIP_OUTPUT_PORT_B_SRC_FID

Address Offset	0x0000 0028		
Physical Address	0x4897 5528 0x4897 5A28	**Instance**	VIP_Slice0_parser VIP_Slice1_parser
Description	Current and Previous Output Port B Source FID values		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PRTB_SRC15_CURR_SOURCE_FID	PRTB_SRC15_PREV_SOURCE_FID	PRTB_SRC14_CURR_SOURCE_FID	PRTB_SRC14_PREV_SOURCE_FID	PRTB_SRC13_CURR_SOURCE_FID	PRTB_SRC13_PREV_SOURCE_FID	PRTB_SRC12_CURR_SOURCE_FID	PRTB_SRC12_PREV_SOURCE_FID	PRTB_SRC11_CURR_SOURCE_FID	PRTB_SRC11_PREV_SOURCE_FID	PRTB_SRC10_CURR_SOURCE_FID	PRTB_SRC10_PREV_SOURCE_FID	PRTB_SRC9_CURR_SOURCE_FID	PRTB_SRC9_PREV_SOURCE_FID	PRTB_SRC8_CURR_SOURCE_FID	PRTB_SRC8_PREV_SOURCE_FID	PRTB_SRC7_CURR_SOURCE_FID	PRTB_SRC7_PREV_SOURCE_FID	PRTB_SRC6_CURR_SOURCE_FID	PRTB_SRC6_PREV_SOURCE_FID	PRTB_SRC5_CURR_SOURCE_FID	PRTB_SRC5_PREV_SOURCE_FID	PRTB_SRC4_CURR_SOURCE_FID	PRTB_SRC4_PREV_SOURCE_FID	PRTB_SRC3_CURR_SOURCE_FID	PRTB_SRC3_PREV_SOURCE_FID	PRTB_SRC2_CURR_SOURCE_FID	PRTB_SRC2_PREV_SOURCE_FID	PRTB_SRC1_CURR_SOURCE_FID	PRTB_SRC1_PREV_SOURCE_FID	PRTB_SRC0_CURR_SOURCE_FID	PRTB_SRC0_PREV_SOURCE_FID

Bits	Field Name	Description	Type	Reset
31	PRTB_SRC15_CURR_SOURCE_FID	For Source ID 15 from Port B. Source Field ID for Current Field	R	0x0
30	PRTB_SRC15_PREV_SOURCE_FID	For Source ID 15 from Port B. Source Field ID for Previous Field	R	0x0
29	PRTB_SRC14_CURR_SOURCE_FID	For Source ID 14 from Port B. Source Field ID for Current Field	R	0x0
28	PRTB_SRC14_PREV_SOURCE_FID	For Source ID 14 from Port B. Source Field ID for Previous Field	R	0x0
27	PRTB_SRC13_CURR_SOURCE_FID	For Source ID 13 from Port B. Source Field ID for Current Field	R	0x0
26	PRTB_SRC13_PREV_SOURCE_FID	For Source ID 13 from Port B. Source Field ID for Previous Field	R	0x0
25	PRTB_SRC12_CURR_SOURCE_FID	For Source ID 12 from Port B. Source Field ID for Current Field	R	0x0
24	PRTB_SRC12_PREV_SOURCE_FID	For Source ID 12 from Port B. Source Field ID for Previous Field	R	0x0
23	PRTB_SRC11_CURR_SOURCE_FID	For Source ID 11 from Port B. Source Field ID for Current Field	R	0x0
22	PRTB_SRC11_PREV_SOURCE_FID	For Source ID 11. from Port B Source Field ID for Previous Field	R	0x0
21	PRTB_SRC10_CURR_SOURCE_FID	For Source ID 10 from Port B. Source Field ID for Current Field	R	0x0
20	PRTB_SRC10_PREV_SOURCE_FID	For Source ID 10 from Port B. Source Field ID for Previous Field	R	0x0
19	PRTB_SRC9_CURR_SOURCE_FID	For Source ID 9 from Port B. Source Field ID for Current Field	R	0x0
18	PRTB_SRC9_PREV_SOURCE_FID	For Source ID 9 from Port B. Source Field ID for Previous Field	R	0x0
17	PRTB_SRC8_CURR_SOURCE_FID	For Source ID 8 from Port B. Source Field ID for Current Field	R	0x0
16	PRTB_SRC8_PREV_SOURCE_FID	For Source ID 8 from Port B. Source Field ID for Previous Field	R	0x0
15	PRTB_SRC7_CURR_SOURCE_FID	For Source ID 7 from Port B. Source Field ID for Current Field	R	0x0
14	PRTB_SRC7_PREV_SOURCE_FID	For Source ID 7 from Port B. Source Field ID for Previous Field	R	0x0
13	PRTB_SRC6_CURR_SOURCE_FID	For Source ID 6 from Port B. Source Field ID for Current Field	R	0x0
12	PRTB_SRC6_PREV_SOURCE_FID	For Source ID 6 from Port B. Source Field ID for Previous Field	R	0x0
11	PRTB_SRC5_CURR_SOURCE_FID	For Source ID 5 from Port B. Source Field ID for Current Field	R	0x0
10	PRTB_SRC5_PREV_SOURCE_FID	For Source ID 5 from Port B. Source Field ID for Previous Field	R	0x0
9	PRTB_SRC4_CURR_SOURCE_FID	For Source ID 4 from Port B. Source Field ID for Current Field	R	0x0
8	PRTB_SRC4_PREV_SOURCE_FID	For Source ID 4 from Port B. Source Field ID for Previous Field	R	0x0
7	PRTB_SRC3_CURR_SOURCE_FID	For Source ID 3 from Port B. Source Field ID for Current Field	R	0x0
6	PRTB_SRC3_PREV_SOURCE_FID	For Source ID 3 from Port B. Source Field ID for Previous Field	R	0x0
5	PRTB_SRC2_CURR_SOURCE_FID	For Source ID 2 from Port B. Source Field ID for Current Field	R	0x0
4	PRTB_SRC2_PREV_SOURCE_FID	For Source ID 2 from Port B. Source Field ID for Previous Field	R	0x0
3	PRTB_SRC1_CURR_SOURCE_FID	For Source ID 1 from Port B. Source Field ID for Current Field	R	0x0

Bits	Field Name	Description	Type	Reset
2	PRTB_SRC1_PREV_SOURCE_FID	For Source ID 1 from Port B. Source Field ID for Previous Field	R	0x0
1	PRTB_SRC0_CURR_SOURCE_FID	For Source ID 0 from Port B. Source Field ID for Current Field	R	0x0
0	PRTB_SRC0_PREV_SOURCE_FID	For Source ID 0 from Port B. Source Field ID for Previous Field	R	0x0

Table 7-139. Register Call Summary for Register VIP_OUTPUT_PORT_B_SRC_FID

VIP Functional Description
- Source Multiplexing: [0]

VIP Register Manual
- VIP Parser Register Summary: [3]

Table 7-140. VIP_OUTPUT_PORT_B_ENC_FID

Address Offset	0x0000 002C		
Physical Address	0x4897 552C 0x4897 5A2C	**Instance**	VIP_Slice0_parser VIP_Slice1_parser
Description	Current and Previous Output Port B Encoder FID values		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PRTB_SRC15_CURR_ENC_FID	PRTB_SRC15_PREV_ENC_FID	PRTB_SRC14_CURR_ENC_FID	PRTB_SRC14_PREV_ENC_FID	PRTB_SRC13_CURR_ENC_FID	PRTB_SRC13_PREV_ENC_FID	PRTB_SRC12_CURR_ENC_FID	PRTB_SRC12_PREV_ENC_FID	PRTB_SRC11_CURR_ENC_FID	PRTB_SRC11_PREV_ENC_FID	PRTB_SRC10_CURR_ENC_FID	PRTB_SRC10_PREV_ENC_FID	PRTB_SRC9_CURR_ENC_FID	PRTB_SRC9_PREV_ENC_FID	PRTB_SRC8_CURR_ENC_FID	PRTB_SRC8_PREV_ENC_FID	PRTB_SRC7_CURR_ENC_FID	PRTB_SRC7_PREV_ENC_FID	PRTB_SRC6_CURR_ENC_FID	PRTB_SRC6_PREV_ENC_FID	PRTB_SRC5_CURR_ENC_FID	PRTB_SRC5_PREV_ENC_FID	PRTB_SRC4_CURR_ENC_FID	PRTB_SRC4_PREV_ENC_FID	PRTB_SRC3_CURR_ENC_FID	PRTB_SRC3_PREV_ENC_FID	PRTB_SRC2_CURR_ENC_FID	PRTB_SRC2_PREV_ENC_FID	PRTB_SRC1_CURR_ENC_FID	PRTB_SRC1_PREV_ENC_FID	PRTB_SRC0_CURR_ENC_FID	PRTB_SRC0_PREV_ENC_FID

Bits	Field Name	Description	Type	Reset
31	PRTB_SRC15_CURR_ENC_FID	For Source ID 15 from Port B. Encoder Field ID for Current Field	R	0x0
30	PRTB_SRC15_PREV_ENC_FID	For Source ID 15 from Port B. Encoder Field ID for Previous Field	R	0x0
29	PRTB_SRC14_CURR_ENC_FID	For Source ID 14 from Port B. Encoder Field ID for Current Field	R	0x0
28	PRTB_SRC14_PREV_ENC_FID	For Source ID 14 from Port B. Encoder Field ID for Previous Field	R	0x0
27	PRTB_SRC13_CURR_ENC_FID	For Source ID 13 from Port B. Encoder Field ID for Current Field	R	0x0
26	PRTB_SRC13_PREV_ENC_FID	For Source ID 13 from Port B. Encoder Field ID for Previous Field	R	0x0
25	PRTB_SRC12_CURR_ENC_FID	For Source ID 12 from Port B. Encoder Field ID for Current Field	R	0x0
24	PRTB_SRC12_PREV_ENC_FID	For Source ID 12 from Port B. Encoder Field ID for Previous Field	R	0x0
23	PRTB_SRC11_CURR_ENC_FID	For Source ID 11 from Port B. Encoder Field ID for Current Field	R	0x0
22	PRTB_SRC11_PREV_ENC_FID	For Source ID 11 from Port B. Encoder Field ID for Previous Field	R	0x0

Bits	Field Name	Description	Type	Reset
21	PRTB_SRC10_CURR_ENC_FID	For Source ID 10 from Port B. Encoder Field ID for Current Field	R	0x0
20	PRTB_SRC10_PREV_ENC_FID	For Source ID 10 from Port B. Encoder Field ID for Previous Field	R	0x0
19	PRTB_SRC9_CURR_ENC_FID	For Source ID 9 from Port B. Encoder Field ID for Current Field	R	0x0
18	PRTB_SRC9_PREV_ENC_FID	For Source ID 9 from Port B. Encoder Field ID for Previous Field	R	0x0
17	PRTB_SRC8_CURR_ENC_FID	For Source ID 8 from Port B. Encoder Field ID for Current Field	R	0x0
16	PRTB_SRC8_PREV_ENC_FID	For Source ID 8 from Port B. Encoder Field ID for Previous Field	R	0x0
15	PRTB_SRC7_CURR_ENC_FID	For Source ID 7 from Port B. Encoder Field ID for Current Field	R	0x0
14	PRTB_SRC7_PREV_ENC_FID	For Source ID 7 from Port B. Encoder Field ID for Previous Field	R	0x0
13	PRTB_SRC6_CURR_ENC_FID	For Source ID 6 from Port B. Encoder Field ID for Current Field	R	0x0
12	PRTB_SRC6_PREV_ENC_FID	For Source ID 6 from Port B. Encoder Field ID for Previous Field	R	0x0
11	PRTB_SRC5_CURR_ENC_FID	For Source ID 5 from Port B. Encoder Field ID for Current Field	R	0x0
10	PRTB_SRC5_PREV_ENC_FID	For Source ID 5 from Port B. Encoder Field ID for Previous Field	R	0x0
9	PRTB_SRC4_CURR_ENC_FID	For Source ID 4 from Port B. Encoder Field ID for Current Field	R	0x0
8	PRTB_SRC4_PREV_ENC_FID	For Source ID 4 from Port B. Encoder Field ID for Previous Field	R	0x0
7	PRTB_SRC3_CURR_ENC_FID	For Source ID 3 from Port B. Encoder Field ID for Current Field	R	0x0
6	PRTB_SRC3_PREV_ENC_FID	For Source ID 3 from Port B. Encoder Field ID for Previous Field	R	0x0
5	PRTB_SRC2_CURR_ENC_FID	For Source ID 2 from Port B. Encoder Field ID for Current Field	R	0x0
4	PRTB_SRC2_PREV_ENC_FID	For Source ID 2 from Port B. Encoder Field ID for Previous Field	R	0x0
3	PRTB_SRC1_CURR_ENC_FID	For Source ID 1 from Port B. Encoder Field ID for Current Field	R	0x0
2	PRTB_SRC1_PREV_ENC_FID	For Source ID 1 from Port B. Encoder Field ID for Previous Field	R	0x0
1	PRTB_SRC0_CURR_ENC_FID	For Source ID 0 from Port B. Encoder Field ID for Current Field	R	0x0
0	PRTB_SRC0_PREV_ENC_FID	For Source ID 0 from Port B. Encoder Field ID for Previous Field	R	0x0

Table 7-141. Register Call Summary for Register VIP_OUTPUT_PORT_B_ENC_FID

VIP Register Manual
- VIP Parser Register Summary: [2]

Table 7-142. VIP_OUTPUT_PORT_A_SRC0_SIZE

Address Offset	0x0000 0030		
Physical Address	0x4897 5530 0x4897 5A30	**Instance**	VIP_Slice0_parser VIP_Slice1_parser
Description	Width and Height for Source 0		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								PRTA_SRC0_WIDTH								RESERVED								PRTA_SRC0_HEIGHT							

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:16	PRTA_SRC0_WIDTH	On Port A. Width of Source ID 0	R	0x0
15:11	RESERVED		R	0x0
10:0	PRTA_SRC0_HEIGHT	On Port A. Height of Source ID 0	R	0x0

Table 7-143. Register Call Summary for Register VIP_OUTPUT_PORT_A_SRC0_SIZE

VIP Functional Description
- Source Multiplexing: [0]

VIP Register Manual
- VIP Parser Register Summary: [3]

Table 7-144. VIP_OUTPUT_PORT_A_SRC1_SIZE

Address Offset	0x0000 0034		
Physical Address	0x4897 5534 0x4897 5A34	**Instance**	VIP_Slice0_parser VIP_Slice1_parser
Description	Width and Height for Source 1		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								PRTA_SRC1_WIDTH								RESERVED								PRTA_SRC1_HEIGHT							

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:16	PRTA_SRC1_WIDTH	On Port A. Width of Source ID 1	R	0x0
15:11	RESERVED		R	0x0
10:0	PRTA_SRC1_HEIGHT	On Port A. Height of Source ID 1	R	0x0

Table 7-145. Register Call Summary for Register VIP_OUTPUT_PORT_A_SRC1_SIZE

VIP Register Manual
- VIP Parser Register Summary: [2]

Table 7-146. VIP_OUTPUT_PORT_A_SRC2_SIZE

Address Offset	0x0000 0038		
Physical Address	0x4897 5538 0x4897 5A38	**Instance**	VIP_Slice0_parser VIP_Slice1_parser
Description	Width and Height for Source 2		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								PRTA_SRC2_WIDTH								RESERVED								PRTA_SRC2_HEIGHT							

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:16	PRTA_SRC2_WIDTH	On Port A. Width of Source ID 2	R	0x0
15:11	RESERVED		R	0x0
10:0	PRTA_SRC2_HEIGHT	On Port A. Height of Source ID 2	R	0x0

Table 7-147. Register Call Summary for Register VIP_OUTPUT_PORT_A_SRC2_SIZE

VIP Register Manual
- VIP Parser Register Summary: [2]

Table 7-148. VIP_OUTPUT_PORT_A_SRC3_SIZE

Address Offset	0x0000 003C		
Physical Address	0x4897 553C 0x4897 5A3C	**Instance**	VIP_Slice0_parser VIP_Slice1_parser
Description	Width and Height for Source 3		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								PRTA_SRC3_WIDTH								RESERVED								PRTA_SRC3_HEIGHT							

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:16	PRTA_SRC3_WIDTH	On Port A. Width of Source ID 3	R	0x0
15:11	RESERVED		R	0x0
10:0	PRTA_SRC3_HEIGHT	On Port A. Height of Source ID 3	R	0x0

Table 7-149. Register Call Summary for Register VIP_OUTPUT_PORT_A_SRC3_SIZE

VIP Register Manual
- VIP Parser Register Summary: [2]

Table 7-150. VIP_OUTPUT_PORT_A_SRC4_SIZE

Address Offset	0x0000 0040		
Physical Address	0x4897 5540 0x4897 5A40	**Instance**	VIP_Slice0_parser VIP_Slice1_parser
Description	Width and Height for Source 4		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								PRTA_SRC4_WIDTH								RESERVED								PRTA_SRC4_HEIGHT							

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:16	PRTA_SRC4_WIDTH	On Port A. Width of Source ID 4	R	0x0
15:11	RESERVED		R	0x0
10:0	PRTA_SRC4_HEIGHT	On Port A. Height of Source ID 4	R	0x0

Table 7-151. Register Call Summary for Register VIP_OUTPUT_PORT_A_SRC4_SIZE

VIP Register Manual
- VIP Parser Register Summary: [2]

Table 7-152. VIP_OUTPUT_PORT_A_SRC5_SIZE

Address Offset	0x0000 0044		
Physical Address	0x4897 5544 0x4897 5A44	**Instance**	VIP_Slice0_parser VIP_Slice1_parser
Description	Width and Height for Source 5		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								PRTA_SRC5_WIDTH								RESERVED								PRTA_SRC5_HEIGHT							

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:16	PRTA_SRC5_WIDTH	On Port A. Width of Source ID 5	R	0x0
15:11	RESERVED		R	0x0
10:0	PRTA_SRC5_HEIGHT	On Port A. Height of Source ID 5	R	0x0

Table 7-153. Register Call Summary for Register VIP_OUTPUT_PORT_A_SRC5_SIZE

VIP Register Manual
- VIP Parser Register Summary: [2]

Table 7-154. VIP_OUTPUT_PORT_A_SRC6_SIZE

Address Offset	0x0000 0048		
Physical Address	0x4897 5548 0x4897 5A48	**Instance**	VIP_Slice0_parser VIP_Slice1_parser
Description	Width and Height for Source 6		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								PRTA_SRC6_WIDTH								RESERVED								PRTA_SRC6_HEIGHT							

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:16	PRTA_SRC6_WIDTH	On Port A. Width of Source ID 6	R	0x0
15:11	RESERVED		R	0x0
10:0	PRTA_SRC6_HEIGHT	On Port A. Height of Source ID 6	R	0x0

Table 7-155. Register Call Summary for Register VIP_OUTPUT_PORT_A_SRC6_SIZE

VIP Register Manual
- VIP Parser Register Summary: [2]

Table 7-156. VIP_OUTPUT_PORT_A_SRC7_SIZE

Address Offset	0x0000 004C		
Physical Address	0x4897 554C 0x4897 5A4C	**Instance**	VIP_Slice0_parser VIP_Slice1_parser
Description	Width and Height for Source 7		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								PRTA_SRC7_WIDTH								RESERVED								PRTA_SRC7_HEIGHT							

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:16	PRTA_SRC7_WIDTH	On Port A. Width of Source ID 7	R	0x0
15:11	RESERVED		R	0x0
10:0	PRTA_SRC7_HEIGHT	On Port A. Height of Source ID 7	R	0x0

Table 7-157. Register Call Summary for Register VIP_OUTPUT_PORT_A_SRC7_SIZE

VIP Register Manual
- VIP Parser Register Summary: [2]

Table 7-158. VIP_OUTPUT_PORT_A_SRC8_SIZE

Address Offset	0x0000 0050		
Physical Address	0x4897 5550 0x4897 5A50	**Instance**	VIP_Slice0_parser VIP_Slice1_parser
Description	Width and Height for Source 8		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								PRTA_SRC8_WIDTH								RESERVED								PRTA_SRC8_HEIGHT							

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:16	PRTA_SRC8_WIDTH	On Port A. Width of Source ID 8	R	0x0
15:11	RESERVED		R	0x0
10:0	PRTA_SRC8_HEIGHT	On Port A. Height of Source ID 8	R	0x0

Table 7-159. Register Call Summary for Register VIP_OUTPUT_PORT_A_SRC8_SIZE

VIP Register Manual
- VIP Parser Register Summary: [2]

Table 7-160. VIP_OUTPUT_PORT_A_SRC9_SIZE

Address Offset	0x0000 0054		
Physical Address	0x4897 5554 0x4897 5A54	**Instance**	VIP_Slice0_parser VIP_Slice1_parser
Description	Width and Height for Source 9		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								PRTA_SRC9_WIDTH								RESERVED								PRTA_SRC9_HEIGHT							

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:16	PRTA_SRC9_WIDTH	On Port A. Width of Source ID 9	R	0x0
15:11	RESERVED		R	0x0
10:0	PRTA_SRC9_HEIGHT	On Port A. Height of Source ID 9	R	0x0

Table 7-161. Register Call Summary for Register VIP_OUTPUT_PORT_A_SRC9_SIZE

VIP Register Manual
- VIP Parser Register Summary: [2]

Table 7-162. VIP_OUTPUT_PORT_A_SRC10_SIZE

Address Offset	0x0000 0058		
Physical Address	0x4897 5558 0x4897 5A58	**Instance**	VIP_Slice0_parser VIP_Slice1_parser
Description	Width and Height for Source 10		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								PRTA_SRC10_WIDTH								RESERVED								PRTA_SRC10_HEIGHT							

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:16	PRTA_SRC10_WIDTH	On Port A. Width of Source ID 10	R	0x0
15:11	RESERVED		R	0x0
10:0	PRTA_SRC10_HEIGHT	On Port A. Height of Source ID 10	R	0x0

Table 7-163. Register Call Summary for Register VIP_OUTPUT_PORT_A_SRC10_SIZE

VIP Register Manual
- VIP Parser Register Summary: [2]

Table 7-164. VIP_OUTPUT_PORT_A_SRC11_SIZE

Address Offset	0x0000 005C		
Physical Address	0x4897 555C 0x4897 5A5C	**Instance**	VIP_Slice0_parser VIP_Slice1_parser
Description	Width and Height for Source 11		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								PRTA_SRC11_WIDTH								RESERVED								PRTA_SRC11_HEIGHT							

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:16	PRTA_SRC11_WIDTH	On Port A. Width of Source ID 11	R	0x0
15:11	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
10:0	PRTA_SRC11_HEIGHT	On Port A. Height of Source ID 11	R	0x0

Table 7-165. Register Call Summary for Register VIP_OUTPUT_PORT_A_SRC11_SIZE

VIP Register Manual
- VIP Parser Register Summary: [2]

Table 7-166. VIP_OUTPUT_PORT_A_SRC12_SIZE

Address Offset	0x0000 0060		
Physical Address	0x4897 5560 0x4897 5A60	**Instance**	VIP_Slice0_parser VIP_Slice1_parser
Description	Width and Height for Source 12		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								PRTA_SRC12_WIDTH								RESERVED								PRTA_SRC12_HEIGHT							

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:16	PRTA_SRC12_WIDTH	On Port A. Width of Source ID 12	R	0x0
15:11	RESERVED		R	0x0
10:0	PRTA_SRC12_HEIGHT	On Port A. Height of Source ID 12	R	0x0

Table 7-167. Register Call Summary for Register VIP_OUTPUT_PORT_A_SRC12_SIZE

VIP Register Manual
- VIP Parser Register Summary: [2]

Table 7-168. VIP_OUTPUT_PORT_A_SRC13_SIZE

Address Offset	0x0000 0064		
Physical Address	0x4897 5564 0x4897 5A64	**Instance**	VIP_Slice0_parser VIP_Slice1_parser
Description	Width and Height for Source 13		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								PRTA_SRC13_WIDTH								RESERVED								PRTA_SRC13_HEIGHT							

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:16	PRTA_SRC13_WIDTH	On Port A. Width of Source ID 13	R	0x0
15:11	RESERVED		R	0x0
10:0	PRTA_SRC13_HEIGHT	On Port A. Height of Source ID 13	R	0x0

Table 7-169. Register Call Summary for Register VIP_OUTPUT_PORT_A_SRC13_SIZE

VIP Register Manual
- VIP Parser Register Summary: [2]

Table 7-170. VIP_OUTPUT_PORT_A_SRC14_SIZE

Address Offset	0x0000 0068		
Physical Address	0x4897 5568 0x4897 5A68	**Instance**	VIP_Slice0_parser VIP_Slice1_parser
Description	Width and Height for Source 14		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								PRTA_SRC14_WIDTH								RESERVED								PRTA_SRC14_HEIGHT							

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:16	PRTA_SRC14_WIDTH	On Port A. Width of Source ID 14	R	0x0
15:11	RESERVED		R	0x0
10:0	PRTA_SRC14_HEIGHT	On Port A. Height of Source ID 14	R	0x0

Table 7-171. Register Call Summary for Register VIP_OUTPUT_PORT_A_SRC14_SIZE

VIP Register Manual
- VIP Parser Register Summary: [2]

Table 7-172. VIP_OUTPUT_PORT_A_SRC15_SIZE

Address Offset	0x0000 006C		
Physical Address	0x4897 556C 0x4897 5A6C	**Instance**	VIP_Slice0_parser VIP_Slice1_parser
Description	Width and Height for Source 15		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								PRTA_SRC15_WIDTH								RESERVED								PRTA_SRC15_HEIGHT							

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:16	PRTA_SRC15_WIDTH	On Port A. Width of Source ID 15	R	0x0
15:11	RESERVED		R	0x0
10:0	PRTA_SRC15_HEIGHT	On Port A. Height of Source ID 15	R	0x0

Table 7-173. Register Call Summary for Register VIP_OUTPUT_PORT_A_SRC15_SIZE

VIP Functional Description
- Source Multiplexing: [0]

VIP Register Manual
- VIP Parser Register Summary: [3]

Table 7-174. VIP_OUTPUT_PORT_B_SRC0_SIZE

Address Offset	0x0000 0070		
Physical Address	0x4897 5570 0x4897 5A70	**Instance**	VIP_Slice0_parser VIP_Slice1_parser
Description	Width and Height for Source 0		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								PRTB_SRC0_WIDTH								RESERVED								PRTB_SRC0_HEIGHT							

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:16	PRTB_SRC0_WIDTH	On Port B. Width of Source ID 0	R	0x0
15:11	RESERVED		R	0x0
10:0	PRTB_SRC0_HEIGHT	On Port B. Height of Source ID 0	R	0x0

Table 7-175. Register Call Summary for Register VIP_OUTPUT_PORT_B_SRC0_SIZE

VIP Functional Description
- Source Multiplexing: [0]

VIP Register Manual
- VIP Parser Register Summary: [3]

Table 7-176. VIP_OUTPUT_PORT_B_SRC1_SIZE

Address Offset	0x0000 0074		
Physical Address	0x4897 5574 0x4897 5A74	**Instance**	VIP_Slice0_parser VIP_Slice1_parser
Description	Width and Height for Source 1		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								PRTB_SRC1_WIDTH								RESERVED								PRTB_SRC1_HEIGHT							

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:16	PRTB_SRC1_WIDTH	On Port B. Width of Source ID 1	R	0x0
15:11	RESERVED		R	0x0
10:0	PRTB_SRC1_HEIGHT	On Port B. Height of Source ID 1	R	0x0

Table 7-177. Register Call Summary for Register VIP_OUTPUT_PORT_B_SRC1_SIZE

VIP Register Manual
- VIP Parser Register Summary: [2]

Table 7-178. VIP_OUTPUT_PORT_B_SRC2_SIZE

Address Offset	0x0000 0078		
Physical Address	0x4897 5578 0x4897 5A78	**Instance**	VIP_Slice0_parser VIP_Slice1_parser
Description	Width and Height for Source 2		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								PRTB_SRC2_WIDTH								RESERVED								PRTB_SRC2_HEIGHT							

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:16	PRTB_SRC2_WIDTH	On Port B. Width of Source ID 2	R	0x0
15:11	RESERVED		R	0x0
10:0	PRTB_SRC2_HEIGHT	On Port B. Height of Source ID 2	R	0x0

Table 7-179. Register Call Summary for Register VIP_OUTPUT_PORT_B_SRC2_SIZE

VIP Register Manual
- VIP Parser Register Summary: [2]

Table 7-180. VIP_OUTPUT_PORT_B_SRC3_SIZE

Address Offset	0x0000 007C		
Physical Address	0x4897 557C 0x4897 5A7C	**Instance**	VIP_Slice0_parser VIP_Slice1_parser
Description	Width and Height for Source 3		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								PRTB_SRC3_WIDTH								RESERVED								PRTB_SRC3_HEIGHT							

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:16	PRTB_SRC3_WIDTH	On Port B. Width of Source ID 3	R	0x0
15:11	RESERVED		R	0x0
10:0	PRTB_SRC3_HEIGHT	On Port B. Height of Source ID 3	R	0x0

Table 7-181. Register Call Summary for Register VIP_OUTPUT_PORT_B_SRC3_SIZE

VIP Register Manual
- VIP Parser Register Summary: [2]

Table 7-182. VIP_OUTPUT_PORT_B_SRC4_SIZE

Address Offset	0x0000 0080		
Physical Address	0x4897 5580 0x4897 5A80	**Instance**	VIP_Slice0_parser VIP_Slice1_parser
Description	Width and Height for Source 4		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								PRTB_SRC4_WIDTH								RESERVED								PRTB_SRC4_HEIGHT							

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:16	PRTB_SRC4_WIDTH	On Port B. Width of Source ID 4	R	0x0
15:11	RESERVED		R	0x0
10:0	PRTB_SRC4_HEIGHT	On Port B. Height of Source ID 4	R	0x0

Table 7-183. Register Call Summary for Register VIP_OUTPUT_PORT_B_SRC4_SIZE

VIP Register Manual
- VIP Parser Register Summary: [2]

Table 7-184. VIP_OUTPUT_PORT_B_SRC5_SIZE

Address Offset	0x0000 0084		
Physical Address	0x4897 5584 0x4897 5A84	**Instance**	VIP_Slice0_parser VIP_Slice1_parser
Description	Width and Height for Source 5		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								PRTB_SRC5_WIDTH								RESERVED								PRTB_SRC5_HEIGHT							

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:16	PRTB_SRC5_WIDTH	On Port B. Width of Source ID 5	R	0x0
15:11	RESERVED		R	0x0
10:0	PRTB_SRC5_HEIGHT	On Port B. Height of Source ID 5	R	0x0

Table 7-185. Register Call Summary for Register VIP_OUTPUT_PORT_B_SRC5_SIZE

VIP Register Manual
- VIP Parser Register Summary: [2]

Table 7-186. VIP_OUTPUT_PORT_B_SRC6_SIZE

Address Offset	0x0000 0088		
Physical Address	0x4897 5588 0x4897 5A88	**Instance**	VIP_Slice0_parser VIP_Slice1_parser
Description	Width and Height for Source 6		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								PRTB_SRC6_WIDTH								RESERVED								PRTB_SRC6_HEIGHT							

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:16	PRTB_SRC6_WIDTH	On Port B. Width of Source ID 6	R	0x0
15:11	RESERVED		R	0x0
10:0	PRTB_SRC6_HEIGHT	On Port B. Height of Source ID 6	R	0x0

Table 7-187. Register Call Summary for Register VIP_OUTPUT_PORT_B_SRC6_SIZE

VIP Register Manual
- VIP Parser Register Summary: [2]

Table 7-188. VIP_OUTPUT_PORT_B_SRC7_SIZE

Address Offset	0x0000 008C		
Physical Address	0x4897 558C 0x4897 5A8C	**Instance**	VIP_Slice0_parser VIP_Slice1_parser
Description	Width and Height for Source 7		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								PRTB_SRC7_WIDTH								RESERVED								PRTB_SRC7_HEIGHT							

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:16	PRTB_SRC7_WIDTH	On Port B. Width of Source ID 7	R	0x0
15:11	RESERVED		R	0x0
10:0	PRTB_SRC7_HEIGHT	On Port B. Height of Source ID 7	R	0x0

Table 7-189. Register Call Summary for Register VIP_OUTPUT_PORT_B_SRC7_SIZE

VIP Register Manual
- VIP Parser Register Summary: [2]

Table 7-190. VIP_OUTPUT_PORT_B_SRC8_SIZE

Address Offset	0x0000 0090		
Physical Address	0x4897 5590 0x4897 5A90	**Instance**	VIP_Slice0_parser VIP_Slice1_parser
Description	Width and Height for Source 8		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								PRTB_SRC8_WIDTH								RESERVED								PRTB_SRC8_HEIGHT							

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:16	PRTB_SRC8_WIDTH	On Port B. Width of Source ID 8	R	0x0
15:11	RESERVED		R	0x0
10:0	PRTB_SRC8_HEIGHT	On Port B. Height of Source ID 8	R	0x0

Table 7-191. Register Call Summary for Register VIP_OUTPUT_PORT_B_SRC8_SIZE

VIP Register Manual
- VIP Parser Register Summary: [2]

Table 7-192. VIP_OUTPUT_PORT_B_SRC9_SIZE

Address Offset	0x0000 0094		
Physical Address	0x4897 5594 0x4897 5A94	**Instance**	VIP_Slice0_parser VIP_Slice1_parser
Description	Width and Height for Source 9		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								PRTB_SRC9_WIDTH								RESERVED								PRTB_SRC9_HEIGHT							

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:16	PRTB_SRC9_WIDTH	On Port B. Width of Source ID 9	R	0x0
15:11	RESERVED		R	0x0
10:0	PRTB_SRC9_HEIGHT	On Port B. Height of Source ID 9	R	0x0

Table 7-193. Register Call Summary for Register VIP_OUTPUT_PORT_B_SRC9_SIZE

VIP Register Manual
- VIP Parser Register Summary: [2]

Table 7-194. VIP_OUTPUT_PORT_B_SRC10_SIZE

Address Offset	0x0000 0098		
Physical Address	0x4897 5598 0x4897 5A98	**Instance**	VIP_Slice0_parser VIP_Slice1_parser
Description	Width and Height for Source 10		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								PRTB_SRC10_WIDTH								RESERVED								PRTB_SRC10_HEIGHT							

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:16	PRTB_SRC10_WIDTH	On Port B. Width of Source ID 10	R	0x0
15:11	RESERVED		R	0x0
10:0	PRTB_SRC10_HEIGHT	On Port B. Height of Source ID 10	R	0x0

Table 7-195. Register Call Summary for Register VIP_OUTPUT_PORT_B_SRC10_SIZE

VIP Register Manual
- VIP Parser Register Summary: [2]

Table 7-196. VIP_OUTPUT_PORT_B_SRC11_SIZE

Address Offset	0x0000 009C		
Physical Address	0x4897 559C 0x4897 5A9C	**Instance**	VIP_Slice0_parser VIP_Slice1_parser
Description	Width and Height for Source 11		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								PRTB_SRC11_WIDTH								RESERVED								PRTB_SRC11_HEIGHT							

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:16	PRTB_SRC11_WIDTH	On Port B. Width of Source ID 11	R	0x0
15:11	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
10:0	PRTB_SRC11_HEIGHT	On Port B. Height of Source ID 11	R	0x0

Table 7-197. Register Call Summary for Register VIP_OUTPUT_PORT_B_SRC11_SIZE

VIP Register Manual
- VIP Parser Register Summary: [2]

Table 7-198. VIP_OUTPUT_PORT_B_SRC12_SIZE

Address Offset	0x0000 00A0		
Physical Address	0x4897 55A0 0x4897 5AA0	**Instance**	VIP_Slice0_parser VIP_Slice1_parser
Description	Width and Height for Source 12		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								PRTB_SRC12_WIDTH								RESERVED								PRTB_SRC12_HEIGHT							

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:16	PRTB_SRC12_WIDTH	On Port B. Width of Source ID 12	R	0x0
15:11	RESERVED		R	0x0
10:0	PRTB_SRC12_HEIGHT	On Port B. Height of Source ID 12	R	0x0

Table 7-199. Register Call Summary for Register VIP_OUTPUT_PORT_B_SRC12_SIZE

VIP Register Manual
- VIP Parser Register Summary: [2]

Table 7-200. VIP_OUTPUT_PORT_B_SRC13_SIZE

Address Offset	0x0000 00A4		
Physical Address	0x4897 55A4 0x4897 5AA4	**Instance**	VIP_Slice0_parser VIP_Slice1_parser
Description	Width and Height for Source 13		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								PRTB_SRC13_WIDTH								RESERVED								PRTB_SRC13_HEIGHT							

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:16	PRTB_SRC13_WIDTH	On Port B. Width of Source ID 13	R	0x0
15:11	RESERVED		R	0x0
10:0	PRTB_SRC13_HEIGHT	On Port B. Height of Source ID 13	R	0x0

Table 7-201. Register Call Summary for Register VIP_OUTPUT_PORT_B_SRC13_SIZE

VIP Register Manual
- VIP Parser Register Summary: [2]

Table 7-202. VIP_OUTPUT_PORT_B_SRC14_SIZE

Address Offset	0x0000 00A8		
Physical Address	0x4897 55A8 0x4897 5AA8	**Instance**	VIP_Slice0_parser VIP_Slice1_parser
Description	Width and Height for Source 14		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								PRTB_SRC14_WIDTH								RESERVED								PRTB_SRC14_HEIGHT							

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:16	PRTB_SRC14_WIDTH	On Port B. Width of Source ID 14	R	0x0
15:11	RESERVED		R	0x0
10:0	PRTB_SRC14_HEIGHT	On Port B. Height of Source ID 14	R	0x0

Table 7-203. Register Call Summary for Register VIP_OUTPUT_PORT_B_SRC14_SIZE

VIP Register Manual
- VIP Parser Register Summary: [2]

Table 7-204. VIP_OUTPUT_PORT_B_SRC15_SIZE

Address Offset	0x0000 00AC		
Physical Address	0x4897 55AC 0x4897 5AAC	**Instance**	VIP_Slice0_parser VIP_Slice1_parser
Description	Width and Height for Source 15		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								PRTB_SRC15_WIDTH								RESERVED								PRTB_SRC15_HEIGHT							

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:16	PRTB_SRC15_WIDTH	On Port B. Width of Source ID 15	R	0x0
15:11	RESERVED		R	0x0
10:0	PRTB_SRC15_HEIGHT	On Port B. Height of Source ID 15	R	0x0

Table 7-205. Register Call Summary for Register VIP_OUTPUT_PORT_B_SRC15_SIZE

VIP Functional Description
- Source Multiplexing: [0]

VIP Register Manual
- VIP Parser Register Summary: [3]

Table 7-206. VIP_PORT_A_VDET_VEC

Address Offset	0x0000 00B0		
Physical Address	0x4897 55B0 0x4897 5AB0	**Instance**	VIP_Slice0_parser VIP_Slice1_parser
Description	Each bit represents the VDET bit setting for Line Mux Mode		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PRTA_VDET_VEC																															

Bits	Field Name	Description	Type	Reset
31:0	PRTA_VDET_VEC	For Embedded Sync Only In Line Mux Mode. each bit represents the vdet value on Port A for the corresponding source id. This vector is meaningless for 1x/2x/4x mux modes.	R	0x0

Table 7-207. Register Call Summary for Register VIP_PORT_A_VDET_VEC

VIP Functional Description • VDET Interrupt: [0]
VIP Register Manual • VIP Parser Register Summary: [3]

Table 7-208. VIP_PORT_B_VDET_VEC

Address Offset	0x0000 00B4		
Physical Address	0x4897 55B4 0x4897 5AB4	**Instance**	VIP_Slice0_parser VIP_Slice1_parser
Description	Each bit represents the VDET bit setting for Line Mux Mode		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PRTB_VDET_VEC																															

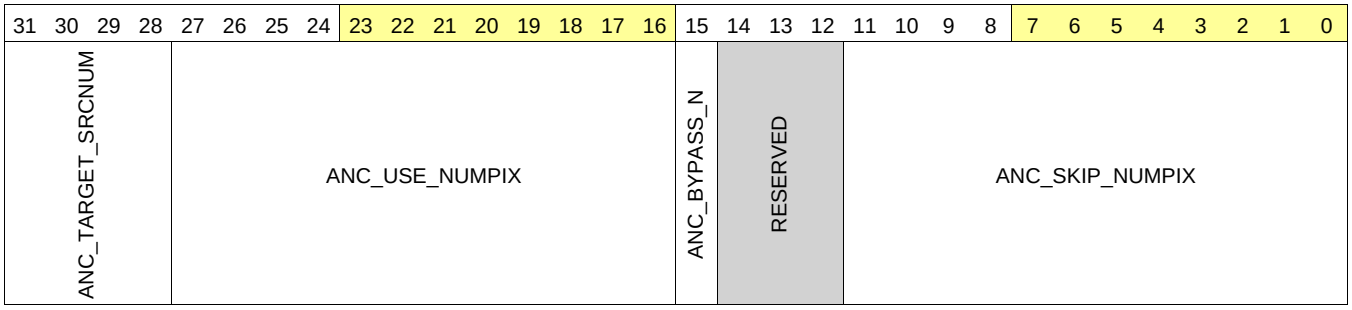
Bits	Field Name	Description	Type	Reset
31:0	PRTB_VDET_VEC	For Embedded Sync Only In Line Mux Mode. each bit represents the vdet value on Port B for the corresponding source id. This vector is meaningless for 1x/2x/4x mux modes.	R	0x0

Table 7-209. Register Call Summary for Register VIP_PORT_B_VDET_VEC

VIP Functional Description • VDET Interrupt: [0]
VIP Register Manual • VIP Parser Register Summary: [3]

Table 7-210. VIP_ANC_CROP_HORZ_PORT_A

Address Offset	0x0000 00B8		
Physical Address	0x4897 55B8 0x4897 5AB8	**Instance**	VIP_Slice0_parser VIP_Slice1_parser
Description	Ancillary Cropping Configuration for Input Port A		
Type	RW		



Bits	Field Name	Description	Type	Reset
31:28	ANC_TARGET_SRCNUM	The cropping module can work on only one srcnum. specified in this field. for each dss_vip_parser output port (Ancillary data).	RW	0x0
27:16	ANC_USE_NUMPIX	When cropping, the number of pixels to keep after the skip_numpix value. skip_numpix + use_numpix must be smaller than the original line width.	RW	0x0
15	ANC_BYPASS_N	0 = Bypass cropping module 1 = Cropping module enabled	RW	0x0
14:12	RESERVED		R	0x0
11:0	ANC_SKIP_NUMPIX	The number of pixels to crop from the beginning of each line.	RW	0x0

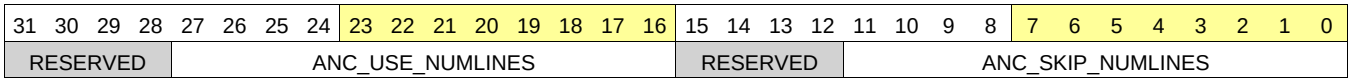
Table 7-211. Register Call Summary for Register VIP_ANC_CROP_HORZ_PORT_A

VIP Functional Description
- Ancillary and Active Video Cropping: [0][1][2][3]

VIP Register Manual
- VIP Parser Register Summary: [6]

Table 7-212. VIP_ANC_CROP_VERT_PORT_A

Address Offset	0x0000 00BC		
Physical Address	0x4897 55BC 0x4897 5ABC	**Instance**	VIP_Slice0_parser VIP_Slice1_parser
Description	Ancillary Cropping Configuration for Input Port A		
Type	RW		



Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:16	ANC_USE_NUMLINES	When cropping, the number of lines to keep after the skip_numlines value. Use_numlines + skip_numlines must be smaller than the total number of lines in the srcnum's ancillary data region.	RW	0x0
15:12	RESERVED		R	0x0
11:0	ANC_SKIP_NUMLINES	The number of lines to crop from the top of the vertical ancillary data region.	RW	0x0

Table 7-213. Register Call Summary for Register VIP_ANC_CROP_VERT_PORT_A

VIP Functional Description
• Ancillary and Active Video Cropping: [0][1]
VIP Register Manual
• VIP Parser Register Summary: [4]

Table 7-214. VIP_CROP_HORZ_PORT_A

Address Offset	0x0000 00C0		
Physical Address	0x4897 55C0 0x4897 5AC0	**Instance**	VIP_Slice0_parser VIP_Slice1_parser
Description	Active Video Cropping Configuration for Input Port A		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ACT_TARGET_SRCNUM				ACT_USE_NUMPIX												ACT_BYPASS_N	RESERVED							ACT_SKIP_NUMPIX							

Bits	Field Name	Description	Type	Reset
31:28	ACT_TARGET_SRCNUM	The cropping module can work on only one srcnum. specified in this field. for each dss_vip_parser output port (Active video).	RW	0x0
27:16	ACT_USE_NUMPIX	When cropping, the number of pixels to keep after the skip_numpix value. skip_numpix + use_numpix must be smaller than the original line width.	RW	0x0
15	ACT_BYPASS_N	0 = Bypass cropping module 1 = Cropping module enabled	RW	0x0
14:12	RESERVED		R	0x0
11:0	ACT_SKIP_NUMPIX	The number of pixels to crop from the beginning of each line.	RW	0x0

Table 7-215. Register Call Summary for Register VIP_CROP_HORZ_PORT_A

VIP Functional Description
• Ancillary and Active Video Cropping: [0][1][2][3]
VIP Register Manual
• VIP Parser Register Summary: [6]

Table 7-216. VIP_CROP_VERT_PORT_A

Address Offset	0x0000 00C4		
Physical Address	0x4897 55C4 0x4897 5AC4	**Instance**	VIP_Slice0_parser VIP_Slice1_parser
Description	Active Video Cropping Configuration for Input Port A		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				ACT_USE_NUMLINES												RESERVED				ACT_SKIP_NUMLINES											

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:16	ACT_USE_NUMLINES	When cropping.. the number of lines to keep after the skip_numlines value. Use_numlines + skip_numlines must be smaller than the total number of lines in the srcnum's active video region.	RW	0x0
15:12	RESERVED		R	0x0
11:0	ACT_SKIP_NUMLINES	The number of lines to crop from the top of the vertical active video region.	RW	0x0

Table 7-217. Register Call Summary for Register VIP_CROP_VERT_PORT_A

VIP Functional Description • Ancillary and Active Video Cropping: [0][1]
VIP Register Manual • VIP Parser Register Summary: [4]

Table 7-218. VIP_ANC_VIP_CROP_HORZ_PORT_B

Address Offset	0x0000 00C8		
Physical Address	0x4897 55C8 0x4897 5AC8	**Instance**	VIP_Slice0_parser VIP_Slice1_parser
Description	Ancillary Cropping Configuration for Input Port B		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ANC_TARGET_SRCNUM				ANC_USE_NUMPIX												ANC_BYPASS_N	RESERVED			ANC_SKIP_NUMPIX											

Bits	Field Name	Description	Type	Reset
31:28	ANC_TARGET_SRCNUM	The cropping module can work on only one srcnum. specified in this field. for each dss_vip_parser output port (Ancillary data).	RW	0x0
27:16	ANC_USE_NUMPIX	When cropping, the number of pixels to keep after the skip_numpix value. skip_numpix + use_numpix must be smaller than the original line width.	RW	0x0
15	ANC_BYPASS_N	0 = Bypass cropping module 1 = Cropping module enabled	RW	0x0
14:12	RESERVED		R	0x0
11:0	ANC_SKIP_NUMPIX	The number of pixels to crop from the beginning of each line.	RW	0x0

Table 7-219. Register Call Summary for Register VIP_ANC_VIP_CROP_HORZ_PORT_B

VIP Register Manual • VIP Parser Register Summary: [2]

Table 7-220. VIP_ANC_VIP_CROP_VERT_PORT_B

Address Offset	0x0000 00CC		
Physical Address	0x4897 55CC 0x4897 5ACC	**Instance**	VIP_Slice0_parser VIP_Slice1_parser
Description	Ancillary Cropping Configuration for Input Port B		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								ANC_USE_NUMLINES								RESERVED								ANC_SKIP_NUMLINES							

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:16	ANC_USE_NUMLINES	When cropping, the number of lines to keep after the skip_numlines value. Use_numlines + skip_numlines must be smaller than the total number of lines in the srcnums active video region.	RW	0x0
15:12	RESERVED		R	0x0
11:0	ANC_SKIP_NUMLINES	The number of lines to crop from the top of the vertical ancillary data region.	RW	0x0

Table 7-221. Register Call Summary for Register VIP_ANC_VIP_CROP_VERT_PORT_B

VIP Register Manual
- VIP Parser Register Summary: [2]

Table 7-222. VIP_CROP_HORZ_PORT_B

Address Offset	0x0000 00D0		
Physical Address	0x4897 55D0 0x4897 5AD0	**Instance**	VIP_Slice0_parser VIP_Slice1_parser
Description	Active Video Cropping Configuration for Input Port B		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ACT_TARGET_SRCNUM								ACT_USE_NUMPIX								ACT_BYPASS_N	RESERVED							ACT_SKIP_NUMPIX							

Bits	Field Name	Description	Type	Reset
31:28	ACT_TARGET_SRCNUM	The cropping module can work on only one srcnum. specified in this field. for each dss_vip_parser output port (Active video).	RW	0x0
27:16	ACT_USE_NUMPIX	When cropping, the number of pixels to keep after the skip_numpix value. skip_numpix + use_numpix must be smaller than the original line width.	RW	0x0
15	ACT_BYPASS_N	0 = Bypass cropping module 1 = Cropping module enabled	RW	0x0
14:12	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
11:0	ACT_SKIP_NUMPIX	The number of pixels to crop from the beginning of each line.	RW	0x0

Table 7-223. Register Call Summary for Register VIP_CROP_HORZ_PORT_B

VIP Register Manual
- VIP Parser Register Summary: [2]

Table 7-224. VIP_CROP_VERT_PORT_B

Address Offset	0x0000 00D4		
Physical Address	0x4897 55D4 0x4897 5AD4	**Instance**	VIP_Slice0_parser VIP_Slice1_parser
Description	Active Video Cropping Configuration for Input Port B		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								ACT_USE_NUMLINES								RESERVED								ACT_SKIP_NUMLINES							

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:16	ACT_USE_NUMLINES	When cropping, the number of lines to keep after the skip_numlines value. Use_numlines + skip_numlines must be smaller than the total number of lines in the srcnum's active video region.	RW	0x0
15:12	RESERVED		R	0x0
11:0	ACT_SKIP_NUMLINES	The number of lines to crop from the top of the vertical active video region.	RW	0x0

Table 7-225. Register Call Summary for Register VIP_CROP_VERT_PORT_B

VIP Register Manual
- VIP Parser Register Summary: [2]

Table 7-226. VIP_XTRA6_PORT_A

Address Offset	0x0000 00D8		
Physical Address	0x4897 55D8 0x4897 5AD8	**Instance**	VIP_Slice0_parser VIP_Slice1_parser
Description	Cfg Disable Active Srcnum Vector Input for Port A		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
YUV_SRCNUM_STOP_IMMEDIATELY																ANC_SRCNUM_STOP_IMMEDIATELY															

Bits	Field Name	Description	Type	Reset
31:16	YUV_SRCNUM_STOP_IMMEDIATELY	For the Active Video Port to the VPDMA, logic exists to ensure that a complete frame is sent out to the VPDMA following cfg_enable transitioning inactive for that port. Each bit in this vector represents a srcnum (remapped srcnum for TI line mux mode) going to the VPDMA. For example, bit 0 is srcnum 0, bit 1 is srcnum 1, etc. A ?0? in a bit position means that the hardware will wait for that srcnum, if it is in the middle of a frame, to continue until the end of the frame before stopping. A ?1? in a bit position means that it is ok for a srcnum to stop in the middle of a frame. For example, suppose a source is removed and the input will never complete sending a frame. If the bit position representing that srcnum is set to ?0?, the port will never disable.	RW	0x0
15:0	ANC_SRCNUM_STOP_IMMEDIATELY	For the Ancillary Data Port to the VPDMA, logic exists to ensure that a complete frame is sent out to the VPDMA following cfg_enable transitioning inactive for that port. Each bit in this vector represents a srcnum (remapped srcnum for TI line mux mode) going to the VPDMA. For example, bit 0 is srcnum 0, bit 1 is srcnum 1, etc. A ?0? in a bit position means that the hardware will wait for that srcnum, if it is in the middle of a frame, to continue until the end of the frame before stopping. A ?1? in a bit position means that it is ok for a srcnum to stop in the middle of a frame. For example, suppose a source is removed and the input will never complete sending a frame. If the bit position representing that srcnum is set to ?0?, the port will never disable.	RW	0x0

Table 7-227. Register Call Summary for Register VIP_XTRA6_PORT_A

VIP Functional Description
- VIP Overflow Detection and Recovery: [0][1][2][3]

VIP Register Manual
- VIP Parser Register Summary: [6]

Table 7-228. VIP_XTRA7_PORT_B

Address Offset	0x0000 00DC		
Physical Address	0x4897 55DC 0x4897 5ADC	**Instance**	VIP_Slice0_parser VIP_Slice1_parser
Description	Cfg Disable Active Srcnum Vector Input for Port B		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
YUV_SRCNUM_STOP_IMMEDIATELY																ANC_SRCNUM_STOP_IMMEDIATELY															

Bits	Field Name	Description	Type	Reset
31:16	YUV_SRCNUM_STOP_IMMEDIATELY	For the Active Video Port to the VPDMA, logic exists to ensure that a complete frame is sent out to the VPDMA following cfg_enable transitioning inactive for that port. Each bit in this vector represents a srcnum (remapped srcnum for TI line mux mode) going to the VPDMA. For example, bit 0 is srcnum 0, bit 1 is srcnum 1, etc. A ?0? in a bit position means that the hardware will wait for that srcnum, if it is in the middle of a frame, to continue until the end of the frame before stopping. A ?1? in a bit position means that it is ok for a srcnum to stop in the middle of a frame. For example, suppose a source is removed and the input will never complete sending a frame. If the bit position representing that srcnum is set to ?0?, the port will never disable.	RW	0x0

Bits	Field Name	Description	Type	Reset
15:0	ANC_SRCNUM_STOP_IMMEDIATELY	For the Ancillary Data Port to the VPDMA, logic exists to ensure that a complete frame is sent out to the VPDMA following cfg_enable transitioning inactive for that port. Each bit in this vector represents a srcnum (remapped srcnum for TI line mux mode) going to the VPDMA. For example, bit 0 is srcnum 0, bit 1 is srcnum 1, etc. A ?0? in a bit position means that the hardware will wait for that srcnum, if it is in the middle of a frame, to continue until the end of the frame before stopping. A ?1? in a bit position means that it is ok for a srcnum to stop in the middle of a frame. For example, suppose a source is removed and the input will never complete sending a frame. If the bit position representing that srcnum is set to ?0?, the port will never disable.	RW	0x0

Table 7-229. Register Call Summary for Register VIP_XTRA7_PORT_B

VIP Register Manual
- VIP Parser Register Summary: [2]

Table 7-230. VIP_XTRA8_PORT_A

Address Offset	0x0000 00E0		
Physical Address	0x4897 55E0 0x4897 5AE0	**Instance**	VIP_Slice0_parser VIP_Slice1_parser
Description	Reserved Register for Port A		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															

Bits	Field Name	Description	Type	Reset
31:0	RESERVED	ust be 0x0 at all times.	RW	0x0

Table 7-231. Register Call Summary for Register VIP_XTRA8_PORT_A

VIP Register Manual
- VIP Parser Register Summary: [2]

Table 7-232. VIP_XTRA9_PORT_B

Address Offset	0x0000 00E4		
Physical Address	0x4897 55E4 0x4897 5AE4	**Instance**	VIP_Slice0_parser VIP_Slice1_parser
Description	Reserved Register for Port B		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															

Bits	Field Name	Description	Type	Reset
31:0	RESERVED	ust be 0x0 at all times.	RW	0x0

Table 7-233. Register Call Summary for Register VIP_XTRA9_PORT_B

VIP Register Manual
- VIP Parser Register Summary: [2]

7.5.4 VIP CSC Registers

7.5.4.1 VIP CSC Register Summary

Table 7-234. VIP CSC Registers Mapping Summary 1

Register Name	Type	Register Width (Bits)	Address Offset	VIP_Slice0_csc Base Address	VIP_Slice1_csc Base Address
VIP_CSC00	RW	32	0x0000 0000	0x4897 5700	0x4897 5C00
VIP_CSC01	RW	32	0x0000 0004	0x4897 5704	0x4897 5C04
VIP_CSC02	RW	32	0x0000 0008	0x4897 5708	0x4897 5C08
VIP_CSC03	RW	32	0x0000 000C	0x4897 570C	0x4897 5C0C
VIP_CSC04	RW	32	0x0000 0010	0x4897 5710	0x4897 5C10
VIP_CSC05	RW	32	0x0000 0014	0x4897 5714	0x4897 5C14

7.5.4.2 VIP CSC Register Description

Table 7-235. VIP_CSC00

Address Offset	0x0000 0000		
Physical Address	0x4897 5700 0x4897 5C00	**Instance**	VIP_Slice0_csc VIP_Slice1_csc
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								B0								RESERVED								A0							

Bits	Field Name	Description	Type	Reset
31:29	RESERVED		RW	0x0
28:16	B0	Coefficients of color space converter. This coefficient is a real number in the range of 4. The MSB is sign bit. (Same format conversion as A0)	RW	0x0
15:13	RESERVED		RW	0x0
12:0	A0	Coefficients of color space converter. This coefficient is a real number in the range of 4. The MSB is sign bit. Rules for converting a real number coefficient to a 12-bit hex number for this register: - If the real number is positive, then simply multiply it by 1024, and convert the integer part to hex format. For example, 0.673 X 1024 = 689.152, then 0x2B1 should fill in to this register - If the real number is negative, then multiply it by 1024, and convert it to 2's compliment format in 12-bit. For example, if a coefficient is -1.893, by *1024 to this number, it becomes -1938. The 2'S compliment format of -1938 is 0x186E (in 13-bit width). Then 0x186E should be the number assigned to this register.	RW	0x0

Table 7-236. Register Call Summary for Register VIP_CSC00

VIP Functional Description
• CSC Functional Description: [0][1][2][3][4][5][6][7][8][9]
VIP Register Manual
• VIP CSC Register Summary: [11]

Table 7-237. VIP_CSC01

Address Offset	0x0000 0004		
Physical Address	0x4897 5704 0x4897 5C04	**Instance**	VIP_Slice0_csc VIP_Slice1_csc
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								A1								RESERVED								C0							

Bits	Field Name	Description	Type	Reset
31:29	RESERVED		RW	0x0
28:16	A1	Coefficients of color space converter. This coefficient is a real number in the range of 4. The MSB is sign bit. (Same format conversion as A0)	RW	0x0
15:13	RESERVED		RW	0x0
12:0	C0	Coefficients of color space converter. This coefficient is a real number in the range of 4. The MSB is sign bit. (Same format conversion as A0)	RW	0x0

Table 7-238. Register Call Summary for Register VIP_CSC01

VIP Functional Description
• CSC Functional Description: [0][1][2][3][4][5][6][7][8][9]
VIP Register Manual
• VIP CSC Register Summary: [11]

Table 7-239. VIP_CSC02

Address Offset	0x0000 0008		
Physical Address	0x4897 5708 0x4897 5C08	**Instance**	VIP_Slice0_csc VIP_Slice1_csc
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								C1								RESERVED								B1							

Bits	Field Name	Description	Type	Reset
31:29	RESERVED		RW	0x0
28:16	C1	Coefficients of color space converter. This coefficient is a real number in the range of 4. The MSB is sign bit. (Same format conversion as A0)	RW	0x0
15:13	RESERVED		RW	0x0
12:0	B1	Coefficients of color space converter. This coefficient is a real number in the range of 4. The MSB is sign bit. (Same format conversion as A0)	RW	0x0

Table 7-240. Register Call Summary for Register VIP_CSC02

VIP Functional Description
• CSC Functional Description: [0][1][2][3][4][5][6][7][8][9]
VIP Register Manual
• VIP CSC Register Summary: [11]

Table 7-241. VIP_CSC03

Address Offset	0x0000 000C		
Physical Address	0x4897 570C 0x4897 5C0C	**Instance**	VIP_Slice0_csc VIP_Slice1_csc
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								B2								RESERVED								A2							

Bits	Field Name	Description	Type	Reset
31:29	RESERVED		RW	0x0
28:16	B2	Coefficients of color space converter. This coefficient is a real number in the range of 4. The MSB is sign bit. (Same format conversion as A0)	RW	0x0
15:13	RESERVED		RW	0x0
12:0	A2	Coefficients of color space converter. This coefficient is a real number in the range of 4. The MSB is sign bit. (Same format conversion as A0)	RW	0x0

Table 7-242. Register Call Summary for Register VIP_CSC03

VIP Functional Description
• CSC Functional Description: [0][1][2][3][4][5][6][7][8][9]
VIP Register Manual
• VIP CSC Register Summary: [11]

Table 7-243. VIP_CSC04

Address Offset	0x0000 0010		
Physical Address	0x4897 5710 0x4897 5C10	**Instance**	VIP_Slice0_csc VIP_Slice1_csc
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								D0								RESERVED								C2							

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		RW	0x0
27:16	D0	Coefficients of color space converter. This coefficient is an integer number in the range of 2048. It is in 12-bit wide 2's compliment format. The MSB is sign bit. For example, if this coefficient is 749, then 0x2ED (hex format) should be assigned to this register. Another example, if this coefficient is -1021, then 0xC03 should be assigned to this register.	RW	0x0
15:13	RESERVED		RW	0x0
12:0	C2	Coefficients of color space converter. This coefficient is a real number in the range of 4. The MSB is sign bit. (Same format conversion as A0)	RW	0x0

Table 7-244. Register Call Summary for Register VIP_CSC04

VIP Functional Description
• CSC Functional Description: [0][1][2][3][4][5][6][7][8][9]
VIP Register Manual
• VIP CSC Register Summary: [11]

Table 7-245. VIP_CSC05

Address Offset	0x0000 0014		
Physical Address	0x4897 5714 0x4897 5C14	**Instance**	VIP_Slice0_csc VIP_Slice1_csc
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED			BYPASS	D2												RESERVED				D1											

Bits	Field Name	Description	Type	Reset
31:29	RESERVED		RW	0x0
28	BYPASS	Full CSC bypass mode	RW	0x0
27:16	D2	Coefficients of color space converter. This coefficient is an integer number in the range of 2048. It is in 12-bit wide 2's compliment format. The MSB is sign bit. (Same format conversion as D0)	RW	0x0
15:12	RESERVED		RW	0x0
11:0	D1	Coefficients of color space converter. This coefficient is an integer number in the range of 2048. It is in 12-bit wide 2's compliment format. The MSB is sign bit. (Same format conversion as D0)	RW	0x0

Table 7-246. Register Call Summary for Register VIP_CSC05

VIP Functional Description • CSC Functional Description: [0][1][2][3][4][5][6][7][8][9] • CSC Bypass Mode: [10]
VIP Register Manual • VIP CSC Register Summary: [12]

7.5.5 VIP SC registers

7.5.5.1 VIP SC Register Summary

Table 7-247. VIP SC Registers Mapping Summary 1

Register Name	Type	Register Width (Bits)	Address Offset	VIP_Slice0_sc Base Address	VIP_Slice1_sc Base Address
VIP_CFG_SC0	RW	32	0x0000 0000	0x4897 5800	0x4897 5D00
VIP_CFG_SC1	RW	32	0x0000 0004	0x4897 5804	0x4897 5D04
VIP_CFG_SC2	RW	32	0x0000 0008	0x4897 5808	0x4897 5D08
VIP_CFG_SC3	RW	32	0x0000 000C	0x4897 580C	0x4897 5D0C
VIP_CFG_SC4	RW	32	0x0000 0010	0x4897 5810	0x4897 5D10
VIP_CFG_SC5	RW	32	0x0000 0014	0x4897 5814	0x4897 5D14
VIP_CFG_SC6	RW	32	0x0000 0018	0x4897 5818	0x4897 5D18
RESERVED	R	32	0x0000 001C	0x4897 581C	0x4897 5D1C
VIP_CFG_SC8	RW	32	0x0000 0020	0x4897 5820	0x4897 5D20
VIP_CFG_SC9	RW	32	0x0000 0024	0x4897 5824	0x4897 5D24
VIP_CFG_SC10	RW	32	0x0000 0028	0x4897 5828	0x4897 5D28
VIP_CFG_SC11	RW	32	0x0000 002C	0x4897 582C	0x4897 5D2C
VIP_CFG_SC12	RW	32	0x0000 0030	0x4897 5830	0x4897 5D30
VIP_CFG_SC13	RW	32	0x0000 0034	0x4897 5834	0x4897 5D34
RESERVED	R	32	0x0000 0038	0x4897 5838	0x4897 5D38
RESERVED	R	32	0x0000 003C	0x4897 583C	0x4897 5D3C
RESERVED	R	32	0x0000 0040	0x4897 5840	0x4897 5D40
RESERVED	R	32	0x0000 0044	0x4897 5844	0x4897 5D44
VIP_CFG_SC18	RW	32	0x0000 0048	0x4897 5848	0x4897 5D48
VIP_CFG_SC19	RW	32	0x0000 004C	0x4897 584C	0x4897 5D4C
VIP_CFG_SC20	RW	32	0x0000 0050	0x4897 5850	0x4897 5D50
VIP_CFG_SC21	RW	32	0x0000 0054	0x4897 5854	0x4897 5D54
VIP_CFG_SC22	RW	32	0x0000 0058	0x4897 5858	0x4897 5D58
RESERVED	R	32	0x0000 005C	0x4897 585C	0x4897 5D5C
VIP_CFG_SC24	RW	32	0x0000 0060	0x4897 5860	0x4897 5D60
VIP_CFG_SC25	RW	32	0x0000 0064	0x4897 5864	0x4897 5D64
RESERVED	R	32	0x0000 0068	0x4897 5868	0x4897 5D68
RESERVED	R	32	0x0000 006C	0x4897 586C	0x4897 5D6C
RESERVED	R	32	0x0000 0070	0x4897 5870	0x4897 5D70
RESERVED	R	32	0x0000 0074	0x4897 5874	0x4897 5D74
RESERVED	R	32	0x0000 0078	0x4897 5878	0x4897 5D78
RESERVED	R	32	0x0000 007C	0x4897 587C	0x4897 5D7C

7.5.5.2 VIP SC Register Description

Table 7-248. VIP_CFG_SC0

Address Offset	0x0000 0000		
Physical Address	0x4897 5800 0x4897 5D00	**Instance**	VIP_Slice0_sc VIP_Slice1_sc
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED															CFG_FID_SELFGEN	CFG_TRIM	CFG_Y_PK_EN	RESERVED			CFG_INTERLACE_I	CFG_HP_BYPASS	CFG_DCM_4X	CFG_DCM_2X	CFG_AUTO_HS	CFG_ENABLE_EV	CFG_USE_RAV	CFG_INVT_FID	CFG_SC_BYPASS	CFG_LINEAR	CFG_INTERLACE_O

Bits	Field Name	Description	Type	Reset
31:17	RESERVED		R	0x0
16	CFG_FID_SELFGEN	FID self generate enable. When input is progressive and this bit is set, the SC generates self-toggling (top/bottom) output FID when performing interlacing.	RW	0x0
15	CFG_TRIM	Trimming enable. When 1, the input image whose size is specified by orgW and orgH registers is trimmed to the size with srcW and srcH from the offset specified by offW and offH. 0: disable trimming 1: enable trimming	RW	0x0
14	CFG_Y_PK_EN	This parameter is used by peaking block. 0: disable luma peaking 1: enable luma peaking	RW	0x0
13:11	RESERVED		R	0x0
10	CFG_INTERLACE_I	This parameter is used by both horizontal and vertical scaling 0: the input video format is progressive 1: the input video format is interlace	RW	0x0
9	CFG_HP_BYPASS	This parameter is used by horizontal scaling. If cfg_auto_hs is 0, horizontal polyphase filter is always enabled. In this case, this register is DON?T CARE. If cfg_auto_hs is 1, 0 : The polyphase scaler is always used regardless of the scaling ratio. 1 : The polyphase scaler is bypassed only when (tar_w == src_w) or (tar_w == src_w/2) or (tar_w == src_w/4)	RW	0x0
8	CFG_DCM_4X	This parameter is used by horizontal scaling. 0: the 4X decimation filter is disabled 1: the 4X decimation filter is enabled Note: (1) Either 2X or 4X can be enabled, but they cannot be enabled simultaneously. (2) This register is only set to 1 when it makes sense to do so. Typically, it is used when (horizontal scale ratio 0.25). (3) This register is DON?T CARE when cfg_auto_hs = 1.	RW	0x0
7	CFG_DCM_2X	This parameter is used by horizontal scaling. 0: the 2X decimation filter is disabled 1: the 2X decimation filter is enabled Note: (1) Either 2X or 4X can be enabled, but they cannot be enabled simultaneously. (2) This register is only set to 1 when it makes sense to do so. Typically, it is used when (0.25 horizontal scale ratio 0.5). (3) This register is DON?T CARE when cfg_auto_hs = 1.	RW	0x0

Bits	Field Name	Description	Type	Reset
6	CFG_AUTO_HS	This parameter is used by horizontal scaling. 0: the cfg_dcm_2x and cfg_dcm_4x bits will enable appropriate decimation filters 1: HW will decide whether up-scaling or down-scaling is required based on horizontal scaling ratio (SR). SR 0.5 : horizontal polyphase filter is enabled, all decimation filters are disabled SR = 0.5 : dcm_2x is enabled, horizontal polyphase filter is enabled or disabled based on cfg_hp_bypass 0.5 SR 0.25 : dcm_2x and horizontal polyphase filter both are enabled SR = 0.25 : dcm_4x is enabled, horizontal polyphase filter is enabled or disabled based on cfg_hp_bypass 0.25 SR 0.125 : dcm_4x and horizontal polyphase filter are both enabled SR = 0.125 : Functionally supported, but not recommended in auto mode for image quality concerns	RW	0x0
5	CFG_ENABLE_EV	This parameter is used by the edge-detection block. 0: The output of edge-detection block will be force to ?0? 1: The calculation results of edge-detection block will be output normally	RW	0x0
4	CFG_USE_RAV	This parameter is used by vertical scaling. 0: Poly-phase filter will be used for the vertical scaling 1: Running average filter will be used for the vertical scaling (down scaling only)	RW	0x0
3	CFG_INVT_FID	This parameter is used by vertical scaling. 0: Progressive input 1: Interlaced input Must be set to 1 when CFG_INTERFACE_I = 1.	RW	0x0
2	CFG_SC_BYPASS	This parameter is a general purpose. 0: Scaling module will engaged 1: Scaling module will be bypassed	RW	0x0
1	CFG_LINEAR	This parameter is used by horizontal scaling. 0: Anamorphic scaling 1: Linear scaling	RW	0x0
0	CFG_INTERLACE_O	This parameter is used by vertical scaling. 0: The output format of SC is progressive 1: The output format of SC is interlace	RW	0x0

Table 7-249. Register Call Summary for Register VIP_CFG_SC0

VIP Functional Description
- SC Functional Description: [0][1][2][3][4][5][6][7][8][9][10][11][12][13]
- SC Code: [14][15][16][17][18][19][20][21][22]

VIP Register Manual
- VIP SC Register Summary: [24]

Table 7-250. VIP_CFG_SC1

Address Offset	0x0000 0004		
Physical Address	0x4897 5804 0x4897 5D04	**Instance**	VIP_Slice0_sc VIP_Slice1_sc
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								CFG_ROW_ACC_INC																							

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:0	CFG_ROW_ACC_INC	This parameter is used by vertical scaling. It defines the increment of the row accumulator in vertical poly-phase filter. It can be calculated by following formula: row_acc_inc = round(2^16 *(src_h)/(tar_h)) In case of interlaced input, srcH is input field height In case of interlaced output, tarH is output field height.	RW	0x0

Table 7-251. Register Call Summary for Register VIP_CFG_SC1

VIP Functional Description • SC Functional Description: [0] • SC Code:
VIP Register Manual • VIP SC Register Summary: [8]

Table 7-252. VIP_CFG_SC2

Address Offset	0x0000 0008		
Physical Address	0x4897 5808 0x4897 5D08	**Instance**	VIP_Slice0_sc VIP_Slice1_sc
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								CFG_ROW_ACC_OFFSET																									

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:0	CFG_ROW_ACC_OFFSET	This parameter is used by vertical scaling. It defines the vertical offset during vertical scaling. In progressive mode: this offset will be applied to a frame. In interlace mode: this offset will be applied to the top field.	RW	0x0

Table 7-253. Register Call Summary for Register VIP_CFG_SC2

VIP Functional Description • SC Functional Description: [0] • SC Code:
VIP Register Manual • VIP SC Register Summary: [3]

Table 7-254. VIP_CFG_SC3

Address Offset	0x0000 000C		
Physical Address	0x4897 580C 0x4897 5D0C	**Instance**	VIP_Slice0_sc VIP_Slice1_sc
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								CFG_ROW_ACC_OFFSET_B																									

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:0	CFG_ROW_ACC_OFFSET_B	This parameter is used by vertical scaling. It defines the vertical offset during vertical scaling. In progressive mode: this parameter will not be used. In interlace mode: this offset will be applied to the bottom field.	RW	0x0

Table 7-255. Register Call Summary for Register VIP_CFG_SC3

VIP Functional Description
• SC Functional Description: [0]
VIP Register Manual
• VIP SC Register Summary: [2]

Table 7-256. VIP_CFG_SC4

Address Offset	0x0000 0010		
Physical Address	0x4897 5810 0x4897 5D10	**Instance**	VIP_Slice0_sc VIP_Slice1_sc
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	CFG_NLIN_ACC_INIT_U			RESERVED	CFG_LIN_ACC_INC_U			RESERVED	CFG_TAR_W							RESERVED	CFG_TAR_H														

Bits	Field Name	Description	Type	Reset
31	RESERVED		R	0x0
30:28	CFG_NLIN_ACC_INIT_U	This parameter is used by horizontal scaling. The 3 MSBbits of 'nlin_acc_init' that is defined in CFG_SC10	RW	0x0
27	RESERVED		R	0x0
26:24	CFG_LIN_ACC_INC_U	This parameter is used by horizontal scaling. The 3 MSBbits of 'lin_acc_inc' that is defined in CFG_SC9	RW	0x0
23	RESERVED		R	0x0
22:12	CFG_TAR_W	This parameter is a general purpose. Scaled target picture width. unit is pixel. This parameter defines the final output picture size	RW	0x0
11	RESERVED		R	0x0
10:0	CFG_TAR_H	This parameter is a general purpose. Scaled target picture height.. unit is line... This parameter defines the final output picture size. For the interlace output.. it should be the number of lines per field.	RW	0x0

Table 7-257. Register Call Summary for Register VIP_CFG_SC4

VIP Functional Description
• SC Functional Description: [0][1][2][3][4][5][6]
VIP Register Manual
• VIP SC Register Summary: [8]

Table 7-258. VIP_CFG_SC5

Address Offset	0x0000 0014		
Physical Address	0x4897 5814 0x4897 5D14	**Instance**	VIP_Slice0_sc VIP_Slice1_sc
Description			

Table 7-258. VIP_CFG_SC5 (continued)

Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED					CFG_NLIN_ACC_INC_U			RESERVED				CFG_SRC_W								RESERVED				CFG_SRC_H							

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:24	CFG_NLIN_ACC_INC_U	This parameter is used by horizontal scaling. The 3 MSBbits of ?nlin_acc_inc? that is defined in CFG_SC11	RW	0x0
23	RESERVED		R	0x0
22:12	CFG_SRC_W	This parameter is a general purpose. This parameter defines the width of the source image	RW	0x0
11	RESERVED		R	0x0
10:0	CFG_SRC_H	This parameter is a general purpose. This parameter defines the height of the source image. For the interlace input.. it should be the number of lines per field.	RW	0x0

Table 7-259. Register Call Summary for Register VIP_CFG_SC5

VIP Functional Description
- SC Functional Description: [0][1][2][3][4]

VIP Register Manual
- VIP SC Register Summary: [6]

Table 7-260. VIP_CFG_SC6

Address Offset	0x0000 0018		
Physical Address	0x4897 5818 0x4897 5D18	**Instance**	VIP_Slice0_sc VIP_Slice1_sc
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								CFG_ROW_ACC_INIT_RAV_B																CFG_ROW_ACC_INIT_RAV							

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19:10	CFG_ROW_ACC_INIT_RAV_B	This parameter is used by vertical scaling. it is used only when the input is interlace format. In vertical down scaling.. the running average filter is applied. This parameter sets the initialization value of the row accumulator in running average filter (for bottom field of interlace format)	RW	0x0

Bits	Field Name	Description	Type	Reset
9:0	CFG_ROW_ACC_INIT_RAV	This parameter is used by vertical scaling. In vertical down scaling.. the running average filter is applied. This parameter sets the initialization value of the row accumulator in running average filter (for progressive format or top field of interlace format)	RW	0x0

Table 7-261. Register Call Summary for Register VIP_CFG_SC6

VIP Functional Description
- SC Functional Description: [0][1][2][3]

VIP Register Manual
- VIP SC Register Summary: [5]

Table 7-262. VIP_CFG_SC8

Address Offset	0x0000 0020		
Physical Address	0x4897 5820 0x4897 5D20	**Instance**	VIP_Slice0_sc VIP_Slice1_sc
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								CFG_NLIN_RIGHT								RESERVED	CFG_NLIN_LEFT														

Bits	Field Name	Description	Type	Reset
31:23	RESERVED		R	0x0
22:12	CFG_NLIN_RIGHT	This parameter is used by horizontal scaling. In anamorphic mode. this parameter defines the width of the strip on right-hand side. In other words. it defines the location of the last pixel where the linear scaling is ended. The unit is the 'pixel location' in an active video line. This parameter will not be used in linear scaling	RW	0x0
11	RESERVED		R	0x0
10:0	CFG_NLIN_LEFT	This parameter is used by horizontal scaling. In anamorphic mode. this parameter defines the width of the strip on left-hand side. In other words. it defines the location of the last pixel in the left-sidenonlinear strip. The unit is the 'pixel location' in an active video line. This parameter will not be used in linear scaling	RW	0x0

Table 7-263. Register Call Summary for Register VIP_CFG_SC8

VIP Functional Description
- SC Functional Description: [0][1]

VIP Register Manual
- VIP SC Register Summary: [3]

Table 7-264. VIP_CFG_SC9

Address Offset	0x0000 0024		
Physical Address	0x4897 5824 0x4897 5D24	**Instance**	VIP_Slice0_sc VIP_Slice1_sc
Description			

Table 7-264. VIP_CFG_SC9 (continued)

Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CFG_LIN_ACC_INC																															

Bits	Field Name	Description	Type	Reset
31:0	CFG_LIN_ACC_INC	This parameter is used by horizontal scaling. It defines the increment of the linear accumulator. if SR 0.5 then lin_acc_inc = round(2^24*(srcWi -1) /(tarWi -1)) else if 0.25 SR ? 0.5 lin_acc_inc = round(2^24*(srcWi/2 -1) /(tarWi - 1)) else if SR ? 0.25 lin_acc_inc = round(2^24*(srcWi/4 -1) /(tarWi - 1)) where srcWi and tarWi are the inner source width and the inner target width respectively.	RW	0x0

Table 7-265. Register Call Summary for Register VIP_CFG_SC9

VIP Functional Description
• SC Functional Description: [0][1]
VIP Register Manual
• VIP SC Register Summary: [3]

Table 7-266. VIP_CFG_SC10

Address Offset	0x0000 0028		
Physical Address	0x4897 5828 0x4897 5D28	**Instance**	VIP_Slice0_sc VIP_Slice1_sc
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CFG_NLIN_ACC_INIT																															

Bits	Field Name	Description	Type	Reset
31:0	CFG_NLIN_ACC_INIT	This parameter is used by horizontal scaling. It is used by nonlinear scaling only. It defines the initialization value of the nonlinear accumulator. nlin_acc_init = K*(1-2*d) Here the definitions of K and d are the same as in CFG_SC11	RW	0x0

Table 7-267. Register Call Summary for Register VIP_CFG_SC10

VIP Register Manual
• VIP SC Register Summary: [1]

Table 7-268. VIP_CFG_SC11

Address Offset	0x0000 002C		
Physical Address	0x4897 582C 0x4897 5D2C	**Instance**	VIP_Slice0_sc VIP_Slice1_sc
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CFG_NLIN_ACC_INC																															

Bits	Field Name	Description	Type	Reset
31:0	CFG_NLIN_ACC_INC	This parameter is used by horizontal scaling. It is used by nonlinear scaling only. It defines the increment of the nonlinear accumulator. if upscaling then d = 0 if Ltar !=0 then K =round[2^24*Lsrc/(Ltar*Ltar)] where Lsrc= (srcW-srcWi)/2 else K = 0 elseif downscaling d = (tarW-1)/2 if Ltar!=0 then K = round[2^24 * Lsrc / (Ltar*(Ltar-2d))] where Lsrc= (srcW-srcWi)/(2n) and n=1..2 or 4 else K = 0 nlin_acc_inc = 2*K (negative for downscaling)	RW	0x0

Table 7-269. Register Call Summary for Register VIP_CFG_SC11

VIP Register Manual
- VIP SC Register Summary: [1]

Table 7-270. VIP_CFG_SC12

Address Offset	0x0000 0030		
Physical Address	0x4897 5830 0x4897 5D30	**Instance**	VIP_Slice0_sc VIP_Slice1_sc
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								CFG_COL_ACC_OFFSET																									

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:0	CFG_COL_ACC_OFFSET	This parameter is used in horizontal scaling. It defines the luma accumulator's offset. Normally this parameter can be set as 0 if no horizontal offset is involved. In some applications.. such as Pan and Scan.. a corresponding offset value should be set. The format is 1.24.	RW	0x0

Table 7-271. Register Call Summary for Register VIP_CFG_SC12

VIP Functional Description
- SC Code: [0]

VIP Register Manual
- VIP SC Register Summary: [2]

Table 7-272. VIP_CFG_SC13

Address Offset	0x0000 0034		
Physical Address	0x4897 5834 0x4897 5D34	**Instance**	VIP_Slice0_sc VIP_Slice1_sc
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						CFG_SC_FACTOR_RAV									

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9:0	CFG_SC_FACTOR_RAV	This parameter is used by vertical scaling. Vertical scaling factor: It is defined as following: 1024*tarH/srcH. It is used for downscaling by the running average filter	RW	0x0

Table 7-273. Register Call Summary for Register VIP_CFG_SC13

VIP Functional Description
- SC Functional Description: [0][1][2]
- SC Code: [3][4]

VIP Register Manual
- VIP SC Register Summary: [6]

Table 7-274. VIP_CFG_SC18

Address Offset	0x0000 0048		
Physical Address	0x4897 5848 0x4897 5D48	**Instance**	VIP_Slice0_sc VIP_Slice1_sc
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								CFG_HS_FACTOR							

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9:0	CFG_HS_FACTOR	This parameter is used by horizontal scaling. Horizontal-scaling-factor = tarWi/srcWi. Numerical format: 6.4 (6 bit integer and 4 bit fraction)	RW	0x0

Table 7-275. Register Call Summary for Register VIP_CFG_SC18

VIP Functional Description
- SC Code: [0]

VIP Register Manual
- VIP SC Register Summary: [2]

Table 7-276. VIP_CFG_SC19

Address Offset	0x0000 004C		
Physical Address	0x4897 584C 0x4897 5D4C	**Instance**	VIP_Slice0_sc VIP_Slice1_sc
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CFG_HPF_COEF3								CFG_HPF_COEF2								CFG_HPF_COEF1								CFG_HPF_COEF0							

Bits	Field Name	Description	Type	Reset
31:24	CFG_HPF_COEF3	This parameter is used by the peaking block. Defines the coefficient 3 of the HPF used in the peaking filter. Signed. Decimal point is defined by hpf_norm_shift.	RW	0x0
23:16	CFG_HPF_COEF2	This parameter is used by the peaking block. Defines the coefficient 2 of the HPF used in the peaking filter. Signed. Decimal point is defined by hpf_norm_shift.	RW	0x0
15:8	CFG_HPF_COEF1	This parameter is used by the peaking block. Defines the coefficient 1 of the HPF used in the peaking filter. Signed. Decimal point is defined by hpf_norm_shift.	RW	0x0
7:0	CFG_HPF_COEF0	This parameter is used by the peaking block. Defines the coefficient 0 of the HPF used in the peaking filter. Signed. Decimal point is defined by hpf_norm_shift.	RW	0x0

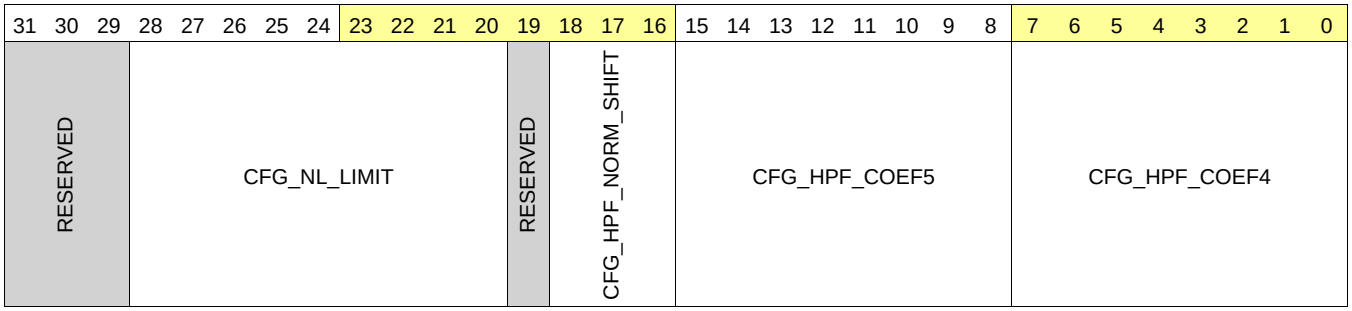
Table 7-277. Register Call Summary for Register VIP_CFG_SC19

VIP Functional Description
- SC Functional Description: [0][1]
- SC Code: [2]

VIP Register Manual
- VIP SC Register Summary: [4]

Table 7-278. VIP_CFG_SC20

Address Offset	0x0000 0050		
Physical Address	0x4897 5850 0x4897 5D50	**Instance**	VIP_Slice0_sc VIP_Slice1_sc
Description			
Type	RW		



Bits	Field Name	Description	Type	Reset
31:29	RESERVED		R	0x0
28:20	CFG_NL_LIMIT	This parameter is used by the peaking block. The maximum of clipping.	RW	0x0
19	RESERVED		R	0x0
18:16	CFG_HPF_NORM_SHIFT	This parameter is used by the peaking block. Defines the decimal point of the hpf coefficient.	RW	0x0
15:8	CFG_HPF_COEF5	This parameter is used by the peaking block. Defines the coefficient 5 of the HPF used in the peaking filter. Signed. Decimal point is defined by hpf_norm_shift.	RW	0x0
7:0	CFG_HPF_COEF4	This parameter is used by the peaking block. Defines the coefficient 4 of the HPF used in the peaking filter. Signed. Decimal point is defined by hpf_norm_shift.	RW	0x0

Table 7-279. Register Call Summary for Register VIP_CFG_SC20

VIP Functional Description • SC Functional Description: [0][1][2] • SC Code: [3]
VIP Register Manual • VIP SC Register Summary: [5]

Table 7-280. VIP_CFG_SC21

Address Offset	0x0000 0054		
Physical Address	0x4897 5854 0x4897 5D54	**Instance**	VIP_Slice0_sc VIP_Slice1_sc
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								CFG_NL_LO_SLOPE								RESERVED								CFG_NL_LO_THR							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	CFG_NL_LO_SLOPE	This parameter is used by the peaking block. Slope of the nonlinear peaking function. The format is fixed point 4.4.	RW	0x0
15:9	RESERVED		R	0x0
8:0	CFG_NL_LO_THR	This parameter is used by the peaking block. Threshold for the nonlinear peaking function. Must be 0	RW	0x0

Table 7-281. Register Call Summary for Register VIP_CFG_SC21

VIP Functional Description • SC Functional Description: [0][1] • SC Code: [2]
VIP Register Manual • VIP SC Register Summary: [4]

Table 7-282. VIP_CFG_SC22

Address Offset	0x0000 0058		
Physical Address	0x4897 5858 0x4897 5D58	**Instance**	VIP_Slice0_sc VIP_Slice1_sc
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														CFG_NL_HI_SLOPE_SHIFT		RESERVED								CFG_NL_HI_THR							

Bits	Field Name	Description	Type	Reset
31:19	RESERVED		R	0x0
18:16	CFG_NL_HI_SLOPE_SHIFT	This parameter is used by the peaking block. Slope of the nonlinear peaking function. The gain is 2^(nl_hi_slope_shift-3).	RW	0x0
15:9	RESERVED		R	0x0
8:0	CFG_NL_HI_THR	This parameter is used by the peaking block. Threshold for the nonlinear peaking function. Must be nl_hi_thr.	RW	0x0

Table 7-283. Register Call Summary for Register VIP_CFG_SC22

VIP Functional Description
- SC Functional Description: [0][1][2]
- SC Code: [3]

VIP Register Manual
- VIP SC Register Summary: [5]

Table 7-284. VIP_CFG_SC24

Address Offset	0x0000 0060		
Physical Address	0x4897 5860 0x4897 5D60	**Instance**	VIP_Slice0_sc VIP_Slice1_sc
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								CFG_ORG_W								RESERVED								CFG_ORG_H							

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:16	CFG_ORG_W	This parameter is used by the trimmer. Horizontal offset from the left of the original input image.	RW	0x0
15:11	RESERVED		R	0x0
10:0	CFG_ORG_H	This parameter is used by the trimmer. Vertical offset from the top of the original input image.	RW	0x0

Table 7-285. Register Call Summary for Register VIP_CFG_SC24

VIP Functional Description
- SC Functional Description: [0][1]

VIP Register Manual
- VIP SC Register Summary: [3]

Table 7-286. VIP_CFG_SC25

Address Offset	0x0000 0064		
Physical Address	0x4897 5864 0x4897 5D64	**Instance**	VIP_Slice0_sc VIP_Slice1_sc
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								CFG_OFF_W								RESERVED								CFG_OFF_H							

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:16	CFG_OFF_W	This parameter is used by the trimmer. Horizontal offset from the left of the original input image.	RW	0x0
15:11	RESERVED		R	0x0
10:0	CFG_OFF_H	This parameter is used by the trimmer. Vertical offset from the top of the original input image.	RW	0x0

Table 7-287. Register Call Summary for Register VIP_CFG_SC25

VIP Functional Description • SC Functional Description: [0][1]
VIP Register Manual • VIP SC Register Summary: [3]

7.5.6 VIP VPDMA Registers

NOTE: The functionality of the following sets of registers is not supported by VIP VPDMA in this family of devices:

- All VIP_INT2_* registers
- All VIP_INT3_* registers

The following channels are not used by VIP VPDMA in this family of devices. All register bit-fields corresponding to these channels should be kept at their reset value.

- HQ_*
- GRPX_*
- SCALER_OUT
- SCALER_LUMA
- SCALER_CHROMA
- NF_*
- TRANSCODE1_*
- TRANSCODE2_*
- AUX_IN
- PIP_FRAME
- POST_COMP_WR
- VBI_SD_VENC

The following clients are not used by VIP VPDMA in this family of devices. All register bit-fields corresponding to these clients should be kept at their reset value.

- DEI_HQ_*
- TRANS1_LUMA
- TRANS1_CHROMA
- TRANS2_LUMA
- TRANS2_CHROMA
- HDMI_WRBK
- VBI_SDVENC
- NF_420_UV_OUT
- NF_420_Y_OUT
- NF_420_UV_IN
- NF_420_Y_IN
- NF_422_IN
- GRPX1_ST
- GRPX2_ST
- GRPX3_ST
- GRPX1_DATA
- GRPX2_DATA
- GRPX3_DATA
- PIP_WRBK
- SC_IN_*
- SC_OUT
- COMP_WRBK

7.5.6.1 VIP VPDMA Register Summary

Table 7-288. VIP VPDMA Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	VIP_VPDMA Base Address
VIP_PID	R	32	0x0000 0000	0x4897 D000
VIP_LIST_ADDR	RW	32	0x0000 0004	0x4897 D004
VIP_LIST_ATTR	RW	32	0x0000 0008	0x4897 D008
VIP_LIST_STAT_SYNC	RW	32	0x0000 000C	0x4897 D00C
VIP_BG_RGB	RW	32	0x0000 0018	0x4897 D018
VIP_BG_YUV	RW	32	0x0000 001C	0x4897 D01C
VIP_VPDMA_SETUP	RW	32	0x0000 0030	0x4897 D030
VIP_MAX_SIZE1	RW	32	0x0000 0034	0x4897 D034
VIP_MAX_SIZE2	RW	32	0x0000 0038	0x4897 D038
VIP_MAX_SIZE3	RW	32	0x0000 003C	0x4897 D03C
VIP_INT0_CHANNEL0_INT_STAT	RW	32	0x0000 0040	0x4897 D040
VIP_INT0_CHANNEL0_INT_MASK	RW	32	0x0000 0044	0x4897 D044
VIP_INT0_CHANNEL1_INT_STAT	RW	32	0x0000 0048	0x4897 D048
VIP_INT0_CHANNEL1_INT_MASK	RW	32	0x0000 004C	0x4897 D04C
VIP_INT0_CHANNEL2_INT_STAT	RW	32	0x0000 0050	0x4897 D050
VIP_INT0_CHANNEL2_INT_MASK	RW	32	0x0000 0054	0x4897 D054
VIP_INT0_CHANNEL3_INT_STAT	RW	32	0x0000 0058	0x4897 D058
VIP_INT0_CHANNEL3_INT_MASK	RW	32	0x0000 005C	0x4897 D05C
VIP_INT0_CHANNEL4_INT_STAT	RW	32	0x0000 0060	0x4897 D060
VIP_INT0_CHANNEL4_INT_MASK	RW	32	0x0000 0064	0x4897 D064
VIP_INT0_CHANNEL5_INT_STAT	RW	32	0x0000 0068	0x4897 D068
VIP_INT0_CHANNEL5_INT_MASK	RW	32	0x0000 006C	0x4897 D06C
VIP_INT0_CLIENT0_INT_STAT	RW	32	0x0000 0078	0x4897 D078
VIP_INT0_CLIENT0_INT_MASK	RW	32	0x0000 007C	0x4897 D07C
VIP_INT0_CLIENT1_INT_STAT	RW	32	0x0000 0080	0x4897 D080
VIP_INT0_CLIENT1_INT_MASK	RW	32	0x0000 0084	0x4897 D084
VIP_INT0_LIST0_INT_STAT	RW	32	0x0000 0088	0x4897 D088
VIP_INT0_LIST0_INT_MASK	RW	32	0x0000 008C	0x4897 D08C
VIP_INT1_CHANNEL0_INT_STAT	RW	32	0x0000 0090	0x4897 D090
VIP_INT1_CHANNEL0_INT_MASK	RW	32	0x0000 0094	0x4897 D094
VIP_INT1_CHANNEL1_INT_STAT	RW	32	0x0000 0098	0x4897 D098

Table 7-288. VIP VPDMA Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	VIP_VPDMA Base Address
VIP_INT1_CHANNEL1_INT_MASK	RW	32	0x0000 009C	0x4897 D09C
VIP_INT1_CHANNEL2_INT_STAT	RW	32	0x0000 00A0	0x4897 D0A0
VIP_INT1_CHANNEL2_INT_MASK	RW	32	0x0000 00A4	0x4897 D0A4
VIP_INT1_CHANNEL3_INT_STAT	RW	32	0x0000 00A8	0x4897 D0A8
VIP_INT1_CHANNEL3_INT_MASK	RW	32	0x0000 00AC	0x4897 D0AC
VIP_INT1_CHANNEL4_INT_STAT	RW	32	0x0000 00B0	0x4897 D0B0
VIP_INT1_CHANNEL4_INT_MASK	RW	32	0x0000 00B4	0x4897 D0B4
VIP_INT1_CHANNEL5_INT_STAT	RW	32	0x0000 00B8	0x4897 D0B8
VIP_INT1_CHANNEL5_INT_MASK	RW	32	0x0000 00BC	0x4897 D0BC
VIP_INT1_CLIENT0_INT_STAT	RW	32	0x0000 00C8	0x4897 D0C8
VIP_INT1_CLIENT0_INT_MASK	RW	32	0x0000 00CC	0x4897 D0CC
VIP_INT1_CLIENT1_INT_STAT	RW	32	0x0000 00D0	0x4897 D0D0
VIP_INT1_CLIENT1_INT_MASK	RW	32	0x0000 00D4	0x4897 D0D4
VIP_INT1_LIST0_INT_STAT	RW	32	0x0000 00D8	0x4897 D0D8
VIP_INT1_LIST0_INT_MASK	RW	32	0x0000 00DC	0x4897 D0DC
VIP_INT2_CHANNEL0_INT_MASK	RW	32	0x0000 00E4	0x4897 D0E4
VIP_INT2_CHANNEL1_INT_STAT	RW	32	0x0000 00E8	0x4897 D0E8
VIP_INT2_CHANNEL1_INT_MASK	RW	32	0x0000 00EC	0x4897 D0EC
VIP_INT2_CHANNEL2_INT_STAT	RW	32	0x0000 00F0	0x4897 D0F0
VIP_INT2_CHANNEL2_INT_MASK	RW	32	0x0000 00F4	0x4897 D0F4
VIP_INT2_CHANNEL3_INT_STAT	RW	32	0x0000 00F8	0x4897 D0F8
VIP_INT2_CHANNEL3_INT_MASK	RW	32	0x0000 00FC	0x4897 D0FC
VIP_INT2_CHANNEL4_INT_STAT	RW	32	0x0000 0100	0x4897 D100
VIP_INT2_CHANNEL4_INT_MASK	RW	32	0x0000 0104	0x4897 D104
VIP_INT2_CHANNEL5_INT_STAT	RW	32	0x0000 0108	0x4897 D108
VIP_INT2_CHANNEL5_INT_MASK	RW	32	0x0000 010C	0x4897 D10C
VIP_INT2_CLIENT0_INT_STAT	RW	32	0x0000 0118	0x4897 D118

Table 7-288. VIP VPDMA Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	VIP_VPDMA Base Address
VIP_INT2_CLIENT0_INT_MASK	RW	32	0x0000 011C	0x4897 D11C
VIP_INT2_LIST0_INT_STAT	RW	32	0x0000 0128	0x4897 D128
VIP_INT2_LIST0_INT_MASK	RW	32	0x0000 012C	0x4897 D12C
VIP_INT3_CHANNEL0_INT_STAT	RW	32	0x0000 0130	0x4897 D130
VIP_INT3_CHANNEL0_INT_MASK	RW	32	0x0000 0134	0x4897 D134
VIP_INT3_CHANNEL1_INT_STAT	RW	32	0x0000 0138	0x4897 D138
VIP_INT3_CHANNEL1_INT_MASK	RW	32	0x0000 013C	0x4897 D13C
VIP_INT3_CHANNEL2_INT_STAT	RW	32	0x0000 0140	0x4897 D140
VIP_INT3_CHANNEL2_INT_MASK	RW	32	0x0000 0144	0x4897 D144
VIP_INT3_CHANNEL3_INT_STAT	RW	32	0x0000 0148	0x4897 D148
VIP_INT3_CHANNEL3_INT_MASK	RW	32	0x0000 014C	0x4897 D14C
VIP_INT3_CHANNEL4_INT_STAT	RW	32	0x0000 0150	0x4897 D150
VIP_INT3_CHANNEL4_INT_MASK	RW	32	0x0000 0154	0x4897 D154
VIP_INT3_CHANNEL5_INT_STAT	RW	32	0x0000 0158	0x4897 D158
VIP_INT3_CHANNEL5_INT_MASK	RW	32	0x0000 015C	0x4897 D15C
VIP_INT3_CLIENT0_INT_STAT	RW	32	0x0000 0168	0x4897 D168
VIP_INT3_CLIENT0_INT_MASK	RW	32	0x0000 016C	0x4897 D16C
VIP_INT3_LIST0_INT_STAT	RW	32	0x0000 0178	0x4897 D178
VIP_INT3_LIST0_INT_MASK	RW	32	0x0000 017C	0x4897 D17C
VIP_PERF_MON0	RW	32	0x0000 0200	0x4897 D200
VIP_PERF_MON1	RW	32	0x0000 0204	0x4897 D204
VIP_PERF_MON2	RW	32	0x0000 0208	0x4897 D208
VIP_PERF_MON3	RW	32	0x0000 020C	0x4897 D20C
VIP_PERF_MON4	RW	32	0x0000 0210	0x4897 D210
VIP_PERF_MON5	RW	32	0x0000 0214	0x4897 D214
VIP_PERF_MON6	RW	32	0x0000 0218	0x4897 D218
VIP_PERF_MON7	RW	32	0x0000 021C	0x4897 D21C
VIP_PERF_MON8	RW	32	0x0000 0220	0x4897 D220
VIP_PERF_MON9	RW	32	0x0000 0224	0x4897 D224
VIP_PERF_MON10	RW	32	0x0000 0228	0x4897 D228
VIP_PERF_MON11	RW	32	0x0000 022C	0x4897 D22C
VIP_PERF_MON12	RW	32	0x0000 0230	0x4897 D230
VIP_PERF_MON13	RW	32	0x0000 0234	0x4897 D234
VIP_PERF_MON14	RW	32	0x0000 0238	0x4897 D238

Table 7-288. VIP VPDMA Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	VIP_VPDMA Base Address
VIP_PERF_MON15	RW	32	0x0000 023C	0x4897 D23C
VIP_PERF_MON16	RW	32	0x0000 0240	0x4897 D240
VIP_PERF_MON17	RW	32	0x0000 0244	0x4897 D244
VIP_PERF_MON18	RW	32	0x0000 0248	0x4897 D248
VIP_PERF_MON19	RW	32	0x0000 024C	0x4897 D24C
VIP_PERF_MON20	RW	32	0x0000 0250	0x4897 D250
VIP_PERF_MON21	RW	32	0x0000 0254	0x4897 D254
VIP_PERF_MON22	RW	32	0x0000 0258	0x4897 D258
VIP_PERF_MON23	RW	32	0x0000 025C	0x4897 D25C
VIP_PERF_MON24	RW	32	0x0000 0260	0x4897 D260
VIP_PERF_MON25	RW	32	0x0000 0264	0x4897 D264
VIP_PERF_MON26	RW	32	0x0000 0268	0x4897 D268
VIP_PERF_MON27	RW	32	0x0000 026C	0x4897 D26C
VIP_PERF_MON28	RW	32	0x0000 0270	0x4897 D270
VIP_PERF_MON29	RW	32	0x0000 0274	0x4897 D274
VIP_PERF_MON30	RW	32	0x0000 0278	0x4897 D278
VIP_PERF_MON31	RW	32	0x0000 027C	0x4897 D27C
VIP_PERF_MON32	RW	32	0x0000 0280	0x4897 D280
VIP_PERF_MON33	RW	32	0x0000 0284	0x4897 D284
VIP_PERF_MON34	RW	32	0x0000 0288	0x4897 D288
VIP_PERF_MON35	RW	32	0x0000 028C	0x4897 D28C
VIP_PERF_MON36	RW	32	0x0000 0290	0x4897 D290
VIP_PERF_MON37	RW	32	0x0000 0294	0x4897 D294
VIP_PERF_MON38	RW	32	0x0000 0298	0x4897 D298
VIP_PERF_MON39	RW	32	0x0000 029C	0x4897 D29C
VIP_PERF_MON40	RW	32	0x0000 02A0	0x4897 D2A0
VIP_PERF_MON41	RW	32	0x0000 02A4	0x4897 D2A4
VIP_PERF_MON42	RW	32	0x0000 02A8	0x4897 D2A8
VIP_PERF_MON43	RW	32	0x0000 02AC	0x4897 D2AC
VIP_PERF_MON44	RW	32	0x0000 02B0	0x4897 D2B0
VIP_PERF_MON45	RW	32	0x0000 02B4	0x4897 D2B4
VIP_PERF_MON46	RW	32	0x0000 02B8	0x4897 D2B8
VIP_PERF_MON47	RW	32	0x0000 02BC	0x4897 D2BC
VIP_PERF_MON48	RW	32	0x0000 02C0	0x4897 D2C0
VIP_PERF_MON49	RW	32	0x0000 02C4	0x4897 D2C4
VIP_PERF_MON50	RW	32	0x0000 02C8	0x4897 D2C8
VIP_PERF_MON51	RW	32	0x0000 02CC	0x4897 D2CC
VIP_PERF_MON52	RW	32	0x0000 02D0	0x4897 D2D0
VIP_PERF_MON53	RW	32	0x0000 02D4	0x4897 D2D4
VIP_PERF_MON54	RW	32	0x0000 02D8	0x4897 D2D8
VIP_PERF_MON55	RW	32	0x0000 02DC	0x4897 D2DC
VIP_PERF_MON56	RW	32	0x0000 02E0	0x4897 D2E0
VIP_PERF_MON57	RW	32	0x0000 02E4	0x4897 D2E4
VIP_PERF_MON58	RW	32	0x0000 02E8	0x4897 D2E8
VIP_PERF_MON59	RW	32	0x0000 02EC	0x4897 D2EC
VIP_PERF_MON60	RW	32	0x0000 02F0	0x4897 D2F0
VIP_PERF_MON61	RW	32	0x0000 02F4	0x4897 D2F4

Table 7-288. VIP VPDMA Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	VIP_VPDMA Base Address
VIP0_LO_Y_CSTAT	RW	32	0x0000 0388	0x4897 D388
VIP0_LO_UV_CSTAT	RW	32	0x0000 038C	0x4897 D38C
VIP0_UP_Y_CSTAT	RW	32	0x0000 0390	0x4897 D390
VIP0_UP_UV_CSTAT	RW	32	0x0000 0394	0x4897 D394
VIP1_LO_Y_CSTAT	RW	32	0x0000 0398	0x4897 D398
VIP1_LO_UV_CSTAT	RW	32	0x0000 039C	0x4897 D39C
VIP1_UP_Y_CSTAT	RW	32	0x0000 03A0	0x4897 D3A0
VIP1_UP_UV_CSTAT	RW	32	0x0000 03A4	0x4897 D3A4
VPI_CTL_CSTAT	RW	32	0x0000 03D0	0x4897 D3D0
VIP0_ANC_A_CSTAT	RW	32	0x0000 03E8	0x4897 D3E8
VIP0_ANC_B_CSTAT	RW	32	0x0000 03EC	0x4897 D3EC
VIP1_ANC_A_CSTAT	RW	32	0x0000 03F0	0x4897 D3F0
VIP1_ANC_B_CSTAT	RW	32	0x0000 03F4	0x4897 D3F4

7.5.6.2 VIP VPDMA Register Description

Table 7-289. VIP_PID

Address Offset	0x0000 0000		
Physical Address	0x4897 D000	**Instance**	VIP_VPDMA
Description	PID VIP VPDMA register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PID																															

Bits	Field Name	Description	Type	Reset
31:0	PID	PID of VPDMA module	R	0x0

Table 7-290. Register Call Summary for Register VIP_PID

VIP Register Manual
- VIP VPDMA Register Summary: [2]

Table 7-291. VIP_LIST_ADDR

Address Offset	0x0000 0004		
Physical Address	0x4897 D004	**Instance**	VIP_VPDMA
Description	The location of a new list to begin processing.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0

Bits	Field Name	Description	Type	Reset
31:0	VIP_LIST_ADDR	Location of a new list of descriptors. This register must be written with the VPDMA Configuration Location after reset.	RW	0x0

Table 7-292. Register Call Summary for Register VIP_LIST_ADDR

VIP Functional Description
- VPDMA Introduction:
- VPDMA Basic Definitions: [1][2]
- VPDMA Configuration: [3][4][5]

VIP Register Manual
- VIP VPDMA Register Summary: [8]
- VIP VPDMA Register Description: [9][10][11][12]

Table 7-293. VIP_LIST_ATTR

Address Offset	0x0000 0008		
Physical Address	0x4897 D008	**Instance**	VIP_VPDMA
Description	The attributes of a new list. This register should always be written after VIP_LIST_ADDR.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED					LIST_NUM			RESERVED			STOP	RDY	LIST_TYPE			LIST_SIZE																

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:24	LIST_NUM	The list number that should be assigned to the list located at VIP_LIST_ADDR. If the list is still active this will block all future list writes until the list is available.	RW	0x0
23:21	RESERVED		R	0x0
20	STOP	This bit is written with the LIST_NUMBER field to stop a self-modifying list. When this bit is written a one the list specified by the LIST_NUMBER is sent a stop signal and will finish the current frame of transfers and then free the list resources.	RW	0x0
19	RDY	This bit is low when a new list cannot be written to the VIP_LIST_ADDR register. The reasons this bit would be low are at initial startup if the LIST_MANAGER State Machine image has not completed loading. It also would be low if the last write to the VIP_LIST_ATTR attempted to start a list that is currently active. When this bit is low any writes to the list address register will cause access to not be accepted until this bit has set by the previous list having completed.	R	0x0
18:16	LIST_TYPE	The type of list that has been generated.\\n0: Normal List\\n1: Self-Modifying List\\n2: List Doorbell\\nOthers Reserved for future use	RW	0x0
15:0	LIST_SIZE	Number of 128 bit word in the new list of descriptors. Writes to this register will activate the list in the list stack of the list manager and begin transfer of the list into VPDMA. This size can not be 0.	RW	0x0

Table 7-294. Register Call Summary for Register VIP_LIST_ATTR

VIP Functional Description
- VPDMA Introduction:
- VPDMA Basic Definitions: [2][3]
- VPDMA Configuration: [4][5][6][7]

Table 7-294. Register Call Summary for Register VIP_LIST_ATTR (continued)

VIP Register Manual

- VIP VPDMA Register Summary: [10]
- VIP VPDMA Register Description: [11]

Table 7-295. VIP_LIST_STAT_SYNC

Address Offset	0x0000 000C		
Physical Address	0x4897 D00C	**Instance**	VIP_VPDMA
Description	The register is used for processor to List Manager syncronization and status registers for the list.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								LIST7_BUSY	LIST6_BUSY	LIST5_BUSY	LIST4_BUSY	LIST3_BUSY	LIST2_BUSY	LIST1_BUSY	LIST0_BUSY	RESERVED								SYNC_LISTS7	SYNC_LISTS6	SYNC_LISTS5	SYNC_LISTS4	SYNC_LISTS3	SYNC_LISTS2	SYNC_LISTS1	SYNC_LISTS0

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23	LIST7_BUSY	The list 7 is currently running. Any attempt to load a new list to list 7 will result in the LM_ADDR and LM_ATTR registers to be locked until the list is complete and this value goes to 0.	R	0x0
22	LIST6_BUSY	The list 6 is currently running. Any attempt to load a new list to list 6 will result in the LM_ADDR and LM_ATTR registers to be locked until the list is complete and this value goes to 0.	R	0x0
21	LIST5_BUSY	The list 5 is currently running. Any attempt to load a new list to list 5 will result in the LM_ADDR and LM_ATTR registers to be locked until the list is complete and this value goes to 0.	R	0x0
20	LIST4_BUSY	The list 4 is currently running. Any attempt to load a new list to list 4 will result in the LM_ADDR and LM_ATTR registers to be locked until the list is complete and this value goes to 0.	R	0x0
19	LIST3_BUSY	The list 3 is currently running. Any attempt to load a new list to list 3 will result in the LM_ADDR and LM_ATTR registers to be locked until the list is complete and this value goes to 0.	R	0x0
18	LIST2_BUSY	The list 2 is currently running. Any attempt to load a new list to list 2 will result in the LM_ADDR and LM_ATTR registers to be locked until the list is complete and this value goes to 0.	R	0x0
17	LIST1_BUSY	The list 1 is currently running. Any attempt to load a new list to list 1 will result in the LM_ADDR and LM_ATTR registers to be locked until the list is complete and this value goes to 0.	R	0x0
16	LIST0_BUSY	The list 0 is currently running. Any attempt to load a new list to list 0 will result in the LM_ADDR and LM_ATTR registers to be locked until the list is complete and this value goes to 0.	R	0x0
15:8	RESERVED	Reserved	R	0x0
7	SYNC_LISTS7	Writing a 1 to this field causes a sync event to fire that clears a Control Descriptor in List 7 waiting on it.	RW	0x0
6	SYNC_LISTS6	Writing a 1 to this field causes a sync event to fire that clears a Control Descriptor in List 6 waiting on it.	RW	0x0

Bits	Field Name	Description	Type	Reset
5	SYNC_LISTS5	Writing a 1 to this field causes a sync event to fire that clears a Control Descriptor in List 5 waiting on it.	RW	0x0
4	SYNC_LISTS4	Writing a 1 to this field causes a sync event to fire that clears a Control Descriptor in List 4 waiting on it.	RW	0x0
3	SYNC_LISTS3	Writing a 1 to this field causes a sync event to fire that clears a Control Descriptor in List 3 waiting on it.	RW	0x0
2	SYNC_LISTS2	Writing a 1 to this field causes a sync event to fire that clears a Control Descriptor in List 2 waiting on it.	RW	0x0
1	SYNC_LISTS1	Writing a 1 to this field causes a sync event to fire that clears a Control Descriptor in List 1 waiting on it.	RW	0x0
0	SYNC_LISTS0	Writing a 1 to this field causes a sync event to fire that clears a Control Descriptor in List 0 waiting on it.	RW	0x0

Table 7-296. Register Call Summary for Register VIP_LIST_STAT_SYNC

VIP Register Manual
- VIP VPDMA Register Summary: [2]

Table 7-297. VIP_BG_RGB

Address Offset	0x0000 0018		
Physical Address	0x4897 D018	**Instance**	VIP_VPDMA
Description	The registers used to set the background color for RGB		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RED								GREEN								BLUE								BLEND							

Bits	Field Name	Description	Type	Reset
31:24	RED	The red value to give on an RGB data port for a blank pixel when using virtual video buffering	RW	0x0
23:16	GREEN	The green value to give on an RGB data port for a blank pixel when using virtual video buffering	RW	0x0
15:8	BLUE	The blue value to give on an RGB data port for a blank pixel when using virtual video buffering	RW	0x0
7:0	BLEND	The blend value to give on an RGB data port for a blank pixel when using virtual video buffering	RW	0x0

Table 7-298. Register Call Summary for Register VIP_BG_RGB

VIP Register Manual
- VIP VPDMA Register Summary: [2]

Table 7-299. VIP_BG_YUV

Address Offset	0x0000 001C		
Physical Address	0x4897 D01C	**Instance**	VIP_VPDMA
Description	The registers used to set the background color for YUV		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								Y								CR								CB							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED	Reserved	R	0x0
23:16	Y	The Y value to give on a YUV data port for a blank pixel when using virtual video buffering	RW	0x0
15:8	CR	The Cr value to give on a YUV data port for a blank pixel when using virtual video buffering	RW	0x0
7:0	CB	The Cb value to give on a YUV data port for a blank pixel when using virtual video buffering	RW	0x0

Table 7-300. Register Call Summary for Register VIP_BG_YUV

VIP Register Manual
- VIP VPDMA Register Summary: [2]

Table 7-301. VIP_VPDMA_SETUP

Address Offset	0x0000 0030		
Physical Address	0x4897 D030	**Instance**	VIP_VPDMA
Description	Configures global parameters that are shared by all clients.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															SEC_BASE_CH

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	SEC_BASE_CH	Use Secondary Channels for Mosaic mode	RW	0x0

Table 7-302. Register Call Summary for Register VIP_VPDMA_SETUP

VIP Register Manual
- VIP VPDMA Register Summary: [2]

Table 7-303. VIP_MAX_SIZE1

Address Offset	0x0000 0034		
Physical Address	0x4897 D034	**Instance**	VIP_VPDMA
Description	Configures maximum width and maximum height global parameters that are shared by all clients to allow for configurable max width and max height when setting is 1 in write descriptor.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MAX_WIDTH																MAX_HEIGHT															

Bits	Field Name	Description	Type	Reset
31:16	MAX_WIDTH	The maximum width to use for setting of max_width 1 in a write descriptor. The value is the number of pixels + 1 so if 1024 pixels are required then set the value to 1023.	RW	0x0
15:0	MAX_HEIGHT	The maximum height to use for setting of max_height 1 in a write descriptor. The value is the number of lines + 1 so if 1024 lines are required then set the value to 1023.	RW	0x0

Table 7-304. Register Call Summary for Register VIP_MAX_SIZE1

VIP Functional Description • VPDMA Descriptors: [0][1]
VIP Register Manual • VIP VPDMA Register Summary: [4]

Table 7-305. VIP_MAX_SIZE2

Address Offset	0x0000 0038		
Physical Address	0x4897 D038	**Instance**	VIP_VPDMA
Description	Configures maximum width and maximum height global parameters that are shared by all clients to allow for configurable max width and max height when setting is 2 in write descriptor.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MAX_WIDTH																MAX_HEIGHT															

Bits	Field Name	Description	Type	Reset
31:16	MAX_WIDTH	The maximum width to use for setting of max_width 2 in a write descriptor. The value is the number of pixels + 1 so if 1024 pixels are required then set the value to 1023.	RW	0x0
15:0	MAX_HEIGHT	The maximum height to use for setting of max_height 2 in a write descriptor. The value is the number of lines + 1 so if 1024 lines are required then set the value to 1023.	RW	0x0

Table 7-306. Register Call Summary for Register VIP_MAX_SIZE2

VIP Functional Description • VPDMA Descriptors: [0][1]
VIP Register Manual • VIP VPDMA Register Summary: [4]

Table 7-307. VIP_MAX_SIZE3

Address Offset	0x0000 003C		
Physical Address	0x4897 D03C	**Instance**	VIP_VPDMA
Description	Configures maximum width and maximum height global parameters that are shared by all clients to allow for configurable max width and max height when setting is 3 in write descriptor.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MAX_WIDTH																MAX_HEIGHT															

Bits	Field Name	Description	Type	Reset
31:16	MAX_WIDTH	The maximum width to use for setting of max_width 3 in a write descriptor. The value is the number of pixels + 1 so if 1024 pixels are required then set the value to 1023.	RW	0x0
15:0	MAX_HEIGHT	The maximum height to use for setting of max_height 3 in a write descriptor. The value is the number of lines + 1 so if 1024 lines are required then set the value to 1023.	RW	0x0

Table 7-308. Register Call Summary for Register VIP_MAX_SIZE3

VIP Functional Description
- VPDMA Descriptors: [0][1]

VIP Register Manual
- VIP VPDMA Register Summary: [4]

Table 7-309. VIP_INT0_CHANNEL0_INT_STAT

Address Offset	0x0000 0040		
Physical Address	0x4897 D040	**Instance**	VIP_VPDMA
Description	This register gives the information of the interrupts that have triggered since last cleared by the process that is servicing vpdma_int0.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INT_STAT_GRPX3	INT_STAT_GRPX2	INT_STAT_GRPX1	INT_STAT_SCALER_OUT	RESERVED								INT_STAT_SCALER_CHROMA	INT_STAT_SCALER_LUMA	INT_STAT_HQ_SCALER	RESERVED	INT_STAT_HQ_MV_OUT	RESERVED		INT_STAT_HQ_MV	RESERVED				INT_STAT_HQ_VID3_CHROMA	INT_STAT_HQ_VID3_LUMA	INT_STAT_HQ_VID2_CHROMA	INT_STAT_HQ_VID2_LUMA	INT_STAT_HQ_VID1_CHROMA	INT_STAT_HQ_VID1_LUMA		

Bits	Field Name	Description	Type	Reset
31	INT_STAT_GRPX3	The last read DMA transaction has occurred for channel grpx3 and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client grpx3_data will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
30	INT_STAT_GRPX2	The last read DMA transaction has occurred for channel grpx2 and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client grpx2_data will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
29	INT_STAT_GRPX1	The last read DMA transaction has occurred for channel grpx1 and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client grpx1_data will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
28	INT_STAT_SCALER_OUT	The last write DMA transaction has completed for channel scaler_out. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client sc_out then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value	RW	0x0
27:20	RESERVED	Reserved	R	0x00
19	INT_STAT_SCALER_CHROMA	The last write DMA transaction has completed for channel scaler_chroma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
18	INT_STAT_SCALER_LUMA	The last write DMA transaction has completed for channel scaler_luma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
17	INT_STAT_HQ_SCALER	The last write DMA transaction has completed for channel hq_scaler. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client dei_sc_out then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
16	RESERVED	Reserved	R	0x0
15	INT_STAT_HQ_MV_OUT	The last write DMA transaction has completed for channel hq_mv_out. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client dei_hq_mv_out then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
14:13	RESERVED	Reserved	R	0x0
12	INT_STAT_HQ_MV	The last read DMA transaction has occurred for channel hq_mv and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client dei_hq_mv_in will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
11:6	RESERVED	Reserved	R	0x00
5	INT_STAT_HQ_VID3_CHROMA	The last write DMA transaction has completed for channel hq_vid3_chroma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
4	INT_STAT_HQ_VID3_LUMA	The last write DMA transaction has completed for channel hq_vid3_luma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
3	INT_STAT_HQ_VID2_CHROMA	The last write DMA transaction has completed for channel hq_vid2_chroma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
2	INT_STAT_HQ_VID2_LUMA	The last write DMA transaction has completed for channel hq_vid2_luma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
1	INT_STAT_HQ_VID1_CHROMA	The last write DMA transaction has completed for channel hq_vid1_chroma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
0	INT_STAT_HQ_VID1_LUMA	The last write DMA transaction has completed for channel hq_vid1_luma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Table 7-310. Register Call Summary for Register VIP_INT0_CHANNEL0_INT_STAT

VIP Functional Description
- VPDMA Interrupts: [0]

VIP Register Manual
- VIP VPDMA Register Summary: [3]

Table 7-311. VIP_INT0_CHANNEL0_INT_MASK

Address Offset	0x0000 0044		
Physical Address	0x4897 D044	**Instance**	VIP_VPDMA
Description	The register gives the information of the interrupts that should be masked and not generate an interrupt for vpdma_int0.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INT_MASK_GRPX3	INT_MASK_GRPX2	INT_MASK_GRPX1	INT_MASK_SCALER_OUT	RESERVED								INT_MASK_SCALER_CHROMA	INT_MASK_SCALER_LUMA	INT_MASK_HQ_SCALER	RESERVED	INT_MASK_HQ_MV_OUT	RESERVED		INT_MASK_HQ_MV	RESERVED				INT_MASK_HQ_VID3_CHROMA	INT_MASK_HQ_VID3_LUMA	INT_MASK_HQ_VID2_CHROMA	INT_MASK_HQ_VID2_LUMA	INT_MASK_HQ_VID1_CHROMA	INT_MASK_HQ_VID1_LUMA	

Bits	Field Name	Description	Type	Reset
31	INT_MASK_GRPX3	The interrupt for Graphcis 2 Data should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
30	INT_MASK_GRPX2	The interrupt for Graphics 1 Data should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
29	INT_MASK_GRPX1	The interrupt for Graphics 0 Data should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
28	INT_MASK_SCALER_OUT	The interrupt for Low Cost DEI Scalar Write to Memory should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
27:20	RESERVED	Reserved	R	0x00
19	INT_MASK_SCALER_CHROMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
18	INT_MASK_SCALER_LUMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
17	INT_MASK_HQ_SCALER	The interrupt for High Quality DEI Scaler Write to Memory should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
16	RESERVED	Reserved	R	0x0
15	INT_MASK_HQ_MV_OUT	The interrupt for Low Cost DEI Motion Vector Write should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
14:13	RESERVED	Reserved	R	0x0
12	INT_MASK_HQ_MV	The interrupt for Low Cost DEI Motion Vector should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
11:6	RESERVED	Reserved	R	0x00
5	INT_MASK_HQ_VID3_CHROMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
4	INT_MASK_HQ_VID3_LUMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
3	INT_MASK_HQ_VID2_CHROMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
2	INT_MASK_HQ_VID2_LUMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
1	INT_MASK_HQ_VID1_CHROMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
0	INT_MASK_HQ_VID1_LUMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0

Table 7-312. Register Call Summary for Register VIP_INT0_CHANNEL0_INT_MASK

VIP Functional Description • VPDMA Interrupts: [0]
VIP Register Manual • VIP VPDMA Register Summary: [3]

Table 7-313. VIP_INT0_CHANNEL1_INT_STAT

Address Offset	0x0000 0048		
Physical Address	0x4897 D048	**Instance**	VIP_VPDMA
Description	This register gives the information of the interrupts that have triggered since last cleared by the process that is servicing vpdma_int0.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INT_STAT_VIP1_MULT_PORTB_SRC9	INT_STAT_VIP1_MULT_PORTB_SRC8	INT_STAT_VIP1_MULT_PORTB_SRC7	INT_STAT_VIP1_MULT_PORTB_SRC6	INT_STAT_VIP1_MULT_PORTB_SRC5	INT_STAT_VIP1_MULT_PORTB_SRC4	INT_STAT_VIP1_MULT_PORTB_SRC3	INT_STAT_VIP1_MULT_PORTB_SRC2	INT_STAT_VIP1_MULT_PORTB_SRC1	INT_STAT_VIP1_MULT_PORTB_SRC0	INT_STAT_VIP1_MULT_PORTA_SRC15	INT_STAT_VIP1_MULT_PORTA_SRC14	INT_STAT_VIP1_MULT_PORTA_SRC13	INT_STAT_VIP1_MULT_PORTA_SRC12	INT_STAT_VIP1_MULT_PORTA_SRC11	INT_STAT_VIP1_MULT_PORTA_SRC10	INT_STAT_VIP1_MULT_PORTA_SRC9	INT_STAT_VIP1_MULT_PORTA_SRC8	INT_STAT_VIP1_MULT_PORTA_SRC7	INT_STAT_VIP1_MULT_PORTA_SRC6	INT_STAT_VIP1_MULT_PORTA_SRC5	INT_STAT_VIP1_MULT_PORTA_SRC4	INT_STAT_VIP1_MULT_PORTA_SRC3	INT_STAT_VIP1_MULT_PORTA_SRC2	INT_STAT_VIP1_MULT_PORTA_SRC1	INT_STAT_VIP1_MULT_PORTA_SRC0	RESERVED						

Bits	Field Name	Description	Type	Reset
31	INT_STAT_VIP1_MULT_PORTB_SRC9	The last write DMA transaction has completed for channel vip1_mult_portb_src9. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
30	INT_STAT_VIP1_MULT_PORTB_SRC8	The last write DMA transaction has completed for channel vip1_mult_portb_src8. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
29	INT_STAT_VIP1_MULT_PORTB_SRC7	The last write DMA transaction has completed for channel vip1_mult_portb_src7. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
28	INT_STAT_VIP1_MULT_PORTB_SRC6	The last write DMA transaction has completed for channel vip1_mult_portb_src6. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
27	INT_STAT_VIP1_MULT_PORTB_SRC5	The last write DMA transaction has completed for channel vip1_mult_portb_src5. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
26	INT_STAT_VIP1_MULT_PORTB_SRC4	The last write DMA transaction has completed for channel vip1_mult_portb_src4. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
25	INT_STAT_VIP1_MULT_PORTB_SRC3	The last write DMA transaction has completed for channel vip1_mult_portb_src3. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
24	INT_STAT_VIP1_MULT_PORTB_SRC2	The last write DMA transaction has completed for channel vip1_mult_portb_src2. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
23	INT_STAT_VIP1_MULT_PORTB_SRC1	The last write DMA transaction has completed for channel vip1_mult_portb_src1. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
22	INT_STAT_VIP1_MULT_PORTB_SRC0	The last write DMA transaction has completed for channel vip1_mult_portb_src0. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
21	INT_STAT_VIP1_MULT_PORTA_SRC15	The last write DMA transaction has completed for channel vip1_mult_porta_src15. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
20	INT_STAT_VIP1_MULT_PORTA_SRC14	The last write DMA transaction has completed for channel vip1_mult_porta_src14. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
19	INT_STAT_VIP1_MULT_PORTA_SRC13	The last write DMA transaction has completed for channel vip1_mult_porta_src13. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
18	INT_STAT_VIP1_MULT_PORTA_SRC12	The last write DMA transaction has completed for channel vip1_mult_porta_src12. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
17	INT_STAT_VIP1_MULT_PORTA_SRC11	The last write DMA transaction has completed for channel vip1_mult_porta_src11. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
16	INT_STAT_VIP1_MULT_PORTA_SRC10	The last write DMA transaction has completed for channel vip1_mult_porta_src10. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
15	INT_STAT_VIP1_MULT_PORTA_SRC9	The last write DMA transaction has completed for channel vip1_mult_porta_src9. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
14	INT_STAT_VIP1_MULT_PORTA_SRC8	The last write DMA transaction has completed for channel vip1_mult_porta_src8. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
13	INT_STAT_VIP1_MULT_PORTA_SRC7	The last write DMA transaction has completed for channel vip1_mult_porta_src7. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
12	INT_STAT_VIP1_MULT_PORTA_SRC6	The last write DMA transaction has completed for channel vip1_mult_porta_src6. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
11	INT_STAT_VIP1_MULT_PORTA_SRC5	The last write DMA transaction has completed for channel vip1_mult_porta_src5. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
10	INT_STAT_VIP1_MULT_PORTA_SRC4	The last write DMA transaction has completed for channel vip1_mult_porta_src4. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
9	INT_STAT_VIP1_MULT_PORTA_SRC3	The last write DMA transaction has completed for channel vip1_mult_porta_src3. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
8	INT_STAT_VIP1_MULT_PORTA_SRC2	The last write DMA transaction has completed for channel vip1_mult_porta_src2. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
7	INT_STAT_VIP1_MULT_PORTA_SRC1	The last write DMA transaction has completed for channel vip1_mult_porta_src1. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
6	INT_STAT_VIP1_MULT_PORTA_SRC0	The last write DMA transaction has completed for channel vip1_mult_porta_src0. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
5:0	RESERVED	Reserved	R	0x00

Table 7-314. Register Call Summary for Register VIP_INT0_CHANNEL1_INT_STAT

VIP Functional Description
- VPDMA Interrupts: [0]

VIP Register Manual
- VIP VPDMA Register Summary: [3]

Table 7-315. VIP_INT0_CHANNEL1_INT_MASK

Address Offset	0x0000 004C		
Physical Address	0x4897 D04C	**Instance**	VIP_VPDMA
Description	The register gives the information of the interrupts that should be masked and not generate an interrupt for vpdma_int0.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INT_MASK_VIP1_MULT_PORTB_SRC9	INT_MASK_VIP1_MULT_PORTB_SRC8	INT_MASK_VIP1_MULT_PORTB_SRC7	INT_MASK_VIP1_MULT_PORTB_SRC6	INT_MASK_VIP1_MULT_PORTB_SRC5	INT_MASK_VIP1_MULT_PORTB_SRC4	INT_MASK_VIP1_MULT_PORTB_SRC3	INT_MASK_VIP1_MULT_PORTB_SRC2	INT_MASK_VIP1_MULT_PORTB_SRC1	INT_MASK_VIP1_MULT_PORTB_SRC0	INT_MASK_VIP1_MULT_PORTA_SRC15	INT_MASK_VIP1_MULT_PORTA_SRC14	INT_MASK_VIP1_MULT_PORTA_SRC13	INT_MASK_VIP1_MULT_PORTA_SRC12	INT_MASK_VIP1_MULT_PORTA_SRC11	INT_MASK_VIP1_MULT_PORTA_SRC10	INT_MASK_VIP1_MULT_PORTA_SRC9	INT_MASK_VIP1_MULT_PORTA_SRC8	INT_MASK_VIP1_MULT_PORTA_SRC7	INT_MASK_VIP1_MULT_PORTA_SRC6	INT_MASK_VIP1_MULT_PORTA_SRC5	INT_MASK_VIP1_MULT_PORTA_SRC4	INT_MASK_VIP1_MULT_PORTA_SRC3	INT_MASK_VIP1_MULT_PORTA_SRC2	INT_MASK_VIP1_MULT_PORTA_SRC1	INT_MASK_VIP1_MULT_PORTA_SRC0	RESERVED						

Bits	Field Name	Description	Type	Reset
31	INT_MASK_VIP1_MULT_PORTB_SRC9	The interrupt for Video Input 1 Port B Channel 9 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
30	INT_MASK_VIP1_MULT_PORTB_SRC8	The interrupt for Video Input 1 Port B Channel 8 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
29	INT_MASK_VIP1_MULT_PORTB_SRC7	The interrupt for Video Input 1 Port B Channel 7 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
28	INT_MASK_VIP1_MULT_PORTB_SRC6	The interrupt for Video Input 1 Port B Channel 6 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
27	INT_MASK_VIP1_MULT_PORTB_SRC5	The interrupt for Video Input 1 Port B Channel 5 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
26	INT_MASK_VIP1_MULT_PORTB_SRC4	The interrupt for Video Input 1 Port B Channel 4 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
25	INT_MASK_VIP1_MULT_PORTB_SRC3	The interrupt for Video Input 1 Port B Channel 3 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
24	INT_MASK_VIP1_MULT_PORTB_SRC2	The interrupt for Video Input 1 Port B Channel 2 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
23	INT_MASK_VIP1_MULT_PORTB_SRC1	The interrupt for Video Input 1 Port B Channel 1 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
22	INT_MASK_VIP1_MULT_PORTB_SRC0	The interrupt for Video Input 1 Port B Channel 0 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
21	INT_MASK_VIP1_MULT_PORTA_SRC15	The interrupt for Video Input 1 Port A Channel 15 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
20	INT_MASK_VIP1_MULT_PORTA_SRC14	The interrupt for Video Input 1 Port A Channel 14 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
19	INT_MASK_VIP1_MULT_PORTA_SRC13	The interrupt for Video Input 1 Port A Channel 13 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
18	INT_MASK_VIP1_MULT_PORTA_SRC12	The interrupt for Video Input 1 Port A Channel 12 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
17	INT_MASK_VIP1_MULT_PORTA_SRC11	The interrupt for Video Input 1 Port A Channel 11 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
16	INT_MASK_VIP1_MULT_PORTA_SRC10	The interrupt for Video Input 1 Port A Channel 10 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
15	INT_MASK_VIP1_MULT_PORTA_SRC9	The interrupt for Video Input 1 Port A Channel 9 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
14	INT_MASK_VIP1_MULT_PORTA_SRC8	The interrupt for Video Input 1 Port A Channel 8 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
13	INT_MASK_VIP1_MULT_PORTA_SRC7	The interrupt for Video Input 1 Port A Channel 7 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
12	INT_MASK_VIP1_MULT_PORTA_SRC6	The interrupt for Video Input 1 Port A Channel 6 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0

Bits	Field Name	Description	Type	Reset
11	INT_MASK_VIP1_MULT_PORTA_SRC5	The interrupt for Video Input 1 Port A Channel 5 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
10	INT_MASK_VIP1_MULT_PORTA_SRC4	The interrupt for Video Input 1 Port A Channel 4 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
9	INT_MASK_VIP1_MULT_PORTA_SRC3	The interrupt for Video Input 1 Port A Channel 3 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
8	INT_MASK_VIP1_MULT_PORTA_SRC2	The interrupt for Video Input 1 Port A Channel 2 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
7	INT_MASK_VIP1_MULT_PORTA_SRC1	The interrupt for Video Input 1 Port A Channel 1 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
6	INT_MASK_VIP1_MULT_PORTA_SRC0	The interrupt for Video Input 1 Port A Channel 0 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
5:0	RESERVED	Reserved	R	0x00

Table 7-316. Register Call Summary for Register VIP_INT0_CHANNEL1_INT_MASK

VIP Functional Description
- VPDMA Interrupts: [0]

VIP Register Manual
- VIP VPDMA Register Summary: [3]

Table 7-317. VIP_INT0_CHANNEL2_INT_STAT

Address Offset	0x0000 0050		
Physical Address	0x4897 D050	**Instance**	VIP_VPDMA
Description	This register gives the information of the interrupts that have triggered since last cleared by the process that is servicing vpdma_int0.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INT_STAT_VIP1_MULT_ANCB_SRC9	INT_STAT_VIP1_MULT_ANCB_SRC8	INT_STAT_VIP1_MULT_ANCB_SRC7	INT_STAT_VIP1_MULT_ANCB_SRC6	INT_STAT_VIP1_MULT_ANCB_SRC5	INT_STAT_VIP1_MULT_ANCB_SRC4	INT_STAT_VIP1_MULT_ANCB_SRC3	INT_STAT_VIP1_MULT_ANCB_SRC2	INT_STAT_VIP1_MULT_ANCB_SRC1	INT_STAT_VIP1_MULT_ANCB_SRC0	INT_STAT_VIP1_MULT_ANCA_SRC15	INT_STAT_VIP1_MULT_ANCA_SRC14	INT_STAT_VIP1_MULT_ANCA_SRC13	INT_STAT_VIP1_MULT_ANCA_SRC12	INT_STAT_VIP1_MULT_ANCA_SRC11	INT_STAT_VIP1_MULT_ANCA_SRC10	INT_STAT_VIP1_MULT_ANCA_SRC9	INT_STAT_VIP1_MULT_ANCA_SRC8	INT_STAT_VIP1_MULT_ANCA_SRC7	INT_STAT_VIP1_MULT_ANCA_SRC6	INT_STAT_VIP1_MULT_ANCA_SRC5	INT_STAT_VIP1_MULT_ANCA_SRC4	INT_STAT_VIP1_MULT_ANCA_SRC3	INT_STAT_VIP1_MULT_ANCA_SRC2	INT_STAT_VIP1_MULT_ANCA_SRC1	INT_STAT_VIP1_MULT_ANCA_SRC0	INT_STAT_VIP1_MULT_PORTB_SRC15	INT_STAT_VIP1_MULT_PORTB_SRC14	INT_STAT_VIP1_MULT_PORTB_SRC13	INT_STAT_VIP1_MULT_PORTB_SRC12	INT_STAT_VIP1_MULT_PORTB_SRC11	INT_STAT_VIP1_MULT_PORTB_SRC10

Bits	Field Name	Description	Type	Reset
31	INT_STAT_VIP1_MULT_ANCB_SRC9	The last write DMA transaction has completed for channel vip1_mult_ancb_src9. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
30	INT_STAT_VIP1_MULT_ANCB_SRC8	The last write DMA transaction has completed for channel vip1_mult_ancb_src8. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
29	INT_STAT_VIP1_MULT_ANCB_SRC7	The last write DMA transaction has completed for channel vip1_mult_ancb_src7. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
28	INT_STAT_VIP1_MULT_ANCB_SRC6	The last write DMA transaction has completed for channel vip1_mult_ancb_src6. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
27	INT_STAT_VIP1_MULT_ANCB_SRC5	The last write DMA transaction has completed for channel vip1_mult_ancb_src5. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
26	INT_STAT_VIP1_MULT_ANCB_SRC4	The last write DMA transaction has completed for channel vip1_mult_ancb_src4. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
25	INT_STAT_VIP1_MULT_ANCB_SRC3	The last write DMA transaction has completed for channel vip1_mult_ancb_src3. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
24	INT_STAT_VIP1_MULT_ANCB_SRC2	The last write DMA transaction has completed for channel vip1_mult_ancb_src2. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
23	INT_STAT_VIP1_MULT_ANCB_SRC1	The last write DMA transaction has completed for channel vip1_mult_ancb_src1. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
22	INT_STAT_VIP1_MULT_ANCB_SRC0	The last write DMA transaction has completed for channel vip1_mult_ancb_src0. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
21	INT_STAT_VIP1_MULT_ANCA_SRC15	The last write DMA transaction has completed for channel vip1_mult_anca_src15. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
20	INT_STAT_VIP1_MULT_ANCA_SRC14	The last write DMA transaction has completed for channel vip1_mult_anca_src14. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
19	INT_STAT_VIP1_MULT_ANCA_SRC13	The last write DMA transaction has completed for channel vip1_mult_anca_src13. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
18	INT_STAT_VIP1_MULT_ANCA_SRC12	The last write DMA transaction has completed for channel vip1_mult_anca_src12. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
17	INT_STAT_VIP1_MULT_ANCA_SRC11	The last write DMA transaction has completed for channel vip1_mult_anca_src11. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
16	INT_STAT_VIP1_MULT_ANCA_SRC10	The last write DMA transaction has completed for channel vip1_mult_anca_src10. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
15	INT_STAT_VIP1_MULT_ANCA_SRC9	The last write DMA transaction has completed for channel vip1_mult_anca_src9. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
14	INT_STAT_VIP1_MULT_ANCA_SRC8	The last write DMA transaction has completed for channel vip1_mult_anca_src8. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
13	INT_STAT_VIP1_MULT_ANCA_SRC7	The last write DMA transaction has completed for channel vip1_mult_anca_src7. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
12	INT_STAT_VIP1_MULT_ANCA_SRC6	The last write DMA transaction has completed for channel vip1_mult_anca_src6. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
11	INT_STAT_VIP1_MULT_ANCA_SRC5	The last write DMA transaction has completed for channel vip1_mult_anca_src5. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
10	INT_STAT_VIP1_MULT_ANCA_SRC4	The last write DMA transaction has completed for channel vip1_mult_anca_src4. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
9	INT_STAT_VIP1_MULT_ANCA_SRC3	The last write DMA transaction has completed for channel vip1_mult_anca_src3. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
8	INT_STAT_VIP1_MULT_ANCA_SRC2	The last write DMA transaction has completed for channel vip1_mult_anca_src2. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
7	INT_STAT_VIP1_MULT_ANCA_SRC1	The last write DMA transaction has completed for channel vip1_mult_anca_src1. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
6	INT_STAT_VIP1_MULT_ANCA_SRC0	The last write DMA transaction has completed for channel vip1_mult_anca_src0. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
5	INT_STAT_VIP1_MULT_PORTB_SRC15	The last write DMA transaction has completed for channel vip1_mult_portb_src15. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
4	INT_STAT_VIP1_MULT_PORTB_SRC14	The last write DMA transaction has completed for channel vip1_mult_portb_src14. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
3	INT_STAT_VIP1_MULT_PORTB_SRC13	The last write DMA transaction has completed for channel vip1_mult_portb_src13. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
2	INT_STAT_VIP1_MULT_PORTB_SRC12	The last write DMA transaction has completed for channel vip1_mult_portb_src12. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
1	INT_STAT_VIP1_MULT_PORTB_SRC11	The last write DMA transaction has completed for channel vip1_mult_portb_src11. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
0	INT_STAT_VIP1_MULT_PORTB_SRC10	The last write DMA transaction has completed for channel vip1_mult_portb_src10. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Table 7-318. Register Call Summary for Register VIP_INT0_CHANNEL2_INT_STAT

VIP Functional Description • VPDMA Interrupts: [0]
VIP Register Manual • VIP VPDMA Register Summary: [3]

Table 7-319. VIP_INT0_CHANNEL2_INT_MASK

Address Offset	0x0000 0054		
Physical Address	0x4897 D054	**Instance**	VIP_VPDMA
Description	The register gives the information of the interrupts that should be masked and not generate an interrupt for vpdma_int0.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INT_MASK_VIP1_MULT_ANCB_SRC9	INT_MASK_VIP1_MULT_ANCB_SRC8	INT_MASK_VIP1_MULT_ANCB_SRC7	INT_MASK_VIP1_MULT_ANCB_SRC6	INT_MASK_VIP1_MULT_ANCB_SRC5	INT_MASK_VIP1_MULT_ANCB_SRC4	INT_MASK_VIP1_MULT_ANCB_SRC3	INT_MASK_VIP1_MULT_ANCB_SRC2	INT_MASK_VIP1_MULT_ANCB_SRC1	INT_MASK_VIP1_MULT_ANCB_SRC0	INT_MASK_VIP1_MULT_ANCA_SRC15	INT_MASK_VIP1_MULT_ANCA_SRC14	INT_MASK_VIP1_MULT_ANCA_SRC13	INT_MASK_VIP1_MULT_ANCA_SRC12	INT_MASK_VIP1_MULT_ANCA_SRC11	INT_MASK_VIP1_MULT_ANCA_SRC10	INT_MASK_VIP1_MULT_ANCA_SRC9	INT_MASK_VIP1_MULT_ANCA_SRC8	INT_MASK_VIP1_MULT_ANCA_SRC7	INT_MASK_VIP1_MULT_ANCA_SRC6	INT_MASK_VIP1_MULT_ANCA_SRC5	INT_MASK_VIP1_MULT_ANCA_SRC4	INT_MASK_VIP1_MULT_ANCA_SRC3	INT_MASK_VIP1_MULT_ANCA_SRC2	INT_MASK_VIP1_MULT_ANCA_SRC1	INT_MASK_VIP1_MULT_ANCA_SRC0	INT_MASK_VIP1_MULT_PORTB_SRC15	INT_MASK_VIP1_MULT_PORTB_SRC14	INT_MASK_VIP1_MULT_PORTB_SRC13	INT_MASK_VIP1_MULT_PORTB_SRC12	INT_MASK_VIP1_MULT_PORTB_SRC11	INT_MASK_VIP1_MULT_PORTB_SRC10

Bits	Field Name	Description	Type	Reset
31	INT_MASK_VIP1_MULT_ANCB_SRC9	The interrupt for Video Input 1 Port B Ancillary Data Channel 9 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
30	INT_MASK_VIP1_MULT_ANCB_SRC8	The interrupt for Video Input 1 Port B Ancillary Data Channel 8 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
29	INT_MASK_VIP1_MULT_ANCB_SRC7	The interrupt for Video Input 1 Port B Ancillary Data Channel 7 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
28	INT_MASK_VIP1_MULT_ANCB_SRC6	The interrupt for Video Input 1 Port B Ancillary Data Channel 6 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
27	INT_MASK_VIP1_MULT_ANCB_SRC5	The interrupt for Video Input 1 Port B Ancillary Data Channel 5 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
26	INT_MASK_VIP1_MULT_ANCB_SRC4	The interrupt for Video Input 1 Port B Ancillary Data Channel 4 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
25	INT_MASK_VIP1_MULT_ANCB_SRC3	The interrupt for Video Input 1 Port B Ancillary Data Channel 3 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
24	INT_MASK_VIP1_MULT_ANCB_SRC2	The interrupt for Video Input 1 Port B Ancillary Data Channel 2 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
23	INT_MASK_VIP1_MULT_ANCB_SRC1	The interrupt for Video Input 1 Port B Ancillary Data Channel 1 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
22	INT_MASK_VIP1_MULT_ANCB_SRC0	The interrupt for Video Input 1 Port B Ancillary Data Channel 0 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
21	INT_MASK_VIP1_MULT_ANCA_SRC15	The interrupt for Video Input 1 Port A Ancillary Data Channel 15 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0

Bits	Field Name	Description	Type	Reset
20	INT_MASK_VIP1_MULT_ANCA_SRC14	The interrupt for Video Input 1 Port A Ancillary Data Channel 14 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
19	INT_MASK_VIP1_MULT_ANCA_SRC13	The interrupt for Video Input 1 Port A Ancillary Data Channel 13 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
18	INT_MASK_VIP1_MULT_ANCA_SRC12	The interrupt for Video Input 1 Port A Ancillary Data Channel 12 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
17	INT_MASK_VIP1_MULT_ANCA_SRC11	The interrupt for Video Input 1 Port A Ancillary Data Channel 11 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
16	INT_MASK_VIP1_MULT_ANCA_SRC10	The interrupt for Video Input 1 Port A Ancillary Data Channel 10 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
15	INT_MASK_VIP1_MULT_ANCA_SRC9	The interrupt for Video Input 1 Port A Ancillary Data Channel 9 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
14	INT_MASK_VIP1_MULT_ANCA_SRC8	The interrupt for Video Input 1 Port A Ancillary Data Channel 8 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
13	INT_MASK_VIP1_MULT_ANCA_SRC7	The interrupt for Video Input 1 Port A Ancillary Data Channel 7 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
12	INT_MASK_VIP1_MULT_ANCA_SRC6	The interrupt for Video Input 1 Port A Ancillary Data Channel 6 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
11	INT_MASK_VIP1_MULT_ANCA_SRC5	The interrupt for Video Input 1 Port A Ancillary Data Channel 5 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
10	INT_MASK_VIP1_MULT_ANCA_SRC4	The interrupt for Video Input 1 Port A Ancillary Data Channel 4 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
9	INT_MASK_VIP1_MULT_ANCA_SRC3	The interrupt for Video Input 1 Port A Ancillary Data Channel 3 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
8	INT_MASK_VIP1_MULT_ANCA_SRC2	The interrupt for Video Input 1 Port A Ancillary Data Channel 2 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
7	INT_MASK_VIP1_MULT_ANCA_SRC1	The interrupt for Video Input 1 Port A Ancillary Data Channel 1 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
6	INT_MASK_VIP1_MULT_ANCA_SRC0	The interrupt for Video Input 1 Port A Ancillary Data Channel 0 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
5	INT_MASK_VIP1_MULT_PORTB_SRC15	The interrupt for Video Input 1 Port B Channel 15 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0

Bits	Field Name	Description	Type	Reset
4	INT_MASK_VIP1_MULT_PORTB_SRC14	The interrupt for Video Input 1 Port B Channel 14 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
3	INT_MASK_VIP1_MULT_PORTB_SRC13	The interrupt for Video Input 1 Port B Channel 13 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
2	INT_MASK_VIP1_MULT_PORTB_SRC12	The interrupt for Video Input 1 Port B Channel 12 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
1	INT_MASK_VIP1_MULT_PORTB_SRC11	The interrupt for Video Input 1 Port B Channel 11 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
0	INT_MASK_VIP1_MULT_PORTB_SRC10	The interrupt for Video Input 1 Port B Channel 10 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0

Table 7-320. Register Call Summary for Register VIP_INT0_CHANNEL2_INT_MASK

VIP Functional Description
- VPDMA Interrupts: [0]

VIP Register Manual
- VIP VPDMA Register Summary: [3]

Table 7-321. VIP_INT0_CHANNEL3_INT_STAT

Address Offset	0x0000 0058		
Physical Address	0x4897 D058	**Instance**	VIP_VPDMA
Description	This register gives the information of the interrupts that have triggered since last cleared by the process that is servicing vpdma_int0.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INT_STAT_VIP2_MULT_PORTB_SRC3	INT_STAT_VIP2_MULT_PORTB_SRC2	INT_STAT_VIP2_MULT_PORTB_SRC1	INT_STAT_VIP2_MULT_PORTB_SRC0	INT_STAT_VIP2_MULT_PORTA_SRC15	INT_STAT_VIP2_MULT_PORTA_SRC14	INT_STAT_VIP2_MULT_PORTA_SRC13	INT_STAT_VIP2_MULT_PORTA_SRC12	INT_STAT_VIP2_MULT_PORTA_SRC11	INT_STAT_VIP2_MULT_PORTA_SRC10	INT_STAT_VIP2_MULT_PORTA_SRC9	INT_STAT_VIP2_MULT_PORTA_SRC8	INT_STAT_VIP2_MULT_PORTA_SRC7	INT_STAT_VIP2_MULT_PORTA_SRC6	INT_STAT_VIP2_MULT_PORTA_SRC5	INT_STAT_VIP2_MULT_PORTA_SRC4	INT_STAT_VIP2_MULT_PORTA_SRC3	INT_STAT_VIP2_MULT_PORTA_SRC2	INT_STAT_VIP2_MULT_PORTA_SRC1	INT_STAT_VIP2_MULT_PORTA_SRC0	INT_STAT_VIP1_PORTB_RGB	INT_STAT_VIP1_PORTA_RGB	INT_STAT_VIP1_PORTB_CHROMA	INT_STAT_VIP1_PORTB_LUMA	INT_STAT_VIP1_PORTA_CHROMA	INT_STAT_VIP1_PORTA_LUMA	INT_STAT_VIP1_MULT_ANCB_SRC15	INT_STAT_VIP1_MULT_ANCB_SRC14	INT_STAT_VIP1_MULT_ANCB_SRC13	INT_STAT_VIP1_MULT_ANCB_SRC12	INT_STAT_VIP1_MULT_ANCB_SRC11	INT_STAT_VIP1_MULT_ANCB_SRC10

Bits	Field Name	Description	Type	Reset
31	INT_STAT_VIP2_MULT_PORTB_SRC3	The last write DMA transaction has completed for channel vip2_mult_portb_src3. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
30	INT_STAT_VIP2_MULT_PORTB_SRC2	The last write DMA transaction has completed for channel vip2_mult_portb_src2. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
29	INT_STAT_VIP2_MULT_PORTB_SRC1	The last write DMA transaction has completed for channel vip2_mult_portb_src1. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
28	INT_STAT_VIP2_MULT_PORTB_SRC0	The last write DMA transaction has completed for channel vip2_mult_portb_src0. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
27	INT_STAT_VIP2_MULT_PORTA_SRC15	The last write DMA transaction has completed for channel vip2_mult_porta_src15. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
26	INT_STAT_VIP2_MULT_PORTA_SRC14	The last write DMA transaction has completed for channel vip2_mult_porta_src14. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
25	INT_STAT_VIP2_MULT_PORTA_SRC13	The last write DMA transaction has completed for channel vip2_mult_porta_src13. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
24	INT_STAT_VIP2_MULT_PORTA_SRC12	The last write DMA transaction has completed for channel vip2_mult_porta_src12. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
23	INT_STAT_VIP2_MULT_PORTA_SRC11	The last write DMA transaction has completed for channel vip2_mult_porta_src11. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
22	INT_STAT_VIP2_MULT_PORTA_SRC10	The last write DMA transaction has completed for channel vip2_mult_porta_src10. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
21	INT_STAT_VIP2_MULT_PORTA_SRC9	The last write DMA transaction has completed for channel vip2_mult_porta_src9. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
20	INT_STAT_VIP2_MULT_PORTA_SRC8	The last write DMA transaction has completed for channel vip2_mult_porta_src8. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
19	INT_STAT_VIP2_MULT_PORTA_SRC7	The last write DMA transaction has completed for channel vip2_mult_porta_src7. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
18	INT_STAT_VIP2_MULT_PORTA_SRC6	The last write DMA transaction has completed for channel vip2_mult_porta_src6. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
17	INT_STAT_VIP2_MULT_PORTA_SRC5	The last write DMA transaction has completed for channel vip2_mult_porta_src5. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
16	INT_STAT_VIP2_MULT_PORTA_SRC4	The last write DMA transaction has completed for channel vip2_mult_porta_src4. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
15	INT_STAT_VIP2_MULT_PORTA_SRC3	The last write DMA transaction has completed for channel vip2_mult_porta_src3. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
14	INT_STAT_VIP2_MULT_PORTA_SRC2	The last write DMA transaction has completed for channel vip2_mult_porta_src2. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
13	INT_STAT_VIP2_MULT_PORTA_SRC1	The last write DMA transaction has completed for channel vip2_mult_porta_src1. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
12	INT_STAT_VIP2_MULT_PORTA_SRC0	The last write DMA transaction has completed for channel vip2_mult_porta_src0. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
11	INT_STAT_VIP1_PORTB_RGB	The last write DMA transaction has completed for channel vip1_portb_rgb. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
10	INT_STAT_VIP1_PORTA_RGB	The last write DMA transaction has completed for channel vip1_porta_rgb. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_up_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
9	INT_STAT_VIP1_PORTB_CHROMA	The last write DMA transaction has completed for channel vip1_portb_chroma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
8	INT_STAT_VIP1_PORTB_LUMA	The last write DMA transaction has completed for channel vip1_portb_luma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
7	INT_STAT_VIP1_PORTA_CHROMA	The last write DMA transaction has completed for channel vip1_porta_chroma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
6	INT_STAT_VIP1_PORTA_LUMA	The last write DMA transaction has completed for channel vip1_porta_luma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
5	INT_STAT_VIP1_MULT_ANCB_SRC15	The last write DMA transaction has completed for channel vip1_mult_ancb_src15. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
4	INT_STAT_VIP1_MULT_ANCB_SRC14	The last write DMA transaction has completed for channel vip1_mult_ancb_src14. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
3	INT_STAT_VIP1_MULT_ANCB_SRC13	The last write DMA transaction has completed for channel vip1_mult_ancb_src13. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
2	INT_STAT_VIP1_MULT_ANCB_SRC12	The last write DMA transaction has completed for channel vip1_mult_ancb_src12. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
1	INT_STAT_VIP1_MULT_ANCB_SRC11	The last write DMA transaction has completed for channel vip1_mult_ancb_src11. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
0	INT_STAT_VIP1_MULT_ANCB_SRC10	The last write DMA transaction has completed for channel vip1_mult_ancb_src10. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Table 7-322. Register Call Summary for Register VIP_INT0_CHANNEL3_INT_STAT

VIP Functional Description
- VPDMA Interrupts: [0]

VIP Register Manual
- VIP VPDMA Register Summary: [3]

Table 7-323. VIP_INT0_CHANNEL3_INT_MASK

Address Offset	0x0000 005C		
Physical Address	0x4897 D05C	**Instance**	VIP_VPDMA
Description	The register gives the information of the interrupts that should be masked and not generate an interrupt for vpdma_int0.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INT_MASK_VIP2_MULT_PORTB_SRC3	INT_MASK_VIP2_MULT_PORTB_SRC2	INT_MASK_VIP2_MULT_PORTB_SRC1	INT_MASK_VIP2_MULT_PORTB_SRC0	INT_MASK_VIP2_MULT_PORTA_SRC15	INT_MASK_VIP2_MULT_PORTA_SRC14	INT_MASK_VIP2_MULT_PORTA_SRC13	INT_MASK_VIP2_MULT_PORTA_SRC12	INT_MASK_VIP2_MULT_PORTA_SRC11	INT_MASK_VIP2_MULT_PORTA_SRC10	INT_MASK_VIP2_MULT_PORTA_SRC9	INT_MASK_VIP2_MULT_PORTA_SRC8	INT_MASK_VIP2_MULT_PORTA_SRC7	INT_MASK_VIP2_MULT_PORTA_SRC6	INT_MASK_VIP2_MULT_PORTA_SRC5	INT_MASK_VIP2_MULT_PORTA_SRC4	INT_MASK_VIP2_MULT_PORTA_SRC3	INT_MASK_VIP2_MULT_PORTA_SRC2	INT_MASK_VIP2_MULT_PORTA_SRC1	INT_MASK_VIP2_MULT_PORTA_SRC0	INT_MASK_VIP1_PORTB_RGB	INT_MASK_VIP1_PORTA_RGB	INT_MASK_VIP1_PORTB_CHROMA	INT_MASK_VIP1_PORTB_LUMA	INT_MASK_VIP1_PORTA_CHROMA	INT_MASK_VIP1_PORTA_LUMA	INT_MASK_VIP1_MULT_ANCB_SRC15	INT_MASK_VIP1_MULT_ANCB_SRC14	INT_MASK_VIP1_MULT_ANCB_SRC13	INT_MASK_VIP1_MULT_ANCB_SRC12	INT_MASK_VIP1_MULT_ANCB_SRC11	INT_MASK_VIP1_MULT_ANCB_SRC10

Bits	Field Name	Description	Type	Reset
31	INT_MASK_VIP2_MULT_PORTB_SRC3	The interrupt for Video Input 2 Port B Channel 3 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
30	INT_MASK_VIP2_MULT_PORTB_SRC2	The interrupt for Video Input 2 Port B Channel 2 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
29	INT_MASK_VIP2_MULT_PORTB_SRC1	The interrupt for Video Input 2 Port B Channel 1 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
28	INT_MASK_VIP2_MULT_PORTB_SRC0	The interrupt for Video Input 2 Port B Channel 0 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
27	INT_MASK_VIP2_MULT_PORTA_SRC15	The interrupt for Video Input 2 Port A Channel 15 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
26	INT_MASK_VIP2_MULT_PORTA_SRC14	The interrupt for Video Input 2 Port A Channel 14 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
25	INT_MASK_VIP2_MULT_PORTA_SRC13	The interrupt for Video Input 2 Port A Channel 13 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
24	INT_MASK_VIP2_MULT_PORTA_SRC12	The interrupt for Video Input 2 Port A Channel 12 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
23	INT_MASK_VIP2_MULT_PORTA_SRC11	The interrupt for Video Input 2 Port A Channel 11 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
22	INT_MASK_VIP2_MULT_PORTA_SRC10	The interrupt for Video Input 2 Port A Channel 10 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
21	INT_MASK_VIP2_MULT_PORTA_SRC9	The interrupt for Video Input 2 Port A Channel 9 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
20	INT_MASK_VIP2_MULT_PORTA_SRC8	The interrupt for Video Input 2 Port A Channel 8 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
19	INT_MASK_VIP2_MULT_PORTA_SRC7	The interrupt for Video Input 2 Port A Channel 7 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
18	INT_MASK_VIP2_MULT_PORTA_SRC6	The interrupt for Video Input 2 Port A Channel 6 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
17	INT_MASK_VIP2_MULT_PORTA_SRC5	The interrupt for Video Input 2 Port A Channel 5 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
16	INT_MASK_VIP2_MULT_PORTA_SRC4	The interrupt for Video Input 2 Port A Channel 4 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
15	INT_MASK_VIP2_MULT_PORTA_SRC3	The interrupt for Video Input 2 Port A Channel 3 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
14	INT_MASK_VIP2_MULT_PORTA_SRC2	The interrupt for Video Input 2 Port A Channel 2 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
13	INT_MASK_VIP2_MULT_PORTA_SRC1	The interrupt for Video Input 2 Port A Channel 1 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
12	INT_MASK_VIP2_MULT_PORTA_SRC0	The interrupt for Video Input 2 Port A Channel 0 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0

Bits	Field Name	Description	Type	Reset
11	INT_MASK_VIP1_PORTB_RGB	The interrupt for Video Input 1 Port B RGB Data should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
10	INT_MASK_VIP1_PORTA_RGB	The interrupt for Video Input 1 Port A RGB Data should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
9	INT_MASK_VIP1_PORTB_CHROMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
8	INT_MASK_VIP1_PORTB_LUMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
7	INT_MASK_VIP1_PORTA_CHROMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
6	INT_MASK_VIP1_PORTA_LUMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
5	INT_MASK_VIP1_MULT_ANCB_SRC15	The interrupt for Video Input 1 Port B Ancillary Data Channel 15 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
4	INT_MASK_VIP1_MULT_ANCB_SRC14	The interrupt for Video Input 1 Port B Ancillary Data Channel 14 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
3	INT_MASK_VIP1_MULT_ANCB_SRC13	The interrupt for Video Input 1 Port B Ancillary Data Channel 13 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
2	INT_MASK_VIP1_MULT_ANCB_SRC12	The interrupt for Video Input 1 Port B Ancillary Data Channel 12 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
1	INT_MASK_VIP1_MULT_ANCB_SRC11	The interrupt for Video Input 1 Port B Ancillary Data Channel 11 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
0	INT_MASK_VIP1_MULT_ANCB_SRC10	The interrupt for Video Input 1 Port B Ancillary Data Channel 10 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0

Table 7-324. Register Call Summary for Register VIP_INT0_CHANNEL3_INT_MASK

VIP Functional Description
- VPDMA Interrupts: [0]

VIP Register Manual
- VIP VPDMA Register Summary: [3]

Table 7-325. VIP_INT0_CHANNEL4_INT_STAT

Address Offset	0x0000 0060		
Physical Address	0x4897 D060	**Instance**	VIP_VPDMA
Description	This register gives the information of the interrupts that have triggered since last cleared by the process that is servicing vpdma_int0.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INT_STAT_VIP2_MULT_ANCB_SRC3	INT_STAT_VIP2_MULT_ANCB_SRC2	INT_STAT_VIP2_MULT_ANCB_SRC1	INT_STAT_VIP2_MULT_ANCB_SRC0	INT_STAT_VIP2_MULT_ANCA_SRC15	INT_STAT_VIP2_MULT_ANCA_SRC14	INT_STAT_VIP2_MULT_ANCA_SRC13	INT_STAT_VIP2_MULT_ANCA_SRC12	INT_STAT_VIP2_MULT_ANCA_SRC11	INT_STAT_VIP2_MULT_ANCA_SRC10	INT_STAT_VIP2_MULT_ANCA_SRC9	INT_STAT_VIP2_MULT_ANCA_SRC8	INT_STAT_VIP2_MULT_ANCA_SRC7	INT_STAT_VIP2_MULT_ANCA_SRC6	INT_STAT_VIP2_MULT_ANCA_SRC5	INT_STAT_VIP2_MULT_ANCA_SRC4	INT_STAT_VIP2_MULT_ANCA_SRC3	INT_STAT_VIP2_MULT_ANCA_SRC2	INT_STAT_VIP2_MULT_ANCA_SRC1	INT_STAT_VIP2_MULT_ANCA_SRC0	INT_STAT_VIP2_MULT_PORTB_SRC15	INT_STAT_VIP2_MULT_PORTB_SRC14	INT_STAT_VIP2_MULT_PORTB_SRC13	INT_STAT_VIP2_MULT_PORTB_SRC12	INT_STAT_VIP2_MULT_PORTB_SRC11	INT_STAT_VIP2_MULT_PORTB_SRC10	INT_STAT_VIP2_MULT_PORTB_SRC9	INT_STAT_VIP2_MULT_PORTB_SRC8	INT_STAT_VIP2_MULT_PORTB_SRC7	INT_STAT_VIP2_MULT_PORTB_SRC6	INT_STAT_VIP2_MULT_PORTB_SRC5	INT_STAT_VIP2_MULT_PORTB_SRC4

Bits	Field Name	Description	Type	Reset
31	INT_STAT_VIP2_MULT_ANCB_SRC3	The last write DMA transaction has completed for channel vip2_mult_ancb_src3. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
30	INT_STAT_VIP2_MULT_ANCB_SRC2	The last write DMA transaction has completed for channel vip2_mult_ancb_src2. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
29	INT_STAT_VIP2_MULT_ANCB_SRC1	The last write DMA transaction has completed for channel vip2_mult_ancb_src1. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
28	INT_STAT_VIP2_MULT_ANCB_SRC0	The last write DMA transaction has completed for channel vip2_mult_ancb_src0. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
27	INT_STAT_VIP2_MULT_ANCA_SRC15	The last write DMA transaction has completed for channel vip2_mult_anca_src15. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
26	INT_STAT_VIP2_MULT_ANCA_SRC14	The last write DMA transaction has completed for channel vip2_mult_anca_src14. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
25	INT_STAT_VIP2_MULT_ANCA_SRC13	The last write DMA transaction has completed for channel vip2_mult_anca_src13. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
24	INT_STAT_VIP2_MULT_ANCA_SRC12	The last write DMA transaction has completed for channel vip2_mult_anca_src12. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
23	INT_STAT_VIP2_MULT_ANCA_SRC11	The last write DMA transaction has completed for channel vip2_mult_anca_src11. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
22	INT_STAT_VIP2_MULT_ANCA_SRC10	The last write DMA transaction has completed for channel vip2_mult_anca_src10. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
21	INT_STAT_VIP2_MULT_ANCA_SRC9	The last write DMA transaction has completed for channel vip2_mult_anca_src9. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
20	INT_STAT_VIP2_MULT_ANCA_SRC8	The last write DMA transaction has completed for channel vip2_mult_anca_src8. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
19	INT_STAT_VIP2_MULT_ANCA_SRC7	The last write DMA transaction has completed for channel vip2_mult_anca_src7. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
18	INT_STAT_VIP2_MULT_ANCA_SRC6	The last write DMA transaction has completed for channel vip2_mult_anca_src6. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
17	INT_STAT_VIP2_MULT_ANCA_SRC5	The last write DMA transaction has completed for channel vip2_mult_anca_src5. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
16	INT_STAT_VIP2_MULT_ANCA_SRC4	The last write DMA transaction has completed for channel vip2_mult_anca_src4. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
15	INT_STAT_VIP2_MULT_ANCA_SRC3	The last write DMA transaction has completed for channel vip2_mult_anca_src3. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
14	INT_STAT_VIP2_MULT_ANCA_SRC2	The last write DMA transaction has completed for channel vip2_mult_anca_src2. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
13	INT_STAT_VIP2_MULT_ANCA_SRC1	The last write DMA transaction has completed for channel vip2_mult_anca_src1. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
12	INT_STAT_VIP2_MULT_ANCA_SRC0	The last write DMA transaction has completed for channel vip2_mult_anca_src0. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
11	INT_STAT_VIP2_MULT_PORTB_SRC15	The last write DMA transaction has completed for channel vip2_mult_portb_src15. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
10	INT_STAT_VIP2_MULT_PORTB_SRC14	The last write DMA transaction has completed for channel vip2_mult_portb_src14. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
9	INT_STAT_VIP2_MULT_PORTB_SRC13	The last write DMA transaction has completed for channel vip2_mult_portb_src13. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
8	INT_STAT_VIP2_MULT_PORTB_SRC12	The last write DMA transaction has completed for channel vip2_mult_portb_src12. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
7	INT_STAT_VIP2_MULT_PORTB_SRC11	The last write DMA transaction has completed for channel vip2_mult_portb_src11. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
6	INT_STAT_VIP2_MULT_PORTB_SRC10	The last write DMA transaction has completed for channel vip2_mult_portb_src10. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
5	INT_STAT_VIP2_MULT_PORTB_SRC9	The last write DMA transaction has completed for channel vip2_mult_portb_src9. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
4	INT_STAT_VIP2_MULT_PORTB_SRC8	The last write DMA transaction has completed for channel vip2_mult_portb_src8. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
3	INT_STAT_VIP2_MULT_PORTB_SRC7	The last write DMA transaction has completed for channel vip2_mult_portb_src7. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
2	INT_STAT_VIP2_MULT_PORTB_SRC6	The last write DMA transaction has completed for channel vip2_mult_portb_src6. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
1	INT_STAT_VIP2_MULT_PORTB_SRC5	The last write DMA transaction has completed for channel vip2_mult_portb_src5. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
0	INT_STAT_VIP2_MULT_PORTB_SRC4	The last write DMA transaction has completed for channel vip2_mult_portb_src4. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Table 7-326. Register Call Summary for Register VIP_INT0_CHANNEL4_INT_STAT

VIP Functional Description
- VPDMA Interrupts: [0]

VIP Register Manual
- VIP VPDMA Register Summary: [3]

Table 7-327. VIP_INT0_CHANNEL4_INT_MASK

Address Offset	0x0000 0064		
Physical Address	0x4897 D064	**Instance**	VIP_VPDMA
Description	The register gives the information of the interrupts that should be masked and not generate an interrupt for vpdma_int0.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INT_MASK_VIP2_MULT_ANCB_SRC3	INT_MASK_VIP2_MULT_ANCB_SRC2	INT_MASK_VIP2_MULT_ANCB_SRC1	INT_MASK_VIP2_MULT_ANCB_SRC0	INT_MASK_VIP2_MULT_ANCA_SRC15	INT_MASK_VIP2_MULT_ANCA_SRC14	INT_MASK_VIP2_MULT_ANCA_SRC13	INT_MASK_VIP2_MULT_ANCA_SRC12	INT_MASK_VIP2_MULT_ANCA_SRC11	INT_MASK_VIP2_MULT_ANCA_SRC10	INT_MASK_VIP2_MULT_ANCA_SRC9	INT_MASK_VIP2_MULT_ANCA_SRC8	INT_MASK_VIP2_MULT_ANCA_SRC7	INT_MASK_VIP2_MULT_ANCA_SRC6	INT_MASK_VIP2_MULT_ANCA_SRC5	INT_MASK_VIP2_MULT_ANCA_SRC4	INT_MASK_VIP2_MULT_ANCA_SRC3	INT_MASK_VIP2_MULT_ANCA_SRC2	INT_MASK_VIP2_MULT_ANCA_SRC1	INT_MASK_VIP2_MULT_ANCA_SRC0	INT_MASK_VIP2_MULT_PORTB_SRC15	INT_MASK_VIP2_MULT_PORTB_SRC14	INT_MASK_VIP2_MULT_PORTB_SRC13	INT_MASK_VIP2_MULT_PORTB_SRC12	INT_MASK_VIP2_MULT_PORTB_SRC11	INT_MASK_VIP2_MULT_PORTB_SRC10	INT_MASK_VIP2_MULT_PORTB_SRC9	INT_MASK_VIP2_MULT_PORTB_SRC8	INT_MASK_VIP2_MULT_PORTB_SRC7	INT_MASK_VIP2_MULT_PORTB_SRC6	INT_MASK_VIP2_MULT_PORTB_SRC5	INT_MASK_VIP2_MULT_PORTB_SRC4

Bits	Field Name	Description	Type	Reset
31	INT_MASK_VIP2_MULT_ANCB_SRC3	The interrupt for Video Input 2 Port B Ancillary Data Channel 3 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
30	INT_MASK_VIP2_MULT_ANCB_SRC2	The interrupt for Video Input 2 Port B Ancillary Data Channel 2 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
29	INT_MASK_VIP2_MULT_ANCB_SRC1	The interrupt for Video Input 2 Port B Ancillary Data Channel 1 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
28	INT_MASK_VIP2_MULT_ANCB_SRC0	The interrupt for Video Input 2 Port B Ancillary Data Channel 0 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
27	INT_MASK_VIP2_MULT_ANCA_SRC15	The interrupt for Video Input 2 Port A Ancillary Data Channel 15 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
26	INT_MASK_VIP2_MULT_ANCA_SRC14	The interrupt for Video Input 2 Port A Ancillary Data Channel 14 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
25	INT_MASK_VIP2_MULT_ANCA_SRC13	The interrupt for Video Input 2 Port A Ancillary Data Channel 13 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
24	INT_MASK_VIP2_MULT_ANCA_SRC12	The interrupt for Video Input 2 Port A Ancillary Data Channel 12 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
23	INT_MASK_VIP2_MULT_ANCA_SRC11	The interrupt for Video Input 2 Port A Ancillary Data Channel 11 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
22	INT_MASK_VIP2_MULT_ANCA_SRC10	The interrupt for Video Input 2 Port A Ancillary Data Channel 10 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
21	INT_MASK_VIP2_MULT_ANCA_SRC9	The interrupt for Video Input 2 Port A Ancillary Data Channel 9 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0

Bits	Field Name	Description	Type	Reset
20	INT_MASK_VIP2_MULT_ANCA_SRC8	The interrupt for Video Input 2 Port A Ancillary Data Channel 8 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
19	INT_MASK_VIP2_MULT_ANCA_SRC7	The interrupt for Video Input 2 Port A Ancillary Data Channel 7 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
18	INT_MASK_VIP2_MULT_ANCA_SRC6	The interrupt for Video Input 2 Port A Ancillary Data Channel 6 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
17	INT_MASK_VIP2_MULT_ANCA_SRC5	The interrupt for Video Input 2 Port A Ancillary Data Channel 5 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
16	INT_MASK_VIP2_MULT_ANCA_SRC4	The interrupt for Video Input 2 Port A Ancillary Data Channel 4 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
15	INT_MASK_VIP2_MULT_ANCA_SRC3	The interrupt for Video Input 2 Port A Ancillary Data Channel 3 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
14	INT_MASK_VIP2_MULT_ANCA_SRC2	The interrupt for Video Input 2 Port A Ancillary Data Channel 2 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
13	INT_MASK_VIP2_MULT_ANCA_SRC1	The interrupt for Video Input 2 Port A Ancillary Data Channel 1 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
12	INT_MASK_VIP2_MULT_ANCA_SRC0	The interrupt for Video Input 2 Port A Ancillary Data Channel 0 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
11	INT_MASK_VIP2_MULT_PORTB_SRC15	The interrupt for Video Input 2 Port B Channel 15 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
10	INT_MASK_VIP2_MULT_PORTB_SRC14	The interrupt for Video Input 2 Port B Channel 14 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
9	INT_MASK_VIP2_MULT_PORTB_SRC13	The interrupt for Video Input 2 Port B Channel 13 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
8	INT_MASK_VIP2_MULT_PORTB_SRC12	The interrupt for Video Input 2 Port B Channel 12 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
7	INT_MASK_VIP2_MULT_PORTB_SRC11	The interrupt for Video Input 2 Port B Channel 11 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
6	INT_MASK_VIP2_MULT_PORTB_SRC10	The interrupt for Video Input 2 Port B Channel 10 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
5	INT_MASK_VIP2_MULT_PORTB_SRC9	The interrupt for Video Input 2 Port B Channel 9 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
4	INT_MASK_VIP2_MULT_PORTB_SRC8	The interrupt for Video Input 2 Port B Channel 8 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
3	INT_MASK_VIP2_MULT_PORTB_SRC7	The interrupt for Video Input 2 Port B Channel 7 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0

Bits	Field Name	Description	Type	Reset
2	INT_MASK_VIP2_MULT_PORTB_SRC6	The interrupt for Video Input 2 Port B Channel 6 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
1	INT_MASK_VIP2_MULT_PORTB_SRC5	The interrupt for Video Input 2 Port B Channel 5 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
0	INT_MASK_VIP2_MULT_PORTB_SRC4	The interrupt for Video Input 2 Port B Channel 4 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0

Table 7-328. Register Call Summary for Register VIP_INT0_CHANNEL4_INT_MASK

VIP Functional Description
- VPDMA Interrupts: [0]

VIP Register Manual
- VIP VPDMA Register Summary: [3]

Table 7-329. VIP_INT0_CHANNEL5_INT_STAT

Address Offset	0x0000 0068		
Physical Address	0x4897 D068	**Instance**	VIP_VPDMA
Description	This register gives the information of the interrupts that have triggered since last cleared by the process that is servicing vpdma_int0.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INT_STAT_TRANSCODE2_CHROMA	INT_STAT_TRANSCODE2_LUMA	INT_STAT_TRANSCODE1_CHROMA	INT_STAT_TRANSCODE1_LUMA	INT_STAT_AUX_IN	INT_STAT_PIP_FRAME	INT_STAT_POST_COMP_WR	INT_STAT_VBI_SD_VENC	RESERVED	INT_STAT_NF_LAST_CHROMA	INT_STAT_NF_LAST_LUMA	INT_STAT_NF_WRITE_CHROMA	INT_STAT_NF_WRITE_LUMA	INT_STAT_OTHER	INT_STAT_VIP2_PORTB_RGB	INT_STAT_VIP2_PORTA_RGB	INT_STAT_VIP2_PORTB_CHROMA	INT_STAT_VIP2_PORTB_LUMA	INT_STAT_VIP2_PORTA_CHROMA	INT_STAT_VIP2_PORTA_LUMA	INT_STAT_VIP2_MULT_ANCB_SRC15	INT_STAT_VIP2_MULT_ANCB_SRC14	INT_STAT_VIP2_MULT_ANCB_SRC13	INT_STAT_VIP2_MULT_ANCB_SRC12	INT_STAT_VIP2_MULT_ANCB_SRC11	INT_STAT_VIP2_MULT_ANCB_SRC10	INT_STAT_VIP2_MULT_ANCB_SRC9	INT_STAT_VIP2_MULT_ANCB_SRC8	INT_STAT_VIP2_MULT_ANCB_SRC7	INT_STAT_VIP2_MULT_ANCB_SRC6	INT_STAT_VIP2_MULT_ANCB_SRC5	INT_STAT_VIP2_MULT_ANCB_SRC4

Bits	Field Name	Description	Type	Reset
31	INT_STAT_TRANSCODE2_CHROMA	The last write DMA transaction has completed for channel transcode2_chroma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
30	INT_STAT_TRANSCODE2_LUMA	The last write DMA transaction has completed for channel transcode2_luma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
29	INT_STAT_TRANSCODE1_CHROMA	The last write DMA transaction has completed for channel transcode1_luma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
28	INT_STAT_TRANSCODE1_LUMA	The last write DMA transaction has completed for channel transcode1_luma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
27	INT_STAT_AUX_IN	The last read DMA transaction has occurred for channel aux_in and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client comp_wrbk will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
26	INT_STAT_PIP_FRAME	The last read DMA transaction has occurred for channel pip_frame and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client pip_wrbk will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
25	INT_STAT_POST_COMP_WR	The last write DMA transaction has completed for channel post_comp_wr. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client hdmi_wrbk_out then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
24	INT_STAT_VBI_SD_VENC	The last read DMA transaction has occurred for channel vbi_sd_venc and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client vbi_sdvenc will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
23	RESERVED	Reserved	R	0x0
22	INT_STAT_NF_LAST_CHROMA	The last write DMA transaction has completed for channel nf_last_chroma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
21	INT_STAT_NF_LAST_LUMA	The last write DMA transaction has completed for channel nf_last_luma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
20	INT_STAT_NF_WRITE_CHROMA	The last write DMA transaction has completed for channel nf_write_chroma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
19	INT_STAT_NF_WRITE_LUMA	The last write DMA transaction has completed for channel nf_write_luma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
18	INT_STAT_OTHER	This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
17	INT_STAT_VIP2_PORTB_RGB	The last write DMA transaction has completed for channel vip2_portb_rgb. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
16	INT_STAT_VIP2_PORTA_RGB	The last write DMA transaction has completed for channel vip2_porta_rgb. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_up_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
15	INT_STAT_VIP2_PORTB_CHROMA	The last write DMA transaction has completed for channel vip2_portb_chroma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
14	INT_STAT_VIP2_PORTB_LUMA	The last write DMA transaction has completed for channel vip2_portb_luma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
13	INT_STAT_VIP2_PORTA_CHROMA	The last write DMA transaction has completed for channel vip2_porta_chroma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
12	INT_STAT_VIP2_PORTA_LUMA	The last write DMA transaction has completed for channel vip2_porta_luma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
11	INT_STAT_VIP2_MULT_ANCB_SRC15	The last write DMA transaction has completed for channel vip2_mult_ancb_src15. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
10	INT_STAT_VIP2_MULT_ANCB_SRC14	The last write DMA transaction has completed for channel vip2_mult_ancb_src14. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
9	INT_STAT_VIP2_MULT_ANCB_SRC13	The last write DMA transaction has completed for channel vip2_mult_ancb_src13. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
8	INT_STAT_VIP2_MULT_ANCB_SRC12	The last write DMA transaction has completed for channel vip2_mult_ancb_src12. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
7	INT_STAT_VIP2_MULT_ANCB_SRC11	The last write DMA transaction has completed for channel vip2_mult_ancb_src11. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
6	INT_STAT_VIP2_MULT_ANCB_SRC10	The last write DMA transaction has completed for channel vip2_mult_ancb_src10. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
5	INT_STAT_VIP2_MULT_ANCB_SRC9	The last write DMA transaction has completed for channel vip2_mult_ancb_src9. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
4	INT_STAT_VIP2_MULT_ANCB_SRC8	The last write DMA transaction has completed for channel vip2_mult_ancb_src8. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
3	INT_STAT_VIP2_MULT_ANCB_SRC7	The last write DMA transaction has completed for channel vip2_mult_ancb_src7. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
2	INT_STAT_VIP2_MULT_ANCB_SRC6	The last write DMA transaction has completed for channel vip2_mult_ancb_src6. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
1	INT_STAT_VIP2_MULT_ANCB_SRC5	The last write DMA transaction has completed for channel vip2_mult_ancb_src5. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
0	INT_STAT_VIP2_MULT_ANCB_SRC4	The last write DMA transaction has completed for channel vip2_mult_ancb_src4. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Table 7-330. Register Call Summary for Register VIP_INT0_CHANNEL5_INT_STAT

VIP Functional Description
- VPDMA Interrupts: [0]

VIP Register Manual
- VIP VPDMA Register Summary: [3]

Table 7-331. VIP_INT0_CHANNEL5_INT_MASK

Address Offset	0x0000 006C		
Physical Address	0x4897 D06C	**Instance**	VIP_VPDMA
Description	The register gives the information of the interrupts that should be masked and not generate an interrupt for vpdma_int0.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INT_MASK_TRANSCODE2_CHROMA	INT_MASK_TRANSCODE2_LUMA	INT_MASK_TRANSCODE1_CHROMA	INT_MASK_TRANSCODE1_LUMA	INT_MASK_AUX_IN	INT_MASK_PIP_FRAME	INT_MASK_POST_COMP_WR	INT_MASK_VBI_SD_VENC	RESERVED	INT_MASK_NF_LAST_CHROMA	INT_MASK_NF_LAST_LUMA	INT_MASK_NF_WRITE_CHROMA	INT_MASK_NF_WRITE_LUMA	INT_MASK_OTHER	INT_MASK_VIP2_PORTB_RGB	INT_MASK_VIP2_PORTA_RGB	INT_MASK_VIP2_PORTB_CHROMA	INT_MASK_VIP2_PORTB_LUMA	INT_MASK_VIP2_PORTA_CHROMA	INT_MASK_VIP2_PORTA_LUMA	INT_MASK_VIP2_MULT_ANCB_SRC15	INT_MASK_VIP2_MULT_ANCB_SRC14	INT_MASK_VIP2_MULT_ANCB_SRC13	INT_MASK_VIP2_MULT_ANCB_SRC12	INT_MASK_VIP2_MULT_ANCB_SRC11	INT_MASK_VIP2_MULT_ANCB_SRC10	INT_MASK_VIP2_MULT_ANCB_SRC9	INT_MASK_VIP2_MULT_ANCB_SRC8	INT_MASK_VIP2_MULT_ANCB_SRC7	INT_MASK_VIP2_MULT_ANCB_SRC6	INT_MASK_VIP2_MULT_ANCB_SRC5	INT_MASK_VIP2_MULT_ANCB_SRC4

Bits	Field Name	Description	Type	Reset
31	INT_MASK_TRANSCODE2_CHROMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
30	INT_MASK_TRANSCODE2_LUMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
29	INT_MASK_TRANSCODE1_CHROMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
28	INT_MASK_TRANSCODE1_LUMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
27	INT_MASK_AUX_IN	The interrupt for Auxilary Data for the Compositor Frame From Memory should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0

Bits	Field Name	Description	Type	Reset
26	INT_MASK_PIP_FRAME	The interrupt for PIP Data for the Compositor Frame From Memory should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
25	INT_MASK_POST_COMP_WR	The interrupt for Post Compositer Writeback to Memory should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
24	INT_MASK_VBI_SD_VENC	The interrupt for SD Video Encoder VBI Data should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
23	RESERVED	Reserved	R	0x0
22	INT_MASK_NF_LAST_CHROMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
21	INT_MASK_NF_LAST_LUMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
20	INT_MASK_NF_WRITE_CHROMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
19	INT_MASK_NF_WRITE_LUMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
18	INT_MASK_OTHER	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
17	INT_MASK_VIP2_PORTB_RGB	The interrupt for Video Input 2 Port B RGB Data should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
16	INT_MASK_VIP2_PORTA_RGB	The interrupt for Video Input 2 Port A RGB Data should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
15	INT_MASK_VIP2_PORTB_CHROMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
14	INT_MASK_VIP2_PORTB_LUMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
13	INT_MASK_VIP2_PORTA_CHROMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
12	INT_MASK_VIP2_PORTA_LUMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
11	INT_MASK_VIP2_MULT_ANCB_SRC15	The interrupt for Video Input 2 Port B Ancillary Data Channel 15 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
10	INT_MASK_VIP2_MULT_ANCB_SRC14	The interrupt for Video Input 2 Port B Ancillary Data Channel 14 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
9	INT_MASK_VIP2_MULT_ANCB_SRC13	The interrupt for Video Input 2 Port B Ancillary Data Channel 13 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
8	INT_MASK_VIP2_MULT_ANCB_SRC12	The interrupt for Video Input 2 Port B Ancillary Data Channel 12 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0

Bits	Field Name	Description	Type	Reset
7	INT_MASK_VIP2_MULT_ANCB_SRC11	The interrupt for Video Input 2 Port B Ancillary Data Channel 11 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
6	INT_MASK_VIP2_MULT_ANCB_SRC10	The interrupt for Video Input 2 Port B Ancillary Data Channel 10 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
5	INT_MASK_VIP2_MULT_ANCB_SRC9	The interrupt for Video Input 2 Port B Ancillary Data Channel 9 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
4	INT_MASK_VIP2_MULT_ANCB_SRC8	The interrupt for Video Input 2 Port B Ancillary Data Channel 8 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
3	INT_MASK_VIP2_MULT_ANCB_SRC7	The interrupt for Video Input 2 Port B Ancillary Data Channel 7 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
2	INT_MASK_VIP2_MULT_ANCB_SRC6	The interrupt for Video Input 2 Port B Ancillary Data Channel 6 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
1	INT_MASK_VIP2_MULT_ANCB_SRC5	The interrupt for Video Input 2 Port B Ancillary Data Channel 5 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
0	INT_MASK_VIP2_MULT_ANCB_SRC4	The interrupt for Video Input 2 Port B Ancillary Data Channel 4 should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0

Table 7-332. Register Call Summary for Register VIP_INT0_CHANNEL5_INT_MASK

VIP Functional Description • VPDMA Interrupts: [0]
VIP Register Manual • VIP VPDMA Register Summary: [3]

Table 7-333. VIP_INT0_CLIENT0_INT_STAT

Address Offset	0x0000 0078		
Physical Address	0x4897 D078	**Instance**	VIP_VPDMA
Description	This register gives the information of the interrupts that have triggered since last cleared by the process that is servicing vpdma_int0.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INT_STAT_GRPX1_DATA	INT_STAT_COMP_WRBK	INT_STAT_SC_OUT	RESERVED								INT_STAT_SC_IN_LUMA	INT_STAT_SC_IN_CHROMA	INT_STAT_PIP_WRBK	INT_STAT_DEI_SC_OUT	RESERVED	INT_STAT_DEI_HQ_MV_OUT	RESERVED			INT_STAT_DEI_HQ_MV_IN	RESERVED						INT_STAT_DEI_HQ_3_CHROMA	INT_STAT_DEI_HQ_3_LUMA	INT_STAT_DEI_HQ_2_CHROMA	INT_STAT_DEI_HQ_2_LUMA	INT_STAT_DEI_HQ_1_LUMA	INT_STAT_DEI_HQ_1_CHROMA

Bits	Field Name	Description	Type	Reset
31	INT_STAT_GRPX1_DATA	The client interface grpx1_data has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
30	INT_STAT_COMP_WRBK	The client interface comp_wrbk has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
29	INT_STAT_SC_OUT	The client interface sc_out has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
28:21	RESERVED	Reserved	R	0x00
20	INT_STAT_SC_IN_LUMA	The client interface sc_in_luma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
19	INT_STAT_SC_IN_CHROMA	The client interface sc_in_chroma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
18	INT_STAT_PIP_WRBK	The client interface pip_wrbk has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
17	INT_STAT_DEI_SC_OUT	The client interface dei_sc_out has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
16	RESERVED	Reserved	R	0x0
15	INT_STAT_DEI_HQ_MV_OUT	The client interface dei_hq_mv_out has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
14:13	RESERVED	Reserved	R	0x0
12	INT_STAT_DEI_HQ_MV_IN	The client interface dei_hq_mv_in has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
11:6	RESERVED	Reserved	R	0x00
5	INT_STAT_DEI_HQ_3_CHROMA	The client interface dei_hq_3_chroma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
4	INT_STAT_DEI_HQ_3_LUMA	The client interface dei_hq_3_luma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
3	INT_STAT_DEI_HQ_2_CHROMA	The client interface dei_hq_2_chroma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
2	INT_STAT_DEI_HQ_2_LUMA	The client interface dei_hq_2_luma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
1	INT_STAT_DEI_HQ_1_LUMA	The client interface dei_hq_1_luma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
0	INT_STAT_DEI_HQ_1_CHROMA	The client interface dei_hq_1_chroma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Table 7-334. Register Call Summary for Register VIP_INT0_CLIENT0_INT_STAT

VIP Functional Description
- VPDMA Interrupts: [0]

VIP Register Manual
- VIP VPDMA Register Summary: [3]

Table 7-335. VIP_INT0_CLIENT0_INT_MASK

Address Offset	0x0000 007C		
Physical Address	0x4897 D07C	**Instance**	VIP_VPDMA
Description	The register gives the information of the interrupts that should be masked and not generate an interrupt for vpdma_int0.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INT_MASK_GRPX1_DATA	INT_MASK_COMP_WRBK	INT_MASK_SC_OUT	RESERVED								INT_MASK_SC_IN_LUMA	INT_MASK_SC_IN_CHROMA	INT_MASK_PIP_WRBK	INT_MASK_DEI_SC_OUT	RESERVED	INT_MASK_DEI_HQ_MV_OUT	RESERVED		INT_MASK_DEI_HQ_MV_IN	RESERVED						INT_MASK_DEI_HQ_3_CHROMA	INT_MASK_DEI_HQ_3_LUMA	INT_MASK_DEI_HQ_2_CHROMA	INT_MASK_DEI_HQ_2_LUMA	INT_MASK_DEI_HQ_1_LUMA	INT_MASK_DEI_HQ_1_CHROMA

Bits	Field Name	Description	Type	Reset
31	INT_MASK_GRPX1_DATA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
30	INT_MASK_COMP_WRBK	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
29	INT_MASK_SC_OUT	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
28:21	RESERVED	Reserved	R	0x00
20	INT_MASK_SC_IN_LUMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
19	INT_MASK_SC_IN_CHROMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
18	INT_MASK_PIP_WRBK	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0

Bits	Field Name	Description	Type	Reset
17	INT_MASK_DEI_SC_OUT	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
16	RESERVED	Reserved	R	0x0
15	INT_MASK_DEI_HQ_MV_OUT	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
14:13	RESERVED	Reserved	R	0x0
12	INT_MASK_DEI_HQ_MV_IN	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
11:6	RESERVED	Reserved	R	0x00
5	INT_MASK_DEI_HQ_3_CHROMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
4	INT_MASK_DEI_HQ_3_LUMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
3	INT_MASK_DEI_HQ_2_CHROMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
2	INT_MASK_DEI_HQ_2_LUMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
1	INT_MASK_DEI_HQ_1_LUMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
0	INT_MASK_DEI_HQ_1_CHROMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0

Table 7-336. Register Call Summary for Register VIP_INT0_CLIENT0_INT_MASK

VIP Functional Description
- VPDMA Interrupts: [0]

VIP Register Manual
- VIP VPDMA Register Summary: [3]

Table 7-337. VIP_INT0_CLIENT1_INT_STAT

Address Offset	0x0000 0080		
Physical Address	0x4897 D080	**Instance**	VIP_VPDMA
Description	This register gives the information of the interrupts that have triggered since last cleared by the process that is servicing vpdma_int0.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	RESERVED	INT_STAT_VIP2_ANC_B	INT_STAT_VIP2_ANC_A	INT_STAT_VIP1_ANC_B	INT_STAT_VIP1_ANC_A	INT_STAT_TRANS2_LUMA	INT_STAT_TRANS2_CHROMA	INT_STAT_TRANS1_LUMA	INT_STAT_TRANS1_CHROMA	INT_STAT_HDMI_WRBK_OUT	INT_STAT_VPI_CTL	INT_STAT_VBI_SDVENC	RESERVED	INT_STAT_NF_420_UV_OUT	INT_STAT_NF_420_Y_OUT	INT_STAT_NF_420_UV_IN	INT_STAT_NF_420_Y_IN	INT_STAT_NF_422_IN	INT_STAT_GRPX3_ST	INT_STAT_GRPX2_ST	INT_STAT_GRPX1_ST	INT_STAT_VIP2_UP_UV	INT_STAT_VIP2_UP_Y	INT_STAT_VIP2_LO_UV	INT_STAT_VIP2_LO_Y	INT_STAT_VIP1_UP_UV	INT_STAT_VIP1_UP_Y	INT_STAT_VIP1_LO_UV	INT_STAT_VIP1_LO_Y	INT_STAT_GRPX3_DATA	INT_STAT_GRPX2_DATA

Bits	Field Name	Description	Type	Reset
31	RESERVED	Reserved	R	0
30	RESERVED	Reserved	R	0
29	INT_STAT_VIP2_ANC_B	The client interface vip2_anc_b has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
28	INT_STAT_VIP2_ANC_A	The client interface vip2_anc_a has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
27	INT_STAT_VIP1_ANC_B	The client interface vip1_anc_b has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
26	INT_STAT_VIP1_ANC_A	The client interface vip1_anc_a has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
25	INT_STAT_TRANS2_LUMA	The client interface trans2_luma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
24	INT_STAT_TRANS2_CHROMA	The client interface trans2_chroma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
23	INT_STAT_TRANS1_LUMA	The client interface trans1_luma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
22	INT_STAT_TRANS1_CHROMA	The client interface trans1_chroma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0

Bits	Field Name	Description	Type	Reset
21	INT_STAT_HDMI_WRBK_OUT	The client interface hdmi_wrbk_out has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
20	INT_STAT_VPI_CTL	The client interface vpi_ctl has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
19	INT_STAT_VBI_SDVENC	The client interface vbi_sdvenc has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
18	RESERVED	Reserved	R	0
17	INT_STAT_NF_420_UV_OUT	The client interface nf_420_uv_out has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
16	INT_STAT_NF_420_Y_OUT	The client interface nf_420_y_out has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
15	INT_STAT_NF_420_UV_IN	The client interface nf_420_uv_in has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
14	INT_STAT_NF_420_Y_IN	The client interface nf_420_y_in has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
13	INT_STAT_NF_422_IN	The client interface nf_422_in has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0

Bits	Field Name	Description	Type	Reset
12	INT_STAT_GRPX3_ST	The client interface grpx3_st has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
11	INT_STAT_GRPX2_ST	The client interface grpx2_st has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
10	INT_STAT_GRPX1_ST	The client interface grpx1_st has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
9	INT_STAT_VIP2_UP_UV	The client interface vip2_up_uv has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
8	INT_STAT_VIP2_UP_Y	The client interface vip2_up_y has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
7	INT_STAT_VIP2_LO_UV	The client interface vip2_lo_uv has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
6	INT_STAT_VIP2_LO_Y	The client interface vip2_lo_y has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
5	INT_STAT_VIP1_UP_UV	The client interface vip1_up_uv has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0

Bits	Field Name	Description	Type	Reset
4	INT_STAT_VIP1_UP_Y	The client interface vip1_up_y has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
3	INT_STAT_VIP1_LO_UV	The client interface vip1_lo_uv has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
2	INT_STAT_VIP1_LO_Y	The client interface vip1_lo_y has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
1	INT_STAT_GRPX3_DATA	The client interface grpx3_data has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
0	INT_STAT_GRPX2_DATA	The client interface grpx2_data has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0

Table 7-338. Register Call Summary for Register VIP_INT0_CLIENT1_INT_STAT

VIP Functional Description • VPDMA Interrupts: [0]
VIP Register Manual • VIP VPDMA Register Summary: [3]

Table 7-339. VIP_INT0_CLIENT1_INT_MASK

Address Offset	0x0000 0084		
Physical Address	0x4897 D084	**Instance**	VIP_VPDMA
Description	The register gives the information of the interrupts that should be masked and not generate an interrupt for vpdma_int0.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	RESERVED	INT_MASK_VIP2_ANC_B	INT_MASK_VIP2_ANC_A	INT_MASK_VIP1_ANC_B	INT_MASK_VIP1_ANC_A	INT_MASK_TRANS2_LUMA	INT_MASK_TRANS2_CHROMA	INT_MASK_TRANS1_LUMA	INT_MASK_TRANS1_CHROMA	INT_MASK_HDMI_WRBK_OUT	INT_MASK_VPI_CTL	INT_MASK_VBI_SDVENC	RESERVED	INT_MASK_NF_420_UV_OUT	INT_MASK_NF_420_Y_OUT	INT_MASK_NF_420_UV_IN	INT_MASK_NF_420_Y_IN	INT_MASK_NF_422_IN	INT_MASK_GRPX3_ST	INT_MASK_GRPX2_ST	INT_MASK_GRPX1_ST	INT_MASK_VIP2_UP_UV	INT_MASK_VIP2_UP_Y	INT_MASK_VIP2_LO_UV	INT_MASK_VIP2_LO_Y	INT_MASK_VIP1_UP_UV	INT_MASK_VIP1_UP_Y	INT_MASK_VIP1_LO_UV	INT_MASK_VIP1_LO_Y	INT_MASK_GRPX3_DATA	INT_MASK_GRPX2_DATA

Bits	Field Name	Description	Type	Reset
31	RESERVED	Reserved	R	0
30	RESERVED	Reserved	R	0
29	INT_MASK_VIP2_ANC_B	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
28	INT_MASK_VIP2_ANC_A	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
27	INT_MASK_VIP1_ANC_B	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
26	INT_MASK_VIP1_ANC_A	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
25	INT_MASK_TRANS2_LUMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
24	INT_MASK_TRANS2_CHROMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
23	INT_MASK_TRANS1_LUMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
22	INT_MASK_TRANS1_CHROMA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
21	INT_MASK_HDMI_WRBK_OUT	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
20	INT_MASK_VPI_CTL	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
19	INT_MASK_VBI_SDVENC	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
18	RESERVED	Reserved	R	0
17	INT_MASK_NF_420_UV_OUT	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
16	INT_MASK_NF_420_Y_OUT	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
15	INT_MASK_NF_420_UV_IN	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0

Bits	Field Name	Description	Type	Reset
14	INT_MASK_NF_420_Y_IN	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
13	INT_MASK_NF_422_IN	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
12	INT_MASK_GRPX3_ST	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
11	INT_MASK_GRPX2_ST	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
10	INT_MASK_GRPX1_ST	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
9	INT_MASK_VIP2_UP_UV	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
8	INT_MASK_VIP2_UP_Y	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
7	INT_MASK_VIP2_LO_UV	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
6	INT_MASK_VIP2_LO_Y	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
5	INT_MASK_VIP1_UP_UV	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
4	INT_MASK_VIP1_UP_Y	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
3	INT_MASK_VIP1_LO_UV	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
2	INT_MASK_VIP1_LO_Y	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
1	INT_MASK_GRPX3_DATA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
0	INT_MASK_GRPX2_DATA	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0

Table 7-340. Register Call Summary for Register VIP_INT0_CLIENT1_INT_MASK

VIP Functional Description
- VPDMA Interrupts: [0]

VIP Register Manual
- VIP VPDMA Register Summary: [3]

Table 7-341. VIP_INT0_LIST0_INT_STAT

Address Offset	0x0000 0088		
Physical Address	0x4897 D088	**Instance**	VIP_VPDMA
Description	This register gives the information of the interrupts that have triggered since last cleared by the process that is servicing vpdma_int0.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INT_STAT_CONTROL_DESCRIPTOR_INT15	INT_STAT_CONTROL_DESCRIPTOR_INT14	INT_STAT_CONTROL_DESCRIPTOR_INT13	INT_STAT_CONTROL_DESCRIPTOR_INT12	INT_STAT_CONTROL_DESCRIPTOR_INT11	INT_STAT_CONTROL_DESCRIPTOR_INT10	INT_STAT_CONTROL_DESCRIPTOR_INT9	INT_STAT_CONTROL_DESCRIPTOR_INT8	INT_STAT_CONTROL_DESCRIPTOR_INT7	INT_STAT_CONTROL_DESCRIPTOR_INT6	INT_STAT_CONTROL_DESCRIPTOR_INT5	INT_STAT_CONTROL_DESCRIPTOR_INT4	INT_STAT_CONTROL_DESCRIPTOR_INT3	INT_STAT_CONTROL_DESCRIPTOR_INT2	INT_STAT_CONTROL_DESCRIPTOR_INT1	INT_STAT_CONTROL_DESCRIPTOR_INT0	INT_STAT_LIST7_NOTIFY	INT_STAT_LIST7_COMPLETE	INT_STAT_LIST6_NOTIFY	INT_STAT_LIST6_COMPLETE	INT_STAT_LIST5_NOTIFY	INT_STAT_LIST5_COMPLETE	INT_STAT_LIST4_NOTIFY	INT_STAT_LIST4_COMPLETE	INT_STAT_LIST3_NOTIFY	INT_STAT_LIST3_COMPLETE	INT_STAT_LIST2_NOTIFY	INT_STAT_LIST2_COMPLETE	INT_STAT_LIST1_NOTIFY	INT_STAT_LIST1_COMPLETE	INT_STAT_LIST0_NOTIFY	INT_STAT_LIST0_COMPLETE

Bits	Field Name	Description	Type	Reset
31	INT_STAT_CONTROL_DESCRIPTOR_INT15	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 15. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
30	INT_STAT_CONTROL_DESCRIPTOR_INT14	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 14. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
29	INT_STAT_CONTROL_DESCRIPTOR_INT13	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 13. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
28	INT_STAT_CONTROL_DESCRIPTOR_INT12	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 12. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
27	INT_STAT_CONTROL_DESCRIPTOR_INT11	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 11. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
26	INT_STAT_CONTROL_DESCRIPTOR_INT10	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 10. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
25	INT_STAT_CONTROL_DESCRIPTOR_INT9	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 9. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
24	INT_STAT_CONTROL_DESCRIPTOR_INT8	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 8. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
23	INT_STAT_CONTROL_DESCRIPTOR_INT7	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 7. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
22	INT_STAT_CONTROL_DESCRIPTOR_INT6	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 6. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
21	INT_STAT_CONTROL_DESCRIPTOR_INT5	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 5. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
20	INT_STAT_CONTROL_DESCRIPTOR_INT4	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 4. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
19	INT_STAT_CONTROL_DESCRIPTOR_INT3	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 3. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
18	INT_STAT_CONTROL_DESCRIPTOR_INT2	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 2. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
17	INT_STAT_CONTROL_DESCRIPTOR_INT1	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 1. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
16	INT_STAT_CONTROL_DESCRIPTOR_INT0	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 0. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
15	INT_STAT_LIST7_NOTIFY	A channel set by List 7 has completed and the Notify bit had been set in the descriptor for that channel. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
14	INT_STAT_LIST7_COMPLETE	List 7 has completed and a new list can be loaded. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
13	INT_STAT_LIST6_NOTIFY	A channel set by List 6 has completed and the Notify bit had been set in the descriptor for that channel. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
12	INT_STAT_LIST6_COMPLETE	List 6 has completed and a new list can be loaded. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
11	INT_STAT_LIST5_NOTIFY	A channel set by List 5 has completed and the Notify bit had been set in the descriptor for that channel. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
10	INT_STAT_LIST5_COMPLETE	List 5 has completed and a new list can be loaded. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
9	INT_STAT_LIST4_NOTIFY	A channel set by List 4 has completed and the Notify bit had been set in the descriptor for that channel. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
8	INT_STAT_LIST4_COMPLETE	List 4 has completed and a new list can be loaded. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
7	INT_STAT_LIST3_NOTIFY	A channel set by List 3 has completed and the Notify bit had been set in the descriptor for that channel. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
6	INT_STAT_LIST3_COMPLETE	List 3 has completed and a new list can be loaded. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
5	INT_STAT_LIST2_NOTIFY	A channel set by List 2 has completed and the Notify bit had been set in the descriptor for that channel. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
4	INT_STAT_LIST2_COMPLETE	List 2 has completed and a new list can be loaded. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
3	INT_STAT_LIST1_NOTIFY	A channel set by List 1 has completed and the Notify bit had been set in the descriptor for that channel. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
2	INT_STAT_LIST1_COMPLETE	List 1 has completed and a new list can be loaded. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
1	INT_STAT_LIST0_NOTIFY	A channel set by List 0 has completed and the Notify bit had been set in the descriptor for that channel. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
0	INT_STAT_LIST0_COMPLETE	List 0 has completed and a new list can be loaded. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Table 7-342. Register Call Summary for Register VIP_INT0_LIST0_INT_STAT

VIP Functional Description • VPDMA Interrupts: [0][1][2]
VIP Register Manual • VIP VPDMA Register Summary: [5]

Table 7-343. VIP_INT0_LIST0_INT_MASK

Address Offset	0x0000 008C		
Physical Address	0x4897 D08C	**Instance**	VIP_VPDMA
Description	The register gives the information of the interrupts that should be masked and not generate an interrupt for vpdma_int0.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INT_MASK_CONTROL_DESCRIPTOR_INT15	INT_MASK_CONTROL_DESCRIPTOR_INT14	INT_MASK_CONTROL_DESCRIPTOR_INT13	INT_MASK_CONTROL_DESCRIPTOR_INT12	INT_MASK_CONTROL_DESCRIPTOR_INT11	INT_MASK_CONTROL_DESCRIPTOR_INT10	INT_MASK_CONTROL_DESCRIPTOR_INT9	INT_MASK_CONTROL_DESCRIPTOR_INT8	INT_MASK_CONTROL_DESCRIPTOR_INT7	INT_MASK_CONTROL_DESCRIPTOR_INT6	INT_MASK_CONTROL_DESCRIPTOR_INT5	INT_MASK_CONTROL_DESCRIPTOR_INT4	INT_MASK_CONTROL_DESCRIPTOR_INT3	INT_MASK_CONTROL_DESCRIPTOR_INT2	INT_MASK_CONTROL_DESCRIPTOR_INT1	INT_MASK_CONTROL_DESCRIPTOR_INT0	INT_MASK_LIST7_NOTIFY	INT_MASK_LIST7_COMPLETE	INT_MASK_LIST6_NOTIFY	INT_MASK_LIST6_COMPLETE	INT_MASK_LIST5_NOTIFY	INT_MASK_LIST5_COMPLETE	INT_MASK_LIST4_NOTIFY	INT_MASK_LIST4_COMPLETE	INT_MASK_LIST3_NOTIFY	INT_MASK_LIST3_COMPLETE	INT_MASK_LIST2_NOTIFY	INT_MASK_LIST2_COMPLETE	INT_MASK_LIST1_NOTIFY	INT_MASK_LIST1_COMPLETE	INT_MASK_LIST0_NOTIFY	INT_MASK_LIST0_COMPLETE

Bits	Field Name	Description	Type	Reset
31	INT_MASK_CONTROL_DESCRIPTOR_INT15	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
30	INT_MASK_CONTROL_DESCRIPTOR_INT14	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
29	INT_MASK_CONTROL_DESCRIPTOR_INT13	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
28	INT_MASK_CONTROL_DESCRIPTOR_INT12	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
27	INT_MASK_CONTROL_DESCRIPTOR_INT11	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
26	INT_MASK_CONTROL_DESCRIPTOR_INT10	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
25	INT_MASK_CONTROL_DESCRIPTOR_INT9	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
24	INT_MASK_CONTROL_DESCRIPTOR_INT8	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
23	INT_MASK_CONTROL_DESCRIPTOR_INT7	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
22	INT_MASK_CONTROL_DESCRIPTOR_INT6	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
21	INT_MASK_CONTROL_DESCRIPTOR_INT5	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0

Bits	Field Name	Description	Type	Reset
20	INT_MASK_CONTROL_DESCRIPTOR_INT4	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
19	INT_MASK_CONTROL_DESCRIPTOR_INT3	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
18	INT_MASK_CONTROL_DESCRIPTOR_INT2	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
17	INT_MASK_CONTROL_DESCRIPTOR_INT1	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
16	INT_MASK_CONTROL_DESCRIPTOR_INT0	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
15	INT_MASK_LIST7_NOTIFY	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
14	INT_MASK_LIST7_COMPLETE	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
13	INT_MASK_LIST6_NOTIFY	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
12	INT_MASK_LIST6_COMPLETE	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
11	INT_MASK_LIST5_NOTIFY	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
10	INT_MASK_LIST5_COMPLETE	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
9	INT_MASK_LIST4_NOTIFY	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
8	INT_MASK_LIST4_COMPLETE	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
7	INT_MASK_LIST3_NOTIFY	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
6	INT_MASK_LIST3_COMPLETE	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
5	INT_MASK_LIST2_NOTIFY	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
4	INT_MASK_LIST2_COMPLETE	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
3	INT_MASK_LIST1_NOTIFY	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
2	INT_MASK_LIST1_COMPLETE	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
1	INT_MASK_LIST0_NOTIFY	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0

Bits	Field Name	Description	Type	Reset
0	INT_MASK_LIST0_COMPLETE	The interrupt for should generate an interrupt on interrupt vpdma_int0. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0

Table 7-344. Register Call Summary for Register VIP_INT0_LIST0_INT_MASK

VIP Functional Description
- VPDMA Interrupts: [0]

VIP Register Manual
- VIP VPDMA Register Summary: [3]

Table 7-345. VIP_INT1_CHANNEL0_INT_STAT

Address Offset	0x0000 0090		
Physical Address	0x4897 D090	**Instance**	VIP_VPDMA
Description	This register gives the information of the interrupts that have triggered since last cleared by the process that is servicing vpdma_int1.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INT_STAT_GRPX3	INT_STAT_GRPX2	INT_STAT_GRPX1	INT_STAT_SCALER_OUT	RESERVED								INT_STAT_SCALER_CHROMA	INT_STAT_SCALER_LUMA	INT_STAT_HQ_SCALER	RESERVED	INT_STAT_HQ_MV_OUT	RESERVED		INT_STAT_HQ_MV	RESERVED				INT_STAT_HQ_VID3_CHROMA	INT_STAT_HQ_VID3_LUMA	INT_STAT_HQ_VID2_CHROMA	INT_STAT_HQ_VID2_LUMA	INT_STAT_HQ_VID1_CHROMA	INT_STAT_HQ_VID1_LUMA		

Bits	Field Name	Description	Type	Reset
31	INT_STAT_GRPX3	The last write DMA transaction has completed for channel scaler_out. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client sc_out then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
30	INT_STAT_GRPX2	The last write DMA transaction has completed for channel scaler_out. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client sc_out then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
29	INT_STAT_GRPX1	The last write DMA transaction has completed for channel scaler_out. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client sc_out then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
28	INT_STAT_SCALER_OUT	The last write DMA transaction has completed for channel scaler_out. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client sc_out then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
27:20	RESERVED	Reserved	R	0x00
19	INT_STAT_SCALER_CHROMA	The last write DMA transaction has completed for channel scaler_luma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
18	INT_STAT_SCALER_LUMA	The last write DMA transaction has completed for channel scaler_luma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
17	INT_STAT_HQ_SCALER	The last write DMA transaction has completed for channel hq_scaler. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client dei_sc_out then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
16	RESERVED	Reserved	R	0x0
15	INT_STAT_HQ_MV_OUT	The last write DMA transaction has completed for channel hq_mv_out. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client dei_hq_mv_out then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
14:13	RESERVED	Reserved	R	0x0
12	INT_STAT_HQ_MV	The last read DMA transaction has occurred for channel hq_mv and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client dei_hq_mv_in will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
11:6	RESERVED	Reserved	R	0x00
5	INT_STAT_HQ_VID3_CHROMA	The last write DMA transaction has completed for channel hq_vid3_chroma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
4	INT_STAT_HQ_VID3_LUMA	The last write DMA transaction has completed for channel hq_vid3_luma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
3	INT_STAT_HQ_VID2_CHROMA	The last write DMA transaction has completed for channel hq_vid2_chroma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
2	INT_STAT_HQ_VID2_LUMA	The last write DMA transaction has completed for channel hq_vid2_luma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
1	INT_STAT_HQ_VID1_CHROMA	The last write DMA transaction has completed for channel hq_vid1_chroma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
0	INT_STAT_HQ_VID1_LUMA	The last write DMA transaction has completed for channel hq_vid1_luma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Table 7-346. Register Call Summary for Register VIP_INT1_CHANNEL0_INT_STAT

VIP Functional Description
- VPDMA Interrupts: [0]

VIP Register Manual
- VIP VPDMA Register Summary: [3]

Table 7-347. VIP_INT1_CHANNEL0_INT_MASK

Address Offset	0x0000 0094		
Physical Address	0x4897 D094	**Instance**	VIP_VPDMA
Description	The register gives the information of the interrupts that should be masked and not generate an interrupt for vpdma_int1.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INT_MASK_GRPX3	INT_MASK_GRPX2	INT_MASK_GRPX1	INT_MASK_SCALER_OUT	RESERVED								INT_MASK_SCALER_CHROMA	INT_MASK_SCALER_LUMA	INT_MASK_HQ_SCALER	RESERVED	INT_MASK_HQ_MV_OUT	RESERVED		INT_MASK_HQ_MV	RESERVED				INT_MASK_HQ_VID3_CHROMA	INT_MASK_HQ_VID3_LUMA	INT_MASK_HQ_VID2_CHROMA	INT_MASK_HQ_VID2_LUMA	INT_MASK_HQ_VID1_CHROMA	INT_MASK_HQ_VID1_LUMA	

Bits	Field Name	Description	Type	Reset
31	INT_MASK_GRPX3	The interrupt for Graphcis 2 Data should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
30	INT_MASK_GRPX2	The interrupt for Graphics 1 Data should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
29	INT_MASK_GRPX1	The interrupt for Graphics 0 Data should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
28	INT_MASK_SCALER_OUT	The interrupt for Low Cost DEI Scalar Write to Memory should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
27:20	RESERVED	Reserved	R	0x00
19	INT_MASK_SCALER_CHROMA	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
18	INT_MASK_SCALER_LUMA	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
17	INT_MASK_HQ_SCALER	The interrupt for High Quality DEI Scaler Write to Memory should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
16	RESERVED	Reserved	R	0x0
15	INT_MASK_HQ_MV_OUT	The interrupt for Low Cost DEI Motion Vector Write should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
14:13	RESERVED	Reserved	R	0x0
12	INT_MASK_HQ_MV	The interrupt for Low Cost DEI Motion Vector should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
11:6	RESERVED	Reserved	R	0x00
5	INT_MASK_HQ_VID3_CHROMA	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
4	INT_MASK_HQ_VID3_LUMA	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
3	INT_MASK_HQ_VID2_CHROMA	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
2	INT_MASK_HQ_VID2_LUMA	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
1	INT_MASK_HQ_VID1_CHROMA	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
0	INT_MASK_HQ_VID1_LUMA	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0

Table 7-348. Register Call Summary for Register VIP_INT1_CHANNEL0_INT_MASK

VIP Functional Description • VPDMA Interrupts: [0]
VIP Register Manual • VIP VPDMA Register Summary: [3]

Table 7-349. VIP_INT1_CHANNEL1_INT_STAT

Address Offset	0x0000 0098		
Physical Address	0x4897 D098	**Instance**	VIP_VPDMA
Description	This register gives the information of the interrupts that have triggered since last cleared by the process that is servicing vpdma_int1.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INT_STAT_VIP1_MULT_PORTB_SRC9	INT_STAT_VIP1_MULT_PORTB_SRC8	INT_STAT_VIP1_MULT_PORTB_SRC7	INT_STAT_VIP1_MULT_PORTB_SRC6	INT_STAT_VIP1_MULT_PORTB_SRC5	INT_STAT_VIP1_MULT_PORTB_SRC4	INT_STAT_VIP1_MULT_PORTB_SRC3	INT_STAT_VIP1_MULT_PORTB_SRC2	INT_STAT_VIP1_MULT_PORTB_SRC1	INT_STAT_VIP1_MULT_PORTB_SRC0	INT_STAT_VIP1_MULT_PORTA_SRC15	INT_STAT_VIP1_MULT_PORTA_SRC14	INT_STAT_VIP1_MULT_PORTA_SRC13	INT_STAT_VIP1_MULT_PORTA_SRC12	INT_STAT_VIP1_MULT_PORTA_SRC11	INT_STAT_VIP1_MULT_PORTA_SRC10	INT_STAT_VIP1_MULT_PORTA_SRC9	INT_STAT_VIP1_MULT_PORTA_SRC8	INT_STAT_VIP1_MULT_PORTA_SRC7	INT_STAT_VIP1_MULT_PORTA_SRC6	INT_STAT_VIP1_MULT_PORTA_SRC5	INT_STAT_VIP1_MULT_PORTA_SRC4	INT_STAT_VIP1_MULT_PORTA_SRC3	INT_STAT_VIP1_MULT_PORTA_SRC2	INT_STAT_VIP1_MULT_PORTA_SRC1	INT_STAT_VIP1_MULT_PORTA_SRC0	RESERVED						

Bits	Field Name	Description	Type	Reset
31	INT_STAT_VIP1_MULT_PORTB_SRC9	The last write DMA transaction has completed for channel vip1_mult_portb_src9. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
30	INT_STAT_VIP1_MULT_PORTB_SRC8	The last write DMA transaction has completed for channel vip1_mult_portb_src8. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
29	INT_STAT_VIP1_MULT_PORTB_SRC7	The last write DMA transaction has completed for channel vip1_mult_portb_src7. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
28	INT_STAT_VIP1_MULT_PORTB_SRC6	The last write DMA transaction has completed for channel vip1_mult_portb_src6. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
27	INT_STAT_VIP1_MULT_PORTB_SRC5	The last write DMA transaction has completed for channel vip1_mult_portb_src5. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
26	INT_STAT_VIP1_MULT_PORTB_SRC4	The last write DMA transaction has completed for channel vip1_mult_portb_src4. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
25	INT_STAT_VIP1_MULT_PORTB_SRC3	The last write DMA transaction has completed for channel vip1_mult_portb_src3. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
24	INT_STAT_VIP1_MULT_PORTB_SRC2	The last write DMA transaction has completed for channel vip1_mult_portb_src2. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
23	INT_STAT_VIP1_MULT_PORTB_SRC1	The last write DMA transaction has completed for channel vip1_mult_portb_src1. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
22	INT_STAT_VIP1_MULT_PORTB_SRC0	The last write DMA transaction has completed for channel vip1_mult_portb_src0. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
21	INT_STAT_VIP1_MULT_PORTA_SRC15	The last write DMA transaction has completed for channel vip1_mult_porta_src15. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
20	INT_STAT_VIP1_MULT_PORTA_SRC14	The last write DMA transaction has completed for channel vip1_mult_porta_src14. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
19	INT_STAT_VIP1_MULT_PORTA_SRC13	The last write DMA transaction has completed for channel vip1_mult_porta_src13. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
18	INT_STAT_VIP1_MULT_PORTA_SRC12	The last write DMA transaction has completed for channel vip1_mult_porta_src12. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
17	INT_STAT_VIP1_MULT_PORTA_SRC11	The last write DMA transaction has completed for channel vip1_mult_porta_src11. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
16	INT_STAT_VIP1_MULT_PORTA_SRC10	The last write DMA transaction has completed for channel vip1_mult_porta_src10. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
15	INT_STAT_VIP1_MULT_PORTA_SRC9	The last write DMA transaction has completed for channel vip1_mult_porta_src9. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
14	INT_STAT_VIP1_MULT_PORTA_SRC8	The last write DMA transaction has completed for channel vip1_mult_porta_src8. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
13	INT_STAT_VIP1_MULT_PORTA_SRC7	The last write DMA transaction has completed for channel vip1_mult_porta_src7. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
12	INT_STAT_VIP1_MULT_PORTA_SRC6	The last write DMA transaction has completed for channel vip1_mult_porta_src6. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
11	INT_STAT_VIP1_MULT_PORTA_SRC5	The last write DMA transaction has completed for channel vip1_mult_porta_src5. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
10	INT_STAT_VIP1_MULT_PORTA_SRC4	The last write DMA transaction has completed for channel vip1_mult_porta_src4. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
9	INT_STAT_VIP1_MULT_PORTA_SRC3	The last write DMA transaction has completed for channel vip1_mult_porta_src3. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
8	INT_STAT_VIP1_MULT_PORTA_SRC2	The last write DMA transaction has completed for channel vip1_mult_porta_src2. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
7	INT_STAT_VIP1_MULT_PORTA_SRC1	The last write DMA transaction has completed for channel vip1_mult_porta_src1. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
6	INT_STAT_VIP1_MULT_PORTA_SRC0	The last write DMA transaction has completed for channel vip1_mult_porta_src0. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
5:0	RESERVED	Reserved	R	0x00

Table 7-350. Register Call Summary for Register VIP_INT1_CHANNEL1_INT_STAT

VIP Functional Description
- VPDMA Interrupts: [0]

VIP Register Manual
- VIP VPDMA Register Summary: [3]

Table 7-351. VIP_INT1_CHANNEL1_INT_MASK

Address Offset	0x0000 009C		
Physical Address	0x4897 D09C	**Instance**	VIP_VPDMA
Description	The register gives the information of the interrupts that should be masked and not generate an interrupt for vpdma_int1.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INT_MASK_VIP1_MULT_PORTB_SRC9	INT_MASK_VIP1_MULT_PORTB_SRC8	INT_MASK_VIP1_MULT_PORTB_SRC7	INT_MASK_VIP1_MULT_PORTB_SRC6	INT_MASK_VIP1_MULT_PORTB_SRC5	INT_MASK_VIP1_MULT_PORTB_SRC4	INT_MASK_VIP1_MULT_PORTB_SRC3	INT_MASK_VIP1_MULT_PORTB_SRC2	INT_MASK_VIP1_MULT_PORTB_SRC1	INT_MASK_VIP1_MULT_PORTB_SRC0	INT_MASK_VIP1_MULT_PORTA_SRC15	INT_MASK_VIP1_MULT_PORTA_SRC14	INT_MASK_VIP1_MULT_PORTA_SRC13	INT_MASK_VIP1_MULT_PORTA_SRC12	INT_MASK_VIP1_MULT_PORTA_SRC11	INT_MASK_VIP1_MULT_PORTA_SRC10	INT_MASK_VIP1_MULT_PORTA_SRC9	INT_MASK_VIP1_MULT_PORTA_SRC8	INT_MASK_VIP1_MULT_PORTA_SRC7	INT_MASK_VIP1_MULT_PORTA_SRC6	INT_MASK_VIP1_MULT_PORTA_SRC5	INT_MASK_VIP1_MULT_PORTA_SRC4	INT_MASK_VIP1_MULT_PORTA_SRC3	INT_MASK_VIP1_MULT_PORTA_SRC2	INT_MASK_VIP1_MULT_PORTA_SRC1	INT_MASK_VIP1_MULT_PORTA_SRC0	RESERVED						

Bits	Field Name	Description	Type	Reset
31	INT_MASK_VIP1_MULT_PORTB_SRC9	The interrupt for Video Input 1 Port B Channel 9 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
30	INT_MASK_VIP1_MULT_PORTB_SRC8	The interrupt for Video Input 1 Port B Channel 8 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
29	INT_MASK_VIP1_MULT_PORTB_SRC7	The interrupt for Video Input 1 Port B Channel 7 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
28	INT_MASK_VIP1_MULT_PORTB_SRC6	The interrupt for Video Input 1 Port B Channel 6 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
27	INT_MASK_VIP1_MULT_PORTB_SRC5	The interrupt for Video Input 1 Port B Channel 5 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
26	INT_MASK_VIP1_MULT_PORTB_SRC4	The interrupt for Video Input 1 Port B Channel 4 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
25	INT_MASK_VIP1_MULT_PORTB_SRC3	The interrupt for Video Input 1 Port B Channel 3 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
24	INT_MASK_VIP1_MULT_PORTB_SRC2	The interrupt for Video Input 1 Port B Channel 2 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
23	INT_MASK_VIP1_MULT_PORTB_SRC1	The interrupt for Video Input 1 Port B Channel 1 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
22	INT_MASK_VIP1_MULT_PORTB_SRC0	The interrupt for Video Input 1 Port B Channel 0 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
21	INT_MASK_VIP1_MULT_PORTA_SRC15	The interrupt for Video Input 1 Port A Channel 15 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
20	INT_MASK_VIP1_MULT_PORTA_SRC14	The interrupt for Video Input 1 Port A Channel 14 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
19	INT_MASK_VIP1_MULT_PORTA_SRC13	The interrupt for Video Input 1 Port A Channel 13 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
18	INT_MASK_VIP1_MULT_PORTA_SRC12	The interrupt for Video Input 1 Port A Channel 12 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
17	INT_MASK_VIP1_MULT_PORTA_SRC11	The interrupt for Video Input 1 Port A Channel 11 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
16	INT_MASK_VIP1_MULT_PORTA_SRC10	The interrupt for Video Input 1 Port A Channel 10 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
15	INT_MASK_VIP1_MULT_PORTA_SRC9	The interrupt for Video Input 1 Port A Channel 9 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
14	INT_MASK_VIP1_MULT_PORTA_SRC8	The interrupt for Video Input 1 Port A Channel 8 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
13	INT_MASK_VIP1_MULT_PORTA_SRC7	The interrupt for Video Input 1 Port A Channel 7 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
12	INT_MASK_VIP1_MULT_PORTA_SRC6	The interrupt for Video Input 1 Port A Channel 6 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0

Bits	Field Name	Description	Type	Reset
11	INT_MASK_VIP1_MULT_PORTA_SRC5	The interrupt for Video Input 1 Port A Channel 5 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
10	INT_MASK_VIP1_MULT_PORTA_SRC4	The interrupt for Video Input 1 Port A Channel 4 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
9	INT_MASK_VIP1_MULT_PORTA_SRC3	The interrupt for Video Input 1 Port A Channel 3 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
8	INT_MASK_VIP1_MULT_PORTA_SRC2	The interrupt for Video Input 1 Port A Channel 2 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
7	INT_MASK_VIP1_MULT_PORTA_SRC1	The interrupt for Video Input 1 Port A Channel 1 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
6	INT_MASK_VIP1_MULT_PORTA_SRC0	The interrupt for Video Input 1 Port A Channel 0 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
5:0	RESERVED	Reserved	R	0x00

Table 7-352. Register Call Summary for Register VIP_INT1_CHANNEL1_INT_MASK

VIP Functional Description
- VPDMA Interrupts: [0]

VIP Register Manual
- VIP VPDMA Register Summary: [3]

Table 7-353. VIP_INT1_CHANNEL2_INT_STAT

Address Offset	0x0000 00A0		
Physical Address	0x4897 D0A0	**Instance**	VIP_VPDMA
Description	This register gives the information of the interrupts that have triggered since last cleared by the process that is servicing vpdma_int1.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INT_STAT_VIP1_MULT_ANCB_SRC9	INT_STAT_VIP1_MULT_ANCB_SRC8	INT_STAT_VIP1_MULT_ANCB_SRC7	INT_STAT_VIP1_MULT_ANCB_SRC6	INT_STAT_VIP1_MULT_ANCB_SRC5	INT_STAT_VIP1_MULT_ANCB_SRC4	INT_STAT_VIP1_MULT_ANCB_SRC3	INT_STAT_VIP1_MULT_ANCB_SRC2	INT_STAT_VIP1_MULT_ANCB_SRC1	INT_STAT_VIP1_MULT_ANCB_SRC0	INT_STAT_VIP1_MULT_ANCA_SRC15	INT_STAT_VIP1_MULT_ANCA_SRC14	INT_STAT_VIP1_MULT_ANCA_SRC13	INT_STAT_VIP1_MULT_ANCA_SRC12	INT_STAT_VIP1_MULT_ANCA_SRC11	INT_STAT_VIP1_MULT_ANCA_SRC10	INT_STAT_VIP1_MULT_ANCA_SRC9	INT_STAT_VIP1_MULT_ANCA_SRC8	INT_STAT_VIP1_MULT_ANCA_SRC7	INT_STAT_VIP1_MULT_ANCA_SRC6	INT_STAT_VIP1_MULT_ANCA_SRC5	INT_STAT_VIP1_MULT_ANCA_SRC4	INT_STAT_VIP1_MULT_ANCA_SRC3	INT_STAT_VIP1_MULT_ANCA_SRC2	INT_STAT_VIP1_MULT_ANCA_SRC1	INT_STAT_VIP1_MULT_ANCA_SRC0	INT_STAT_VIP1_MULT_PORTB_SRC15	INT_STAT_VIP1_MULT_PORTB_SRC14	INT_STAT_VIP1_MULT_PORTB_SRC13	INT_STAT_VIP1_MULT_PORTB_SRC12	INT_STAT_VIP1_MULT_PORTB_SRC11	INT_STAT_VIP1_MULT_PORTB_SRC10

Bits	Field Name	Description	Type	Reset
31	INT_STAT_VIP1_MULT_ANCB_SRC9	The last write DMA transaction has completed for channel vip1_mult_ancb_src9. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
30	INT_STAT_VIP1_MULT_ANCB_SRC8	The last write DMA transaction has completed for channel vip1_mult_ancb_src8. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
29	INT_STAT_VIP1_MULT_ANCB_SRC7	The last write DMA transaction has completed for channel vip1_mult_ancb_src7. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
28	INT_STAT_VIP1_MULT_ANCB_SRC6	The last write DMA transaction has completed for channel vip1_mult_ancb_src6. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
27	INT_STAT_VIP1_MULT_ANCB_SRC5	The last write DMA transaction has completed for channel vip1_mult_ancb_src5. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
26	INT_STAT_VIP1_MULT_ANCB_SRC4	The last write DMA transaction has completed for channel vip1_mult_ancb_src4. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
25	INT_STAT_VIP1_MULT_ANCB_SRC3	The last write DMA transaction has completed for channel vip1_mult_ancb_src3. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
24	INT_STAT_VIP1_MULT_ANCB_SRC2	The last write DMA transaction has completed for channel vip1_mult_ancb_src2. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
23	INT_STAT_VIP1_MULT_ANCB_SRC1	The last write DMA transaction has completed for channel vip1_mult_ancb_src1. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
22	INT_STAT_VIP1_MULT_ANCB_SRC0	The last write DMA transaction has completed for channel vip1_mult_ancb_src0. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
21	INT_STAT_VIP1_MULT_ANCA_SRC15	The last write DMA transaction has completed for channel vip1_mult_anca_src15. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
20	INT_STAT_VIP1_MULT_ANCA_SRC14	The last write DMA transaction has completed for channel vip1_mult_anca_src14. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
19	INT_STAT_VIP1_MULT_ANCA_SRC13	The last write DMA transaction has completed for channel vip1_mult_anca_src13. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
18	INT_STAT_VIP1_MULT_ANCA_SRC12	The last write DMA transaction has completed for channel vip1_mult_anca_src12. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
17	INT_STAT_VIP1_MULT_ANCA_SRC11	The last write DMA transaction has completed for channel vip1_mult_anca_src11. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
16	INT_STAT_VIP1_MULT_ANCA_SRC10	The last write DMA transaction has completed for channel vip1_mult_anca_src10. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
15	INT_STAT_VIP1_MULT_ANCA_SRC9	The last write DMA transaction has completed for channel vip1_mult_anca_src9. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
14	INT_STAT_VIP1_MULT_ANCA_SRC8	The last write DMA transaction has completed for channel vip1_mult_anca_src8. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
13	INT_STAT_VIP1_MULT_ANCA_SRC7	The last write DMA transaction has completed for channel vip1_mult_anca_src7. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
12	INT_STAT_VIP1_MULT_ANCA_SRC6	The last write DMA transaction has completed for channel vip1_mult_anca_src6. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
11	INT_STAT_VIP1_MULT_ANCA_SRC5	The last write DMA transaction has completed for channel vip1_mult_anca_src5. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
10	INT_STAT_VIP1_MULT_ANCA_SRC4	The last write DMA transaction has completed for channel vip1_mult_anca_src4. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
9	INT_STAT_VIP1_MULT_ANCA_SRC3	The last write DMA transaction has completed for channel vip1_mult_anca_src3. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
8	INT_STAT_VIP1_MULT_ANCA_SRC2	The last write DMA transaction has completed for channel vip1_mult_anca_src2. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
7	INT_STAT_VIP1_MULT_ANCA_SRC1	The last write DMA transaction has completed for channel vip1_mult_anca_src1. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
6	INT_STAT_VIP1_MULT_ANCA_SRC0	The last write DMA transaction has completed for channel vip1_mult_anca_src0. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
5	INT_STAT_VIP1_MULT_PORTB_SRC15	The last write DMA transaction has completed for channel vip1_mult_portb_src15. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
4	INT_STAT_VIP1_MULT_PORTB_SRC14	The last write DMA transaction has completed for channel vip1_mult_portb_src14. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
3	INT_STAT_VIP1_MULT_PORTB_SRC13	The last write DMA transaction has completed for channel vip1_mult_portb_src13. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
2	INT_STAT_VIP1_MULT_PORTB_SRC12	The last write DMA transaction has completed for channel vip1_mult_portb_src12. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
1	INT_STAT_VIP1_MULT_PORTB_SRC11	The last write DMA transaction has completed for channel vip1_mult_portb_src11. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
0	INT_STAT_VIP1_MULT_PORTB_SRC10	The last write DMA transaction has completed for channel vip1_mult_portb_src10. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Table 7-354. Register Call Summary for Register VIP_INT1_CHANNEL2_INT_STAT

VIP Functional Description • VPDMA Interrupts: [0]
VIP Register Manual • VIP VPDMA Register Summary: [3]

Table 7-355. VIP_INT1_CHANNEL2_INT_MASK

Address Offset	0x0000 00A4		
Physical Address	0x4897 D0A4	**Instance**	VIP_VPDMA
Description	The register gives the information of the interrupts that should be masked and not generate an interrupt for vpdma_int1.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INT_MASK_VIP1_MULT_ANCB_SRC9	INT_MASK_VIP1_MULT_ANCB_SRC8	INT_MASK_VIP1_MULT_ANCB_SRC7	INT_MASK_VIP1_MULT_ANCB_SRC6	INT_MASK_VIP1_MULT_ANCB_SRC5	INT_MASK_VIP1_MULT_ANCB_SRC4	INT_MASK_VIP1_MULT_ANCB_SRC3	INT_MASK_VIP1_MULT_ANCB_SRC2	INT_MASK_VIP1_MULT_ANCB_SRC1	INT_MASK_VIP1_MULT_ANCB_SRC0	INT_MASK_VIP1_MULT_ANCA_SRC15	INT_MASK_VIP1_MULT_ANCA_SRC14	INT_MASK_VIP1_MULT_ANCA_SRC13	INT_MASK_VIP1_MULT_ANCA_SRC12	INT_MASK_VIP1_MULT_ANCA_SRC11	INT_MASK_VIP1_MULT_ANCA_SRC10	INT_MASK_VIP1_MULT_ANCA_SRC9	INT_MASK_VIP1_MULT_ANCA_SRC8	INT_MASK_VIP1_MULT_ANCA_SRC7	INT_MASK_VIP1_MULT_ANCA_SRC6	INT_MASK_VIP1_MULT_ANCA_SRC5	INT_MASK_VIP1_MULT_ANCA_SRC4	INT_MASK_VIP1_MULT_ANCA_SRC3	INT_MASK_VIP1_MULT_ANCA_SRC2	INT_MASK_VIP1_MULT_ANCA_SRC1	INT_MASK_VIP1_MULT_ANCA_SRC0	INT_MASK_VIP1_MULT_PORTB_SRC15	INT_MASK_VIP1_MULT_PORTB_SRC14	INT_MASK_VIP1_MULT_PORTB_SRC13	INT_MASK_VIP1_MULT_PORTB_SRC12	INT_MASK_VIP1_MULT_PORTB_SRC11	INT_MASK_VIP1_MULT_PORTB_SRC10

Bits	Field Name	Description	Type	Reset
31	INT_MASK_VIP1_MULT_ANCB_SRC9	The interrupt for Video Input 1 Port B Ancillary Data Channel 9 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
30	INT_MASK_VIP1_MULT_ANCB_SRC8	The interrupt for Video Input 1 Port B Ancillary Data Channel 8 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
29	INT_MASK_VIP1_MULT_ANCB_SRC7	The interrupt for Video Input 1 Port B Ancillary Data Channel 7 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
28	INT_MASK_VIP1_MULT_ANCB_SRC6	The interrupt for Video Input 1 Port B Ancillary Data Channel 6 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
27	INT_MASK_VIP1_MULT_ANCB_SRC5	The interrupt for Video Input 1 Port B Ancillary Data Channel 5 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
26	INT_MASK_VIP1_MULT_ANCB_SRC4	The interrupt for Video Input 1 Port B Ancillary Data Channel 4 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
25	INT_MASK_VIP1_MULT_ANCB_SRC3	The interrupt for Video Input 1 Port B Ancillary Data Channel 3 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
24	INT_MASK_VIP1_MULT_ANCB_SRC2	The interrupt for Video Input 1 Port B Ancillary Data Channel 2 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
23	INT_MASK_VIP1_MULT_ANCB_SRC1	The interrupt for Video Input 1 Port B Ancillary Data Channel 1 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
22	INT_MASK_VIP1_MULT_ANCB_SRC0	The interrupt for Video Input 1 Port B Ancillary Data Channel 0 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
21	INT_MASK_VIP1_MULT_ANCA_SRC15	The interrupt for Video Input 1 Port A Ancillary Data Channel 15 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0

Bits	Field Name	Description	Type	Reset
20	INT_MASK_VIP1_MULT_ANCA_SRC14	The interrupt for Video Input 1 Port A Ancillary Data Channel 14 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
19	INT_MASK_VIP1_MULT_ANCA_SRC13	The interrupt for Video Input 1 Port A Ancillary Data Channel 13 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
18	INT_MASK_VIP1_MULT_ANCA_SRC12	The interrupt for Video Input 1 Port A Ancillary Data Channel 12 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
17	INT_MASK_VIP1_MULT_ANCA_SRC11	The interrupt for Video Input 1 Port A Ancillary Data Channel 11 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
16	INT_MASK_VIP1_MULT_ANCA_SRC10	The interrupt for Video Input 1 Port A Ancillary Data Channel 10 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
15	INT_MASK_VIP1_MULT_ANCA_SRC9	The interrupt for Video Input 1 Port A Ancillary Data Channel 9 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
14	INT_MASK_VIP1_MULT_ANCA_SRC8	The interrupt for Video Input 1 Port A Ancillary Data Channel 8 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
13	INT_MASK_VIP1_MULT_ANCA_SRC7	The interrupt for Video Input 1 Port A Ancillary Data Channel 7 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
12	INT_MASK_VIP1_MULT_ANCA_SRC6	The interrupt for Video Input 1 Port A Ancillary Data Channel 6 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
11	INT_MASK_VIP1_MULT_ANCA_SRC5	The interrupt for Video Input 1 Port A Ancillary Data Channel 5 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
10	INT_MASK_VIP1_MULT_ANCA_SRC4	The interrupt for Video Input 1 Port A Ancillary Data Channel 4 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
9	INT_MASK_VIP1_MULT_ANCA_SRC3	The interrupt for Video Input 1 Port A Ancillary Data Channel 3 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
8	INT_MASK_VIP1_MULT_ANCA_SRC2	The interrupt for Video Input 1 Port A Ancillary Data Channel 2 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
7	INT_MASK_VIP1_MULT_ANCA_SRC1	The interrupt for Video Input 1 Port A Ancillary Data Channel 1 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
6	INT_MASK_VIP1_MULT_ANCA_SRC0	The interrupt for Video Input 1 Port A Ancillary Data Channel 0 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
5	INT_MASK_VIP1_MULT_PORTB_SRC15	The interrupt for Video Input 1 Port B Channel 15 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0

Bits	Field Name	Description	Type	Reset
4	INT_MASK_VIP1_MULT_PORTB_SRC14	The interrupt for Video Input 1 Port B Channel 14 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
3	INT_MASK_VIP1_MULT_PORTB_SRC13	The interrupt for Video Input 1 Port B Channel 13 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
2	INT_MASK_VIP1_MULT_PORTB_SRC12	The interrupt for Video Input 1 Port B Channel 12 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
1	INT_MASK_VIP1_MULT_PORTB_SRC11	The interrupt for Video Input 1 Port B Channel 11 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
0	INT_MASK_VIP1_MULT_PORTB_SRC10	The interrupt for Video Input 1 Port B Channel 10 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0

Table 7-356. Register Call Summary for Register VIP_INT1_CHANNEL2_INT_MASK

VIP Functional Description
- VPDMA Interrupts: [0]

VIP Register Manual
- VIP VPDMA Register Summary: [3]

Table 7-357. VIP_INT1_CHANNEL3_INT_STAT

Address Offset	0x0000 00A8		
Physical Address	0x4897 D0A8	**Instance**	VIP_VPDMA
Description	This register gives the information of the interrupts that have triggered since last cleared by the process that is servicing vpdma_int1.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INT_STAT_VIP2_MULT_PORTB_SRC3	INT_STAT_VIP2_MULT_PORTB_SRC2	INT_STAT_VIP2_MULT_PORTB_SRC1	INT_STAT_VIP2_MULT_PORTB_SRC0	INT_STAT_VIP2_MULT_PORTA_SRC15	INT_STAT_VIP2_MULT_PORTA_SRC14	INT_STAT_VIP2_MULT_PORTA_SRC13	INT_STAT_VIP2_MULT_PORTA_SRC12	INT_STAT_VIP2_MULT_PORTA_SRC11	INT_STAT_VIP2_MULT_PORTA_SRC10	INT_STAT_VIP2_MULT_PORTA_SRC9	INT_STAT_VIP2_MULT_PORTA_SRC8	INT_STAT_VIP2_MULT_PORTA_SRC7	INT_STAT_VIP2_MULT_PORTA_SRC6	INT_STAT_VIP2_MULT_PORTA_SRC5	INT_STAT_VIP2_MULT_PORTA_SRC4	INT_STAT_VIP2_MULT_PORTA_SRC3	INT_STAT_VIP2_MULT_PORTA_SRC2	INT_STAT_VIP2_MULT_PORTA_SRC1	INT_STAT_VIP2_MULT_PORTA_SRC0	INT_STAT_VIP1_PORTB_RGB	INT_STAT_VIP1_PORTA_RGB	INT_STAT_VIP1_PORTB_CHROMA	INT_STAT_VIP1_PORTB_LUMA	INT_STAT_VIP1_PORTA_CHROMA	INT_STAT_VIP1_PORTA_LUMA	INT_STAT_VIP1_MULT_ANCB_SRC15	INT_STAT_VIP1_MULT_ANCB_SRC14	INT_STAT_VIP1_MULT_ANCB_SRC13	INT_STAT_VIP1_MULT_ANCB_SRC12	INT_STAT_VIP1_MULT_ANCB_SRC11	INT_STAT_VIP1_MULT_ANCB_SRC10

Bits	Field Name	Description	Type	Reset
31	INT_STAT_VIP2_MULT_PORTB_SRC3	The last write DMA transaction has completed for channel vip2_mult_portb_src3. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
30	INT_STAT_VIP2_MULT_PORTB_SRC2	The last write DMA transaction has completed for channel vip2_mult_portb_src2. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
29	INT_STAT_VIP2_MULT_PORTB_SRC1	The last write DMA transaction has completed for channel vip2_mult_portb_src1. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
28	INT_STAT_VIP2_MULT_PORTB_SRC0	The last write DMA transaction has completed for channel vip2_mult_portb_src0. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
27	INT_STAT_VIP2_MULT_PORTA_SRC15	The last write DMA transaction has completed for channel vip2_mult_porta_src15. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
26	INT_STAT_VIP2_MULT_PORTA_SRC14	The last write DMA transaction has completed for channel vip2_mult_porta_src14. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
25	INT_STAT_VIP2_MULT_PORTA_SRC13	The last write DMA transaction has completed for channel vip2_mult_porta_src13. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
24	INT_STAT_VIP2_MULT_PORTA_SRC12	The last write DMA transaction has completed for channel vip2_mult_porta_src12. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
23	INT_STAT_VIP2_MULT_PORTA_SRC11	The last write DMA transaction has completed for channel vip2_mult_porta_src11. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
22	INT_STAT_VIP2_MULT_PORTA_SRC10	The last write DMA transaction has completed for channel vip2_mult_porta_src10. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
21	INT_STAT_VIP2_MULT_PORTA_SRC9	The last write DMA transaction has completed for channel vip2_mult_porta_src9. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
20	INT_STAT_VIP2_MULT_PORTA_SRC8	The last write DMA transaction has completed for channel vip2_mult_porta_src8. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
19	INT_STAT_VIP2_MULT_PORTA_SRC7	The last write DMA transaction has completed for channel vip2_mult_porta_src7. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
18	INT_STAT_VIP2_MULT_PORTA_SRC6	The last write DMA transaction has completed for channel vip2_mult_porta_src6. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
17	INT_STAT_VIP2_MULT_PORTA_SRC5	The last write DMA transaction has completed for channel vip2_mult_porta_src5. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
16	INT_STAT_VIP2_MULT_PORTA_SRC4	The last write DMA transaction has completed for channel vip2_mult_porta_src4. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
15	INT_STAT_VIP2_MULT_PORTA_SRC3	The last write DMA transaction has completed for channel vip2_mult_porta_src3. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
14	INT_STAT_VIP2_MULT_PORTA_SRC2	The last write DMA transaction has completed for channel vip2_mult_porta_src2. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
13	INT_STAT_VIP2_MULT_PORTA_SRC1	The last write DMA transaction has completed for channel vip2_mult_porta_src1. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
12	INT_STAT_VIP2_MULT_PORTA_SRC0	The last write DMA transaction has completed for channel vip2_mult_porta_src0. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
11	INT_STAT_VIP1_PORTB_RGB	The last write DMA transaction has completed for channel vip1_portb_rgb. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
10	INT_STAT_VIP1_PORTA_RGB	The last write DMA transaction has completed for channel vip1_porta_rgb. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_up_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
9	INT_STAT_VIP1_PORTB_CHROMA	The last write DMA transaction has completed for channel vip1_portb_chroma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
8	INT_STAT_VIP1_PORTB_LUMA	The last write DMA transaction has completed for channel vip1_portb_luma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
7	INT_STAT_VIP1_PORTA_CHROMA	The last write DMA transaction has completed for channel vip1_porta_chroma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
6	INT_STAT_VIP1_PORTA_LUMA	The last write DMA transaction has completed for channel vip1_porta_luma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
5	INT_STAT_VIP1_MULT_ANCB_SRC15	The last write DMA transaction has completed for channel vip1_mult_ancb_src15. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
4	INT_STAT_VIP1_MULT_ANCB_SRC14	The last write DMA transaction has completed for channel vip1_mult_ancb_src14. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
3	INT_STAT_VIP1_MULT_ANCB_SRC13	The last write DMA transaction has completed for channel vip1_mult_ancb_src13. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
2	INT_STAT_VIP1_MULT_ANCB_SRC12	The last write DMA transaction has completed for channel vip1_mult_ancb_src12. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
1	INT_STAT_VIP1_MULT_ANCB_SRC11	The last write DMA transaction has completed for channel vip1_mult_ancb_src11. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
0	INT_STAT_VIP1_MULT_ANCB_SRC10	The last write DMA transaction has completed for channel vip1_mult_ancb_src10. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip1_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Table 7-358. Register Call Summary for Register VIP_INT1_CHANNEL3_INT_STAT

VIP Functional Description
- VPDMA Interrupts: [0]

VIP Register Manual
- VIP VPDMA Register Summary: [3]

Table 7-359. VIP_INT1_CHANNEL3_INT_MASK

Address Offset	0x0000 00AC		
Physical Address	0x4897 D0AC	**Instance**	VIP_VPDMA
Description	The register gives the information of the interrupts that should be masked and not generate an interrupt for vpdma_int1.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INT_MASK_VIP2_MULT_PORTB_SRC3	INT_MASK_VIP2_MULT_PORTB_SRC2	INT_MASK_VIP2_MULT_PORTB_SRC1	INT_MASK_VIP2_MULT_PORTB_SRC0	INT_MASK_VIP2_MULT_PORTA_SRC15	INT_MASK_VIP2_MULT_PORTA_SRC14	INT_MASK_VIP2_MULT_PORTA_SRC13	INT_MASK_VIP2_MULT_PORTA_SRC12	INT_MASK_VIP2_MULT_PORTA_SRC11	INT_MASK_VIP2_MULT_PORTA_SRC10	INT_MASK_VIP2_MULT_PORTA_SRC9	INT_MASK_VIP2_MULT_PORTA_SRC8	INT_MASK_VIP2_MULT_PORTA_SRC7	INT_MASK_VIP2_MULT_PORTA_SRC6	INT_MASK_VIP2_MULT_PORTA_SRC5	INT_MASK_VIP2_MULT_PORTA_SRC4	INT_MASK_VIP2_MULT_PORTA_SRC3	INT_MASK_VIP2_MULT_PORTA_SRC2	INT_MASK_VIP2_MULT_PORTA_SRC1	INT_MASK_VIP2_MULT_PORTA_SRC0	INT_MASK_VIP1_PORTB_RGB	INT_MASK_VIP1_PORTA_RGB	INT_MASK_VIP1_PORTB_CHROMA	INT_MASK_VIP1_PORTB_LUMA	INT_MASK_VIP1_PORTA_CHROMA	INT_MASK_VIP1_PORTA_LUMA	INT_MASK_VIP1_MULT_ANCB_SRC15	INT_MASK_VIP1_MULT_ANCB_SRC14	INT_MASK_VIP1_MULT_ANCB_SRC13	INT_MASK_VIP1_MULT_ANCB_SRC12	INT_MASK_VIP1_MULT_ANCB_SRC11	INT_MASK_VIP1_MULT_ANCB_SRC10

Bits	Field Name	Description	Type	Reset
31	INT_MASK_VIP2_MULT_PORTB_SRC3	The interrupt for Video Input 2 Port B Channel 3 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
30	INT_MASK_VIP2_MULT_PORTB_SRC2	The interrupt for Video Input 2 Port B Channel 2 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
29	INT_MASK_VIP2_MULT_PORTB_SRC1	The interrupt for Video Input 2 Port B Channel 1 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
28	INT_MASK_VIP2_MULT_PORTB_SRC0	The interrupt for Video Input 2 Port B Channel 0 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
27	INT_MASK_VIP2_MULT_PORTA_SRC15	The interrupt for Video Input 2 Port A Channel 15 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
26	INT_MASK_VIP2_MULT_PORTA_SRC14	The interrupt for Video Input 2 Port A Channel 14 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
25	INT_MASK_VIP2_MULT_PORTA_SRC13	The interrupt for Video Input 2 Port A Channel 13 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
24	INT_MASK_VIP2_MULT_PORTA_SRC12	The interrupt for Video Input 2 Port A Channel 12 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
23	INT_MASK_VIP2_MULT_PORTA_SRC11	The interrupt for Video Input 2 Port A Channel 11 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
22	INT_MASK_VIP2_MULT_PORTA_SRC10	The interrupt for Video Input 2 Port A Channel 10 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
21	INT_MASK_VIP2_MULT_PORTA_SRC9	The interrupt for Video Input 2 Port A Channel 9 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
20	INT_MASK_VIP2_MULT_PORTA_SRC8	The interrupt for Video Input 2 Port A Channel 8 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
19	INT_MASK_VIP2_MULT_PORTA_SRC7	The interrupt for Video Input 2 Port A Channel 7 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
18	INT_MASK_VIP2_MULT_PORTA_SRC6	The interrupt for Video Input 2 Port A Channel 6 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
17	INT_MASK_VIP2_MULT_PORTA_SRC5	The interrupt for Video Input 2 Port A Channel 5 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
16	INT_MASK_VIP2_MULT_PORTA_SRC4	The interrupt for Video Input 2 Port A Channel 4 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
15	INT_MASK_VIP2_MULT_PORTA_SRC3	The interrupt for Video Input 2 Port A Channel 3 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
14	INT_MASK_VIP2_MULT_PORTA_SRC2	The interrupt for Video Input 2 Port A Channel 2 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
13	INT_MASK_VIP2_MULT_PORTA_SRC1	The interrupt for Video Input 2 Port A Channel 1 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
12	INT_MASK_VIP2_MULT_PORTA_SRC0	The interrupt for Video Input 2 Port A Channel 0 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0

Bits	Field Name	Description	Type	Reset
11	INT_MASK_VIP1_PORTB_RGB	The interrupt for Video Input 1 Port B RGB Data should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
10	INT_MASK_VIP1_PORTA_RGB	The interrupt for Video Input 1 Port A RGB Data should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
9	INT_MASK_VIP1_PORTB_CHROMA	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
8	INT_MASK_VIP1_PORTB_LUMA	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
7	INT_MASK_VIP1_PORTA_CHROMA	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
6	INT_MASK_VIP1_PORTA_LUMA	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
5	INT_MASK_VIP1_MULT_ANCB_SRC15	The interrupt for Video Input 1 Port B Ancillary Data Channel 15 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
4	INT_MASK_VIP1_MULT_ANCB_SRC14	The interrupt for Video Input 1 Port B Ancillary Data Channel 14 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
3	INT_MASK_VIP1_MULT_ANCB_SRC13	The interrupt for Video Input 1 Port B Ancillary Data Channel 13 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
2	INT_MASK_VIP1_MULT_ANCB_SRC12	The interrupt for Video Input 1 Port B Ancillary Data Channel 12 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
1	INT_MASK_VIP1_MULT_ANCB_SRC11	The interrupt for Video Input 1 Port B Ancillary Data Channel 11 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
0	INT_MASK_VIP1_MULT_ANCB_SRC10	The interrupt for Video Input 1 Port B Ancillary Data Channel 10 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0

Table 7-360. Register Call Summary for Register VIP_INT1_CHANNEL3_INT_MASK

VIP Functional Description
- VPDMA Interrupts: [0]

VIP Register Manual
- VIP VPDMA Register Summary: [3]

Table 7-361. VIP_INT1_CHANNEL4_INT_STAT

Address Offset	0x0000 00B0		
Physical Address	0x4897 D0B0	**Instance**	VIP_VPDMA
Description	This register gives the information of the interrupts that have triggered since last cleared by the process that is servicing vpdma_int1.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INT_STAT_VIP2_MULT_ANCB_SRC3	INT_STAT_VIP2_MULT_ANCB_SRC2	INT_STAT_VIP2_MULT_ANCB_SRC1	INT_STAT_VIP2_MULT_ANCB_SRC0	INT_STAT_VIP2_MULT_ANCA_SRC15	INT_STAT_VIP2_MULT_ANCA_SRC14	INT_STAT_VIP2_MULT_ANCA_SRC13	INT_STAT_VIP2_MULT_ANCA_SRC12	INT_STAT_VIP2_MULT_ANCA_SRC11	INT_STAT_VIP2_MULT_ANCA_SRC10	INT_STAT_VIP2_MULT_ANCA_SRC9	INT_STAT_VIP2_MULT_ANCA_SRC8	INT_STAT_VIP2_MULT_ANCA_SRC7	INT_STAT_VIP2_MULT_ANCA_SRC6	INT_STAT_VIP2_MULT_ANCA_SRC5	INT_STAT_VIP2_MULT_ANCA_SRC4	INT_STAT_VIP2_MULT_ANCA_SRC3	INT_STAT_VIP2_MULT_ANCA_SRC2	INT_STAT_VIP2_MULT_ANCA_SRC1	INT_STAT_VIP2_MULT_ANCA_SRC0	INT_STAT_VIP2_MULT_PORTB_SRC15	INT_STAT_VIP2_MULT_PORTB_SRC14	INT_STAT_VIP2_MULT_PORTB_SRC13	INT_STAT_VIP2_MULT_PORTB_SRC12	INT_STAT_VIP2_MULT_PORTB_SRC11	INT_STAT_VIP2_MULT_PORTB_SRC10	INT_STAT_VIP2_MULT_PORTB_SRC9	INT_STAT_VIP2_MULT_PORTB_SRC8	INT_STAT_VIP2_MULT_PORTB_SRC7	INT_STAT_VIP2_MULT_PORTB_SRC6	INT_STAT_VIP2_MULT_PORTB_SRC5	INT_STAT_VIP2_MULT_PORTB_SRC4

Bits	Field Name	Description	Type	Reset
31	INT_STAT_VIP2_MULT_ANCB_SRC3	The last write DMA transaction has completed for channel vip2_mult_ancb_src3. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
30	INT_STAT_VIP2_MULT_ANCB_SRC2	The last write DMA transaction has completed for channel vip2_mult_ancb_src2. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
29	INT_STAT_VIP2_MULT_ANCB_SRC1	The last write DMA transaction has completed for channel vip2_mult_ancb_src1. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
28	INT_STAT_VIP2_MULT_ANCB_SRC0	The last write DMA transaction has completed for channel vip2_mult_ancb_src0. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
27	INT_STAT_VIP2_MULT_ANCA_SRC15	The last write DMA transaction has completed for channel vip2_mult_anca_src15. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
26	INT_STAT_VIP2_MULT_ANCA_SRC14	The last write DMA transaction has completed for channel vip2_mult_anca_src14. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
25	INT_STAT_VIP2_MULT_ANCA_SRC13	The last write DMA transaction has completed for channel vip2_mult_anca_src13. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
24	INT_STAT_VIP2_MULT_ANCA_SRC12	The last write DMA transaction has completed for channel vip2_mult_anca_src12. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
23	INT_STAT_VIP2_MULT_ANCA_SRC11	The last write DMA transaction has completed for channel vip2_mult_anca_src11. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
22	INT_STAT_VIP2_MULT_ANCA_SRC10	The last write DMA transaction has completed for channel vip2_mult_anca_src10. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
21	INT_STAT_VIP2_MULT_ANCA_SRC9	The last write DMA transaction has completed for channel vip2_mult_anca_src9. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
20	INT_STAT_VIP2_MULT_ANCA_SRC8	The last write DMA transaction has completed for channel vip2_mult_anca_src8. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
19	INT_STAT_VIP2_MULT_ANCA_SRC7	The last write DMA transaction has completed for channel vip2_mult_anca_src7. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
18	INT_STAT_VIP2_MULT_ANCA_SRC6	The last write DMA transaction has completed for channel vip2_mult_anca_src6. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
17	INT_STAT_VIP2_MULT_ANCA_SRC5	The last write DMA transaction has completed for channel vip2_mult_anca_src5. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
16	INT_STAT_VIP2_MULT_ANCA_SRC4	The last write DMA transaction has completed for channel vip2_mult_anca_src4. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
15	INT_STAT_VIP2_MULT_ANCA_SRC3	The last write DMA transaction has completed for channel vip2_mult_anca_src3. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
14	INT_STAT_VIP2_MULT_ANCA_SRC2	The last write DMA transaction has completed for channel vip2_mult_anca_src2. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
13	INT_STAT_VIP2_MULT_ANCA_SRC1	The last write DMA transaction has completed for channel vip2_mult_anca_src1. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
12	INT_STAT_VIP2_MULT_ANCA_SRC0	The last write DMA transaction has completed for channel vip2_mult_anca_src0. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_a then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
11	INT_STAT_VIP2_MULT_PORTB_SRC15	The last write DMA transaction has completed for channel vip2_mult_portb_src15. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
10	INT_STAT_VIP2_MULT_PORTB_SRC14	The last write DMA transaction has completed for channel vip2_mult_portb_src14. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
9	INT_STAT_VIP2_MULT_PORTB_SRC13	The last write DMA transaction has completed for channel vip2_mult_portb_src13. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
8	INT_STAT_VIP2_MULT_PORTB_SRC12	The last write DMA transaction has completed for channel vip2_mult_portb_src12. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
7	INT_STAT_VIP2_MULT_PORTB_SRC11	The last write DMA transaction has completed for channel vip2_mult_portb_src11. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
6	INT_STAT_VIP2_MULT_PORTB_SRC10	The last write DMA transaction has completed for channel vip2_mult_portb_src10. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
5	INT_STAT_VIP2_MULT_PORTB_SRC9	The last write DMA transaction has completed for channel vip2_mult_portb_src9. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
4	INT_STAT_VIP2_MULT_PORTB_SRC8	The last write DMA transaction has completed for channel vip2_mult_portb_src8. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
3	INT_STAT_VIP2_MULT_PORTB_SRC7	The last write DMA transaction has completed for channel vip2_mult_portb_src7. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
2	INT_STAT_VIP2_MULT_PORTB_SRC6	The last write DMA transaction has completed for channel vip2_mult_portb_src6. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
1	INT_STAT_VIP2_MULT_PORTB_SRC5	The last write DMA transaction has completed for channel vip2_mult_portb_src5. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
0	INT_STAT_VIP2_MULT_PORTB_SRC4	The last write DMA transaction has completed for channel vip2_mult_portb_src4. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_uv then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Table 7-362. Register Call Summary for Register VIP_INT1_CHANNEL4_INT_STAT

VIP Functional Description
- VPDMA Interrupts: [0]

VIP Register Manual
- VIP VPDMA Register Summary: [3]

Table 7-363. VIP_INT1_CHANNEL4_INT_MASK

Address Offset	0x0000 00B4		
Physical Address	0x4897 D0B4	**Instance**	VIP_VPDMA
Description	The register gives the information of the interrupts that should be masked and not generate an interrupt for vpdma_int1.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INT_MASK_VIP2_MULT_ANCB_SRC3	INT_MASK_VIP2_MULT_ANCB_SRC2	INT_MASK_VIP2_MULT_ANCB_SRC1	INT_MASK_VIP2_MULT_ANCB_SRC0	INT_MASK_VIP2_MULT_ANCA_SRC15	INT_MASK_VIP2_MULT_ANCA_SRC14	INT_MASK_VIP2_MULT_ANCA_SRC13	INT_MASK_VIP2_MULT_ANCA_SRC12	INT_MASK_VIP2_MULT_ANCA_SRC11	INT_MASK_VIP2_MULT_ANCA_SRC10	INT_MASK_VIP2_MULT_ANCA_SRC9	INT_MASK_VIP2_MULT_ANCA_SRC8	INT_MASK_VIP2_MULT_ANCA_SRC7	INT_MASK_VIP2_MULT_ANCA_SRC6	INT_MASK_VIP2_MULT_ANCA_SRC5	INT_MASK_VIP2_MULT_ANCA_SRC4	INT_MASK_VIP2_MULT_ANCA_SRC3	INT_MASK_VIP2_MULT_ANCA_SRC2	INT_MASK_VIP2_MULT_ANCA_SRC1	INT_MASK_VIP2_MULT_ANCA_SRC0	INT_MASK_VIP2_MULT_PORTB_SRC15	INT_MASK_VIP2_MULT_PORTB_SRC14	INT_MASK_VIP2_MULT_PORTB_SRC13	INT_MASK_VIP2_MULT_PORTB_SRC12	INT_MASK_VIP2_MULT_PORTB_SRC11	INT_MASK_VIP2_MULT_PORTB_SRC10	INT_MASK_VIP2_MULT_PORTB_SRC9	INT_MASK_VIP2_MULT_PORTB_SRC8	INT_MASK_VIP2_MULT_PORTB_SRC7	INT_MASK_VIP2_MULT_PORTB_SRC6	INT_MASK_VIP2_MULT_PORTB_SRC5	INT_MASK_VIP2_MULT_PORTB_SRC4

Bits	Field Name	Description	Type	Reset
31	INT_MASK_VIP2_MULT_ANCB_SRC3	The interrupt for Video Input 2 Port B Ancillary Data Channel 3 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
30	INT_MASK_VIP2_MULT_ANCB_SRC2	The interrupt for Video Input 2 Port B Ancillary Data Channel 2 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
29	INT_MASK_VIP2_MULT_ANCB_SRC1	The interrupt for Video Input 2 Port B Ancillary Data Channel 1 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
28	INT_MASK_VIP2_MULT_ANCB_SRC0	The interrupt for Video Input 2 Port B Ancillary Data Channel 0 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
27	INT_MASK_VIP2_MULT_ANCA_SRC15	The interrupt for Video Input 2 Port A Ancillary Data Channel 15 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
26	INT_MASK_VIP2_MULT_ANCA_SRC14	The interrupt for Video Input 2 Port A Ancillary Data Channel 14 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
25	INT_MASK_VIP2_MULT_ANCA_SRC13	The interrupt for Video Input 2 Port A Ancillary Data Channel 13 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
24	INT_MASK_VIP2_MULT_ANCA_SRC12	The interrupt for Video Input 2 Port A Ancillary Data Channel 12 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
23	INT_MASK_VIP2_MULT_ANCA_SRC11	The interrupt for Video Input 2 Port A Ancillary Data Channel 11 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
22	INT_MASK_VIP2_MULT_ANCA_SRC10	The interrupt for Video Input 2 Port A Ancillary Data Channel 10 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
21	INT_MASK_VIP2_MULT_ANCA_SRC9	The interrupt for Video Input 2 Port A Ancillary Data Channel 9 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0

Bits	Field Name	Description	Type	Reset
20	INT_MASK_VIP2_MULT_ANCA_SRC8	The interrupt for Video Input 2 Port A Ancillary Data Channel 8 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
19	INT_MASK_VIP2_MULT_ANCA_SRC7	The interrupt for Video Input 2 Port A Ancillary Data Channel 7 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
18	INT_MASK_VIP2_MULT_ANCA_SRC6	The interrupt for Video Input 2 Port A Ancillary Data Channel 6 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
17	INT_MASK_VIP2_MULT_ANCA_SRC5	The interrupt for Video Input 2 Port A Ancillary Data Channel 5 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
16	INT_MASK_VIP2_MULT_ANCA_SRC4	The interrupt for Video Input 2 Port A Ancillary Data Channel 4 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
15	INT_MASK_VIP2_MULT_ANCA_SRC3	The interrupt for Video Input 2 Port A Ancillary Data Channel 3 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
14	INT_MASK_VIP2_MULT_ANCA_SRC2	The interrupt for Video Input 2 Port A Ancillary Data Channel 2 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
13	INT_MASK_VIP2_MULT_ANCA_SRC1	The interrupt for Video Input 2 Port A Ancillary Data Channel 1 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
12	INT_MASK_VIP2_MULT_ANCA_SRC0	The interrupt for Video Input 2 Port A Ancillary Data Channel 0 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
11	INT_MASK_VIP2_MULT_PORTB_SRC15	The interrupt for Video Input 2 Port B Channel 15 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
10	INT_MASK_VIP2_MULT_PORTB_SRC14	The interrupt for Video Input 2 Port B Channel 14 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
9	INT_MASK_VIP2_MULT_PORTB_SRC13	The interrupt for Video Input 2 Port B Channel 13 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
8	INT_MASK_VIP2_MULT_PORTB_SRC12	The interrupt for Video Input 2 Port B Channel 12 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
7	INT_MASK_VIP2_MULT_PORTB_SRC11	The interrupt for Video Input 2 Port B Channel 11 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
6	INT_MASK_VIP2_MULT_PORTB_SRC10	The interrupt for Video Input 2 Port B Channel 10 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
5	INT_MASK_VIP2_MULT_PORTB_SRC9	The interrupt for Video Input 2 Port B Channel 9 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
4	INT_MASK_VIP2_MULT_PORTB_SRC8	The interrupt for Video Input 2 Port B Channel 8 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
3	INT_MASK_VIP2_MULT_PORTB_SRC7	The interrupt for Video Input 2 Port B Channel 7 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0

Bits	Field Name	Description	Type	Reset
2	INT_MASK_VIP2_MULT_PORTB_SRC6	The interrupt for Video Input 2 Port B Channel 6 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
1	INT_MASK_VIP2_MULT_PORTB_SRC5	The interrupt for Video Input 2 Port B Channel 5 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
0	INT_MASK_VIP2_MULT_PORTB_SRC4	The interrupt for Video Input 2 Port B Channel 4 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0

Table 7-364. Register Call Summary for Register VIP_INT1_CHANNEL4_INT_MASK

VIP Functional Description
- VPDMA Interrupts: [0]

VIP Register Manual
- VIP VPDMA Register Summary: [3]

Table 7-365. VIP_INT1_CHANNEL5_INT_STAT

Address Offset	0x0000 00B8		
Physical Address	0x4897 D0B8	**Instance**	VIP_VPDMA
Description	This register gives the information of the interrupts that have triggered since last cleared by the process that is servicing vpdma_int1.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INT_STAT_TRANSCODE2_CHROMA	INT_STAT_TRANSCODE2_LUMA	INT_STAT_TRANSCODE1_CHROMA	INT_STAT_TRANSCODE1_LUMA	INT_STAT_AUX_IN	INT_STAT_PIP_FRAME	INT_STAT_POST_COMP_WR	INT_STAT_VBI_SD_VENC	RESERVED	INT_STAT_NF_LAST_CHROMA	INT_STAT_NF_LAST_LUMA	INT_STAT_NF_WRITE_CHROMA	INT_STAT_NF_WRITE_LUMA	INT_STAT_OTHER	INT_STAT_VIP2_PORTB_RGB	INT_STAT_VIP2_PORTA_RGB	INT_STAT_VIP2_PORTB_CHROMA	INT_STAT_VIP2_PORTB_LUMA	INT_STAT_VIP2_PORTA_CHROMA	INT_STAT_VIP2_PORTA_LUMA	INT_STAT_VIP2_MULT_ANCB_SRC15	INT_STAT_VIP2_MULT_ANCB_SRC14	INT_STAT_VIP2_MULT_ANCB_SRC13	INT_STAT_VIP2_MULT_ANCB_SRC12	INT_STAT_VIP2_MULT_ANCB_SRC11	INT_STAT_VIP2_MULT_ANCB_SRC10	INT_STAT_VIP2_MULT_ANCB_SRC9	INT_STAT_VIP2_MULT_ANCB_SRC8	INT_STAT_VIP2_MULT_ANCB_SRC7	INT_STAT_VIP2_MULT_ANCB_SRC6	INT_STAT_VIP2_MULT_ANCB_SRC5	INT_STAT_VIP2_MULT_ANCB_SRC4

Bits	Field Name	Description	Type	Reset
31	INT_STAT_TRANSCODE2_CHROMA	The last write DMA transaction has completed for channel transcode2_chroma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
30	INT_STAT_TRANSCODE2_LUMA	The last write DMA transaction has completed for channel transcode2_luma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
29	INT_STAT_TRANSCODE1_CHROMA	The last write DMA transaction has completed for channel transcode1_chroma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
28	INT_STAT_TRANSCODE1_LUMA	The last write DMA transaction has completed for channel transcode1_luma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
27	INT_STAT_AUX_IN	The last read DMA transaction has occurred for channel aux_in and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client comp_wrbk will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
26	INT_STAT_PIP_FRAME	The last read DMA transaction has occurred for channel pip_frame and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client pip_wrbk will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
25	INT_STAT_POST_COMP_WR	The last write DMA transaction has completed for channel post_comp_wr. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client hdmi_wrbk_out then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
24	INT_STAT_VBI_SD_VENC	The last read DMA transaction has occurred for channel vbi_sd_venc and the channel is free to be updated for the next transfer. This will fire before the destination has received the data as it will have just been stored in the internal buffer. The client vbi_sdvenc will now accept a new descriptor from the List Manager. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
23	RESERVED	Reserved	R	0x0
22	INT_STAT_NF_LAST_CHROMA	The last write DMA transaction has completed for channel nf_last_chroma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
21	INT_STAT_NF_LAST_LUMA	The last write DMA transaction has completed for channel nf_last_luma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
20	INT_STAT_NF_WRITE_CHROMA	The last write DMA transaction has completed for channel nf_write_chroma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
19	INT_STAT_NF_WRITE_LUMA	The last write DMA transaction has completed for channel nf_write_luma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
18	INT_STAT_OTHER	This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
17	INT_STAT_VIP2_PORTB_RGB	The last write DMA transaction has completed for channel vip2_portb_rgb. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_lo_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
16	INT_STAT_VIP2_PORTA_RGB	The last write DMA transaction has completed for channel vip2_porta_rgb. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_up_y then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
15	INT_STAT_VIP2_PORTB_CHROMA	The last write DMA transaction has completed for channel vip2_portb_chroma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
14	INT_STAT_VIP2_PORTB_LUMA	The last write DMA transaction has completed for channel vip2_portb_luma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
13	INT_STAT_VIP2_PORTA_CHROMA	The last write DMA transaction has completed for channel vip2_porta_chroma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
12	INT_STAT_VIP2_PORTA_LUMA	The last write DMA transaction has completed for channel vip2_porta_luma. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
11	INT_STAT_VIP2_MULT_ANCB_SRC15	The last write DMA transaction has completed for channel vip2_mult_ancb_src15. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
10	INT_STAT_VIP2_MULT_ANCB_SRC14	The last write DMA transaction has completed for channel vip2_mult_ancb_src14. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
9	INT_STAT_VIP2_MULT_ANCB_SRC13	The last write DMA transaction has completed for channel vip2_mult_ancb_src13. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
8	INT_STAT_VIP2_MULT_ANCB_SRC12	The last write DMA transaction has completed for channel vip2_mult_ancb_src12. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
7	INT_STAT_VIP2_MULT_ANCB_SRC11	The last write DMA transaction has completed for channel vip2_mult_ancb_src11. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
6	INT_STAT_VIP2_MULT_ANCB_SRC10	The last write DMA transaction has completed for channel vip2_mult_ancb_src10. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
5	INT_STAT_VIP2_MULT_ANCB_SRC9	The last write DMA transaction has completed for channel vip2_mult_ancb_src9. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
4	INT_STAT_VIP2_MULT_ANCB_SRC8	The last write DMA transaction has completed for channel vip2_mult_ancb_src8. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
3	INT_STAT_VIP2_MULT_ANCB_SRC7	The last write DMA transaction has completed for channel vip2_mult_ancb_src7. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
2	INT_STAT_VIP2_MULT_ANCB_SRC6	The last write DMA transaction has completed for channel vip2_mult_ancb_src6. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
1	INT_STAT_VIP2_MULT_ANCB_SRC5	The last write DMA transaction has completed for channel vip2_mult_ancb_src5. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
0	INT_STAT_VIP2_MULT_ANCB_SRC4	The last write DMA transaction has completed for channel vip2_mult_ancb_src4. All data from the channel has been sent and received by the external memory. If a new channel has not been setup for the client vip2_anc_b then the client will be fully empty at this point. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Table 7-366. Register Call Summary for Register VIP_INT1_CHANNEL5_INT_STAT

VIP Functional Description
- VPDMA Interrupts: [0]

VIP Register Manual
- VIP VPDMA Register Summary: [3]

Table 7-367. VIP_INT1_CHANNEL5_INT_MASK

Address Offset	0x0000 00BC		
Physical Address	0x4897 D0BC	**Instance**	VIP_VPDMA
Description	The register gives the information of the interrupts that should be masked and not generate an interrupt for vpdma_int1.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INT_MASK_TRANSCODE2_CHROMA	INT_MASK_TRANSCODE2_LUMA	INT_MASK_TRANSCODE1_CHROMA	INT_MASK_TRANSCODE1_LUMA	INT_MASK_AUX_IN	INT_MASK_PIP_FRAME	INT_MASK_POST_COMP_WR	INT_MASK_VBI_SD_VENC	RESERVED	INT_MASK_NF_LAST_CHROMA	INT_MASK_NF_LAST_LUMA	INT_MASK_NF_WRITE_CHROMA	INT_MASK_NF_WRITE_LUMA	INT_MASK_OTHER	INT_MASK_VIP2_PORTB_RGB	INT_MASK_VIP2_PORTA_RGB	INT_MASK_VIP2_PORTB_CHROMA	INT_MASK_VIP2_PORTB_LUMA	INT_MASK_VIP2_PORTA_CHROMA	INT_MASK_VIP2_PORTA_LUMA	INT_MASK_VIP2_MULT_ANCB_SRC15	INT_MASK_VIP2_MULT_ANCB_SRC14	INT_MASK_VIP2_MULT_ANCB_SRC13	INT_MASK_VIP2_MULT_ANCB_SRC12	INT_MASK_VIP2_MULT_ANCB_SRC11	INT_MASK_VIP2_MULT_ANCB_SRC10	INT_MASK_VIP2_MULT_ANCB_SRC9	INT_MASK_VIP2_MULT_ANCB_SRC8	INT_MASK_VIP2_MULT_ANCB_SRC7	INT_MASK_VIP2_MULT_ANCB_SRC6	INT_MASK_VIP2_MULT_ANCB_SRC5	INT_MASK_VIP2_MULT_ANCB_SRC4

Bits	Field Name	Description	Type	Reset
31	INT_MASK_TRANSCODE2_CHROMA	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
30	INT_MASK_TRANSCODE2_LUMA	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
29	INT_MASK_TRANSCODE1_CHROMA	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
28	INT_MASK_TRANSCODE1_LUMA	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
27	INT_MASK_AUX_IN	The interrupt for Auxilary Data for the Compositor Frame From Memory should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0

Bits	Field Name	Description	Type	Reset
26	INT_MASK_PIP_FRAME	The interrupt for PIP Data for the Compositor Frame From Memory should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
25	INT_MASK_POST_COMP_WR	The interrupt for Post Compositer Writeback to Memory should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
24	INT_MASK_VBI_SD_VENC	The interrupt for SD Video Encoder VBI Data should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
23	RESERVED	Reserved	R	0x0
22	INT_MASK_NF_LAST_CHROMA	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
21	INT_MASK_NF_LAST_LUMA	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
20	INT_MASK_NF_WRITE_CHROMA	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
19	INT_MASK_NF_WRITE_LUMA	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
18	INT_MASK_OTHER	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
17	INT_MASK_VIP2_PORTB_RGB	The interrupt for Video Input 2 Port B RGB Data should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
16	INT_MASK_VIP2_PORTA_RGB	The interrupt for Video Input 2 Port A RGB Data should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
15	INT_MASK_VIP2_PORTB_CHROMA	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
14	INT_MASK_VIP2_PORTB_LUMA	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
13	INT_MASK_VIP2_PORTA_CHROMA	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
12	INT_MASK_VIP2_PORTA_LUMA	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
11	INT_MASK_VIP2_MULT_ANCB_SRC15	The interrupt for Video Input 2 Port B Ancillary Data Channel 15 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
10	INT_MASK_VIP2_MULT_ANCB_SRC14	The interrupt for Video Input 2 Port B Ancillary Data Channel 14 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
9	INT_MASK_VIP2_MULT_ANCB_SRC13	The interrupt for Video Input 2 Port B Ancillary Data Channel 13 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
8	INT_MASK_VIP2_MULT_ANCB_SRC12	The interrupt for Video Input 2 Port B Ancillary Data Channel 12 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0

Bits	Field Name	Description	Type	Reset
7	INT_MASK_VIP2_MULT_ANCB_SRC11	The interrupt for Video Input 2 Port B Ancillary Data Channel 11 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
6	INT_MASK_VIP2_MULT_ANCB_SRC10	The interrupt for Video Input 2 Port B Ancillary Data Channel 10 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
5	INT_MASK_VIP2_MULT_ANCB_SRC9	The interrupt for Video Input 2 Port B Ancillary Data Channel 9 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
4	INT_MASK_VIP2_MULT_ANCB_SRC8	The interrupt for Video Input 2 Port B Ancillary Data Channel 8 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
3	INT_MASK_VIP2_MULT_ANCB_SRC7	The interrupt for Video Input 2 Port B Ancillary Data Channel 7 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
2	INT_MASK_VIP2_MULT_ANCB_SRC6	The interrupt for Video Input 2 Port B Ancillary Data Channel 6 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
1	INT_MASK_VIP2_MULT_ANCB_SRC5	The interrupt for Video Input 2 Port B Ancillary Data Channel 5 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
0	INT_MASK_VIP2_MULT_ANCB_SRC4	The interrupt for Video Input 2 Port B Ancillary Data Channel 4 should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0

Table 7-368. Register Call Summary for Register VIP_INT1_CHANNEL5_INT_MASK

VIP Functional Description • VPDMA Interrupts: [0]
VIP Register Manual • VIP VPDMA Register Summary: [3]

Table 7-369. VIP_INT1_CLIENT0_INT_STAT

Address Offset	0x0000 00C8		
Physical Address	0x4897 D0C8	**Instance**	VIP_VPDMA
Description	This register gives the information of the interrupts that have triggered since last cleared by the process that is servicing vpdma_int1.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INT_MASK_TRANSCODE2_CHROMA	INT_MASK_TRANSCODE2_LUMA	INT_MASK_TRANSCODE1_CHROMA	INT_MASK_TRANSCODE1_LUMA	INT_MASK_AUX_IN	INT_MASK_PIP_FRAME	INT_MASK_POST_COMP_WR	INT_MASK_VBI_SD_VENC	RESERVED	INT_MASK_NF_LAST_CHROMA	INT_MASK_NF_LAST_LUMA	INT_MASK_NF_WRITE_CHROMA	INT_MASK_NF_WRITE_LUMA	INT_MASK_NF_READ	INT_MASK_VIP2_PORTB_RGB	INT_MASK_VIP2_PORTA_RGB	INT_STAT_DEI_HQ_MV_OUT	RESERVED		INT_STAT_DEI_HQ_MV_IN	RESERVED					INT_STAT_DEI_HQ_3_CHROMA	INT_STAT_DEI_HQ_3_LUMA	INT_STAT_DEI_HQ_2_CHROMA	INT_STAT_DEI_HQ_2_LUMA	INT_STAT_DEI_HQ_1_LUMA	INT_STAT_DEI_HQ_1_CHROMA

Bits	Field Name	Description	Type	Reset
31	INT_MASK_TRANSCODE2_CHROMA	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
30	INT_MASK_TRANSCODE2_LUMA	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
29	INT_MASK_TRANSCODE1_CHROMA	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
28	INT_MASK_TRANSCODE1_LUMA	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
27	INT_MASK_AUX_IN	The interrupt for Auxilary Data for the Compositor Frame From Memory should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
26	INT_MASK_PIP_FRAME	The interrupt for PIP Data for the Compositor Frame From Memory should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
25	INT_MASK_POST_COMP_WR	The interrupt for Post Compositer Writeback to Memory should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
24	INT_MASK_VBI_SD_VENC	The interrupt for SD Video Encoder VBI Data should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
23	RESERVED	Reserved	R	0x0
22	INT_MASK_NF_LAST_CHROMA	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
21	INT_MASK_NF_LAST_LUMA	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
20	INT_MASK_NF_WRITE_CHROMA	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
19	INT_MASK_NF_WRITE_LUMA	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
18	INT_MASK_NF_READ	The interrupt for Noise Filter Input Data 422 Interleaved should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0

Bits	Field Name	Description	Type	Reset
17	INT_MASK_VIP2_PORTB_RGB	The interrupt for Video Input 2 Port B RGB Data should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
16	INT_MASK_VIP2_PORTA_RGB	The interrupt for Video Input 2 Port A RGB Data should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
15	INT_STAT_DEI_HQ_MV_OUT	The client interface dei_hq_mv_out has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
14:13	RESERVED	Reserved	R	0x0
12	INT_STAT_DEI_HQ_MV_IN	The client interface dei_hq_mv_in has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
11:6	RESERVED	Reserved	R	0x00
5	INT_STAT_DEI_HQ_3_CHROMA	The client interface dei_hq_3_chroma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
4	INT_STAT_DEI_HQ_3_LUMA	The client interface dei_hq_3_luma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
3	INT_STAT_DEI_HQ_2_CHROMA	The client interface dei_hq_2_chroma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
2	INT_STAT_DEI_HQ_2_LUMA	The client interface dei_hq_2_luma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
1	INT_STAT_DEI_HQ_1_LUMA	The client interface dei_hq_1_luma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
0	INT_STAT_DEI_HQ_1_CHROMA	The client interface dei_hq_1_chroma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Table 7-370. Register Call Summary for Register VIP_INT1_CLIENT0_INT_STAT

VIP Functional Description
- VPDMA Interrupts: [0]

VIP Register Manual
- VIP VPDMA Register Summary: [3]

Table 7-371. VIP_INT1_CLIENT0_INT_MASK

Address Offset	0x0000 00CC		
Physical Address	0x4897 D0CC	**Instance**	VIP_VPDMA
Description	The register gives the information of the interrupts that should be masked and not generate an interrupt for vpdma_int1.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INT_MASK_GRPX1_DATA	INT_MASK_COMP_WRBK	INT_MASK_SC_OUT	RESERVED								INT_MASK_SC_IN_LUMA	INT_MASK_SC_IN_CHROMA	INT_MASK_PIP_WRBK	INT_MASK_DEI_SC_OUT	RESERVED	INT_MASK_DEI_HQ_MV_OUT	RESERVED		INT_MASK_DEI_HQ_MV_IN	RESERVED						INT_MASK_DEI_HQ_3_CHROMA	INT_MASK_DEI_HQ_3_LUMA	INT_MASK_DEI_HQ_2_CHROMA	INT_MASK_DEI_HQ_2_LUMA	INT_MASK_DEI_HQ_1_LUMA	INT_MASK_DEI_HQ_1_CHROMA

Bits	Field Name	Description	Type	Reset
31	INT_MASK_GRPX1_DATA	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
30	INT_MASK_COMP_WRBK	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
29	INT_MASK_SC_OUT	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
28:21	RESERVED	Reserved	R	0x00
20	INT_MASK_SC_IN_LUMA	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
19	INT_MASK_SC_IN_CHROMA	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
18	INT_MASK_PIP_WRBK	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0

Bits	Field Name	Description	Type	Reset
17	INT_MASK_DEI_SC_OUT	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
16	RESERVED	Reserved	R	0x0
15	INT_MASK_DEI_HQ_MV_OUT	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
14:13	RESERVED	Reserved	R	0x0
12	INT_MASK_DEI_HQ_MV_IN	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
11:6	RESERVED	Reserved	R	0x00
5	INT_MASK_DEI_HQ_3_CHROMA	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
4	INT_MASK_DEI_HQ_3_LUMA	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
3	INT_MASK_DEI_HQ_2_CHROMA	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
2	INT_MASK_DEI_HQ_2_LUMA	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
1	INT_MASK_DEI_HQ_1_LUMA	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
0	INT_MASK_DEI_HQ_1_CHROMA	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0

Table 7-372. Register Call Summary for Register VIP_INT1_CLIENT0_INT_MASK

VIP Functional Description
- VPDMA Interrupts: [0]

VIP Register Manual
- VIP VPDMA Register Summary: [3]

Table 7-373. VIP_INT1_CLIENT1_INT_STAT

Address Offset	0x0000 00D0		
Physical Address	0x4897 D0D0	**Instance**	VIP_VPDMA
Description	This register gives the information of the interrupts that have triggered since last cleared by the process that is servicing vpdma_int1.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	RESERVED	INT_STAT_VIP2_ANC_B	INT_STAT_VIP2_ANC_A	INT_STAT_VIP1_ANC_B	INT_STAT_VIP1_ANC_A	INT_STAT_TRANS2_LUMA	INT_STAT_TRANS2_CHROMA	INT_STAT_TRANS1_LUMA	INT_STAT_TRANS1_CHROMA	INT_STAT_HDMI_WRBK_OUT	INT_STAT_VPI_CTL	INT_STAT_VBI_SDVENC	RESERVED	INT_STAT_NF_420_UV_OUT	INT_STAT_NF_420_Y_OUT	INT_STAT_NF_420_UV_IN	INT_STAT_NF_420_Y_IN	INT_STAT_NF_422_IN	INT_STAT_GRPX3_ST	INT_STAT_GRPX2_ST	INT_STAT_GRPX1_ST	INT_STAT_VIP2_UP_UV	INT_STAT_VIP2_UP_Y	INT_STAT_VIP2_LO_UV	INT_STAT_VIP2_LO_Y	INT_STAT_VIP1_UP_UV	INT_STAT_VIP1_UP_Y	INT_STAT_VIP1_LO_UV	INT_STAT_VIP1_LO_Y	INT_STAT_GRPX3_DATA	INT_STAT_GRPX2_DATA

Bits	Field Name	Description	Type	Reset
31	RESERVED	Reserved	R	0
30	RESERVED	Reserved	R	0
29	INT_STAT_VIP2_ANC_B	The client interface vip2_anc_b has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
28	INT_STAT_VIP2_ANC_A	The client interface vip2_anc_a has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
27	INT_STAT_VIP1_ANC_B	The client interface vip1_anc_b has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
26	INT_STAT_VIP1_ANC_A	The client interface vip1_anc_a has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
25	INT_STAT_TRANS2_LUMA	The client interface trans2_luma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
24	INT_STAT_TRANS2_CHROMA	The client interface trans2_chroma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
23	INT_STAT_TRANS1_LUMA	The client interface trans1_luma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
22	INT_STAT_TRANS1_CHROMA	The client interface trans1_chroma has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0

Bits	Field Name	Description	Type	Reset
21	INT_STAT_HDMI_WRBK_OUT	The client interface hdmi_wrbk_out has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
20	INT_STAT_VPI_CTL	The client interface vpi_ctl has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
19	INT_STAT_VBI_SDVENC	The client interface vbi_sdvenc has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
18	RESERVED	Reserved	R	0
17	INT_STAT_NF_420_UV_OUT	The client interface nf_420_uv_out has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
16	INT_STAT_NF_420_Y_OUT	The client interface nf_420_y_out has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
15	INT_STAT_NF_420_UV_IN	The client interface nf_420_uv_in has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
14	INT_STAT_NF_420_Y_IN	The client interface nf_420_y_in has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
13	INT_STAT_NF_422_IN	The client interface nf_422_in has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0

Bits	Field Name	Description	Type	Reset
12	INT_STAT_GRPX3_ST	The client interface grpx3_st has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
11	INT_STAT_GRPX2_ST	The client interface grpx2_st has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
10	INT_STAT_GRPX1_ST	The client interface grpx1_st has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
9	INT_STAT_VIP2_UP_UV	The client interface vip2_up_uv has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
8	INT_STAT_VIP2_UP_Y	The client interface vip2_up_y has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
7	INT_STAT_VIP2_LO_UV	The client interface vip2_lo_uv has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
6	INT_STAT_VIP2_LO_Y	The client interface vip2_lo_y has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
5	INT_STAT_VIP1_UP_UV	The client interface vip1_up_uv has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0

Bits	Field Name	Description	Type	Reset
4	INT_STAT_VIP1_UP_Y	The client interface vip1_up_y has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
3	INT_STAT_VIP1_LO_UV	The client interface vip1_lo_uv has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
2	INT_STAT_VIP1_LO_Y	The client interface vip1_lo_y has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having received the End of Frame signal from the transmitting module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
1	INT_STAT_GRPX3_DATA	The client interface grpx3_data has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0
0	INT_STAT_GRPX2_DATA	The client interface grpx2_data has reached its current configured interrupt event as specified by the last received control descriptor for this client. If no control descriptor has been configured this will default to having sent the End of Frame signal to the receiving module. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW W0toClr	0

Table 7-374. Register Call Summary for Register VIP_INT1_CLIENT1_INT_STAT

VIP Functional Description • VPDMA Interrupts: [0]
VIP Register Manual • VIP VPDMA Register Summary: [3]

Table 7-375. VIP_INT1_CLIENT1_INT_MASK

Address Offset	0x0000 00D4		
Physical Address	0x4897 D0D4	**Instance**	VIP_VPDMA
Description	The register gives the information of the interrupts that should be masked and not generate an interrupt for vpdma_int1.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	RESERVED	INT_MASK_VIP2_ANC_B	INT_MASK_VIP2_ANC_A	INT_MASK_VIP1_ANC_B	INT_MASK_VIP1_ANC_A	INT_MASK_TRANS2_LUMA	INT_MASK_TRANS2_CHROMA	INT_MASK_TRANS1_LUMA	INT_MASK_TRANS1_CHROMA	INT_MASK_HDMI_WRBK_OUT	INT_MASK_VPI_CTL	INT_MASK_VBI_SDVENC	RESERVED	INT_MASK_NF_420_UV_OUT	INT_MASK_NF_420_Y_OUT	INT_MASK_NF_420_UV_IN	INT_MASK_NF_420_Y_IN	INT_MASK_NF_422_IN	INT_MASK_GRPX3_ST	INT_MASK_GRPX2_ST	INT_MASK_GRPX1_ST	INT_MASK_VIP2_UP_UV	INT_MASK_VIP2_UP_Y	INT_MASK_VIP2_LO_UV	INT_MASK_VIP2_LO_Y	INT_MASK_VIP1_UP_UV	INT_MASK_VIP1_UP_Y	INT_MASK_VIP1_LO_UV	INT_MASK_VIP1_LO_Y	INT_MASK_GRPX3_DATA	INT_MASK_GRPX2_DATA

Bits	Field Name	Description	Type	Reset
31	RESERVED	Reserved	R	0
30	RESERVED	Reserved	R	0
29	INT_MASK_VIP2_ANC_B	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
28	INT_MASK_VIP2_ANC_A	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
27	INT_MASK_VIP1_ANC_B	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
26	INT_MASK_VIP1_ANC_A	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
25	INT_MASK_TRANS2_LUMA	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
24	INT_MASK_TRANS2_CHROMA	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
23	INT_MASK_TRANS1_LUMA	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
22	INT_MASK_TRANS1_CHROMA	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
21	INT_MASK_HDMI_WRBK_OUT	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
20	INT_MASK_VPI_CTL	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
19	INT_MASK_VBI_SDVENC	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
18	RESERVED	Reserved	R	0
17	INT_MASK_NF_420_UV_OUT	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
16	INT_MASK_NF_420_Y_OUT	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
15	INT_MASK_NF_420_UV_IN	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0

Bits	Field Name	Description	Type	Reset
14	INT_MASK_NF_420_Y_IN	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
13	INT_MASK_NF_422_IN	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
12	INT_MASK_GRPX3_ST	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
11	INT_MASK_GRPX2_ST	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
10	INT_MASK_GRPX1_ST	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
9	INT_MASK_VIP2_UP_UV	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
8	INT_MASK_VIP2_UP_Y	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
7	INT_MASK_VIP2_LO_UV	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
6	INT_MASK_VIP2_LO_Y	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
5	INT_MASK_VIP1_UP_UV	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
4	INT_MASK_VIP1_UP_Y	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
3	INT_MASK_VIP1_LO_UV	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
2	INT_MASK_VIP1_LO_Y	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
1	INT_MASK_GRPX3_DATA	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0
0	INT_MASK_GRPX2_DATA	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0

Table 7-376. Register Call Summary for Register VIP_INT1_CLIENT1_INT_MASK

VIP Functional Description • VPDMA Interrupts: [0]
VIP Register Manual • VIP VPDMA Register Summary: [3]

Table 7-377. VIP_INT1_LIST0_INT_STAT

Address Offset	0x0000 00D8		
Physical Address	0x4897 D0D8	**Instance**	VIP_VPDMA
Description	This register gives the information of the interrupts that have triggered since last cleared by the process that is servicing vpdma_int1.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INT_STAT_CONTROL_DESCRIPTOR_INT15	INT_STAT_CONTROL_DESCRIPTOR_INT14	INT_STAT_CONTROL_DESCRIPTOR_INT13	INT_STAT_CONTROL_DESCRIPTOR_INT12	INT_STAT_CONTROL_DESCRIPTOR_INT11	INT_STAT_CONTROL_DESCRIPTOR_INT10	INT_STAT_CONTROL_DESCRIPTOR_INT9	INT_STAT_CONTROL_DESCRIPTOR_INT8	INT_STAT_CONTROL_DESCRIPTOR_INT7	INT_STAT_CONTROL_DESCRIPTOR_INT6	INT_STAT_CONTROL_DESCRIPTOR_INT5	INT_STAT_CONTROL_DESCRIPTOR_INT4	INT_STAT_CONTROL_DESCRIPTOR_INT3	INT_STAT_CONTROL_DESCRIPTOR_INT2	INT_STAT_CONTROL_DESCRIPTOR_INT1	INT_STAT_CONTROL_DESCRIPTOR_INT0	INT_STAT_LIST7_NOTIFY	INT_STAT_LIST7_COMPLETE	INT_STAT_LIST6_NOTIFY	INT_STAT_LIST6_COMPLETE	INT_STAT_LIST5_NOTIFY	INT_STAT_LIST5_COMPLETE	INT_STAT_LIST4_NOTIFY	INT_STAT_LIST4_COMPLETE	INT_STAT_LIST3_NOTIFY	INT_STAT_LIST3_COMPLETE	INT_STAT_LIST2_NOTIFY	INT_STAT_LIST2_COMPLETE	INT_STAT_LIST1_NOTIFY	INT_STAT_LIST1_COMPLETE	INT_STAT_LIST0_NOTIFY	INT_STAT_LIST0_COMPLETE

Bits	Field Name	Description	Type	Reset
31	INT_STAT_CONTROL_DESCRIPTOR_INT15	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 15. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
30	INT_STAT_CONTROL_DESCRIPTOR_INT14	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 14. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
29	INT_STAT_CONTROL_DESCRIPTOR_INT13	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 13. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
28	INT_STAT_CONTROL_DESCRIPTOR_INT12	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 12. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
27	INT_STAT_CONTROL_DESCRIPTOR_INT11	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 11. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
26	INT_STAT_CONTROL_DESCRIPTOR_INT10	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 10. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
25	INT_STAT_CONTROL_DESCRIPTOR_INT9	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 9. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
24	INT_STAT_CONTROL_DESCRIPTOR_INT8	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 8. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
23	INT_STAT_CONTROL_DESCRIPTOR_INT7	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 7. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
22	INT_STAT_CONTROL_DESCRIPTOR_INT6	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 6. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
21	INT_STAT_CONTROL_DESCRIPTOR_INT5	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 5. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
20	INT_STAT_CONTROL_DESCRIPTOR_INT4	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 4. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
19	INT_STAT_CONTROL_DESCRIPTOR_INT3	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 3. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
18	INT_STAT_CONTROL_DESCRIPTOR_INT2	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 2. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
17	INT_STAT_CONTROL_DESCRIPTOR_INT1	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 1. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
16	INT_STAT_CONTROL_DESCRIPTOR_INT0	A Send Interrupt Control Descriptor has been received by the list manager with a source value of 0. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
15	INT_STAT_LIST7_NOTIFY	A channel set by List 7 has completed and the Notify bit had been set in the descriptor for that channel. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
14	INT_STAT_LIST7_COMPLETE	List 7 has completed and a new list can be loaded. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
13	INT_STAT_LIST6_NOTIFY	A channel set by List 6 has completed and the Notify bit had been set in the descriptor for that channel. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
12	INT_STAT_LIST6_COMPLETE	List 6 has completed and a new list can be loaded. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
11	INT_STAT_LIST5_NOTIFY	A channel set by List 5 has completed and the Notify bit had been set in the descriptor for that channel. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
10	INT_STAT_LIST5_COMPLETE	List 5 has completed and a new list can be loaded. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Bits	Field Name	Description	Type	Reset
9	INT_STAT_LIST4_NOTIFY	A channel set by List 4 has completed and the Notify bit had been set in the descriptor for that channel. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
8	INT_STAT_LIST4_COMPLETE	List 4 has completed and a new list can be loaded. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
7	INT_STAT_LIST3_NOTIFY	A channel set by List 3 has completed and the Notify bit had been set in the descriptor for that channel. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
6	INT_STAT_LIST3_COMPLETE	List 3 has completed and a new list can be loaded. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
5	INT_STAT_LIST2_NOTIFY	A channel set by List 2 has completed and the Notify bit had been set in the descriptor for that channel. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
4	INT_STAT_LIST2_COMPLETE	List 2 has completed and a new list can be loaded. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
3	INT_STAT_LIST1_NOTIFY	A channel set by List 1 has completed and the Notify bit had been set in the descriptor for that channel. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
2	INT_STAT_LIST1_COMPLETE	List 1 has completed and a new list can be loaded. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
1	INT_STAT_LIST0_NOTIFY	A channel set by List 0 has completed and the Notify bit had been set in the descriptor for that channel. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0
0	INT_STAT_LIST0_COMPLETE	List 0 has completed and a new list can be loaded. This event will cause a one to be set in this register until cleared by software. Write a 1 to this field to clear the value.	RW	0x0

Table 7-378. Register Call Summary for Register VIP_INT1_LIST0_INT_STAT

VIP Functional Description • VPDMA Interrupts: [0][1][2]
VIP Register Manual • VIP VPDMA Register Summary: [5]

Table 7-379. VIP_INT1_LIST0_INT_MASK

Address Offset	0x0000 00DC		
Physical Address	0x4897 D0DC	**Instance**	VIP_VPDMA
Description	The register gives the information of the interrupts that should be masked and not generate an interrupt for vpdma_int1.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INT_MASK_CONTROL_DESCRIPTOR_INT15	INT_MASK_CONTROL_DESCRIPTOR_INT14	INT_MASK_CONTROL_DESCRIPTOR_INT13	INT_MASK_CONTROL_DESCRIPTOR_INT12	INT_MASK_CONTROL_DESCRIPTOR_INT11	INT_MASK_CONTROL_DESCRIPTOR_INT10	INT_MASK_CONTROL_DESCRIPTOR_INT9	INT_MASK_CONTROL_DESCRIPTOR_INT8	INT_MASK_CONTROL_DESCRIPTOR_INT7	INT_MASK_CONTROL_DESCRIPTOR_INT6	INT_MASK_CONTROL_DESCRIPTOR_INT5	INT_MASK_CONTROL_DESCRIPTOR_INT4	INT_MASK_CONTROL_DESCRIPTOR_INT3	INT_MASK_CONTROL_DESCRIPTOR_INT2	INT_MASK_CONTROL_DESCRIPTOR_INT1	INT_MASK_CONTROL_DESCRIPTOR_INT0	INT_MASK_LIST7_NOTIFY	INT_MASK_LIST7_COMPLETE	INT_MASK_LIST6_NOTIFY	INT_MASK_LIST6_COMPLETE	INT_MASK_LIST5_NOTIFY	INT_MASK_LIST5_COMPLETE	INT_MASK_LIST4_NOTIFY	INT_MASK_LIST4_COMPLETE	INT_MASK_LIST3_NOTIFY	INT_MASK_LIST3_COMPLETE	INT_MASK_LIST2_NOTIFY	INT_MASK_LIST2_COMPLETE	INT_MASK_LIST1_NOTIFY	INT_MASK_LIST1_COMPLETE	INT_MASK_LIST0_NOTIFY	INT_MASK_LIST0_COMPLETE

Bits	Field Name	Description	Type	Reset
31	INT_MASK_CONTROL_DESCRIPTOR_INT15	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
30	INT_MASK_CONTROL_DESCRIPTOR_INT14	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
29	INT_MASK_CONTROL_DESCRIPTOR_INT13	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
28	INT_MASK_CONTROL_DESCRIPTOR_INT12	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
27	INT_MASK_CONTROL_DESCRIPTOR_INT11	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
26	INT_MASK_CONTROL_DESCRIPTOR_INT10	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
25	INT_MASK_CONTROL_DESCRIPTOR_INT9	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
24	INT_MASK_CONTROL_DESCRIPTOR_INT8	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
23	INT_MASK_CONTROL_DESCRIPTOR_INT7	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
22	INT_MASK_CONTROL_DESCRIPTOR_INT6	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
21	INT_MASK_CONTROL_DESCRIPTOR_INT5	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0

Bits	Field Name	Description	Type	Reset
20	INT_MASK_CONTROL_DESCRIPTOR_INT4	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
19	INT_MASK_CONTROL_DESCRIPTOR_INT3	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
18	INT_MASK_CONTROL_DESCRIPTOR_INT2	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
17	INT_MASK_CONTROL_DESCRIPTOR_INT1	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
16	INT_MASK_CONTROL_DESCRIPTOR_INT0	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
15	INT_MASK_LIST7_NOTIFY	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
14	INT_MASK_LIST7_COMPLETE	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
13	INT_MASK_LIST6_NOTIFY	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
12	INT_MASK_LIST6_COMPLETE	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
11	INT_MASK_LIST5_NOTIFY	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
10	INT_MASK_LIST5_COMPLETE	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
9	INT_MASK_LIST4_NOTIFY	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
8	INT_MASK_LIST4_COMPLETE	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
7	INT_MASK_LIST3_NOTIFY	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
6	INT_MASK_LIST3_COMPLETE	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
5	INT_MASK_LIST2_NOTIFY	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
4	INT_MASK_LIST2_COMPLETE	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
3	INT_MASK_LIST1_NOTIFY	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
2	INT_MASK_LIST1_COMPLETE	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0
1	INT_MASK_LIST0_NOTIFY	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0

Bits	Field Name	Description	Type	Reset
0	INT_MASK_LIST0_COMPLETE	The interrupt for should generate an interrupt on interrupt vpdma_int1. Write a 1 for the interrupt event to trigger the interrupt signal.	RW	0x0

Table 7-380. Register Call Summary for Register VIP_INT1_LIST0_INT_MASK

VIP Functional Description • VPDMA Interrupts: [0]
VIP Register Manual • VIP VPDMA Register Summary: [3]

Table 7-381. VIP_PERF_MON0

Address Offset	0x0000 0200		
Physical Address	0x4897 D200	**Instance**	VIP_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA\\n		
Type	RW		

31 30	29 28	27	26 25 24	23 22	21 20	19	18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT	

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0: 1: vip2_anc_b 2: 3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0: 1: vip2_anc_b 2: 3:	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 7-382. Register Call Summary for Register VIP_PERF_MON0

VIP Register Manual • VIP VPDMA Register Summary: [2]

Table 7-383. VIP_PERF_MON1

Address Offset	0x0000 0204		
Physical Address	0x4897 D204	**Instance**	VIP_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA\\n		
Type	RW		

31 30	29 28	27	26 25 24	23 22	21 20	19	18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT	

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0: 1: 2: 3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0: 1: 2: 3:	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 7-384. Register Call Summary for Register VIP_PERF_MON1

VIP Register Manual

- VIP VPDMA Register Summary: [2]

Table 7-385. VIP_PERF_MON2

Address Offset	0x0000 0208		
Physical Address	0x4897 D208	**Instance**	VIP_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA\\n		
Type	RW		

31 30	29 28	27	26 25 24	23 22	21 20	19	18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT	

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0: 1: 2: 3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0: 1: 2: 3:	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 7-386. Register Call Summary for Register VIP_PERF_MON2

VIP Register Manual

- VIP VPDMA Register Summary: [2]

Table 7-387. VIP_PERF_MON3

Address Offset	0x0000 020C		
Physical Address	0x4897 D20C	**Instance**	VIP_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA\\n		
Type	RW		

31 30	29 28	27	26 25 24	23 22	21 20	19	18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT	

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0

Bits	Field Name	Description	Type	Reset
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0: 1: 2: 3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0: 1: 2: 3:	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 7-388. Register Call Summary for Register VIP_PERF_MON3

VIP Register Manual
- VIP VPDMA Register Summary: [2]

Table 7-389. VIP_PERF_MON4

Address Offset	0x0000 0210		
Physical Address	0x4897 D210	**Instance**	VIP_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA\\n		
Type	RW		

31 30	29 28	27	26 25 24	23 22	21 20	19	18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT	

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0: 1: 2: 3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0: 1: 2: 3:	RW	0x0
19	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 7-390. Register Call Summary for Register VIP_PERF_MON4

VIP Register Manual
- VIP VPDMA Register Summary: [2]

Table 7-391. VIP_PERF_MON5

Address Offset	0x0000 0214		
Physical Address	0x4897 D214	**Instance**	VIP_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA\\n		
Type	RW		

31 30	29 28	27	26 25 24	23 22	21 20	19	18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT	

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0: 1: 2: 3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0: 1: 2: 3:	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 7-392. Register Call Summary for Register VIP_PERF_MON5

VIP Register Manual
- VIP VPDMA Register Summary: [2]

Table 7-393. VIP_PERF_MON6

Address Offset	0x0000 0218		
Physical Address	0x4897 D218	**Instance**	VIP_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA\\n		
Type	RW		

31 30	29 28	27	26 25 24	23 22	21 20	19	18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT	

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0: 1: 2: 3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0: 1: 2: 3:	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 7-394. Register Call Summary for Register VIP_PERF_MON6

VIP Register Manual

- VIP VPDMA Register Summary: [2]

Table 7-395. VIP_PERF_MON7

Address Offset	0x0000 021C		
Physical Address	0x4897 D21C	**Instance**	VIP_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA\\n		
Type	RW		

31 30	29 28	27	26 25 24	23 22	21 20	19	18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT	

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0: 1: 2: 3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0: 1: 2: 3:	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 7-396. Register Call Summary for Register VIP_PERF_MON7

VIP Register Manual

- VIP VPDMA Register Summary: [2]

Table 7-397. VIP_PERF_MON8

Address Offset	0x0000 0220		
Physical Address	0x4897 D220	**Instance**	VIP_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA\\n		
Type	RW		

31 30	29 28	27	26 25 24	23 22	21 20	19	18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT	

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0

Bits	Field Name	Description	Type	Reset
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0: 1: 2: 3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0: 1: 2: 3:	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 7-398. Register Call Summary for Register VIP_PERF_MON8

VIP Register Manual
- VIP VPDMA Register Summary: [2]

Table 7-399. VIP_PERF_MON9

Address Offset	0x0000 0224		
Physical Address	0x4897 D224	**Instance**	VIP_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA\\n		
Type	RW		

31 30	29 28	27	26 25 24	23 22	21 20	19	18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT	

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0: 1: 2: 3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0: 1: 2: 3:	RW	0x0
19	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 7-400. Register Call Summary for Register VIP_PERF_MON9

VIP Register Manual
- VIP VPDMA Register Summary: [2]

Table 7-401. VIP_PERF_MON10

Address Offset	0x0000 0228		
Physical Address	0x4897 D228	**Instance**	VIP_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA\\n		
Type	RW		

31 30	29 28	27	26 25 24	23 22	21 20	19	18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT	

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0: 1: 2: 3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0: 1: 2: 3:	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 7-402. Register Call Summary for Register VIP_PERF_MON10

VIP Register Manual
- VIP VPDMA Register Summary: [2]

Table 7-403. VIP_PERF_MON11

Address Offset	0x0000 022C		
Physical Address	0x4897 D22C	**Instance**	VIP_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA\\n		
Type	RW		

31 30	29 28	27	26 25 24	23 22	21 20	19	18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT	

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0: 1: 2: 3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0: 1: 2: 3:	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 7-404. Register Call Summary for Register VIP_PERF_MON11

VIP Register Manual

- VIP VPDMA Register Summary: [2]

Table 7-405. VIP_PERF_MON12

Address Offset	0x0000 0230		
Physical Address	0x4897 D230	**Instance**	VIP_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA\\n		
Type	RW		

31 30	29 28	27	26 25 24	23 22	21 20	19	18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT	

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0: 1: 2: 3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0: 1: 2: 3:	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 7-406. Register Call Summary for Register VIP_PERF_MON12

VIP Register Manual

- VIP VPDMA Register Summary: [2]

Table 7-407. VIP_PERF_MON13

Address Offset	0x0000 0234		
Physical Address	0x4897 D234	**Instance**	VIP_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA\\n		
Type	RW		

31 30	29 28	27	26 25 24	23 22	21 20	19	18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT	

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0

Bits	Field Name	Description	Type	Reset
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0: 1: 2: 3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0: 1: 2: 3:	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 7-408. Register Call Summary for Register VIP_PERF_MON13

VIP Register Manual
- VIP VPDMA Register Summary: [2]

Table 7-409. VIP_PERF_MON14

Address Offset	0x0000 0238		
Physical Address	0x4897 D238	**Instance**	VIP_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA\\n		
Type	RW		

31 30	29 28	27	26 25 24	23 22	21 20	19	18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT	

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0: 1: 2: 3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0: 1: 2: 3:	RW	0x0
19	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 7-410. Register Call Summary for Register VIP_PERF_MON14

VIP Register Manual
- VIP VPDMA Register Summary: [2]

Table 7-411. VIP_PERF_MON15

Address Offset	0x0000 023C		
Physical Address	0x4897 D23C	**Instance**	VIP_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA\\n		
Type	RW		

31 30	29 28	27	26 25 24	23 22	21 20	19	18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT	

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0: 1: 2: 3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0: 1: 2: 3:	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 7-412. Register Call Summary for Register VIP_PERF_MON15

VIP Register Manual
- VIP VPDMA Register Summary: [2]

Table 7-413. VIP_PERF_MON16

Address Offset	0x0000 0240		
Physical Address	0x4897 D240	**Instance**	VIP_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA\\n		
Type	RW		

31 30	29 28	27	26 25 24	23 22	21 20	19	18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT	

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0: 1: 2: 3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0: 1: 2: 3:	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 7-414. Register Call Summary for Register VIP_PERF_MON16

VIP Register Manual

- VIP VPDMA Register Summary: [2]

Table 7-415. VIP_PERF_MON17

Address Offset	0x0000 0244		
Physical Address	0x4897 D244	**Instance**	VIP_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA\\n		
Type	RW		

31 30	29 28	27	26 25 24	23 22	21 20	19	18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT	

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0: 1: 2: 3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0: 1: 2: 3:	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 7-416. Register Call Summary for Register VIP_PERF_MON17

VIP Register Manual

- VIP VPDMA Register Summary: [2]

Table 7-417. VIP_PERF_MON18

Address Offset	0x0000 0248		
Physical Address	0x4897 D248	**Instance**	VIP_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA\\n		
Type	RW		

31 30	29 28	27	26 25 24	23 22	21 20	19	18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT	

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0

Bits	Field Name	Description	Type	Reset
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0: 1: 2: 3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0: 1: 2: 3:	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 7-418. Register Call Summary for Register VIP_PERF_MON18

VIP Register Manual
- VIP VPDMA Register Summary: [2]

Table 7-419. VIP_PERF_MON19

Address Offset	0x0000 024C		
Physical Address	0x4897 D24C	**Instance**	VIP_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA\\n		
Type	RW		

31 30	29 28	27	26 25 24	23 22	21 20	19	18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT	

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0: 1: 2: 3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0: 1: 2: 3:	RW	0x0
19	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 7-420. Register Call Summary for Register VIP_PERF_MON19

VIP Register Manual
- VIP VPDMA Register Summary: [2]

Table 7-421. VIP_PERF_MON20

Address Offset	0x0000 0250		
Physical Address	0x4897 D250	**Instance**	VIP_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA\\n		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CAPTURE_MODE		STOP_CLIENT		RESERVED		STOP_COUNT		RESERVED		START_CLIENT		RESERVED		START_COUNT		CURR_COUNT															

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0: 1: 2: 3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0: 1: 2: 3:	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 7-422. Register Call Summary for Register VIP_PERF_MON20

VIP Register Manual
- VIP VPDMA Register Summary: [2]

Table 7-423. VIP_PERF_MON21

Address Offset	0x0000 0254		
Physical Address	0x4897 D254	**Instance**	VIP_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA\\n		
Type	RW		

31 30	29 28	27	26 25 24	23 22	21 20	19	18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT	

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0: 1: 2: 3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0: 1: 2: 3:	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 7-424. Register Call Summary for Register VIP_PERF_MON21

VIP Register Manual

- VIP VPDMA Register Summary: [2]

Table 7-425. VIP_PERF_MON22

Address Offset	0x0000 0258		
Physical Address	0x4897 D258	**Instance**	VIP_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA\\n		
Type	RW		

31 30	29 28	27	26 25 24	23 22	21 20	19	18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT	

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0: 1: 2: 3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0: 1: 2: 3:	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 7-426. Register Call Summary for Register VIP_PERF_MON22

VIP Register Manual

- VIP VPDMA Register Summary: [2]

Table 7-427. VIP_PERF_MON23

Address Offset	0x0000 025C		
Physical Address	0x4897 D25C	**Instance**	VIP_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA\\n		
Type	RW		

31 30	29 28	27	26 25 24	23 22	21 20	19	18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT	

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0

Bits	Field Name	Description	Type	Reset
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0: 1: 2: 3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0: 1: 2: 3:	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 7-428. Register Call Summary for Register VIP_PERF_MON23

VIP Register Manual
- VIP VPDMA Register Summary: [2]

Table 7-429. VIP_PERF_MON24

Address Offset	0x0000 0260		
Physical Address	0x4897 D260	**Instance**	VIP_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA\\n		
Type	RW		

31 30	29 28	27	26 25 24	23 22	21 20	19	18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT	

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0: 1: 2: 3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0: 1: 2: 3:	RW	0x0
19	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 7-430. Register Call Summary for Register VIP_PERF_MON24

VIP Register Manual
- VIP VPDMA Register Summary: [2]

Table 7-431. VIP_PERF_MON25

Address Offset	0x0000 0264		
Physical Address	0x4897 D264	**Instance**	VIP_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA\\n		
Type	RW		

31 30	29 28	27	26 25 24	23 22	21 20	19	18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT	

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0: 1: 2: 3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0: 1: 2: 3:	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 7-432. Register Call Summary for Register VIP_PERF_MON25

VIP Register Manual
- VIP VPDMA Register Summary: [2]

Table 7-433. VIP_PERF_MON26

Address Offset	0x0000 0268		
Physical Address	0x4897 D268	**Instance**	VIP_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA\\n		
Type	RW		

31 30	29 28	27	26 25 24	23 22	21 20	19	18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT	

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0: 1: 2: 3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0: 1: 2: 3:	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 7-434. Register Call Summary for Register VIP_PERF_MON26

VIP Register Manual

- VIP VPDMA Register Summary: [2]

Table 7-435. VIP_PERF_MON27

Address Offset	0x0000 026C		
Physical Address	0x4897 D26C	**Instance**	VIP_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA\\n		
Type	RW		

31 30	29 28	27	26 25 24	23 22	21 20	19	18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT	

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0: 1: 2: 3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0: 1: 2: 3:	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 7-436. Register Call Summary for Register VIP_PERF_MON27

VIP Register Manual

- VIP VPDMA Register Summary: [2]

Table 7-437. VIP_PERF_MON28

Address Offset	0x0000 0270		
Physical Address	0x4897 D270	**Instance**	VIP_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA\\n		
Type	RW		

31 30	29 28	27	26 25 24	23 22	21 20	19	18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT	

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0

Bits	Field Name	Description	Type	Reset
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0: 1: 2: 3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0: 1: 2: 3:	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 7-438. Register Call Summary for Register VIP_PERF_MON28

VIP Register Manual
- VIP VPDMA Register Summary: [2]

Table 7-439. VIP_PERF_MON29

Address Offset	0x0000 0274		
Physical Address	0x4897 D274	**Instance**	VIP_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA\\n		
Type	RW		

31 30	29 28	27	26 25 24	23 22	21 20	19	18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT	

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0: 1: 2: 3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0: 1: 2: 3:	RW	0x0
19	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 7-440. Register Call Summary for Register VIP_PERF_MON29

VIP Register Manual
- VIP VPDMA Register Summary: [2]

Table 7-441. VIP_PERF_MON30

Address Offset	0x0000 0278		
Physical Address	0x4897 D278	**Instance**	VIP_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA\\n		
Type	RW		

31 30	29 28	27	26 25 24	23 22	21 20	19	18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT	

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0: 1: 2: 3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0: 1: 2: 3:	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 7-442. Register Call Summary for Register VIP_PERF_MON30

VIP Register Manual
- VIP VPDMA Register Summary: [2]

Table 7-443. VIP_PERF_MON31

Address Offset	0x0000 027C		
Physical Address	0x4897 D27C	**Instance**	VIP_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA\\n		
Type	RW		

31 30	29 28	27	26 25 24	23 22	21 20	19	18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT	

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0: 1: 2: 3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0: 1: 2: 3:	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 7-444. Register Call Summary for Register VIP_PERF_MON31

VIP Register Manual

- VIP VPDMA Register Summary: [2]

Table 7-445. VIP_PERF_MON32

Address Offset	0x0000 0280		
Physical Address	0x4897 D280	**Instance**	VIP_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA\\n		
Type	RW		

31 30	29 28	27	26 25 24	23 22	21 20	19	18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT	

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0: 1: 2: 3: vip1_lo_y	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0: 1: 2: 3: vip1_lo_y	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 7-446. Register Call Summary for Register VIP_PERF_MON32

VIP Register Manual

- VIP VPDMA Register Summary: [2]

Table 7-447. VIP_PERF_MON33

Address Offset	0x0000 0284		
Physical Address	0x4897 D284	**Instance**	VIP_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA\\n		
Type	RW		

31 30	29 28	27	26 25 24	23 22	21 20	19	18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT	

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0

Bits	Field Name	Description	Type	Reset
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0: 1: 2: vip1_lo_y 3: vip1_lo_uv	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0: 1: 2: vip1_lo_y 3: vip1_lo_uv	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 7-448. Register Call Summary for Register VIP_PERF_MON33

VIP Register Manual
- VIP VPDMA Register Summary: [2]

Table 7-449. VIP_PERF_MON34

Address Offset	0x0000 0288		
Physical Address	0x4897 D288	**Instance**	VIP_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA\\n		
Type	RW		

31 30	29 28	27	26 25 24	23 22	21 20	19	18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT	

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0: vip1_lo_y 1: 2: vip1_lo_uv 3: vip1_up_y	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0: vip1_lo_y 1: 2: vip1_lo_uv 3: vip1_up_y	RW	0x0
19	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 7-450. Register Call Summary for Register VIP_PERF_MON34

VIP Register Manual
- VIP VPDMA Register Summary: [2]

Table 7-451. VIP_PERF_MON35

Address Offset	0x0000 028C		
Physical Address	0x4897 D28C	**Instance**	VIP_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA\\n		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CAPTURE_MODE				STOP_CLIENT				RESERVED				STOP_COUNT				RESERVED				START_CLIENT				RESERVED				START_COUNT			
CURR_COUNT																															

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0: vip1_lo_uv 1: vip1_lo_y 2: vip1_up_y 3: vip1_up_uv	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0: vip1_lo_uv 1: vip1_lo_y 2: vip1_up_y 3: vip1_up_uv	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 7-452. Register Call Summary for Register VIP_PERF_MON35

VIP Register Manual
- VIP VPDMA Register Summary: [2]

Table 7-453. VIP_PERF_MON36

Address Offset	0x0000 0290		
Physical Address	0x4897 D290	**Instance**	VIP_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA\\n		
Type	RW		

31 30	29 28	27	26 25 24	23 22	21 20	19	18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT	

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0: vip1_up_y 1: vip1_lo_uv 2: vip1_up_uv 3: vip2_lo_y	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0: vip1_up_y 1: vip1_lo_uv 2: vip1_up_uv 3: vip2_lo_y	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 7-454. Register Call Summary for Register VIP_PERF_MON36

VIP Register Manual

- VIP VPDMA Register Summary: [2]

Table 7-455. VIP_PERF_MON37

Address Offset	0x0000 0294		
Physical Address	0x4897 D294	**Instance**	VIP_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA\\n		
Type	RW		

31 30	29 28	27	26 25 24	23 22	21 20	19	18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT	

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0: vip1_up_uv 1: vip1_up_y 2: vip2_lo_y 3: vip2_lo_uv	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT		RW	0x0
23:22	RESERVED	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0: vip1_up_uv 1: vip1_up_y 2: vip2_lo_y 3: vip2_lo_uv	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 7-456. Register Call Summary for Register VIP_PERF_MON37

VIP Register Manual

- VIP VPDMA Register Summary: [2]

Table 7-457. VIP_PERF_MON38

Address Offset	0x0000 0298		
Physical Address	0x4897 D298	**Instance**	VIP_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA\\n		
Type	RW		

31 30	29 28	27	26 25 24	23 22	21 20	19	18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT	

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0: vip2_lo_y 1: vip1_up_uv 2: vip2_lo_uv 3: vip2_up_y	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0: vip2_lo_y 1: vip1_up_uv 2: vip2_lo_uv 3: vip2_up_y	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 7-458. Register Call Summary for Register VIP_PERF_MON38

VIP Register Manual

- VIP VPDMA Register Summary: [2]

Table 7-459. VIP_PERF_MON39

Address Offset	0x0000 029C		
Physical Address	0x4897 D29C	**Instance**	VIP_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA\\n		
Type	RW		

31 30	29 28	27	26 25 24	23 22	21 20	19	18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT	

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0: vip2_lo_uv 1: vip2_lo_y 2: vip2_up_y 3: vip2_up_uv	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0: vip2_lo_uv 1: vip2_lo_y 2: vip2_up_y 3: vip2_up_uv	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 7-460. Register Call Summary for Register VIP_PERF_MON39

VIP Register Manual

- VIP VPDMA Register Summary: [2]

Table 7-461. VIP_PERF_MON40

Address Offset	0x0000 02A0		
Physical Address	0x4897 D2A0	**Instance**	VIP_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA\\n		
Type	RW		

31 30	29 28	27	26 25 24	23 22	21 20	19	18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT	

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0: vip2_up_y 1: vip2_lo_uv 2: vip2_up_uv 3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0: vip2_up_y 1: vip2_lo_uv 2: vip2_up_uv 3:	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 7-462. Register Call Summary for Register VIP_PERF_MON40

VIP Register Manual

- VIP VPDMA Register Summary: [2]

Table 7-463. VIP_PERF_MON41

Address Offset	0x0000 02A4		
Physical Address	0x4897 D2A4	**Instance**	VIP_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA\\n		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CAPTURE_MODE		STOP_CLIENT		RESERVED		STOP_COUNT		RESERVED		START_CLIENT		RESERVED		START_COUNT		CURR_COUNT															

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0: vip2_up_uv 1: vip2_up_y 2: 3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0: vip2_up_uv 1: vip2_up_y 2: 3:	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 7-464. Register Call Summary for Register VIP_PERF_MON41

VIP Register Manual

- VIP VPDMA Register Summary: [2]

Table 7-465. VIP_PERF_MON42

Address Offset	0x0000 02A8		
Physical Address	0x4897 D2A8	**Instance**	VIP_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA\\n		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CAPTURE_MODE		STOP_CLIENT		RESERVED		STOP_COUNT		RESERVED		START_CLIENT		RESERVED		START_COUNT		CURR_COUNT																	

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0: 1: vip2_up_uv 2: 3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0: 1: vip2_up_uv 2: 3:	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 7-466. Register Call Summary for Register VIP_PERF_MON42

VIP Register Manual

- VIP VPDMA Register Summary: [2]

Table 7-467. VIP_PERF_MON43

Address Offset	0x0000 02AC		
Physical Address	0x4897 D2AC	**Instance**	VIP_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA\\n		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CAPTURE_MODE		STOP_CLIENT		RESERVED		STOP_COUNT		RESERVED		START_CLIENT		RESERVED		START_COUNT		CURR_COUNT																	

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0

Bits	Field Name	Description	Type	Reset
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0: 1: 2: 3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0: 1: 2: 3:	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 7-468. Register Call Summary for Register VIP_PERF_MON43

VIP Register Manual
- VIP VPDMA Register Summary: [2]

Table 7-469. VIP_PERF_MON44

Address Offset	0x0000 02B0		
Physical Address	0x4897 D2B0	**Instance**	VIP_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA\\n		
Type	RW		

31 30	29 28	27	26 25 24	23 22	21 20	19	18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT	

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0: 1: 2: 3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0: 1: 2: 3:	RW	0x0
19	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 7-470. Register Call Summary for Register VIP_PERF_MON44

VIP Register Manual
- VIP VPDMA Register Summary: [2]

Table 7-471. VIP_PERF_MON45

Address Offset	0x0000 02B4		
Physical Address	0x4897 D2B4	**Instance**	VIP_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA\\n		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CAPTURE_MODE		STOP_CLIENT		RESERVED		STOP_COUNT		RESERVED		START_CLIENT		RESERVED		START_COUNT		CURR_COUNT															

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0: 1: 2: 3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0: 1: 2: 3:	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 7-472. Register Call Summary for Register VIP_PERF_MON45

VIP Register Manual
- VIP VPDMA Register Summary: [2]

Table 7-473. VIP_PERF_MON46

Address Offset	0x0000 02B8		
Physical Address	0x4897 D2B8	**Instance**	VIP_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA\\n		
Type	RW		

31 30	29 28	27	26 25 24	23 22	21 20	19	18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT	

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0: 1: 2: 3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0: 1: 2: 3:	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 7-474. Register Call Summary for Register VIP_PERF_MON46

VIP Register Manual

- VIP VPDMA Register Summary: [2]

Table 7-475. VIP_PERF_MON47

Address Offset	0x0000 02BC		
Physical Address	0x4897 D2BC	**Instance**	VIP_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA\\n		
Type	RW		

31 30	29 28	27	26 25 24	23 22	21 20	19	18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT	

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0: 1: 2: 3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0: 1: 2: 3:	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 7-476. Register Call Summary for Register VIP_PERF_MON47

VIP Register Manual

- VIP VPDMA Register Summary: [2]

Table 7-477. VIP_PERF_MON48

Address Offset	0x0000 02C0		
Physical Address	0x4897 D2C0	**Instance**	VIP_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA\\n		
Type	RW		

31 30	29 28	27	26 25 24	23 22	21 20	19	18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT	

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0

Bits	Field Name	Description	Type	Reset
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0: 1: 2: 3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0: 1: 2: 3:	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 7-478. Register Call Summary for Register VIP_PERF_MON48

VIP Register Manual
- VIP VPDMA Register Summary: [2]

Table 7-479. VIP_PERF_MON49

Address Offset	0x0000 02C4		
Physical Address	0x4897 D2C4	**Instance**	VIP_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA\\n		
Type	RW		

31 30	29 28	27	26 25 24	23 22	21 20	19	18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT	

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0: 1: 2: 3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0: 1: 2: 3:	RW	0x0
19	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 7-480. Register Call Summary for Register VIP_PERF_MON49

VIP Register Manual
- VIP VPDMA Register Summary: [2]

Table 7-481. VIP_PERF_MON50

Address Offset	0x0000 02C8		
Physical Address	0x4897 D2C8	**Instance**	VIP_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA\\n		
Type	RW		

31 30	29 28	27	26 25 24	23 22	21 20	19	18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT	

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0: 1: 2: 3: vpi_ctl	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0: 1: 2: 3: vpi_ctl	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 7-482. Register Call Summary for Register VIP_PERF_MON50

VIP Register Manual
- VIP VPDMA Register Summary: [2]

Table 7-483. VIP_PERF_MON51

Address Offset	0x0000 02CC		
Physical Address	0x4897 D2CC	**Instance**	VIP_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA\\n		
Type	RW		

31 30	29 28	27	26 25 24	23 22	21 20	19	18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT	

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0: 1: 2: vpi_ctl 3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0: 1: 2: vpi_ctl 3:	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 7-484. Register Call Summary for Register VIP_PERF_MON51

VIP Register Manual

- VIP VPDMA Register Summary: [2]

Table 7-485. VIP_PERF_MON52

Address Offset	0x0000 02D0		
Physical Address	0x4897 D2D0	**Instance**	VIP_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA\\n		
Type	RW		

31 30	29 28	27	26 25 24	23 22	21 20	19	18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT	

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0: vpi_ctl 1: 2: 3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0: vpi_ctl 1: 2: 3:	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 7-486. Register Call Summary for Register VIP_PERF_MON52

VIP Register Manual

- VIP VPDMA Register Summary: [2]

Table 7-487. VIP_PERF_MON53

Address Offset	0x0000 02D4		
Physical Address	0x4897 D2D4	**Instance**	VIP_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA\\n		
Type	RW		

31 30	29 28	27	26 25 24	23 22	21 20	19	18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT	

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0

Bits	Field Name	Description	Type	Reset
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0: 1: vpi_ctl 2: 3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0: 1: vpi_ctl 2: 3:	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 7-488. Register Call Summary for Register VIP_PERF_MON53

VIP Register Manual
- VIP VPDMA Register Summary: [2]

Table 7-489. VIP_PERF_MON54

Address Offset	0x0000 02D8		
Physical Address	0x4897 D2D8	**Instance**	VIP_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA\\n		
Type	RW		

31 30	29 28	27	26 25 24	23 22	21 20	19	18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT	

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0: 1: 2: 3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0: 1: 2: 3:	RW	0x0
19	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 7-490. Register Call Summary for Register VIP_PERF_MON54

VIP Register Manual
- VIP VPDMA Register Summary: [2]

Table 7-491. VIP_PERF_MON55

Address Offset	0x0000 02DC		
Physical Address	0x4897 D2DC	**Instance**	VIP_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA\\n		
Type	RW		

31 30	29 28	27	26 25 24	23 22	21 20	19	18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT	

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0: 1: 2: 3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0: 1: 2: 3:	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 7-492. Register Call Summary for Register VIP_PERF_MON55

VIP Register Manual
- VIP VPDMA Register Summary: [2]

Table 7-493. VIP_PERF_MON56

Address Offset	0x0000 02E0		
Physical Address	0x4897 D2E0	**Instance**	VIP_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA\\n		
Type	RW		

31 30	29 28	27	26 25 24	23 22	21 20	19	18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT	

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0
29:28	STOP_CLIENT		RW	0x0
27	RESERVED	Sets the client whose event stops the performance monitor counter. 0: 1: 2: 3: vip1_anc_a	R	0x0
26:24	STOP_COUNT		RW	0x0
23:22	RESERVED	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	R	0x0
21:20	START_CLIENT		RW	0x0
19	RESERVED	Sets the client whose event starts the performance monitor counter. 0: 1: 2: 3: vip1_anc_a	R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 7-494. Register Call Summary for Register VIP_PERF_MON56

VIP Register Manual

- VIP VPDMA Register Summary: [2]

Table 7-495. VIP_PERF_MON57

Address Offset	0x0000 02E4		
Physical Address	0x4897 D2E4	**Instance**	VIP_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA\\n		
Type	RW		

31 30	29 28	27	26 25 24	23 22	21 20	19	18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT	

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0: 1: 2: vip1_anc_a 3: vip1_anc_b	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0: 1: 2: vip1_anc_a 3: vip1_anc_b	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 7-496. Register Call Summary for Register VIP_PERF_MON57

VIP Register Manual

- VIP VPDMA Register Summary: [2]

Table 7-497. VIP_PERF_MON58

Address Offset	0x0000 02E8		
Physical Address	0x4897 D2E8	**Instance**	VIP_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA\\n		
Type	RW		

31 30	29 28	27	26 25 24	23 22	21 20	19	18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT	

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0

Bits	Field Name	Description	Type	Reset
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0: vip1_anc_a 1: 2: vip1_anc_b 3: vip2_anc_a	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0: vip1_anc_a 1: 2: vip1_anc_b 3: vip2_anc_a	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 7-498. Register Call Summary for Register VIP_PERF_MON58

VIP Register Manual
- VIP VPDMA Register Summary: [2]

Table 7-499. VIP_PERF_MON59

Address Offset	0x0000 02EC		
Physical Address	0x4897 D2EC	**Instance**	VIP_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA\\n		
Type	RW		

31 30	29 28	27	26 25 24	23 22	21 20	19	18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT	

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0: vip1_anc_b 1: vip1_anc_a 2: vip2_anc_a 3: vip2_anc_b	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0: vip1_anc_b 1: vip1_anc_a 2: vip2_anc_a 3: vip2_anc_b	RW	0x0

Bits	Field Name	Description	Type	Reset
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 7-500. Register Call Summary for Register VIP_PERF_MON59

VIP Register Manual
- VIP VPDMA Register Summary: [2]

Table 7-501. VIP_PERF_MON60

Address Offset	0x0000 02F0		
Physical Address	0x4897 D2F0	**Instance**	VIP_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA\\n		
Type	RW		

31 30	29 28	27	26 25 24	23 22	21 20	19	18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT	

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0: vip2_anc_a 1: vip1_anc_b 2: vip2_anc_b 3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0: vip2_anc_a 1: vip1_anc_b 2: vip2_anc_b 3:	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 7-502. Register Call Summary for Register VIP_PERF_MON60

VIP Register Manual
- VIP VPDMA Register Summary: [2]

Table 7-503. VIP_PERF_MON61

Address Offset	0x0000 02F4		
Physical Address	0x4897 D2F4	**Instance**	VIP_VPDMA
Description	The register can be used to capture timing differences between events in the VPDMA\\n		
Type	RW		

31 30	29 28	27	26 25 24	23 22	21 20	19	18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
CAPTURE_MODE	STOP_CLIENT	RESERVED	STOP_COUNT	RESERVED	START_CLIENT	RESERVED	START_COUNT	CURR_COUNT	

Bits	Field Name	Description	Type	Reset
31:30	CAPTURE_MODE	Sets how the counter should be updated. Updating this value will also clear the current counter stored value. 0: Running Average 1: Minimum Value 2: Maximum Value 3: Last Value	RW	0x0
29:28	STOP_CLIENT	Sets the client whose event stops the performance monitor counter. 0: vip2_anc_b 1: vip2_anc_a 2: 3:	RW	0x0
27	RESERVED		R	0x0
26:24	STOP_COUNT	Sets the value that stops the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
23:22	RESERVED		R	0x0
21:20	START_CLIENT	Sets the client whose event starts the performance monitor counter. 0: vip2_anc_b 1: vip2_anc_a 2: 3:	RW	0x0
19	RESERVED		R	0x0
18:16	START_COUNT	Sets the value that starts the performance monitor counter. 0: command request 1: command accept 2: data request 3: data rcvd 4: data empty 5: data full 6: frame start 7: frame end	RW	0x0
15:0	CURR_COUNT	The current value of the perfomance monitor counter	R	0x0

Table 7-504. Register Call Summary for Register VIP_PERF_MON61

VIP Register Manual

- VIP VPDMA Register Summary: [2]

Table 7-505. VIP0_LO_Y_CSTAT

Address Offset	0x0000 0388		
Physical Address	0x4897 D388	**Instance**	VIP_VPDMA
Description	The register holds status information and control for the client.\\n		
Type	RW		

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15	14	13 12 11 10	9 8	7 6 5 4 3 2 1 0
REQ_DELAY	REQ_RATE	BUSY	DMA_ACTIVE	FRAME_START	RESERVED	

Bits	Field Name	Description	Type	Reset
31:24	REQ_DELAY	The minimum number of clock cycles between requests being issued. This value is multiplied by 32 to get the actual number of cycles..This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins and the first request of a frame will go as soon as possible.	RW	0x0
23:16	REQ_RATE	The number of clock cycles between the last two requests issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins.	R	0x0
15	BUSY	Signals if the client is currently active. This bit is set as soon as we the channel is received by the client from the list manager and is cleared when the channel is cleared from the shared memory.	R	0x0
14	DMA_ACTIVE	Signals if the client is currently actively sending DMA requests	R	0x0
13:10	FRAME_START	The source of the start frame event for the client.\\n0 : Change in value of hdmi_field_id\\n1 : Change in value of dvo2_field_id\\n2 : Change in value of hdcomp_field_id\\n3 : Change in value of sd_field_id\\n4 : Use List Manager Internal Field0\\n5 : Use List Manager Internal Field1\\n6 : Use List Manager Internal Field2\\n7 : Start on channel active	RW	0x0
9:0	RESERVED		R	0x0

Table 7-506. Register Call Summary for Register VIP0_LO_Y_CSTAT

VIP Functional Description
- VPDMA Basic Definitions:

VIP Register Manual
- VIP VPDMA Register Summary: [3]

Table 7-507. VIP0_LO_UV_CSTAT

Address Offset	0x0000 038C		
Physical Address	0x4897 D38C	**Instance**	VIP_VPDMA
Description	The register holds status information and control for the client.\\n		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REQ_DELAY								REQ_RATE								BUSY	DMA_ACTIVE	FRAME_START				RESERVED									

Bits	Field Name	Description	Type	Reset
31:24	REQ_DELAY	The minimum number of clock cycles between requests being issued. This value is multiplied by 32 to get the actual number of cycles..This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins and the first request of a frame will go as soon as possible.	RW	0x0
23:16	REQ_RATE	The number of clock cycles between the last two requests issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins.	R	0x0

Bits	Field Name	Description	Type	Reset
15	BUSY	Signals if the client is currently active. This bit is set as soon as we the channel is received by the client from the list manager and is cleared when the channel is cleared from the shared memory.	R	0x0
14	DMA_ACTIVE	Signals if the client is currently actively sending DMA requests	R	0x0
13:10	FRAME_START	The source of the start frame event for the client.\\n0 : Change in value of hdmi_field_id\\n1 : Change in value of dvo2_field_id\\n2 : Change in value of hdcomp_field_id\\n3 : Change in value of sd_field_id\\n4 : Use List Manager Internal Field0\\n5 : Use List Manager Internal Field1\\n6 : Use List Manager Internal Field2\\n7 : Start on channel active	RW	0x0
9:0	RESERVED		R	0x0

Table 7-508. Register Call Summary for Register VIP0_LO_UV_CSTAT

VIP Functional Description
- VPDMA Basic Definitions:

VIP Register Manual
- VIP VPDMA Register Summary: [3]

Table 7-509. VIP0_UP_Y_CSTAT

Address Offset	0x0000 0390		
Physical Address	0x4897 D390	**Instance**	VIP_VPDMA
Description	The register holds status information and control for the client.\\n		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REQ_DELAY								REQ_RATE								BUSY	DMA_ACTIVE	FRAME_START						RESERVED							

Bits	Field Name	Description	Type	Reset
31:24	REQ_DELAY	The minimum number of clock cycles between requests being issued. This value is multiplied by 32 to get the actual number of cycles..This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins and the first request of a frame will go as soon as possible.	RW	0x0
23:16	REQ_RATE	The number of clock cycles between the last two requests issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins.	R	0x0
15	BUSY	Signals if the client is currently active. This bit is set as soon as we the channel is received by the client from the list manager and is cleared when the channel is cleared from the shared memory.	R	0x0
14	DMA_ACTIVE	Signals if the client is currently actively sending DMA requests	R	0x0

Bits	Field Name	Description	Type	Reset
13:10	FRAME_START	The source of the start frame event for the client.\\n0 : Change in value of hdmi_field_id\\n1 : Change in value of dvo2_field_id\\n2 : Change in value of hdcomp_field_id\\n3 : Change in value of sd_field_id\\n4 : Use List Manager Internal Field0\\n5 : Use List Manager Internal Field1\\n6 : Use List Manager Internal Field2\\n7 : Start on channel active	RW	0x0
9:0	RESERVED		R	0x0

Table 7-510. Register Call Summary for Register VIP0_UP_Y_CSTAT

VIP Functional Description
- VPDMA Basic Definitions:

VIP Register Manual
- VIP VPDMA Register Summary: [3]

Table 7-511. VIP0_UP_UV_CSTAT

Address Offset	0x0000 0394		
Physical Address	0x4897 D394	**Instance**	VIP_VPDMA
Description	The register holds status information and control for the client.\\n		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REQ_DELAY								REQ_RATE								BUSY	DMA_ACTIVE	FRAME_START						RESERVED							

Bits	Field Name	Description	Type	Reset
31:24	REQ_DELAY	The minimum number of clock cycles between requests being issued. This value is multiplied by 32 to get the actual number of cycles..This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins and the first request of a frame will go as soon as possible.	RW	0x0
23:16	REQ_RATE	The number of clock cycles between the last two requests issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins.	R	0x0
15	BUSY	Signals if the client is currently active. This bit is set as soon as we the channel is received by the client from the list manager and is cleared when the channel is cleared from the shared memory.	R	0x0
14	DMA_ACTIVE	Signals if the client is currently actively sending DMA requests	R	0x0
13:10	FRAME_START	The source of the start frame event for the client.\\n0 : Change in value of hdmi_field_id\\n1 : Change in value of dvo2_field_id\\n2 : Change in value of hdcomp_field_id\\n3 : Change in value of sd_field_id\\n4 : Use List Manager Internal Field0\\n5 : Use List Manager Internal Field1\\n6 : Use List Manager Internal Field2\\n7 : Start on channel active	RW	0x0
9:0	RESERVED		R	0x0

Table 7-512. Register Call Summary for Register VIP0_UP_UV_CSTAT

VIP Functional Description • VPDMA Basic Definitions:
VIP Register Manual • VIP VPDMA Register Summary: [3]

Table 7-513. VIP1_LO_Y_CSTAT

Address Offset	0x0000 0398		
Physical Address	0x4897 D398	**Instance**	VIP_VPDMA
Description	The register holds status information and control for the client.\\n		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REQ_DELAY								REQ_RATE								BUSY	DMA_ACTIVE	FRAME_START						RESERVED							

Bits	Field Name	Description	Type	Reset
31:24	REQ_DELAY	The minimum number of clock cycles between requests being issued. This value is multiplied by 32 to get the actual number of cycles..This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins and the first request of a frame will go as soon as possible.	RW	0x0
23:16	REQ_RATE	The number of clock cycles between the last two requests issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins.	R	0x0
15	BUSY	Signals if the client is currently active. This bit is set as soon as we the channel is received by the client from the list manager and is cleared when the channel is cleared from the shared memory.	R	0x0
14	DMA_ACTIVE	Signals if the client is currently actively sending DMA requests	R	0x0
13:10	FRAME_START	The source of the start frame event for the client.\\n0 : Change in value of hdmi_field_id\\n1 : Change in value of dvo2_field_id\\n2 : Change in value of hdcomp_field_id\\n3 : Change in value of sd_field_id\\n4 : Use List Manager Internal Field0\\n5 : Use List Manager Internal Field1\\n6 : Use List Manager Internal Field2\\n7 : Start on channel active	RW	0x0
9:0	RESERVED		R	0x0

Table 7-514. Register Call Summary for Register VIP1_LO_Y_CSTAT

VIP Functional Description • VPDMA Basic Definitions:
VIP Register Manual • VIP VPDMA Register Summary: [3]

Table 7-515. VIP1_LO_UV_CSTAT

Address Offset	0x0000 039C		
Physical Address	0x4897 D39C	**Instance**	VIP_VPDMA
Description	The register holds status information and control for the client.\\n		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REQ_DELAY								REQ_RATE								BUSY	DMA_ACTIVE	FRAME_START						RESERVED							

Bits	Field Name	Description	Type	Reset
31:24	REQ_DELAY	The minimum number of clock cycles between requests being issued. This value is multiplied by 32 to get the actual number of cycles..This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins and the first request of a frame will go as soon as possible.	RW	0x0
23:16	REQ_RATE	The number of clock cycles between the last two requests issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins.	R	0x0
15	BUSY	Signals if the client is currently active. This bit is set as soon as we the channel is received by the client from the list manager and is cleared when the channel is cleared from the shared memory.	R	0x0
14	DMA_ACTIVE	Signals if the client is currently actively sending DMA requests	R	0x0
13:10	FRAME_START	The source of the start frame event for the client.\\n0 : Change in value of hdmi_field_id\\n1 : Change in value of dvo2_field_id\\n2 : Change in value of hdcomp_field_id\\n3 : Change in value of sd_field_id\\n4 : Use List Manager Internal Field0\\n5 : Use List Manager Internal Field1\\n6 : Use List Manager Internal Field2\\n7 : Start on channel active	RW	0x0
9:0	RESERVED		R	0x0

Table 7-516. Register Call Summary for Register VIP1_LO_UV_CSTAT

VIP Functional Description

- VPDMA Basic Definitions:

VIP Register Manual

- VIP VPDMA Register Summary: [3]

Table 7-517. VIP1_UP_Y_CSTAT

Address Offset	0x0000 03A0		
Physical Address	0x4897 D3A0	**Instance**	VIP_VPDMA
Description	The register holds status information and control for the client.\\n		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REQ_DELAY								REQ_RATE								BUSY	DMA_ACTIVE	FRAME_START				RESERVED									

Bits	Field Name	Description	Type	Reset
31:24	REQ_DELAY	The minimum number of clock cycles between requests being issued. This value is multiplied by 32 to get the actual number of cycles..This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins and the first request of a frame will go as soon as possible.	RW	0x0
23:16	REQ_RATE	The number of clock cycles between the last two requests issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins.	R	0x0
15	BUSY	Signals if the client is currently active. This bit is set as soon as we the channel is received by the client from the list manager and is cleared when the channel is cleared from the shared memory.	R	0x0
14	DMA_ACTIVE	Signals if the client is currently actively sending DMA requests	R	0x0
13:10	FRAME_START	The source of the start frame event for the client.\\n0 : Change in value of hdmi_field_id\\n1 : Change in value of dvo2_field_id\\n2 : Change in value of hdcomp_field_id\\n3 : Change in value of sd_field_id\\n4 : Use List Manager Internal Field0\\n5 : Use List Manager Internal Field1\\n6 : Use List Manager Internal Field2\\n7 : Start on channel active	RW	0x0
9:0	RESERVED		R	0x0

Table 7-518. Register Call Summary for Register VIP1_UP_Y_CSTAT

VIP Functional Description
- VPDMA Basic Definitions:

VIP Register Manual
- VIP VPDMA Register Summary: [3]

Table 7-519. VIP1_UP_UV_CSTAT

Address Offset	0x0000 03A4		
Physical Address	0x4897 D3A4	**Instance**	VIP_VPDMA
Description	The register holds status information and control for the client.\\n		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REQ_DELAY								REQ_RATE								BUSY	DMA_ACTIVE	FRAME_START				RESERVED									

Bits	Field Name	Description	Type	Reset
31:24	REQ_DELAY	The minimum number of clock cycles between requests being issued. This value is multiplied by 32 to get the actual number of cycles..This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins and the first request of a frame will go as soon as possible.	RW	0x0
23:16	REQ_RATE	The number of clock cycles between the last two requests issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins.	R	0x0
15	BUSY	Signals if the client is currently active. This bit is set as soon as we the channel is received by the client from the list manager and is cleared when the channel is cleared from the shared memory.	R	0x0
14	DMA_ACTIVE	Signals if the client is currently actively sending DMA requests	R	0x0
13:10	FRAME_START	The source of the start frame event for the client.\\n0 : Change in value of hdmi_field_id\\n1 : Change in value of dvo2_field_id\\n2 : Change in value of hdcomp_field_id\\n3 : Change in value of sd_field_id\\n4 : Use List Manager Internal Field0\\n5 : Use List Manager Internal Field1\\n6 : Use List Manager Internal Field2\\n7 : Start on channel active	RW	0x0
9:0	RESERVED		R	0x0

Table 7-520. Register Call Summary for Register VIP1_UP_UV_CSTAT

VIP Functional Description
- VPDMA Basic Definitions:

VIP Register Manual
- VIP VPDMA Register Summary: [3]

Table 7-521. VPI_CTL_CSTAT

Address Offset	0x0000 03D0		
Physical Address	0x4897 D3D0	**Instance**	VIP_VPDMA
Description	The register holds status information and control for the client.\\n		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REQ_DELAY								REQ_RATE								BUSY	DMA_ACTIVE	FRAME_START						RESERVED							

Bits	Field Name	Description	Type	Reset
31:24	REQ_DELAY	The minimum number of clock cycles between requests being issued. This value is multiplied by 32 to get the actual number of cycles..This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins and the first request of a frame will go as soon as possible.	RW	0x0
23:16	REQ_RATE	The number of clock cycles between the last two requests issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins.	R	0x0

Bits	Field Name	Description	Type	Reset
15	BUSY	Signals if the client is currently active. This bit is set as soon as we the channel is received by the client from the list manager and is cleared when the channel is cleared from the shared memory.	R	0x0
14	DMA_ACTIVE	Signals if the client is currently actively sending DMA requests	R	0x0
13:10	FRAME_START	The source of the start frame event for the client.\\n0 : Change in value of hdmi_field_id\\n1 : Change in value of dvo2_field_id\\n2 : Change in value of hdcomp_field_id\\n3 : Change in value of sd_field_id\\n4 : Use List Manager Internal Field0\\n5 : Use List Manager Internal Field1\\n6 : Use List Manager Internal Field2\\n7 : Start on channel active	RW	0x0
9:0	RESERVED		R	0x0

Table 7-522. Register Call Summary for Register VPI_CTL_CSTAT

VIP Functional Description • VPDMA Basic Definitions:
VIP Register Manual • VIP VPDMA Register Summary: [3]

Table 7-523. VIP0_ANC_A_CSTAT

Address Offset	0x0000 03E8		
Physical Address	0x4897 D3E8	**Instance**	VIP_VPDMA
Description	The register holds status information and control for the client.\\n		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REQ_DELAY								REQ_RATE								BUSY	DMA_ACTIVE	FRAME_START						RESERVED							

Bits	Field Name	Description	Type	Reset
31:24	REQ_DELAY	The minimum number of clock cycles between requests being issued. This value is multiplied by 32 to get the actual number of cycles..This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins and the first request of a frame will go as soon as possible.	RW	0x0
23:16	REQ_RATE	The number of clock cycles between the last two requests issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins.	R	0x0
15	BUSY	Signals if the client is currently active. This bit is set as soon as we the channel is received by the client from the list manager and is cleared when the channel is cleared from the shared memory.	R	0x0
14	DMA_ACTIVE	Signals if the client is currently actively sending DMA requests	R	0x0

Bits	Field Name	Description	Type	Reset
13:10	FRAME_START	The source of the start frame event for the client.\\n0 : Change in value of hdmi_field_id\\n1 : Change in value of dvo2_field_id\\n2 : Change in value of hdcomp_field_id\\n3 : Change in value of sd_field_id\\n4 : Use List Manager Internal Field0\\n5 : Use List Manager Internal Field1\\n6 : Use List Manager Internal Field2\\n7 : Start on channel active	RW	0x0
9:0	RESERVED		R	0x0

Table 7-524. Register Call Summary for Register VIP0_ANC_A_CSTAT

VIP Functional Description
- VPDMA Basic Definitions:

VIP Register Manual
- VIP VPDMA Register Summary: [3]

Table 7-525. VIP0_ANC_B_CSTAT

Address Offset	0x0000 03EC		
Physical Address	0x4897 D3EC	**Instance**	VIP_VPDMA
Description	The register holds status information and control for the client.\\n		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REQ_DELAY								REQ_RATE								BUSY	DMA_ACTIVE	FRAME_START						RESERVED							

Bits	Field Name	Description	Type	Reset
31:24	REQ_DELAY	The minimum number of clock cycles between requests being issued. This value is multiplied by 32 to get the actual number of cycles..This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins and the first request of a frame will go as soon as possible.	RW	0x0
23:16	REQ_RATE	The number of clock cycles between the last two requests issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins.	R	0x0
15	BUSY	Signals if the client is currently active. This bit is set as soon as we the channel is received by the client from the list manager and is cleared when the channel is cleared from the shared memory.	R	0x0
14	DMA_ACTIVE	Signals if the client is currently actively sending DMA requests	R	0x0
13:10	FRAME_START	The source of the start frame event for the client.\\n0 : Change in value of hdmi_field_id\\n1 : Change in value of dvo2_field_id\\n2 : Change in value of hdcomp_field_id\\n3 : Change in value of sd_field_id\\n4 : Use List Manager Internal Field0\\n5 : Use List Manager Internal Field1\\n6 : Use List Manager Internal Field2\\n7 : Start on channel active	RW	0x0
9:0	RESERVED		R	0x0

Table 7-526. Register Call Summary for Register VIP0_ANC_B_CSTAT

VIP Functional Description
• VPDMA Basic Definitions:
VIP Register Manual
• VIP VPDMA Register Summary: [3]

Table 7-527. VIP1_ANC_A_CSTAT

Address Offset	0x0000 03F0		
Physical Address	0x4897 D3F0	**Instance**	VIP_VPDMA
Description	The register holds status information and control for the client.\\n		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REQ_DELAY								REQ_RATE								BUSY	DMA_ACTIVE	FRAME_START						RESERVED							

Bits	Field Name	Description	Type	Reset
31:24	REQ_DELAY	The minimum number of clock cycles between requests being issued. This value is multiplied by 32 to get the actual number of cycles..This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins and the first request of a frame will go as soon as possible.	RW	0x0
23:16	REQ_RATE	The number of clock cycles between the last two requests issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins.	R	0x0
15	BUSY	Signals if the client is currently active. This bit is set as soon as we the channel is received by the client from the list manager and is cleared when the channel is cleared from the shared memory.	R	0x0
14	DMA_ACTIVE	Signals if the client is currently actively sending DMA requests	R	0x0
13:10	FRAME_START	The source of the start frame event for the client.\\n0 : Change in value of hdmi_field_id\\n1 : Change in value of dvo2_field_id\\n2 : Change in value of hdcomp_field_id\\n3 : Change in value of sd_field_id\\n4 : Use List Manager Internal Field0\\n5 : Use List Manager Internal Field1\\n6 : Use List Manager Internal Field2\\n7 : Start on channel active	RW	0x0
9:0	RESERVED		R	0x0

Table 7-528. Register Call Summary for Register VIP1_ANC_A_CSTAT

VIP Functional Description
• VPDMA Basic Definitions:
VIP Register Manual
• VIP VPDMA Register Summary: [3]

Table 7-529. VIP1_ANC_B_CSTAT

Address Offset	0x0000 03F4		
Physical Address	0x4897 D3F4	**Instance**	VIP_VPDMA
Description	The register holds status information and control for the client.\\n		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REQ_DELAY								REQ_RATE								BUSY	DMA_ACTIVE	FRAME_START						RESERVED							

Bits	Field Name	Description	Type	Reset
31:24	REQ_DELAY	The minimum number of clock cycles between requests being issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins and the first request of a frame will go as soon as possible.	RW	0x0
23:16	REQ_RATE	The number of clock cycles between the last two requests issued. This value is multiplied by 32 to get the actual number of cycles.This value is only accurate for the current frame. The internal counters used to calculate the rate are reset when a new frame begins.	R	0x0
15	BUSY	Signals if the client is currently active. This bit is set as soon as we the channel is received by the client from the list manager and is cleared when the channel is cleared from the shared memory.	R	0x0
14	DMA_ACTIVE	Signals if the client is currently actively sending DMA requests	R	0x0
13:10	FRAME_START	The source of the start frame event for the client.\\n0 : Change in value of hdmi_field_id\\n1 : Change in value of dvo2_field_id\\n2 : Change in value of hdcomp_field_id\\n3 : Change in value of sd_field_id\\n4 : Use List Manager Internal Field0\\n5 : Use List Manager Internal Field1\\n6 : Use List Manager Internal Field2\\n7 : Start on channel active	RW	0x0
9:0	RESERVED		R	0x0

Table 7-530. Register Call Summary for Register VIP1_ANC_B_CSTAT

VIP Functional Description

- VPDMA Basic Definitions:

VIP Register Manual

- VIP VPDMA Register Summary: [3]

Display Subsystem

This chapter describes the display subsystem for the device.

Topic **Page**

8.1 Display Subsystem Overview

The Display Subsystem (DSS) provides the logic to interface display peripherals. DSS integrates a DMA engine as part of DISPC module, which allows direct access to the memory frame buffer. Various pixel processing capabilities are supported, such as: color space conversion, filtering, scaling, blending, color keying, etc.

The supported display interfaces are:

- One parallel CMOS output, which can be used for MIPI® DPI 2.0, or BT-656 or BT-1120.
- One TV output, which is connected to the internal Video Encoder module (VENC). The VENC drives a single video digital-to-analog converter (SD_DAC) supporting composite video mode.

The modules integrated in the display subsystem are:

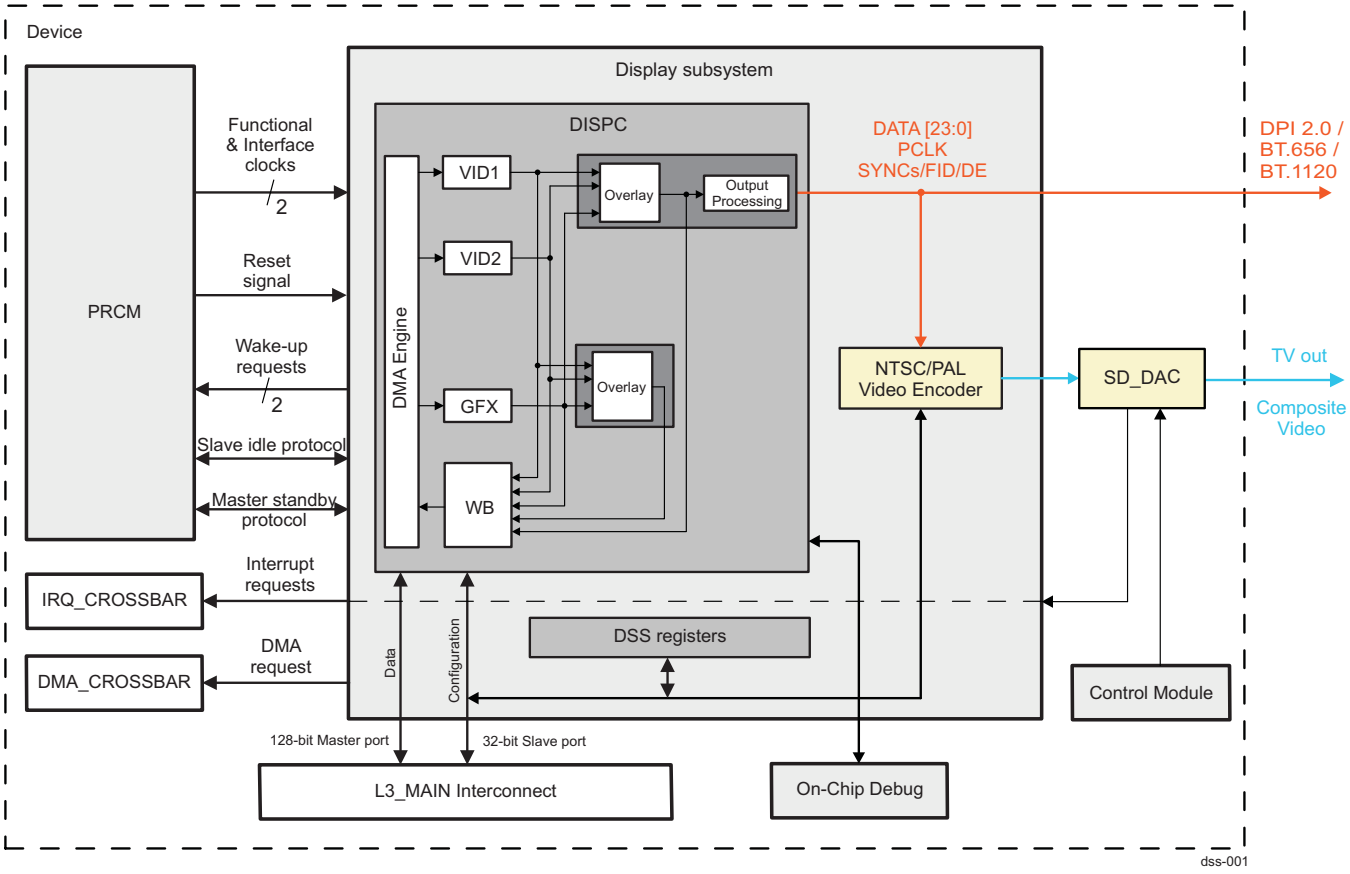
- Display controller (DISPC), with the following main features
  - One direct memory access (DMA) engine
  - One graphics pipeline (GFX), two video pipelines (VID1 and VID2), and one write-back pipeline (WB)
  - Two overlay managers
  - Active Matrix color support for 12/16/18/24-bit (truncated or dithered encoded pixel values)
  - One Video Port (VP) with programmable timing generator to support:
    - DPI: up to 165 MHz pixel clock video formats defined in CEA-861-E and VESA DMT standards
    - VENC: NTSC/PAL standards with 60Hz/50Hz refresh rates
  - Supported maximum FrameBuffer width of 4096 for all pixel formats
  - Configurable output mode: progressive or interlaced
  - Selection between RGB and YUV422 output pixel formats (YUV4:2:2 only available when BT-656 or BT-1120 output mode is enabled)
  - Refer to Section 8.2, *Display Controller*, for more details
- Video Encoder (VENC) with 10-bit standard definition video DAC (SD_DAC). Refer to Section 8.3, *Video Encoder*, for more details.

DSS provides two interfaces to L3_MAIN interconnect

- One 128-bit master port (with MFLAG support). The DMA engine in DISPC uses this single master to read/write data from/to device system memory.
- One 32-bit slave port. Used for registers configuration. It is further connected internally to DISPC and VENC modules.

Figure 8-1 is a high-level diagram of the display subsystem.

Figure 8-1. Display Subsystem Overview



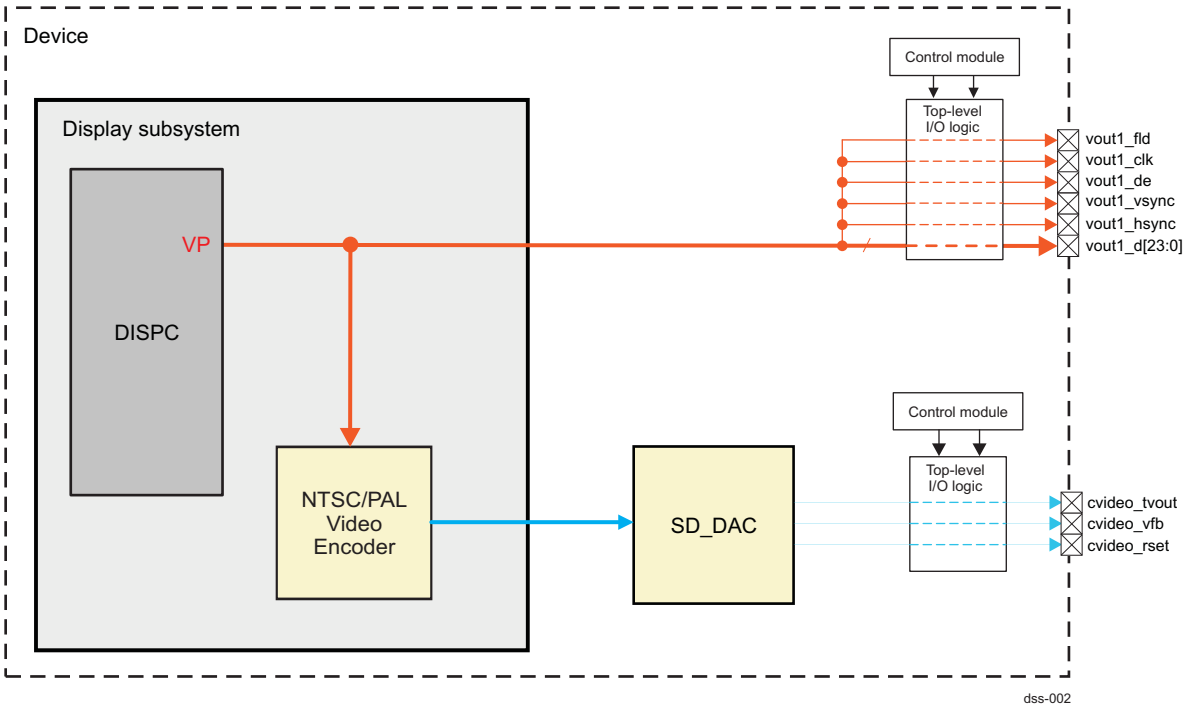
8.1.1 Display Subsystem Environment

This section describes the various outputs handled by the display subsystem:

- Parallel interface (MIPI DPI 2.0, or BT-656 or BT-1120) support
- TV (composite video) output support

Figure 8-2 is a diagram of the display subsystem environment.

Figure 8-2. Display Subsystem Environment



NOTE: The path from a module pin to device pad(s) is defined at the device I/O logic level. The I/O logic maps the module signals to the different pads of the device and is programmable in the control module registers and dedicated IP registers. For more information, see Section 13.4.6.3 *Pad Configuration Registers* in Chapter 13, *Control Module*.

CAUTION

The parallel and TV out interfaces cannot be used simultaneously.

- When the DSS_DPI_CTRL[0] DPI_ENABLE is set to 0x1, the parallel DPI output is enabled.
- When the DSS_DPI_CTRL[0] DPI_ENABLE is set to 0x0, the parallel DPI output is disabled. The TV composite output through VENC and SD_DAC modules can be used.

8.1.1.1 Display Subsystem Parallel Interface

In a parallel interface configuration, the DISPC outputs the VP data and interface signals on device I/O pads: DATA[23:0] (data bus), HSYNC (horizontal synchronization signal), VSYNC (vertical synchronization signal), DE (data enable), FID (field identification, odd/even) and PCLK (pixel clock). The required data and control signals can be provided directly to an external MIPI DPI-compatible LCD panels.

For more details on the output pixel data formats for the parallel interface, see Section 8.2.2 *DISPC Environment*, in *Display Controller*.

8.1.1.2 Display Subsystem TV Output

In a TV output configuration, the DISPC, VENC and video SD_DAC modules are used in the data path. The DISPC receives synchronization signals from the VENC and synchronously sends pixel data to the VENC with these signals. The digital output of the DISPC is always a 24-bit RGB value based on a pixel request from the VENC. The VENC converts RGB video signals to conform to the NTSC/PAL standards for analog video.

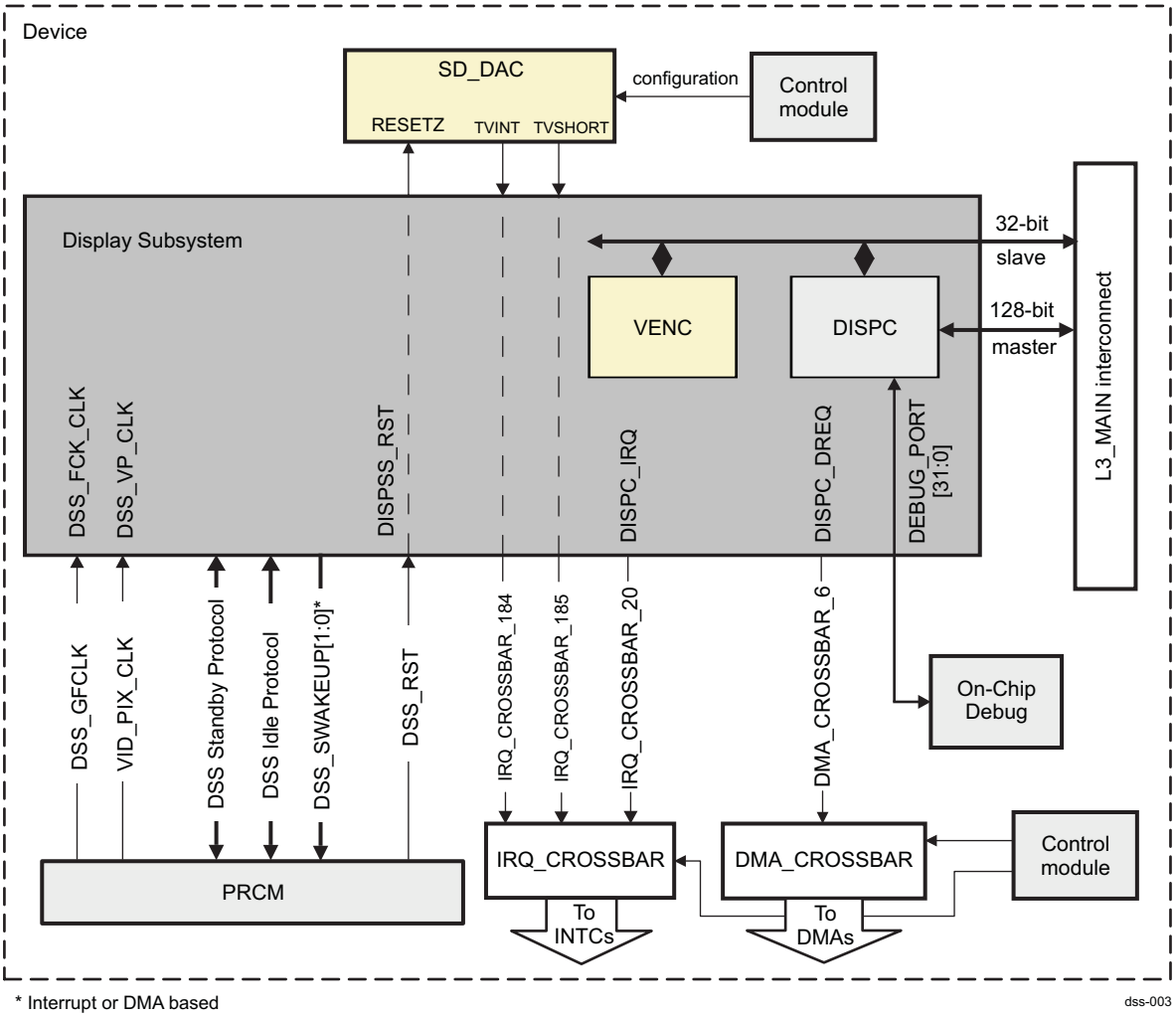
For more details, see Section 8.3, *Video Encoder*.

8.1.2 Display Subsystem Integration

This section describes the integration of the display subsystem module in the device, including information about clocks, resets, and hardware requests.

Figure 8-3 shows the integration of the display subsystem in the device.

Figure 8-3. Display Subsystem Integration



8.1.2.1 Display Subsystem Interrupt and DMA Requests

Table 8-1 shows the interrupt and DMA requests generated by the DISPC and SD_DAC module.

Table 8-1. Display Subsystem Hardware Requests

Interrupt Requests				
Module Instance	**Source Signal Name**	**Destination IRQ_CROSSBAR Input**	**Default Mapping**	**Description**
DISPC	DISPC_IRQ	IRQ_CROSSBAR_20	DSP1_IRQ_51 DSP2_IRQ_51 IPU1_IRQ_23	DISPC interrupt request
SD_DAC	SD_DAC_IRQ_TVINT	IRQ_CROSSBAR_184	-	Interrupt for TV load detection from SD_DAC
	SD_DAC_IRQ_TVSHORT	IRQ_CROSSBAR_185	-	Interrupt for TV short detection from SD_DAC

Table 8-1. Display Subsystem Hardware Requests (continued)

DMA Requests				
Module Instance	**Source Signal Name**	**Destination DMA_CROSSBAR Input**	**Default Mapping**	**Description**
DISPC	DISPC_DREQ	DMA_CROSSBAR_6	DMA_EDMA_DREQ_5	The line trigger signal used for synchronization of a logical channel in the device-level EDMA for memory-to-memory transfer, generated by the DISPC module. There is no data received by the DISPC based on the DMA request.

NOTE: The "**Default Mapping**" column in Table 8-1 *Display Subsystem Hardware Requests* shows the default mapping of module IRQ and DREQ source signals. These module IRQ and DREQ source signals can also be mapped to other lines of each device Interrupt or DMA controller through the IRQ_CROSSBAR and DMA_CROSSBAR modules, respectively.
For more information about the IRQ_CROSSBAR module, see Section 13.4.6.4, *IRQ_CROSSBAR Module Functional Description*, in Chapter 13, *Control Module*.
For more information about the DMA_CROSSBAR module, see Section 13.4.6.5, *DMA_CROSSBAR Module Functional Description*, in Chapter 13, *Control Module*.
For more information about the device interrupt controllers, see Chapter 12, *Interrupt Controllers*.
For more information about the device EDMA module, see Chapter 11, *Enhanced DMA*.

8.1.2.2 Display Subsystem Clocks

The power, reset, and clock management (PRCM) module provides the clock signals, and status for their activity, to the display subsystem.

Figure 8-4 shows the details of the display subsystem clock tree.

Figure 8-4. Display Subsystem Clock Tree

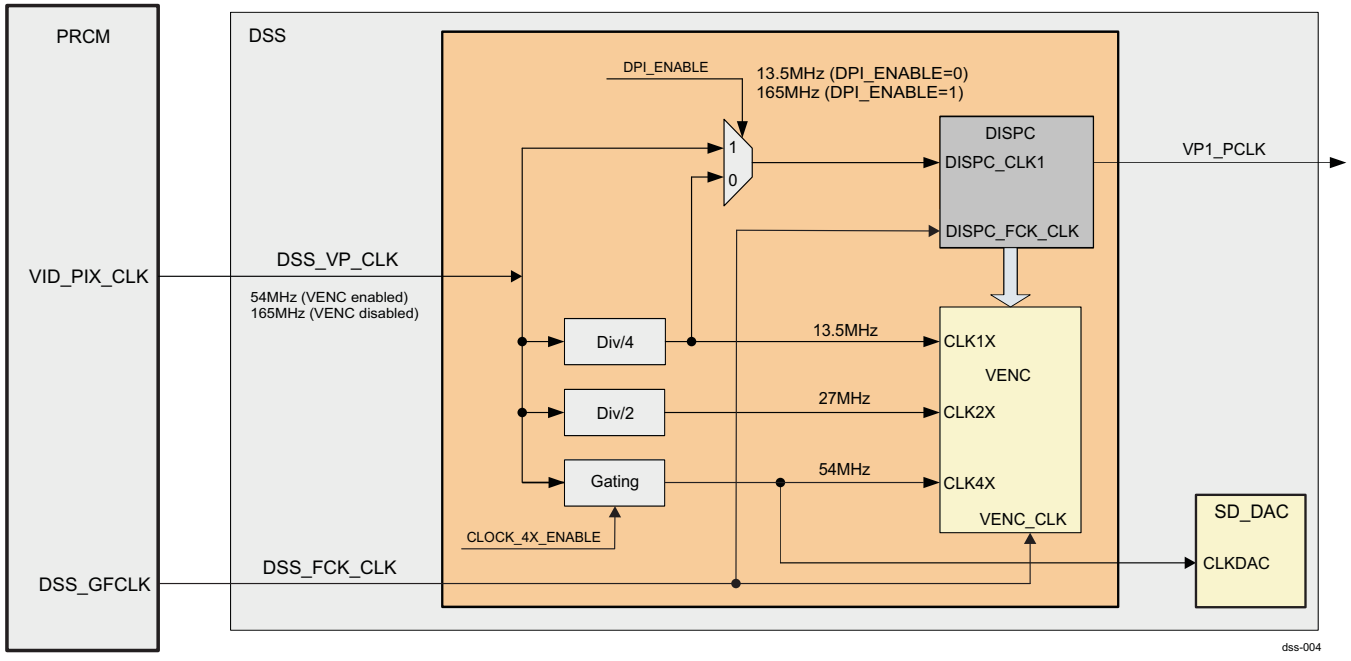


Table 8-2 lists the main DSS clocks and their sources.

Table 8-2. Display Subsystem Clocks

Clocks				
Module Instance	**Destination Signal Name**	**Source Signal Name**	**Source**	**Description**
DSS	DSS_VP_CLK	VID_PIX_CLK	PRCM module	Functional clock used for the VP pixel clock, Sync buffer and Timing generator modules within DISPC, and VENC/SD_DAC modules
	DSS_FCK_CLK	DSS_GFCLK	PRCM module	DISPC functional and interface clock; VENC interface clock

The DPI_ENABLE signal in Figure 8-4 is driven by DSS_DPI_CTRL[0] DPI_ENABLE register bit.

The gating of VENC clock CLK4X is controlled via DSS_VENC_CTRL[0] VENC_CLOCK_4X_ENABLE bit.

8.1.2.3 Display Subsystem Reset

The PRCM module provides one HW reset signal (DSS_RST) to the display subsystem, which is propagated to the DISPC, VENC and SD_DAC modules.

The DSS submodules can also be reset by a corresponding SW reset on submodule level.

The DSS_SYSSTATUS[0] DSS_RESETDONE register bit indicates the completion of the DISPC reset, following a HW or SW reset.

In order VENC reset status to be indicated in the DSS_SYSSTATUS[6] VENC_RESETDONE bit, the VENC functional clock (VENC_CLK) must be running.

8.1.2.4 Display Subsystem Power Management

The display subsystem and submodules are in the PD_DSS power domain.

The DSS supports no-idle mode, force-idle mode, and smart-idle modes. The mode can be selected by programming the appropriate value in the DSS_SYSCONFIG[1:0] SIDLEMODE register bitfield.

8.1.2.4.1 Display Subsystem Standby Mode

As part of the system-wide power-management scheme, the display subsystem supports the MStandby/MWait and SIdleReq/SIdleAck protocols:

- MStandby/MWait
  - DISPC
- SIdleReq/SIdleAck
  - DISPC

The PRCM module asserts the MWait and receives MStandby directly from DISPC. When the display subsystem initiates a standby procedure, it also initiates a master standby/wait protocols with the PRCM module that lets the PRCM cut the display subsystem clocks. For information about the conditions that allow the subsystem to exit standby mode, see Section 8.2.4.3.2 *DISPC StandBy Mode*, in *Section 8.2 Display Controller*.

The PRCM also asserts the SIdleReq. Then, it is split at the display subsystem level and send to the appropriate modules. Consequently, all SIdleAck are merged into one and sent back to PRCM.

8.1.2.4.2 Display Subsystem Wake-Up Mode

The DISPC module supports the wake-up protocol. DSS_SWAKEUP[1:0] signals are associated with DISPC_DREQ and DISPC_IRQ. For the events that generate an Swakeup and the description and configuration of the registers,see Section 8.2.4.3.3 *DISPC StandBy Mode*, in *Section 8.2 Display Controller*..

8.1.3 Display Subsystem Register Manual

8.1.3.1 Display Subsystem Instance Summary

Table 8-3. Display Subsystem Instance Summary

Module Name	L3_MAIN Base Address	Size
DSS	0x5800 0000	1 KiB

8.1.3.2 Display Subsystem Registers

8.1.3.2.1 Display Subsystem Registers Mapping Summary

Table 8-4 summarizes the display subsystem register mapping.

Table 8-4. DSS Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	L3_MAIN Physical Address
DSS_REVISION	R	32	0x0000 0000	0x5800 0000
DSS_SYSCONFIG	RW	32	0x0000 0010	0x5800 0010
DSS_SYSSTATUS	R	32	0x0000 0014	0x5800 0014
DSS_VENC_CTRL	RW	32	0x0000 0018	0x5800 0018
DSS_DPI_CTRL	RW	32	0x0000 001C	0x5800 001C
DSS_DEBUG_CFG	RW	32	0x0000 0040	0x5800 0040

8.1.3.2.2 Display Subsystem Register Description

Table 8-5. DSS_REVISION

Address Offset	0x0000 0000		
Physical Address	0x5800 0000	**Instance**	DSS
Description	This register contains the DSS revision number.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REVISION																															

Bits	Field Name	Description	Type	Reset
31:0	REVISION	IP revision	R	See (1)

(1) TI internal data

Table 8-6. Register Call Summary for Register DSS_REVISION

Display Subsystem Overview
- Display Subsystem Registers Mapping Summary: [0]

Table 8-7. DSS_SYSCONFIG

Address Offset	0x0000 0010		
Physical Address	0x5800 0010	**Instance**	DSS
Description	This register controls the various parameters of the OCP interface		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED															RESERVED	RESERVED											RESERVED	RESERVED		SIDLEMODE	

Bits	Field Name	Description	Type	Reset
31:17	RESERVED		R	0x0
16	RESERVED		R	0x0
15:5	RESERVED		R	0x0
4	RESERVED		R	0x0
3:2	RESERVED		R	0x0
1:0	SIDLEMODE	0x0: Force-idle.An idle request is acknowledged unconditionally 0x1: No-idle. An idle request is never acknowledged 0x3: Reserved 0x2: Smart-idle. Acknowledgement to an idle request is given based on the internal activity of the module.	RW	0x2

Table 8-8. Register Call Summary for Register DSS_SYSCONFIG

Display Subsystem Overview
- Display Subsystem Power Management: [0]
- Display Subsystem Registers Mapping Summary: [1]

Table 8-9. DSS_SYSSTATUS

Address Offset	0x0000 0014		
Physical Address	0x5800 0014	**Instance**	DSS
Description	This register provides status information about the module.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															RESETDONE

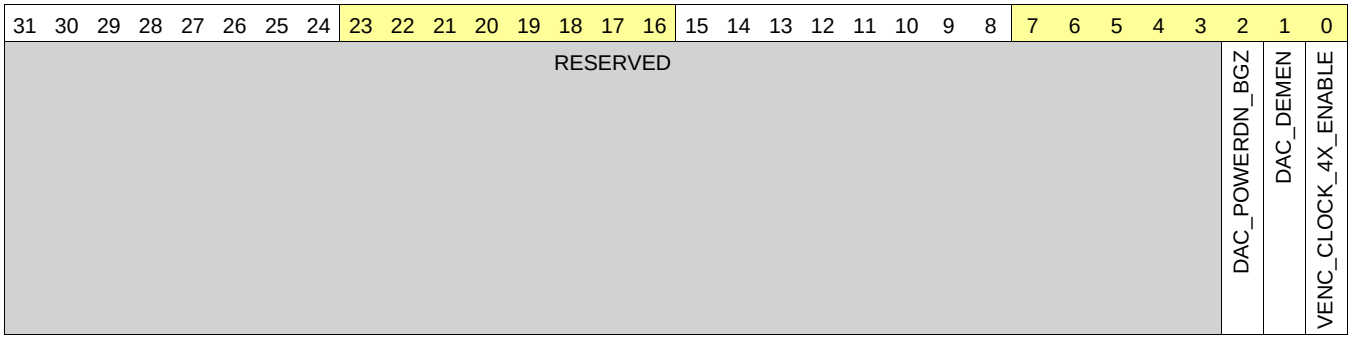
Bits	Field Name	Description	Type	Reset
31:7	RESERVED		R	0x0
6	VENC_RESETDONE	Reset status of VENC module 0x0: Internal module reset is on-going 0x1: Reset completed	R	0x0
5	RESERVED		R	0x0
4	RESERVED		R	0x0
3	RESERVED		R	0x0
2	RESERVED		R	0x0
1	RESERVED		R	0x0
0	DSS_RESETDONE	Reset status of DISPC/DSS 0x0: Internal module reset is on-going 0x1: Reset completed	R	0x0

Table 8-10. Register Call Summary for Register DSS_SYSSTATUS

Display Subsystem Overview
• Display Subsystem Reset: [0][1]
• Display Subsystem Registers Mapping Summary: [2]

Table 8-11. DSS_VENC_CTRL

Address Offset	0x0000 0018		
Physical Address	0x5800 0018	**Instance**	DSS
Description	This register contains control bits corresponding to the VENC instance in DSS		
Type	RW		



Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0
2	DAC_POWERDN_BGZ	DAC power down band gap control 0x0: DAC power down band gap disabled 0x1: DAC power down band gap enabled	RW	0x0
1	DAC_DEMEN	DAC Dynamic Element Matching Enable 0x0: DAC Dynamic Element Matching disabled 0x1: DAC Dynamic Element Matching enabled	RW	0x0
0	VENC_CLOCK_4X_ENABLE	VENC Clock CLK4X Enable. This bit is used to control the CLK4X clock gating. 0x0: Disable 0x1: Enable	RW	0x0

Table 8-12. Register Call Summary for Register DSS_VENC_CTRL

Display Subsystem Overview
• Display Subsystem Clocks: [0]
• Display Subsystem Registers Mapping Summary: [1]

Table 8-13. DSS_DPI_CTRL

Address Offset	0x0000 001C		
Physical Address	0x5800 001C	**Instance**	DSS
Description	This register contains control bits corresponding to the DPI interface in DSS		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															DPI_ENABLE

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	DPI_ENABLE	Enable DPI interface. This bit is used to enable the DPI interface. 0x0: Disable - PCLK is gated 0x1: Enable	RW	0x1

Table 8-14. Register Call Summary for Register DSS_DPI_CTRL

Display Subsystem Overview
- Display Subsystem Environment: [0][1]
- Display Subsystem Clocks: [2]
- Display Subsystem Registers Mapping Summary: [3]

Table 8-15. DSS_DEBUG_CFG

Address Offset	0x0000 0040		
Physical Address	0x5800 0040	**Instance**	DSS
Description	Debug configuration		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														CFG	

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0
2:0	CFG	Defines which debug bus to provide on the DSS debug bus connected at the top. Only values 0, 1, 2, 3, 4, and 5 can be used. 0x0: select DISPC Debug bus **The following values are not supported on device level, as the features listed are not supported in this family of devices.** 0x1: select DSI1_A Debug bus 0x2: select DSI1_B Debug bus 0x3: select DP_A Debug bus 0x4: select DP_B Debug bus 0x5: select HDMI Debug bus	RW	0x0

Table 8-16. Register Call Summary for Register DSS_DEBUG_CFG

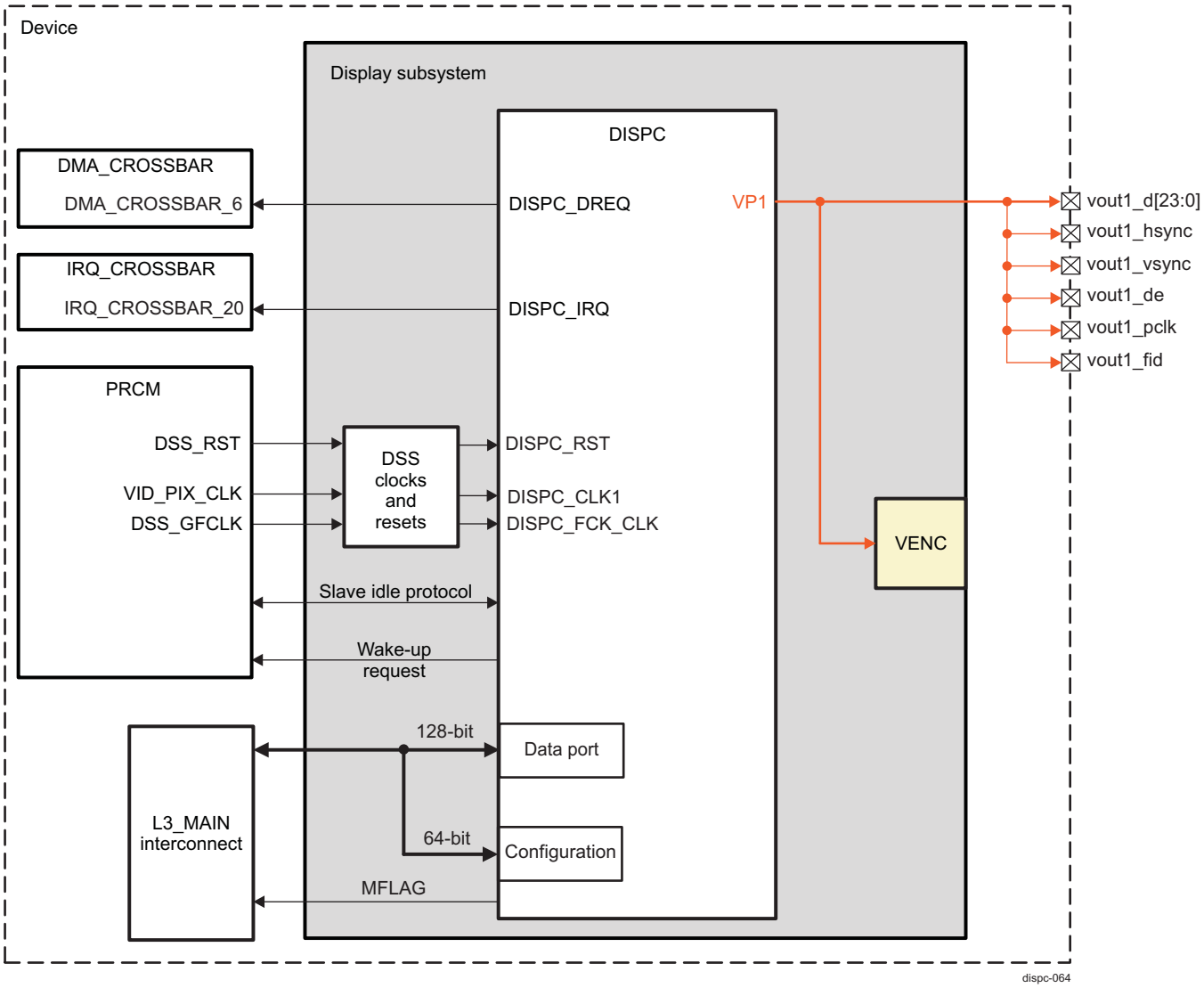
Display Subsystem Overview
- Display Subsystem Registers Mapping Summary: [0]

8.2 Display Controller

8.2.1 DISPC Overview

Figure 8-5 shows a block diagram of the display controller (DISPC) within the display subsystem.

Figure 8-5. DISPC Overview



The DISPC includes the following main features:

- Pipelines for processing:
  - One Graphics (GFX) pipeline:
    - Pixel formats: ARGB16-4444, ABGR16-4444, RGBA16-4444, RGB16-565, BGR16-565, ARGB16-1555, ABGR16-1555, ARGB32-8888, ABGR32-8888, BGRA32-8888, RGBA32-8888, RGB24-888, ARGB32-2101010, ABGR32-2101010, ARGB48-12121212, RGBA48-12121212, ARGB64-16161616, RGBA64-16161616, BITMAP1, BITMAP2, BITMAP4, BITMAP8. (A component can be ignored by not selecting Alpha pixel)
    - Premultiplied ARGB and RGBA formats
    - Selection of the color depth expansion from ARGB16-4444, RGBA16-4444, and ARGB16-1555 to ARGB32-8888, and from xRGB12-4444, RGBx12-4444, and xRGB16-1555 to xRGB32-8888 (replication of the most significant bits [MSBs])

- Support for color look-up table (CLUT): 256 × 24-bit entries palette in RGB
- Support for anti-aliasing on RGB pixel formats using 3-tap filter
- Up to 4K-by-4K frame buffer size

– Two Video pipelines (VID1 and VID2):
  - Pixel formats: ARGB16-4444, ABGR16-4444, RGBA16-4444, RGB16-565, BGR16-565, ARGB16-1555, ABGR16-1555, ARGB32-8888, ABGR32-8888, BGRA32-8888, RGBA32-8888, RGB24-888, ARGB32-2101010, ABGR32-2101010, ARGB48-12121212, RGBA48-12121212, ARGB64-16161616, RGBA64-16161616, RGB565-A8, BITMAP1, BITMAP2, BITMAP4, BITMAP8, YUV422-UYVY, YUV422-YUV2, YUV420-NV12, YUV420-NV21 and in addition RGBx, xRGB, xBGR, and BGRx pixel formats defined considering that A component of RGBA, ARGB, ABGR, BGRA pixel formats is ignored by HW (e.g. ARGB->xRGB)
  - Premultiplied ARGB and RGBA formats
  - Support for color look-up table (CLUT): 256 × 24-bit entries palette in RGB
  - Programmable poly-phase filter:
    - Independent horizontal and vertical resampling. Upsampling up to x16, and downsampling down to 1/4
    - Maximum input width of 1280 pixels (2560 in 3-tap mode)
    - No limitation on the input height
    - Supported input formats are all above listed formats, except BITMAP formats. The alpha blending factor is rescaled like the R, G and B color components. 16 phases with symmetrical coefficients are implemented.
  - Programmable color space conversion from YUV4:2:2 (YUV4:4:4, YUV4:2:0 after Chroma upsampling through the scaler) into ARGB48-12121212.
  - Programmable VC-1 range mapping
  - Up to 4K-by-4K frame buffer size

– One Write-back (WB) pipeline: Allows the use of the hardware processing available inside the DISPC, such as color space conversion, rescaling, and compositing to perform memory-to-memory transfer with data processing or capturing a displayed frame
  - Pixel formats: ARGB16-4444, ABGR16-4444, RGBA16-4444, RGB16-565, BGR16-565, ARGB16-1555, ABGR16-1555, ARGB32-8888, ABGR32-8888, BGRA32-8888, RGBA32-8888, RGB24-888, ARGB32-2101010, ABGR32-2101010, ARGB48-12121212, RGBA48-12121212, ARGB64-16161616, RGBA64-16161616, RGB565-A8, YUV422-UYVY, YUV422-YUV2, YUV420-NV12, YUV420-NV21 and in addition RGBx, xRGB, xBGR, and BGRx pixel formats defined considering that A component of RGBA, ARGB, ABGR, BGRA pixel formats is ignored by HW (e.g. ARGB->xRGB)
  - Programmable color space conversion RGB24 into YUV4:2:2-UYVY, YUV4:2:2-YUV2, YUV4:2:0-NV12, or YUV4:2:0-NV21
  - Programmable poly-phase filter:
    - Independent horizontal and vertical resampling. Upsampling up to x16, and downsampling down to 1/4. When the output format of WB pipeline includes a format change from RGB or YUV4:2:2 to YUV4:2:0, the maximum downscaling provided by the WB scaler is 1/2.
    - Maximum input width of 1280 pixels using 32-bit pixels and 5-tap, and 2560 pixels using 32-bit pixels and 3-tap
    - No limitation on the input height
    - The supported formats are RGB888/RGBA-8888/ARGB32-8888, ARGB48-12121212, RGB565-A8, YUV422-UYVY, YUV422-YUV2, YUV420-NV12 and YUV420-NV21
    - Alpha blending factor is rescaled like the R, G, and B color components
    - 16 phases with symmetrical coefficients are implemented
  - Selection of the source of the data:
    - One of the overlay manager outputs
    - One of the pipeline outputs

- One Video Port (VP1) output:
  - Up to 24-bit per pixel on the output interface, with selection between 12, 16, 18 or 24-bit
  - Independent programmable timing generator to support up to 165 MHz pixel clock video formats defined in CEA-861-E and VESA-DMT standards
  - Independent programmable multiple cycles output format on 8/9/12/16-bit interface (TDM)
  - Configurable output mode: progressive or interlaced mode
  - Selection between RGB and YUV422 output pixel format (YUV4:2:2 only available when BT output is enabled)
- Two Overlay Managers (OVR1 and OVR2):
  - Input Pixel format: ARGB48-12121212
  - Output Pixel format: ARGB48-12121212 (A component is only used for Write-back)
  - Overlay of the graphics and video pipelines
  - Transparency color keys (source and destination)
  - Global and pixel alpha blending
  - Programmable Z-order (full flexibility)
- Common:
  - Programmable 8-bit gamma curve support on VP1 output
  - Programmable color phase rotation (CPR)
  - Alpha blending support:
    - Embedded pixel factor (ARGB and RGBA)
    - Global pixel
    - Combination of global pixel and pixel factor
- DMA (internal to the DISPC):
  - No support for rotation, flip-flop, mirroring and memory fragmentation
  - Integrated shared buffers between DMA engine and pipelines
  - Programmable buffer thresholds
  - Bandwidth limiter on write request (insertion on idle cycles between requests)
- Advanced:
  - Mode outputting data on display only from the DMA buffer (self-refresh using the DMA FIFO)
  - Arbitration between normal and low priority pipelines (GFX, VID1, VID2 and WB pipelines)
- Power modes:
  - Low-power saving modes
  - Support on-the-fly dynamic voltage and frequency scaling (DVFS)
  - Merge capability of the DMA buffers to support greater OFF period on the L3_MAIN interconnect
    - All buffers associated to a single pipeline
    - Reallocation of the buffers of the nonactive pipelines to the active pipelines

NOTE: The display resolution is programmable and can be any width in the range [1:4096] pixels. The following limitations, related to the type of display or the processing done, apply:

- Active Matrix screen + dithering forces a width multiple of 2 pixels
- Active Matrix + TDM may force a width multiple of 2 pixels

The display buffers in the system memory consist of contiguous pixels.

8.2.2 DISPC Environment

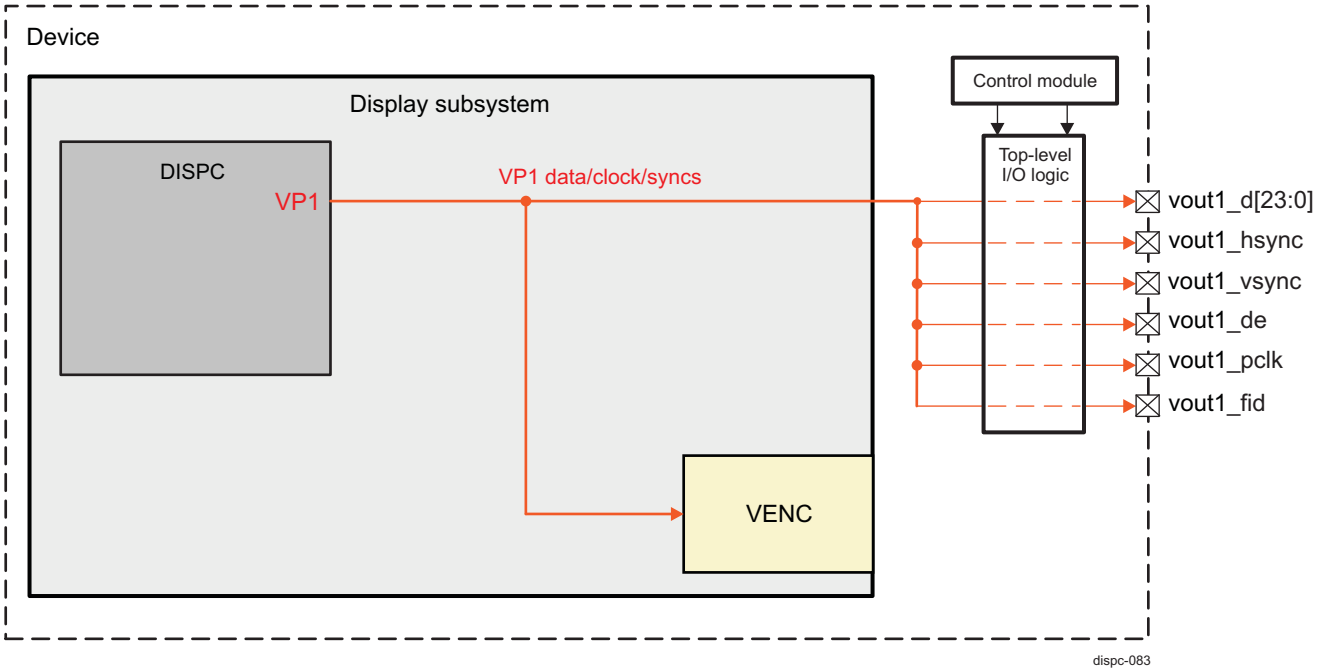
The DISPC VP1 provides the required data and control signals to interface:

- Directly to device CMOS pads for MIPI DPI 2.0, or BT.656 or BT.1120 interface support
- Video Encoder (VENC) module with dedicated standard definition video DAC (SD_DAC) for composite TV out support

Figure 8-6 shows the supported interfaces.

NOTE: The parallel and TV out interfaces cannot be used simultaneously. The selection can be done at DSS top level. For more information, see Section 8.1.1, *Display Subsystem Environment*.

Figure 8-6. DISPC VP1 Output Interfaces



NOTE: The path from a module pin to device pad (or pads) is defined at the device I/O logic level. The I/O logic maps the module signals to the different pads of the device and is programmable in the control module registers and dedicated IP registers. For more information, see Section 13.4.6.1, Pad Configuration Registers, in Chapter 13 *Control Module*.

Table 8-17 describes the VP1 interface signals.

Table 8-17. DISPC VP1 Interface Signals

Signal Name	Type[1]	Description
vout1_d[23:0]	O	Pixel data
vout1_pclk	O	Pixel clock
vout1_vsync	O	Vertical synchronization. The frame synchronization pulse (vsync) toggles after all the lines in a frame are transmitted and a programmable number of line clock cycles has elapsed at the beginning and the end of each frame.

(1) I = Input, O = Output, I/O = Input/Output

Table 8-17. DISPC VP1 Interface Signals (continued)

Signal Name	Type[1]	Description
vout1_hsync	O	Horizontal synchronization. The line synchronization pulse (hsync) toggles after all pixels in a line are transmitted and a programmable number of pixel clock wait-states has elapsed at the beginning and the end of each line.
vout1_de	O	Pixel data output-enable signal to indicate when data must be latched using the pixel clock.
vout1_fid	O	The FID signal indicates the output field identifier: • 0 means even • 1 means odd The FID signal is 0 when the progressive mode is selected for the VP output.

8.2.2.1 DISPC VP1 Output and Data Formats

This section describes the pixel data bus and shows timing diagrams of transactions and synchronizations.

In the Active matrix display type, one pixel per pixel clock is displayed. The diagrams represent the configuration of assertion of the data on the rising edge of the pixel clock. It is possible to program the interface timing to output the data on the falling edge of the pixel clock.

Active matrix displays bypass the STN dithering logic block and the output FIFO. Each line represents one pixel.Figure 8-7 through Figure 8-10 show 12-, 16-, 18-, and 24-active matrix displays, respectively. The width of the data bus can be configured through DISPC_VP1_CONTROL[10:8] DATALINES register bitfield.

Figure 8-7. DISPC VP1 Pixel Data Color-12 Active Matrix

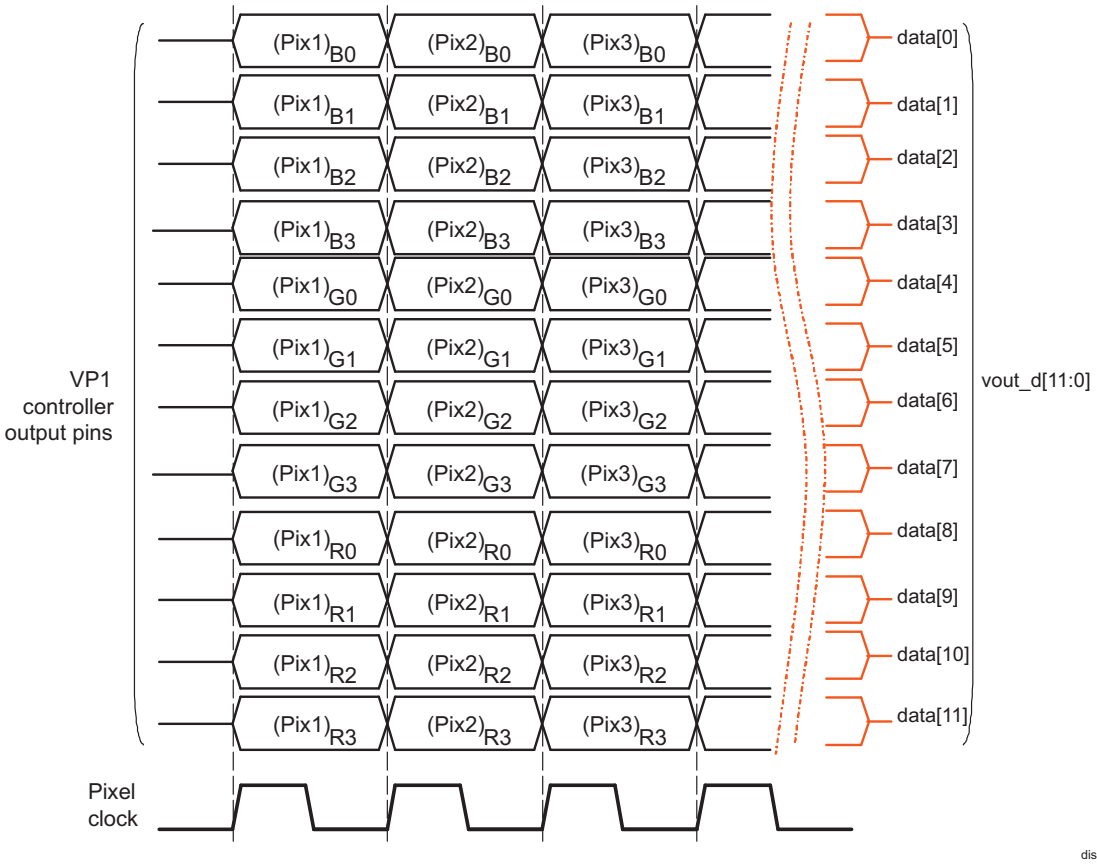


Figure 8-8. DISPC VP1 Pixel Data Color-16 Active Matrix

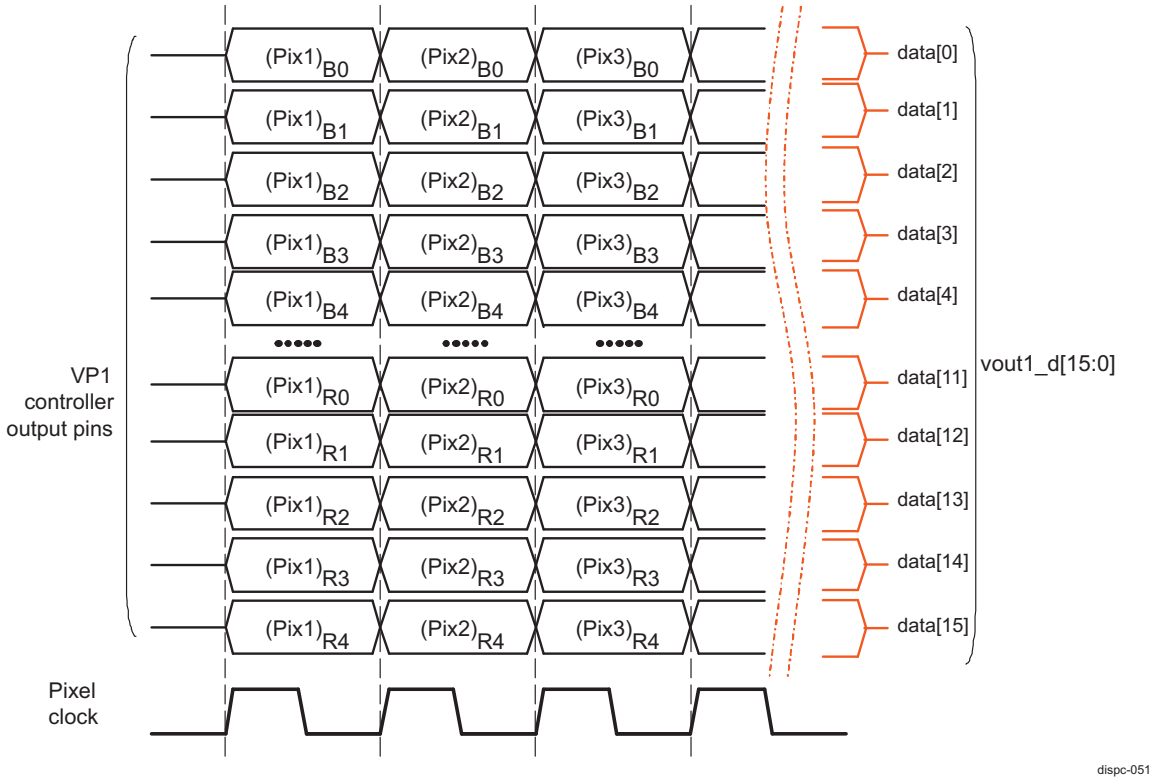


Figure 8-9. DISPC VP1 Pixel Data Color-18 Active Matrix

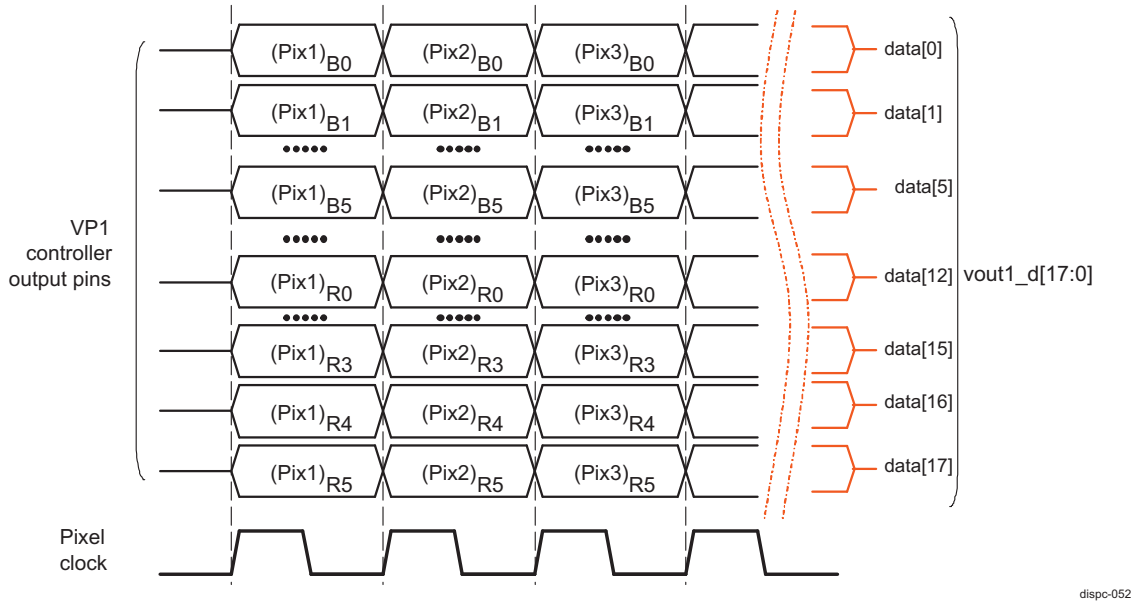


Figure 8-10. DISPC VP1 Pixel Data Color-24 Active Matrix

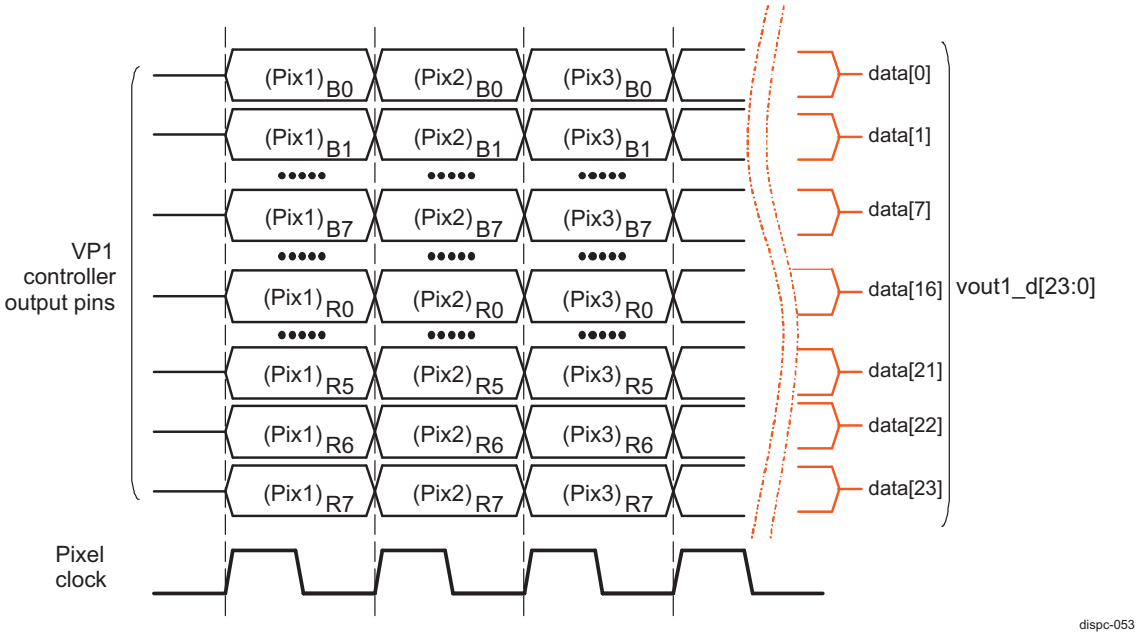


Table 8-18 summarizes the mapping of RGB color components to DSS output data signals, with corresponding settings of DISPC_VP1_CONTROL[10:8] DATALINES register bit-field.

Table 8-18. DSS Output Data Signals to RGB Color Components Mapping

Color Assignment	12-bit Output Mode	16-bit Output Mode	18-bit Output Mode	24-bit Output Mode
	DATALINES=00b	DATALINES=01b	DATALINES=10b	DATALINES=11b
Red 7 (MS bit)	-	-	-	vout1_d23
Red 6	-	-	-	vout1_d22
Red 5	-	-	vout1_d17	vout1_d21
Red 4	-	vout1_d15	vout1_d16	vout1_d20
Red 3	vout1_d11	vout1_d14	vout1_d15	vout1_d19
Red 2	vout1_d10	vout1_d13	vout1_d14	vout1_d18
Red 1	vout1_d9	vout1_d12	vout1_d13	vout1_d17
Red 0	vout1_d8	vout1_d11	vout1_d12	vout1_d16
Green 7	-	-	-	vout1_d15
Green 6	-	-	-	vout1_d14
Green 5	-	vout1_d10	vout1_d11	vout1_d13
Green 4	-	vout1_d9	vout1_d10	vout1_d12
Green 3	vout1_d7	vout1_d8	vout1_d9	vout1_d11
Green 2	vout1_d6	vout1_d7	vout1_d8	vout1_d10
Green 1	vout1_d5	vout1_d6	vout1_d7	vout1_d9
Green 0	vout1_d4	vout1_d5	vout1_d6	vout1_d8
Blue 7	-	-	-	vout1_d7
Blue 6	-	-	-	vout1_d6
Blue 5	-	-	vout1_d5	vout1_d5
Blue 4	-	vout1_d4	vout1_d4	vout1_d4
Blue 3	vout1_d3	vout1_d3	vout1_d3	vout1_d3
Blue 2	vout1_d2	vout1_d2	vout1_d2	vout1_d2
Blue 1	vout1_d1	vout1_d1	vout1_d1	vout1_d1
Blue 0 (LS bit)	vout1_d0	vout1_d0	vout1_d0	vout1_d0

8.2.2.2 DISPC VP1 Active Marix Display Timing Diagrams

Figure 8-11 through Figure 8-14 show timing diagrams of synchronization signals and pixel clocks for active matrix panels. The DISPC directly drives these signals, which are related to the programmable fields listed in Table 8-19.

Table 8-19. DISPC VP1 Programmable Fields for Active Matrix Display

Name	Register	Description
PPL	DISPC_VP1_SIZE_SCREEN[11:0] PPL value + 1	Pixels per line
LPP	DISPC_VP1_SIZE_SCREEN[27:16] LPP value + 1	Lines per panel
HBP	DISPC_VP1_TIMING_H[31:20] HBP value + 1	Horizontal back porch
HFP	DISPC_VP1_TIMING_H[19:8] HFP value + 1	Horizontal front porch
HSW	DISPC_VP1_TIMING_H[7:0] HSW value + 1	Horizontal synchronization pulse width
VBP	DISPC_VP1_TIMING_V[31:20] VBP value	Vertical back porch
VFP	DISPC_VP1_TIMING_V[19:8] VFP value	Vertical front porch
VSW	DISPC_VP1_TIMING_V[7:0] VSW value + 1	Vertical synchronization pulse width
ONOFF(1)	DISPC_VP1_POL_FREQ[17] ONOFF	DISPC_HSYNC and DISPC_VSYNC pixel clock control
RF(2)	DISPC_VP1_POL_FREQ[16] RF	DISPC_HSYNC and DISPC_VSYNC pixel clock edge control
IEO	DISPC_VP1_POL_FREQ[15] IEO	Invert output enable
IPC(3)	DISPC_VP1_POL_FREQ[14] IPC	Invert DISPC_PCLK
IHS	DISPC_VP1_POL_FREQ[13] IHS	Invert DISPC_HSYNC
IVS	DISPC_VP1_POL_FREQ[12] IVS	Invert DISPC_VSYNC

(1) DISPC_VP1_POL_FREQ[17] ONOFF and Control Module register CTRL_CORE_SMA_SW_1[]DSS_CH0_ON_OFF must match.
(2) DISPC_VP1_POL_FREQ[16] RF and Control Module register CTRL_CORE_SMA_SW_1[]DSS_CH0_RF must match.
(3) DISPC_VP1_POL_FREQ[14] IPC and Control Module register CTRL_CORE_SMA_SW_1[]DSS_CH0_IPC must match.

- Active matrix timing configuration 1:
  - DISPC_VP1_POL_FREQ[17] ONOFF = 0 and CTRL_CORE_SMA_SW_1[]DSS_CH0_ON_OFF = 0
  - DISPC_VP1_POL_FREQ[16] RF = 0 and CTRL_CORE_SMA_SW_1[]DSS_CH0_RF = 0
    The HSYNC and VSYNC signals are driven on the opposite edge of PCLK from the pixel data.
  - DISPC_VP1_POL_FREQ[15] IEO = 0
    The DE signal is active high.
  - DISPC_VP1_POL_FREQ[14] IPC = 0 and CTRL_CORE_SMA_SW_1[]DSS_CH0_IPC = 0
    The pixel data are driven on the rising edge of PCLK.
  - DISPC_VP1_POL_FREQ[13] IHS = 0
    The HSYNC signal is active high.
  - DISPC_VP1_POL_FREQ[12] IVS = 0
    The VSYNC signal is active high.

Figure 8-11. DISPC Active Matrix Timing Diagram of Configuration 1 (Start of Frame)

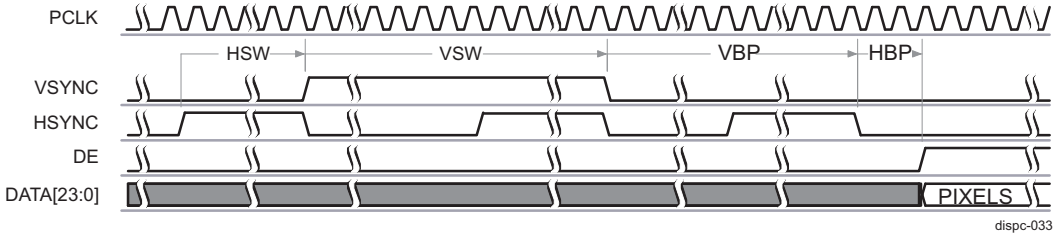


Figure 8-12. DISPC Active Matrix Timing Diagram of Configuration 1 (Between Lines)

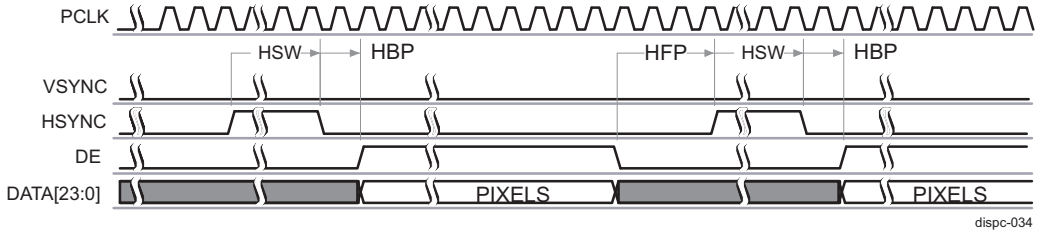


Figure 8-13. DISPC Active Matrix Timing Diagram of Configuration 1 (Between Frames)

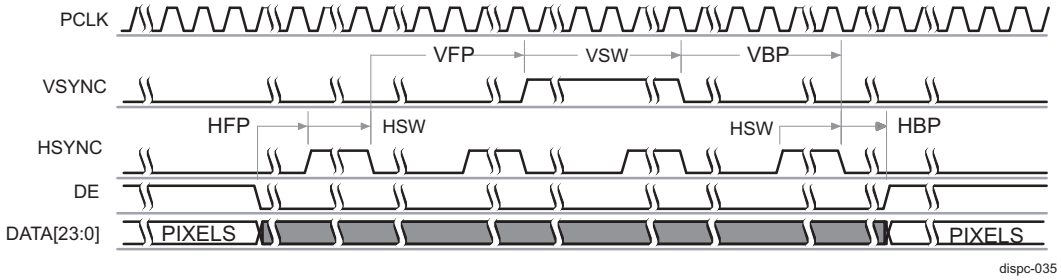
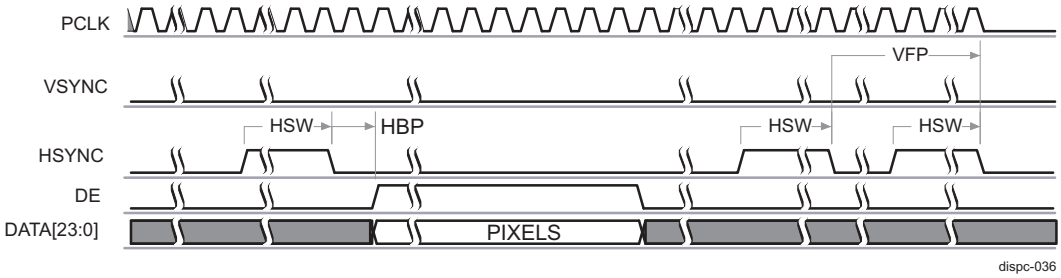


Figure 8-14. DISPC Active Matrix Timing Diagram of Configuration 1 (End of Frame)



- Active matrix timing configuration 2:
  - DISPC_VP1_POL_FREQ[17] ONOFF = 1 and CTRL_CORE_SMA_SW_1[]DSS_CH0_ON_OFF = 1
  - DISPC_VP1_POL_FREQ[16] RF = 1 and CTRL_CORE_SMA_SW_1[]DSS_CH0_RF = 1
    The HSYNC and VSYNC signals are driven on the rising edge of PCLK.
  - DISPC_VP1_POL_FREQ[15] IEO = 1
    The DE signal is active low.
  - DISPC_VP1_POL_FREQ[14] IPC = 1 and CTRL_CORE_SMA_SW_1[]DSS_CH0_IPC = 1
    The pixel data is driven on the falling edge of PCLK.
  - DISPC_VP1_POL_FREQ[13] IHS = 1
    The HSYNC signal is active low.
  - DISPC_VP1_POL_FREQ[12] IVS = 1
    The VSYNC signal is active low.

Figure 8-15. DISPC Active Matrix Timing Diagram of Configuration 2 (Start of Frame)

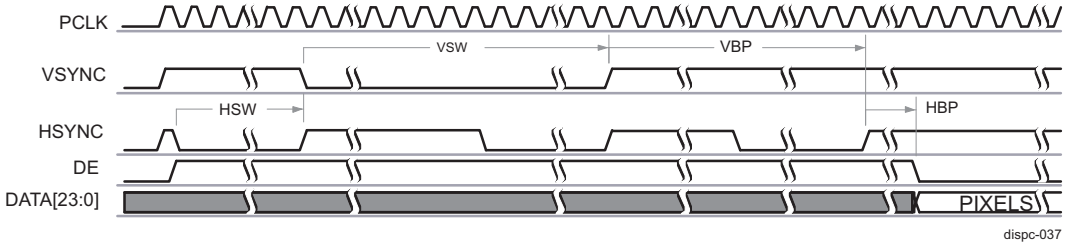


Figure 8-16. DISPC Active Matrix Timing Diagram of Configuration 2 (Between Lines)

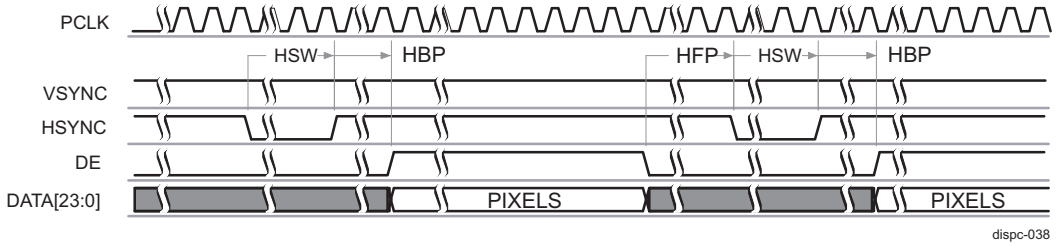


Figure 8-17. DISPC Active Matrix Timing Diagram of Configuration 2 (Between Frames)

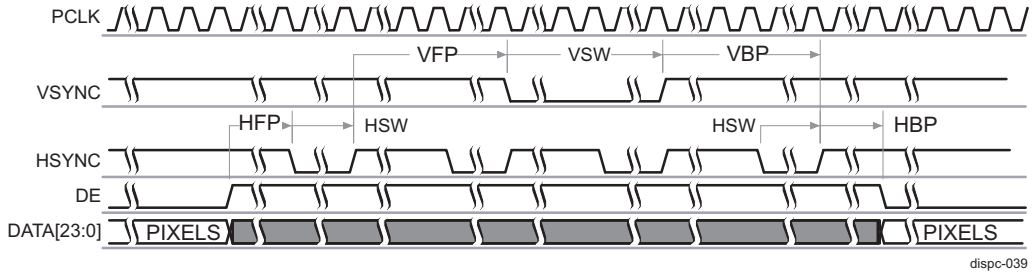
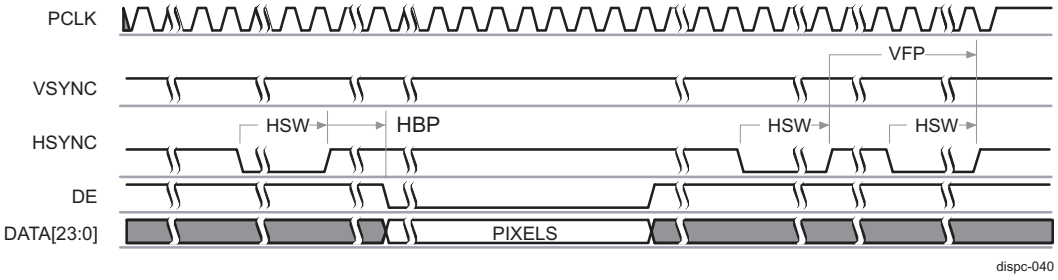


Figure 8-18. DISPC Active Matrix Timing Diagram of Configuration 2 (End of Frame)



- Active matrix timing configuration 3:
  - DISPC_VP1_POL_FREQ[17] ONOFF = 1 and CTRL_CORE_SMA_SW_1[]DSS_CH0_ON_OFF = 1
  - DISPC_VP1_POL_FREQ[16] RF = 1 and CTRL_CORE_SMA_SW_1[]DSS_CH0_RF = 1
    The HSYNC and VSYNC signals are driven on the rising edge of PCLK.
  - DISPC_VP1_POL_FREQ[15] IEO = 0
    The DE signal is active high.
  - DISPC_VP1_POL_FREQ[14] IPC = 0 and CTRL_CORE_SMA_SW_1[]DSS_CH0_IPC = 0
    The pixel data are driven on the rising edge of PCLK.
  - DISPC_VP1_POL_FREQ[13] IHS = 0
    The HSYNC signal is active high.

– DISPC_VP1_POL_FREQ[12] IVS = 0
The VSYNC signal is active high.

Figure 8-19. DISPC Active Matrix Timing Diagram of Configuration 3 (Start of Frame)

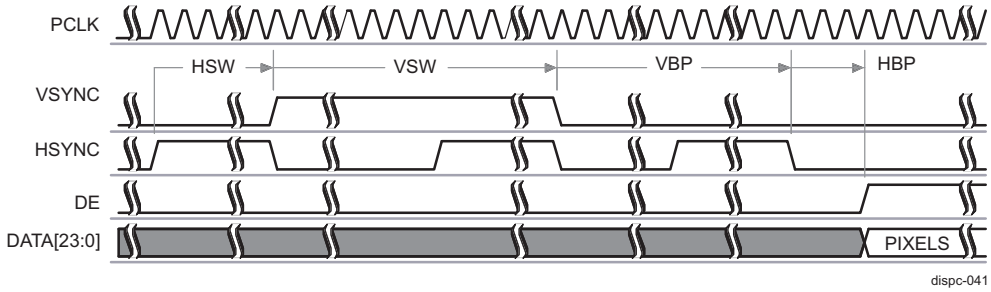


Figure 8-20. DISPC Active Matrix Timing Diagram of Configuration 3 (Between Lines)

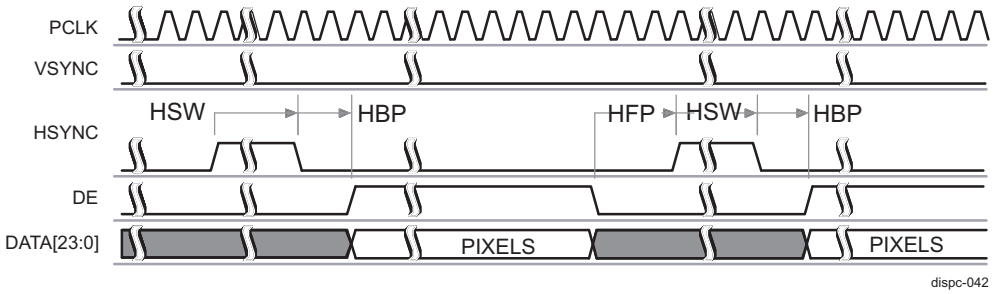


Figure 8-21. DISPC Active Matrix Timing Diagram of Configuration 3 (Between Frames)

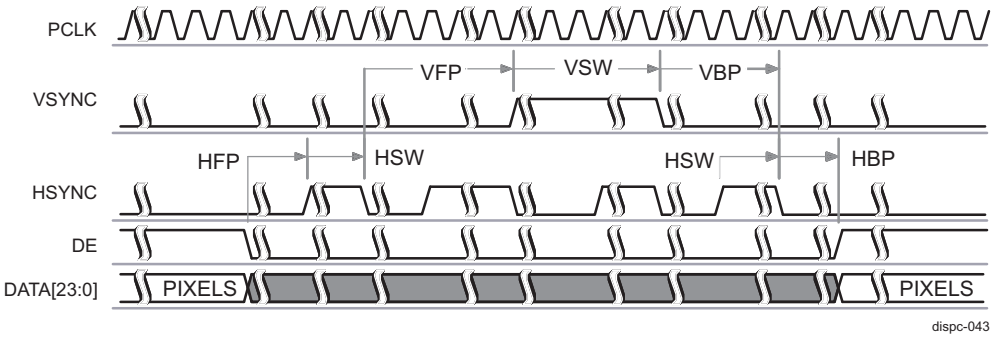
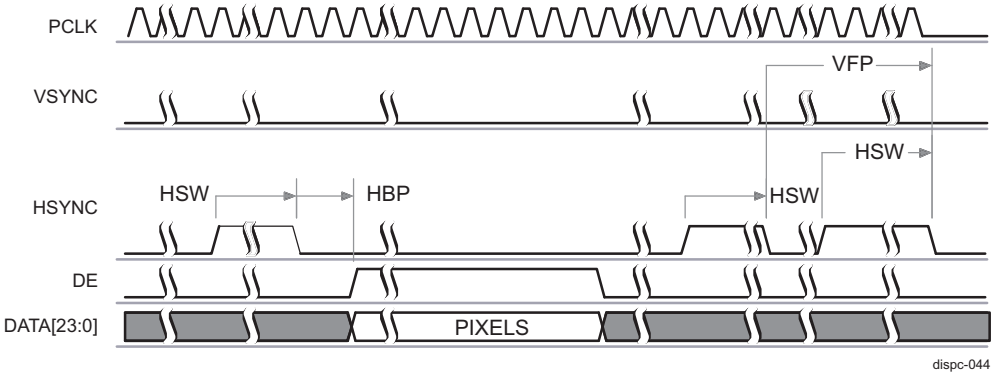


Figure 8-22. DISPC Active Matrix Timing Diagram of Configuration 3 (End of Frame)



8.2.3 DISPC Integration

This section describes the DISPC integration in the device (see Figure 8-23). For complete details about clocks and resets, see Section 8.1, *Display Subsystem Overview*.

Figure 8-23. DISPC Integration

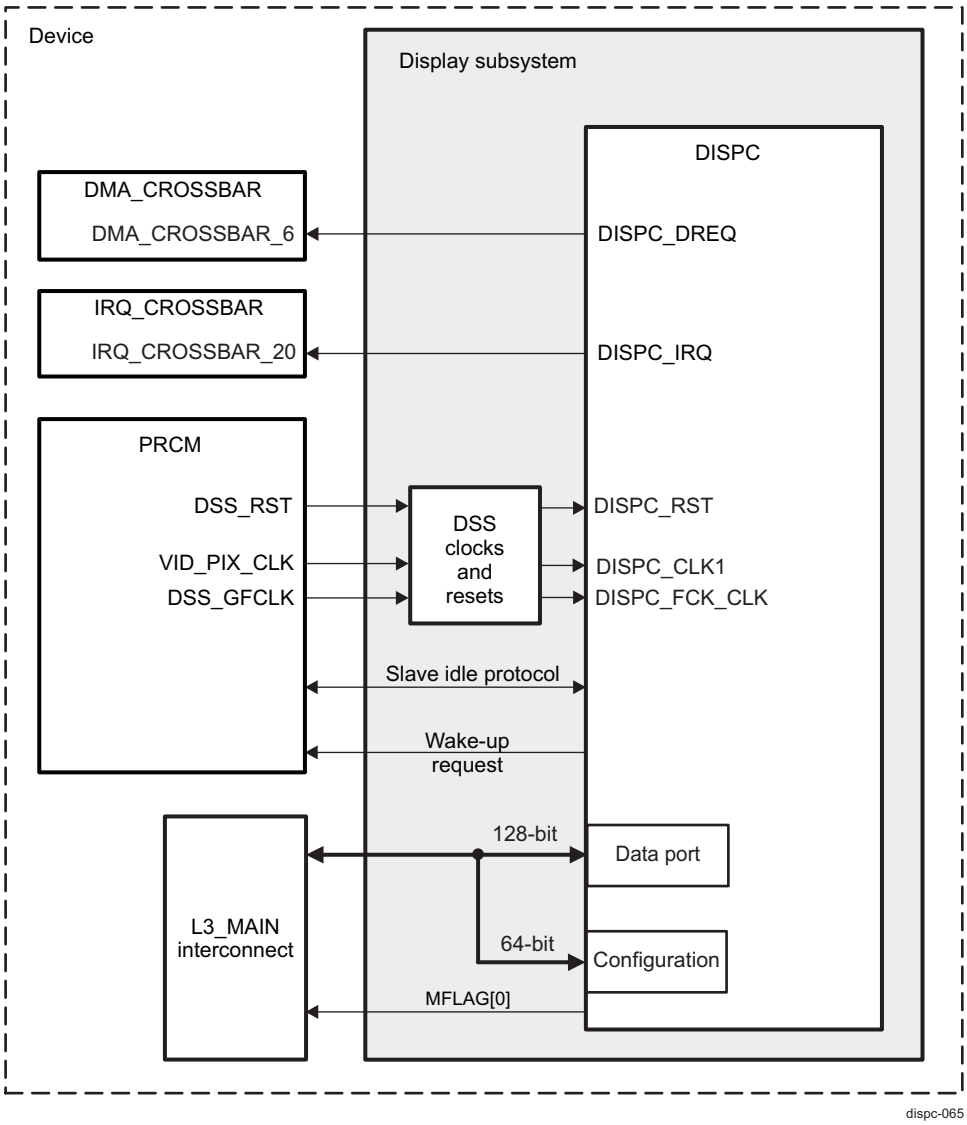


Table 8-20 and Table 8-21 list the integration attributes and clock and resets, respectively.

Table 8-20. DISPC Integration Attributes

Module Instance	Attributes	
	Power Domain	Interconnect
DISPC	PD_DSS	128-bit master port to L3_MAIN for data transfer to/from system memory 32-bit slave port to L3_MAIN for configuration

Table 8-21. DISPC Clocks and Resets

Clocks				
Module Instance	**Destination Signal Name**	**Source Signal Name**	**Source**	**Description**
DISPC	DISPC_FCK_CLK	DSS_GFCLK	PRCM	Functional clock for the internal DISPC logic. Also DISPC interface clock. For additional details, see Section 8.1.2, *Display Subsystem Integration*.
	DISPC_CLK1	VID_PIX_CLK	PRCM	Clock used to generate the pixel clock (VP1_PCLK) for the video port (VP1) interface. For additional details, see Section 8.1.2, *Display Subsystem Integration*.
Resets				
Module Instance	**Destination Signal Name**	**Source Signal Name**	**Source**	**Description**
DISPC	DISPC_RST	DSS_RST	PRCM	Hardware reset is coming from the PRCM module. For additional details, see Section 8.1.2, *Display Subsystem Integration*. **NOTE:** The DSS_RST signal is provided to the entire display subsystem. When inside the display subsystem boundaries, it is named DISPSS_RST, which on its end is provided to the DISPC and is named DISPC_RST.

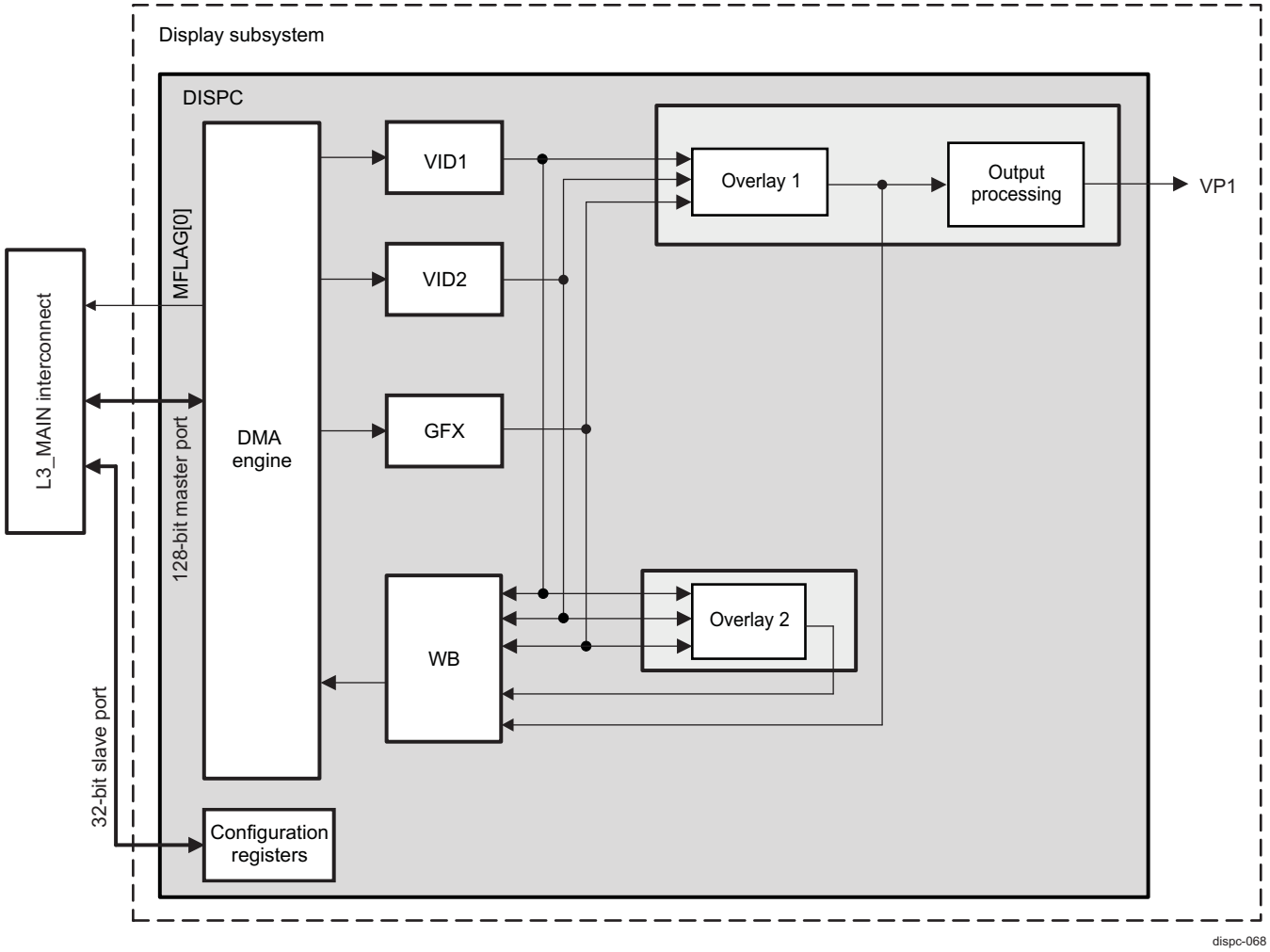
Table 8-22. DISPC Hardware Requests

Interrupt Requests				
Module Instance	**Source Signal Name**	**Destination IRQ_CROSSBAR Input**	**Default Mapping**	**Description**
DISPC	DISPC_IRQ	IRQ_CROSSBAR_20	DSP1_IRQ_51 DSP2_IRQ_51 IPU1_IRQ_23	DISPC interrupt request
DMA Requests				
Module Instance	**Source Signal Name**	**Destination DMA_CROSSBAR Input**	**Default Mapping**	**Description**
DISPC	DISPC_DREQ	DMA_CROSSBAR_6	DMA_EDMA_DREQ_5	Associated with the line trigger signal used for synchronization of a logical channel in the device-level EDMA for memory-to-memory transfer, generated by the DISPC module. There is no data received by the DISPC based on the DMA request. Indicates when the programmable line number matches the current primary VP1 panel refresh line number. See Section 8.2.4.5, *DISPC DMA Requests*, for more details.

NOTE: The "**Default Mapping**" column in Table 8-22 *DISPC Hardware Requests* shows the default mapping of module IRQ and DREQ source signals. These module IRQ and DREQ source signals can also be mapped to other lines of each device Interrupt or DMA controller through the IRQ_CROSSBAR and DMA_CROSSBAR modules, respectively.
For more information about the IRQ_CROSSBAR module, see Section 13.4.6.4, *IRQ_CROSSBAR Module Functional Description*, in Chapter 13, *Control Module*.
For more information about the DMA_CROSSBAR module, see Section 13.4.6.5, *DMA_CROSSBAR Module Functional Description*, in Chapter 13, *Control Module*.
For more information about the device interrupt controllers, see Chapter 12, *Interrupt Controllers*.
For more information about the device EDMA module, see Chapter 11, *Enhanced DMA*.

8.2.4 DISPC Functional Description

Figure 8-24. DISPC Architecture Overview



The DISPC can read and display the encoded pixel data stored in memory and write the output of one of the overlays or one of the pipelines into system memory.

Several processes can be configured to manage the graphics pipeline (antialiasing) and video pipelines (VC-1, color space conversion, scaling, and so forth).

The data coming out of a pipeline is sent to the overlay managers (and from there to VP1 or WB), or to the WB pipeline directly. An overlay manager manages inputs of multiple pipelines. Transparency settings and user timing configurations for VP1 output are available, based on VESA-DMT and CEA-861 standards.

The DISPC allows the capturing of one of the pipeline outputs, or one of the overlay manager outputs, to redirect it into the WB pipeline. It allows the use of the hardware processing available inside the DISPC, such as color space conversion, rescaling, and compositing, and so forth, to perform memory-to-memory transfer with data processing. It can be used as a HW front end compositing unit in order to pre-compose the output frame and write in memory before being fetched again by DISPC for display on VP1.

NOTE: DISPC does not support any tiled frame buffer, nor any compressed frame buffer, and is not able to generate any compressed output to memory through the write-back pipeline. The DISPC has no capability to support by itself rotation of the frame buffer.

DISPC can be used for composition using memory to memory transfer. In that case the frame buffer on the input and output can be split into smaller blocks of pixels, processed separately by DISPC. Scaling/filtering of the split blocks requires fetching extra pixels around the processed pixel-block. This can be achieved by properly programming the initial accumulator (accu) value of the polyphase filter inside VID/WB pipelines. The number of taps used by the polyphase filter implies the number of extra pixels to fetch (5-tap requires 2 pixels on left/right/bottom/top to be fetched).

8.2.4.1 DISPC Clock Configuration

The VP1_PCLK clock for the VP1 output is provided directly from the DISPC_CLK1 input clock. There is no internal divisor on this clock.

The pixel clock transitions continuously as long as the VP1 is enabled and DISPC_VP1_CONFIG[0] PIXELGATED register bit is 0x0.

The DISPC functional clock FCK_CLK is the internal logic clock, and also acts as the interface clock.

The FCK_CLK has to be greater or equal to the VP1_PCLK, in order for DISPC internal logic to function properly.

The FCK_CLK must be faster or equal to the highest clock required by the user application, in order to support the processing for the VP1 output.

The minimum ratio between the pixel clock frequency and the core functional clock does not depend on the vertical scaling ratio. It is possible to rescale vertically between 16x and 0.25x with a clock ratio between functional clock and pixel clock of 1x. For the horizontal scaling, the minimum clock ratio does not depend on the video window width but on the display width. Considering for example 1080p video frame downscaled by 0.25 on a 1080p screen, the minimum clock ratio is 1x.

The pixel clock and the core functional clocks are asynchronous. There is re-synchronization logic between internal logic and the VP1 output interface.

8.2.4.2 DISPC Software Reset

To perform a software reset on the DISPC, set the DISPC_SYSCONFIG[1] SOFTRESET bit to 0x1. The DISPC_SYSSTATUS[0] DISPC_FUNC_RESETDONE bit indicates that the software reset is complete when its value is 0x1. When the software reset completes, the DISPC_SYSCONFIG[1] SOFTRESET bit is automatically reset. Software must ensure that the software reset completes before performing DISPC operations.

The completion of the software reset for the VP logic is indicated in the DISPC_SYSSTATUS[1] DISPC_VP1_RESETDONE bit.

8.2.4.3 DISPC Power Management

The DISPC supports the MStandby/Wait, IdleReq/SidleAck, Disconnect and Wake-up protocols as defined in Chapter 3, *Power, Reset, and Clock Management*.

8.2.4.3.1 DISPC Idle Mode

The DISPC supports no-idle mode, force-idle mode, and smart-idle mode. The mode can be selected by programming the appropriate value in the DISPC_SYSCONFIG[4:3] SIDLEMODE bit field.

Condition of assertion of the SIdleAck signal:

- In no-idle mode: SIdleAck is never asserted.
- In force-idle mode: SIdleAck is asserted unconditionally with a 1-configuration port interface clock cycle delay with respect to an IdleReq assertion.

NOTE: The proper use of force-idle mode assumes that no interrupt needs to be generated.

- In smart-idle mode: SIdleAck is asserted when at least the following conditions are satisfied:
  - No interrupt is pending.
  - The DISPC no longer uses the interface clock for the slave port.

Once SIdleAck is asserted, the DISPC interface clock used by the slave port can be shut down at any time. Any transactions on the configuration port are ignored.

The conditions of deassertion of the SIdleAck signal are:

- In force-idle mode: SIdleAck is deasserted with a 1-configuration port interface clock cycle delay with respect to an IdleReq deassertion.
- In smart-idle mode: SIdleAck is deasserted with a 1-configuration port interface clock cycle delay with respect to an IdleReq deassertion.

Once SIdleAck is released, the DISPC is fully operational and a DMA request can be processed normally.

8.2.4.3.2 DISPC StandBy Mode

The DISPC supports no-standby mode, force-standby mode, and a single smart-standby mode. The mode is set in the DISPC_SYSCONFIG[13:12] MIDLEMODE bit field. The functional clock is always active, if the module is enabled. The L3_MAIN clock can be shut down at any time independently of the status of MStandby.

The conditions of assertion of the MStandby signal are:

- In no-standby mode: MStandby is never asserted.
- In force-standby mode: MStandby is asserted when the module is disabled.
- In smart-standby: In the case of one of the following conditions:
  - The GFX pipeline is disabled or enabled and the data fetch is complete for the GFX window, or the GFX pipeline is enabled but the data fetch did not complete and data in the DMA buffer is greater than the high threshold value.
  - A VID pipeline (VID1 or VID2) is disabled or enabled and the data fetch is complete for the VID window, or the VID pipeline is enabled but the data fetch did not complete and data in the DMA buffer is greater than the high threshold value.
  - The WB pipeline is disabled or enabled and the data store to memory is complete for the WB picture, or the WB pipeline is enabled but the data storage did not complete and data in the DMA buffer is lower than the low threshold value.

The MStandby signal asserts whenever all the above events have occurred or the DISPC is disabled. While MStandby is asserted, the DISPC does not generate any transaction on the L3_MAIN master port.

The conditions of deassertion of the MStandby signal are:

- In force-standby mode: MStandby is deasserted only when the DISPC is enabled.
- In smart-standby mode: In the case of one of the following conditions:
  - The GFX pipeline is enabled but the data fetch did not complete for the GFX window, and the data in the DMA buffer is less than the low threshold value.
  - A VID pipeline (VID1 or VID2) is enabled but the data fetch did not complete for the VID window, and the data in the DMA buffer is less than the low threshold value.

– The WB pipeline is enabled but the data store did not complete for the WB picture, and the data in the DMA buffer more than the high threshold value.

The above events are conditions for MStandby signal de-assertion. MStandby signal is de-asserted whenever any one the above event happens.

Detection of the deassertion conditions assumes that the interface clocks are active.

8.2.4.3.3 DISPC Wakeup

The DISPC supports wake-up signaling. The mode can be selected by programming the appropriate value in the DISPC_SYSCONFIG[2] ENAWAKEUP bit. Because the SWakeUp signal is asynchronous, it does not require the interface clock.

The conditions of assertion of the SWakeUp signal are:

- The GFX pipeline is enabled but the data fetch did not complete for the GFX window, and the data in the DMA buffer is less than the low threshold value.
- At least one of the the video pipelines (VID1 or VID2) is enabled but the data fetch did not complete for the VID window, and the data in the DMA buffer is less than the low threshold value.
- The WB pipeline is enabled and the data in the DMA buffer is more than the high threshold value.
- The last pixel displayed into the VP1 panel if it is not the last frame

The SWakeUp signal is asserted whenever any one of the abpveevents occurs and IdleAck is asserted, and Idlemode is set to smart-idle.

The condition of deassertion of the SWakeUp signal is:

- Immediately after deassertion of IdleReq

8.2.4.4 DISPC Interrupt Requests

The interrupt line, DISPC_IRQ, indicates when one or more events are detected by the hardware. Each event is independently maskable by setting the DISPC_IRQENABLE_SET register.

To check when a particular interrupt event occurs and to reset a particular event, the DISPC_IRQSTATUS register must be accessed. This register regroups the status of internal events in the module that generate an interrupt (read 0: no interrupt occurred; read 1: interrupt occurred; write 1: status bit reset).

There are two level of interrupts. The first level, controlled by the DISPC_IRQENABLE_SET, DISPC_IRQENABLE_CLR and DISPC_IRQSTATUS registers, is used to indicate common events and also source for the second level of interrupt. The second level of interrupts consists of status and enable interrupt registers for each pipeline and the video port.

Table 8-23. DISPC Interrupts - First Level

Interrupt Name	Description
VP1_IRQ	At least one event of the Video Port 1 interrupt events occurred
GFX_IRQ	At least one event of the GFX pipeline interrupt events occurred
VID1_IRQ	At least one event of the VID1 pipeline interrupt events occurred
VID2_IRQ	At least one event of the VID2 pipeline interrupt events occurred
WB_IRQ	At least one event of the WB pipeline interrupt events occurred
WAKEUP_IRQ	WakeUp: Occurs when the SWakeUp signal is asserted.

The second level of interrupts for GXF pipeline is controlled through the DISPC_GFX1_IRQENABLE and DISPC_GFX1_IRQSTATUS registers.

Table 8-24. DISPC Interrupts - Second Level - GFX Pipeline

Interrupt Name	Description
GFXENDWINDOW_IRQ	End of the graphics window: The DMA engine has fetched all the data from memory for the graphics for the current frame.

Table 8-24. DISPC Interrupts - Second Level - GFX Pipeline (continued)

Interrupt Name	Description
GFXBUFFERUNDERFLOW_IRQ	GFX DMA buffer underflow: The input graphics DMA buffer goes underflow.
GFXREGIONBASEDPIPESTART_IRQ	Graphics window started on overlay. Used for Region-based feature.
GFXREGIONBASEDPIPEEND_IRQ	Graphics window ended on overlay. Used for Region-based feature.

The second level of interrupts for VID pipelines is controlled through the DISPC_VID_IRQENABLE and DISPC_VID_IRQSTATUS registers.

Table 8-25. DISPC Interrupts - Second Level - VID Pipelines

Interrupt Name	Description
VIDENDWINDOW_IRQ	End of the VID window: The DMA engine has fetched all the data from memory for the VID for the current frame.
VIDBUFFERUNDERFLOW_IRQ	VID DMA buffer underflow: The input VID DMA buffer goes underflow. Does not necessary means that the buffer is empty (out of order refill), but simply that the required pixel is not in yet.
VIDREGIONBASEDPIPESTART_IRQ	Video window started on overlay. Used for Region-based feature.
VIDREGIONBASEDPIPEEND_IRQ	Video window ended on overlay. Used for Region-based feature.

The second level of interrupts for WB pipeline is controlled through the DISPC_WB_IRQENABLE and DISPC_WB_IRQSTATUS registers.

Table 8-26. DISPC Interrupts - Second Level - WB Pipeline

Interrupt Name	Description
WBUNCOMPLETEERROR_IRQ	The WB pipe is reset before all data of the frame currently written back are output to the system memory.
WBBUFFEROVERFLOW_IRQ	Write-back DMA buffer Overflow. The output Write-back DMA buffer goes overflow. It can not occur when write-back channel is used in memory to memory transfer mode, but only in capture mode. In capture mode the timings are defined by the timer associated with the output. In memory to memory mode, there is timing constraint.
WBREGIONBASEDEVENT_IRQ	For Write-back region-based feature, event indicating end of current window.
WBSYNC_IRQ	Indicates that Write-back has synced to new configuration (configuration copied from shadow registers to work).
WBFRAMEDONE_IRQ	FrameDone interrupt for WriteBack. Indicates that the WB frame has been completely flushed out of WB-DMA.

The second level of interrupts for VP1 ouput is controlled through the DISPC_VP1_IRQENABLE and DISPC_VP1_IRQSTATUS registers.

Table 8-27. DISPC Interrupts - Second Level - VP1 Output

Interrupt Name	Description
FRAMEDONE_IRQ	Frame Done for VP1 output. After disabling the VP output of the DISPC, the interrupt is set when the active frame related to the VP has completed.
VSYNC_IRQ	VSYNC for VP1 output: VSYNC interrupt for the VP1 has occurred at the end of the frame.
VSYNC_ODD_IRQ	VSYNC for odd field. VSYNC_ODD interrupt has occurred at the end of the frame(EVSYNC received and the field polarity is odd).
PROGRAMMEDLINENUMBER_IRQ	Programmed line number: The VP1 has reached the user programmed line number.
SYNCLOST_IRQ	Synchronization lost on VP1 output: Occurs when VSYNC width/front or back porches are not wide enough to load the pipelines with data (VP1 output).

8.2.4.5 DISPC DMA Requests

The DISPC DMA request signal, DISPC_DREQ, is not a classical one, but rather a synchronization signal between the DISPC and device DMA controller. The device DMA controller is informed that a programmable number of lines are output to the VP1 and that a system memory can be updated. This request is related to the interrupt event PROGRAMMEDLINENUMBER_IRQ described in Table 8-27. This allows the device DMA controller channel to be synchronized with the internal DMA controller of the DISPC.

In other words, it allows synchronizing a memory-to-memory frame buffer update based on the scan line of the frame buffer in system memory (SDRAM or SRAM) by the DISPC. The DISPC_DREQ request is generated at a programmable line number defined in the DISPC_VP1_LINE_NUMBER[11:0] LINENUMBER bit field. This process allows an application to use a single frame buffer and to update it after a certain number of lines are read by the DISPC.

8.2.4.6 DISPC DMA Engine

The DISPC DMA engine:

- Requests and supplies data from system memory to the GFX, VID1 and VID2 pipelines through the interconnect, based on the configuration of the DISPC and GFX/VID pipelines settings.
- Stores composed frames from GFX/VID pipelines or overlay managers to system memory through the WB pipeline and interconnect, based on the configuration of the DISPC and WB pipeline settings.
- Is fully programmable to support configuration with only 1D burst.

Each pipeline has a dedicated buffer and a channel with independent settings. Table 8-28 lists the default size and allocation of the DMA buffer. Each DMA buffer is divided into two spaces, top and bottom. Depending on the application, a DMA buffer space can be associated to a pipeline or merged with other spaces. The total number of spaces for each pipeline is from 0 (pipeline inactive) to the number of pipelines × 2 (in that case, all the DMA buffers are associated to a single pipeline). The sum of the number of spaces allocated for each pipeline must not be greater than the maximum number of available spaces. The correct number of spaces must be allocated to ensure no underflow. The spaces allocated to each pipeline must be greater than or equal to the minimum number of spaces required to support the throughput and system latency. The space assignations are done in the DISPC_GLOBAL_BUFFER register.

Table 8-28. DISPC DMA Buffer Size

Pipelines	DMA Buffer Size
GFX	4 lines × 640 × 128 bits
VID1	4 lines × 640 × 128 bits
VID2	4 lines × 640 × 128 bits
WB	4 lines × 640 × 128 bits

8.2.4.6.1 DISPC DMA Addressing and Bursts

For each line to be fetched/stored, the DMA engine:

- Calculates the pixel address
- Aligns the address
- Defines the burst structure:
  - Type of burst (1D only)
  - Length of the burst

The DMA engine generates scan addresses to read and write data to and from system memory. The base address defines the start address of the first pixel, and then the address is incremented based on the number of pixels per line, offset between two consecutive lines and number of lines. The ROW_INC registers of GFX/VID pipelines allow to access a frame using 1D bursts, but as a two-dimensional block by adding a fixed address offset at the end of a line. The ROW_INC can also be used to skip lines from the input frame. The byte address of each pixel in the frame buffer in the system memory is determined by:

Pixel address = Base address + x × bpp + (y × ((width × bpp) + increment))

where:

- Base address corresponds to the base address (for YUV–NV12 or YUV4:2:0-NV21 format) defined by:
  - DISPC_GFX1_BA_j[31:0] BA bit fields
  - DISPC_VID_BA_j[31:0] BA bit fields
  - DISPC_VID_BA_UV_j[31:0] BA bit fields
- bpp corresponds to the number of bits per pixel defined by the DISPC_GFX1_ATTRIBUTES[6:1] FORMAT bit field or the DISPC_VID_ATTRIBUTES[6:1] FORMAT bit field.
- width corresponds to the number of pixels per line defined by the DISPC_GFX1_SIZE[11:0] SIZEX + 1 bit field or DISPC_VID_SIZE[11:0] SIZEX + 1 bit field.
- increment corresponds to the number of bytes to skip between two contiguous lines defined by the DISPC_GFX1_ROW_INC[31:0] ROWINC – 1 bit field or the DISPC_VID_ROW_INC[31:0] ROWINC – 1 bit field.
- x corresponds to the pixel position on the x-axis.
- y corresponds to the pixel position on the y-axis.

NOTE: For YUV420-NV12 or YUV420-NV21 format, the pixel values are defined in two buffers (Y and UV). The base address of the Y buffers is defined in the DISPC_VID_BA_j[31:0] BA bit field. The base address of the UV buffers is defined in the DISPC_VID_BA_UV_j[31:0] BA bit field.

In case of interlaced mode, DISPC_VID_BA_0 and DISPC_VID_BA_UV_0 registers define the base address of the even field, and DISPC_VID_BA_1 and DISPC_VID_BA_UV_1 registers define the base address of the odd field.

Table 8-29 summarizes the register settings for a simple access of a picture in the system memory.

Table 8-29. DISPC Register Settings for Accessing Image in Internal Memory

Registers	Value
DISPC_GFX1_BA_j/DISPC_VID_BA_j/DISPC_WB_BA_j	PBA, the physical base address of image in the memory
DISPC_VID_BA_UV_j/DISPC_WB_BA_UV_j	PBA, the physical base address of UV buffers image in the memory
DISPC_GFX1_PIXEL_INC/DISPC_VID_PIXEL_INC	1 or other in pixel incremental value
DISPC_GFX1_ROW_INC/DISPC_VID_ROW_INC	1 or other in row incremental value

An interconnect request (128 bits) corresponds to one or several pixels, depending on the bits per pixel. Therefore, the DMA engine determines the appropriate burst sequence to optimize the fetching/storing of each new line. The DMA engine must prevent a single burst from crossing two lines. The DMA engine supports only 1D burst. 1D burst is used, if the fetch/storage is linear in memory. The size of the burst can be one of the following values: 1x128bit, 2x128-bit, 4x128-bit or 8x128-bit. The default burst size at reset time is 8 × 128 bits. Because the burst size must be aligned to the burst boundary, in case of misalignment, the DMA engine may issue one or more smaller burst requests.

8.2.4.6.2 DISPC DMA Buffers

8.2.4.6.2.1 DISPC READ DMA Buffers (GFX and VID Pipelines)

When the vertical front porch (VFP) period starts after the last horizontal front porch (HFP) of the last line, the DMA buffers are flushed according to the selected output associated with the pipeline. The DMA engine restarts fetching data from the memory through the L3_MAIN interconnect. Enabling or disabling the DISPC flushes the DMA buffers (except the WB DMA buffers).

Programmable high and low thresholds, independent for each DMA buffer, are used by the DMA engine to start and stop requesting data to the L3_MAIN interconnect.

- When low threshold (set in the DISPC_GFX1_BUF_THRESHOLD[15:0] BUFLOWTHRESHOLD or DISPC_VID_BUF_THRESHOLD[15:0] BUFLOWTHRESHOLD bit field) is reached, the DMA engine starts a request on the L3_MAIN interconnect to fill the DMA buffer.
- When high threshold (set in the DISPC_GFX1_BUF_THRESHOLD[31:16] BUFHIGHTHRESHOLD or DISPC_VID_BUF_THRESHOLD[31:16] BUFHIGHTHRESHOLD bit field) is reached, the DMA engine stops requesting encoded pixels.

NOTE: The following limitations on DISPC_VID_BUF_THRESHOLD[15:0] BUFLOWTHRESHOLD must be considered:

- If VID scaler is disabled, BUFLOWTHRESHOLD can be programmed as low as interconnect latency and pixel output rate allow it
- If VID scaler is enabled, BUFLOWTHRESHOLD must be programmed to guarantee that at least four full lines can be stored

To avoid underflow at the beginning of a frame and have sufficient encoded pixel data to start some processing, a preloading of the DMA buffer is configurable between a fixed value of bytes or the high threshold value. The preload ensures a minimum number of pixels present in the buffer. When the preload value is reached, the associated channel must start pulling pixels out of the DMA buffer. To enable the preload based on the value entered in the DISPC_GFX1_PRELOAD[11:0] PRELOAD or DISPC_VID_PRELOAD[11:0] PRELOAD bit field, the DISPC_GFX1_ATTRIBUTES[11] BUFPRELOAD bit or the DISPC_VID_ATTRIBUTES[19] BUFPRELOAD bit must be set to 0x0.

NOTE: When self-refresh mode is selected, meaning the data in the DMA buffers are used for multiple frames, and at the end of each frame, the DMA buffers are not flushed.

8.2.4.6.2.2 DISPC WRITE DMA Buffer (WB Pipeline)

Two modes are supported by the WB channel, selectable through the DISPC_WB_ATTRIBUTES[19] WRITEBACKMODE bit:

- Capture mode, WRITEBACKMODE bit set to 0: The OVR1 output, connected to external interface, is captured and at the same time the data are sent on the VP1 output. The WB timings are controlled by the VP1 timings. The output of a VID/GFX pipeline is not sent to the write back, when the WB pipeline is connected to an overlay used in capture mode.
- Memory-to-memory mode, WRITEBACKMODE bit set to 1: One of the overlay outputs or one of the pipeline outputs is captured to perform a memory-to-memory transfer, with some processing by the DISPC (rescaling, overlaying, color space conversion, etc.).

In capture mode: The DMA engine starts storing data to memory through the L3_MAIN-based interconnect as soon as enough data is available for the programmed burst size. When enabling/disabling the DISPC, the DMA buffer is flushed. The programmable thresholds low and high are used by the DMA engine to start and stop sending data to the L3_MAIN interconnect.

NOTE: In WB capture mode, if a new frame starts before the WB DMA buffers contents are fully written onto external memory, then the contents of the WB DMA buffers are lost (implying last few pixels/lines are corrupted in the captured frame in memory). The DISPC_WB_IRQSTATUS[1] WBUNCOMPLETEERROR_IRQ interrupt bit indicates this situation and triggers every frame. The WBUNCOMPLETEERROR interrupt can be enabled through the DISPC_WB_IRQSTATUS[1] WBUNCOMPLETEERROR_EN register bit.

In memory-to-memory mode: The WB pipeline is not synchronized to any internal or external timing generator. The WB pipeline stores the output of one of the overlay outputs or one of the pipelines. When enabling or disabling the DISPC, the DMA buffers are flushed. The programmable thresholds low and high are used by the DMA engine to start and stop sending data to the L3_MAIN interconnect.

Programmable high and low thresholds are used by the DMA engine to start and stop sending data to the L3_MAIN interconnect.

- When high threshold (set in the DISPC_WB_BUF_THRESHOLD[31:16] BUFHIGHTHRESHOLD bit

field) is reached, the DMA engine starts sending data on the L3_MAIN interconnect to empty buffer.

- When low threshold (set in the DISPC_WB_BUF_THRESHOLD[15:0] BUFLOWTHRESHOLD bit field) is reached, the DMA engine stops sending encoded pixels.

At the end of the frame, to completely drain the DMA buffer, some smaller bursts (even single requests) must be issued. To limit the number of interconnect requests from the DISPC, a number of IDLE cycles between requests can be inserted. IDLE cycles can be inserted only when WB is used in memory-to-memory mode. It is ignored when WB is in capture mode.

The number of IDLE cycles between requests can be activated and determined by:

- Setting the DISPC_WB_ATTRIBUTES[27] IDLESIZE bit to 0x0 (default value) and entering the number of idles between requests in the DISPC_WB_ATTRIBUTES[31:28] IDLENUMBER bit field. Idle numbers vary from 0 to 15.
- Setting the DISPC_WB_ATTRIBUTES[27] IDLESIZE bit to 0x1, which considers the size of the burst to determine the number of IDLE cycles. In this case, the number of IDLE cycles equals IDLENUMBER × 8 (0 to 120).

8.2.4.6.3 DISPC DMA MFLAG Mechanism and Arbitration

The MFLAG mechanism allows a dynamic increase of the priority of DISPC real-time traffic, when required, based on the fullness of the DISPC DMA read and write buffers.

The mechanism is implemented for all DMA buffers (GFX, VID1, VID2, and WB).

Programmable buffer thresholds (hysteresis) for each pipeline are used to indicate when the local MFLAG signal for each pipeline is generated. All local MFLAG signals are OR-ed to generate a single DSS MFLAG out band signal, which is provided to the L3 interconnect for granting OCP requests. The out band DSS MFLAG signal is asynchronous to any ongoing OCP transaction.

The threshold for each pipeline corresponds to the fullness of the associated DMA buffer, and is defined by two threshold parameters:

- HT_MFLAG: High threshold.
  - For read access from the GFX, VID1 and VID2 pipelines, when the corresponding pipeline buffer reaches the programmed value, the associated local MFLAG signal goes low (deasserted).
  - For write access from the WB pipeline, when the corresponding pipeline buffer reaches the programmed value, the associated local MFLAG signal goes high (asserted).
  - This threshold can be programmed in the following bit fields:
    - For the GFX pipeline in the DISPC_GFX1_MFLAG_THRESHOLD[31:16] HT_MFLAG bit field
    - For the VID pipelines in the DISPC_VID_MFLAG_THRESHOLD[31:16] HT_MFLAG bit fields
    - For the WB pipeline in the DISPC_WB_MFLAG_THRESHOLD[31:16] HT_MFLAG bit field
- LT_MFLAG: Low threshold.
  - For read access from the GFX, VID1 and VID2 pipelines, when the corresponding pipeline buffer reaches the programmed value, the associated local MFLAG signal goes high (asserted).
  - For write access from the WB pipeline, when the corresponding pipeline buffer reaches the programmed value, the associated local MFLAG signal goes low (deasserted).
  - This threshold can be programmed in the following bit fields:
    - For the GFX pipeline in the DISPC_GFX1_MFLAG_THRESHOLD[15:0] LT_MFLAG bit field
    - For the VID pipelines in the DISPC_VID_MFLAG_THRESHOLD[15:0] LT_MFLAG bit fields
    - For the WB pipeline in the DISPC_WB_MFLAG_THRESHOLD [15:0] LT_MFLAG bit field

By default, the MFLAG mechanism is disabled (DISPC_GLOBAL_MFLAG_ATTRIBUTE[1:0] MFLAG_CTRL bit field = 0x0), and the DSS MFLAG out band signal is low (de-asserted). The arbitration scheme for the DISPC pipelines is the same as described in Section 8.2.4.6.5, *DISPC Arbitration*. That is, round-robin either between high-priority pipelines, or between normal-priority pipelines (if all pipelines are of normal priority).

When the DISPC_GLOBAL_MFLAG_ATTRIBUTE[1:0] MFLAG_CTRL bit field is set to 0x2, the MFLAG mechanism is enabled, and the DSS MFLAG out band signal is dynamically set to 0 or 1, depending on DMA buffers fullness and programmed threshold levels, as explained previously in this section. In this case, the arbitration scheme for DISPC pipelines is round-robin between those high-priority pipelines, which have asserted local MFLAG signals. If there are no high-priority pipelines with asserted local MFLAG signals, then the arbitration scheme is the same as described in Section 8.2.4.6.5, *DISPC Arbitration*.

The DISPC_GLOBAL_MFLAG_ATTRIBUTE[2] MFLAG_START bit defines additional rules for the MFLAG mechanism:

- If the MFLAG_START bit is set to 0x0 (default value), then in the beginning of the frame when the DMA buffer is empty, the local MFLAG signal of each pipeline is kept at 0 until PRELOAD is reached (for more information on preloading, see Section 8.2.4.6.2.1, *DISPC READ DMA Buffers (GFX and VID Pipelines)*. Then, based on the setting of the DISPC_GLOBAL_MFLAG_ATTRIBUTE[1:0] MFLAG_CTRL bit field, the MFLAG signal is generated and internal logic is arbitrating between pipeline requests.
- If the MFLAG_START bit is set to 0x1, then even in the beginning of the frame when the DMA buffer is empty, the DISPC_GLOBAL_MFLAG_ATTRIBUTE[1:0] MFLAG_CTRL bit field defines the generation of the MFLAG signal for each pipeline.

8.2.4.6.4 DISPC DMA Predecimation

The predecimation process consists of downscaling an image by fetching only the necessary pixels in the memory. Vertical and horizontal predecimation are possible:

- Vertical predecimation: The picture stored in memory can be predecimated vertically by skipping lines. Burst mode is used to fetch the data when skipping lines. Only the lines that will be used by the DISPC are fetched from memory; the other lines are skipped. The DMA engine sends requests only for the useful lines using 1D burst. The base address indicates the first valid pixel to fetch from memory. The number of lines to skip is set in the DISPC_GFX1_ROW_INC[31:0] ROWINC or DISPC_VID_ROW_INC[31:0] ROWINC bit field.
- Horizontal predecimation: When fetching data from memory, it is possible to skip 1 of 2 pixel data containers, up to 1 of 2047 pixel data containers, by setting the DISPC_GFX1_PIXEL_INC[7:0] PIXELINC or DISPC_VID_PIXEL_INC[7:0] PIXELINC bit field to the number of pixel data containers to skip (n), multiplied by the size of a pixel data container (in bytes), + 1. See the following note for more details.

NOTE: The restriction to horizontal predecimation is that there is at least one useful pixel per 128-bit request. In that case, the DMA engine uses burst mode instead of singles to optimize the requests in terms of latency and SDRAM efficiency.

No decimation is supported when the input format is 1, 2, 4, or 8-bit BITMAP.

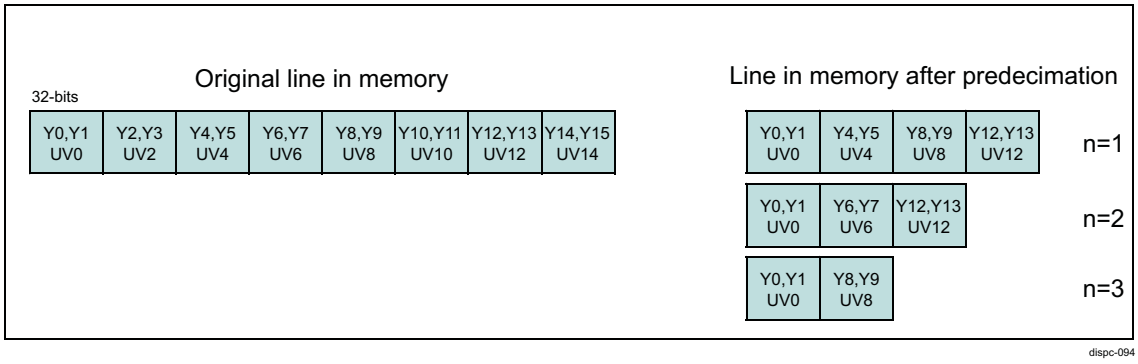
For RGB and YUV4:2:0 data formats, each pixel data container in memory holds 1 pixel. Thus, when configuring the PIXELINC bit field, the value of n equals the number of pixels to skip:

- For RGB format, one pixel data container = 32 bits = 1 pixel
- For YUV format:
  - One Y pixel data container = 8 bits = 1 pixel
  - One UV pixel data container = 16 bits = 1 pixel

For YUV4:2:2 format, each 32-bit pixel data container holds the Luma components for 2 pixels, and the Chrominance component of 1 pixel (see Figure 8-25). Therefore, for the valid values of the PIXELINC bit field in the case of the following YUV4:2:2 format, caution must be taken because n equals the number of pixel data containers to skip, not the number of pixels:

- For n = 1, PIXELINC = 5
- For n = 2, PIXELINC = 9
- For n = 3, PIXELINC = 13
- For n = 4, PIXELINC = 17, etc.

Figure 8-25. DISPC YUV4:2:2 Predecimation



8.2.4.6.5 DISPC DMA Arbitration

The requests (reads or writes) sent to the L3_MAIN interconnect are pipelined and arbitrated in a round-robin scheme. The default arbitration scheme must be modified by setting the priority attribute of each pipeline as defined in the following bits:

- DISPC_GFX1_ATTRIBUTES[14] ARBITRATION
- DISPC_VID_ATTRIBUTES[14] ARBITRATION
- DISPC_WB_ATTRIBUTES[14] ARBITRATION

By default, all pipelines have the same priority (normal), which means all pipeline requests are treated in a round-robin order manner. If one or more pipelines require a higher number of requests going to the L3_MAIN interconnect, its priority can be moved up to high priority. In this case, the high-priority pipeline is granted access before any pipeline in normal priority. If more than one active pipeline is in high priority, the behavior is the same as all active pipelines in normal priority. Normal active pipelines are not treated until all high active pipelines are finished. The ARBITRATION bit cannot be modified during the entire frame.

This functionality balances the bandwidth of the pipeline depending on its constraint. It can be used to give higher priority to the pipelines with real-time constraints versus non-real-time pipelines. For example, pipelines associated with the WB (used in memory-to-memory mode) must have lower priority than the pipelines associated with VP1 output.

8.2.4.6.6 DISPC DMA Power Modes

8.2.4.6.6.1 DISPC DMA Low-Power Mode

Each DMA buffer is divided into two spaces. Each space can be associated with the pipeline or merged with other DMA buffers. The total number of DMA buffers for each pipeline is from 0 (pipeline inactive) to the number of pipelines × 2 (in that case all the DMA buffers are associated with a single pipeline). The sum of the number of DMA buffers allocated for each pipeline must not be greater than the maximum available. The correct number of DMA buffers must be allocated to ensure no underflow. The number of DMA buffers allocated to each pipeline must be greater than or equal to the minimum required to support the throughput and the system latency.

NOTE: When the number of buffers is changed, the thresholds must be reprogrammed to reflect the new configuration of the DMA buffer.

8.2.4.6.6.2 DISPC DMA Ultralow-Power Mode

In low-power mode, the L3_MAIN interconnect is used to fill up the DMA buffers to store all the data required to display a full frame. The L3_MAIN interconnect is not used to fetch new pixels for the following frames. The data in the DMA buffer are reused to display on the screen.

The setting of the mode is independent for each pipeline. One pipeline may have all the frame pixels in its DMA buffer and the other pipelines may have to refill their respective DMA buffers along the display scan because the frame buffer is too big to be stored in the DMA buffer.

The DMA buffers can be merged to optimize the L3_MAIN interconnect off time. Merging the DMA buffers into a single buffer can be used at the same time to improve ultralow-power mode (see Section 8.2.4.6.6.1, *Low-Power Mode*).

During the time in which the frames are fetched in the internal DMA buffer, MStandby must be asserted if the DISPC_SYSCONFIG[13:12] MIDLEMODE bit field is set to 0x2 (smart-standby mode).

Two ultralow-power modes can be entered manually or automatically:

- Self-refresh mode: Starting self-refresh mode is done manually by setting the DISPC_GFX1_ATTRIBUTES[15] SELFREFRESH or DISPC_VID_ATTRIBUTES[15] SELFREFRESH bit to 0x1 after capturing a frame in the DMA buffers. Self-refresh mode is stopped by setting the SELFREFRESH bit to 0x0.
- Automatic self-refresh mode: By setting the DISPC_GFX1_ATTRIBUTES[17] SELFREFRESHAUTO or DISPC_VID_ATTRIBUTES[17] SELFREFRESHAUTO bit to 0x1, the transition from off to on self-refresh mode is done by hardware after capturing the first frame. The hardware reflects the status of the self-refresh mode by setting the SELFREFRESH bit to 0x1, which means that the data are read inside the DMA buffer without accessing the interconnect and system memory during the frame. Setting the SELFREFRESH bit to 0x0 updates the DMA buffer.

NOTE: The WB pipeline does not support ultralow-power mode.

8.2.4.7 DISPC Memory Formats

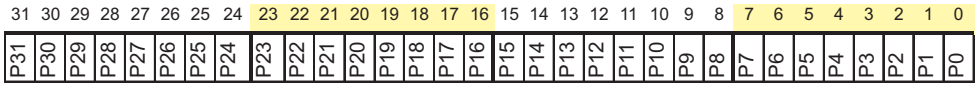
The graphic, video and write-back pipelines support various types of memory formats. Table 8-30 lists all supported formats for each pipeline.

For the BITMAP formats in GFX pipeline the nibble mode can be enabled by setting the DISPC_GFX1_ATTRIBUTES[7] NIBBLEMODE bit to 0x1. For VID pipelines the nibble mode can be selected in DISPC_VID_ATTRIBUTES[10] NIBBLEMODE register bit.

Table 8-30. DISPC Memory Formats Supported

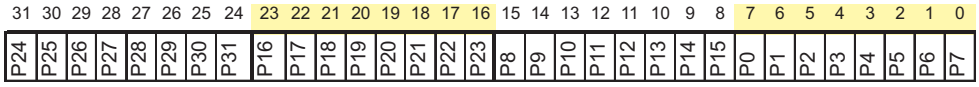
Formats	GFX	VID1	VID2	WB
BITMAP 1-bpp	x	x	x	
BITMAP 2-bpp	x	x	x	
BITMAP 4-bpp	x	x	x	
BITMAP 8-bpp	x	x	x	
xRGB12-4444	x	x	x	x
RGBx12-4444	x	x	x	x
ARGB16-4444	x	x	x	x
RGBA16-4444	x	x	x	x
RGB16-565	x	x	x	x
xRGB16-1555	x	x	x	x
ARGB16-1555	x	x	x	x
xRGB24-8888	x	x	x	x
RGBx24-8888	x	x	x	x
RGB24-888	x	x	x	x
ARGB32-8888	x	x	x	x
RGBA32-8888	x	x	x	x
BGRA32-8888	x	x	x	x
UYVY 4:2:2		x	x	x
YUV2 4:2:2		x	x	x
YUV4:2:0 – NV12		x	x	x
YUV4:2:0 – NV21		x	x	x

- BITMAP 1-bpp data memory organization (CLUT)



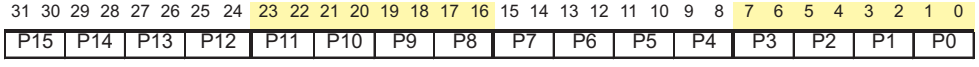
dss-T039

- BITMAP 1-bpp data memory organization (CLUT) in nibble mode



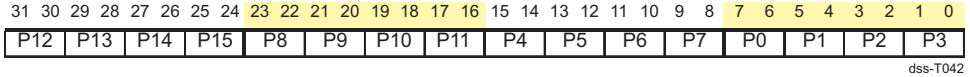
dss-T040

- BITMAP 2-bpp data memory organization (CLUT)



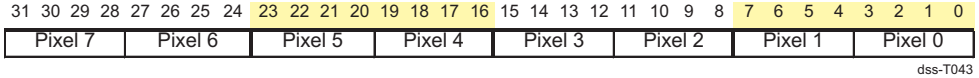
dss-T041

- BITMAP 2-bpp data memory organization (CLUT) in nibble mode



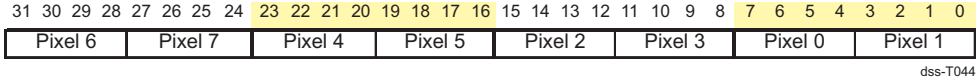
dss-T042

- BITMAP 4-bpp data memory organization (CLUT)



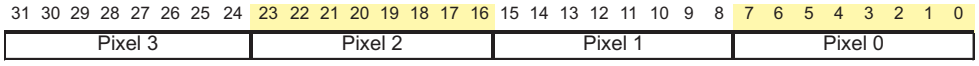
dss-T043

- BITMAP 4-bpp data memory organization (CLUT) in nibble mode



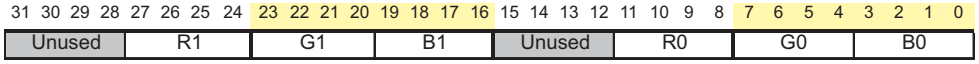
dss-T044

- BITMAP 8-bpp data memory organization (CLUT)



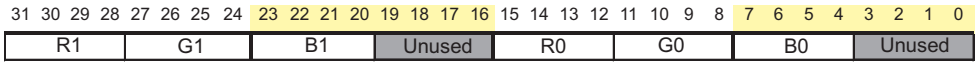
dss-T045

- xRGB12-4444 bpp data memory organization



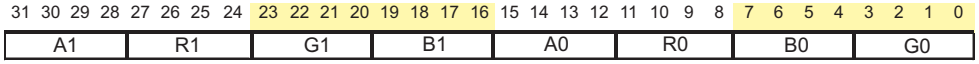
dss-T046

- RGBx12-4444 bpp data memory organization



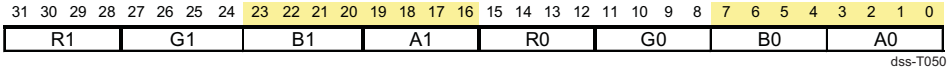
dss-T047

- ARGB16-4444 bpp data memory organization



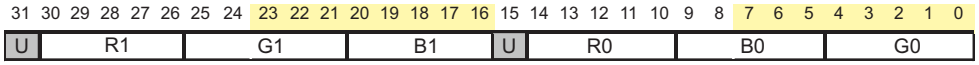
dss-T048

- RGBA16-4444 bpp data memory organization



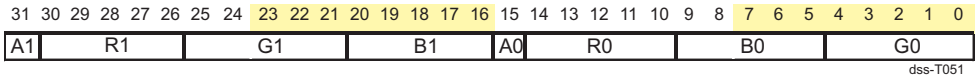
dss-T050

- xRGB16-1555 bpp data memory organization



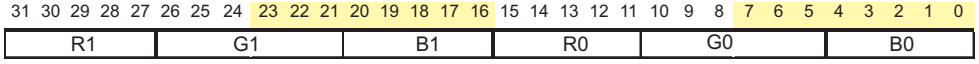
dss-T062

- ARGB16-1555 bpp data memory organization



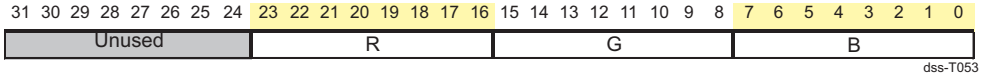
dss-T051

- RGB16-565 bpp data memory organization



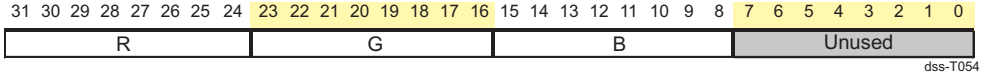
dss-T052

- xRGB24-8888 bpp data memory organization



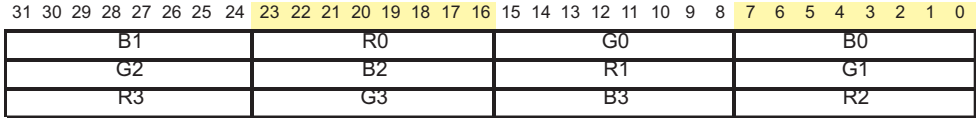
dss-T053

- RGBx24-8888 bpp data memory organization



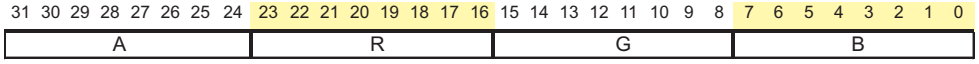
dss-T054

- RGB24-888 bpp packed data memory organization



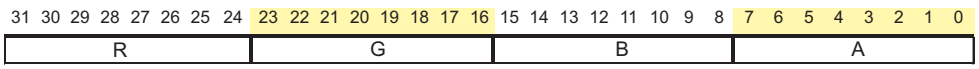
dss-T055

- ARGB32-8888 bpp data memory organization



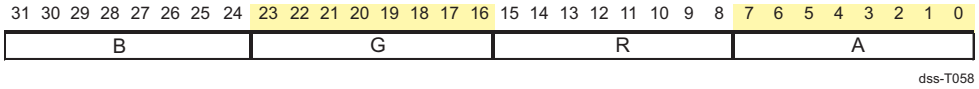
dss-T056

- RGBA32-8888 bpp data memory organization

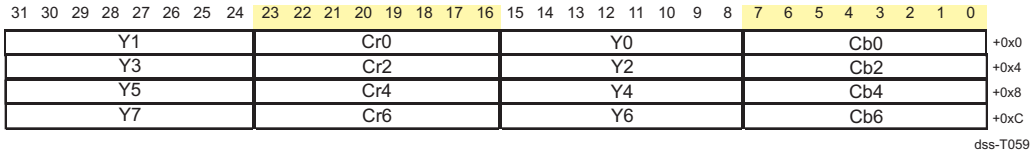


dss-T057

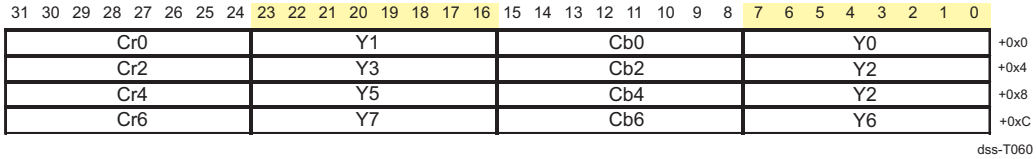
- BGRA32-8888 bpp data memory organization



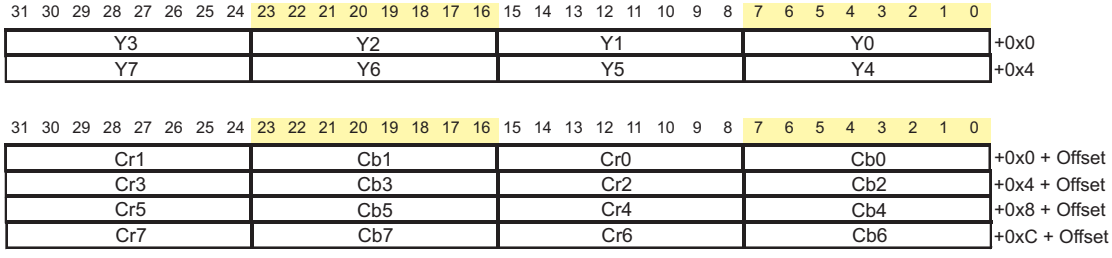
- UYVY4:2:2 data memory organization



- YUV2 4:2:2 data memory organization



- YUV4:2:0-NV12 data memory organization (same for YUV4:2:0-NV21, with only UV in reverse order)



8.2.4.8 DISPC Graphics Pipeline

The input port of the graphics pipeline (GFX) is connected to the graphics DMA buffer controller. The output port of the GFX pipeline is connected to the overlay managers or WB pipeline. The pixel output is directed to the OVR1, OVR2 or WB path by setting the DISPC_GFX1_ATTRIBUTES[8] CHANNELOUT bit. Table 8-41 lists the bit field settings to orient a pipeline to OVR1, OVR2 or WB output. The default value directs the GFX pipeline to OVR1. The GFX pipeline can be enabled by setting the DISPC_GFX1_ATTRIBUTES[0] ENABLE bit to 0x1.

NOTE: It is not possible to change the direction of the GFX pipeline on the fly. If the graphics pipeline must be connected to an overlay manager different from the one to which it is currently connected, then the following steps must be performed:

1. Disable the GFX pipeline by setting the DISPC_GFX1_ATTRIBUTES[0] ENABLE bit to 0x0.
2. Direct the GFX pipeline to the new overlay manager by modifying the DISPC_GFX1_ATTRIBUTES[8] CHANNELOUT bit.
3. Disable the DISPC output which corresponds with the overlay manager to which the GFX pipeline will be connected next. This is done by setting the in the following bits to 0x0 for the listed outputs:
   - The DISPC_VP1_CONTROL1[0] VPENABLE bit for VP1 output
4. Enable the GFX pipeline by setting the DISPC_GFX1_ATTRIBUTES[0] ENABLE bit to 0x1.
5. Enable the DISPC output that corresponds with the overlay manager to which the GFX pipeline will be connected. This is done by setting the following bits to 0x1 for the listed outputs:
   - The DISPC_VP1_CONTROL1[0] VPENABLE bit for VP1 output

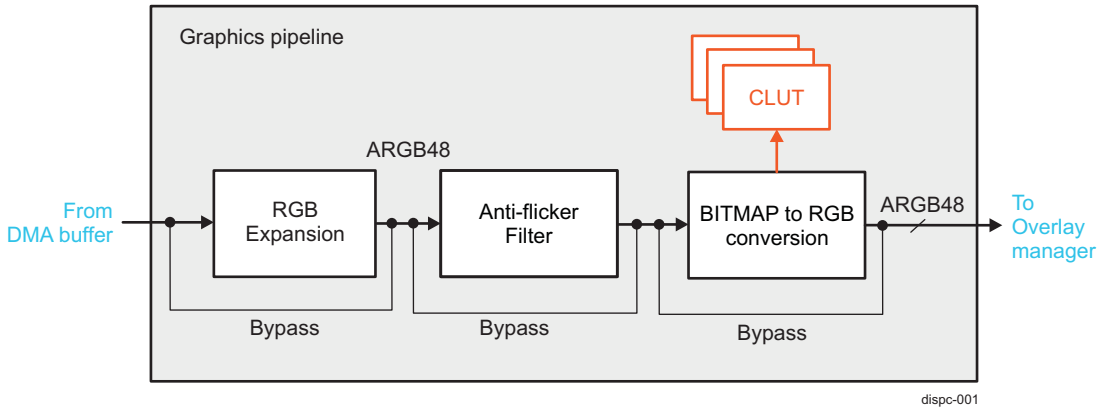
The pipeline contains replication logic, an anti-aliasing filter, and one 256-entry Color Look-up Table (CLUT). The anti-aliasing filter processes the graphics data in RGB format to remove some of the vertical aliasing. The replication logic is used to convert the RGB, ARGB, RGBA pixel formats into ARGB48 format. The 256-entry CLUT is used to convert BITMAP (1, 2, 4, or 8-bit indexed formats) into RGB format.

NOTE: The GFX pipeline does not include a scaler.

Table 8-30 lists the input formats supported by the graphics pipeline.

Figure 8-26 shows the graphics pipeline.

Figure 8-26. DISPC Graphics Pipeline



8.2.4.8.1 DISPC GFX Replication Logic

The replication logic (RGB expansion) increases the color depth of the graphics-encoded pixels (from true color RGB 12 and 16, to 48 bpp).

- The replication logic is always enabled, and the MSBs are copied to the missing LSBs. Table 8-31 describes the remapping of the RGB pixels into ARGB 48-bit values.

Table 8-31. DISPC GFX Replication: RGB Pixel Formats Remapping Into ARGB48-12.12.12.12

	A[11:0]	R[11:0]	G[11:0]	B[11:0]
Format	**MSB LSB**	**MSB LSB**	**MSB LSB**	**MSB LSB**
xRGB12-4444	111111111111	R[3:0]R[3:0]R[3:0]	G[3:0]G[3:0]G[3:0]	B[3:0]B[3:0]B[3:0]
RGBx12-4444	111111111111	R[3:0]R[3:0]R[3:0]	G[3:0]G[3:0]G[3:0]	B[3:0]B[3:0]B[3:0]
RGB16-565	111111111111	R[4:0]R[4:0]R[4:3]	G[5:0]G[5:0]	B[4:0]B[4:0]B[4:3]
xRGB16-1555	111111111111	R[4:0]R[4:0]R[4:3]	G[4:0]G[4:0]G[4:3]	B[4:0]B[4:0]B[4:3]
ARGB16-1555	AAAAAAAAAAAA	R[4:0]R[4:0]R[4:3]	G[4:0]G[4:0]G[4:3]	B[4:0]B[4:0]B[4:3]
ARGB16-4444	A[3:0]A[3:0]A[3:0]	R[3:0]R[3:0]R[3:0]	G[3:0]G[3:0]G[3:0]	B[3:0]B[3:0]B[3:0]
RGBA16-4444	A[3:0]A[3:0]A[3:0]	R[3:0]R[3:0]R[3:0]	G[3:0]G[3:0]G[3:0]	B[3:0]B[3:0]B[3:0]

8.2.4.8.2 DISPC GFX Anti-Aliasing Filter

The anti-aliasing filter processes the graphics data to remove some of the vertical aliasing. It is based on a 3-tap FIR filter with fixed coefficients. For each pixel to be output from the graphics pipeline, the pixel above and below the current line must be read from the DMA graphics FIFO. Therefore, three lines of pixels must be stored in the DMA graphics FIFO.

The anti-aliasinging equations for A, R, G, and B components are:

$A_{out}(x,y) = 0.25 \times A_{in}(x,y-1) + 0.5 \times A_{in}(x,y) + 0.25 \times A_{in}(x,y+1)$

$R_{out}(x,y) = 0.25 \times R_{in}(x,y-1) + 0.5 \times R_{in}(x,y) + 0.25 \times R_{in}(x,y+1)$

$G_{out}(x,y) = 0.25 \times G_{in}(x,y-1) + 0.5 \times G_{in}(x,y) + 0.25 \times G_{in}(x,y+1)$

$B_{out}(x,y) = 0.25 \times B_{in}(x,y-1) + 0.5 \times B_{in}(x,y) + 0.25 \times B_{in}(x,y+1)$

For the first line of processing, because there is no pixel above, the value (x,y) is duplicated.

$Out(x,y) = 0.25 \times In(x,y) + 0.5 \times In(x,y) + 0.25 \times In(x,y+1)$

For the last line of processing, because there is no pixel below, the value (x,y) is duplicated.

$Out(x,y) = 0.25 \times In(x,y-1) + 0.5 \times In(x,y) + 0.25 \times In(x,y)$

NOTE: Anti-aliasing filtering is supported only in RGB formats and not in BITMAP formats.

Anti-aliasinging is not supported for pictures with fewer than two lines. In this case, the user must disable the anti-aliasinging processing.

By default, the anti-aliasing filtering is disabled. It can be enabled by setting the DISPC_GFX1_ATTRIBUTES[24] anti-aliasing bit to 0x1.

8.2.4.8.3 DISPC GFX Color Look-Up Table (CLUT)

The graphics pipeline supports conversion of BITMAP formats (1, 2, 4, or 8-bit indexed) into RGB24 format through a Color Look-Up Table (CLUT).

The look-up table consists of 3 separate 256 x 8-bit memories and is indexed by the source BITMAP data. The table is loaded through direct register access by writing to DISPC_GFX1_CLUT register.

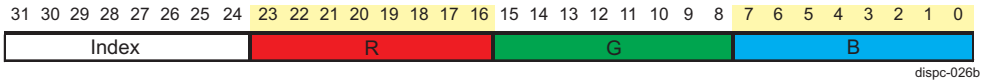
The sequence to load the table is:

1. SW writes (only writes are supported) 32-bit values using single access, or burst access in linear increment burst mode, into DISPC_GFX1_CLUT register. The LSB 24 bits [23:0] are used for the value, and the MSB 8 bits [31:24] are used for the index into the table (see Figure 8-27).
2. Loop to Step 1, if there is a new access to the CLUT register. The SW can access other registers between two accesses to the CLUT register.

SW needs to ensure that there is no visible effect when modifying the table, since it is not under HW control.

The usage of the CLUT is activated when a BITMAP format is selected in the DISPC_GFX1_ATTRIBUTES[6:1] FORMAT register bit-field.

Figure 8-27. DISPC GFX CLUT Data Memory Organization



8.2.4.9 DISPC Video Pipelines

Two identical video pipelines are available, VID1 and VID2. Each video pipeline is connected to the video DMA buffer controller for the input port, and to the overlay managers or WB pipeline. The pixel output is directed to the OVR1, OVR2 or WB path by configuring the DISPC_VID_ATTRIBUTES[16] CHANNELOUT register bit. Table 8-41 summarizes the bit field settings to orient a pipeline to the OVR1, OVR2 or VP1 output. The default value directs all video pipelines to OVR1.

NOTE: It is not possible to change the direction of the video pipelines on the fly. If a video pipeline needs to be connected to a different overlay manager than what it is currently connected, then the following steps need to be performed:

1. Disable the VIDp pipeline by setting the DISPC_VID_ATTRIBUTES[0] ENABLE bit to 0x0.
2. Direct the VIDp pipeline to the new overlay manager by modifying the [16] CHANNELOUT bit in the DISPC_VID_ATTRIBUTES register.
3. Disable the DISPC output that corresponds with the overlay manager to which the VIDp pipeline will be connected next. This is done by setting the following bits to 0x0 for the listed outputs:
   - The DISPC_VP1_CONTROL1[0] VPENABLE bit for VP1 output
4. Enable the VIDp pipeline by writing 0x1 to the DISPC_VID_ATTRIBUTES [0] ENABLE bit.
5. Enable the DISPC output that corresponds with the overlay manager to which the VIDp pipeline will be connected next. This is done by setting the following bits to 0x1 for the listed outputs:
   - The DISPC_VP1_CONTROL1[0] VPENABLE bit for VP1 output

A video pipeline consists of a scaler unit, color space conversion (CSC) unit, VC-1 range mapping unit, one 256-entry Color Look-up Table (CLUT), and some programmable replication logic. The order of the video pipeline unit is configured like follows (see Figure 8-28):

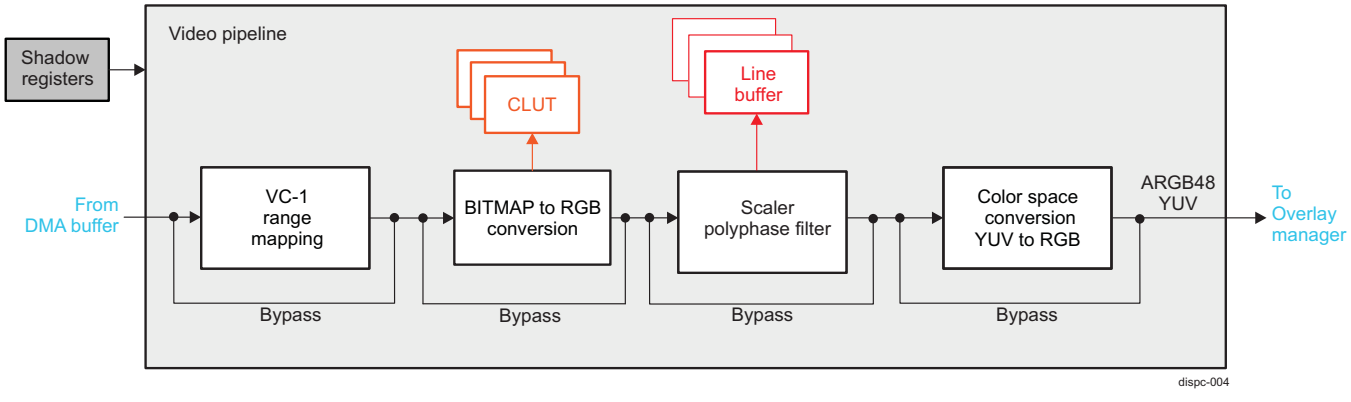
- VC-1 range mapping unit, followed by a color look-up table, then a scaler unit and then a CSC unit. The configuration is used to support RGB, ARGB, and RGBA formats, YUV4:2:2 and YUV420-NV12 (NV21) formats, taking advantage of the scaler to resample the chrominance using five taps horizontally and three or five taps vertically. Each block can be independently bypassed. The 256-entry CLUT is used to convert BITMAP (1, 2, 4, or 8-bit indexed formats) into RGB format.

NOTE: For a BITMAP format data, scaling is not supported. Scaling and color look-up table features are mutually exclusive. If color look-up feature is enabled, then video pipeline scaler has to be disabled.

Table 8-30 lists the input formats supported by the video pipelines.

The video pipeline is enabled by setting the DISPC_VID_ATTRIBUTES[0] ENABLE bit to 0x1.

Figure 8-28. DISPC Video Pipeline Configuration



8.2.4.9.1 DISPC VID Replication Logic

The replication logic (RGB expansion) is used to convert the ARGB pixels into ARGB48 format.

Table 8-32. DISPC VID Replication: ARGB Pixel Formats Remapping Into ARGB48-12121212

Formats	A[11:0] MSB LSB	R[11:0] MSB LSB	G[11:0] MSB LSB	B[11:0] MSB LSB
xRGB12-4444	111111111111	R[3:0]R[3:0]R[3:0]	G[3:0]G[3:0]G[3:0]	B[3:0]B[3:0]B[3:0]
RGBx12-4444	111111111111	R[3:0]R[3:0]R[3:0]	G[3:0]G[3:0]G[3:0]	B[3:0]B[3:0]B[3:0]
RGB16-565	111111111111	R[4:0]R[4:0]R[4:3]	G[5:0]G[5:0]	B[4:0]B[4:0]B[4:3]
xRGB16-1555	111111111111	R[4:0]R[4:0]R[4:3]	G[4:0]G[4:0]G[4:3]	B[4:0]B[4:0]B[4:3]
ARGB16-1555	AAAAAAAAAA	R[4:0]R[4:0]R[4:3]	G[4:0]G[4:0]G[4:3]	B[4:0]B[4:0]B[4:3]
ARGB16-4444	A[3:0]A[3:0]A[3:0]	R[3:0]R[3:0]R[3:0]	G[3:0]G[3:0]G[3:0]	B[3:0]B[3:0]B[3:0]
RGBA16-4444	A[3:0]A[3:0]A[3:0]	R[3:0]R[3:0]R[3:0]	G[3:0]G[3:0]G[3:0]	B[3:0]B[3:0]B[3:0]

8.2.4.9.2 DISPC VID VC-1 Range Mapping Unit

The VC-1 range mapping unit is used when the video frame picture is decoded using a VC-1 codec by the video accelerator. It remaps the Y, Cb, and Cr components. The unit is used primarily for YUV4:2:0-NV12 (NV21) pixel format, but also can be applied to YUV4:2:2 pixel formats (YUV2 and UYVY).

The VC-1 range mapping unit is enabled by setting the DISPC_VID_ATTRIBUTES2[0] VC1ENABLE bit to 0x1. The DISPC_VID_ATTRIBUTES2[3:1] VC1_RANGE_Y and DISPC_VID_ATTRIBUTES2[6:4] VC1_RANGE_CBCR bit fields are two 3-bit values programmed by the user and are independent for each video pipeline. The module is governed by the equations:

Y_{out} = CLIP((((Y_{int} – 128) × (VC1_RANGE_Y + 9) + 4) / 8) + 128)

C_b = CLIP((((C_b – 128) × (VC1_RANGE_CBCR + 9) + 4) / 8) + 128)

C_r = CLIP((((C_r – 128) × (VC1_RANGE_CBCR + 9) + 4) / 8) + 128)

NOTE: The input and output pixel values are unsigned (Y, Cr, and Cb).

The function CLIP () clips to 0 or 255 when minimum or maximum, respectively, are reached; otherwise, the resulting output remains identical.

8.2.4.9.3 DISPC VID Color Look-Up Table (CLUT)

The video pipelines support conversion of BITMAP formats (1, 2, 4, or 8-bit indexed) into RGB24 format through a Color Look-Up Table (CLUT).

The look-up table consists of 3 separate 256 x 8-bit memories and is indexed by the source BITMAP data. The table is loaded through direct register access by writing to DISPC_VID_CLUT register.

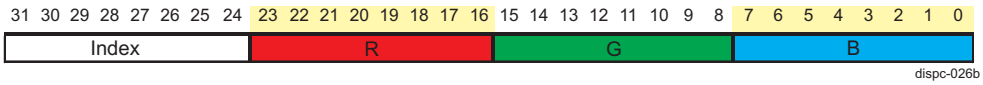
The sequence to load the table is:

1. SW writes (only writes are supported) 32-bit values using single access, or burst access in linear increment burst mode, into DISPC_VID_CLUT register. The LSB 24 bits [23:0] are used for the value, and the MSB 8 bits [31:24] are used for the index into the table (see Figure 8-29).
2. Loop to Step 1, if there is a new access to the CLUT register. The SW can access other registers between two accesses to the CLUT register.

SW needs to ensure that there is no visible effect when modifying the table, since it is not under HW control.

The usage of the CLUT is activated when a BITMAP format is selected in the DISPC_VID_ATTRIBUTES[6:1] FORMAT register bit-field.

Figure 8-29. DISPC VID CLUT Data Memory Organization



8.2.4.9.4 DISPC VID CSC Unit YUV to RGB

The CSC unit converts the video-encoded pixel values from YUV4:4:4 format into RGB36 format (12-bit value per color component: red, green, and blue).

In case of YUV4:2:0 or YUV4:2:2 formats, a chrominance resampling to YUV4:4:4 is required before converting the YUV into RGB values (see Section 8.2.4.9.4.1, *Chrominance Resampling*). YUV4:2:2 or YUV4:2:0 to YUV4:4:4 chrominance resampling is a preprocessing to the color space conversion.

Figure 8-30 through Figure 8-31 show the 3 × 3 11-bit coefficients used to convert from YUV4:4:4 into RGB36. The coefficients are set according to the standard used to encode the pixel data in YUV color space. Table 8-33 summarizes the coefficients with their respective register bit fields.

Table 8-33. DISPC VID Color Space Conversion YUV to RGB Register Bitfield Settings

Coefficients	Bit Field Registers
R_Y	DISPC_VID_CONV_COEF0[10:0] RY
R_{Cr}	DISPC_VID_CONV_COEF0[26:16] RCR
R_{Cb}	DISPC_VID_CONV_COEF1[10:0] RCB
G_Y	DISPC_VID_CONV_COEF1[26:16] GY
G_{Cr}	DISPC_VID_CONV_COEF2[10:0] GCR
G_{Cb}	DISPC_VID_CONV_COEF2[26:16] GCB
B_Y	DISPC_VID_CONV_COEF3[10:0] BY
B_{Cr}	DISPC_VID_CONV_COEF3[26:16] BCR
B_{Cb}	DISPC_VID_CONV_COEF4[10:0] BCB
G offset	DISPC_VID_CONV_COEF5[31:19] GOFFSET
R offset	DISPC_VID_CONV_COEF5[15:3] ROFFSET
B offset	DISPC_VID_CONV_COEF6[15:3] BOFFSET

If the active range for the luminance samples (Y) is [256:3760] and [256:3840] for the chrominance samples (Cb and Cr), the range selection is done by setting the DISPC_VID_ATTRIBUTES[11] FULLRANGE bit to 0x0. The values of R, G, and B output components are clipped to the range [0:4095].

NOTE: The scaling and CSC clipping is set by the same bit, DISPC_VID_ATTRIBUTES[11] FULLRANGE.

Figure 8-30. DISPC VID YCbCr to RGB Registers (FULLRANGE = 0), 12-Bit Outputs

$$\begin{bmatrix} R_{OUT} \\ G_{OUT} \\ B_{OUT} \end{bmatrix} = \frac{1}{256} * \begin{bmatrix} R_Y & R_{cr} & R_{cb} \\ G_Y & G_{cr} & G_{cb} \\ B_Y & B_{cr} & B_{cb} \end{bmatrix} * \begin{bmatrix} Y_{IN} - 256 \\ Cr_{IN} - 2048 \\ Cb_{IN} - 2048 \end{bmatrix}$$

dispc-005

If the active range for the luminance samples (Y) and chrominance samples (Cb and Cr) is [0:4095], the range selection is done by setting the DISPC_VID_ATTRIBUTES[11] FULLRANGE bit to 0x1. The values of R, G, and B output components are clipped to the range [0:4095].

Figure 8-31. DISPC VID YCbCr to RGB Registers (FULLRANGE = 1), 12-Bit Outputs

$$\begin{bmatrix} R_{OUT} \\ G_{OUT} \\ B_{OUT} \end{bmatrix} = \frac{1}{256} * \begin{bmatrix} R_Y & R_{cr} & R_{cb} \\ G_Y & G_{cr} & G_{cb} \\ B_Y & B_{cr} & B_{cb} \end{bmatrix} * \begin{bmatrix} Y_{IN} \\ Cr_{IN} - 2048 \\ Cb_{IN} - 2048 \end{bmatrix}$$

dispc-006

8.2.4.9.4.1 DISPC VID Chrominance Resampling

The chrominance resampling is always performed using filtering (the scaler unit filter). Chrominance resampling and rescaling can be combined to support native rescaling of YUV format.

The usage of the scaler unit for resampling the chrominance of YUV4:2:0 and YUV4:2:2 is shown in Figure 8-32 and Figure 8-33, respectively. The settings of the scaler unit to perform chrominance resampling are described in Section 8.2.4.9.5, *DISPC Scaler Unit*.

Figure 8-32. DISPC VID YUV4:2:0 to RGB36 Using Scaler Unit for Resampling Chrominance

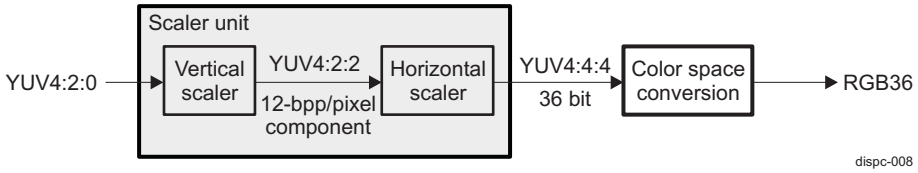
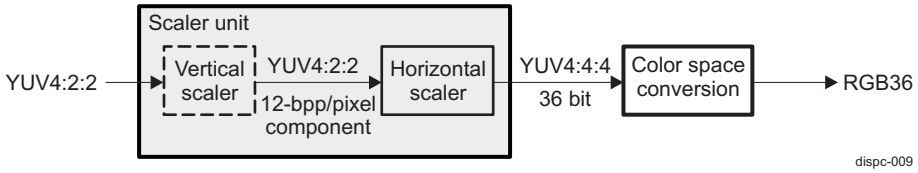


Figure 8-33. DISPC VID YUV4:2:2 to RGB36 Using Scaler Unit for Resampling Chrominance



8.2.4.9.5 DISPC VID Scaler Unit

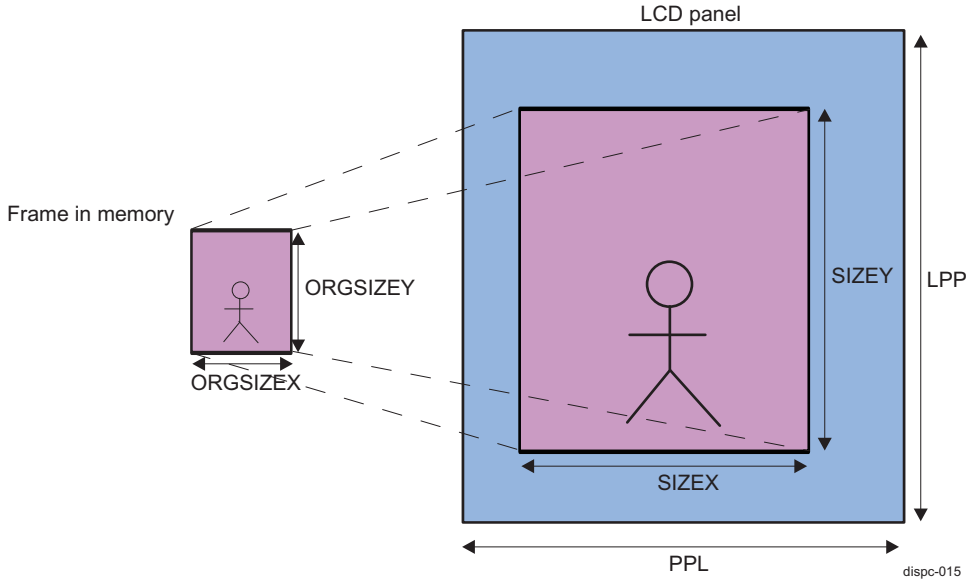
All video formats are supported, including formats with alpha blending. Alpha blending is scaled with the same parameters as RGB color components. For the YUV formats, Y and Cb/Cr are processed independently. The filter is based on a finite impulse response (FIR) filter with 16 phases. The filter is a 5-tap for horizontal filtering, and can be configured for 3 or 5 taps for vertical filtering.The filtering can be used for different processing:

- Upsampling of the picture
- Downsampling of the picture
- Anti-aliasing reduction
- Chrominance resampling in case of YUV data formats

NOTE: The user must ensure that the resizing frame displays in the screen boundaries.

Figure 8-34 shows an example of video upsampling.

Figure 8-34. DISPC Video Upsampling



The upsampling/downsampling filter is a polyphase filter with 5 taps and 16 phases for the horizontal filter, and a programmable number of taps (3 or 5) and 16 phases for the vertical filter. The input buffer has six input memory lines. The following limitations must be considered:

- The upsampling ratio is up to x16.
- The downsampling ratio using 3-tap configuration is down to x0.5 for RGB format.
- The downsampling ratio using 5-tap configuration is down to x0.25 for RGB format.
- If the input format is changed from YUV4:2:2 to YUV4:2:0 (WB pipeline), the downsampling ratio is further reduced:
  - Using 5-tap configuration, the ratio is down to x0.5 for RGB format.
  - Using 3-tap configuration, no downscaling is available.
- Scaling capability of VID pipelines and WB pipeline can be combined to double the maximum rescaling factor.

For vertical upsampling and downsampling in a 3-tap configuration, the equations are:

For RGB formats

$$A_{out}(n)=\left(\sum_{i=-1}^{i=1} C_{vi}(\Phi_y)*A_{in}(n+i)\right)>>7$$

$$R_{out}(n)=\left(\sum_{i=-1}^{i=1} C_{vi}(\Phi_y)*R_{in}(n+i)\right)>>7$$

$$G_{out}(n)=\left(\sum_{i=-1}^{i=1} C_{vi}(\Phi_y)*G_{in}(n+i)\right)>>7$$

$$B_{out}(n)=\left(\sum_{i=-1}^{i=1} C_{vi}(\Phi_y)*B_{in}(n+i)\right)>>7$$

For YUV formats

$$Y_{out}(n)=\left(\sum_{i=-1}^{i=1} C_{vi}(\Phi_y)*Y_{in}(n+i)\right)>>7$$

$$Cr_{out}(n)=\left(\sum_{i=-1}^{i=1} C_{vci}(\Phi_c)*Cb_{in}(n+i)\right)>>7$$

$$Cb_{out}(n)=\left(\sum_{i=-1}^{i=1} C_{vci}(\Phi_c)*Cr_{in}(n+i)\right)>>7$$

dispc-013

(1)

For vertical upsampling and downsampling in a 5-tap configuration, the equations are:

For RGB formats

$$A_{out}(n)=\left(\sum_{i=-2}^{i=2} C_{vi}(\Phi)*A_{in}(n+i)\right)>>7$$

$$R_{out}(n)=\left(\sum_{i=-2}^{i=2} C_{vi}(\Phi)*R_{in}(n+i)\right)>>7$$

$$G_{out}(n)=\left(\sum_{i=-2}^{i=2} C_{vi}(\Phi)*G_{in}(n+i)\right)>>7$$

$$B_{out}(n)=\left(\sum_{i=-2}^{i=2} C_{vi}(\Phi)*B_{in}(n+i)\right)>>7$$

For YUV formats

$$Y_{out}(n)=\left(\sum_{i=-2}^{i=2} C_{vi}(\Phi_y)*Y_{in}(n+i)\right)>>7$$

$$Cb_{out}(n)=\left(\sum_{i=-2}^{i=2} C_{vci}(\Phi_c)*Cb_{in}(n+i)\right)>>7$$

$$Cr_{out}(n)=\left(\sum_{i=-2}^{i=2} C_{vci}(\Phi_c)*Cr_{in}(n+i)\right)>>7$$

dispc-012

(2)

For horizontal upsampling and downsampling in a 5-tap configuration, the equations are:

For RGB formats

$$A_{out}(n)=\left(\sum_{i=-2}^{i=2} C_{hi}(\Phi)*A_{in}(n+i)\right)>>7$$

$$R_{out}(n)=\left(\sum_{i=-2}^{i=2} C_{hi}(\Phi)*R_{in}(n+i)\right)>>7$$

$$G_{out}(n)=\left(\sum_{i=-2}^{i=2} C_{hi}(\Phi)*G_{in}(n+i)\right)>>7$$

$$B_{out}(n)=\left(\sum_{i=-2}^{i=2} C_{hi}(\Phi)*B_{in}(n+i)\right)>>7$$

For YUV formats

$$Y_{out}(n)=\left(\sum_{i=-2}^{i=2} C_{hi}(\Phi_y)*Y_{in}(n+i)\right)>>7$$

$$Cb_{out}(n)=\left(\sum_{i=-2}^{i=2} C_{hci}(\Phi_c)*Cb_{in}(n+i)\right)>>7$$

$$Cr_{out}(n)=\left(\sum_{i=-2}^{i=2} C_{hci}(\Phi_c)*Cr_{in}(n+i)\right)>>7$$

dispc-014

(3)

NOTE: The pixel (n + 1) is the previous pixel with respect to pixel (n). The line (n + 1) is the previous line with respect to line (n).

The coefficients Ci() depend on the phase between input and output pixels.

NOTE: The coefficients are different for Y and Cr, Cb filtering because the calculations are independent due to the chrominance resampling for YUV4:2:2 and YUV4:2:0.

First, the vertical filter is applied to the encoded input pixel data, and then the horizontal filter is applied on the resulting pixel values to generate the output pixel values. The vertical input of the filter consists of six lines of 1280 × 32 bits for 5-tap configurations (see Table 8-34). In case of 3-tap configuration, the scaler input width capability is extended to 2560 pixels wide.

Table 8-34. DISPC VID Line Buffer Width for Scaler Unit

Vertical Taps	Maximum Input Width (Pixels)
3	2560
5	1280 (1)

(1) For the 5-tap configuration, the 6th video line is used on the output of the horizontal scaler, so that horizontal down-scaling can be supported with a minimum clock ratio equal to 1. The maximum output width is limited to 1280 pixels in order to be able to support clock ratio of 1 due to the 6th video line buffer.

At the beginning of frame scaling processing, the first line may be duplicated multiple times depending on the initial vertical phase programmed for the poly-phase filter.

At the end of frame scaling processing, the last line is duplicated if the scaling logic requires loading more lines and the last line has been reached.

Similarly, the first pixel may be duplicated multiple times depending on the initial horizontal phase programmed for the poly-phase filter. The last pixel is duplicated if the scaling logic requires loading more pixels and the last pixel has been reached.

The programmable coefficients of the polyphase filters are signed 10-bit values (except for the central coefficient, which is unsigned). The vertical video scaler has an 8-bit input and a 10-bit output. The vertical scaling changes the 8-bit input into a 10-bit clipped output, and the horizontal scaling takes the 10-bit input.

Figure 8-35 and Figure 8-36 show the scaler macro-architecture for the component A, R, G, B, and Y. Figure 8-37 and Figure 8-38 show the scaler macro-architecture for component Cr and Cb.

NOTE: The scaling and CSC clipping is set by the same bit, DISPC_VID_ATTRIBUTES[11] FULLRANGE.

Figure 8-35. DISPC VID Macro-Architecture of the Horizontal Scaling for A, R, G, B, and Y Components (5-tap Restriction)

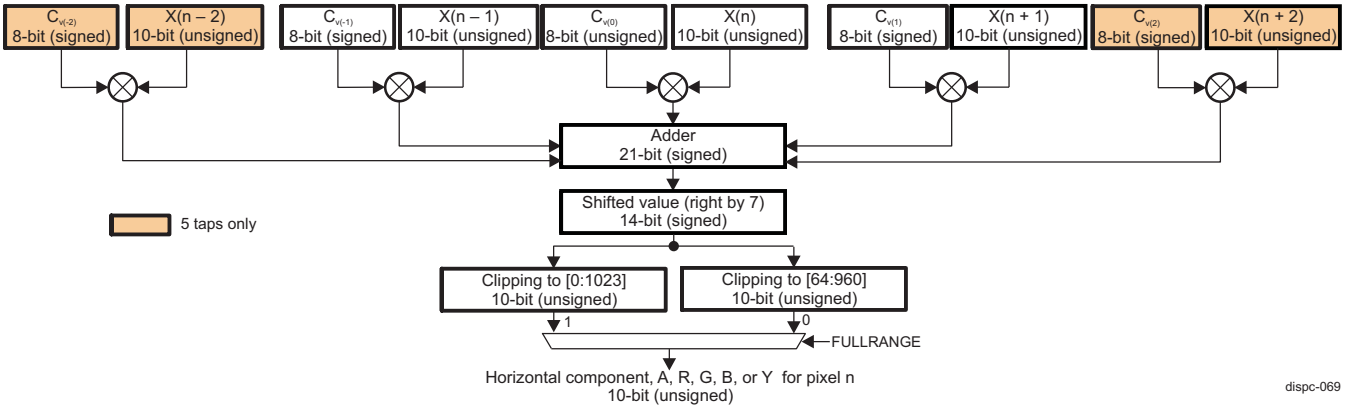


Figure 8-36. DISPC VID Macro-Architecture of the Vertical Scaling for A, R, G, B, and Y Components (5 and 3 taps)

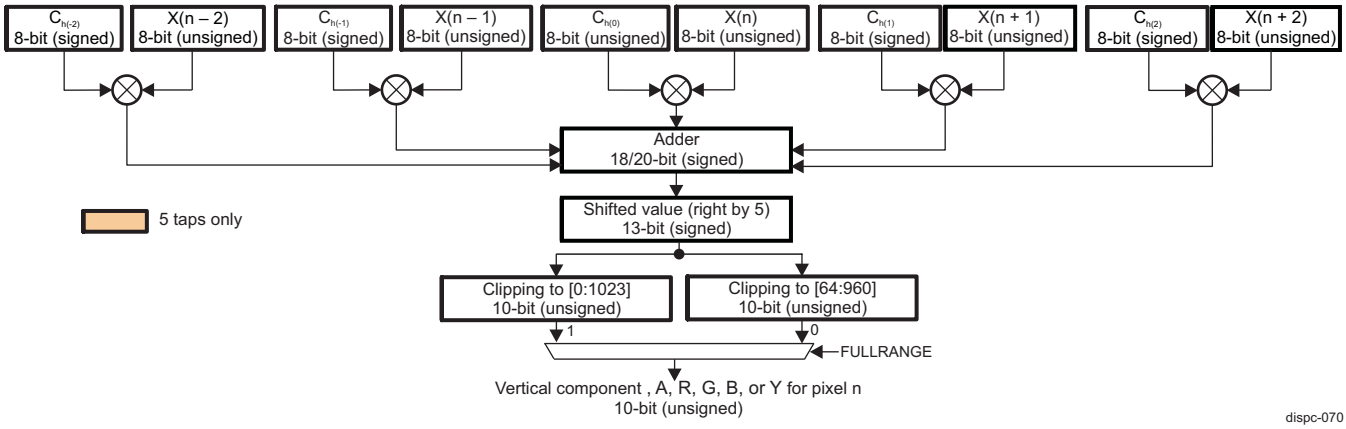


Figure 8-37. DISPC VID Macro-Architecture of the Horizontal Scaling for Cr and Cb Components (5-tap Restriction)

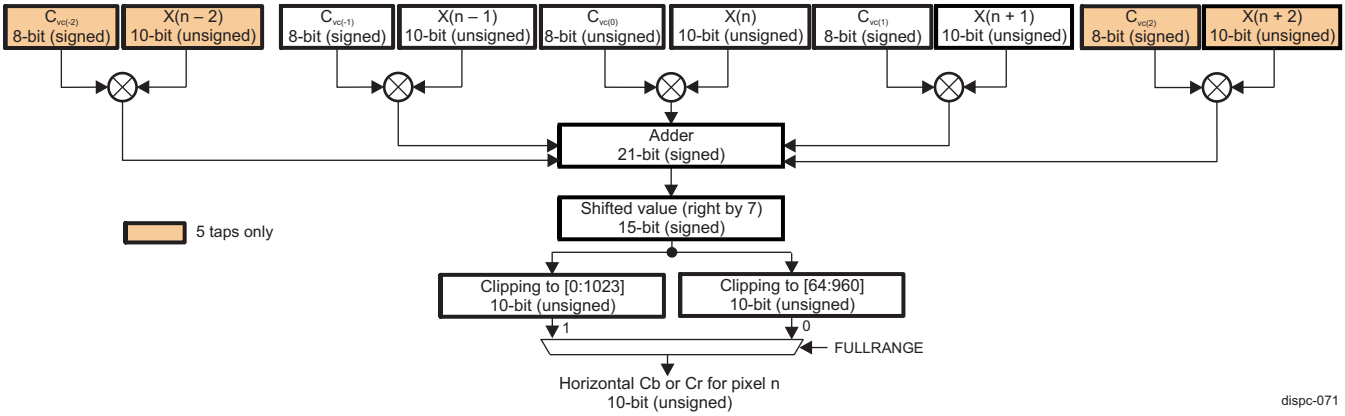


Figure 8-38. DISPC VID Macro-Architecture of the Vertical Scaling for Cr and Cb Components (5 and 3 taps)

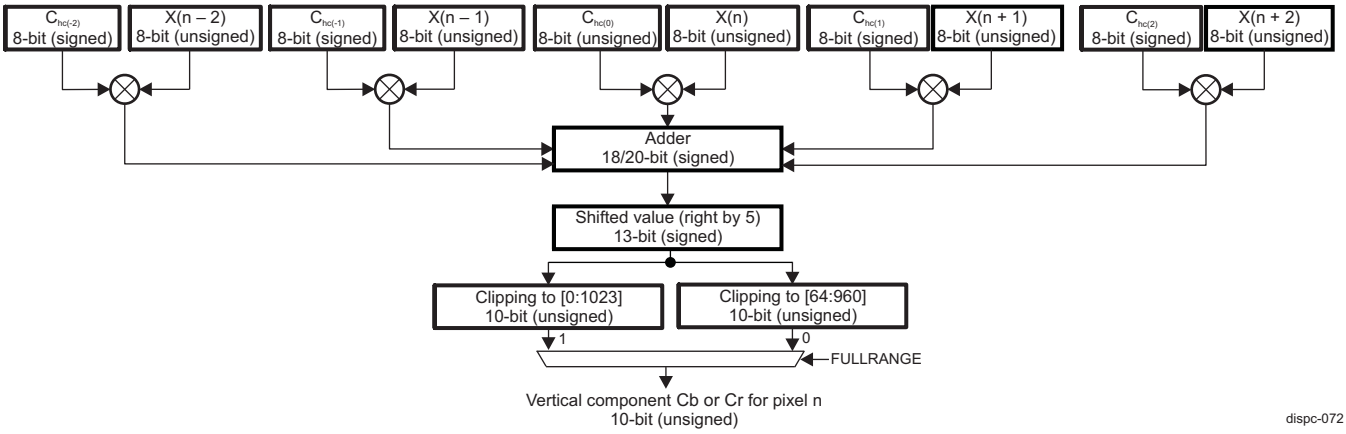


Table 8-35 list the bit fields in the function to the coefficients for the VID horizontal scaler in the DISPC_VID_FIR_COEF_H0_i and DISPC_VID_FIR_COEF_H12_k registers.

Table 8-35. DISPC Register Bit Fields Associated to Coefficients for ARGB and Y Configuration in VID Horizontal Scaler

Phases	Ch(2)	Ch(1)	Ch(0)	Ch(-1)	Ch(-2)
	Signed coefficient [29:20] FIRHC2 bitfield	Signed coefficient [19:10] FIRHC1 bitfield	Unsigned central coefficient [9:0] FIRHC0 bitfield	Signed coefifcient [19:10] FIRHC1 bitfield	Signed coefficient [29:20] FIRHC2 bitfield
0	DISPC_VID_FIR_COEF_H12_0	DISPC_VID_FIR_COEF_H12_0	DISPC_VID_FIR_COEF_H0_0	DISPC_VID_FIR_COEF_H12_0	DISPC_VID_FIR_COEF_H12_0
1	DISPC_VID_FIR_COEF_H12_1	DISPC_VID_FIR_COEF_H12_1	DISPC_VID_FIR_COEF_H0_1	DISPC_VID_FIR_COEF_H12_15	DISPC_VID_FIR_COEF_H12_15
2	DISPC_VID_FIR_COEF_H12_2	DISPC_VID_FIR_COEF_H12_2	DISPC_VID_FIR_COEF_H0_2	DISPC_VID_FIR_COEF_H12_14	DISPC_VID_FIR_COEF_H12_14
3	DISPC_VID_FIR_COEF_H12_3	DISPC_VID_FIR_COEF_H12_3	DISPC_VID_FIR_COEF_H0_3	DISPC_VID_FIR_COEF_H12_13	DISPC_VID_FIR_COEF_H12_13
4	DISPC_VID_FIR_COEF_H12_4	DISPC_VID_FIR_COEF_H12_4	DISPC_VID_FIR_COEF_H0_4	DISPC_VID_FIR_COEF_H12_12	DISPC_VID_FIR_COEF_H12_12
5	DISPC_VID_FIR_COEF_H12_5	DISPC_VID_FIR_COEF_H12_5	DISPC_VID_FIR_COEF_H0_5	DISPC_VID_FIR_COEF_H12_11	DISPC_VID_FIR_COEF_H12_11
6	DISPC_VID_FIR_COEF_H12_6	DISPC_VID_FIR_COEF_H12_6	DISPC_VID_FIR_COEF_H0_6	DISPC_VID_FIR_COEF_H12_10	DISPC_VID_FIR_COEF_H12_10

Table 8-35. DISPC Register Bit Fields Associated to Coefficients for ARGB and Y Configuration in VID Horizontal Scaler (continued)

Phases	Ch(2)	Ch(1)	Ch(0)	Ch(-1)	Ch(-2)
7	DISPC_VID_FIR_COEF_H12_7	DISPC_VID_FIR_COEF_H12_7	DISPC_VID_FIR_COEF_H0_7	DISPC_VID_FIR_COEF_H12_9	DISPC_VID_FIR_COEF_H12_9
8	DISPC_VID_FIR_COEF_H12_8	DISPC_VID_FIR_COEF_H12_8	DISPC_VID_FIR_COEF_H0_8	DISPC_VID_FIR_COEF_H12_8	DISPC_VID_FIR_COEF_H12_8
9	DISPC_VID_FIR_COEF_H12_9	DISPC_VID_FIR_COEF_H12_9	DISPC_VID_FIR_COEF_H0_7	DISPC_VID_FIR_COEF_H12_7	DISPC_VID_FIR_COEF_H12_7
10	DISPC_VID_FIR_COEF_H12_10	DISPC_VID_FIR_COEF_H12_10	DISPC_VID_FIR_COEF_H0_6	DISPC_VID_FIR_COEF_H12_6	DISPC_VID_FIR_COEF_H12_6
11	DISPC_VID_FIR_COEF_H12_11	DISPC_VID_FIR_COEF_H12_11	DISPC_VID_FIR_COEF_H0_5	DISPC_VID_FIR_COEF_H12_5	DISPC_VID_FIR_COEF_H12_5
12	DISPC_VID_FIR_COEF_H12_12	DISPC_VID_FIR_COEF_H12_12	DISPC_VID_FIR_COEF_H0_4	DISPC_VID_FIR_COEF_H12_4	DISPC_VID_FIR_COEF_H12_4
13	DISPC_VID_FIR_COEF_H12_13	DISPC_VID_FIR_COEF_H12_13	DISPC_VID_FIR_COEF_H0_3	DISPC_VID_FIR_COEF_H12_3	DISPC_VID_FIR_COEF_H12_3
14	DISPC_VID_FIR_COEF_H12_14	DISPC_VID_FIR_COEF_H12_14	DISPC_VID_FIR_COEF_H0_2	DISPC_VID_FIR_COEF_H12_2	DISPC_VID_FIR_COEF_H12_2
15	DISPC_VID_FIR_COEF_H12_15	DISPC_VID_FIR_COEF_H12_15	DISPC_VID_FIR_COEF_H0_1	DISPC_VID_FIR_COEF_H12_1	DISPC_VID_FIR_COEF_H12_1

NOTE: The table cells without color are duplicated from the grey cells.

Similar table approach applies to the vertical scaler (registers DISPC_VID_FIR_COEF_V0_i and DISPC_VID_FIR_COEF_V12_k are used).

Similar table approach applies to the coefficients for CbCr filtering in case of YUV format (registers DISPC_VID_FIR_COEF_H0_C_i and DISPC_VID_FIR_COEF_H12_C_k, and DISPC_VID_FIR_COEF_V0_C_i and DISPC_VID_FIR_COEF_V12_C_k) .

The VID scaler unit vertical or/and horizontal sampling is defined by setting/resetting the DISPC_VID_ATTRIBUTES[8:7] RESIZEENABLE bit field.

A set of configurations must be valid before enabling the video upsampling and downsampling block.

The following fields define the configuration of the video upsampling downsampling block for VID pipelines:

- Vertical upsampling and downsampling increments the value of the [23:0] FIRVINC bit field in DISPC_VID_FIRV and DISPC_VID_FIRV2 registers. The unsigned integer value range is 2^{23}. Software calculates the value using the following equation:

$$FIRVINC = 2^{21} * \left(\frac{MEMSIZEY+1}{SIZEY+1}\right)$$

dispc-066 (4)

- Horizontal upsampling and downsampling increments the value of the the [23:0] FIRHINC bit field in the DISPC_VID_FIRH and DISPC_VID_FIRH2 registers. The unsigned integer value range is 2^{23}. Software calculates the value using the following equation:

$$FIRHINC = 2^{21} * \left(\frac{MEMSIZEH+1}{SIZEH+1}\right)$$

dispc-067 (5)

- Vertical up/downsampling accumulator value, [23:0] VERTICALACCU bit field in DISPC_VID_ACCUV_j and DISPC_VID_ACCUV2_j registers: The accumulator value indicates on which phase the vertical filtering starts. The register DISPC_VID_ACCUV_0 is used for progressive output, and for interlace output DISPC_VID_ACCUV_0 and DISPC_VID_ACCUV_1 registers are used. Similarly, DISPC_VID_ACCUV2_0 and DISPC_VID_ACCUV2_1 are used in progressive or interlace output to set the accumulator value of the Cb and Cr components when scaling YUV format.
- Vertical upsampling and downsampling line buffer configuration DISPC_VID_ATTRIBUTES[21]

VERTICALTAPS bit: The default value at reset time is 0x0 (3-tap configuration is used). If the bit field is reset, the 3-tap configuration is used.

- Horizontal upsampling and downsampling accumulator value, [23:0] HORIZONTALACCU bit field in DISPC_VID_ACCUH_j and DISPC_VID_ACCUH2_j registers: The accumulator value indicates on which phase the horizontal filtering starts. The register DISPC_VID_ACCUH_0 is used for progressive output, and for interlace output the DISPC_VID_ACCUH_0 and DISPC_VID_ACCUH_1 registers are used. Similarly, DISPC_VID_ACCUH2_0 and DISPC_VID_ACCUH2_1 are used in progressive or interlace output to set the accumulator value of the Cb and Cr components when scaling YUV format.

Table 8-36 lists the DISPC vertical and horizonatal accumulator values and phases.

Table 8-36. DISPC VID Vertical and Horizontal Accumulator Phases

Accumulator Value (MSB bits)	Phases f
0	0
256 or -3840	1
512 or -3584	2
768 or -3328	3
1024 or -3072	4
1280 or -2816	5
1536 or -2560	6
1792 or -2304	7
2048 or -2048	8
2304 or -1792	9
2560 or -1536	10
2816 or -1280	11
3072 or -1024	12
3328 or -768	13
3584 or -512	14
3840 or -256	15

- Vertical upsampling and downsampling central coefficients:
  - The vertical upsampling and downsampling central coefficients are defined in the DISPC_VID_FIR_COEF_V0_i registers. There are 9 registers for the 16 phases with 1 coefficient for each of them. Symetrical implementation is used, so only 9 coefficients are used. Each register contains one 10-bit unsigned coefficient (the central one).
  - Four YUV, the vertical upsampling and downsampling central coefficients are set in DISPC_VID_FIR_COEF_V0_C_i registers.
- Vertical upsampling and downsampling coefficients:
  - The vertical upsampling and downsampling coefficients are defined in the DISPC_VID_FIR_COEF_V12_k registers. There are 16 registers for the 16 phases with 2 coefficient for each of them, so a total of 32 programmable coefficients for the vertical up/down-sampling block. Each register contains two 10-bit signed coefficients.
  - Four YUV, the vertical upsampling and downsampling coefficients are set in DISPC_VID_FIR_COEF_V12_C_k registers.
- Horizontal upsampling and downsampling central coefficients:
  - The horizontal upsampling and downsampling central coefficients are defined in the DISPC_VID_FIR_COEF_H0_i registers. There are 9 registers for the 16 phases with 1 coefficient for each of them. Symetrical implementation is used, so only 9 coefficients are used. Each register contains one 10-bit unsigned coefficient (the central one).
  - Four YUV, the horizontal upsampling and downsampling central coefficients are set in DISPC_VID_FIR_COEF_H0_C_i registers.
- Horizontal upsampling and downsampling coefficients:

– The horizontal upsampling and downsampling coefficients are defined in the DISPC_VID_FIR_COEF_H12_k registers. There are 16 registers for the 16 phases with 2 coefficient for each of them, so a total of 32 programmable coefficients for the horizontal up/down-sampling block. Each register contains two 10-bit signed coefficients.
– Four YUV, the horizontal upsampling and downsampling coefficients are set in DISPC_VID_FIR_COEF_H12_C_k registers.

8.2.4.9.5.1 DISPC Scaling Algorithms

Figure 8-39 and Figure 8-40 show details of the vertical and horizontal upsampling and downsampling finite state-machines (FSMs), respectively.

Figure 8-39. DISPC Vertical Upsampling and Downsampling Algorithm

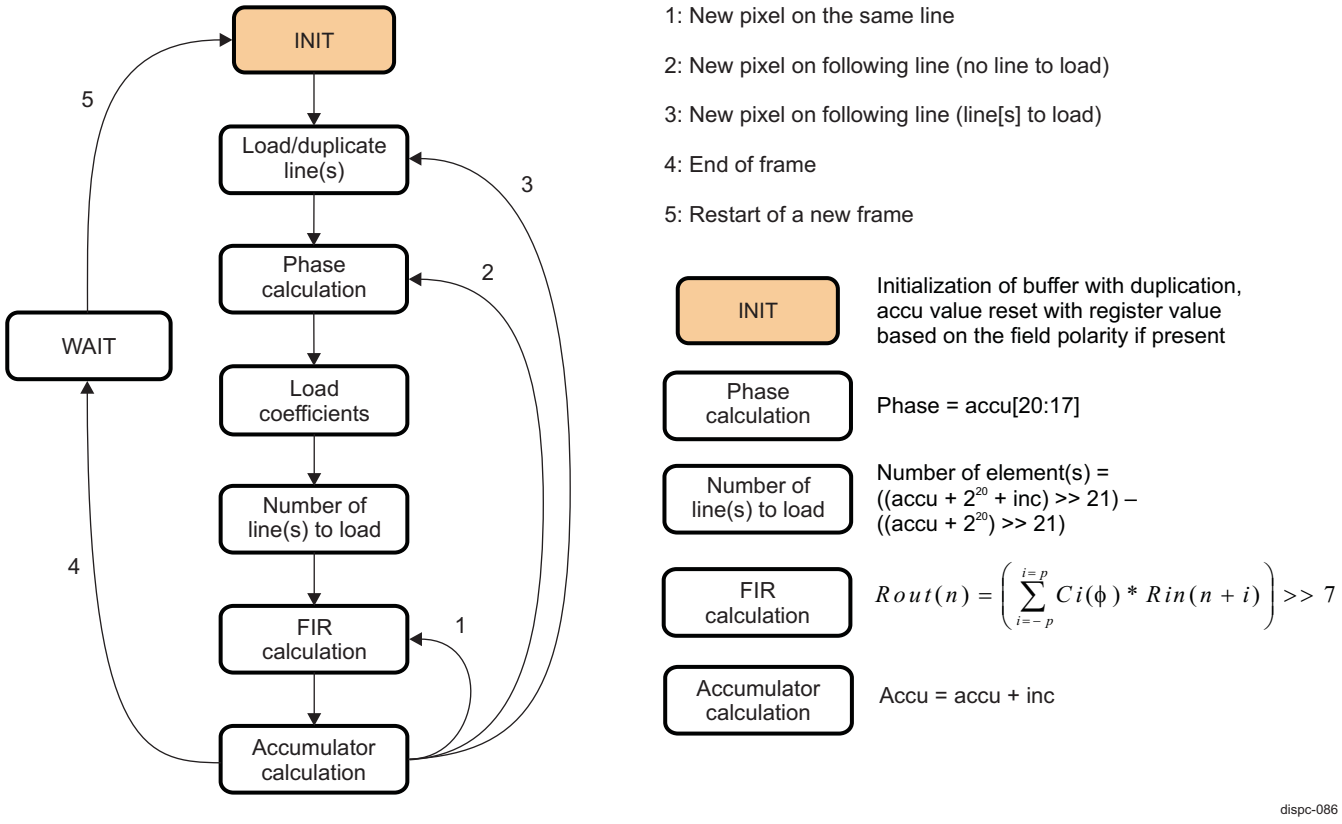
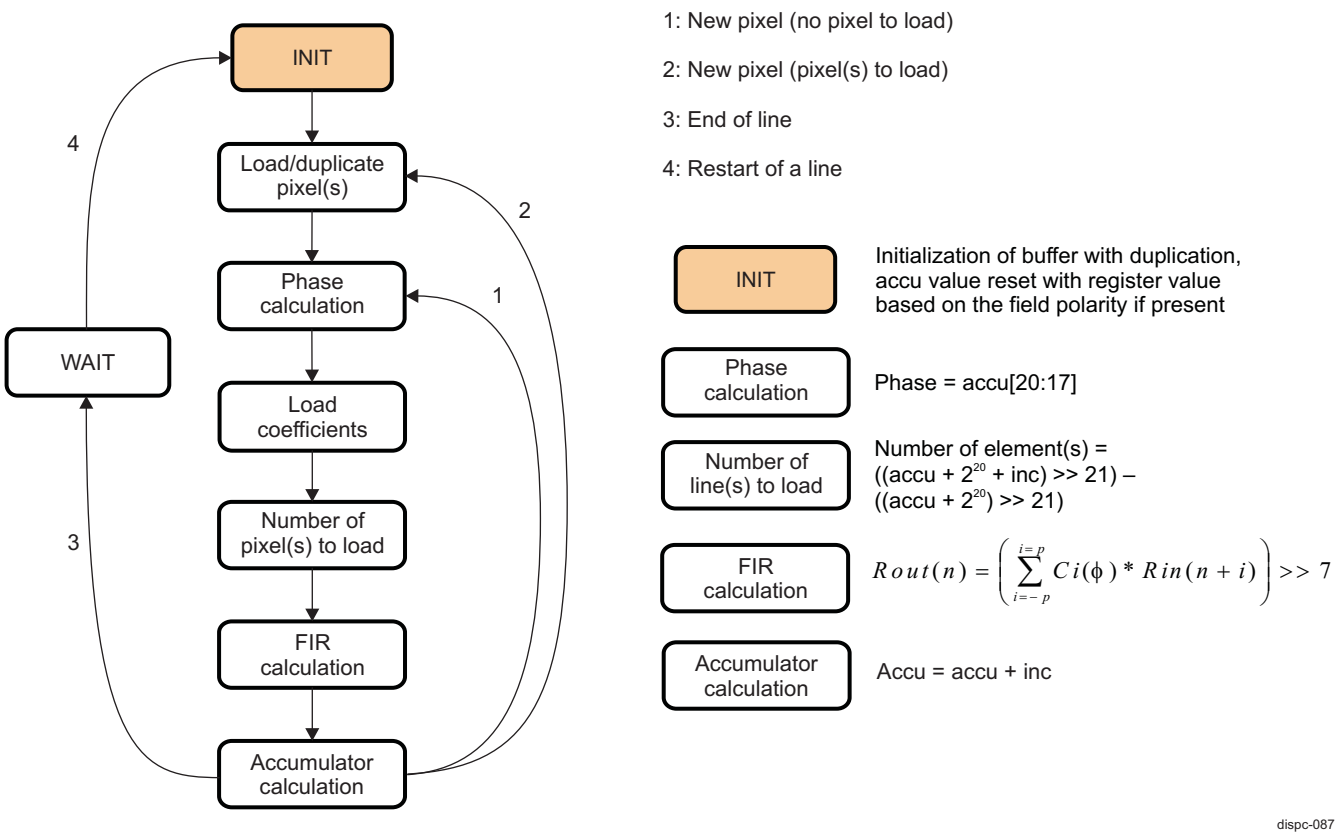


Figure 8-40 shows the horizontal up/downsampling FSM.

Figure 8-40. DISPC Horizontal Up/Downsampling Algorithm



8.2.4.9.6 DISPC VID Progressive to Interlace conversion

It is possible, in case of YUV420 format, to fetch only one out of two lines of the Y buffer and fetch all the CbCr lines in order to read a field (even lines, if the base address of the Y buffer points to an even line and odd lines, if it points to an odd line) from a YUV420 frame buffers (Y and CrCr buffers). The register configuration required is as follows:

- DISPC_VID_ATTRIBUTES[6:1] FORMAT = 0x3D (YUV420 NV12/NV21 format)
- DISPC_VID_BA_j[31] and DISPC_VID_BA_UV_j[31]= 0x0
- DISPC_VID_ATTRIBUTES[22] DOUBLESTRIDE = 0x0

8.2.4.10 DISPC Write-Back Pipeline

The write-back pipeline is used to store in the system memory the capture of one of the overlay outputs, or the output of one of the pipelines. The WB pipeline consists of a CSC unit, a scaler unit, and an RGB truncation logic. Because the overlay works on ARGB48 format and the video accelerator works on YUV format, the color space conversion from RGB to YUV is used to directly output to memory the format that can be encoded with no extra processing.

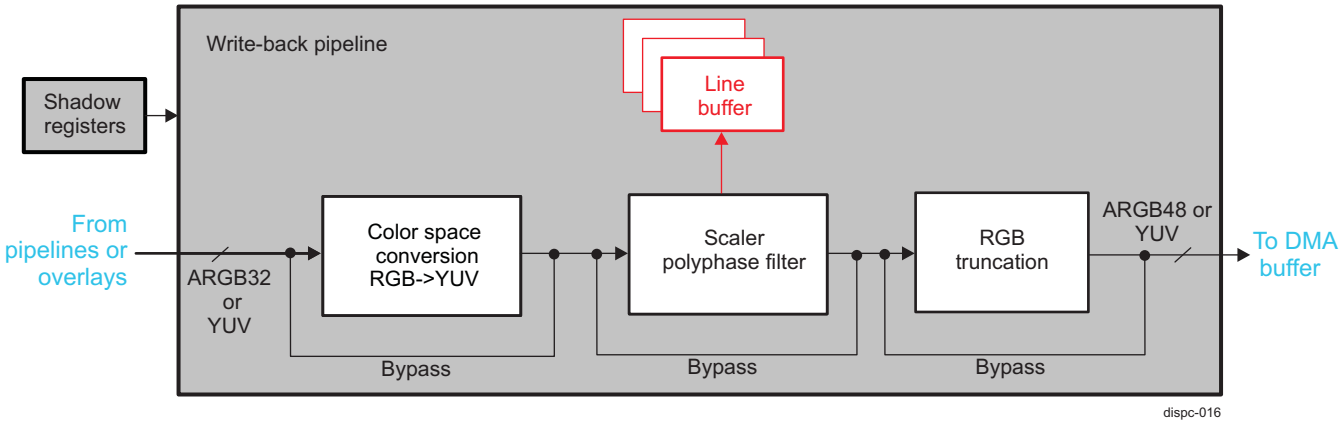
The write-back pipeline is connected to either one of the pipeline outputs (GFX, VID1 or VID2), or to one of the outputs of the overlay managers (OVR1 or OVR2). The input is selected by setting the DISPC_WB_ATTRIBUTES[18:15] CHANNELIN bit field, and the capture frame rate is set in the DISPC_WB_ATTRIBUTES[26:24] CAPTUREMODE bit field.

The output format of the overlay managers is ARGB48-1212121212. The graphics pipeline output is ARGB48. The video pipeline outputs are YUV4:2:2, YUV4:2:0, or ARGB48.

The WB pipeline is enabled by setting the DISPC_WB_ATTRIBUTES[0] ENABLE bit to 0x1.

Figure 8-41 shows the WB pipeline.

Figure 8-41. DISPC Write-Back Pipeline



8.2.4.10.1 DISPC WB CSC Unit RGB to YUV

The RGB-to-YUV CSC unit converts the encoded pixel values from RGB24 into YUV4:4:4 format. For YUV4:2:0 or YUV4:2:2 formats, a chrominance sub-sampling is required after converting the RGB into YUV values. Because of the subsampling, the following limitations must be considered:

- When converting RGB into YUV4:2:0 NV12 (NV21) format:
  - Maximum horizontal downscale = x0.5
  - Maximum vertical downscale = x0.5
- When converting RGB into YUV4:2:2 format:
  - Maximum horizontal downscale = x0.5
  - Maximum vertical downscale = x0.25

Figure 8-42 and Figure 8-43 show the 3 × 3 11-bit coefficients used to convert from RGB24 into YUV4:4:4. The user sets the coefficients according to the standard used to encode the pixel data in YUV color space. Table 8-37 lists the coefficients with their respective bit fields.

Table 8-37. DISPC WB CSC RGB to YUV Bit Field Setting

Coefficients	Bit Fields
Y_R	DISPC_WB_CONV_COEF0[10:0] YR
Y_G	DISPC_WB_CONV_COEF0[26:16] YG
Y_B	DISPC_WB_CONV_COEF1[10:0] YB
Cr_R	DISPC_WB_CONV_COEF1[26:16] CRR
Cr_G	DISPC_WB_CONV_COEF2[10:0] CRG
Cr_B	DISPC_WB_CONV_COEF2[26:16] CRB
Cb_R	DISPC_WB_CONV_COEF3[10:0] CBR
Cb_G	DISPC_WB_CONV_COEF3[26:16] CBG
Cb_B	DISPC_WB_CONV_COEF4[10:0] CBB
G offset	DISPC_WB_CONV_COEF5[31:19] GOFFSET
R offset	DISPC_WB_CONV_COEF5[15:3] ROFFSET
B offset	DISPC_WB_CONV_COEF6[15:3] BOFFSET

If the active range for the luminance samples (Y) is [16:235] and [16:240] for the chrominance samples (Cb and Cr), the values of Y, Cb, and Cr output components are clipped to the range [0:255]. The range selection is done by setting the DISPC_WB_ATTRIBUTES[12] FULLRANGE bit to 0x0.

Figure 8-42. DISPC WB RGB to YCbCr (FULLRANGE = 0)

$$\begin{bmatrix} Y_{OUT} \\ Cb_{OUT} \\ Cr_{OUT} \end{bmatrix} = \frac{1}{256} * \begin{bmatrix} Y_R & Y_G & Y_B \\ Cb_R & Cb_G & Cb_B \\ Cr_R & Cr_G & Cr_B \end{bmatrix} * \begin{bmatrix} R_{IN} \\ G_{IN} \\ B_{IN} \end{bmatrix} + \begin{bmatrix} 256 \\ 2048 \\ 2048 \end{bmatrix}$$

dispc-017

If the active range for the luminance samples (Y) and or the chrominance samples (Cb and Cr)is [0:255], the values of Y, Cb, and Cr output components are clipped to the range [0:255]. The range selection is done by setting the DISPC_WB_ATTRIBUTES[12] FULLRANGE bit to 0x1.

Figure 8-43. DISPC WB RGB to YCbCr (FULLRANGE = 1)

$$\begin{bmatrix} Y_{OUT} \\ Cb_{OUT} \\ Cr_{OUT} \end{bmatrix} = \frac{1}{256} * \begin{bmatrix} Y_R & Y_G & Y_B \\ Cb_R & Cb_G & Cb_B \\ Cr_R & Cr_G & Cr_B \end{bmatrix} * \begin{bmatrix} R_{IN} \\ G_{IN} \\ B_{IN} \end{bmatrix} + \begin{bmatrix} 0 \\ 2048 \\ 2048 \end{bmatrix}$$

dispc-018

8.2.4.10.2 DISPC WB Scaler Unit

The functional aspect of the WB pipeline scaler unit is identical to the video pipeline scaler unit (see Section 8.2.4.9.5, *DISPC VID Scaler Unit*), except in the output width when scaling ARGB components. The resulting output format is ARGB32 instead of ARGB48.

The programmable coefficients of the polyphase filters are signed 10-bit values (except for the central coefficient, which is unsigned). The WB scaler component has an 8-bit input and an 8-bit output.

Figure 8-44 and Figure 8-45 show the scaler macro-architecture for the component A, R, G, B, and Y. Figure 8-46 and Figure 8-47 show the scaler macro-architecture for component Cr and Cb.

The scaling output can be clipped to an output range of [0:255] or [16:240] by configuring the DISPC_WB_ATTRIBUTES[12] FULLRANGE bit.

Figure 8-44. DISPC WB Macro-Architecture of the Vertical Scaling for A, R, G, B, and Y Components

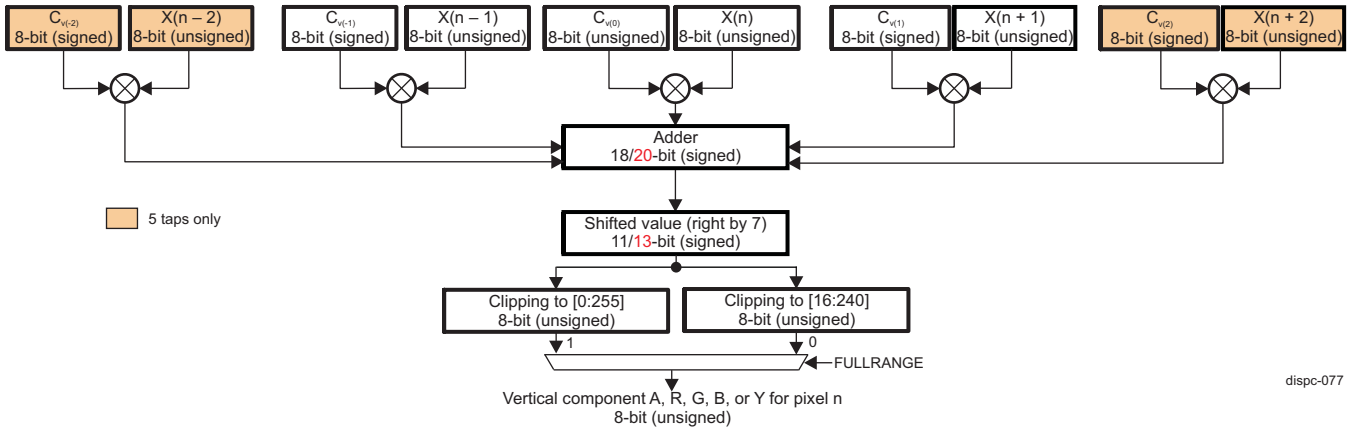


Figure 8-45. DISPC WB Macro-Architecture of the Horizontal Scaling for A, R, G, B, and Y Components

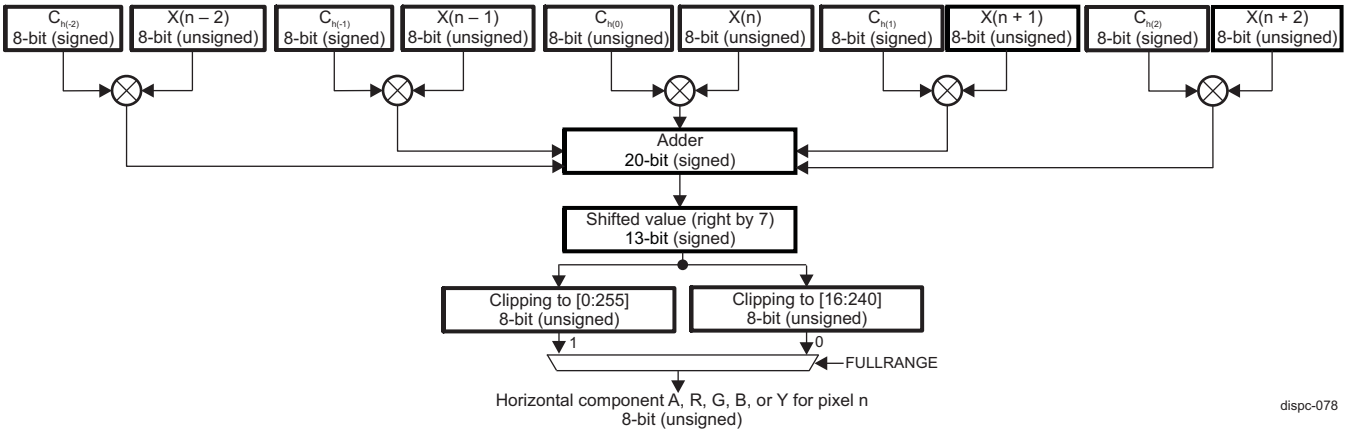


Figure 8-46. DISPC WB Macro-Architecture of the Vertical Scaling for Cr and Cb Components

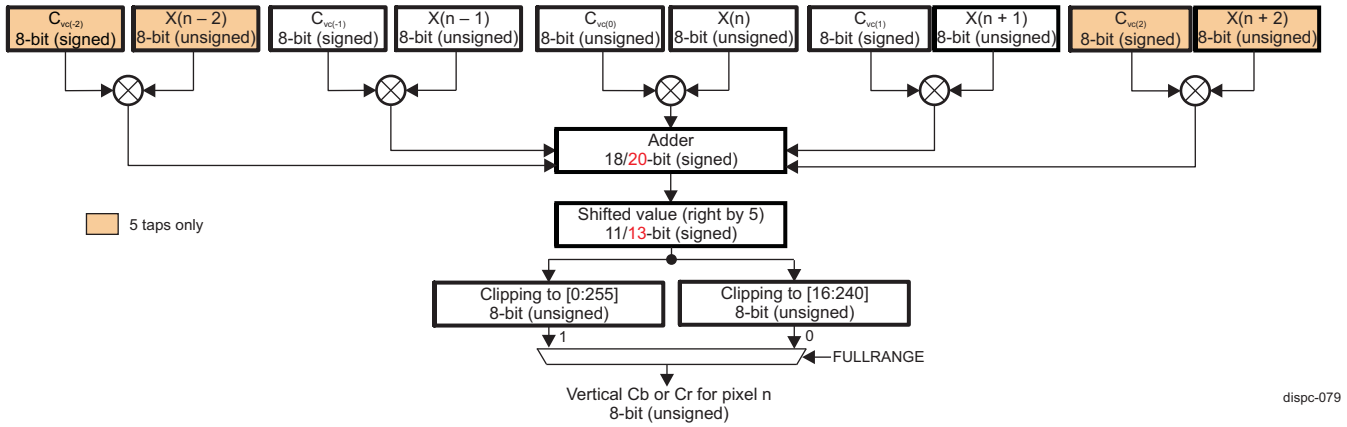


Figure 8-47. DISPC WB Macro-Architecture of the Horizontal Scaling for Cr and Cb Components

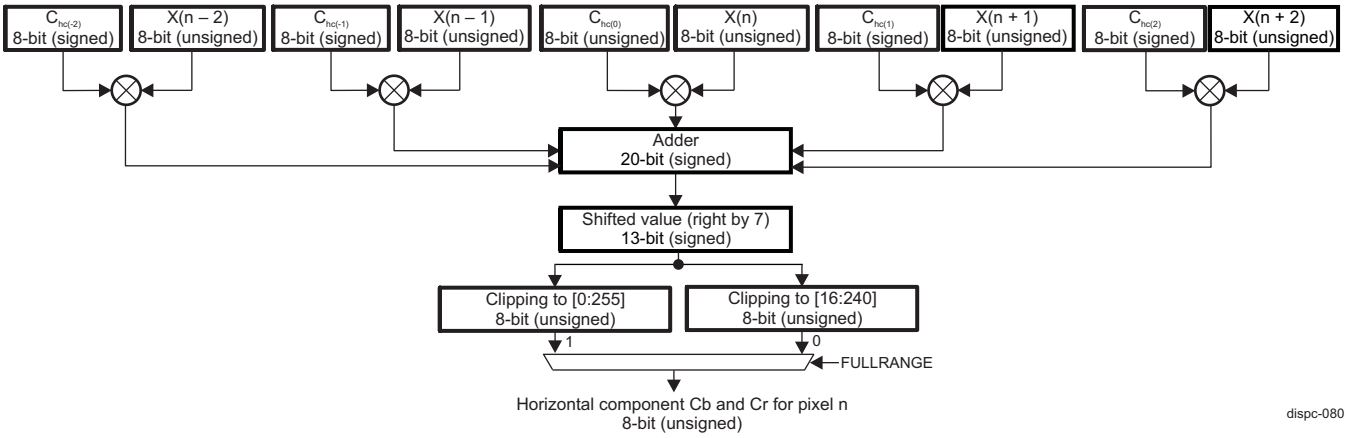


Table 8-38 list all the bit fields in the function to set the coefficients for the WB horizontal scaler in the DISPC_WB_FIR_COEF_H0_i and DISPC_WB_FIR_COEF_H12_k registers.

Table 8-38. DISPC Register Bit Fields Associated to Coefficients for ARGB and Y Configuration in WB Horizontal Scaler

Phases	Ch(2)	Ch(1)	Ch(0)	Ch(-1)	Ch(-2)
	Signed coefficient [29:20] FIRHC2 bitfield	Signed coefficient [19:10] FIRHC1 bitfield	Unsigned central coefficient [9:0] FIRHC0 bitfield	Signed coefifcient [19:10] FIRHC1 bitfield	Signed coefficient [29:20] FIRHC2 bitfield
0	DISPC_WB_FIR_COEF_H12_0	DISPC_WB_FIR_COEF_H12_0	DISPC_WB_FIR_COEF_H0_0	DISPC_WB_FIR_COEF_H12_0	DISPC_WB_FIR_COEF_H12_0
1	DISPC_WB_FIR_COEF_H12_1	DISPC_WB_FIR_COEF_H12_1	DISPC_WB_FIR_COEF_H0_1	DISPC_WB_FIR_COEF_H12_15	DISPC_WB_FIR_COEF_H12_15
2	DISPC_WB_FIR_COEF_H12_2	DISPC_WB_FIR_COEF_H12_2	DISPC_WB_FIR_COEF_H0_2	DISPC_WB_FIR_COEF_H12_14	DISPC_WB_FIR_COEF_H12_14
3	DISPC_WB_FIR_COEF_H12_3	DISPC_WB_FIR_COEF_H12_3	DISPC_WB_FIR_COEF_H0_3	DISPC_WB_FIR_COEF_H12_13	DISPC_WB_FIR_COEF_H12_13
4	DISPC_WB_FIR_COEF_H12_4	DISPC_WB_FIR_COEF_H12_4	DISPC_WB_FIR_COEF_H0_4	DISPC_WB_FIR_COEF_H12_12	DISPC_WB_FIR_COEF_H12_12
5	DISPC_WB_FIR_COEF_H12_5	DISPC_WB_FIR_COEF_H12_5	DISPC_WB_FIR_COEF_H0_5	DISPC_WB_FIR_COEF_H12_11	DISPC_WB_FIR_COEF_H12_11
6	DISPC_WB_FIR_COEF_H12_6	DISPC_WB_FIR_COEF_H12_6	DISPC_WB_FIR_COEF_H0_6	DISPC_WB_FIR_COEF_H12_10	DISPC_WB_FIR_COEF_H12_10
7	DISPC_WB_FIR_COEF_H12_7	DISPC_WB_FIR_COEF_H12_7	DISPC_WB_FIR_COEF_H0_7	DISPC_WB_FIR_COEF_H12_9	DISPC_WB_FIR_COEF_H12_9
8	DISPC_WB_FIR_COEF_H12_8	DISPC_WB_FIR_COEF_H12_8	DISPC_WB_FIR_COEF_H0_8	DISPC_WB_FIR_COEF_H12_8	DISPC_WB_FIR_COEF_H12_8
9	DISPC_WB_FIR_COEF_H12_9	DISPC_WB_FIR_COEF_H12_9	DISPC_WB_FIR_COEF_H0_7	DISPC_WB_FIR_COEF_H12_7	DISPC_WB_FIR_COEF_H12_7
10	DISPC_WB_FIR_COEF_H12_10	DISPC_WB_FIR_COEF_H12_10	DISPC_WB_FIR_COEF_H0_6	DISPC_WB_FIR_COEF_H12_6	DISPC_WB_FIR_COEF_H12_6
11	DISPC_WB_FIR_COEF_H12_11	DISPC_WB_FIR_COEF_H12_11	DISPC_WB_FIR_COEF_H0_5	DISPC_WB_FIR_COEF_H12_5	DISPC_WB_FIR_COEF_H12_5
12	DISPC_WB_FIR_COEF_H12_12	DISPC_WB_FIR_COEF_H12_12	DISPC_WB_FIR_COEF_H0_4	DISPC_WB_FIR_COEF_H12_4	DISPC_WB_FIR_COEF_H12_4
13	DISPC_WB_FIR_COEF_H12_13	DISPC_WB_FIR_COEF_H12_13	DISPC_WB_FIR_COEF_H0_3	DISPC_WB_FIR_COEF_H12_3	DISPC_WB_FIR_COEF_H12_3
14	DISPC_WB_FIR_COEF_H12_14	DISPC_WB_FIR_COEF_H12_14	DISPC_WB_FIR_COEF_H0_2	DISPC_WB_FIR_COEF_H12_2	DISPC_WB_FIR_COEF_H12_2
15	DISPC_WB_FIR_COEF_H12_15	DISPC_WB_FIR_COEF_H12_15	DISPC_WB_FIR_COEF_H0_1	DISPC_WB_FIR_COEF_H12_1	DISPC_WB_FIR_COEF_H12_1

NOTE: The Table 8-38 cells without color are duplicated from the grey cells.

Similar table approach applies to the vertical scaler (registers DISPC_WB_FIR_COEF_V0_i and DISPC_WB_FIR_COEF_V12_k).

Similar table approach applies to the coefficients for CbCr filtering in case of YUV format (registers DISPC_WB_FIR_COEF_H0_C_i and DISPC_WB_FIR_COEF_H12_C_k, and DISPC_WB_FIR_COEF_V0_C_i and DISPC_WB_FIR_COEF_V12_C_k) .

The WB scaler unit vertical or/and horizontal sampling is defined by setting/resetting the DISPC_WB_ATTRIBUTES[8:7] RESIZEENABLE bit field.

A set of configuration must be valid before enabling the video up/downsampling block.

The following fields define the configuration of the video up/downsampling block for WB:

- Vertical up/downsampling increments the value of [23:0] FIRVINC bitfield in the DISPC_WB_FIRV and DISPC_WB_FIRV2 registers. The unsigned integer value range is [1:2^{23}]. Software calculates the value using the following equation:

$$FIRVINC = 2^{21} * \left(\frac{SIZEY+1}{MEMSIZEY+1} \right)$$

dispc-066wb (6)

- Horizontal up/downsampling increments the value of the [23:0] FIRHINC bit field in the DISPC_WB_FIRH and DISPC_WB_FIRH2 registers. The unsigned integer value range is [1:2^{23}]. Software calculates the value using the following equation:

$$FIRHINC = 2^{21} * \left(\frac{SIZEX+1}{MEMSIZEX+1} \right)$$

dispc-067wb (7)

- Vertical up/downsampling accumulator value, [23:0] VERTICALACCU bit field in DISPC_WB_ACCUV_j and DISPC_WB_ACCUV2_j registers. The accumulator value indicates on which phase the vertical filtering starts. The register DISPC_WB_ACCUV_0 is used for progressive output, and for interlace output the DISPC_WB_ACCUV_0 and DISPC_WB_ACCUV_1 registers are used. Similarly, DISPC_WB_ACCUV2_0 and DISPC_WB_ACCUV2_1 are used in progressive or interlace output to set the accumulator value of the Cb and Cr components when scaling YUV format.
- Vertical up/downsampling line buffer configuration DISPC_WB_ATTRIBUTES[21] VERTICALTAPS bit: The default value at reset time is 0x0 (3-tap configuration is used). If the bit field is reset, the 3-tap configuration is used.
- Horizontal up/downsampling accumulator value, [23:0] HORIZONTALACCU bit field in DISPC_WB_ACCUH_j and DISPC_WB_ACCUH2_j registers. The accumulator value indicates on which phase the horizontal filtering starts. The register DISPC_WB_ACCUH_0 is used for progressive output, and for interlace output the DISPC_WB_ACCUH_0 and DISPC_WB_ACCUH_1 registers are used. Similarly, DISPC_WB_ACCUH2_0 and DISPC_WB_ACCUH2_1 are used in progressive or interlace output to set the accumulator value of Cb and Cr components when scaling YUV format.

Table 8-39 lists the DISPC WB vertical and horizontal accumulator values and phases.

Table 8-39. DISPC WB Vertical/Horizontal Accumulator Phase

Accumulator Value (MSB bits)	Phases f
0	0
256 or -3840	1
512 or -3584	2
768 or -3328	3
1024 or -3072	4
1280 or -2816	5
1536 or -2560	6
1792 or -2304	7
2048 or -2048	8
2304 or -1792	9
2560 or -1536	10
2816 or -1280	11
3072 or -1024	12
3328 or -768	13
3584 or -512	14
3840 or -256	15

- Vertical up/downsampling coefficients:
  - The vertical up/downsampling central coefficients are defined in the DISPC_WB_FIR_COEF_V0_i registers. There are 9 registers for the 16 phases with 1 coefficient for each of them. Symetrical implementation is used, so only 9 coefficients are required. Each register contains 10-bit unsigned coefficient (the central one).
  - The vertical up/downsampling coefficients are defined in the DISPC_WB_FIR_COEF_V12_k

registers. There are 16 registers for the 16 phases with 2 coefficients for each of them, so a total of 32 programmable coefficients for the vertical up/downsampling block are used. Each register contains two 10-bit signed coefficients.

Four YUV, the vertical up/downsampling coefficients are set in the DISPC_WB_FIR_COEF_V0_C_i and DISPC_WB_FIR_COEF_V12_C_k registers. Table 8-38 summarizes all coefficients and their respective registers.

- Horizontal up/downsampling coefficients:
  - The horizontal up/downsampling central coefficients are defined in the DISPC_WB_FIR_COEF_H0_i registers. There are 9 registers for the 16 phases with 1 coefficient for each of them. Symetrical implementation is used, so only 9 coefficients are required. Each register contains 10-bit unsigned coefficient (the central one).
  - The vertical up/downsampling coefficients are defined in the DISPC_WB_FIR_COEF_H12_k registers. There are 16 registers for the 16 phases with 2 coefficients for each of them, so a total of 32 programmable coefficients for the vertical up/downsampling block are used. Each register contains two 10-bit signed coefficients.

  Four YUV horizontal up/downsampling coefficients are set in the DISPC_WB_FIR_COEF_H0_C_i and DISPC_WB_FIR_COEF_H12_C_k registers. Table 8-38 summarizes all coefficients and their respective registers.

8.2.4.10.3 DISPC WB RGB Truncation Logic

Truncation logic is used to convert a pixel from ARGB 32-bit format into a lower color depth: 12- or 16-bit format based. The truncation is done by removing the necessary LSB of each component to match the output format.

NOTE: If there is no alpha field in the pixel format description, 0s are used. For example, in RGB12 pixel format, the upper 4 bits are set to 0s because the RGB value is only 12 bits inside a 16-bit container.

8.2.4.11 DISPC Region-Based Mechanism

8.2.4.11.1 Region-Based Mechanism Overview

The region-based mechanism is used in order to display multiple windows using the same layer. The layer can be graphics layer or a video layer. The reconfiguration of a pipeline is performed based on the different region-based interrupts (see Section 8.2.4.4, *DISPC Interrupt Requests*). When all the pixels for the current window have been output, then the pipeline uses the new configuration provided through the shadow registers in order to display the new window re-using the same layer.

It is possible to display unlimited number of windows on the screen using same layer considering the following restriction: only one window horizontally per layer at a given time.

In order to enable the feature, the SW needs to set to 0x1 the [24] REGION_BASED bit within DISPC_GFX1_ATTRIBUTES2, DISPC_VID_ATTRIBUTES2 or DISPC_WB_ATTRIBUTES2 registers.

For the WB pipeline, an additional register DISPC_WB_REGION_BASED_TOTAL_PICTURE_SIZEY, defining the total SIZEY value (total number of lines) of all the windows written back to the memory, needs to be programmed when the Region-based mechanism is enabled for the WB pipeline.

8.2.4.11.2 Region-Based Mechanism for a Single Region Write-Back

A typical use-case for the Region-Based Mechanism in the Write-Back pipeline is to capture one single region at a time per frame, while doing write-back in capture mode. This allows writing back only the region of interest from the entire frame displayed, thereby reducing the write-back bandwidth compared to when the whole frame is written back in a non region-based mode.

In a generic write-back region-based mechanism multiple regions are captured per frame. The last region to capture in a frame is indicated to the HW by resetting the DISPC_WB_ATTRIBUTES2[24] REGION_BASED bit. Due to this, when only a single region capture is needed, a specific programming sequence needs to be followed, which is described in Table 8-40, *DISPC Write-Back Region-Based Configuration Steps*.

Table 8-40 summarizes the configuration steps when the region-based mechanism for a single region is enabled for the WB pipeline.

Table 8-40. DISPC Write-Back Region-Based Configuration Steps

Step	Register/Bit-Field	Value
Initial programming (the very first frame)		
Enable the regional-based feature for WB pipeline.	DISPC_WB_ATTRIBUTES2[24] REGION_BASED	0x1
Configure the attributes of the region.	DISPC_WB_POSITION DISPC_WB_SIZE DISPC_WB_PICTURE_SIZE	- - -
Configure the total number of lines to be captured in the frame to be same as the number of lines in the region DISPC_WB_REGION_BASED_TOTAL_PICTURE_SIZEY.	DISPC_WB_REGION_BASED_TOTAL_PICTURE_SIZEY = DISPC_WB_PICTURE_SIZE[27:16] MEMSIZEY	-
Configure all other shadow registers, associated with the use case.		
Enable WBREGIONBASEDEVENT_IRQ.	DISPC_WB_IRQENABLE[3] WBREGIONBASEDEVENT_EN	0x1
Enable WBSYNC_IRQ.	DISPC_WB_IRQENABLE[4] WBSYNC_EN	0x1
Enable the Video Port (VP1) output.	DISPC_VP1_CONTROL[0] VPENABLE	0x1
Wait for the GOBIT bit to get reset by HW.	DISPC_VP1_CONTROL[5] GOBIT	0x0
From here on the region-based feature is active and will trigger both WBREGIONBASEDEVENT_IRQ and WBSYNC_IRQ every frame until the feature is completely disabled.		
The following actions need to be taken by SW in WBSYNC_IRQ and WBREGIONBASEDEVENT_IRQ routines every time those interrupts are triggered, until region-based feature is disabled.		
In WBSYNC_IRQ routine, reset the regional-based bit.	DISPC_WB_ATTRIBUTES2[24] REGION_BASED	0x0
In WBREGIONBASEDEVENT_IRQ routine, set the region-based bit.	DISPC_WB_ATTRIBUTES2[24] REGION_BASED	0x1
If the attributes of the region to be captured during next frame (position, size, base address) are different than the current region, then the corresponding changes must applied to WB registers.		

8.2.4.12 DISPC Overlay Managers

DISPC implements two identical Overlay Managers (see Figure 8-24, *DISPC Architecture Overview*):

- OVR1, with output assigned to the Video Port (VP1).
- OVR2, with output asiggned to the WB pipeline.

The overlay mechanism consists of displaying more than one layer (GFX, VID1 or VID2 pipeline layers) using:

- A priority rule based on a Z-order: Application can set the ordering layer of the frames.
- Transparency color keys: Destination and source transparency color keys can be set.
- Alpha blending values: Using the A component of a pixel or a blending set by the user for a layer, a level of transparency can be determined.

All pipelines (GFX, VID1 and VID2) can be assigned to one of the overlay managers (OVR1 or OVR2), or directly to the WB pipeline. The overlay managers can be connected to all pipeline outputs simultaneously. The GFX pipeline output is directed using the DISPC_GFX1_ATTRIBUTES[10:8] CHANNELOUT bitfield, while the VID1 and VID2 pipeline outputs are directed using the DISPC_VID_ATTRIBUTES[16:14] CHANNELOUT bit fields. Table 8-41 summarizes the bit field settings to direct a pipeline to the OVR1, OVR2 or WB. The default register value directs all pipelines to OVR1.

Table 8-41. DISPC Pipeline Connection to VP1 or WB Output

	DISPC_GFX1_ATTRIBUTES	DISPC_VID_ATTRIBUTES
Overlay Manager/Output	**[10:8] CHANNELOUT Bitfield**	**[16:14] CHANNELOUT Bitfield**
OVR1/VP1	0x0	0x0
OVR2	0x1	0x1
WB	0x4	0x4

The output of OVR1 is connected to VP1 Color Phase Rotation (CPR) unit through the Gamma table.

NOTE:

- When the pixel format is ARGB or RGBA, the color key match logic uses only the RGB value defined by ARGB or RGBA. The alpha blending factor is ignored.

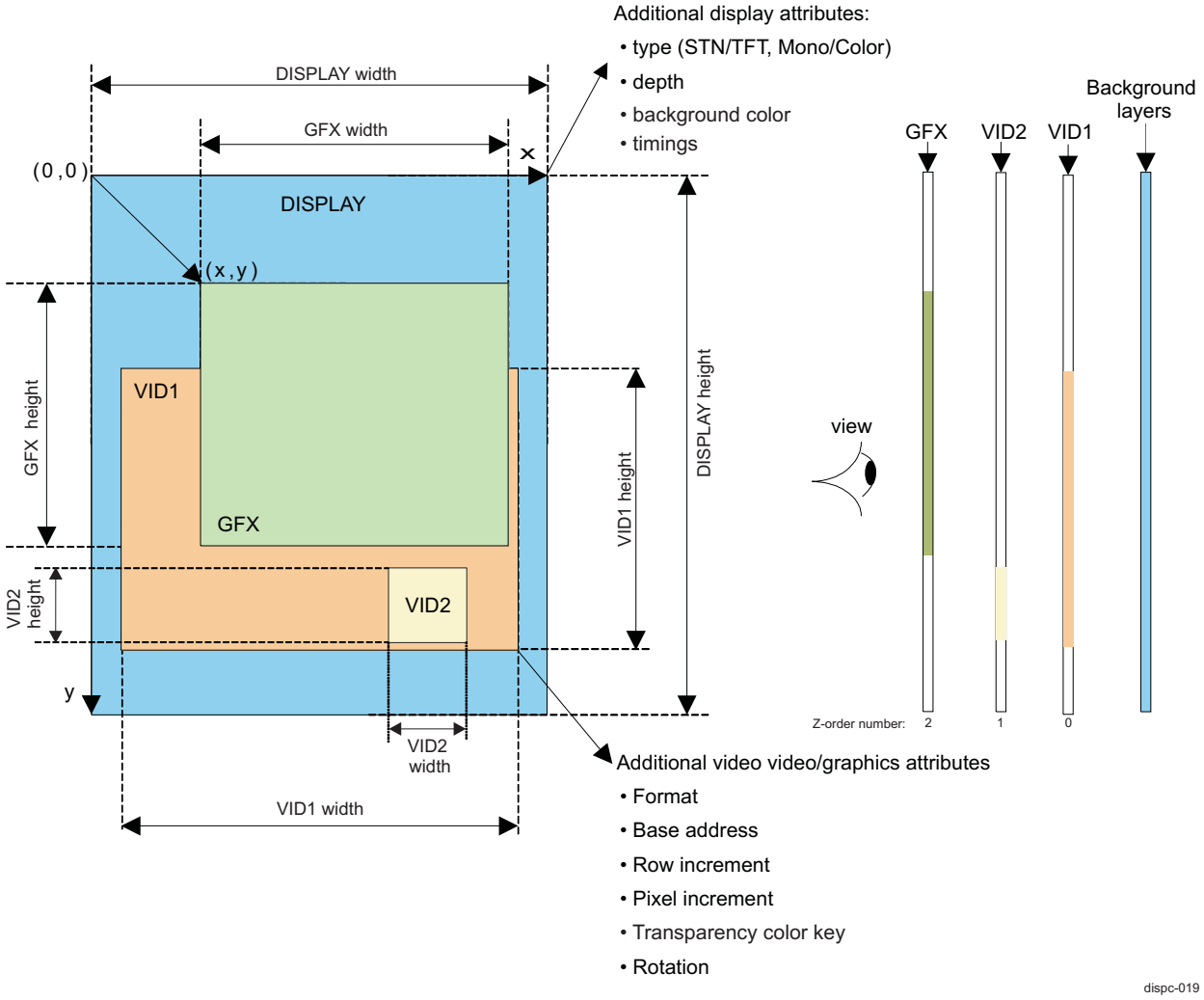
8.2.4.12.1 DISPC Overlay Priority Rule

The overlay manager is configured using the Z-order parameter. The Z-order value defined for each pipeline indicates the visibility order to the window on the screen. If the Z-order value of window A is lower than the Z-order to layer B, layer A is displayed below layer B. The transparency color keys and the alpha blending factors are then used to blend the layers together (see Section 8.2.4.12.2, *DISPC Alpha Blender*, and Section 8.2.4.12.3, *DISPC Transparency Color Keys*). The Z-order for each pipeline is defined in the DISPC_GFX1_ATTRIBUTES[27:25] ZORDER and DISPC_VID_ATTRIBUTES[27:25] ZORDER register bit fields. Each ZORDER bit-field setting must be different for each active pipeline. It is not possible to use the same value for more than one pipeline.

The height and width of each enabled layer (pipeline) must be defined in the [11:0] SIZEX and [27:16] SIZEY bit fields of DISPC_GFX1_SIZE and DISPC_VID_SIZE registers. The x and y positions of each layer must be defined in the [11:0] POSX and [27:16] POSY bit fields of DISPC_GFX1_POSITION and DISPC_VID_POSITION registers. If there are no graphics or video-encoded pixels at a specific position, the programmable, solid background color appears. The solid background color is set in the DISPC_OVR_DEFAULT_COLOR register. Figure 8-48 is an example of priority rule.

The Z-order reordering block must always map the pipelines to the blender logic in the same order - from background to foreground.

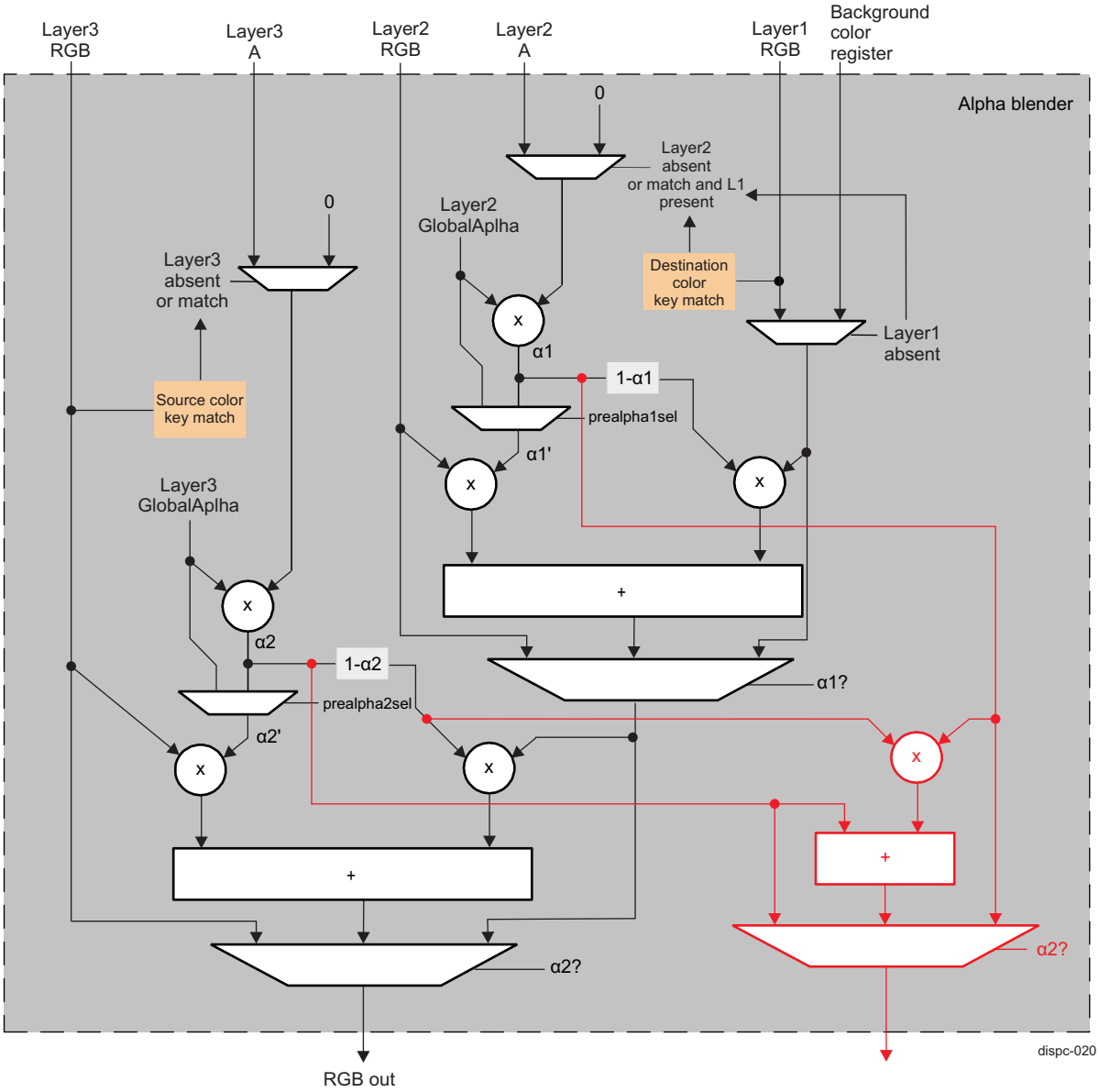
Figure 8-48. DISPC Overlay Example of Priority Rule: From Lower to Higher VID1, VID2, GFX



8.2.4.12.2 DISPC Overlay Alpha Blender

Figure 8-49 shows the alpha blending processing in detail.

Figure 8-49. DISPC Overlay Alpha Blending Architecture With Premultiplied Alpha Support



CAUTION

The Z order of the Source Transparency Layer must have the value 2 (0x2: Z-order 2: layer above all the other layers).

NOTE: 1-alpha operator corresponds to the basic 1s-complement operation.

The alpha blending value is defined by:

- The component value A when using an ARGB or RGBA pixel format.
  - For ARGB-1555, the alpha blending is defined using a 1-bit value. It is converted into an 8-bit value by duplicating the 1-bit value (see Table 8-42).
  - For ARGB-4444, the alpha blending is defined using a 4-bit value. It is converted into an 8-bit value by duplicating the 4-bit value (see Table 8-42).

  - If the pixel format contains no alpha blending value, the pixel alpha value is considered to be 0xFF, and if alpha is equal to 0xFF, there is no multiplication.
  - For BITMAP or YUV formats, there is no alpha blending factor associated with each pixel value. Only the global alpha blending factor associated with the window displaying the BITMAP or YUV format is used.
- Each 12x12 multiplication gives a result on 24 bits, which are shifted to the right to become an 12-bit value. All the values are unsigned.
- The global alpha blending value is set in the DISPC_GFX1_GLOBAL_ALPHA and DISPC_VID_GLOBAL_ALPHA registers, and is updated in synch with the selected output channel.

Table 8-42 lists the percentage of alpha blending in the function of the alpha blending value on 8 bits.

Table 8-42. DISPC Overlay Alpha Blending – ARGB

Alpha Blending 1-Bit Value (ARGB16-1555)	Alpha Blending 4-Bit Value (ARGB16-4444)	Alpha Blending 8-Bit Value (Converted Value or Resulting Alpha)	Percent Blending
0x0	0x0	0x00	100 (transparent)
N/A	0x1	0x11	93.33
N/A	0x2	0x22	86.6
N/A	...	...	...
N/A	0xE	0xEE	6.6
0x1	0xF	0xFF	0 (opaque)

Premultiplied Alpha

The image ARGB may have its RGB component already premultiplied with the alpha (ARGB) where:

- R = A*R
- G = A*G
- B = A*B

In that case, the processing is as follows:

- Color component of premultiplied layers are multiplied with the Global Alpha, if Global Alpha is not equal to 0.
- Color component of the composed underlying layers are multiplied with (1 - A × Global Alpha).

The additional premultiplied alpha option is associated with the pipelines GFX, VID1 and VID2. The option is accessible through the [28] PREMULTIPLYALPHA bit for the respective pipeline register:

- DISPC_GFX1_ATTRIBUTES
- DISPC_VID_ATTRIBUTES

The following settings are available:

- PREMULTIPLYALPHA bit = 0: Source is not premultiplied with alpha. Full blending is done in the DISPC.
- PREMULTIPLYALPHA bit = 1: Source is premultiplied with alpha. Partial blending is done.

NOTE: The *prealphasel* controls in Figure 8-49, correspond to the PREMULTIPLYALPHA of the pipelines mapped on the respective layers.

The logic marked in red in Figure 8-49 corresponds to the alpha value, computed when the write-back channel copies back to memory the premultiplied color component:
A (destination) = A (source) + (1-A (source) × A (destination))

The default value for DISPC_WB_ATTRIBUTES[7] ALPHAENABLE bit is 0x0. When the WB is configured to copy back one of the output channels (output of overlay), the following configurations are available:

- The ALPHAENABLE bit is set to 0x1: The WB pipe copies back to memory the premultiplied alpha calculated through the overlay.

- The ALPHAENABLE bit is set to 0x0: The alpha value is not written back.

NOTE: The DISPC_WB_ATTRIBUTES[7] ALPHAENABLE bit is effective only when one of the output channels is written back, otherwise it is ignored.

8.2.4.12.3 DISPC Overlay Transparency Color Keys

There are two transparency color keys: the source transparency color key and the destination transparency color key. The source transparency color key can be used with BITMAP formats (1-, 2-, 4-, and 8-bpp), YUV, and RGB formats (ARGB, RGB, RGBA, xRGB, and RGBx). In this case the A information is ignored for the comparison between the pixel value and the color key value. It is possible to use YUV formats with some caution, because the comparison is between the input pixel value of the overlay manager from pipeline (GFX or one of the VID pipelines depending on the Z-order) and the color key value. The YUV data is converted to RGB format. If the original format is YUV, the user must consider the color space conversion processing to define the RGB color key value used for the comparison.

The transparency color key is enabled by setting the following bits to 0x1:

- DISPC_OVR_CONFIG[10] TCKLCDENABLE

The transparency color key is determined in the TRANSCOLORKEY bit fields of the following registers:

- DISPC_OVR_TRANS_COLOR_MAX, DISPC_OVR_TRANS_COLOR_MAX2, DISPC_OVR_TRANS_COLOR_MIN and DISPC_OVR_TRANS_COLOR_MIN2

NOTE:

- The video source transparency color key and graphics destination transparency color key cannot be active at the same time.
- For CLUT bitmaps, the palette index is compared to the transparency color key and not to the palette value pointed out by the palette index.

- **Source transparency color key**

  The value of the video source transparency color key defines the range of encoded pixel data considered as a transparent pixel. The encoded pixel values with the source color key value inside the valid range are not visible, and the encoded pixel values of the under layers or solid background color are visible.

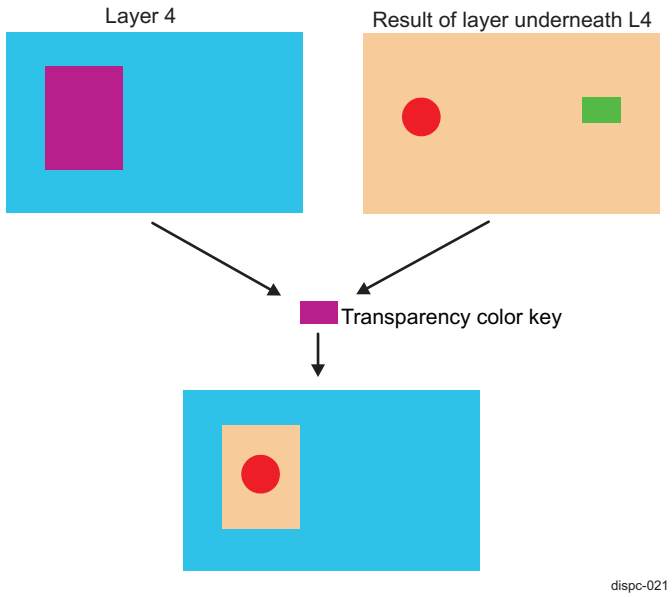
  The scaler can be enabled as a preprocessor in the VID pipelines, but it is necessary to consider the pixel scaling preprocessing in order to define the color key value to be used after the rescaling for the comparison between the input pixel value to the overlay manager and the color key value.

  The source transparency color key mode is selected by setting the following bit to 0x1:

  – DISPC_OVR_CONFIG[11] TCKLCDSELECTION

  Figure 8-50 shows an example of source color key. The pixels with the transparency color key are not displayed; instead, pixels of the resulting layer underneath are shown.

Figure 8-50. DISPC Overlay Source Transparency Color Key Example



- **Destination transparency color key**

  The graphics destination transparency color key values define the range of the encoded pixels in layer 1, which are not displayed. Other layer 1 pixels (not in the range of destination transparency color key value) are displayed over layer 2. The encoded pixels within the range of destination color key values are pixels not visible on the screen because pixels at the same position in layer 2 are visible; otherwise, encoded pixels are visible above layer 2 . The destination transparency color key applies only if layer 1 overlaps layer 2 (see the Z-order section for details on layer position depending on the Z-order parameter in Section 8.2.4.12.1, *DISPC Overlay Priority Rule*); otherwise, the destination transparency color key is ignored.

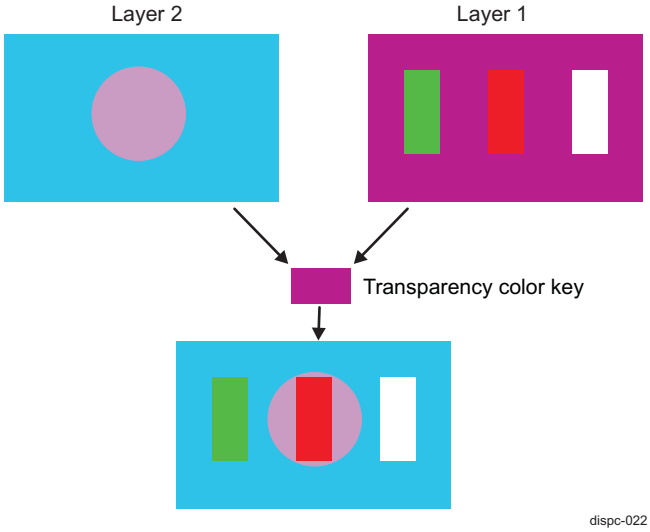
  The scaler can be enabled as a preprocessor in the VID pipelines. It is necessary, however, to consider the pixel scaling preprocessing in order to define the color key value to be used after rescaling for the comparison between the input pixel value to the overlay manager and the color key value.

  The destination transparency color key mode is selected by setting the following bit to 0x0:

  – DISPC_OVR_CONFIG[11] TCKLCDSELECTION

  Figure 8-51 shows an example of the destination color key. The pixels, equal to the transparency color key, are not displayed and are replaced by layer 2 pixels. All other layer 1 pixels, different from the transparency color key, are displayed over layer 2.

Figure 8-51. DISPC Overlay Destination Transparency Color Key Example



8.2.4.13 DISPC Video Port Output

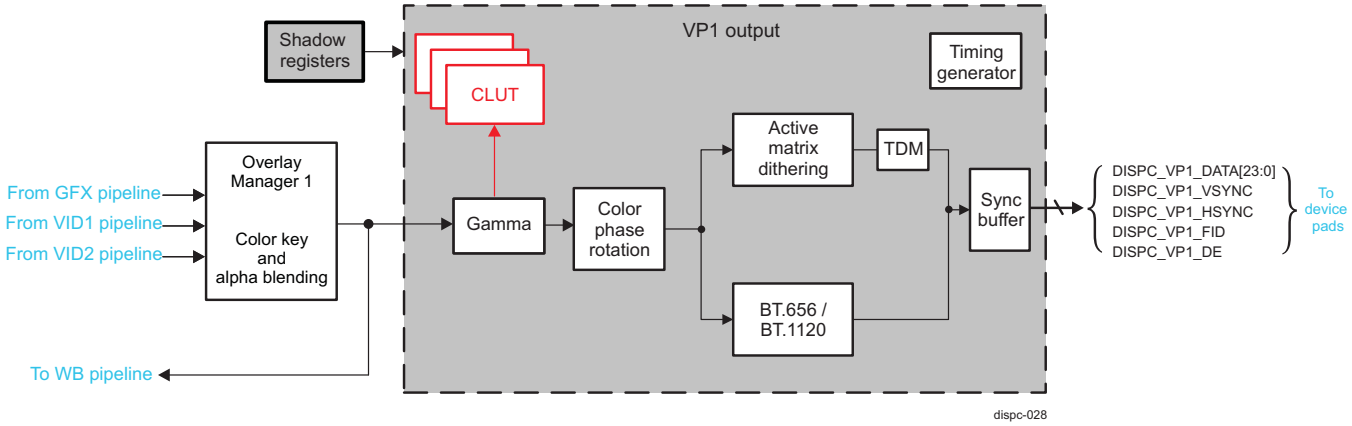
DISPC implemets a single video port output (VP1). It processes data received from Overlay Manager 1 (OVR1). The VP1 output path consist of several processing blocks (see Figure 8-52):

- Gamma correction unit
- Color phase rotation (CPR) (also used for RGB-to-YUV conversion)
- Active matrix dithering with TDM
- BT.656
- BT.1120
- Timing generator

The display subsystem supports active matrix display technology. The configuration of colors depends on the color depth:

- 24 bpp supports 16,777,216 colors.
- 18 bpp supports 262,144 colors.
- 16 bpp supports 65,536 colors.
- 12 bpp supports 4096 colors.

Figure 8-52. DISPC VP1 Output Architecture



NOTE: In BT.656 mode only bits 9 through 0 are used from DISPC_VP1_DATA[23:0] data bus.

8.2.4.13.1 DISPC VP1 Gamma Correction Unit

When performing gamma correction, the selected encoded pixel values based on the color keys by the overlay manager from the video or graphics paths are sent to the gamma curve table. Each component of encoded pixel value is used as a pointer to index 1 out of 256 24-bit gamma curve entries in the table. Each 8-bit component is replaced with the 8-bit table value corresponding to R, G, or B component. The table is loaded by software. It is possible to load only part of the table. For each access to the table, the 24-bit value is associated with index in the table by concatenating the 24-bit value (LSB of 32-bit access) and the 8-bit index value (MSB of the 32-bit access).

The sequence to load the gamma table is:

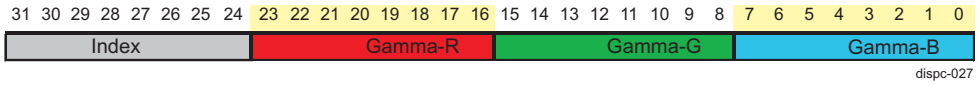
1. SW writes (only writes are supported) 32-bit gamma correction values using single access, or burst access in streaming mode, into DISPC_VP1_GAMMA_TABLE register. The LSB 24 bits [23:0] are used for the value, and the MSB 8 bits [31:24] are used for the index into the table.
2. Loop to Step 1, if there is a new access to the gamma table register. The SW can access other registers between two accesses to the gamma table register.

SW needs to ensure that there is no visible effect when modifying the table, since it is not under HW control.

The usage of the gamma table is activated by setting DISPC_VP1_CONFIG[2] GAMMAENABLE register bit to 0x1.

Figure 8-53 describes the format of one of the gamma curve values in the memory.

Figure 8-53. DISPC Data Memory Organization for Gamma Mode in VP1 Output



8.2.4.13.2 DISPC VP1 Color Phase Rotation Unit

The Color Phase Rotation (CPR) unit can be used to correct the display output colorimetry in case of nonpure white backlight.

The CPR is enabled by setting the DISPC_VP1_CONFIG[15] CPR bit to 0x1. The coefficients are programmed in the following registers:

- Red 10-bit signed coefficients in DISPC_VP1_CPR_COEF_R
- Green 10-bit signed coefficients in DISPC_VP1_CPR_COEF_G
- Blue 10-bit signed coefficients in DISPC_VP1_CPR_COEF_B

The CPR logic is integrated after the overlay manager while using the gamma correction and before the spatial/temporal dithering. The CPR can be selected to correct the nonpure white backlight of the display module by using a programmable matrix to convert the 36-bit RGB pixel value into a new 30-bit RGB pixel value. The matrix is programmed through a set of nine 10-bit signed coefficients. The output of the calculation is clipped to [0:4095]. The CPR is processed by the equation shown in Figure 8-54. Table 8-43 lists all coefficients with their respective register bitfields for settings.

Figure 8-54. DISPC VP1 CPR Matrix

$$\begin{bmatrix} R_{out} \\ G_{out} \\ B_{out} \end{bmatrix} = \begin{bmatrix} Rr & Rg & Rb \\ Gr & Gg & Gb \\ Br & Bg & Bb \end{bmatrix} * \begin{bmatrix} R_{in} \\ G_{in} \\ B_{in} \end{bmatrix}$$

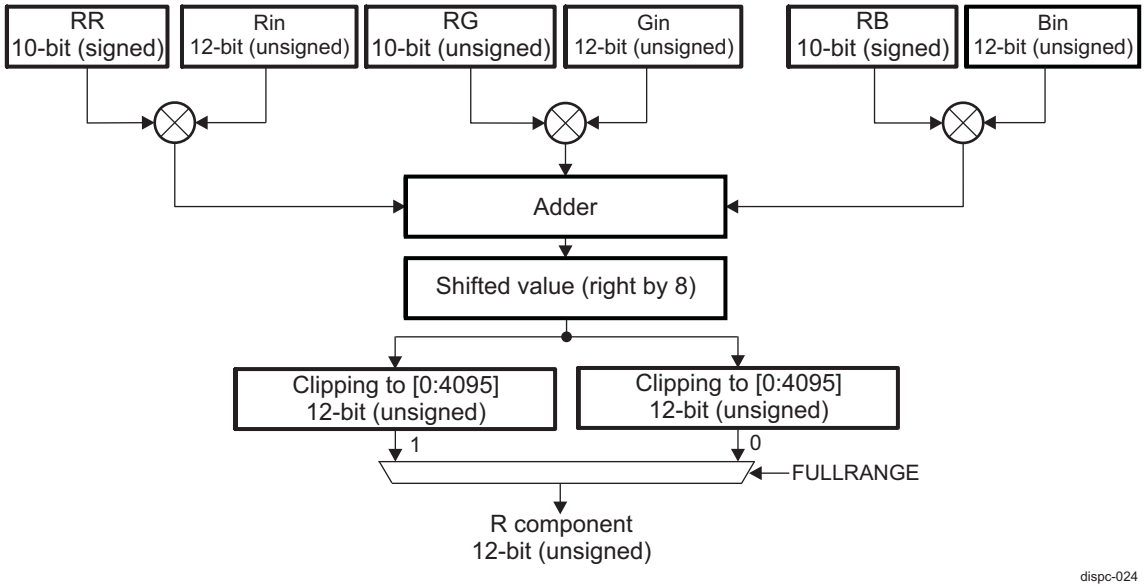
dispc-023

Table 8-43. DISPC VP1 CPR, or RGB to YUV Conversion Coefficients with Associated Register Bitfields

Registers	Bit Field	Color Phase Rotation	RGB to YUV
DISPC_VP1_CPR_COEF_R	RR	Rr	Yr
	RG	Rg	Yg
	RB	Rb	Yb
DISPC_VP1_CPR_COEF_G	GR	Gr	Cbr
	GG	Gg	Cbg
	GB	Gb	Cbb
DISPC_VP1_CPR_COEF_B	BR	Br	Crr
	BG	Bg	Crg
	BB	Bb	Crb

Figure 8-55 shows the CPR macro-architecture.

Figure 8-55. DISPC VP1 CPR Macro-Architecture



8.2.4.13.3 DISPC VP1 Color Space Conversion

The CPR block can be used to perform Color Space Conversion (CSC). Table 8-43 lists the associated coefficients with their respective register bit-fields. The CSC logic uses the CPR registers for the coefficients. Only the upper 10 bits of each color component are used for the CSC processing. The CSC processing is enabled by setting the DISPC_VP1_CONFIG[24] COLORCONVENABLE register bit to 0x1.

The color space conversion on the VP output can be selected in order to convert from 30-bit RGB to YUV444. The matrix is programmed through a set of nine 10-bit signed coefficients. The result is clipped to [64:940] for the luminance and [64:960] for the chrominance, when full-range equals zero (DISPC_VP1_CONFIG[25] FULLRANGE = 0).

Figure 8-56. DISPC VP1 CSC RGB to YUV Registers (FullRange=0)

$$\begin{bmatrix} Y \\ Cr \\ Cb \end{bmatrix} = \frac{1}{256} * \begin{bmatrix} YR & YG & YB \\ CrR & CrG & CrB \\ CbR & CbG & CbB \end{bmatrix} * \begin{bmatrix} R \\ G \\ B \end{bmatrix} + \begin{bmatrix} 0 \\ 128 \\ 128 \end{bmatrix}$$

dispc-098

If the programmed active range for the luminance samples (Y) and chrominance samples (Cb and Cr) is [0:1023], the values Y, Cb, and Cr are clipped to the range [0:1023]. The following equation gives the 11-bit coefficients of the RGB to YUV color space conversion, for full-range (DISPC_VP1_CONFIG[25] FULLRANGE = 1)

Figure 8-57. DISPC VP1 CSC RGB to YUV Registers (FullRange=1)

$$\begin{bmatrix} Y \\ Cr \\ Cb \end{bmatrix} = \frac{1}{256} * \begin{bmatrix} YR & YG & YB \\ CrR & CrG & CrB \\ CbR & CbG & CbB \end{bmatrix} * \begin{bmatrix} R \\ G \\ B \end{bmatrix} + \begin{bmatrix} 0 \\ 128 \\ 128 \end{bmatrix}$$

dispc-098

In order to provide YUV422 data to the BT-656 or BT-1120 blocks, the YUV444 format is converted to YUV422 by sub-sampling the chrominance values. The hardware does this by averaging two-by-two the chrominance samples. The conversion from YUV444 to YUV422 is performed when the color space conversion is enabled.

8.2.4.13.4 DISPC VP1 BT.656 and BT.1120 Modes

The VP1 output can be configured in BT.656 or BT.1120 mode. The following standards are not supported in BT.656 mode:

- BT.470 (Support for conventional Analog TV Systems)
- BT.803 (The avoidance of interference generated by digital television studio equipment)
- BT.1364 (Format of ancillary data signals carried in digital component studio interfaces)

Unsupported formats when BT.1120 mode is used:

- BT.1364 (Support for Ancillary Data during blanking period)

BT.656/BT.1120 modes use embedded EAV/SAV syncs.

Enabling BT.656 or BT.1120 format for VP1 ouput is done by setting DISPC_VP1_CONFIG[20] BT656ENABLE or [21] BT1120ENABLE register bits.

NOTE: It is not posssible to enable BT.656 and BT.1120 mode simultaneously on the same output.

Figure 8-58 shows signal mapping on DISPC_VP1_DATA[23:0] data bus. Bits 9 to 0 are used in BT.656 mode (10-bit).

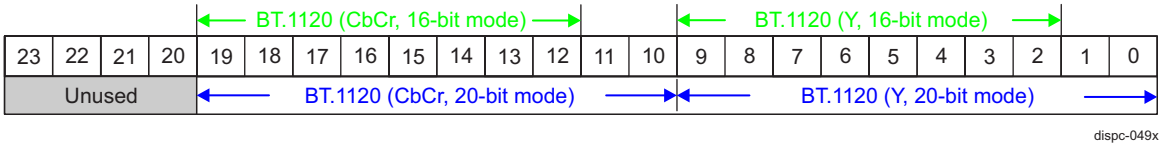
Figure 8-58. DISPC VP1 Data Mapping in BT.656 Mode

23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Unused														BT.656									

NOTE: For compatibility with existing 8-bit interfaces, the two LSB are ignored, and only bits [9:2] are used.

Figure 8-59 shows signal mapping on DISPC_VP1_DATA[23:0] data bus for BT.1120 mode. Bits [19:10] (CbCr) and [9:0] (Y) are used in 20-bit mode. Bits [19:12] (CbCr) and [9:2] (Y) are used in 16-bit mode (YCbCr 4:2:2).

Figure 8-59. DISPC VP1 Data Mapping in BT.1120 Mode



VP1 supports progressive output for embedded syncs interface, when it is configured in BT.656/BT.1120 modes. For more information on timings configuration, see Section 8.2.4.13.7, *DISPC VP1 Timing Generator and Panel Settings*.

8.2.4.13.4.1 DISPC BT Mode Blanking

During the transmission of the video signal, the portion of the stream in-between active video data segments is known as the horizontal blanking interval.

Strictly speaking this entire region is the blanking interval, but this interval also includes the EAV and SAV codes. The remaining bytes of information in a digital blanking interval are filled with values corresponding to the blanking levels of the Cb, Y and Cr signals respectively, and in accordance with the standard multiplex sequence for the stream (CbYCrY..). The blanking levels are as follows:

- Cb = 80h
- Y = 10h
- Cr = 80h.

  The sequence in the BLANKING region of the data stream is therefore: *80h, 10h, 80h, 10h.....80h, 10h*

  For more details on setting the blanking timing values for BT.1120 and/or BT.656 mode, see Section 8.2.4.13.7, *DISPC Timing Generator and Panel Settings*.

8.2.4.13.4.2 DISPC BT Mode EAV and SAV

The End of Active Video(EAV) and Start of Active Video (SAV) parts of the stream are timing codes. Their function can be summarized as follows:

- EAV – marks the end of the active video data within the current line and therefore also the start of the subsequent line.
- SAV – heralds the start of the active video data within the current line.

These codes are embedded within the BT.656 video data stream, thereby eliminating the need for additional timing signals (HSYNC, VSYNC) to be included as part of the interface.

Both EAV and SAV codes are comprised of a sequence of four bytes (FFh – 00h – 00h - XY). The first three bytes in the sequence constitute a fixed preamble. The fourth byte, contains information about the field being transmitted (Field 1 or Field 2 in an interlaced video signal), the state of field blanking (Vertical) and the state of line blanking (Horizontal). The bit assignment for this byte of the code is shown in Figure 8-60, with the function of each bit described inTable 8-44.

Figure 8-60. DISPC BT Mode Bit-Assignment for the Fourth Byte of EAV/SAV Codes

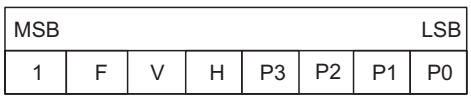


Table 8-44. DISPC BT Mode Bit Function

Bit	Symbol	Function
7	1	Always set to '1'.
6	F	Field bit. 0 – Field 1 1 – Field 2
5	V	Vertical Blanking Status bit. This bit goes High during a vertical field blanking interval, otherwise it remains Low.

Table 8-44. DISPC BT Mode Bit Function (continued)

Bit	Symbol	Function
4	H	Horizontal Blanking Status bit. 0 – byte is part of SAV code (i.e. stream is entering an active video data region for the current line) 1 – byte is part of EAV code (i.e. stream has entered a horizontal blanking interval - start of a new line)
3	P3	Protection bit 3
2	P2	Protection bit 2
1	P1	Protection bit 1
0	P0	Protection bit 0

The protection bits allow for detection and correction of 1-bit errors and the detection of 2-bit errors. The status of P3, P2, P1 and P0 depend on the states of bits F, V and H. This dependency is shown in Table 8-45.

Table 8-45. DISPC BT Mode Status of Protection Bits as F, V and H Vary

F	V	H	P3	P2	P1	P0
0	0	0	0	0	0	0
0	0	1	1	1	0	1
0	1	0	1	0	1	1
0	1	1	0	1	1	0
1	0	0	0	1	1	1
1	0	1	1	0	1	0
1	1	0	1	1	0	0
1	1	1	0	0	0	1

8.2.4.13.5 DISPC VP1 Spatial/Temporal Dithering

The spatial/temporal dithering logic can be selected to enhance the quality of the active matrix outputs. The dithering logic is integrated after the CPR and before the TDM. The encoded pixel values are used by spatial/temporal dithering logic to display the data in a lower color depth on the display panel. The spatial/temporal dithering algorithm is based on the (x,y) pixel position and frame rate control. The dithering logic can process the pixels over one frame, two frames, or four frames. The number of frames is selected by setting the DISPC_VP1_CONTROL[31:30] SPATIALTEMPORALDITHERINGFRAMES bit field. In the case of a single frame, only spatial processing is applied. In case of multiple frames, spatial and temporal processing are applied to the pixels. The spatial/temporal dithering logic is enabled by setting the DISPC_VP1_CONTROL[7] STDITHERENABLE bit to 0x1.

NOTE:

- If the interface data bus is smaller than the pixel format size and spatial/temporal dithering is not enabled, the MSBs of the pixel color components are output on the interface data bus.
- If the interface data bus is larger than the pixel format size, by programming the pixel components replication active/inactive, the MSB is replicated to the LSB of the interface data bus.

8.2.4.13.6 DISPC VP1 Multiple Cycle Output Format (TDM)

The pixels are formatted on one or multiple cycles (a maximum of three cycles). On three cycles, two pixels can concatenate and send to the panel. The cycle format is selected through the DISPC_VP1_CONTROL[24:23] TDMCYCLEFORMAT bit field. The number of bits for each cycle is set in the DISPC_VP1_DATA_CYCLE_0 register for the first cycle, the DISPC_VP1_DATA_CYCLE1 register for the second cycle, and the DISPC_VP1_DATA_CYCLE2 register for the third cycle. The interface data bus width, when TDM mode is enabled (DISPC_VP1_CONTROL[20] TDMENABLE = 1), can be 8, 9, 12, or 16 bits, configurable through the DISPC_VP1_CONTROL[22:21] TDMPARALLELMODE bit field.

When the TDM is disabled (DISPC_VP1_CONTROL[20] TDMENABLE = 0), the DISPC outputs the pixels using the conventional formats: active matrix display monochrome/color. When TDM is disabled, the interface data bus width is configured through the DISPC_VP1_CONTROL[10:8] DATALINES bit field.

When using TDM mode, only up to 24-bit per pixel can be output on the interface. For higher color depth, only the upper bits are kept before converting each pixel into TDM output.

Figure 8-61 through Figure 8-64 show various examples of TDM settings in the function of pixel data formats and the interface data bus width.

Figure 8-61. DISPC VP1 TDM 8-Bit Interface Settings

	24-bpp		
	1st cycle	2nd cycle	3rd cycle
Data[7]	R0[7]	G0[7]	B0[7]
Data[6]	R0[6]	G0[6]	B0[6]
Data[5]	R0[5]	G0[5]	B0[5]
Data[4]	R0[4]	G0[4]	B0[4]
Data[3]	R0[3]	G0[3]	B0[3]
Data[2]	R0[2]	G0[2]	B0[2]
Data[1]	R0[1]	G0[1]	B0[1]
Data[0]	R0[0]	G0[0]	B0[0]

DISPC_VP1_CONTROL.TDMCYCLEFORMAT = 0x2
DISPC_VP1_DATA_CYCLE0 = 0x00000008
DISPC_VP1_DATA_CYCLE1 = 0x00000008
DISPC_VP1_DATA_CYCLE2 = 0x00000008

	18-bpp		
	1st cycle	2nd cycle	3rd cycle
Data[7]	R0[5]	G0[3]	x
Data[6]	R0[4]	G0[2]	x
Data[5]	R0[3]	G0[1]	x
Data[4]	R0[2]	G0[0]	x
Data[3]	R0[1]	B0[5]	x
Data[2]	R0[0]	B0[4]	x
Data[1]	G0[5]	B0[3]	B0[1]
Data[0]	G0[4]	B0[2]	B0[0]

DISPC_VP1_CONTROL.TDMCycleFormat = 0x2
DISPC_VP1_DATA_CYCLE0 = 0x00000008
DISPC_VP1_DATA_CYCLE1 = 0x00000008
DISPC_VP1_DATA_CYCLE2 = 0x00000002

	16-bpp	
	1st cycle	2nd cycle
Data[7]	R0[4]	G0[2]
Data[6]	R0[3]	G0[1]
Data[5]	R0[2]	G0[0]
Data[4]	R0[1]	B0[4]
Data[3]	R0[0]	B0[3]
Data[2]	G0[5]	B0[2]
Data[1]	G0[4]	B0[1]
Data[0]	G0[3]	B0[0]

DISPC_VP1_CONTROL.TDMCYCLEFORMAT = 0x1
DISPC_VP1_DATA_CYCLE0 = 0x00000008
DISPC_VP1_DATA_CYCLE1 = 0x00000008

	12-bpp	
	1st cycle	2nd cycle
Data[7]	R0[3]	x
Data[6]	R0[2]	x
Data[5]	R0[1]	x
Data[4]	R0[0]	x
Data[3]	G0[3]	B0[3]
Data[2]	G0[2]	B0[2]
Data[1]	G0[1]	B0[1]
Data[0]	G0[0]	B0[0]

DISPC_VP1_CONTROL.TDMCYCLEFORMAT = 0x1
DISPC_VP1_DATA_CYCLE0 = 0x00000008
DISPC_VP1_DATA_CYCLE1 = 0x00000004

dispc-057

Figure 8-62. DISPC VP1 TDM 9-Bit Interface Settings

	24-bpp		
	1st cycle	2nd cycle	3rd cycle
Data[8]	R0[7]	G0[6]	x
Data[7]	R0[6]	G0[5]	x
Data[6]	R0[5]	G0[4]	x
Data[5]	R0[4]	G0[3]	B0[5]
Data[4]	R0[3]	G0[2]	B0[4]
Data[3]	R0[2]	G0[1]	B0[3]
Data[2]	R0[1]	G0[0]	B0[2]
Data[1]	R0[0]	B0[7]	B0[1]
Data[0]	G0[7]	B0[6]	B0[0]

DISPC_VP1_CONTROL.TDMCycleFormat = 0x2
DISPC_VP1_DATA_CYCLE0 = 0x00000009
DISPC_VP1_DATA_CYCLE1 = 0x00000009
DISPC_VP1_DATA_CYCLE2 = 0x00000006

	18-bpp	
	1st cycle	2nd cycle
Data[8]	R0[5]	G0[2]
Data[7]	R0[4]	G0[1]
Data[6]	R0[3]	G0[0]
Data[5]	R0[2]	B0[5]
Data[4]	R0[1]	B0[4]
Data[3]	R0[0]	B0[3]
Data[2]	G0[5]	B0[2]
Data[1]	G0[4]	B0[1]
Data[0]	G0[3]	B0[0]

DISPC_VP1_CONTROL.TDMCYCLEFORMAT = 0x1
DISPC_VP1_DATA_CYCLE0 = 0x00000009
DISPC_VP1_DATA_CYCLE1 = 0x00000009

	16-bpp	
	1st cycle	2nd cycle
Data[8]	R0[4]	x
Data[7]	R0[3]	x
Data[6]	R0[2]	G0[1]
Data[5]	R0[1]	G0[0]
Data[4]	R0[0]	B0[4]
Data[3]	G0[5]	B0[3]
Data[2]	G0[4]	B0[2]
Data[1]	G0[3]	B0[1]
Data[0]	G0[2]	B0[0]

DISPC_VP1_CONTROL.TDMCYCLEFORMAT = 0x1
DISPC_VP1_DATA_CYCLE0 = 0x00000009
DISPC_VP1_DATA_CYCLE1 = 0x00000007

	12-bpp	
	1st cycle	2nd cycle
Data[8]	R0[3]	x
Data[7]	R0[2]	x
Data[6]	R0[1]	x
Data[5]	R0[0]	x
Data[4]	G0[3]	x
Data[3]	G0[2]	x
Data[2]	G0[1]	B0[2]
Data[1]	G0[0]	B0[1]
Data[0]	B0[3]	B0[0]

DISPC_VP1_CONTROL.TDMCYCLEFORMAT = 0x1
DISPC_VP1_DATA_CYCLE0 = 0x00000009
DISPC_VP1_DATA_CYCLE1 = 0x00000003

dispc-058

Figure 8-63. DISPC VP1 TDM 12-Bit Interface Settings

	24-bpp	
	1st cycle	2nd cycle
Data[11]	R0[7]	G0[3]
Data[10]	R0[6]	G0[2]
Data[9]	R0[5]	G0[1]
Data[8]	R0[4]	G0[0]
Data[7]	R0[3]	B0[7]
Data[6]	R0[2]	B0[6]
Data[5]	R0[1]	B0[5]
Data[4]	R0[0]	B0[4]
Data[3]	G0[7]	B0[3]
Data[2]	G0[6]	B0[2]
Data[1]	G0[5]	B0[1]
Data[0]	G0[4]	B0[0]

DISPC_VP1_CONTROL.TDMCYCLEFORMAT = 0x1
DISPC_VP1_DATA_CYCLE0 = 0x0000000C
DISPC_VP1_DATA_CYCLE1 = 0x0000000C

	18-bpp		
	1st cycle	2nd cycle	3rd cycle
Data[11]	R0[5]	B0[5]	G1[5]
Data[10]	R0[4]	B0[4]	G1[4]
Data[9]	R0[3]	B0[3]	G1[3]
Data[8]	R0[2]	B0[2]	G1[2]
Data[7]	R0[1]	B0[1]	G1[1]
Data[6]	R0[0]	B0[0]	G1[0]
Data[5]	G0[5]	R1[5]	B1[5]
Data[4]	G0[4]	R1[4]	B1[4]
Data[3]	G0[3]	R1[3]	B1[3]
Data[2]	G0[2]	R1[2]	B1[2]
Data[1]	G0[1]	R1[1]	B1[1]
Data[0]	G0[0]	R1[0]	B1[0]

DISPC_VP1_CONTROL.TDMCYCLEFORMAT = 0x3
DISPC_VP1_DATA_CYCLE0 = 0x0000000C
DISPC_VP1_DATA_CYCLE1 = 0x00060606
DISPC_VP1_DATA_CYCLE2 = 0x000C0000

	16-bpp	
	1st cycle	2nd cycle
Data[11]	R0[4]	x
Data[10]	R0[3]	x
Data[9]	R0[2]	x
Data[8]	R0[1]	x
Data[7]	R0[0]	x
Data[6]	G0[5]	x
Data[5]	G0[4]	x
Data[4]	G0[3]	x
Data[3]	G0[2]	B0[3]
Data[2]	G0[1]	B0[2]
Data[1]	G0[0]	B0[1]
Data[0]	B0[4]	B0[0]

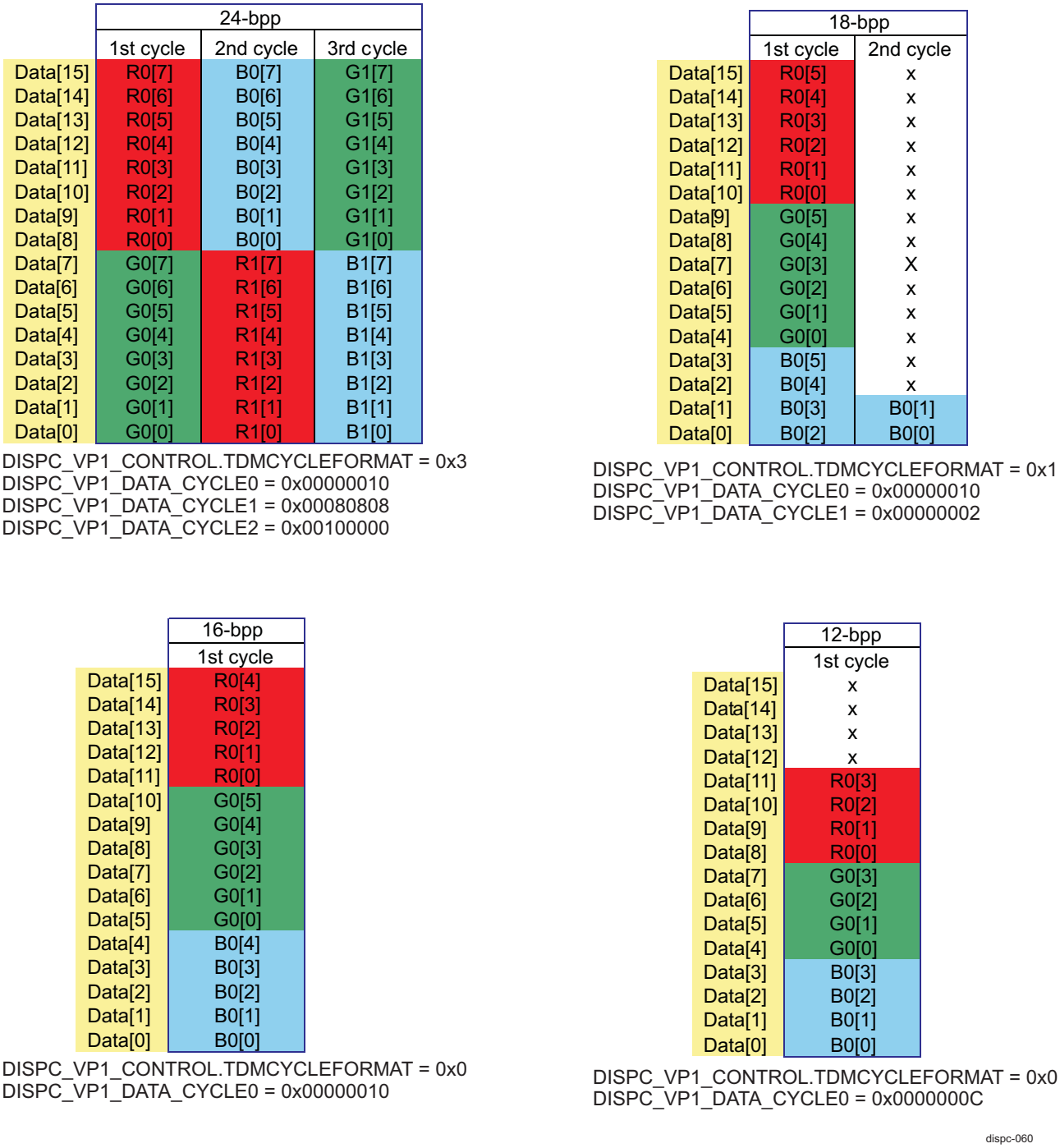
DISPC_VP1_CONTROL.TDMCYCLEFORMAT = 0x1
DISPC_VP1_DATA_CYCLE0 = 0x0000000C
DISPC_VP1_DATA_CYCLE1 = 0x00000004

	12-bpp
	1st cycle
Data[11]	R0[3]
Data[10]	R0[2]
Data[9]	R0[1]
Data[8]	R0[0]
Data[7]	G0[3]
Data[6]	G0[2]
Data[5]	G0[1]
Data[4]	G0[0]
Data[3]	B0[3]
Data[2]	B0[2]
Data[1]	B0[1]
Data[0]	B0[0]

DISPC_VP1_CONTROL.TDMCYCLEFORMAT = 0x0
DISPC_VP1_DATA_CYCLE0 = 0x0000000C

dispc-059

Figure 8-64. DISPC VP1 TDM 16-Bit Interface Settings



8.2.4.13.7 DISPC VP1 Timing Generator and Panel Settings

The VP1 output has a dedicated timing generator supporting progressive and interlaced mode.

The size of the display panel is defined by:

- Number of lines, DISPC_VP1_SIZE_SCREEN[27:16] LPP bit field, with a value from 1 to 4096
- Number of pixels per line, DISPC_VP1_SIZE_SCREEN[11:0] PPL bit field, with a value from 1 to 4096

Standard HSYNC/VSYNC timing generation are programmable for VP1:

- Horizontal front porch is set in the DISPC_VP1_TIMING_H[19:8] HFP bit field.
- Horizontal back porch is set in the DISPC_VP1_TIMING_H[31:20] HBP bit field.
- Horizontal synchronization pulse width is set in the DISPC_VP1_TIMING_H[7:0] HSW bit field.
- Vertical front porch is set in theDISPC_VP1_TIMING_V[19:8] VFP bit field.
- Vertical back porch is set in the DISPC_VP1_TIMING_V[31:20] VBP bit field.
- Vertical synchronization pulse width is set in the DISPC_VP1_TIMING_V[7:0] VSW bit field.

When the output is in BT.1120 or BT.656 mode, the following timing constants are mapped onto the DISPC_VP1_TIMING_H and DISPC_VP1_TIMING_V registers:

- Progressive mode:
  - Horizontal blanking (12 bits) is set in the {DISPC_VP1_TIMING_V[3:0] VSW, DISPC_VP1_TIMING_H[7:0] HSW} bit fields; (up to 2048 bytes of horizontal blanking supported).
  - Vertical frame blanking No 1 is set in the DISPC_VP1_TIMING_V[19:8] VFP bit field.
  - Vertical frame blanking No 2 is set in the DISPC_VP1_TIMING_V[31:20] VBP bit field.
  - Number of lines is set in the DISPC_VP1_SIZE_SCREEN[27:16] LPP bit field
  - Number of pixels per line is set in the DISPC_VP1_SIZE_SCREEN[11:0] PPL bit field.
- Interlaced mode:
  - Horizontal blanking (12 bits) is set in the {DISPC_VP1_TIMING_V[3:0] VSW, DISPC_VP1_TIMING_H[7:0] HSW} bit fields; (up to 2048 bytes of horizontal blanking supported).
  - Vertical field blanking No 1 for Even Field is set in the DISPC_VP1_TIMING_H[19:8] HFP bit field.
  - Vertical field blanking No 2 for Even Field is set in the DISPC_VP1_TIMING_H[31:20] HBP bit field.
  - Vertical field blanking No 1 for Odd Field is set in the DISPC_VP1_TIMING_V[19:8] VFP bit field.
  - Vertical field blanking No 2 for Odd Field is set in the DISPC_VP1_TIMING_V[31:20] VBP bit field.
  - Number of lines per field (even) is set in the DISPC_VP1_SIZE_SCREEN[27:16] LPP bit field.
  - Delta number of odd field compared to even field (in a single line) is set in the DISPC_VP1_SIZE_SCREEN[15:14] DELTA_LPP bit field.
  - Number of pixels per line is set in the DISPC_VP1_SIZE_SCREEN[11:0] PPL bit field.

NOTE: The DELTA_LPP field only controls the output channel and not the size of the field fetched from the frame buffer in memory. This field shall be set to zero for YUV420 format.

Horizontal/vertical synchronization and output enable signals polarity are programmable by setting the DISPC_VP1_POL_FREQ[12] IVS, DISPC_VP1_POL_FREQ[13] IHS, and DISPC_VP1_POL_FREQ[15] IEO bits. These signals can be gated by setting the DISPC_VP1_CONFIG[7] VSYNCGATED and DISPC_VP1_CONFIG[6] HSYNCGATED bits.

The latch of data can be driven on the rising or falling edge of the pixel clock by setting the IPC bit of DISPC_VP1_POL_FREQ[14] IPC and CTRL_CORE_SMA_SW_1[] DSS_CH0_IPC (the values must mach) registers. The drive of the SYNC and VSYNC signals in the function of the pixel clock is done by setting the DISPC_VP1_POL_FREQ[16] RF and CTRL_CORE_SMA_SW_1[] DSS_CH0_RF bit.

The timing generation of FID signal is defined by the DISPC_VP1_CONFIG[22] OUTPUTMODEENABLE bitfield. The selection can be changed only if the VP output is disabled.

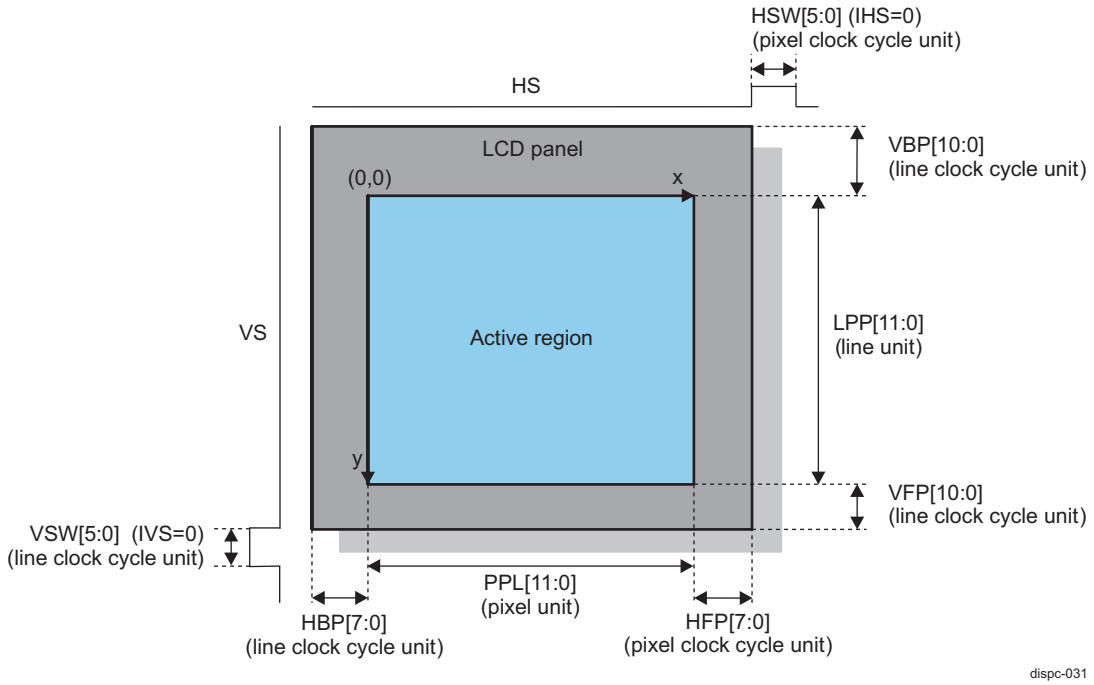
- 0x0 (default): Progressive mode (FID signal is inactive: low level)
- 0x1: Interlaced mode (FID signal transition from low to high and from high to low to indicate on the VP interface the field polarity and internally the FID to be used for the registers with dual value: one for odd field and another one for even field). The FID signal toggles on the rising edge of the VSW pulse.

When in interlaced mode, the DISPC_VP1_CONFIG[23] FIDFIRST bit indicates which field is output first:

- 0x0: Even field first (FID = 0)
- 0x1: Odd field first (FID = 1)

Figure 8-65 shows the timing values description.

Figure 8-65. DISPC VP1 Timing Values (Display Screen)

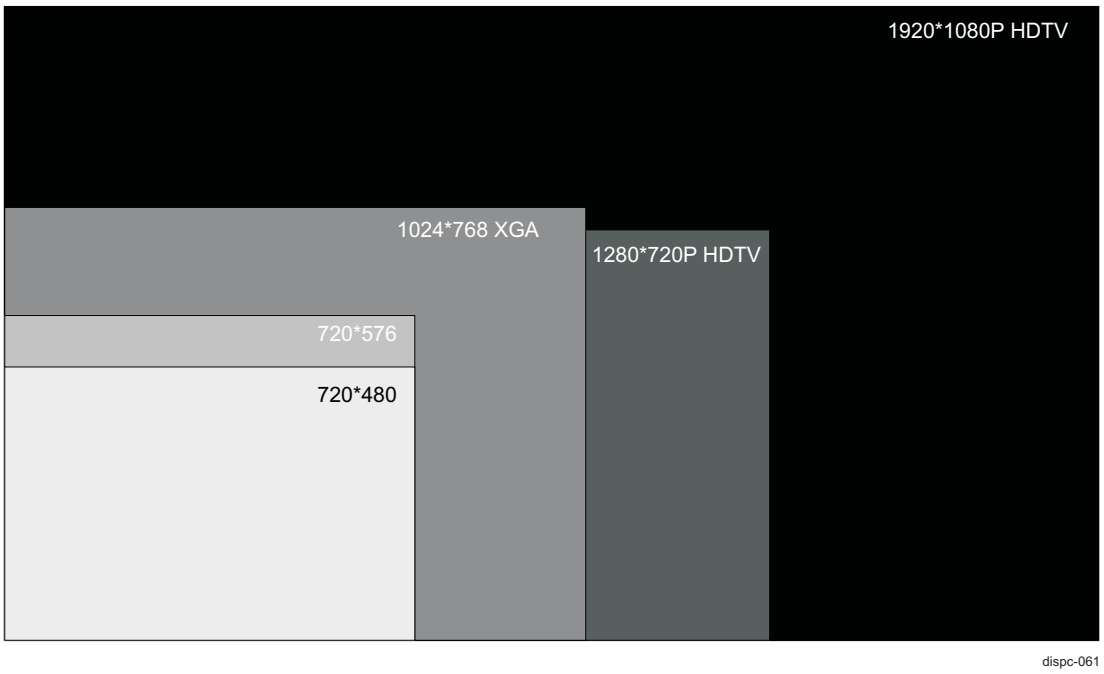


The pixel clock can be gated by setting the DISPC_VP1_CONFIG[5] PIXELCLOCKGATED bit to 0x1.

8.2.4.13.8 DISPC VP1 Configuration for TV Support

Figure 8-66 shows the TV formats supported.

Figure 8-66. DISPC Example TV Timing Formats



The size of a frame (field, if interlaced mode) is defined by the fields:

- Number of lines, DISPC_VP1_SIZE_SCREEN[27:16] LPP bit field, with a value from 1 to 4096

- Number of pixels per line, DISPC_VP1_SIZE_SCREEN[11:0] PPL bit field, with a value from 1 to 4096
- Delta size between odd/even field, DISPC_VP1_SIZE_SCREEN[15:14] DELTA_LPP bit field. This bit field controls only the output channel and not the size of the data field fetched from the frame buffer in memory.

The hold time of the pixels on the data bus is determined in clock cycles by the DISPC_VP1_CONTROL[16:14] HT bit field. The default value at reset time is 0x0 (one cycle).

- When connected to external HDMI encoder, Table 8-46 indicates the DISPC_VP1_SIZE_SCREEN register values for PPL and LPP for each HD standard.

Table 8-46. DISPC VP1 PPL and LLP Value for HD Standard

Standards		Active Pixels/Line	Active Lines	Digital Clock DISPC_VP1_CONTROL[16:14] HT = 0x0	DISPC_VP1_SIZE_SCREEN value
HDTV	720p	1280	720	74.25/74.125 MHz (60/59.99..frames/s)	0x02CF 04FF
	1080i	1920	540	74.25/74.125 MHz (60/59.99..frames/s)	0x021B 077F
	1080p	1920	1080	148.5/148.25 MHz (60/59.99..frames/s)	0x0437 077F
	720p 3D (frame packing)	1280	1470	148.5/148.35/148.5 MHz (60/59.94/50 frames/s)	0x05BD 04FF
	1080p 3D (frame packing)	1920	2205	148.5/148.35 MHz (24/23.98 frames/s)	0x089C 077F
	1080p 3D (side-by-side half)	1920	1080	148.5 MHz (60 frames/s)	0x0437 077F

NOTE: When configuring the DISPC for outputting any supported 3D frame-packing format, the following generic details must be considered:

- As defined by the *HDMI v1.4 Specification*, 3D frame-packing is a video format structure composed of two stereoscopic pictures: left and right.
- The stereoscopic pictures can be processed through the video pipelines (VID1 and VID2). For more information about the configuration of video pipelines, see Section 8.2.4.9, *DISPC Video Pipelines*.
- The 3D frame can be generated by setting the VP1 overlay manager to combine the outputs of the selected video pipelines that hold the pictures. VID1 pipeline must carry the top field of the 3D frame (left stereoscopic picture), and VID2 pipeline must always carry the bottom field of the frame (right stereoscopic picture). The top and bottom fields are separated by an active space area.
- The pipeline carrying each field (top and bottom) of the 3D frame must have its height and width parameters defined in the [27:16] SIZEY and [11:0] SIZEX bit fields of the DISPC_VP1_SIZE_SCREEN register, and its Y and X positions defined in the [26:16] POSY and [10:0] POSX bit fields of the DISPC_VID_POSITION register. The active space area of the 3D frame can be encoded by setting the solid background color for the VP1 output (in the DISPC_OVR_DEFAULT_COLOR and DISPC_OVR_DEFAULT_COLOR2 registers). For more information about the overlay mechanism, see Section 8.2.4.12, *DISPC Overlay Managers*.

3D frame-packing format support using a single pipeline can be achieved using the DISPC region-based feature. Refer to Section 8.2.4.11, *DISPC Region-Based Mechanism*, for more information.

8.2.4.14 DISPC Extended 3D Support

8.2.4.14.1 DISPC Extended 3D Support - Line Alternative Format

The DISPC supports Line Alternative Format for 3D displays, as defined by the HDMI Specification 1.4a.

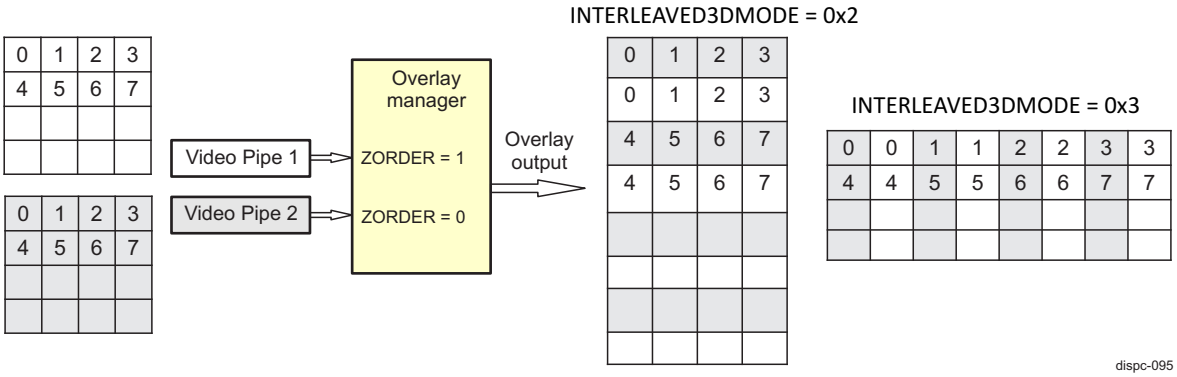
The processing is done in the overlay manager:

- The layers are grouped into two categories based on the z-order. Odd z-order may blend together and will interleave with even z-order pipes, which may blend together.
- Two types of interleaves are possible, line-based interleaving and pixel-based interleaving, to support landscape and portrait screen orientations.

The configuration is done through the DISPC_OVR_CONFIG[9:8] INTERLEAVED3DMODE register bitfield, which specifies the interleave pattern that needs to be added for each overlay manager:

- 0x0: No interleaving happens in the overlay manager
- 0x1: Reserved for future use
- 0x2: All even lines (all pixels) have a contribution from even z-order pipelines. All odd lines (all pixels) have a contribution from the odd z-order pipelines
- 0x3: All even pixels (for all lines) have a contribution from even z-order pipelines. All odd pixels (for all lines) have a contribution from the odd z-order pipelines

Figure 8-67. DISPC Illustration of 3D Interleaving

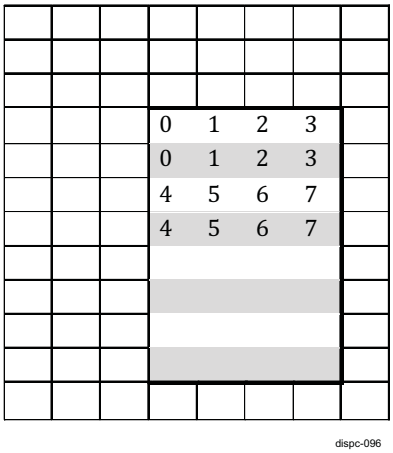


In addition, from Figure 8-67, if DISPC_OVR_CONFIG[9:8] INTERLEAVED3DMODE is programmed as 0x2, in this case the entire screen composited by the overlay manager (defined by PPL x LPP) gets sub-divided into odd and even lines. It is possible for a 3D window created using the two pipelines to have a non-zero position (that is, POSX and POSY not equal to zero).

In such cases, depending on the oddness or evenness of the POSY setting, the first line in the window may come from VID1 or VID2 pipeline.

For instance, if the POSX = 3 and POSY = 3 (set in DISPC_VID_POSITION registers), then the final composited screen with PPL = 8 and LPP = 12 (set in DISPC_VP1_SIZE_SCREEN register) may look like in Figure 8-68:

Figure 8-68. DISPC Illustration of a Non-zero Position of 3D Window



NOTE: In this case the first line comes from the odd-zorder pipeline, that is, the VID1 pipeline.

The following limitations apply to the 3D Line alternative format:

- Software must ensure that the left/right frames are of same size.
- Software must ensure that in case of line interleaving the DISPC_VID_SIZE[27:16] SIZEY bitfield values is not greater than LPP/2.
- Software must ensure that in case of pixel interleaving the DISPC_VID_SIZE[11:0] SIZEX is not greater than PPL/2.
- Software must program POSX and POSY for both left and right frame to be the same.
- Color-keying is not available for 3D formats.
- Since each overlay manager can support 4k x 4k frames, the maximum resolution of each left/right frame must be restricted to 2k x 2k.

8.2.4.14.2 DISPC Extended 3D Support - Frame Packing Format

3D Frame Packing Format support using a single pipeline can be achieved using the region-based feature. Refer to Section 8.2.4.11, *DISPC Region-Based Mechanism*, for more information.

8.2.4.15 DISPC Shadow Registers

Some DISPC registers are termed *shadow registers*. A shadow register change has no direct effect on the configuration of the DISPC. The registers are shadow registers in order the software to change the values of the registers at any time. When all the values for a given configuration are written into the registers, software must set 1 bit only to validate the configuration. When hardware reaches the end of the current frame and sees that the bit field has been set by software, the new configuration is now the configuration used by the hardware.

The bits enabling the hardware to use the new configuration are:

- DISPC_VP1_CONTROL[5] GOBIT bit for all the shadow registers of the pipeliens connected to the VP1 output

NOTE: As a general rule, all shadow registers are updated with the value of their shadow when:

- The interface is enabled, or
- The GOBIT bit-field of the output, which the register is associated to, is set and the output sync is active

8.2.5 DISPC Register Manual

8.2.5.1 DISPC Instance Summary

Table 8-47. DISPC Instance Summary

Module Name	L3_MAIN Base Address	Size
DISPC_COMMON	0x5801 0000	68 Bytes
DISPC_GFX1	0x5801 1000	64 Bytes
DISPC_WB	0x5801 5000	540 Bytes
DISPC_VID1	0x5801 7000	552 Bytes
DISPC_VID2	0x5801 8000	552 Bytes
DISPC_OVR1	0x5801 A800	32 Bytes
DISPC_OVR2	0x5801 A900	32 Bytes
DISPC_VP1	0x5801 AC00	92 Bytes

8.2.5.2 DISPC_COMMON Registers

8.2.5.2.1 DISPC_COMMON Register Summary

Table 8-48. DISPC_COMMON Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	DISPC_COMMON L3_MAIN Physical Address
DISPC_REVISION	R	32	0x0000 0000	0x5801 0000
DISPC_SYSCONFIG	RW	32	0x0000 0004	0x5801 0004
DISPC_SYSSTATUS	R	32	0x0000 0008	0x5801 0008
DISPC_IRQ_EOI	RW	32	0x0000 0020	0x5801 0020
DISPC_IRQSTATUS_RAW	RW	32	0x0000 0024	0x5801 0024
DISPC_IRQSTATUS	RW	32	0x0000 0028	0x5801 0028
DISPC_IRQENABLE_SET	RW	32	0x0000 002C	0x5801 002C
DISPC_IRQENABLE_CLR	RW	32	0x0000 0030	0x5801 0030
DISPC_IRQWAKEEN	RW	32	0x0000 0034	0x5801 0034
DISPC_GLOBAL_MFLAG_ATTRIBUTE	RW	32	0x0000 0040	0x5801 0040
DISPC_GLOBAL_BUFFER	RW	32	0x0000 0044	0x5801 0044
DISPC_BA0_FLIPIMMEDIATE_EN	R	32	0x0000 0048	0x5801 0048
DISPC_DBG_CONTROL	RW	32	0x0000 004C	0x5801 004C
DISPC_DBG_STATUS	R	32	0x0000 0050	0x5801 0050
DISPC_CLKGATING_DISABLE	RW	32	0x0000 0054	0x5801 0054

8.2.5.2.2 DISPC_COMMON Register Description

Table 8-49. DISPC_REVISION

Address Offset	0x0000 0000		
Physical Address	0x5801 0000	**Instance**	DISPC_COMMON
Description	This register contains the IP revision code		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								REV							

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Write 0's for future compatibility. Reads return 0	R	0x0
7:0	REV	IP revision	R	See (1)

(1) TI internal data

Table 8-50. Register Call Summary for Register DISPC_REVISION

Display Controller
- DISPC_COMMON Register Summary: [0]

Table 8-51. DISPC_SYSCONFIG

Address Offset	0x0000 0004		
Physical Address	0x5801 0004	**Instance**	DISPC_COMMON
Description	This register allows to control various parameters of the OCP interface.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																		MIDLEMODE		RESERVED		CLOCKACTIVITY		RESERVED		WARMRESET		SIDLEMODE		ENWAKEUP		SOFTRESET		AUTOIDLE

Bits	Field Name	Description	Type	Reset
31:14	RESERVED	Write 0's for future compatibility. Reads returns 0	R	0x0
13:12	MIDLEMODE	Master interface power management, standby/wait control 0x0: Force-standby.MStandby is only asserted when the module is disabled. MStandby is only asserted when the module is disabled. 0x1: No-Standby:MStandby is never asserted 0x3: Reserved 0x2: Smart-Standby.MStandby is asserted based on the internal activity of the module	RW	0x2
11:10	RESERVED	Write 0's for future compatibility. Reads returns 0	R	0x0

Bits	Field Name	Description	Type	Reset
9:8	CLOCKACTIVITY	Clocks activity during wake up mode period 0x0: OCP and Functional clocks can be switched off 0x1: Functional clocks can be switched off and OCP clocks are mantained during wake up period 0x3: OCP and Functional clocks are maintained during wake up period 0x2: OCP clocks can be switched off and Functional clocks are maintained during wake up period	RW	0x0
7:6	RESERVED	Write 0's for future compatibility. Reads returns 0	R	0x0
5	WARMRESET	Warm reset. Set this bit to 1 triggers a module warm reset. The bit is automatically reset by the hardware. During reads, it always returns 0. The warm reset keep the configuration registers unchanged. 0x0: Normal mode 0x1: the warmreset is set	RW	0x0
4:3	SIDLEMODE	Slave interface power management, Idle req/ack control 0x0: Force-idle.An idle request is acknowledged unconditionally 0x1: No-idle. An idle request is never acknowledged 0x3: Reserved 0x2: Smart-idle. Acknowledgement to an idle request is given based on the internal activity of the module.	RW	0x2
2	ENWAKEUP	WakeUp feature control 0x0: Wakeup is disabled 0x1: Wakeup is enabled	RW	0x0
1	SOFTRESET	Software reset. Set this bit to 1 to trigger a module reset. The bit is automatically reset by the hardware. During reads, it always returns 0. 0x0: Normal mode 0x1: The module is reset	RW	0x0
0	AUTOIDLE	Internal OCP clock gating strategy 0x0: OCP clock is free-running 0x1: Automatic OCP L3 and L4 clocks gating strategy is applied, based on the OCP interface activity.. Automatic functional clock gating is also applied to the functional clock based on the module activity (for instance DISPC_ pipe _ATTRIBUTES.ENABLE)	RW	0x1

Table 8-52. Register Call Summary for Register DISPC_SYSCONFIG

Display Controller

- DISPC Software Reset: [0][1]
- DISPC Idle Mode: [2]
- DISPC StandBy Mode: [3]
- DISPC Wakeup: [4]
- DISPC DMA Power Modes: [5]
- DISPC_COMMON Register Summary: [6]

Table 8-53. DISPC_SYSSTATUS

Address Offset	0x0000 0008		
Physical Address	0x5801 0008	**Instance**	DISPC_COMMON
Description	This register provides status information about the module, excluding the interrupt status information.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																											RESERVED	RESERVED	RESERVED	DISPC_VP1_RESETDONE	DISPC_FUNC_RESETDONE

Bits	Field Name	Description	Type	Reset
31:5	RESERVED		R	0x0
4	RESERVED		R	0x0
3	RESERVED		R	0x0
2	RESERVED		R	0x0
1	DISPC_VP1_RESETDONE	Reset status of DISPC VP1 pixel clock domain 0x0: Internal module reset is on-going 0x1: Reset completed	R	0x0
0	DISPC_FUNC_RESETDONE	Reset status of DISPC Functional clock domain 0x0: Internal module reset is on-going 0x1: Reset completed	R	0x0

Table 8-54. Register Call Summary for Register DISPC_SYSSTATUS

Display Controller

- DISPC Software Reset: [0][1]
- DISPC_COMMON Register Summary: [2]

Table 8-55. DISPC_IRQ_EOI

Address Offset	0x0000 0020		
Physical Address	0x5801 0020	**Instance**	DISPC_COMMON
Description	End Of Interrupt number (for H08 interrupt)		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															LINE_NUMBER

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	LINE_NUMBER	Software End Of Interrupt (EOI) control. Write number of interrupt output. The IP has 1 interrupt compliant with H08. 0x0: Reads always 0 (no EOI memory)	RW	0x0

Table 8-56. Register Call Summary for Register DISPC_IRQ_EOI

Display Controller
- DISPC_COMMON Register Summary: [0]

Table 8-57. DISPC_IRQSTATUS_RAW

Address Offset	0x0000 0024		
Physical Address	0x5801 0024	**Instance**	DISPC_COMMON
Description	Per-end of group (31 down to 0) internal signaling raw interrupt status vector, line #0. Raw status is set even if end of group (31 down to 0) interrupt is not enabled. Write 1 to set the (raw) status, mostly for debug.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																		WAKEUP_IRQ	RESERVED	WB_IRQ	RESERVED	RESERVED	VID2_IRQ	VID1_IRQ	RESERVED	RESERVED	GFX1_IRQ	RESERVED	RESERVED	RESERVED	VP1_IRQ

Bits	Field Name	Description	Type	Reset
31:14	RESERVED		R	0x0
13	WAKEUP_IRQ	Wake-up 0x0: No event pending 0x1: IRQ event pending	RW	0x0
12	RESERVED		R	0x0
11	WB_IRQ	WB IRQ STATUS register indicates the write-back pipeline interrupt events 0x0: No event pending 0x1: IRQ event pending	RW	0x0
10	RESERVED		R	0x0
9	RESERVED		R	0x0
8	VID2_IRQ	VID2 IRQ STATUS register indicates the video pipeline 2 interrupt events 0x0: No event pending 0x1: IRQ event pending	RW	0x0
7	VID1_IRQ	VID1 IRQ STATUS register indicates the video pipeline 1 interrupt events 0x0: No event pending 0x1: IRQ event pending	RW	0x0
6	RESERVED		R	0x0
5	RESERVED		R	0x0
4	GFX1_IRQ	GFX1 IRQ STATUS register indicates the graphics pipeline 1 interrupt events 0x0: No event pending 0x1: IRQ event pending	RW	0x0

Bits	Field Name	Description	Type	Reset
3	RESERVED		R	0x0
2	RESERVED		R	0x0
1	RESERVED		R	0x0
0	VP1_IRQ	VP1 IRQ STATUS register indicates the Video Port 1 interrupt events 0x0: No event pending 0x1: IRQ event pending	RW	0x0

Table 8-58. Register Call Summary for Register DISPC_IRQSTATUS_RAW

Display Controller
- DISPC_COMMON Register Summary: [0]

Table 8-59. DISPC_IRQSTATUS

Address Offset	0x0000 0028		
Physical Address	0x5801 0028	**Instance**	DISPC_COMMON
Description	Per-end of group (31 down to 0) internal signaling 'enabled' interrupt status vector, line #0. Enabled status isn't set unless event is enabled. Write 1 to clear the status after interrupt has been serviced (raw status gets cleared, i.e. even if not enabled).		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																		WAKEUP_IRQ	RESERVED	WB_IRQ	RESERVED	RESERVED	VID2_IRQ	VID1_IRQ	RESERVED	RESERVED	GFX1_IRQ	RESERVED	RESERVED	RESERVED	VP1_IRQ

Bits	Field Name	Description	Type	Reset
31:14	RESERVED		R	0x0
13	WAKEUP_IRQ	Wake-up 0x0: No event pending 0x1: IRQ event pending	RW	0x0
12	RESERVED		R	0x0
11	WB_IRQ	WB IRQ STATUS register indicates the write-back pipeline interrupt events 0x0: No event pending 0x1: IRQ event pending	RW	0x0
10	RESERVED		R	0x0
9	RESERVED		R	0x0
8	VID2_IRQ	VID2 IRQ STATUS register indicates the video pipeline 2 interrupt events 0x0: No event pending 0x1: IRQ event pending	RW	0x0
7	VID1_IRQ	VID1 IRQ STATUS register indicates the video pipeline 1 interrupt events 0x0: No event pending 0x1: IRQ event pending	RW	0x0
6	RESERVED		R	0x0
5	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
4	GFX1_IRQ	GFX1 IRQ STATUS register indicates the graphics pipeline 1 interrupt events 0x0: No event pending 0x1: IRQ event pending	RW	0x0
3	RESERVED		R	0x0
2	RESERVED		R	0x0
1	RESERVED		R	0x0
0	VP1_IRQ	VP1 IRQ STATUS register indicates the Video Port 1 interrupt events 0x0: No event pending 0x1: IRQ event pending	RW	0x0

Table 8-60. Register Call Summary for Register DISPC_IRQSTATUS

Display Controller

- DISPC Interrupt Requests: [0][1]
- DISPC_COMMON Register Summary: [2]

Table 8-61. DISPC_IRQENABLE_SET

Address Offset	0x0000 002C		
Physical Address	0x5801 002C	**Instance**	DISPC_COMMON
Description	Per-end of group (31 down to 0) internal event interrupt enable bit vector, line #0. Write 1 to set (enable interrupt). Readout equal to corresponding _CLR register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																		SET_WAKEUP_IRQ	RESERVED	SET_WB_IRQ	RESERVED	RESERVED	SET_VID2_IRQ	SET_VID1_IRQ	RESERVED	RESERVED	SET_GFX1_IRQ	RESERVED	RESERVED	RESERVED	RESERVED	SET_VP1_IRQ

Bits	Field Name	Description	Type	Reset
31:14	RESERVED		R	0x0
13	SET_WAKEUP_IRQ	Wake Up Mask 0x0: interrupt disabled 0x1: interrupt enabled	RW	0x0
12	RESERVED		R	0x0
11	SET_WB_IRQ	WB IRQ 0x0: interrupt disabled 0x1: interrupt enabled	RW	0x0
10	RESERVED		R	0x0
9	RESERVED		R	0x0
8	SET_VID2_IRQ	VID2 IRQ 0x0: interrupt disabled 0x1: interrupt enabled	RW	0x0
7	SET_VID1_IRQ	VID1 IRQ 0x0: interrupt disabled 0x1: interrupt enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
6	RESERVED		R	0x0
5	RESERVED		R	0x0
4	SET_GFX1_IRQ	GFX1 IRQ 0x0: interrupt disabled 0x1: interrupt enabled	RW	0x0
3	RESERVED		R	0x0
2	RESERVED		R	0x0
1	RESERVED		R	0x0
0	SET_VP1_IRQ	VP1 IRQ 0x0: interrupt disabled 0x1: interrupt enabled	RW	0x0

Table 8-62. Register Call Summary for Register DISPC_IRQENABLE_SET

Display Controller
- DISPC Interrupt Requests: [0][1]
- DISPC_COMMON Register Summary: [2]

Table 8-63. DISPC_IRQENABLE_CLR

Address Offset	0x0000 0030		
Physical Address	0x5801 0030	**Instance**	DISPC_COMMON
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																		CLR_WAKEUP_IRQ	RESERVED	CLR_WB_IRQ	RESERVED	RESERVED	CLR_VID2_IRQ	CLR_VID1_IRQ	RESERVED	RESERVED	CLR_GFX1_IRQ	RESERVED	RESERVED	RESERVED	CLR_VP1_IRQ

Bits	Field Name	Description	Type	Reset
31:14	RESERVED		R	0x0
13	CLR_WAKEUP_IRQ	Wake Up Mask 0x0: interrupt disabled 0x1: interrupt enabled	RW	0x0
12	RESERVED		R	0x0
11	CLR_WB_IRQ	WB IRQ 0x0: interrupt disabled 0x1: interrupt enabled	RW	0x0
10	RESERVED		R	0x0
9	RESERVED		R	0x0
8	CLR_VID2_IRQ	VID2 IRQ 0x0: interrupt disabled 0x1: interrupt enabled	RW	0x0
7	CLR_VID1_IRQ	VID1 IRQ 0x0: interrupt disabled 0x1: interrupt enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
6	RESERVED		R	0x0
5	RESERVED		R	0x0
4	CLR_GFX1_IRQ	GFX1 IRQ 0x0: interrupt disabled 0x1: interrupt enabled	RW	0x0
3	RESERVED		R	0x0
2	RESERVED		R	0x0
1	RESERVED		R	0x0
0	CLR_VP1_IRQ	VP1 IRQ 0x0: interrupt disabled 0x1: interrupt enabled	RW	0x0

Table 8-64. Register Call Summary for Register DISPC_IRQENABLE_CLR

Display Controller
- DISPC Interrupt Requests: [0]
- DISPC_COMMON Register Summary: [1]

Table 8-65. DISPC_IRQWAKEEN

Address Offset	0x0000 0034		
Physical Address	0x5801 0034	**Instance**	DISPC_COMMON
Description	IRQ wake up register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																				WB_IRQWAKEEN	RESERVED	RESERVED	VID2_IRQWAKEEN	VID1_IRQWAKEEN	RESERVED	RESERVED	GFX1_IRQWAKEEN	RESERVED	RESERVED	RESERVED	VP1_IRQWAKEEN

Bits	Field Name	Description	Type	Reset
31:12	RESERVED		R	0x0
11	WB_IRQWAKEEN	wakeupen for WB first level interrupt 0x0: Swakeup is not generated when this interrupt is asserted in idle mode 0x1: Swakeup is generated when this interrupt is asserted in idle mode	RW	0x0
10	RESERVED		R	0x0
9	RESERVED		R	0x0
8	VID2_IRQWAKEEN	wakeupen for VID2 first level interrupt 0x0: Swakeup is not generated when this interrupt is asserted in idle mode 0x1: Swakeup is generated when this interrupt is asserted in idle mode	RW	0x0
7	VID1_IRQWAKEEN	wakeupen for VID1 first level interrupt 0x0: Swakeup is not generated when this interrupt is asserted in idle mode 0x1: Swakeup is generated when this interrupt is asserted in idle mode	RW	0x0

Bits	Field Name	Description	Type	Reset
6	RESERVED		R	0x0
5	RESERVED		R	0x0
4	GFX1_IRQWAKEEN	wakeupen for GFX1 first level interrupt 0x0: Swakeup is not generated when this interrupt is asserted in idle mode 0x1: Swakeup is generated when this interrupt is asserted in idle mode	RW	0x0
3	RESERVED		R	0x0
2	RESERVED		R	0x0
1	RESERVED		R	0x0
0	VP1_IRQWAKEEN	wakeupen for VP1 first level interrupt 0x0: Swakeup is not generated when this interrupt is asserted in idle mode 0x1: Swakeup is generated when this interrupt is asserted in idle mode	RW	0x0

Table 8-66. Register Call Summary for Register DISPC_IRQWAKEEN

Display Controller
- DISPC_COMMON Register Summary: [0]

Table 8-67. DISPC_GLOBAL_MFLAG_ATTRIBUTE

Address Offset	0x0000 0040		
Physical Address	0x5801 0040	**Instance**	DISPC_COMMON
Description	MFLAG control register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																													MFLAG_START	MFLAG_CTRL	

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0
2	MFLAG_START	0x0: reset value, when the DMA buffer is empty at the beginning of the frame, the MFLAG of each pipe is kept at 0 until PRELOAD is reached, then based on MFLAG_CTRL, MFLAG[1:0] are generated and internal logic is arbitrating between pipeline requests 0x1: Even at the beginning of the frame when the DMA buffer is empty, MFLAG_CTRL is used to determine how MFLAG dedicated to each pipe signal shall be driven	RW	0x0
1:0	MFLAG_CTRL	0x0: MFLAG mechanism is disabled: MFLAG[1:0] out band signals are set to 0 0x1: MFLAG mechanism is enabled: MFLAG[1:0] out band signals are always set to 1 0x2: MFLAG mechanism is enabled and MFLAG[1:0] out band signals are dynamically set to 0 or 1 depending on the MFLAG rules	RW	0x0

Table 8-68. Register Call Summary for Register DISPC_GLOBAL_MFLAG_ATTRIBUTE

Display Controller

- DISPC DMA MFLAG Mechanism and Arbitration: [0][1][2][3][4]
- DISPC_COMMON Register Summary: [5]

Table 8-69. DISPC_GLOBAL_BUFFER

Address Offset	0x0000 0044		
Physical Address	0x5801 0044	**Instance**	DISPC_COMMON
Description	The register configures the DMA buffers allocations to the pipeline.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BUFFERFILLING	RESERVED										WB_BUFFER			RESERVED		VID2_BUFFER			VID1_BUFFER			RESERVED			RESERVED			GFX1_BUFFER			

Bits	Field Name	Description	Type	Reset
31	BUFFERFILLING	Controls if the DMA buffers are re-filled only when the LOW threshold is reached or if all DMA buffers are re-filled when at least one of them reaches the LOW threshold. wr: immediate 0x0: Each DMA buffer is re-filled when it reaches LOW threshold. 0x1: All DMA buffers are re-filled up to high threshold when at least one of them reaches the LOW threshold. (only active DMA buffers shall be considered and when reaching the end of the frame the DMA buffer goes to empty condition so no need to fill it again).	RW	0x0
30:21	RESERVED		R	0x0
20:18	WB_BUFFER	WB DMA buffer allocation to one of the pipelines. By default to video 1 pipeline. 0x6: DMA buffer allocated to the vdieo3 pipeline. 0x1: DMA buffer allocated to the gfx1 pipeline. 0x7: DMA buffer allocated to the write-back pipeline. 0x0: DMA buffer is not allocated to a pipeline. 0x2: DMA buffer allocated to the gfx2 pipeline. 0x4: DMA buffer allocated to the video1 pipeline. 0x5: DMA buffer allocated to the vdieo2 pipeline. 0x3: DMA buffer allocated to the gfx3 pipeline.	RW	0x7
17:15	RESERVED		R	0x0
14:12	VID2_BUFFER	Video2 DMA buffer allocation to one of the pipelines. By default to video 2 pipeline. 0x6: DMA buffer allocated to the vdieo3 pipeline. 0x1: DMA buffer allocated to the gfx1 pipeline. 0x7: DMA buffer allocated to the write-back pipeline. 0x0: DMA buffer is not allocated to a pipeline. 0x2: DMA buffer allocated to the gfx2 pipeline. 0x4: DMA buffer allocated to the video1 pipeline. 0x5: DMA buffer allocated to the vdieo2 pipeline. 0x3: DMA buffer allocated to the gfx3 pipeline.	RW	0x5

Bits	Field Name	Description	Type	Reset
11:9	VID1_BUFFER	Video1 DMA buffer allocation to one of the pipelines. By default to video 1 pipeline. 0x6: DMA buffer allocated to the vdieo3 pipeline. 0x1: DMA buffer allocated to the gfx1 pipeline. 0x7: DMA buffer allocated to the write-back pipeline. 0x0: DMA buffer is not allocated to a pipeline. 0x2: DMA buffer allocated to the gfx2 pipeline. 0x4: DMA buffer allocated to the video1 pipeline. 0x5: DMA buffer allocated to the vdieo2 pipeline. 0x3: DMA buffer allocated to the gfx3 pipeline.	RW	0x4
8:6	RESERVED		R	0x0
5:3	RESERVED		R	0x0
2:0	GFX1_BUFFER	Gfx1 DMA buffer allocation to one of the pipelines. By default to gfx1 pipeline. 0x6: DMA buffer allocated to the vdieo3 pipeline. 0x1: DMA buffer allocated to the gfx1 pipeline. 0x7: DMA buffer allocated to the write-back pipeline. 0x0: DMA buffer is not allocated to a pipeline. 0x2: DMA buffer allocated to the gfx2 pipeline. 0x4: DMA buffer allocated to the video1 pipeline. 0x5: DMA buffer allocated to the vdieo2 pipeline. 0x3: DMA buffer allocated to the gfx3 pipeline.	RW	0x1

Table 8-70. Register Call Summary for Register DISPC_GLOBAL_BUFFER

Display Controller
- DISPC DMA Engine: [0]
- DISPC_COMMON Register Summary: [1]

Table 8-71. DISPC_BA0_FLIPIMMEDIATE_EN

Address Offset	0x0000 0048		
Physical Address	0x5801 0048	**Instance**	DISPC_COMMON
Description	**Note: Register is not supported in this family of devices.**		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESERVED		

Bits	Field Name	Description	Type	Reset
31:6	RESERVED		R	0x0
5	RESERVED		R	0x0
4	RESERVED		R	0x0
3	RESERVED		R	0x0
2	RESERVED		R	0x0
1	RESERVED		R	0x0
0	RESERVED		R	0x0

Table 8-72. Register Call Summary for Register DISPC_BA0_FLIPIMMEDIATE_EN

Display Controller

- DISPC_COMMON Register Summary: [0]

Table 8-73. DISPC_DBG_CONTROL

Address Offset	0x0000 004C		
Physical Address	0x5801 004C	**Instance**	DISPC_COMMON
Description	DISPC debug bus control register **Note: GFX2, GFX3, VID-3, OVR3, OVR4, VP2, VP3, VP4, GLBCE1 and GLBCE2 are not supported in this family of devices.**		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								DBGMUXSEL							DBGEN

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8:1	DBGMUXSEL	0xD: Select first [191:160] bits of VID-2 debug bus 0x9: Select first [63:32] bits of VID-2 debug bus 0x15: Select first [191:160] bits of VID-3 debug bus 0x1E: Select GFX2 debug bus 0x8: Select first [31:0] bits of VID-2 debug bus 0x5: Select first [191:160] bits of VID-1 debug bus 0x1B: Select OVR3 debug bus 0x2: Select first [95:64] bits of VID-1 debug bus 0x4: Select first [159:128] bits of VID-1 debug bus 0x23: Select VP4 debug bus 0x1: Select first [63:32] bits of VID-1 debug bus 0x1D: Select GFX1 debug bus 0x24: Select GLBCE1 debug bus 0x0: Select first [31:0] bits of VID-1 debug bus 0xB: Select first [127:96] bits of VID-2 debug bus 0x20: Select VP1 debug bus 0x19: Select OVR1 debug bus 0x17: Select first [255:224] bits of VID-3 debug bus 0x11: Select first [63:32] bits of VID-3 debug bus 0x6: Select first [223:192] bits of VID-1 debug bus 0xA: Select first [95:64] bits of VID-2 debug bus 0x26: Reserved 0x10: Select first [31:0] bits of VID-3 debug bus 0x21: Select VP2 debug bus 0x25: Select GLBCE2 debug bus 0x12: Select first [95:64] bits of VID-3 debug bus 0x13: Select first [127:96] bits of VID-3 debug bus 0x18: Select WB debug bus 0x14: Select first [159:128] bits of VID-3 debug bus 0xE: Select first [223:192] bits of VID-2 debug bus 0x16: Select first [223:192] bits of VID-3 debug bus 0x1C: Select OVR4 debug bus 0x7: Select first [255:224] bits of VID-1 debug bus 0x3: Select first [127:96] bits of VID-1 debug bus 0x1F: Select GFX3 debug bus 0x1A: Select OVR2 debug bus 0xF: Select first [255:224] bits of VID-2 debug bus 0xC: Select first [159:128] bits of VID-2 debug bus 0x22: Select VP3 debug bus	RW	0x0
0	DBGEN	Enable debug ports 0x0: DBGDIS 0x1: DBGEN	RW	0x0

Table 8-74. Register Call Summary for Register DISPC_DBG_CONTROL

Display Controller

- DISPC_COMMON Register Summary: [0]

Table 8-75. DISPC_DBG_STATUS

Address Offset	0x0000 0050		
Physical Address	0x5801 0050	**Instance**	DISPC_COMMON
Description	DISPC debug status register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DBGOUT																															

Bits	Field Name	Description	Type	Reset
31:0	DBGOUT		R	0x0

Table 8-76. Register Call Summary for Register DISPC_DBG_STATUS

Display Controller
- DISPC_COMMON Register Summary: [0]

Table 8-77. DISPC_CLKGATING_DISABLE

Address Offset	0x0000 0054		
Physical Address	0x5801 0054	**Instance**	DISPC_COMMON
Description	Register to control clock gating at DISPC sub-module level **Note: CUR, GFX2, GFX3, VID3, OVR3, OVR4, VP2, VP3, VP4, GLBCE1 and GLBCE2 are not supported in this family of devices.**		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED					VP4	VP3	VP2	VP1	OVR4	OVR3	OVR2	OVR1	CUR	WB	GLBCE2	GLBCE1	GFX3	GFX2	GFX1	VID3	VID2	VID1	DMA_CH8	DMA_CH7	DMA_CH6	DMA_CH5	DMA_CH4	DMA_CH3	DMA_CH2	DMA_CH1	DMA_COMMON

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26	VP4	Clock gating control for VP4 0x0: Clock-Gating is enabled 0x1: Clock-gating is disabled. Clocks are free running	RW	0x0
25	VP3	Clock gating control for VP3 0x0: Clock-Gating is enabled 0x1: Clock-gating is disabled. Clocks are free running	RW	0x0
24	VP2	Clock gating control for VP2 0x0: Clock-Gating is enabled 0x1: Clock-gating is disabled. Clocks are free running	RW	0x0
23	VP1	Clock gating control for VP1 0x0: Clock-Gating is enabled 0x1: Clock-gating is disabled. Clocks are free running	RW	0x0
22	OVR4	Clock gating control for OVR4 0x0: Clock-Gating is enabled 0x1: Clock-gating is disabled. Clocks are free running	RW	0x0

Bits	Field Name	Description	Type	Reset
21	OVR3	Clock gating control for OVR3 0x0: Clock-Gating is enabled 0x1: Clock-gating is disabled. Clocks are free running	RW	0x0
20	OVR2	Clock gating control for OVR2 0x0: Clock-Gating is enabled 0x1: Clock-gating is disabled. Clocks are free running	RW	0x0
19	OVR1	Clock gating control for OVR1 0x0: Clock-Gating is enabled 0x1: Clock-gating is disabled. Clocks are free running	RW	0x0
18	CUR	Clock gating control for CUR 0x0: Clock-Gating is enabled 0x1: Clock-gating is disabled. Clocks are free running	RW	0x0
17	WB	Clock gating control for WB 0x0: Clock-Gating is enabled 0x1: Clock-gating is disabled. Clocks are free running	RW	0x0
16	GLBCE2	Clock gating control for GLBCE2 0x0: Clock-Gating is enabled 0x1: Clock-gating is disabled. Clocks are free running	RW	0x0
15	GLBCE1	Clock gating control for GLBCE1 0x0: Clock-Gating is enabled 0x1: Clock-gating is disabled. Clocks are free running	RW	0x0
14	GFX3	Clock gating control for GFX3 0x0: Clock-Gating is enabled 0x1: Clock-gating is disabled. Clocks are free running	RW	0x0
13	GFX2	Clock gating control for GFX2 0x0: Clock-Gating is enabled 0x1: Clock-gating is disabled. Clocks are free running	RW	0x0
12	GFX1	Clock gating control for GFX1 0x0: Clock-Gating is enabled 0x1: Clock-gating is disabled. Clocks are free running	RW	0x0
11	VID3	Clock gating control for VID3 0x0: Clock-Gating is enabled 0x1: Clock-gating is disabled. Clocks are free running	RW	0x0
10	VID2	Clock gating control for VID2 0x0: Clock-Gating is enabled 0x1: Clock-gating is disabled. Clocks are free running	RW	0x0
9	VID1	Clock gating control for VID1 0x0: Clock-Gating is enabled 0x1: Clock-gating is disabled. Clocks are free running	RW	0x0
8	DMA_CH8	Clock gating control for DMA Channel-8 0x0: Clock-Gating is enabled 0x1: Clock-gating is disabled. Clocks are free running	RW	0x0
7	DMA_CH7	Clock gating control for DMA Channel-7 0x0: Clock-Gating is enabled 0x1: Clock-gating is disabled. Clocks are free running	RW	0x0
6	DMA_CH6	Clock gating control for DMA Channel-6 0x0: Clock-Gating is enabled 0x1: Clock-gating is disabled. Clocks are free running	RW	0x0

Bits	Field Name	Description	Type	Reset
5	DMA_CH5	Clock gating control for DMA Channel-5 0x0: Clock-Gating is enabled 0x1: Clock-gating is disabled. Clocks are free running	RW	0x0
4	DMA_CH4	Clock gating control for DMA Channel-4 0x0: Clock-Gating is enabled 0x1: Clock-gating is disabled. Clocks are free running	RW	0x0
3	DMA_CH3	Clock gating control for DMA Channel-3 0x0: Clock-Gating is enabled 0x1: Clock-gating is disabled. Clocks are free running	RW	0x0
2	DMA_CH2	Clock gating control for DMA Channel-2 0x0: Clock-Gating is enabled 0x1: Clock-gating is disabled. Clocks are free running	RW	0x0
1	DMA_CH1	Clock gating control for DMA Channel-1 0x0: Clock-Gating is enabled 0x1: Clock-gating is disabled. Clocks are free running	RW	0x0
0	DMA_COMMON	Clock gating control for DMA_COMMON module 0x0: Clock-Gating is enabled 0x1: Clock-gating is disabled. Clocks are free running	RW	0x0

Table 8-78. Register Call Summary for Register DISPC_CLKGATING_DISABLE

Display Controller
- DISPC_COMMON Register Summary: [0]

8.2.5.3 DISPC_GFX1 Registers

8.2.5.3.1 DISPC_GFX1 Register Summary

Table 8-79. DISPC_GFX1 Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	DISPC_GFX1 L3_MAIN Physical Address
DISPC_GFX1_ATTRIBUTES	RW	32	0x0000 0000	0x5801 1000
DISPC_GFX1_ATTRIBUTES2	RW	32	0x0000 0004	0x5801 1004
DISPC_GFX1_BA_j[(1)]	RW	32	0x0000 0008 + (0x4 * j)	0x5801 1008 + (0x4 * j)
DISPC_GFX1_BUF_SIZE_STATUS	R	32	0x0000 0010	0x5801 1010
DISPC_GFX1_BUF_THRESHOLD	RW	32	0x0000 0014	0x5801 1014
DISPC_GFX1_GLOBAL_ALPHA	RW	32	0x0000 0018	0x5801 1018
DISPC_GFX1_IRQENABLE	RW	32	0x0000 001C	0x5801 101C
DISPC_GFX1_IRQSTATUS	RW	32	0x0000 0020	0x5801 1020
DISPC_GFX1_MFLAG_THRESHOLD	RW	32	0x0000 0024	0x5801 1024
DISPC_GFX1_PIXEL_INC	RW	32	0x0000 0028	0x5801 1028
DISPC_GFX1_POSITION	RW	32	0x0000 002C	0x5801 102C
DISPC_GFX1_PRELOAD	RW	32	0x0000 0030	0x5801 1030
DISPC_GFX1_ROW_INC	RW	32	0x0000 0034	0x5801 1034
DISPC_GFX1_SIZE	RW	32	0x0000 0038	0x5801 1038
DISPC_GFX1_CLUT	W	32	0x0000 003C	0x5801 103C

(1) j = 0 to 1

8.2.5.3.2 DISPC_GFX1 Register Description

Table 8-80. DISPC_GFX1_ATTRIBUTES

Address Offset	0x0000 0000		
Physical Address	0x5801 1000	**Instance**	DISPC_GFX1
Description	The register configures the graphics attributes. Shadow register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED			PREMULTIPLYALPHA	ZORDER			ANTIFLICKERg	RESERVED					RESERVED	SELFREFRESHAUTO	RESERVED	SELFREFRESH	ARBITRATION	RESERVED		BUFPRELOAD	CHANNELOUT			NIBBLEMODE	FORMAT					ENABLE

Bits	Field Name	Description	Type	Reset
31	RESERVED		R	0x0
30	RESERVED		R	0x0
29	RESERVED		R	0x0
28	PREMULTIPLYALPHA	The field configures the DISPC GFX to process incoming data as premultiplied alpha data or non premultiplied alpha data. Default setting is non premultiplied alpha data. 0x0: Non premultiplyalpha data color component 0x1: Premultiplyalpha data color component	RW	0x0
27:25	ZORDER	Z-Order defining the priority of the layer compared to others when overlaying. It is SW responsibility to ensure that each layer connected to the same overlay manager has a different z-order value. 0x1: Z-order 1: layer above layer with z-order value of 0 and below layers with z-order values of 2 and 3 0x0: Z-order 0: layer above solid background color and below layer with higher Z-order values. 0x2: Z-order 2: layer above layers with z-order value of 0 and 1 and below layer with z-order value of 3 0x4: Z-order 4: layer above layers with z-order value of 0, 1, 2 and 3 and below layer with z-order value of 5 0x5: Z-order 5: layer above all the other layers except cursor 0x3: Z-order 3: layer above layers with z-order value of 0, 1 and 2 and below layer with z-order value of 4	RW	0x0
24	ANTIFLICKER	Anti-aliasing filtering using a 3-tap filter with hardcoded coefficients (1/4, 1/2, 1/4) 0x0: Anti-aliasing disabled. 0x1: Anti-aliasing enabled.	RW	0x0
23:19	RESERVED		R	0x0
18	RESERVED		R	0x0
17	SELFREFRESHAUTO	Automatic self refresh mode 0x0: The transition from Selfrefresh 'disabled' to 'enabled' is controlled by SW 0x1: The transition from Selfrefresh 'disabled' to 'enabled' is controlled only by HW	RW	0x0
16	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
15	SELFREFRESH	Enables the self refresh of the graphics window from its own DMA buffer. This bit should be set only after havin set the GO bit of the channel and read back a zero in its field. 0x0: The graphics pipeline accesses the interconnect to fetch data from the system memory. 0x1: The graphics pipeline does not need anymore to fetch data from memory. Only the graphics DMA buffer is used. It takes effect after the frame has been loaded in the DMA buffer.	RW	0x0
14	ARBITRATION	Determines the priority of the graphics pipeline. When the graphics pipeline is one of the high priority pipelines. The arbitration wheel gives always the priority first to the high priority pipelines using round-robin between them. When there is only normal priority pipelines sending requests, the round-robin applies between them. 0x0: The graphics pipeline is one of the normal priority pipeline. 0x1: The graphics pipeline is one of the high priority pipeline.	RW	0x0
13:12	RESERVED		R	0x0
11	BUFPRELOAD	Graphics Preload Value 0x0: H/W prefetches pixels up to the preload value defined in the preload register 0x1: H/W prefetches pixels up to high threshold value	RW	0x0
10:8	CHANNELOUT	Graphics Channel Out configuration wr: immediate 0x0: OVR1 (VP1) 0x1: OVR2 0x4: WB	RW	0x0
7	NIBBLEMODE	Graphics Nibble mode (only for 1-, 2- and 4-bpp) 0x0: Nibble mode is disabled 0x1: Nibble mode is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
6:1	FORMAT	Graphics format. It defines the pixel format when fetching the graphics picture into memory. 0x9: RGBA32-8888 0x15: BITMAP8 (CLUT is required) 0x8: ABGR32-8888 0x5: ARGB16-1555 0x2E: xRGB32-2101010 0x2: RGBA16-4444 0xB: RGB24-888 (24-bit container) 0x4: BGR16-565 0x32: RESERVED3 0x23: RESERVED1 0x1: ABGR16-4444 0x28: xBGR32_8888 0x0: ARGB16-4444 0x25: xRGB16-1555 0x20: xRGB12-4444 0x3: RGB16-565 0x30: xRGB64-16161616 0x31: RGBX64_16161616 0x11: RGBA64_16161616 0x6: ABGR16-1555 0x2F: xBGR32-2101010 0xA: BGRA32-8888 0x26: xBGR16-1555 0x10: ARGB64-16161616 0x21: xBGR16-4444 0x12: BITMAP1 (CLUT is required) 0x13: BITMAP2 (CLUT is required) 0x2B: RESERVED2 0x14: BITMAP4 (CLUT is required) 0xE: ARGB32-2101010 0x16: RESERVED 0x2A: BGRX32_8888 0x7: ARGB32-8888 0x27: xRGB32-8888 (32-bit container) 0x29: RGBx32-8888 (24-bit RGB aligned on MSB of the 32-bit container) 0xF: ABGR32-2101010 0xC: RESERVED4 0x22: RGBx16-4444	RW	0x0
0	ENABLE	Graphics Enable 0x0: Graphics disabled (graphics pipeline inactive and graphics window not present) 0x1: Graphics enabled (graphics pipeline active and graphics window present on the screen)	RW	0x0

Table 8-81. Register Call Summary for Register DISPC_GFX1_ATTRIBUTES

Display Controller

- DISPC DMA Addressing and Bursts: [0]
- DISPC DMA Buffers: [1]
- DISPC DMA Arbitration: [2]
- DISPC DMA Power Modes: [3][4]
- DISPC Memory Formats: [5]
- DISPC Graphics Pipeline: [6][7][8][9][10]
- DISPC GFX Anti-Aliasing Filter: [11]
- DISPC GFX Color Look-Up Table (CLUT): [12]
- DISPC Overlay Managers: [13][14]
- DISPC Overlay Priority Rule: [15]
- DISPC Overlay Alpha Blender: [17]
- DISPC_GFX1 Register Summary: [18]

Table 8-82. DISPC_GFX1_ATTRIBUTES2

Address Offset	0x0000 0004		
Physical Address	0x5801 1004	**Instance**	DISPC_GFX1
Description	The register configures the graphics attributes. Shadow register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	TAGS					RESERVED	REGION_BASED	RESERVED							SECURE	RESERVED															

Bits	Field Name	Description	Type	Reset
31	RESERVED		R	0x0
30:26	TAGS	Number of OCP TAGS to be used for the pipeline (from 1 to 32)	RW	0x1f
25	RESERVED		R	0x0
24	REGION_BASED	Enable region-based mechanism 0x0: DISABLE 0x1: ENABLE	RW	0x0
23:17	RESERVED		R	0x0
16	SECURE	OCP requests corresponds to pipeline data are secure/unsecure. The bit-field can be modified only by secure transaction using MReqSecure qualifier.	RW	0x0
15:0	RESERVED		R	0x0

Table 8-83. Register Call Summary for Register DISPC_GFX1_ATTRIBUTES2

Display Controller

- Region-Based Mechanism Overview: [0]
- DISPC_GFX1 Register Summary: [1]

Table 8-84. DISPC_GFX1_BA_j

Address Offset	0x0000 0008 + (0x4 * j)		
Physical Address	0x5801 1008 + (0x4 * j)	**Instance**	DISPC_GFX1
Description	The register configures the base address of the graphics buffer displayed in the graphics window (0 & 1 :for ping-pong mechanism with external trigger, based on the field polarity). Shadow register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BA																															

Bits	Field Name	Description	Type	Reset
31:0	BA	Graphics base address When decompression is enabled bit[5:0] shall be set to 0. Base address of the graphics buffer (aligned on pixel size boundary) (in case of RGB24 packed format, 4-pixel alignment is required) When the TILER is addressed, the bits: [28:27] = 0x0 for 8-bit tiled [28:27] = 0x1 for 16-bit tiled [28:27] = 0x2 for 32-bit tiled [28:27] = 0x3 for page mode [31:29] = 0x0 for 0-degree view [31:29] = 0x1 for 180-degree view + mirroring [31:29] = 0x2 for 0-degree view + mirroring [31:29] = 0x3 for 180-degree view [31:29] = 0x4 for 270-degree view + mirroring [31:29] = 0x5 for 270-degree view [31:29] = 0x6 for 90-degree view [31:29] = 0x7 for 90-degree view + mirroring Otherwise the bits indicated the corresponding bit address to access the SDRAM.	RW	0x0

Table 8-85. Register Call Summary for Register DISPC_GFX1_BA_j

Display Controller

- DISPC DMA Addressing and Bursts: [0][1]
- DISPC_GFX1 Register Summary: [2]

Table 8-86. DISPC_GFX1_BUF_SIZE_STATUS

Address Offset	0x0000 0010		
Physical Address	0x5801 1010	**Instance**	DISPC_GFX1
Description	The register defines the Graphics buffer size		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																BUFSIZE															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Write 0's for future compatibility. Reads return 0	R	0x0
15:0	BUFSIZE	DMA buffer Size in number of 128-bits	R	0xa00

Table 8-87. Register Call Summary for Register DISPC_GFX1_BUF_SIZE_STATUS

Display Controller

- DISPC_GFX1 Register Summary: [0]

Table 8-88. DISPC_GFX1_BUF_THRESHOLD

Address Offset	0x0000 0014		
Physical Address	0x5801 1014	**Instance**	DISPC_GFX1
Description	The register configures the graphics buffer. Shadow register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BUFHIGHTHRESHOLD																BUFLOWTHRESHOLD															

Bits	Field Name	Description	Type	Reset
31:16	BUFHIGHTHRESHOLD	DMA buffer High Threshold Number of 128-bits defining the threshold value.	RW	0x9ff
15:0	BUFLOWTHRESHOLD	DMA buffer Low Threshold Number of 128-bits defining the threshold value.	RW	0x9f8

Table 8-89. Register Call Summary for Register DISPC_GFX1_BUF_THRESHOLD

Display Controller

- DISPC DMA Buffers: [0][1]
- DISPC_GFX1 Register Summary: [2]

Table 8-90. DISPC_GFX1_GLOBAL_ALPHA

Address Offset	0x0000 0018		
Physical Address	0x5801 1018	**Instance**	DISPC_GFX1
Description	The register defines the global alpha value for the graphics pipeline. Shadow register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								GLOBALALPHA							

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x0
7:0	GLOBALALPHA	Global alpha value from 0 to 255. 0 corresponds to fully transparent and 255 to fully opaque.	RW	0xff

Table 8-91. Register Call Summary for Register DISPC_GFX1_GLOBAL_ALPHA

Display Controller

- DISPC Overlay Alpha Blender: [0]
- DISPC_GFX1 Register Summary: [1]

Table 8-92. DISPC_GFX1_IRQENABLE

Address Offset	0x0000 001C		
Physical Address	0x5801 101C	**Instance**	DISPC_GFX1
Description	This register allows to mask/unmask the module internal sources of interrupt, on an event-by-event basis		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												GFXREGIONBASEDPIPEEND_EN	GFXREGIONBASEDPIPESTART_EN	GFXENDWINDOW_EN	GFXBUFFERUNDERFLOW_EN

Bits	Field Name	Description	Type	Reset
31:4	RESERVED		R	0x0
3	GFXREGIONBASEDPIPEEND_EN	PIPE end window IRQ for region-based feature 0x0: GFXREGIONBASEDPIPEEND is masked 0x1: GFXREGIONBASEDPIPEEND generates an interrupt when it occurs	RW	0x0
2	GFXREGIONBASEDPIPESTART_EN	PIPE start window IRQ for region-based feature 0x0: GFXREGIONBASEDPIPESTART is masked 0x1: GFXREGIONBASEDPIPESTART generates an interrupt when it occurs	RW	0x0
1	GFXENDWINDOW_EN	The end of Gfx Window has been reached. It is detected by the overlay manager when the full graphics has been displayed. 0x0: GfxEndWindow is masked 0x1: GfxEndWindow generates an interrupt when it occurs	RW	0x0
0	GFXBUFFERUNDERFLOW_EN	Gfx DMA Buffer Underflow. The DMA buffer is not necessarly empty but required data are not present in the DMA buffer (due to out of order responses) 0x0: GfxBufferUnderflow is masked 0x1: GfxBufferUnderflow generates an interrupt when it occurs	RW	0x0

Table 8-93. Register Call Summary for Register DISPC_GFX1_IRQENABLE

Display Controller

- DISPC Interrupt Requests: [0]
- DISPC_GFX1 Register Summary: [1]

Table 8-94. DISPC_GFX1_IRQSTATUS

Address Offset	0x0000 0020		
Physical Address	0x5801 1020	**Instance**	DISPC_GFX1
Description	This register regroups all the status of the module internal events that generate an interrupt. Write 1 to a given bit resets this bit		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												GFXREGIONBASEDPIPEEND_IRQ	GFXREGIONBASEDPIPESTART_IRQ	GFXENDWINDOW_IRQ	GFXBUFFERUNDERFLOW_IRQ

Bits	Field Name	Description	Type	Reset
31:4	RESERVED		R	0x0
3	GFXREGIONBASEDPIPEEND_IRQ	PIPE end window IRQ for region-based feature 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW	0x0
2	GFXREGIONBASEDPIPESTART_IRQ	PIPE start window IRQ for region-based feature 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW	0x0
1	GFXENDWINDOW_IRQ	The end of Gfx Window has been reached. It is detected by the overlay manager when the full graphics has been displayed. 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW	0x0
0	GFXBUFFERUNDERFLOW_IRQ	Gfx DMA Buffer Underflow. The DMA buffer is not necessarly empty but required data are not present in the DMA buffer (due to out of order responses) 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW	0x0

Table 8-95. Register Call Summary for Register DISPC_GFX1_IRQSTATUS

Display Controller

- DISPC Interrupt Requests: [0]
- DISPC_GFX1 Register Summary: [1]

Table 8-96. DISPC_GFX1_MFLAG_THRESHOLD

Address Offset	0x0000 0024		
Physical Address	0x5801 1024	**Instance**	DISPC_GFX1
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HT_MFLAG																LT_MFLAG															

Bits	Field Name	Description	Type	Reset
31:16	HT_MFLAG		RW	0x0
15:0	LT_MFLAG		RW	0x0

Table 8-97. Register Call Summary for Register DISPC_GFX1_MFLAG_THRESHOLD

Display Controller

- DISPC DMA MFLAG Mechanism and Arbitration: [0][1]
- DISPC_GFX1 Register Summary: [2]

Table 8-98. DISPC_GFX1_PIXEL_INC

Address Offset	0x0000 0028		
Physical Address	0x5801 1028	**Instance**	DISPC_GFX1
Description	The register configures the number of bytes to increment between two pixels. Shadow register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								PIXELINC							

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Write 0's for future compatibility. Reads return 0	R	0x0
7:0	PIXELINC	Number of bytes to increment between two pixels. Encoded unsigned value (from 1 to 255) to specify the number of bytes between two pixels in the graphics buffer. The value 0 is invalid. The value 1 means next pixel. The value 1+n*bpp means increment of n pixels.	RW	0x1

Table 8-99. Register Call Summary for Register DISPC_GFX1_PIXEL_INC

Display Controller

- DISPC DMA Addressing and Bursts: [0]
- DISPC DMA Predecimation: [1]
- DISPC_GFX1 Register Summary: [2]

Table 8-100. DISPC_GFX1_POSITION

Address Offset	0x0000 002C		
Physical Address	0x5801 102C	**Instance**	DISPC_GFX1
Description	The register configures the position of the graphics window. Shadow register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								POSY								RESERVED								POSX							

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:16	POSY	Y position of the graphics window. Encoded value (from 0 to 4095) to specify the Y position of the graphics window on the screen. The line at the top has the Y-position 0.	RW	0x0
15:12	RESERVED		R	0x0
11:0	POSX	X position of the graphics window. Encoded value (from 0 to 4095) to specify the X position of the graphics window on the screen. The first pixel on the left of the screen has the X-position 0.	RW	0x0

Table 8-101. Register Call Summary for Register DISPC_GFX1_POSITION

Display Controller

- DISPC Overlay Priority Rule: [0]
- DISPC_GFX1 Register Summary: [2]

Table 8-102. DISPC_GFX1_PRELOAD

Address Offset	0x0000 0030		
Physical Address	0x5801 1030	**Instance**	DISPC_GFX1
Description	The register configures the graphics DMA buffer Shadow register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																				PRELOAD											

Bits	Field Name	Description	Type	Reset
31:12	RESERVED	Write 0's for future compatibility. Reads return 0	R	0x0
11:0	PRELOAD	DMA buffer preload value Number of 128-bit words defining the preload value.	RW	0x100

Table 8-103. Register Call Summary for Register DISPC_GFX1_PRELOAD

Display Controller

- DISPC DMA Buffers: [0]
- DISPC_GFX1 Register Summary: [1]

Table 8-104. DISPC_GFX1_ROW_INC

Address Offset	0x0000 0034		
Physical Address	0x5801 1034	**Instance**	DISPC_GFX1
Description	The register configures the number of bytes to increment at the end of the row. Shadow register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ROWINC																															

Bits	Field Name	Description	Type	Reset
31:0	ROWINC	Number of bytes to increment at the end of the row Encoded unsigned value to specify the number of bytes to increment at the end of the row in the graphics buffer. The value 0 is invalid. The value 1 means next pixel. The value 1+n*bpp means increment of n pixels. The value 1-(n+1)*bpp means decrement of n pixels.	RW	0x1

Table 8-105. Register Call Summary for Register DISPC_GFX1_ROW_INC

Display Controller

- DISPC DMA Addressing and Bursts: [0][1]
- DISPC DMA Predecimation: [2]
- DISPC_GFX1 Register Summary: [3]

Table 8-106. DISPC_GFX1_SIZE

Address Offset	0x0000 0038		
Physical Address	0x5801 1038	**Instance**	DISPC_GFX1
Description	The register configures the size of the graphics window. Shadow register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				SIZEY												RESERVED				SIZEX											

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:16	SIZEY	Number of lines of the graphics window. Encoded value (from 1 to 4096) to specify the number of lines of the graphics window (program size -1).	RW	0x0
15:12	RESERVED		R	0x0
11:0	SIZEX	Number of pixels of the graphics window. Encoded value (from 1 to 4096) to specify the number of pixels per line of the graphics window (program size -1).	RW	0x0

Table 8-107. Register Call Summary for Register DISPC_GFX1_SIZE

Display Controller

- DISPC DMA Addressing and Bursts: [0]
- DISPC Overlay Priority Rule: [1]
- DISPC_GFX1 Register Summary: [3]

Table 8-108. DISPC_GFX1_CLUT

Address Offset	0x0000 003C		
Physical Address	0x5801 103C	**Instance**	DISPC_GFX1
Description	The register configures the Color Look Up Table (CLUT) for GFX pipeline. CLUT is used in conjunction with bitmap formats		
Type	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INDEX								VALUE_R								VALUE_G								VALUE_B							

Bits	Field Name	Description	Type	Reset
31:24	INDEX	Defines the location in the table where the bit-field VALUE is stored	W	0x0
23:16	VALUE_R	8-bit value used to defined the value to store at the location in the table defined by the bit-field INDEX.	W	0x0
15:8	VALUE_G	8-bit value used to defined the value to store at the location in the table defined by the bit-field INDEX.	W	0x0
7:0	VALUE_B	8-bit value used to defined the value to store at the location in the table defined by the bit-field INDEX.	W	0x0

Table 8-109. Register Call Summary for Register DISPC_GFX1_CLUT

Display Controller
- DISPC GFX Color Look-Up Table (CLUT): [0][1]
- DISPC_GFX1 Register Summary: [2]

8.2.5.4 DISPC_WB Registers

8.2.5.4.1 DISPC_WB Register Summary

Table 8-110. DISPC_WB Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	DISPC_WB L3_MAIN Physical Address
DISPC_WB_ACCUH_j [(1)]	RW	32	0x0000 0000 + (0x4 * j)	0x5801 5000 + (0x4 * j)
DISPC_WB_ACCUH2_j [(1)]	RW	32	0x0000 0008 + (0x4 * j)	0x5801 5008 + (0x4 * j)
DISPC_WB_ACCUV_j [(1)]	RW	32	0x0000 0010 + (0x4 * j)	0x5801 5010 + (0x4 * j)
DISPC_WB_ACCUV2_j [(1)]	RW	32	0x0000 0018 + (0x4 * j)	0x5801 5018 + (0x4 * j)
DISPC_WB_ATTRIBUTES	RW	32	0x0000 0020	0x5801 5020
DISPC_WB_ATTRIBUTES2	RW	32	0x0000 0024	0x5801 5024
DISPC_WB_BA_j [(1)]	RW	32	0x0000 0028 + (0x4 * j)	0x5801 5028 + (0x4 * j)
DISPC_WB_BA_UV_j [(1)]	RW	32	0x0000 0030 + (0x4 * j)	0x5801 5030 + (0x4 * j)
DISPC_WB_BUF_SIZE_STATUS	R	32	0x0000 0038	0x5801 5038
DISPC_WB_BUF_THRESHOLD	RW	32	0x0000 003C	0x5801 503C
DISPC_WB_CONV_COEF0	RW	32	0x0000 0040	0x5801 5040
DISPC_WB_CONV_COEF1	RW	32	0x0000 0044	0x5801 5044
DISPC_WB_CONV_COEF2	RW	32	0x0000 0048	0x5801 5048
DISPC_WB_CONV_COEF3	RW	32	0x0000 004C	0x5801 504C
DISPC_WB_CONV_COEF4	RW	32	0x0000 0050	0x5801 5050
DISPC_WB_CONV_COEF5	RW	32	0x0000 0054	0x5801 5054
DISPC_WB_CONV_COEF6	RW	32	0x0000 0058	0x5801 5058

(1) j = 0 to 1

Table 8-110. DISPC_WB Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	DISPC_WB L3_MAIN Physical Address
DISPC_WB_FIRH	RW	32	0x0000 005C	0x5801 505C
DISPC_WB_FIRH2	RW	32	0x0000 0060	0x5801 5060
DISPC_WB_FIRV	RW	32	0x0000 0064	0x5801 5064
DISPC_WB_FIRV2	RW	32	0x0000 0068	0x5801 5068
DISPC_WB_FIR_COEF_H0_i [2]	RW	32	0x0000 006C + (0x4 * i)	0x5801 506C + (0x4 * i)
DISPC_WB_FIR_COEF_H0_C_i [2]	RW	32	0x0000 0090 + (0x4 * i)	0x5801 5090 + (0x4 * i)
DISPC_WB_FIR_COEF_H12_k [3]	RW	32	0x0000 00B4 + (0x4 * k)	0x5801 50B4 + (0x4 * k)
DISPC_WB_FIR_COEF_H12_C_k [3]	RW	32	0x0000 00F4 + (0x4 * k)	0x5801 50F4 + (0x4 * k)
DISPC_WB_FIR_COEF_V0_i [2]	RW	32	0x0000 0134 + (0x4 * i)	0x5801 5134 + (0x4 * i)
DISPC_WB_FIR_COEF_V0_C_i [2]	RW	32	0x0000 0158 + (0x4 * i)	0x5801 5158 + (0x4 * i)
DISPC_WB_FIR_COEF_V12_k [3]	RW	32	0x0000 017C + (0x4 * k)	0x5801 517C + (0x4 * k)
DISPC_WB_FIR_COEF_V12_C_k [3]	RW	32	0x0000 01BC + (0x4 * k)	0x5801 51BC + (0x4 * k)
DISPC_WB_IRQENABLE	RW	32	0x0000 01FC	0x5801 51FC
DISPC_WB_IRQSTATUS	RW	32	0x0000 0200	0x5801 5200
DISPC_WB_MFLAG_THRESHOLD	RW	32	0x0000 0204	0x5801 5204
DISPC_WB_PICTURE_SIZE	RW	32	0x0000 0208	0x5801 5208
DISPC_WB_SIZE	RW	32	0x0000 0210	0x5801 5210
DISPC_WB_POSITION	RW	32	0x0000 0214	0x5801 5214
DISPC_WB_REGION_BASED_TOTAL_PICTURE_SIZEY	RW	32	0x0000 0218	0x5801 5218

(2) i = 0 to 8
(3) k = 0 to 15

8.2.5.4.2 DISPC_WB Register Description

Table 8-111. DISPC_WB_ACCUH_j

Address Offset	0x0000 0000		
Physical Address	0x5801 5000 + (0x4 * j)	**Instance**	DISPC_WB
Description	The register configures the resize accumulator init values for horizontal up/down-sampling of the wirte-back window It is used for ARGB and Y setting. Shadow register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								HORIZONTALACCU																							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:0	HORIZONTALACCU	Horizontal initialization accu signed value.	RW	0x0

Table 8-112. Register Call Summary for Register DISPC_WB_ACCUH_j

Display Controller
- DISPC WB Scaler Unit: [0]
- DISPC_WB Register Summary: [1]

Table 8-113. DISPC_WB_ACCUH2_j

Address Offset	0x0000 0008		
Physical Address	0x5801 5008 + (0x4 * j)	**Instance**	DISPC_WB
Description	The register configures the resize accumulator init value for horizontal up/down-sampling of the write-back window It is used for Cb and Cr setting. It is used only when the pixel format at the input of the filter is one of the YUV formats. If the pixel format at the input of the filter is ARGB (all ARGB, RGB, RGBA are converted to ARGB32-8888 by the color space conversion before going to the filter is the color space conversion is done before the filter). When the register is not used by the HW, any value can be used for the bit-fields. Shadow register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								HORIZONTALACCU																							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:0	HORIZONTALACCU	Horizontal initialization accu signed value	RW	0x0

Table 8-114. Register Call Summary for Register DISPC_WB_ACCUH2_j

Display Controller
- DISPC WB Scaler Unit: [0]
- DISPC_WB Register Summary: [1]

Table 8-115. DISPC_WB_ACCUV_j

Address Offset	0x0000 0010		
Physical Address	0x5801 5010 + (0x4 * j)	**Instance**	DISPC_WB
Description	The register configures the resize accumulator init value for vertical up/down-sampling of the wirte-back window It is used for ARGB and Y setting. Shadow register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								VERTICALACCU																							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:0	VERTICALACCU	Vertical initialization accu signed value.	RW	0x0

Table 8-116. Register Call Summary for Register DISPC_WB_ACCUV_j

Display Controller
- DISPC WB Scaler Unit: [0]
- DISPC_WB Register Summary: [1]

Table 8-117. DISPC_WB_ACCUV2_j

Address Offset	0x0000 0018		
Physical Address	0x5801 5018 + (0x4 * j)	**Instance**	DISPC_WB
Description	The register configures the resize accumulator init value for vertical up/down-sampling of the write-back window It is used for Cb and Cr setting. It is used only when the pixel format at the input of the filter is one of the YUV formats. If the pixel format at the input of the filter is ARGB (all ARGB, RGB, RGBA are converted to ARGB32-8888 by the color space conversion before going to the filter is the color space conversion is done before the filter). When the register is not used by the HW, any value can be used for the bit-fields. Shadow register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								VERTICALACCU																							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:0	VERTICALACCU	Vertical initialization accu signed value.	RW	0x0

Table 8-118. Register Call Summary for Register DISPC_WB_ACCUV2_j

Display Controller

- DISPC WB Scaler Unit: [0]
- DISPC_WB Register Summary: [1]

Table 8-119. DISPC_WB_ATTRIBUTES

Address Offset	0x0000 0020		
Physical Address	0x5801 5020	**Instance**	DISPC_WB
Description	The register configures the attributes of the viwrite back pipeline. Shadow register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IDLENUMBER				IDLESIZE	CAPTUREMODE			ARBITRATION	DOUBLESTRIDE	VERTICALTAPS	RESERVED	WRITEBACKMODE	CHANNELIN			RESERVED			FULLRANGE	COLORCONVENABLE	RESERVED	ALPHAENABLE	RESIZEENABLE	FORMAT							ENABLE

Bits	Field Name	Description	Type	Reset
31:28	IDLENUMBER	Determines the number of idles between requests on the L3 interconnect. It is only used when the write-back pipeline does data transfer from memory to memory. When the output of an overlay is stored in memory through the write-back pipeline in capture mode, the bit-field IDLENUMBER is ignored since a timing generator is used to time the transfer. The number of IDLE cycles is IDLENUMBER (from 0 to 15) if IDLESIZE=0. The number of IDLE cycles is IDLENUMBERx8 (from 0 to 120) if IDLESIZE=1 and BURSTSIZE=2. The number of IDLE cycles is IDLENUMBERx4 (from 0 to 60) if IDLESIZE=1 and BURSTSIZE=1. The number of IDLE cycles is IDLENUMBERx2 (from 0 to 30) if IDLESIZE=1 and BURSTSIZE=0.	RW	0x0

Bits	Field Name	Description	Type	Reset
27	IDLESIZE	Determines if the IDLENUMBER corresponds to a number of bursts or singles. 0x0: The number of idles between requests is defined by IDLENUMBER as number of cycles. 0x1: The number of idles between requests is defined by IDLENUMBER multiplied by burst size as number of cycles.	RW	0x0
26:24	CAPTUREMODE	Defines the frame rate capture. 0x6: Only one out of six frames is captured. The first one is captured then the second one is skipped and so on. 0x1: Only one frame is captured. 0x7: Only one out of seven frames is captured. The first one is captured then the second one is skipped and so on. 0x0: All frames are captures until the write-back channel is disabled or there is no more data generated by the overlay or the pipeline attached to the write-back channel. 0x2: Only one out of two frames is captured. The first one is captured then the second one is skipped and so on. 0x4: Only one out of four frames is captured. The first one is captured then the second one is skipped and so on. 0x5: Only one out of five frames is captured. The first one is captured then the second one is skipped and so on. 0x3: Only one out of three frames is captured. The first one is captured then the second one is skipped and so on.	RW	0x0
23	ARBITRATION	Determines the priority of the write-back pipeline. The write-back pipeline is one of the high priority pipeline. The arbitration gives always the priority first to the high priority pipelines using round-robin between them. When there is only normal priority pipelines sending requests, the round-robin applies between them. 0x0: The write-back pipeline is one of the normal priority pipeline. 0x1: The write-back pipeline is one of the high priority pipeline.	RW	0x0
22	DOUBLESTRIDE	Determines if the stride for CbCr buffer is the 1x or 2x of the Y buffer stride. It is only used in case of YUV420 and 2D access 0x0: The CbCr stride value is equal to the Y stride. 0x1: The CbCr stride value is double to the Y stride.	RW	0x0
21	VERTICALTAPS	Video Vertical Resize Tap Number 0x0: 3 taps are used for the vertical filtering logic. The 2 other taps are not used. 0x1: 5 taps are used for the vertical filtering logic.	RW	0x0
20	RESERVED		R	0x0
19	WRITEBACKMODE	When connected to the overlay output of a channel the write back can operate as a simple transfer from memory to memory (composition engine) or as a capture channel. 0x0: Capture mode (default mode 0x1: Memory to memory mode	RW	0x0

Bits	Field Name	Description	Type	Reset
18:15	CHANNELIN	WB Channel In configuration wr: immediate 0x0: OVR1 (VP1) 0x1: OVR2 0x4: GFX1SEL 0x7: VID1SEL 0x8: VID2SEL	RW	0x0
14:13	RESERVED		R	0x0
12	FULLRANGE	Color Space Conversion full range setting. 0x0: Limited range selected: 16 subtracted from Y before color space conversion 0x1: Full range selected: Y is not modified before the color space conversion	RW	0x0
11	COLORCONVENABLE	Enable the color space conversion. The HW does not enable/disable the conversion based on the pixel format. The bit-field shall be reset when the format is not YUV. 0x0: Disable Color Space Conversion RGB to YUV 0x1: Enable Color Space Conversion RGB to YUV	RW	0x0
10	RESERVED		R	0x0
9	ALPHAENABLE	Premultiplied alpha enable 0x0: disabled.Disabled. This bit also disable the logic present in the associated channel out that compute the alpha component sent to the WB pipe 0x1: enabled.	RW	0x0
8:7	RESIZEENABLE	Resize Enable 0x0: Disable the resize processing 0x1: Enable the horizontal resize processing 0x3: Enable both horizontal and vertical resize processing 0x2: Enable the vertical resize processing	RW	0x0

Bits	Field Name	Description	Type	Reset
6:1	FORMAT	Write-back Format. It defines the pixel format when storing the write-back picture into memory. 0x9: RGBA32-8888 0x8: ABGR32-8888 0x5: ARGB16-1555 0x2: RGBA16-4444 0xB: RGB24-888 (24-bit container) 0x4: BGR16-565 0x32: RESERVED4 0x23: RESERVED2 0x1: ABGR16-4444 0x3F: UYVY 4:2:2 co-sited 0x28: xBGR32_8888 0x0: ARGB16-4444 0x3E: YUV2 4:2:2 co-sited 0x25: xRGB16-1555 0x3D: NV12/NV21 4:2:0 2 buffers (Y + UV) 0x20: xRGB12-4444 0x3: RGB16-565 0x17: BGR565A8 0x30: xRGB64-16161616 0x11: RGBA64_16161616 0x6: ABGR16-1555 0x31: RGBX64_16161616 0xA: BGRA32-8888 0x26: xBGR16-1555 0x10: ARGB64-16161616 0x21: xBGR16-4444 0x12: RESERVED5 0x2B: RESERVED3 0x18: RESERVED1 0x16: RGB565A8 0x2A: BGRX32_8888 0x7: ARGB32-8888 0x27: xRGB32-8888 (32-bit container) 0x29: RGBx32-8888 (24-bit RGB aligned on MSB of the 32-bit container) 0xC: RESERVED 0x22: RGBx16-4444	RW	0x0
0	ENABLE	Write-back Enable. wr: immediate 0x0: Write-back disabled 0x1: Write-back enabled	RW	0x0

Table 8-120. Register Call Summary for Register DISPC_WB_ATTRIBUTES

Display Controller

- DISPC DMA Buffers: [0][1][2][3]
- DISPC DMA Arbitration: [4]
- DISPC Write-Back Pipeline: [5][6][7]
- DISPC WB CSC Unit RGB to YUV: [8][9]
- DISPC WB Scaler Unit: [10][11][12]
- DISPC Overlay Alpha Blender: [13][14]
- DISPC_WB Register Summary: [15]

Table 8-121. DISPC_WB_ATTRIBUTES2

Address Offset	0x0000 0024		
Physical Address	0x5801 5024	**Instance**	DISPC_WB
Description	The register set the WB pipe. Shadow register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	TAGS					RESERVED	REGION_BASED	RESERVED						RESERVED	SECURE	RESERVED															FSC

Bits	Field Name	Description	Type	Reset
31	RESERVED		R	0x0
30:26	TAGS	Number of OCP TAGS to be used for the pipeline (from 1 to 32)	RW	0x1f
25	RESERVED		R	0x0
24	REGION_BASED	Enable region-based mechanism for WBPipe 0x0: DISABLE 0x1: ENABLE	RW	0x0
23:18	RESERVED		R	0x0
17	RESERVED		R	0x0
16	SECURE	OCP requests corresponds to pipeline data are secure/unsecure. The bit-field can be modified only by secure transaction using MReqSecure qualifier.	RW	0x0
15:1	RESERVED		R	0x0
0	FSC	Field Sequential Color generation : R,G, and B buffers are generated from RGB buffer. 0x0: DISABLE 0x1: ENABLE	RW	0x0

Table 8-122. Register Call Summary for Register DISPC_WB_ATTRIBUTES2

Display Controller

- Region-Based Mechanism Overview: [0]
- Region-Based Mechanism for a Single Region Write-Back: [1][2][3][4]
- DISPC_WB Register Summary: [5]

Table 8-123. DISPC_WB_BA_j

Address Offset	0x0000 0028		
Physical Address	0x5801 5028 + (0x4 * j)	**Instance**	DISPC_WB
Description	The register configures the base address of the WB buffer (DISPC_WB_BA__0 DISPC_WB_BA__1 for ping-pong mechanism with external trigger, based on the field polarity otherwise only DISPC_WB_BA__0 is used). Shadow register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BA																															

Bits	Field Name	Description	Type	Reset
31:0	BA	Write-back base address Base address of the WB buffer (aligned on pixel size boundary except in case of RGB24 packed format, 4-pixel alignment is required; in case of YUV422, 2-pixel alignment is required, and YUV420, byte alignment is supported)). It case of YUV 4:2:0 format, it indicates the base address of the Y buffer. When the TILER is addressed, the bits: [28:27] = 0x0 for 8-bit tiled [28:27] = 0x1 for 16-bit tiled [28:27] = 0x2 for 32-bit tiled [28:27] = 0x3 for page mode [31:29] = 0x0 for 0-degree view [31:29] = 0x1 for 180-degree view + mirroring [31:29] = 0x2 for 0-degree view + mirroring [31:29] = 0x3 for 180-degree view [31:29] = 0x4 for 270-degree view + mirroring [31:29] = 0x5 for 270-degree view [31:29] = 0x6 for 90-degree view [31:29] = 0x7 for 90-degree view + mirroring Otherwise the bits indicated the corresponding bit address to access the SDRAM.	RW	0x0

Table 8-124. Register Call Summary for Register DISPC_WB_BA_j

Display Controller

- DISPC DMA Addressing and Bursts: [0]
- DISPC_WB Register Summary: [1]

Table 8-125. DISPC_WB_BA_UV_j

Address Offset	0x0000 0030		
Physical Address	0x5801 5030 + (0x4 * j)	**Instance**	DISPC_WB
Description	The register configures the base address of the UV buffer for the write-back pipeline. (DISPC_WB_BA_UV_0 and DISPC_WB_BA_UV_1 for ping-pong mechanism with external trigger, based on the field polarity, otherwise only DISPC_WB_BA_UV_0 is used)). The register is also used to configure the RGB plane BA for RGB565A8 format. Shadow register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BA																															

Bits	Field Name	Description	Type	Reset
31:0	BA	Video base address alinged on 16-bit boundary Base address of the UV video buffer used only in case of YUV420-NV12 When the TILER is addressed, the bits: [28:27] = 0x0 for 8-bit tiled [28:27] = 0x1 for 16-bit tiled [28:27] = 0x2 for 32-bit tiled [28:27] = 0x3 for page mode [31:29] = 0x0 for 0-degree view [31:29] = 0x1 for 180-degree view + mirroring [31:29] = 0x2 for 0-degree view + mirroring [31:29] = 0x3 for 180-degree view [31:29] = 0x4 for 270-degree view + mirroring [31:29] = 0x5 for 270-degree view [31:29] = 0x6 for 90-degree view [31:29] = 0x7 for 90-degree view + mirroring Otherwise the bits indicated the corresponding bit address to access the SDRAM.	RW	0x0

Table 8-126. Register Call Summary for Register DISPC_WB_BA_UV_j

Display Controller

- DISPC DMA Addressing and Bursts: [0]
- DISPC_WB Register Summary: [1]

Table 8-127. DISPC_WB_BUF_SIZE_STATUS

Address Offset	0x0000 0038		
Physical Address	0x5801 5038	**Instance**	DISPC_WB
Description	The register defines the DMA buufer size for the write back pipeline.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																BUFSIZE															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Write 0's for future compatibility. Reads return 0	R	0x0
15:0	BUFSIZE	DMA buffer Size in number of 128-bits.	R	0xa00

Table 8-128. Register Call Summary for Register DISPC_WB_BUF_SIZE_STATUS

Display Controller

- DISPC_WB Register Summary: [0]

Table 8-129. DISPC_WB_BUF_THRESHOLD

Address Offset	0x0000 003C		
Physical Address	0x5801 503C	**Instance**	DISPC_WB
Description	The register configures the DMA buffer associated with the write-back pipeline. Shadow register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BUFHIGHTHRESHOLD																BUFLOWTHRESHOLD															

Bits	Field Name	Description	Type	Reset
31:16	BUFHIGHTHRESHOLD	DMA buffer High Threshold Number of 128-bits defining the threshold value.	RW	0x9ff
15:0	BUFLOWTHRESHOLD	DMA buffer High Threshold Number of 128-bits defining the threshold value.	RW	0x9f8

Table 8-130. Register Call Summary for Register DISPC_WB_BUF_THRESHOLD

Display Controller

- DISPC DMA Buffers: [0][1]
- DISPC_WB Register Summary: [2]

Table 8-131. DISPC_WB_CONV_COEF0

Address Offset	0x0000 0040		
Physical Address	0x5801 5040	**Instance**	DISPC_WB
Description	The register configures the color space conversion matrix coefficients for the write back pipeline (YUV444 to RGB24) Shadow register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								YG								RESERVED								YR							

Bits	Field Name	Description	Type	Reset
31:27	RESERVED	Write 0's for future compatibility. Reads return 0	R	0x0
26:16	YG	YG Coefficient Encoded signed value (from -1024 to 1023).	RW	0x0
15:11	RESERVED	Write 0's for future compatibility. Reads return 0	R	0x0
10:0	YR	YR Coefficient Encoded signed value (from -1024 to 1023).	RW	0x0

Table 8-132. Register Call Summary for Register DISPC_WB_CONV_COEF0

Display Controller

- DISPC WB CSC Unit RGB to YUV: [0][1]
- DISPC_WB Register Summary: [2]

Table 8-133. DISPC_WB_CONV_COEF1

Address Offset	0x0000 0044		
Physical Address	0x5801 5044	**Instance**	DISPC_WB
Description	The register configures the color space conversion matrix coefficients for the write back pipeline. Shadow register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								CRR								RESERVED								YB							

Bits	Field Name	Description	Type	Reset
31:27	RESERVED	Write 0's for future compatibility. Reads return 0	R	0x0
26:16	CRR	CrR Coefficient Encoded signed value (from -1024 to 1023).	RW	0x0
15:11	RESERVED	Write 0's for future compatibility. Reads return 0	R	0x0
10:0	YB	YB Coefficient Encoded signed value (from -1024 to 1023).	RW	0x0

Table 8-134. Register Call Summary for Register DISPC_WB_CONV_COEF1

Display Controller

- DISPC WB CSC Unit RGB to YUV: [0][1]
- DISPC_WB Register Summary: [2]

Table 8-135. DISPC_WB_CONV_COEF2

Address Offset	0x0000 0048		
Physical Address	0x5801 5048	**Instance**	DISPC_WB
Description	The register configures the color space conversion matrix coefficients for the write back pipeline. Shadow register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								CRB								RESERVED								CRG							

Bits	Field Name	Description	Type	Reset
31:27	RESERVED	Write 0's for future compatibility. Reads return 0	R	0x0
26:16	CRB	CrB Coefficient Encoded signed value (from -1024 to 1023).	RW	0x0
15:11	RESERVED	Write 0's for future compatibility. Reads return 0	R	0x0
10:0	CRG	CrG Coefficient Encoded signed value (from -1024 to 1023).	RW	0x0

Table 8-136. Register Call Summary for Register DISPC_WB_CONV_COEF2

Display Controller
- DISPC WB CSC Unit RGB to YUV: [0][1]
- DISPC_WB Register Summary: [2]

Table 8-137. DISPC_WB_CONV_COEF3

Address Offset	0x0000 004C		
Physical Address	0x5801 504C	**Instance**	DISPC_WB
Description	The register configures the color space conversion matrix coefficients for the write back pipeline. Shadow register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								CBG								RESERVED								CBR							

Bits	Field Name	Description	Type	Reset
31:27	RESERVED	Write 0's for future compatibility. Reads return 0	R	0x0
26:16	CBG	CbG coefficient Encoded signed value (from -1024 to 1023).	RW	0x0
15:11	RESERVED	Write 0's for future compatibility. Reads return 0	R	0x0
10:0	CBR	CbR coefficient Encoded signed value (from -1024 to 1023).	RW	0x0

Table 8-138. Register Call Summary for Register DISPC_WB_CONV_COEF3

Display Controller
- DISPC WB CSC Unit RGB to YUV: [0][1]
- DISPC_WB Register Summary: [2]

Table 8-139. DISPC_WB_CONV_COEF4

Address Offset	0x0000 0050		
Physical Address	0x5801 5050	**Instance**	DISPC_WB
Description	The register configures the color space conversion matrix coefficients for the write back pipeline. Shadow register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																					CBB										

Bits	Field Name	Description	Type	Reset
31:11	RESERVED		R	0x0
10:0	CBB	CbB Coefficient Encoded signed value (from -1024 to 1023).	RW	0x0

Table 8-140. Register Call Summary for Register DISPC_WB_CONV_COEF4

Display Controller

- DISPC WB CSC Unit RGB to YUV: [0]
- DISPC_WB Register Summary: [1]

Table 8-141. DISPC_WB_CONV_COEF5

Address Offset	0x0000 0054		
Physical Address	0x5801 5054	**Instance**	DISPC_WB
Description	The register configures the color space conversion matrix coefficients for the write-back pipeline. Shadow register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GOFFSET													RESERVED			ROFFSET													RESERVED		

Bits	Field Name	Description	Type	Reset
31:19	GOFFSET	G offset Encoded signed value (from -4096 to 4095).	RW	0x0
18:16	RESERVED		R	0x0
15:3	ROFFSET	R offset Encoded signed value (from -4096 to 4095).	RW	0x0
2:0	RESERVED		R	0x0

Table 8-142. Register Call Summary for Register DISPC_WB_CONV_COEF5

Display Controller

- DISPC WB CSC Unit RGB to YUV: [0][1]
- DISPC_WB Register Summary: [2]

Table 8-143. DISPC_WB_CONV_COEF6

Address Offset	0x0000 0058		
Physical Address	0x5801 5058	**Instance**	DISPC_WB
Description	The register configures the color space conversion matrix coefficients for the write-back pipeline. Shadow register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																BOFFSET														RESERVED	

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0
15:3	BOFFSET	B offset Encoded signed value (from -4096 to 4095).	RW	0x0
2:0	RESERVED		R	0x0

Table 8-144. Register Call Summary for Register DISPC_WB_CONV_COEF6

Display Controller

- DISPC WB CSC Unit RGB to YUV: [0]
- DISPC_WB Register Summary: [1]

Table 8-145. DISPC_WB_FIRH

Address Offset	0x0000 005C		
Physical Address	0x5801 505C	**Instance**	DISPC_WB
Description	The register configures the resize factor for horizontal up/down-sampling of the write-back window. It is used for ARGB and Y setting. Shadow register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								FIRHINC																							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:0	FIRHINC	Horizontal increment of the up/down-sampling filter. The value 0 is invalid.	RW	0x200000

Table 8-146. Register Call Summary for Register DISPC_WB_FIRH

Display Controller

- DISPC WB Scaler Unit: [0]
- DISPC_WB Register Summary: [1]

Table 8-147. DISPC_WB_FIRH2

Address Offset	0x0000 0060		
Physical Address	0x5801 5060	**Instance**	DISPC_WB
Description	The register configures the resize factor for horizontal up/down-sampling of the wirte-back window. It is used for Crb and Cr setting. It is used only when the pixel format at the input of the filter is one of the YUV formats. If the pixel format at the input of the filter is ARGB (all ARGB, RGB, RGBA are converted to ARGB32-8888 by the color space conversion before going to the filter is the color space conversion is done before the filter). When the register is not used by the HW, any value can be used for the bit-fields. Shadow register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								FIRHINC																							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:0	FIRHINC	Horizontal increment of the up/down-sampling filter for Cb and Cr. The value 0 is invalid.	RW	0x200000

Table 8-148. Register Call Summary for Register DISPC_WB_FIRH2

Display Controller

- DISPC WB Scaler Unit: [0]
- DISPC_WB Register Summary: [1]

Table 8-149. DISPC_WB_FIRV

Address Offset	0x0000 0064		
Physical Address	0x5801 5064	**Instance**	DISPC_WB
Description	The register configures the resize factor for vertical up/down-sampling of the write-back window It is used for ARGB and Y setting. Shadow register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								FIRVINC																							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:0	FIRVINC	Vertical increment of the up/down-sampling filter. The value 0 is invalid.	RW	0x200000

Table 8-150. Register Call Summary for Register DISPC_WB_FIRV

Display Controller

- DISPC WB Scaler Unit: [0]
- DISPC_WB Register Summary: [1]

Table 8-151. DISPC_WB_FIRV2

Address Offset	0x0000 0068		
Physical Address	0x5801 5068	**Instance**	DISPC_WB
Description	The register configures the resize factor for vertical up/down-sampling of the wirte-back window. It is used for Crb and Cr setting. It is used only when the pixel format at the input of the filter is one of the YUV formats. If the pixel format at the input of the filter is ARGB (all ARGB, RGB, RGBA are converted to ARGB32-8888 by the color space conversion before going to the filter is the color space conversion is done before the filter). When the register is not used by the HW, any value can be used for the bit-fields. Shadow register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								FIRVINC																							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:0	FIRVINC	Vertical increment of the up/down-sampling filter for Cb and Cr. The value 0 is invalid.	RW	0x200000

Table 8-152. Register Call Summary for Register DISPC_WB_FIRV2

Display Controller

- DISPC WB Scaler Unit: [0]
- DISPC_WB Register Summary: [1]

Table 8-153. DISPC_WB_FIR_COEF_H0_i

Address Offset	0x0000 006C		
Physical Address	0x5801 506C + (0x4 * i)	**Instance**	DISPC_WB
Description	The bank of registers configure the up/down-scaling coefficients for the horizontal resize of the video picture associated with the write back pipeline for the phases from 0 to 15. It is used for ARGB and Y setting. Shadow register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED		RESERVED																						FIRHC0							

Bits	Field Name	Description	Type	Reset
31:30	RESERVED		R	0x0
29:10	RESERVED		R	0x0
9:0	FIRHC0	Unsigned coefficient C0 for the horizontal up/down-scaling with the phase n	RW	0x0

Table 8-154. Register Call Summary for Register DISPC_WB_FIR_COEF_H0_i

Display Controller

- DISPC WB Scaler Unit: [0][1]
- DISPC_WB Register Summary: [2]

Table 8-155. DISPC_WB_FIR_COEF_H0_C_i

Address Offset	0x0000 0090		
Physical Address	0x5801 5090 + (0x4 * i)	**Instance**	DISPC_WB
Description	The bank of registers configure the up/down-scaling coefficients for the horizontal resize of the video picture associated with the write back pipeline for the phases from 0 to 15. It is used for Crb and Cr setting. It is used only when the pixel format at the input of the filter is one of the YUV formats. If the pixel format at the input of the filter is ARGB (all ARGB, RGB, RGBA are converted to ARGB32-8888 by the color space conversion before going to the filter is the color space conversion is done before the filter). When the register is not used by the HW, any value can be used for the bit-fields. Shadow register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								RESERVED																FIRHC0							

Bits	Field Name	Description	Type	Reset
31:30	RESERVED		R	0x0
29:10	RESERVED		R	0x0
9:0	FIRHC0	Unsigned coefficient C0 for the horizontal up/down-scaling with the phase n	RW	0x0

Table 8-156. Register Call Summary for Register DISPC_WB_FIR_COEF_H0_C_i

Display Controller

- DISPC WB Scaler Unit: [0][1]
- DISPC_WB Register Summary: [2]

Table 8-157. DISPC_WB_FIR_COEF_H12_k

Address Offset	0x0000 00B4		
Physical Address	0x5801 50B4 + (0x4 * k)	**Instance**	DISPC_WB
Description	The bank of registers configure the up/down-scaling coefficients for the horizontal resize of the video picture associated with the write back pipeline for the phases from 0 to 15. It is used for ARGB and Y setting. Shadow register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								FIRHC2								FIRHC1								RESERVED							

Bits	Field Name	Description	Type	Reset
31:30	RESERVED		R	0x0
29:20	FIRHC2	Signed coefficient C2 for the horizontal up/down-scaling with the phase n	RW	0x0
19:10	FIRHC1	Signed coefficient C1 for the horizontal up/down-scaling with the phase n	RW	0x0
9:0	RESERVED		R	0x0

Table 8-158. Register Call Summary for Register DISPC_WB_FIR_COEF_H12_k

Display Controller
- DISPC WB Scaler Unit: [0][1]
- DISPC_WB Register Summary: [2]

Table 8-159. DISPC_WB_FIR_COEF_H12_C_k

Address Offset	0x0000 00F4		
Physical Address	0x5801 50F4 + (0x4 * k)	**Instance**	DISPC_WB
Description	The bank of registers configure the up/down-scaling coefficients for the horizontal resize of the video picture associated with the write back pipeline for the phases from 0 to 15. It is used for Crb and Cr setting. It is used only when the pixel format at the input of the filter is one of the YUV formats. If the pixel format at the input of the filter is ARGB (all ARGB, RGB, RGBA are converted to ARGB32-8888 by the color space conversion before going to the filter is the color space conversion is done before the filter). When the register is not used by the HW, any value can be used for the bit-fields. Shadow register.		
Type	RW		

31 30	29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
RESERVED	FIRHC2	FIRHC1	RESERVED	

Bits	Field Name	Description	Type	Reset
31:30	RESERVED		R	0x0
29:20	FIRHC2	Signed coefficient C2 for the horizontal up/down-scaling with the phase n	RW	0x0
19:10	FIRHC1	Signed coefficient C1 for the horizontal up/down-scaling with the phase n	RW	0x0
9:0	RESERVED		R	0x0

Table 8-160. Register Call Summary for Register DISPC_WB_FIR_COEF_H12_C_k

Display Controller
- DISPC WB Scaler Unit: [0][1]
- DISPC_WB Register Summary: [2]

Table 8-161. DISPC_WB_FIR_COEF_V0_i

Address Offset	0x0000 0134		
Physical Address	0x5801 5134 + (0x4 * i)	**Instance**	DISPC_WB
Description	The bank of registers configure the down/up/down-scaling coefficients for the vertical resize of the video picture associated with the write back pipeline for the phases from 0 to 15. It is used for ARGB and Y setting. Shadow register.		
Type	RW		

31 30	29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
RESERVED	RESERVED			FIRVC0

Bits	Field Name	Description	Type	Reset
31:30	RESERVED		R	0x0
29:10	RESERVED		R	0x0
9:0	FIRVC0	Unsigned coefficient C0 for the vertical up/down-scaling with the phase n	RW	0x0

Table 8-162. Register Call Summary for Register DISPC_WB_FIR_COEF_V0_i

Display Controller
- DISPC WB Scaler Unit: [0][1]
- DISPC_WB Register Summary: [2]

Table 8-163. DISPC_WB_FIR_COEF_V0_C_i

Address Offset	0x0000 0158		
Physical Address	0x5801 5158 + (0x4 * i)	**Instance**	DISPC_WB
Description	The bank of registers configure the down/up/down-scaling coefficients for the vertical resize of the video picture associated with the write back pipeline for the phases from 0 to 15. It is used for Crb and Cr setting. It is used only when the pixel format at the input of the filter is one of the YUV formats. If the pixel format at the input of the filter is ARGB (all ARGB, RGB, RGBA are converted to ARGB32-8888 by the color space conversion before going to the filter is the color space conversion is done before the filter). When the register is not used by the HW, any value can be used for the bit-fields. Shadow register.		
Type	RW		

31 30	29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10	9 8	7 6 5 4 3 2 1 0
RESERVED	RESERVED				FIRVC0

Bits	Field Name	Description	Type	Reset
31:30	RESERVED		R	0x0
29:10	RESERVED		R	0x0
9:0	FIRVC0	Unsigned coefficient C0 for the vertical up/down-scaling with the phase n	RW	0x0

Table 8-164. Register Call Summary for Register DISPC_WB_FIR_COEF_V0_C_i

Display Controller
- DISPC WB Scaler Unit: [0][1]
- DISPC_WB Register Summary: [2]

Table 8-165. DISPC_WB_FIR_COEF_V12_k

Address Offset	0x0000 017C		
Physical Address	0x5801 517C + (0x4 * k)	**Instance**	DISPC_WB
Description	The bank of registers configure the down/up/down-scaling coefficients for the vertical resize of the video picture associated with the write back pipeline for the phases from 0 to 15. It is used for ARGB and Y setting. Shadow register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								FIRVC2								FIRVC1								RESERVED							

Bits	Field Name	Description	Type	Reset
31:30	RESERVED		R	0x0
29:20	FIRVC2	Signed coefficient C2 for the vertical up/down-scaling with the phase n	RW	0x0
19:10	FIRVC1	Signed coefficient C1 for the vertical up/down-scaling with the phase n	RW	0x0
9:0	RESERVED		R	0x0

Table 8-166. Register Call Summary for Register DISPC_WB_FIR_COEF_V12_k

Display Controller

- DISPC WB Scaler Unit: [0][1]
- DISPC_WB Register Summary: [2]

Table 8-167. DISPC_WB_FIR_COEF_V12_C_k

Address Offset	0x0000 01BC		
Physical Address	0x5801 51BC + (0x4 * k)	**Instance**	DISPC_WB
Description	The bank of registers configure the down/up/down-scaling coefficients for the vertical resize of the video picture associated with the write back pipeline for the phases from 0 to 15. It is used for Crb and Cr setting. It is used only when the pixel format at the input of the filter is one of the YUV formats. If the pixel format at the input of the filter is ARGB (all ARGB, RGB, RGBA are converted to ARGB32-8888 by the color space conversion before going to the filter is the color space conversion is done before the filter). When the register is not used by the HW, any value can be used for the bit-fields. Shadow register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								FIRVC2								FIRVC1								RESERVED							

Bits	Field Name	Description	Type	Reset
31:30	RESERVED		R	0x0
29:20	FIRVC2	Signed coefficient C2 for the vertical up/down-scaling with the phase n	RW	0x0
19:10	FIRVC1	Signed coefficient C1 for the vertical up/down-scaling with the phase n	RW	0x0
9:0	RESERVED		R	0x0

Table 8-168. Register Call Summary for Register DISPC_WB_FIR_COEF_V12_C_k

Display Controller

- DISPC WB Scaler Unit: [0][1]
- DISPC_WB Register Summary: [2]

Table 8-169. DISPC_WB_IRQENABLE

Address Offset	0x0000 01FC		
Physical Address	0x5801 51FC	**Instance**	DISPC_WB
Description	This register allows to mask/unmask the module internal sources of interrupt, on an event-by-event basis		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																											WBSYNC_EN	WBREGIONBASEDEVENT_EN	WBFRAMEDONE_EN	WBUNCOMPLETEERROR_EN	WBBUFFEROVERFLOW_EN

Bits	Field Name	Description	Type	Reset
31:5	RESERVED		R	0x0
4	WBSYNC_EN	Write-back synced (configuration copied from shadow to work) 0x0: WBBufferOverflow is masked 0x1: WBBufferOverflow generates an interrupt when it occurs	RW	0x0
3	WBREGIONBASEDEVENT_EN	For Write-back region-based, event indicating end of current window 0x0: WBBufferOverflow is masked 0x1: WBBufferOverflow generates an interrupt when it occurs	RW	0x0
2	WBFRAMEDONE_EN	Write-back Frame Done. 0x0: WBBufferOverflow is masked 0x1: WBBufferOverflow generates an interrupt when it occurs	RW	0x0
1	WBUNCOMPLETEERROR_EN	The write back buffer has been flushed before been fully drained. Enable 0x0: Interrupt is masked 0x1: Interrupt is enabled	RW	0x0
0	WBBUFFEROVERFLOW_EN	Write-back DMA Buffer Overflow. The DMA buffer is full 0x0: WBBufferOverflow is masked 0x1: WBBufferOverflow generates an interrupt when it occurs	RW	0x0

Table 8-170. Register Call Summary for Register DISPC_WB_IRQENABLE

Display Controller

- DISPC Interrupt Requests: [0]
- Region-Based Mechanism for a Single Region Write-Back: [1][2]
- DISPC_WB Register Summary: [3]

Table 8-171. DISPC_WB_IRQSTATUS

Address Offset	0x0000 0200		
Physical Address	0x5801 5200	**Instance**	DISPC_WB
Description	This register regroups all the status of the module internal events that generate an interrupt. Write 1 to a given bit resets this bit		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																											WBSYNC_IRQ	WBREGIONBASEDEVENT_IRQ	WBFRAMEDONE_IRQ	WBUNCOMPLETEERROR_IRQ	WBBUFFEROVERFLOW_IRQ

Bits	Field Name	Description	Type	Reset
31:5	RESERVED		R	0x0
4	WBSYNC_IRQ	Write-back synced (configuration copied from shadow to work) 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW	0x0
3	WBREGIONBASEDEVENT_IRQ	Write-back Frame Done. 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW	0x0
2	WBFRAMEDONE_IRQ	Write-back Frame Done. 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW	0x0
1	WBUNCOMPLETEERROR_IRQ	Write back DMA buffer is flushed before been completely drained. 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (Pending) WRITE: Status bit is reset	RW	0x0
0	WBBUFFEROVERFLOW_IRQ	Write-back DMA Buffer Overrflow. The DMA buffer is full. 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW	0x0

Table 8-172. Register Call Summary for Register DISPC_WB_IRQSTATUS

Display Controller

- DISPC Interrupt Requests: [0]
- DISPC DMA Buffers: [1][2]
- DISPC_WB Register Summary: [3]

Table 8-173. DISPC_WB_MFLAG_THRESHOLD

Address Offset	0x0000 0204		
Physical Address	0x5801 5204	**Instance**	DISPC_WB
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HT_MFLAG																LT_MFLAG															

Bits	Field Name	Description	Type	Reset
31:16	HT_MFLAG		RW	0x0
15:0	LT_MFLAG		RW	0x0

Table 8-174. Register Call Summary for Register DISPC_WB_MFLAG_THRESHOLD

Display Controller

- DISPC DMA MFLAG Mechanism and Arbitration: [0][1]
- DISPC_WB Register Summary: [2]

Table 8-175. DISPC_WB_PICTURE_SIZE

Address Offset	0x0000 0208		
Physical Address	0x5801 5208	**Instance**	DISPC_WB
Description	The register configures the size of the write-back picture associated with the write back pipeline after up/down-scaling (size of the image stored in DDR memory, generated by WB pipe). Shadow register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				MEMSIZEY												RESERVED				MEMSIZEX											

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:16	MEMSIZEY	Number of lines of the wb picture in memory Encoded value (from 1 to 4096) to specify the number of lines of the picture store in memory (program to value minus one).	RW	0x0
15:12	RESERVED		R	0x0
11:0	MEMSIZEX	Number of pixels of the wb picture in memory. Encoded value (from 1 to 4096) to specify the number of pixels of the picture stored in memory (program to value minus one). In case of 32-bit RGB/ARGB/RGBA pixel formats and 2D burst, the maximum width is limited to 2048 pixels.	RW	0x0

Table 8-176. Register Call Summary for Register DISPC_WB_PICTURE_SIZE

Display Controller

- DISPC WB Scaler Unit:
- Region-Based Mechanism for a Single Region Write-Back: [2][3]
- DISPC_WB Register Summary: [4]

Table 8-177. DISPC_WB_SIZE

Address Offset	0x0000 0210		
Physical Address	0x5801 5210	**Instance**	DISPC_WB
Description	The register configures the size of the output of overlay connected to the write-back pipeline when the overlay output is only used by the write-back pipeline. When the overlay is output on the VP1 output, the size of the frame is defined in the DISPC_VP1_SIZE_SCREEN register. Shadow register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								SIZEY								RESERVED								SIZEX							

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:16	SIZEY	Number of lines of the Write-back picture Encoded value (from 1 to 4096) to specify the number of lines of the write-back picture from overlay or pipeline.	RW	0x0
15:12	RESERVED		R	0x0
11:0	SIZEX	Number of pixels of the Write-back picture Encoded value (from 1 to 4096) to specify the number of pixels of the write-back picture from overlay or pipeline..	RW	0x0

Table 8-178. Register Call Summary for Register DISPC_WB_SIZE

Display Controller

- DISPC WB Scaler Unit:
- Region-Based Mechanism for a Single Region Write-Back: [5]
- DISPC_WB Register Summary: [6]

Table 8-179. DISPC_WB_POSITION

Address Offset	0x0000 0214		
Physical Address	0x5801 5214	**Instance**	DISPC_WB
Description	The register configures the start position of the window on overlay which wb will capture Shadow register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								POSY								RESERVED								POSX							

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:16	POSY	Y position of the video window Encoded value (from 0 to 4095) to specify the Y position of the video window #1 .The line at the top has the Y-position 0.	RW	0x0
15:12	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
11:0	POSX	X position of the video window Encoded value (from 0 to 4095) to specify the X position of the video window #1. The first pixel on the left of the display screen has the X-position 0.	RW	0x0

Table 8-180. Register Call Summary for Register DISPC_WB_POSITION

Display Controller
- Region-Based Mechanism for a Single Region Write-Back: [0]
- DISPC_WB Register Summary: [1]

Table 8-181. DISPC_WB_REGION_BASED_TOTAL_PICTURE_SIZEY

Address Offset	0x0000 0218		
Physical Address	0x5801 5218	**Instance**	DISPC_WB
Description	The register configures the total sizey (total number of lines) of all the windows written back to the memory in Write-Back Region-based mode. Shadow register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																				SIZEY											

Bits	Field Name	Description	Type	Reset
31:12	RESERVED		R	0x0
11:0	SIZEY	Total number of lines of the Write-back picture in Region-based mode Encoded value (from 1 to 4096) to specify the number of lines of the write-back picture from overlay or pipeline.	RW	0x0

Table 8-182. Register Call Summary for Register DISPC_WB_REGION_BASED_TOTAL_PICTURE_SIZEY

Display Controller
- Region-Based Mechanism Overview: [0]
- Region-Based Mechanism for a Single Region Write-Back: [1][2]
- DISPC_WB Register Summary: [3]

8.2.5.5 DISPC_VID Registers

8.2.5.5.1 DISPC_VID Register Summary

Table 8-183. DISPC_VID Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	DISPC_VID1 L3_MAIN Physical Address	DISPC_VID2 L3_MAIN Physical Address
DISPC_VID_ACCUH_j[(1)]	RW	32	0x0000 0000 + (0x4 * j)	0x5801 7000 + (0x4 * j)	0x5801 8000 + (0x4 * j)
DISPC_VID_ACCUH2_j [(1)]	RW	32	0x0000 0008 + (0x4 * j)	0x5801 7008 + (0x4 * j)	0x5801 8008 + (0x4 * j)
DISPC_VID_ACCUV_j [(1)]	RW	32	0x0000 0010 + (0x4 * j)	0x5801 7010 + (0x4 * j)	0x5801 8010 + (0x4 * j)
DISPC_VID_ACCUV2_j [(1)]	RW	32	0x0000 0018 + (0x4 * j)	0x5801 7018 + (0x4 * j)	0x5801 8018 + (0x4 * j)
DISPC_VID_ATTRIBUTES	RW	32	0x0000 0020	0x5801 7020	0x5801 8020
DISPC_VID_ATTRIBUTES2	RW	32	0x0000 0024	0x5801 7024	0x5801 8024
DISPC_VID_BA_j [(1)]	RW	32	0x0000 0028 + (0x4 * j)	0x5801 7028 + (0x4 * j)	0x5801 8028 + (0x4 * j)

(1) j = 0 to 1

Table 8-183. DISPC_VID Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	DISPC_VID1 L3_MAIN Physical Address	DISPC_VID2 L3_MAIN Physical Address
DISPC_VID_BA_UV_j (1)	RW	32	0x0000 0030 + (0x4 * j)	0x5801 7030 + (0x4 * j)	0x5801 8030 + (0x4 * j)
DISPC_VID_BUF_SIZE_STATUS	R	32	0x0000 0038	0x5801 7038	0x5801 8038
DISPC_VID_BUF_THRESHOLD	RW	32	0x0000 003C	0x5801 703C	0x5801 803C
DISPC_VID_CONV_COEF0	RW	32	0x0000 0040	0x5801 7040	0x5801 8040
DISPC_VID_CONV_COEF1	RW	32	0x0000 0044	0x5801 7044	0x5801 8044
DISPC_VID_CONV_COEF2	RW	32	0x0000 0048	0x5801 7048	0x5801 8048
DISPC_VID_CONV_COEF3	RW	32	0x0000 004C	0x5801 704C	0x5801 804C
DISPC_VID_CONV_COEF4	RW	32	0x0000 0050	0x5801 7050	0x5801 8050
DISPC_VID_CONV_COEF5	RW	32	0x0000 0054	0x5801 7054	0x5801 8054
DISPC_VID_CONV_COEF6	RW	32	0x0000 0058	0x5801 7058	0x5801 8058
DISPC_VID_FIRH	RW	32	0x0000 005C	0x5801 705C	0x5801 805C
DISPC_VID_FIRH2	RW	32	0x0000 0060	0x5801 7060	0x5801 8060
DISPC_VID_FIRV	RW	32	0x0000 0064	0x5801 7064	0x5801 8064
DISPC_VID_FIRV2	RW	32	0x0000 0068	0x5801 7068	0x5801 8068
DISPC_VID_FIR_COEF_H0_i (2)	RW	32	0x0000 006C + (0x4 * i)	0x5801 706C + (0x4 * i)	0x5801 806C + (0x4 * i)
DISPC_VID_FIR_COEF_H0_C_i (2)	RW	32	0x0000 0090 + (0x4 * i)	0x5801 7090 + (0x4 * i)	0x5801 8090 + (0x4 * i)
DISPC_VID_FIR_COEF_H12_k (3)	RW	32	0x0000 00B4 + (0x4 * k)	0x5801 70B4 + (0x4 * k)	0x5801 80B4 + (0x4 * k)
DISPC_VID_FIR_COEF_H12_C_k (3)	RW	32	0x0000 00F4 + (0x4 * k)	0x5801 70F4 + (0x4 * k)	0x5801 80F4 + (0x4 * k)
DISPC_VID_FIR_COEF_V0_i (2)	RW	32	0x0000 0134 + (0x4 * i)	0x5801 7134 + (0x4 * i)	0x5801 8134 + (0x4 * i)
DISPC_VID_FIR_COEF_V0_C_i (2)	RW	32	0x0000 0158 + (0x4 * i)	0x5801 7158 + (0x4 * i)	0x5801 8158 + (0x4 * i)
DISPC_VID_FIR_COEF_V12_k (3)	RW	32	0x0000 017C + (0x4 * k)	0x5801 717C + (0x4 * k)	0x5801 817C + (0x4 * k)
DISPC_VID_FIR_COEF_V12_C_k (3)	RW	32	0x0000 01BC + (0x4 * k)	0x5801 71BC + (0x4 * k)	0x5801 81BC + (0x4 * k)
DISPC_VID_GLOBAL_ALPHA	RW	32	0x0000 01FC	0x5801 71FC	0x5801 81FC
DISPC_VID_IRQENABLE	RW	32	0x0000 0200	0x5801 7200	0x5801 8200
DISPC_VID_IRQSTATUS	RW	32	0x0000 0204	0x5801 7204	0x5801 8204
DISPC_VID_MFLAG_THRESHOLD	RW	32	0x0000 0208	0x5801 7208	0x5801 8208
DISPC_VID_PICTURE_SIZE	RW	32	0x0000 020C	0x5801 720C	0x5801 820C
DISPC_VID_PIXEL_INC	RW	32	0x0000 0210	0x5801 7210	0x5801 8210
DISPC_VID_POSITION	RW	32	0x0000 0214	0x5801 7214	0x5801 8214
DISPC_VID_PRELOAD	RW	32	0x0000 0218	0x5801 7218	0x5801 8218
DISPC_VID_ROW_INC	RW	32	0x0000 021C	0x5801 721C	0x5801 821C
DISPC_VID_SIZE	RW	32	0x0000 0220	0x5801 7220	0x5801 8220
DISPC_VID_CLUT	W	32	0x0000 0224	0x5801 7224	0x5801 8224

(2) i = 0 to 8
(3) k = 0 to 15

8.2.5.5.2 DISPC_VID Register Description

Table 8-184. DISPC_VID_ACCUH_j

Address Offset	0x0000 0000		
Physical Address	0x5801 7000 + (0x4 * j) 0x5801 8000 + (0x4 * j)	**Instance**	DISPC_VID1 DISPC_VID2
Description	The register configures the resize accumulator init values for horizontal up/down-sampling of the video window (DISPC_VIDx_ACCU__0 DISPC_VIDx_ACCU__1 for ping-pong mechanism with external trigger, based on the field polarity) It is used for ARGB and Y setting. Shadow register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								HORIZONTALACCU																							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:0	HORIZONTALACCU	Horizontal initialization accu signed value.	RW	0x0

Table 8-185. Register Call Summary for Register DISPC_VID_ACCUH_j

Display Controller
- DISPC VID Scaler Unit: [0]
- DISPC_VID Register Summary: [1]

Table 8-186. DISPC_VID_ACCUH2_j

Address Offset	0x0000 0008		
Physical Address	0x5801 7008 + (0x4 * j) 0x5801 8008 + (0x4 * j)	**Instance**	DISPC_VID1 DISPC_VID2
Description	The register configures the resize accumulator init value for horizontal up/down-sampling of the video window (DISPC_VID#n_ACCU2__0 DISPC_VID#n_ACCU2__1 for ping-pong mechanism with external trigger, based on the field polarity) It is used for Cb and Cr setting. It is used only when the pixel format at the input of the filter is one of the YUV formats. If the pixel format at the input of the filter is ARGB (all ARGB, RGB, RGBA are converted to ARGB32-8888 by the color space conversion before going to the filter is the color space conversion is done before the filter). When the register is not used by the HW, any value can be used for the bit-fields. Shadow register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								HORIZONTALACCU																							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:0	HORIZONTALACCU	Horizontal initialization accu signed value	RW	0x0

Table 8-187. Register Call Summary for Register DISPC_VID_ACCUH2_j

Display Controller
- DISPC VID Scaler Unit: [0]
- DISPC_VID Register Summary: [1]

Table 8-188. DISPC_VID_ACCUV_j

Address Offset	0x0000 0010		
Physical Address	0x5801 7010 + (0x4 * j) 0x5801 8010 + (0x4 * j)	**Instance**	DISPC_VID1 DISPC_VID2
Description	The register configures the resize accumulator init values for horizontal and vertical up/down-sampling of the video window (DISPC_VIDx_ACCU__0 DISPC_VIDx_ACCU__1 for ping-pong mechanism with external trigger, based on the field polarity) It is used for ARGB and Y setting. Shadow register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								VERTICALACCU																							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:0	VERTICALACCU	Vertical initialization accu signed value.	RW	0x0

Table 8-189. Register Call Summary for Register DISPC_VID_ACCUV_j

Display Controller

- DISPC VID Scaler Unit: [0]
- DISPC_VID Register Summary: [1]

Table 8-190. DISPC_VID_ACCUV2_j

Address Offset	0x0000 0018		
Physical Address	0x5801 7018 + (0x4 * j) 0x5801 8018 + (0x4 * j)	**Instance**	DISPC_VID1 DISPC_VID2
Description	The register configures the resize accumulator init value for vertical up/down-sampling of the video window (DISPC_VID1_ACCU2__0 DISPC_VID1_ACCU2__1 for ping-pong mechanism with external trigger, based on the field polarity) It is used for Cb and Cr setting. It is used only when the pixel format at the input of the filter is one of the YUV formats. If the pixel format at the input of the filter is ARGB (all ARGB, RGB, RGBA are converted to ARGB32-8888 by the color space conversion before going to the filter is the color space conversion is done before the filter). When the register is not used by the HW, any value can be used for the bit-fields. Shadow register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								VERTICALACCU																							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:0	VERTICALACCU	Vertical initialization accu signed value.	RW	0x0

Table 8-191. Register Call Summary for Register DISPC_VID_ACCUV2_j

Display Controller

- DISPC VID Scaler Unit: [0]
- DISPC_VID Register Summary: [1]

Table 8-192. DISPC_VID_ATTRIBUTES

Address Offset	0x0000 0020		
Physical Address	0x5801 7020 0x5801 8020	**Instance**	DISPC_VID1 DISPC_VID2
Description	The register configures the attributes of the video window. Shadow register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	RESERVED	RESERVED	PREMULTIPLYALPHA	ZORDER			SELFREFRESH	ARBITRATION	DOUBLESTRIDE	VERTICALTAPS	RESERVED	BUFFPRELOAD	RESERVED	SELFREFRESHAUTO	CHANNELOUT			RESERVED		FULLRANGE	NIBBLEMODE	COLORCONVENABLE	RESIZEENABLE	FORMAT							ENABLE
R	R	R	RW	RW			RW	RW	RW	RW	R	RW	R	RW	RW			R		RW	RW	RW	RW	RW							RW
0x0	0x0	0x0	0x0	0x0			0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0	0x0			0x0		0x0	0x0	0x0	0x0	0x0							0x0

Bits	Field Name	Description	Type	Reset
31	RESERVED		R	0x0
30	RESERVED		R	0x0
29	RESERVED		R	0x0
28	PREMULTIPLYALPHA	The field configures the DISPC VID1 to process incoming data as premultiplied alpha data or non premultiplied alpha data. Default setting is non premultiplied alpha data. 0x0: Non premultiplyalpha data color component 0x1: Premultiplyalpha data color component	RW	0x0
27:25	ZORDER	Z-Order defining the priority of the layer compared to others when overlaying. It is SW responsibility to ensure that each layer connected to the same overlay manager has a different z-order value. 0x1: Z-order 1: layer above layer with z-order value of 0 and below layers with z-order values of 2 and 3 0x0: Z-order 0: layer above solid background color and below layer with higher Z-order values. 0x2: Z-order 2: layer above layers with z-order value of 0 and 1 and below layer with z-order value of 3 0x4: Z-order 4: layer above layers with z-order value of 0, 1, 2 and 3 and below layer with z-order value of 5 0x5: Z-order 5: layer above all the other layers except cursor 0x3: Z-order 3: layer above layers with z-order value of 0, 1 and 2 and below layer with z-order value of 4	RW	0x0
24	SELFREFRESH	Enables the self refresh of the video window from its own DMA buffer only. 0x0: The video pipeline accesses the interconnect to fetch data from the system memory. 0x1: The video pipeline does not need anymore to fetch data from memory. Only the DMA buffer associated with the video1 is used. It takes effect after the frame has been loaded in the DMA buffer.	RW	0x0

Bits	Field Name	Description	Type	Reset
23	ARBITRATION	Determines the priority of the video pipeline. The video pipeline is one of the high priority pipeline. The arbitration gives always the priority first to the high priority pipelines using round-robin between them. When there is only normal priority pipelines sending requests, the round-robin applies between them. 0x0: The video pipeline is one of the normal priority pipeline. 0x1: The video pipeline is one of the high priority pipeline.	RW	0x0
22	DOUBLESTRIDE	Determines if the stride for CbCr buffer is the 1x or 2x of the Y buffer stride. It is only used in case of YUV420 and 2D access 0x0: The CbCr stride value is equal to the Y stride. 0x1: The CbCr stride value is double to the Y stride.	RW	0x0
21	VERTICALTAPS	Video Vertical Resize Tap Number. The vertial poly-phase filter can be configured in 3-tap or 5-tap configuration. According to the number of taps, the maximum input picture width is double while using 3-tap compared to 5-tap. 0x0: 3 taps are used for the vertical filtering logic. The 2 other taps are not used. The associated bit-fields for the 2 other taps coefficients do not need to be initialized. 0x1: 5 taps are used for the vertical filtering logic.	RW	0x0
20	RESERVED		R	0x0
19	BUFPRELOAD	Video Preload Value 0x0: H/W prefetches pixels up to the preload value defined in the preload register 0x1: H/W prefetches pixels up to high threshold value	RW	0x0
18	RESERVED	Write 0's for future compatibility. Reads return 0.	R	0x0
17	SELFREFRESHAUTO	Automatic self refresh mode 0x0: The transition from SELFREFRESH <disabled> to <enabled> is controlled by SW. 0x1: The transition from SELFREFRESH <disabled> to <enabled> is controlled only by HW.	RW	0x0
16:14	CHANNELOUT	Video Channel Out configuration wr: immediate 0x0: OVR1 (VP1) 0x1: OVR2 0x4: WB	RW	0x0
13:12	RESERVED		R	0x0
11	FULLRANGE	Color Space Conversion full range setting. 0x0: Limited range selected: 16 subtracted from Y before color space conversion 0x1: Full range selected: Y is not modified before the color space conversion	RW	0x0
10	NIBBLEMODE	Video Nibble mode (only for 1-, 2- and 4-bpp) 0x0: Nibble mode is disabled 0x1: Nibble mode is enabled	RW	0x0
9	COLORCONVENABLE	Enable the color space conversion. The HW does not enable/disable the conversion based on the pixel format. The bit-field shall be reset when the format is not YUV. 0x0: Disable Color Space Conversion YUV to RGB 0x1: Enable Color Space Conversion YUV to RGB	RW	0x0

Bits	Field Name	Description	Type	Reset
8:7	RESIZEENABLE	Video Resize Enable 0x0: Disable both horizontal and vertical resize processing 0x1: Enable the horizontal resize processing 0x3: Enable both horizontal and vertical resize processing 0x2: Enable the vertical resize processing	RW	0x0

Bits	Field Name	Description	Type	Reset
6:1	FORMAT	Video Format. It defines the pixel format when fetching the video frame buffer. 0x9: RGBA32-8888 0x15: BITMAP8 (CLUT is required) 0x8: ABGR32-8888 0x5: ARGB16-1555 0x2E: xRGB32-2101010 0x2: RGBA16-4444 0x4: BGR16-565 0x32: RESERVED4 0x23: RESERVED2 0x1: ABGR16-4444 0x3F: UYVY 4:2:2 co-sited 0x28: xBGR32_8888 0x0: ARGB16-4444 0x3E: YUV2 4:2:2 co-sited 0xB: RGB24-888 (24-bit container) 0x3D: NV12/N21 4:2:0 2 buffers (Y + UV) 0x20: xRGB12-4444 0x3: RGB16-565 0x17: BGR565A8 0x30: xRGB64-16161616 0x31: RGBX64_16161616 0x11: RGBA64_16161616 0x6: ABGR16-1555 0x2F: xBGR32-2101010 0xA: BGRA32-8888 0x26: xBGR16-1555 0x10: ARGB64-16161616 0x21: xBGR16-4444 0x25: xRGB16-1555 0x12: BITMAP1 (CLUT is required) 0x13: BITMAP2 (CLUT is required) 0x2B: RESERVED3 0x18: RESERVED1 0x14: BITMAP4 (CLUT is required) 0xE: ARGB32-2101010 0x16: RGB565A8 0x2A: BGRX32_8888 0x7: ARGB32-8888 0x27: xRGB32-8888 (32-bit container) 0x29: RGBx32-8888 (24-bit RGB aligned on MSB of the 32-bit container) 0xF: ABGR32-2101010 0xC: RESERVED 0x22: RGBx16-4444	RW	0x0

Bits	Field Name	Description	Type	Reset
0	ENABLE	Video pipeline Enable 0x0: Video disabled (video pipeline inactive and window not present) 0x1: Video enabled (video pipeline active and window present on the screen)	RW	0x0

Table 8-193. Register Call Summary for Register DISPC_VID_ATTRIBUTES

Display Controller

- DISPC DMA Addressing and Bursts: [0]
- DISPC DMA Buffers: [1]
- DISPC DMA Arbitration: [2]
- DISPC DMA Power Modes: [3][4]
- DISPC Memory Formats: [5]
- DISPC Video Pipelines: [6][7][8][9][10]
- DISPC VID Color Look-Up Table (CLUT): [11]
- DISPC VID CSC Unit YUV to RGB: [12][13][14]
- DISPC VID Scaler Unit: [16][17][18]
- DISPC VID Progressive to Interlace conversion: [19][20]
- DISPC Overlay Managers: [21][22]
- DISPC Overlay Priority Rule: [23]
- DISPC Overlay Alpha Blender: [25]
- DISPC_VID Register Summary: [26]

Table 8-194. DISPC_VID_ATTRIBUTES2

Address Offset	0x0000 0024		
Physical Address	0x5801 7024 0x5801 8024	**Instance**	DISPC_VID1 DISPC_VID2
Description	The register configures the attributes of the video window. Shadow register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	TAGS					RESERVED	REGION_BASED	RESERVED							SECURE	RESERVED								VC1_RANGE_CBCR				VC1_RANGE_Y			VC1ENABLE

Bits	Field Name	Description	Type	Reset
31	RESERVED		R	0x0
30:26	TAGS	Number of OCP TAGS to be used for the pipeline (from 1 to 32)	RW	0x1f
25	RESERVED		R	0x0
24	REGION_BASED	Enable region-based mechanism 0x0: DISABLE 0x1: ENABLE	RW	0x0
23:17	RESERVED		R	0x0
16	SECURE	OCP requests corresponds to pipeline data are secure/unsecure. The bit-field can be modified only by secure transaction using MReqSecure qualifier.	RW	0x0
15:7	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
6:4	VC1_RANGE_CBCR	Defines the VC1 range value for the CbCr component from 0 to 7.	RW	0x0
3:1	VC1_RANGE_Y	Defines the VC1 range value for the Y component from 0 to 7.	RW	0x0
0	VC1ENABLE	Enable/disable the VC1 range mapping processing. The bit-field is ignored if the format is not one of the supported YUV formats. 0x0: VC1 range mapping disabled 0x1: VC1 range mapping enabled	RW	0x0

Table 8-195. Register Call Summary for Register DISPC_VID_ATTRIBUTES2

Display Controller
- DISPC VID VC-1 Range Mapping Unit: [0][1][2]
- Region-Based Mechanism Overview: [3]
- DISPC_VID Register Summary: [4]

Table 8-196. DISPC_VID_BA_j

Address Offset	0x0000 0028		
Physical Address	0x5801 7028 + (0x4 * j) 0x5801 8028 + (0x4 * j)	**Instance**	DISPC_VID1 DISPC_VID2
Description	The register configures the base address of the video buffer for the video window (DISPC_VID1_BA__0 DISPC_VID1_BA__1 for ping-pong mechanism with external trigger, based on the field polarity otherwise only DISPC_VID1_BA__0 is used). Shadow register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BA																															

Bits	Field Name	Description	Type	Reset
31:0	BA	Video base address When decompression is enabled bit[5:0] shall be set to 0. Base address of the video buffer (aligned on pixel size boundary except in case of RGB24 packed format, 4-pixel alignment is required; in case of YUV422, 2-pixel alignment is required, and YUV420, byte alignment is supported)). It case of YUV 4:2:0 format, it indicates the base address of the Y buffer. When the TILER is addressed, the bits: [28:27] = 0x0 for 8-bit tiled [28:27] = 0x1 for 16-bit tiled [28:27] = 0x2 for 32-bit tiled [28:27] = 0x3 for page mode [31:29] = 0x0 for 0-degree view [31:29] = 0x1 for 180-degree view + mirroring [31:29] = 0x2 for 0-degree view + mirroring [31:29] = 0x3 for 180-degree view [31:29] = 0x4 for 270-degree view + mirroring [31:29] = 0x5 for 270-degree view [31:29] = 0x6 for 90-degree view [31:29] = 0x7 for 90-degree view + mirroring Otherwise the bits indicated the corresponding bit address to access the SDRAM.	RW	0x0

Table 8-197. Register Call Summary for Register DISPC_VID_BA_j

Display Controller
- DISPC DMA Addressing and Bursts: [0][1][2]
- DISPC VID Progressive to Interlace conversion: [3]
- DISPC_VID Register Summary: [4]

Table 8-198. DISPC_VID_BA_UV_j

Address Offset	0x0000 0030		
Physical Address	0x5801 7030 + (0x4 * j) 0x5801 8030 + (0x4 * j)	**Instance**	DISPC_VID1 DISPC_VID2
Description	The register configures the base address of the UV buffer for the video window. (DISPC_VID1_BA_UV__0 DISPC_VID1_BA_UV__1 for ping-pong mechanism with external trigger, based on the field polarity otherwise only DISPC_VID1_BA_UV__0 is used)). The register is also used to configure the RGB plane BA for RGB565A8 format Shadow register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BA																															

Bits	Field Name	Description	Type	Reset
31:0	BA	Video base address alinged on 16-bit boundary Base address of the UV video buffer used only in case of YUV420-NV12 When the TILER is addressed, the bits: [28:27] = 0x0 for 8-bit tiled [28:27] = 0x1 for 16-bit tiled [28:27] = 0x2 for 32-bit tiled [28:27] = 0x3 for page mode [31:29] = 0x0 for 0-degree view [31:29] = 0x1 for 180-degree view + mirroring [31:29] = 0x2 for 0-degree view + mirroring [31:29] = 0x3 for 180-degree view [31:29] = 0x4 for 270-degree view + mirroring [31:29] = 0x5 for 270-degree view [31:29] = 0x6 for 90-degree view [31:29] = 0x7 for 90-degree view + mirroring Otherwise the bits indicated the corresponding bit address to access the SDRAM.	RW	0x0

Table 8-199. Register Call Summary for Register DISPC_VID_BA_UV_j

Display Controller

- DISPC DMA Addressing and Bursts: [0][1][2]
- DISPC VID Progressive to Interlace conversion: [3]
- DISPC_VID Register Summary: [4]

Table 8-200. DISPC_VID_BUF_SIZE_STATUS

Address Offset	0x0000 0038		
Physical Address	0x5801 7038 0x5801 8038	**Instance**	DISPC_VID1 DISPC_VID2
Description	The register defines the Video buffer size for the video pipeline.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																BUFSIZE															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Write 0's for future compatibility. Reads return 0	R	0x0
15:0	BUFSIZE	Video DMA buffer Size in number of 128-bits	R	0xa00

Table 8-201. Register Call Summary for Register DISPC_VID_BUF_SIZE_STATUS

Display Controller

- DISPC_VID Register Summary: [0]

Table 8-202. DISPC_VID_BUF_THRESHOLD

Address Offset	0x0000 003C		
Physical Address	0x5801 703C 0x5801 803C	**Instance**	DISPC_VID1 DISPC_VID2
Description	The register configures the video buffer associated with the video pipeline. Shadow register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BUFHIGHTHRESHOLD																BUFLOWTHRESHOLD															

Bits	Field Name	Description	Type	Reset
31:16	BUFHIGHTHRESHOLD	Video DMA buffer High Threshold Number of 128-bits defining the threshold value.	RW	0x9ff
15:0	BUFLOWTHRESHOLD	DMA buffer High Threshold Number of 128-bits defining the threshold value.	RW	0x9f8

Table 8-203. Register Call Summary for Register DISPC_VID_BUF_THRESHOLD

Display Controller

- DISPC DMA Buffers: [0][1][2]
- DISPC_VID Register Summary: [3]

Table 8-204. DISPC_VID_CONV_COEF0

Address Offset	0x0000 0040		
Physical Address	0x5801 7040 0x5801 8040	**Instance**	DISPC_VID1 DISPC_VID2
Description	The register configures the color space conversion matrix coefficients for the video pipeline. Shadow register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								RCR								RESERVED								RY							

Bits	Field Name	Description	Type	Reset
31:27	RESERVED	Write 0's for future compatibility. Reads return 0	R	0x0
26:16	RCR	RCr Coefficient Encoded signed value (from -1024 to 1023).	RW	0x0
15:11	RESERVED	Write 0's for future compatibility. Reads return 0	R	0x0
10:0	RY	RY Coefficient Encoded signed value (from -1024 to 1023).	RW	0x0

Table 8-205. Register Call Summary for Register DISPC_VID_CONV_COEF0

Display Controller

- DISPC VID CSC Unit YUV to RGB: [0][1]
- DISPC_VID Register Summary: [2]

Table 8-206. DISPC_VID_CONV_COEF1

Address Offset	0x0000 0044		
Physical Address	0x5801 7044 0x5801 8044	**Instance**	DISPC_VID1 DISPC_VID2
Description	The register configures the color space conversion matrix coefficients for the video pipeline. Shadow register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								GY								RESERVED								RCB							

Bits	Field Name	Description	Type	Reset
31:27	RESERVED	Write 0's for future compatibility. Reads return 0	R	0x0
26:16	GY	GY Coefficient Encoded signed value (from -1024 to 1023).	RW	0x0
15:11	RESERVED	Write 0's for future compatibility. Reads return 0	R	0x0
10:0	RCB	RCb Coefficient Encoded signed value (from -1024 to 1023).	RW	0x0

Table 8-207. Register Call Summary for Register DISPC_VID_CONV_COEF1

Display Controller
- DISPC VID CSC Unit YUV to RGB: [0][1]
- DISPC_VID Register Summary: [2]

Table 8-208. DISPC_VID_CONV_COEF2

Address Offset	0x0000 0048		
Physical Address	0x5801 7048 0x5801 8048	**Instance**	DISPC_VID1 DISPC_VID2
Description	The register configures the color space conversion matrix coefficients for the video pipeline. Shadow register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								GCB								RESERVED								GCR							

Bits	Field Name	Description	Type	Reset
31:27	RESERVED	Write 0's for future compatibility. Reads return 0	R	0x0
26:16	GCB	GCb Coefficient Encoded signed value (from -1024 to 1023).	RW	0x0
15:11	RESERVED	Write 0's for future compatibility. Reads return 0	R	0x0
10:0	GCR	GCr Coefficient Encoded signed value (from -1024 to 1023).	RW	0x0

Table 8-209. Register Call Summary for Register DISPC_VID_CONV_COEF2

Display Controller
- DISPC VID CSC Unit YUV to RGB: [0][1]
- DISPC_VID Register Summary: [2]

Table 8-210. DISPC_VID_CONV_COEF3

Address Offset	0x0000 004C		
Physical Address	0x5801 704C 0x5801 804C	**Instance**	DISPC_VID1 DISPC_VID2
Description	The register configures the color space conversion matrix coefficients for the video pipeline. Shadow register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								BCR								RESERVED								BY							

Bits	Field Name	Description	Type	Reset
31:27	RESERVED	Write 0's for future compatibility. Reads return 0	R	0x0
26:16	BCR	BCr coefficient Encoded signed value (from -1024 to 1023).	RW	0x0
15:11	RESERVED	Write 0's for future compatibility. Reads return 0	R	0x0
10:0	BY	BY coefficient Encoded signed value (from -1024 to 1023).	RW	0x0

Table 8-211. Register Call Summary for Register DISPC_VID_CONV_COEF3

Display Controller
- DISPC VID CSC Unit YUV to RGB: [0][1]
- DISPC_VID Register Summary: [2]

Table 8-212. DISPC_VID_CONV_COEF4

Address Offset	0x0000 0050		
Physical Address	0x5801 7050 0x5801 8050	**Instance**	DISPC_VID1 DISPC_VID2
Description	The register configures the color space conversion matrix coefficients for the video pipeline. Shadow register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								BCB							

Bits	Field Name	Description	Type	Reset
31:11	RESERVED	Write 0's for future compatibility. Reads return 0	R	0x0
10:0	BCB	BCb Coefficient Encoded signed value (from -1024 to 1023).	RW	0x0

Table 8-213. Register Call Summary for Register DISPC_VID_CONV_COEF4

Display Controller
- DISPC VID CSC Unit YUV to RGB: [0]
- DISPC_VID Register Summary: [1]

Table 8-214. DISPC_VID_CONV_COEF5

Address Offset	0x0000 0054		
Physical Address	0x5801 7054 0x5801 8054	**Instance**	DISPC_VID1 DISPC_VID2
Description	The register configures the color space conversion matrix coefficients for the video pipeline. Shadow register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GOFFSET													RESERVED			ROFFSET								RESERVED							

Bits	Field Name	Description	Type	Reset
31:19	GOFFSET	G offset Encoded signed value (from -4096 to 4095).	RW	0x0
18:16	RESERVED		R	0x0
15:3	ROFFSET	R offset Encoded signed value (from -4096 to 4095).	RW	0x0
2:0	RESERVED		R	0x0

Table 8-215. Register Call Summary for Register DISPC_VID_CONV_COEF5

Display Controller
- DISPC VID CSC Unit YUV to RGB: [0][1]
- DISPC_VID Register Summary: [2]

Table 8-216. DISPC_VID_CONV_COEF6

Address Offset	0x0000 0058		
Physical Address	0x5801 7058 0x5801 8058	**Instance**	DISPC_VID1 DISPC_VID2
Description	The register configures the color space conversion matrix coefficients for the video pipeline. Shadow register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																BOFFSET								RESERVED							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0
15:3	BOFFSET	B offset Encoded signed value (from -4096 to 4095).	RW	0x0
2:0	RESERVED		R	0x0

Table 8-217. Register Call Summary for Register DISPC_VID_CONV_COEF6

Display Controller
- DISPC VID CSC Unit YUV to RGB: [0]
- DISPC_VID Register Summary: [1]

Table 8-218. DISPC_VID_FIRH

Address Offset	0x0000 005C		
Physical Address	0x5801 705C 0x5801 805C	**Instance**	DISPC_VID1 DISPC_VID2
Description	The register configures the resize factor for horizontal up/down-sampling of the video window. It is used for ARGB and Y setting. Shadow register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								FIRHINC																							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:0	FIRHINC	Horizontal increment of the up/down-sampling filter. The value 0 is invalid.	RW	0x200000

Table 8-219. Register Call Summary for Register DISPC_VID_FIRH

Display Controller

- DISPC VID Scaler Unit: [0]
- DISPC_VID Register Summary: [1]

Table 8-220. DISPC_VID_FIRH2

Address Offset	0x0000 0060		
Physical Address	0x5801 7060 0x5801 8060	**Instance**	DISPC_VID1 DISPC_VID2
Description	The register configures the resize factor for horizontal up/down-sampling of the video window. It is used for Crb and Cr setting. It is used only when the pixel format at the input of the filter is one of the YUV formats. If the pixel format at the input of the filter is ARGB (all ARGB, RGB, RGBA are converted to ARGB32-8888 by the color space conversion before going to the filter is the color space conversion is done before the filter). When the register is not used by the HW, any value can be used for the bit-fields. Shadow register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								FIRHINC																							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:0	FIRHINC	Horizontal increment of the up/down-sampling filter for Cb and Cr. The value 0 is invalid.	RW	0x200000

Table 8-221. Register Call Summary for Register DISPC_VID_FIRH2

Display Controller

- DISPC VID Scaler Unit: [0]
- DISPC_VID Register Summary: [1]

Table 8-222. DISPC_VID_FIRV

Address Offset	0x0000 0064		
Physical Address	0x5801 7064 0x5801 8064	**Instance**	DISPC_VID1 DISPC_VID2
Description	The register configures the resize factor for vertical up/down-sampling of the video window. It is used for ARGB and Y setting. Shadow register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								FIRVINC																							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:0	FIRVINC	Vertical increment of the up/down-sampling filter. The value 0 is invalid.	RW	0x200000

Table 8-223. Register Call Summary for Register DISPC_VID_FIRV

Display Controller

- DISPC VID Scaler Unit: [0]
- DISPC_VID Register Summary: [1]

Table 8-224. DISPC_VID_FIRV2

Address Offset	0x0000 0068		
Physical Address	0x5801 7068 0x5801 8068	**Instance**	DISPC_VID1 DISPC_VID2
Description	The register configures the resize factor for vertical up/down-sampling of the video window. It is used for Crb and Cr setting. It is used only when the pixel format at the input of the filter is one of the YUV formats. If the pixel format at the input of the filter is ARGB (all ARGB, RGB, RGBA are converted to ARGB32-8888 by the color space conversion before going to the filter is the color space conversion is done before the filter). When the register is not used by the HW, any value can be used for the bit-fields. Shadow register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								FIRVINC																							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:0	FIRVINC	Vertical increment of the up/down-sampling filter for Cb and Cr. The value 0 is invalid.	RW	0x200000

Table 8-225. Register Call Summary for Register DISPC_VID_FIRV2

Display Controller

- DISPC VID Scaler Unit: [0]
- DISPC_VID Register Summary: [1]

Table 8-226. DISPC_VID_FIR_COEF_H0_i

Address Offset	0x0000 006C		
Physical Address	0x5801 706C + (0x4 * i) 0x5801 806C + (0x4 * i)	**Instance**	DISPC_VID1 DISPC_VID2
Description	The bank of registers configure the up/down-scaling coefficients for the horizontal resize of the video picture associated with the video window for the 16 phases It is used for ARGB and Y setting. Shadow register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								RESERVED																FIRHC0							

Bits	Field Name	Description	Type	Reset
31:30	RESERVED		R	0x0
29:10	RESERVED		R	0x0
9:0	FIRHC0	Unsigned coefficient C0 for the horizontal up/down-scaling with the phase n	RW	0x0

Table 8-227. Register Call Summary for Register DISPC_VID_FIR_COEF_H0_i

Display Controller

- DISPC VID Scaler Unit: [0][1]
- DISPC_VID Register Summary: [2]

Table 8-228. DISPC_VID_FIR_COEF_H0_C_i

Address Offset	0x0000 0090		
Physical Address	0x5801 7090 + (0x4 * i) 0x5801 8090 + (0x4 * i)	**Instance**	DISPC_VID1 DISPC_VID2
Description	The bank of registers configure the up/down-scaling coefficients for the horizontal resize of the video picture associated with the video window for the phases from 0 to 15. It is used for Crb and Cr setting. It is used only when the pixel format at the input of the filter is one of the YUV formats. If the pixel format at the input of the filter is ARGB (all ARGB, RGB, RGBA are converted to ARGB32-8888 by the color space conversion before going to the filter is the color space conversion is done before the filter). When the register is not used by the HW, any value can be used for the bit-fields. Shadow register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								RESERVED																FIRHC0							

Bits	Field Name	Description	Type	Reset
31:30	RESERVED		R	0x0
29:10	RESERVED		R	0x0
9:0	FIRHC0	Unsigned coefficient C0 for the horizontal up/down-scaling with the phase n	RW	0x0

Table 8-229. Register Call Summary for Register DISPC_VID_FIR_COEF_H0_C_i

Display Controller
- DISPC VID Scaler Unit: [0][1]
- DISPC_VID Register Summary: [2]

Table 8-230. DISPC_VID_FIR_COEF_H12_k

Address Offset	0x0000 00B4		
Physical Address	0x5801 70B4 + (0x4 * k) 0x5801 80B4 + (0x4 * k)	**Instance**	DISPC_VID1 DISPC_VID2
Description	The bank of registers configure the up/down-scaling coefficients for the horizontal resize of the video picture associated with the video window for the 16 phases It is used for ARGB and Y setting. Shadow register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								FIRHC2								FIRHC1								RESERVED							

Bits	Field Name	Description	Type	Reset
31:30	RESERVED		R	0x0
29:20	FIRHC2	Signed coefficient C2 for the horizontal up/down-scaling with the phase n	RW	0x0
19:10	FIRHC1	Signed coefficient C1 for the horizontal up/down-scaling with the phase n	RW	0x0
9:0	RESERVED		R	0x0

Table 8-231. Register Call Summary for Register DISPC_VID_FIR_COEF_H12_k

Display Controller
- DISPC VID Scaler Unit: [0][1]
- DISPC_VID Register Summary: [2]

Table 8-232. DISPC_VID_FIR_COEF_H12_C_k

Address Offset	0x0000 00F4		
Physical Address	0x5801 70F4 + (0x4 * k) 0x5801 80F4 + (0x4 * k)	**Instance**	DISPC_VID1 DISPC_VID2
Description	The bank of registers configure the up/down-scaling coefficients for the horizontal resize of the video picture associated with the video window for the phases from 0 to 15. It is used for Crb and Cr setting. It is used only when the pixel format at the input of the filter is one of the YUV formats. If the pixel format at the input of the filter is ARGB (all ARGB, RGB, RGBA are converted to ARGB32-8888 by the color space conversion before going to the filter is the color space conversion is done before the filter). When the register is not used by the HW, any value can be used for the bit-fields. Shadow register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								FIRHC2								FIRHC1								RESERVED							

Bits	Field Name	Description	Type	Reset
31:30	RESERVED		R	0x0
29:20	FIRHC2	Signed coefficient C2 for the horizontal up/down-scaling with the phase n	RW	0x0
19:10	FIRHC1	Signed coefficient C1 for the horizontal up/down-scaling with the phase n	RW	0x0
9:0	RESERVED		R	0x0

Table 8-233. Register Call Summary for Register DISPC_VID_FIR_COEF_H12_C_k

Display Controller
- DISPC VID Scaler Unit: [0][1]
- DISPC_VID Register Summary: [2]

Table 8-234. DISPC_VID_FIR_COEF_V0_i

Address Offset	0x0000 0134		
Physical Address	0x5801 7134 + (0x4 * i) 0x5801 8134 + (0x4 * i)	**Instance**	DISPC_VID1 DISPC_VID2
Description	The bank of registers configure the up/down-scaling coefficients for the vertical resize of the video picture associated with the video window for the 16 phases It is used for ARGB and Y setting. Shadow register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								RESERVED																FIRVC0							

Bits	Field Name	Description	Type	Reset
31:30	RESERVED		R	0x0
29:10	RESERVED		R	0x0
9:0	FIRVC0	Unsigned coefficient C0 for the vertical up/down-scaling with the phase n	RW	0x0

Table 8-235. Register Call Summary for Register DISPC_VID_FIR_COEF_V0_i

Display Controller
- DISPC VID Scaler Unit: [0][1]
- DISPC_VID Register Summary: [2]

Table 8-236. DISPC_VID_FIR_COEF_V0_C_i

Address Offset	0x0000 0158		
Physical Address	0x5801 7158 + (0x4 * i) 0x5801 8158 + (0x4 * i)	**Instance**	DISPC_VID1 DISPC_VID2
Description	The bank of registers configure the up/down-scaling coefficients for the vertical resize of the video picture associated with the video window for the phases from 0 to 15. It is used for Crb and Cr setting. It is used only when the pixel format at the input of the filter is one of the YUV formats. If the pixel format at the input of the filter is ARGB (all ARGB, RGB, RGBA are converted to ARGB32-8888 by the color space conversion before going to the filter is the color space conversion is done before the filter). When the register is not used by the HW, any value can be used for the bit-fields. Shadow register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								RESERVED																FIRVC0							

Bits	Field Name	Description	Type	Reset
31:30	RESERVED		R	0x0
29:10	RESERVED		R	0x0
9:0	FIRVC0	Unsigned coefficient C0 for the vertical up/down-scaling with the phase n	RW	0x0

Table 8-237. Register Call Summary for Register DISPC_VID_FIR_COEF_V0_C_i

Display Controller

- DISPC VID Scaler Unit: [0][1]
- DISPC_VID Register Summary: [2]

Table 8-238. DISPC_VID_FIR_COEF_V12_k

Address Offset	0x0000 017C		
Physical Address	0x5801 717C + (0x4 * k) 0x5801 817C + (0x4 * k)	**Instance**	DISPC_VID1 DISPC_VID2
Description	The bank of registers configure the up/down-scaling coefficients for the vertical resize of the video picture associated with the video window for the 16 phases It is used for ARGB and Y setting. Shadow register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								FIRVC2								FIRVC1								RESERVED							

Bits	Field Name	Description	Type	Reset
31:30	RESERVED		R	0x0
29:20	FIRVC2	Signed coefficient C2 for the vertical up/down-scaling with the phase n	RW	0x0
19:10	FIRVC1	Signed coefficient C1 for the vertical up/down-scaling with the phase n	RW	0x0
9:0	RESERVED		R	0x0

Table 8-239. Register Call Summary for Register DISPC_VID_FIR_COEF_V12_k

Display Controller
- DISPC VID Scaler Unit: [0][1]
- DISPC_VID Register Summary: [2]

Table 8-240. DISPC_VID_FIR_COEF_V12_C_k

Address Offset	0x0000 01BC		
Physical Address	0x5801 71BC + (0x4 * k) 0x5801 81BC + (0x4 * k)	**Instance**	DISPC_VID1 DISPC_VID2
Description	The bank of registers configure the up/down-scaling coefficients for the vertical resize of the video picture associated with the video window for the phases from 0 to 15. It is used for Crb and Cr setting. It is used only when the pixel format at the input of the filter is one of the YUV formats. If the pixel format at the input of the filter is ARGB (all ARGB, RGB, RGBA are converted to ARGB32-8888 by the color space conversion before going to the filter is the color space conversion is done before the filter). When the register is not used by the HW, any value can be used for the bit-fields. Shadow register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								FIRVC2								FIRVC1								RESERVED							

Bits	Field Name	Description	Type	Reset
31:30	RESERVED		R	0x0
29:20	FIRVC2	Signed coefficient C2 for the vertical up/down-scaling with the phase n	RW	0x0
19:10	FIRVC1	Signed coefficient C1 for the vertical up/down-scaling with the phase n	RW	0x0
9:0	RESERVED		R	0x0

Table 8-241. Register Call Summary for Register DISPC_VID_FIR_COEF_V12_C_k

Display Controller
- DISPC VID Scaler Unit: [0][1]
- DISPC_VID Register Summary: [2]

Table 8-242. DISPC_VID_GLOBAL_ALPHA

Address Offset	0x0000 01FC		
Physical Address	0x5801 71FC 0x5801 81FC	**Instance**	DISPC_VID1 DISPC_VID2
Description	The register defines the global alpha value for the video pipeline. Shadow register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								GLOBALALPHA							

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x0
7:0	GLOBALALPHA	Global alpha value from 0 to 255. 0 corresponds to fully transparent and 255 to fully opaque.	RW	0xff

Table 8-243. Register Call Summary for Register DISPC_VID_GLOBAL_ALPHA

Display Controller

- DISPC Overlay Alpha Blender: [0]
- DISPC_VID Register Summary: [1]

Table 8-244. DISPC_VID_IRQENABLE

Address Offset	0x0000 0200		
Physical Address	0x5801 7200 0x5801 8200	**Instance**	DISPC_VID1 DISPC_VID2
Description	This register allows to mask/unmask the module internal sources of interrupt, on an event-by-event basis		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												VIDREGIONBASEDPIPEEND_EN	VIDREGIONBASEDPIPESTART_EN	VIDENDWINDOW_EN	VIDBUFFERUNDERFLOW_EN

Bits	Field Name	Description	Type	Reset
31:4	RESERVED		R	0x0
3	VIDREGIONBASEDPIPEEND_EN	PIPE end window IRQ for region-based feature 0x0: VIDREGIONBASEDPIPEEND is masked 0x1: VIDREGIONBASEDPIPEEND generates an interrupt when it occurs	RW	0x0
2	VIDREGIONBASEDPIPESTART_EN	PIPE start window IRQ for region-based feature 0x0: VIDREGIONBASEDPIPESTART is masked 0x1: VIDREGIONBASEDPIPESTART generates an interrupt when it occurs	RW	0x0
1	VIDENDWINDOW_EN	The end of the video Window has been reached. It is detected by the overlay manager when the full video has been displayed. 0x0: EndVid1Window is masked 0x1: EndVid1Window generates an interrupt when it occurs	RW	0x0
0	VIDBUFFERUNDERFLOW_EN	Video DMA Buffer Underflow. The DMA buffer is not necessary empty but required data are not present in the DMA buffer (due to out of order responses) 0x0: Vid1BufferUnderflow is masked 0x1: Vid1BufferUnderflow generates an interrupt when it occurs	RW	0x0

Table 8-245. Register Call Summary for Register DISPC_VID_IRQENABLE

Display Controller

- DISPC Interrupt Requests: [0]
- DISPC_VID Register Summary: [1]

Table 8-246. DISPC_VID_IRQSTATUS

Address Offset	0x0000 0204		
Physical Address	0x5801 7204 0x5801 8204	**Instance**	DISPC_VID1 DISPC_VID2
Description	This register regroups all the status of the module internal events that generate an interrupt. Write 1 to a given bit resets this bit		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												VIDREGIONBASEDPIPEEND_IRQ	VIDREGIONBASEDPIPESTART_IRQ	VIDENDWINDOW_IRQ	VIDBUFFERUNDERFLOW_IRQ

Bits	Field Name	Description	Type	Reset
31:4	RESERVED		R	0x0
3	VIDREGIONBASEDPIPEEND_IRQ	PIPE end window IRQ for region-based feature 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW	0x0
2	VIDREGIONBASEDPIPESTART_IRQ	PIPE start window IRQ for region-based feature 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW	0x0
1	VIDENDWINDOW_IRQ	The end of the video Window has been reached. It is detected by the overlay manager when the full video has been displayed. 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW	0x0
0	VIDBUFFERUNDERFLOW_IRQ	Video DMA Buffer Underflow. The DMA buffer is not necessarly empty but required data are not present in the DMA buffer (due to out of order responses) 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW	0x0

Table 8-247. Register Call Summary for Register DISPC_VID_IRQSTATUS

Display Controller

- DISPC Interrupt Requests: [0]
- DISPC_VID Register Summary: [1]

Table 8-248. DISPC_VID_MFLAG_THRESHOLD

Address Offset	0x0000 0208		
Physical Address	0x5801 7208 0x5801 8208	**Instance**	DISPC_VID1 DISPC_VID2
Description	MFLAG thresholds for video pipelines. Shadow register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HT_MFLAG																LT_MFLAG															

Bits	Field Name	Description	Type	Reset
31:16	HT_MFLAG	High Thresholds (in 128bits) for MFLAG generation: when FIFO fullness reaches HT_MFLAG level, MFLAG is reset to 0	RW	0x0
15:0	LT_MFLAG	Low Thresholds (in 128bits) for MFLAG generation: when FIFO fullness reaches LT_MFLAG level, MFLAG is set to 1	RW	0x0

Table 8-249. Register Call Summary for Register DISPC_VID_MFLAG_THRESHOLD

Display Controller

- DISPC DMA MFLAG Mechanism and Arbitration: [0][1]
- DISPC_VID Register Summary: [2]

Table 8-250. DISPC_VID_PICTURE_SIZE

Address Offset	0x0000 020C		
Physical Address	0x5801 720C 0x5801 820C	**Instance**	DISPC_VID1 DISPC_VID2
Description	The register configures the size of the video picture associated with the video layer before up/down-scaling. Shadow register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				MEMSIZEY												RESERVED				MEMSIZEX											

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:16	MEMSIZEY	Number of lines of the video picture Encoded value (from 1 to 4096) to specify the number of lines of the video picture in memory (program to value minus one). When predecimation is set, the value represents the size of the image after predecimation but the max size of the unpredecimated image size in memory is still bounded to 2exp(11).	RW	0x0
15:12	RESERVED		R	0x0
11:0	MEMSIZEX	Number of pixels of the video picture Encoded value (from 1 to 4096) to specify the number of pixels of the video picture in memory (program to value minus one). The size is limited to the size of the line buffer of the vertical sampling block in case the video picture is processed by the vertical filtering unit. (program to value minus one). When predecimation is set, the value represents the size of the image after predecimation but the max size of the unpredecimated image size in memory is still bounded to 2exp(11).	RW	0x0

Table 8-251. Register Call Summary for Register DISPC_VID_PICTURE_SIZE

Display Controller

- DISPC VID Scaler Unit:
- DISPC_VID Register Summary: [2]

Table 8-252. DISPC_VID_PIXEL_INC

Address Offset	0x0000 0210		
Physical Address	0x5801 7210 0x5801 8210	**Instance**	DISPC_VID1 DISPC_VID2
Description	The register configures the number of bytes to increment between two pixels for the buffer associated with the video window. The register is used only when the TILER is not present in the system in order to perform low performance rotation. When the TILER IP is present it is highly recommanded to use it for performing the rotation. Shadow register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								PIXELINC							

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Write 0's for future compatibility. Reads return 0	R	0x0
7:0	PIXELINC	Number of bytes to increment between two pixels. Encoded unsigned value (from 1 to 255) to specify the number of bytes between two pixels in the video buffer. The value 0 is invalid. The value 1 means next pixel. The value 1+n*bpp means increment of n pixels. For YUV420, Max supported value is 128.	RW	0x1

Table 8-253. Register Call Summary for Register DISPC_VID_PIXEL_INC

Display Controller

- DISPC DMA Addressing and Bursts: [0]
- DISPC DMA Predecimation: [1]
- DISPC_VID Register Summary: [2]

Table 8-254. DISPC_VID_POSITION

Address Offset	0x0000 0214		
Physical Address	0x5801 7214 0x5801 8214	**Instance**	DISPC_VID1 DISPC_VID2
Description	The register configures the position of the video window. Shadow register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				POSY												RESERVED				POSX											

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:16	POSY	Y position of the video window Encoded value (from 0 to 4095) to specify the Y position of the video window #1 .The line at the top has the Y-position 0.	RW	0x0
15:12	RESERVED		R	0x0
11:0	POSX	X position of the video window Encoded value (from 0 to 4095) to specify the X position of the video window #1. The first pixel on the left of the display screen has the X-position 0.	RW	0x0

Table 8-255. Register Call Summary for Register DISPC_VID_POSITION

Display Controller

- DISPC Overlay Priority Rule: [0]
- DISPC VP1 Configuration for TV Support: [2]
- DISPC Extended 3D Support - Line Alternative Format: [3]
- DISPC_VID Register Summary: [4]

Table 8-256. DISPC_VID_PRELOAD

Address Offset	0x0000 0218		
Physical Address	0x5801 7218 0x5801 8218	**Instance**	DISPC_VID1 DISPC_VID2
Description	The register configures the DMA buffer of the video pipeline. Shadow register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																				PRELOAD											

Bits	Field Name	Description	Type	Reset
31:12	RESERVED	Write 0's for future compatibility. Reads return 0	R	0x0
11:0	PRELOAD	DMA buffer preload value Number of 128-bit words defining the preload value.	RW	0x100

Table 8-257. Register Call Summary for Register DISPC_VID_PRELOAD

Display Controller

- DISPC DMA Buffers: [0]
- DISPC_VID Register Summary: [1]

Table 8-258. DISPC_VID_ROW_INC

Address Offset	0x0000 021C		
Physical Address	0x5801 721C 0x5801 821C	**Instance**	DISPC_VID1 DISPC_VID2
Description	The register configures the number of bytes to increment at the end of the row for the buffer associated with the video window. Shadow register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ROWINC																															

Bits	Field Name	Description	Type	Reset
31:0	ROWINC	Number of bytes to increment at the end of the row Encoded signed value (from -2^31-1 to 2^31) to specify the number of bytes to increment at the end of the row in the video buffer. The value 0 is invalid. The value 1 means next pixel. The value 1+n*bpp means increment of n pixels. The value 1- (n+1)*bpp means decrement of n pixels.	RW	0x1

Table 8-259. Register Call Summary for Register DISPC_VID_ROW_INC

Display Controller

- DISPC DMA Addressing and Bursts: [0][1]
- DISPC DMA Predecimation: [3]
- DISPC_VID Register Summary: [4]

Table 8-260. DISPC_VID_SIZE

Address Offset	0x0000 0220		
Physical Address	0x5801 7220 0x5801 8220	**Instance**	DISPC_VID1 DISPC_VID2
Description	The register configures the size of the video window. Shadow register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								SIZEY								RESERVED								SIZEX							

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:16	SIZEY	Number of lines of the video window. Encoded value (from 1 to 4096) to specify the number of lines of the video window (program size -1).	RW	0x0
15:12	RESERVED		R	0x0
11:0	SIZEX	Number of pixels of the video window. Encoded value (from 1 to 4096) to specify the number of pixels of the video window (program size -1).	RW	0x0

Table 8-261. Register Call Summary for Register DISPC_VID_SIZE

Display Controller

- DISPC DMA Addressing and Bursts: [0]
- DISPC VID Scaler Unit:
- DISPC Overlay Priority Rule: [5]
- DISPC Extended 3D Support - Line Alternative Format: [7][8]
- DISPC_VID Register Summary: [9]

Table 8-262. DISPC_VID_CLUT

Address Offset	0x0000 0224		
Physical Address	0x5801 7224 0x5801 8224	**Instance**	DISPC_VID1 DISPC_VID2
Description	The register configures the Color Look Up Table (CLUT) for VID pipeline. CLUT is used in conjunction with bitmap formats		
Type	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INDEX								VALUE_R								VALUE_G								VALUE_B							

Bits	Field Name	Description	Type	Reset
31:24	INDEX	Defines the location in the table where the bit-field VALUE is stored	W	0x0
23:16	VALUE_R	8-bit value used to defined the value to store at the location in the table defined by the bit-field INDEX.	W	0x0
15:8	VALUE_G	8-bit value used to defined the value to store at the location in the table defined by the bit-field INDEX.	W	0x0
7:0	VALUE_B	8-bit value used to defined the value to store at the location in the table defined by the bit-field INDEX.	W	0x0

Table 8-263. Register Call Summary for Register DISPC_VID_CLUT

Display Controller
- DISPC VID Color Look-Up Table (CLUT): [0][1]
- DISPC_VID Register Summary: [2]

8.2.5.6 DISPC_OVR Registers

8.2.5.6.1 DISPC_OVR Register Summary

Table 8-264. DISPC_OVR Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	DISPC_OVR1 L3_MAIN Physical Address	DISPC_OVR2 L3_MAIN Physical Address
DISPC_OVR_CONFIG	RW	32	0x0000 0000	0x5801 A800	0x5801 A900
RESERVED	R	32	0x0000 0004	0x5801 A804	0x5801 A904
DISPC_OVR_DEFAULT_COLOR	RW	32	0x0000 0008	0x5801 A808	0x5801 A908
DISPC_OVR_DEFAULT_COLOR2	RW	32	0x0000 000C	0x5801 A80C	0x5801 A90C
DISPC_OVR_TRANS_COLOR_MAX	RW	32	0x0000 0010	0x5801 A810	0x5801 A910
DISPC_OVR_TRANS_COLOR_MAX2	RW	32	0x0000 0014	0x5801 A814	0x5801 A914
DISPC_OVR_TRANS_COLOR_MIN	RW	32	0x0000 0018	0x5801 A818	0x5801 A918
DISPC_OVR_TRANS_COLOR_MIN2	RW	32	0x0000 001C	0x5801 A81C	0x5801 A91C

8.2.5.6.2 DISPC_OVR Register Description

Table 8-265. DISPC_OVR_CONFIG

Address Offset	0x0000 0000		
Physical Address	0x5801 A800 0x5801 A900	**Instance**	DISPC_OVR1 DISPC_OVR2
Description	The control register configures the Display Controller module for the VP output. Shadow register. **Note: GLBCE is not supported in this family of devices.**		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																		GLBCESEL	GLBCEEN	TCKLCDSELECTION	TCKLCDENABLE	INTERLEAVED3DMODE		RESERVED							

Bits	Field Name	Description	Type	Reset
31:14	RESERVED		R	0x0
13	GLBCESEL	Selection between GLBCE-0 and GLBCE-1 0x0: GLBCE_0 0x1: GLBCE_1	RW	0x0
12	GLBCEEN	Enable the GLBCE processing 0x0: Disabled 0x1: Enabled	RW	0x0
11	TCKLCDSELECTION	Transparency Color Key Selection Shadow bit-field. 0x0: Destination transparency color key selected 0x1: Source transparency color key selected	RW	0x0
10	TCKLCDENABLE	Transparency Color Key Enabled Shadow bit-field. 0x0: Disable the transparency color key 0x1: Enable the transparency color key	RW	0x0
9:8	INTERLEAVED3DMODE	Define which layer contributes to odd/even lines of the line interleaving 3D format 0x0: No interleaving happens in the overlay manager 0x1: RESERVED 0x3: A even pixels (for all lines) have a contribution from even z-order pipes and odd pixels (for all lines) have a contribution from the odd z-order pipes 0x2: At even lines (all pixels) have a contribution from even z-order pipes and odd lines (all pixels) have a contribution from the odd z-order pipes.	RW	0x0
7:0	RESERVED		R	0x0

Table 8-266. Register Call Summary for Register DISPC_OVR_CONFIG

Display Controller

- DISPC Overlay Transparency Color Keys: [0][1][2]
- DISPC Extended 3D Support - Line Alternative Format: [3][4]
- DISPC_OVR Register Summary: [5]

Table 8-267. DISPC_OVR_DEFAULT_COLOR

Address Offset	0x0000 0008		
Physical Address	0x5801 A808 0x5801 A908	**Instance**	DISPC_OVR1 DISPC_OVR2
Description	The control register configures the default solid background color bits [31:0]. Shadow register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DEFAULTCOLOR																															

Bits	Field Name	Description	Type	Reset
31:0	DEFAULTCOLOR	48-bit ARGB color value to specify the default solid color to display when there is no data from the overlays. Only [31:0] is defined in this register. Refer to DEFAULT_COLOR2 for [47:32] bits.	RW	0x0

Table 8-268. Register Call Summary for Register DISPC_OVR_DEFAULT_COLOR

Display Controller

- DISPC Overlay Priority Rule: [0]
- DISPC VP1 Configuration for TV Support: [1]
- DISPC_OVR Register Summary: [2]

Table 8-269. DISPC_OVR_DEFAULT_COLOR2

Address Offset	0x0000 000C		
Physical Address	0x5801 A80C 0x5801 A90C	**Instance**	DISPC_OVR1 DISPC_OVR2
Description	The control register configures the default solid background color bits [47:32]. Shadow register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																DEFAULTCOLOR															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0
15:0	DEFAULTCOLOR	48-bit ARGB color value to specify the default solid color to display when there is no data from the overlays. Only [47:32] is defined in this register. refer to DEFAULT_COLOR for [31:0] bits	RW	0x0

Table 8-270. Register Call Summary for Register DISPC_OVR_DEFAULT_COLOR2

Display Controller

- DISPC VP1 Configuration for TV Support: [0]
- DISPC_OVR Register Summary: [1]

Table 8-271. DISPC_OVR_TRANS_COLOR_MAX

Address Offset	0x0000 0010		
Physical Address	0x5801 A810 0x5801 A910	**Instance**	DISPC_OVR1 DISPC_OVR2
Description	The register sets the max transparency color value for the overlays. Shadow register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRANSCOLORKEY																															

Bits	Field Name	Description	Type	Reset
31:0	TRANSCOLORKEY	[31:0] Transparency Color Key Value in 36-bit RGB format	RW	0x0

Table 8-272. Register Call Summary for Register DISPC_OVR_TRANS_COLOR_MAX

Display Controller

- DISPC Overlay Transparency Color Keys: [0]
- DISPC_OVR Register Summary: [1]

Table 8-273. DISPC_OVR_TRANS_COLOR_MAX2

Address Offset	0x0000 0014		
Physical Address	0x5801 A814 0x5801 A914	**Instance**	DISPC_OVR1 DISPC_OVR2
Description	The register sets the max transparency color value for the overlays. Shadow register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								TRANSCOLORKEY							

Bits	Field Name	Description	Type	Reset
31:4	RESERVED		R	0x0
3:0	TRANSCOLORKEY	[35:32] Transparency Color Key Value in 36-bit RGB format	RW	0x0

Table 8-274. Register Call Summary for Register DISPC_OVR_TRANS_COLOR_MAX2

Display Controller

- DISPC Overlay Transparency Color Keys: [0]
- DISPC_OVR Register Summary: [1]

Table 8-275. DISPC_OVR_TRANS_COLOR_MIN

Address Offset	0x0000 0018		
Physical Address	0x5801 A818 0x5801 A918	**Instance**	DISPC_OVR1 DISPC_OVR2
Description	The register sets the min transparency color value for the overlays. Shadow register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRANSCOLORKEY																															

Bits	Field Name	Description	Type	Reset
31:0	TRANSCOLORKEY	[31:0] Transparency Color Key Value in 36-bit RGB format	RW	0x0

Table 8-276. Register Call Summary for Register DISPC_OVR_TRANS_COLOR_MIN

Display Controller

- DISPC Overlay Transparency Color Keys: [0]
- DISPC_OVR Register Summary: [1]

Table 8-277. DISPC_OVR_TRANS_COLOR_MIN2

Address Offset	0x0000 001C		
Physical Address	0x5801 A81C 0x5801 A91C	**Instance**	DISPC_OVR1 DISPC_OVR2
Description	The register sets the min transparency color value for the overlays. Shadow register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												TRANSCOLORKEY			

Bits	Field Name	Description	Type	Reset
31:4	RESERVED		R	0x0
3:0	TRANSCOLORKEY	[35:32] Transparency Color Key Value in 36-bit RGB format	RW	0x0

Table 8-278. Register Call Summary for Register DISPC_OVR_TRANS_COLOR_MIN2

Display Controller
- DISPC Overlay Transparency Color Keys: [0]
- DISPC_OVR Register Summary: [1]

8.2.5.7 DISPC_VP1 Registers

8.2.5.7.1 DISPC_VP1 Register Summary

Table 8-279. DISPC_VP1 Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	DISPC_VP1 L3_MAIN Physical Address
DISPC_VP1_CONFIG	RW	32	0x0000 0000	0x5801 AC00
DISPC_VP1_CONTROL	RW	32	0x0000 0004	0x5801 AC04
DISPC_VP1_CPR_COEF_B	RW	32	0x0000 0008	0x5801 AC08
DISPC_VP1_CPR_COEF_G	RW	32	0x0000 000C	0x5801 AC0C
DISPC_VP1_CPR_COEF_R	RW	32	0x0000 0010	0x5801 AC10
DISPC_VP1_DATA_CYCLE_I [(1)]	RW	32	0x0000 0014 + (0x4 * I)	0x5801 AC14 + (0x4 * I)
DISPC_VP1_GAMMA_TABLE	W	32	0x0000 0020	0x5801 AC20
DISPC_VP1_IRQENABLE	RW	32	0x0000 003C	0x5801 AC3C
DISPC_VP1_IRQSTATUS	RW	32	0x0000 0040	0x5801 AC40
DISPC_VP1_LINE_NUMBER	RW	32	0x0000 0044	0x5801 AC44
DISPC_VP1_LINE_STATUS	R	32	0x0000 0048	0x5801 AC48
DISPC_VP1_POL_FREQ	RW	32	0x0000 004C	0x5801 AC4C
DISPC_VP1_SIZE_SCREEN	RW	32	0x0000 0050	0x5801 AC50
DISPC_VP1_TIMING_H	RW	32	0x0000 0054	0x5801 AC54
DISPC_VP1_TIMING_V	RW	32	0x0000 0058	0x5801 AC58

(1) I = 0 to 2

8.2.5.7.2 DISPC_VP1 Register Description

Table 8-280. DISPC_VP1_CONFIG

Address Offset	0x0000 0000		
Physical Address	0x5801 AC00	**Instance**	DISPC_VP1
Description	The control register configures the Display Controller module for the VP output. Shadow register. **Note: DSI, RFBI and HDMI are not supported in this family of devices.**		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED						FULLRANGE	COLORCONVENABLE	FIDFIRST	OUTPUTMODEENABLE	BT1120ENABLE	BT656ENABLE	RESERVED			BUFFERHANDSHAKE	CPR	RESERVED						EXTERNALSYNCEN	VSYNCGATED	HSYNCGATED	PIXELCLOCKGATED	PIXELDATAGATED	HDMIMODE	GAMMAENABLE	DATAENABLEGATED	PIXELGATED

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	FULLRANGE	Color Space Conversion full range setting. 0x0: Limited range selected. 0x1: Full range selected.	RW	0x0
24	COLORCONVENABLE	Enable the color space conversion. It shall be reset when CPR bit-field is set to 0x1. 0x0: Disable Color Space Conversion RGB to YUV 0x1: Enable Color Space Conversion RGB to YUV	RW	0x0
23	FIDFIRST	Selects the first field to output in case of interlace mode. In case of progressive mode, the value is not used. 0x0: First field is even. 0x1: Odd field is first.	RW	0x0
22	OUTPUTMODEENABLE	Selects between progressive and interlace mode for the VP output. 0x0: Progressive mode selected. 0x1: Interlace mode selected.	RW	0x0
21	BT1120ENABLE	Selects BT-1120 format on the VP output. It is not posssible to enable BT656 and BT1120 at the same time on the same VP output. 0x0: BT-1120 is disabled. 0x1: BT-1120 is enabled.	RW	0x0
20	BT656ENABLE	Selects BT-656 format on the VP output. It is not posssible to enable BT656 and BT1120 at the same time on the same VP output. 0x0: BT-656 is disabled. 0x1: BT-656 is enabled.	RW	0x0
19:17	RESERVED	Write 0's for future compatibility. Reads return 0	R	0x0

Bits	Field Name	Description	Type	Reset
16	BUFFERHANDSHAKE	Controls the handsSHAKE between DMA buffer and STALL signal in order to prevent from underflow. The bit shall be set to 0 when the module is not in STALL mode. 0x0: Only the STALL signal (generated by RFBI or DSI2 depending on which IP uses the LCD output) is used regardless of the DMA buffer fullness information in order to provide data to the RFBI or DSI2 module. 0x1: The STALL signal (generated by RFBI or DSI2 depending on which IP uses the LCD output) is used in combination with the DMA buffer fullness information in order to provide data to the RFBI or DSI2 module only when it does not generated buffer underflow.	RW	0x0
15	CPR	Color Phase Rotation Control VP output). It shall be reset when ColorConvEnable bit-field is set to 1. Shadow bit-field. 0x0: Color Phase Rotation Disabled 0x1: Color Phase Rotation Enabled	RW	0x0
14:9	RESERVED	Write 0's for future compatibility. Reads return 0	R	0x0
8	EXTERNALSYNCEN	Selects between external sync and internal sync mode for the VP output. 0x0: Internal sync mode selected 0x1: External sync mode selected.	RW	0x0
7	VSYNCGATED	VSYNC Gated Enabled (VP output) Shadow bit-field. 0x0: VSYNC Gated Disabled 0x1: VSYNC Gated Enabled	RW	0x0
6	HSYNCGATED	HSYNC Gated Enabled (VP output) Shadow bit-field. 0x0: HSYNC Gated Disabled 0x1: HSYNC Gated Enabled	RW	0x0
5	PIXELCLOCKGATED	Pixel Clock Gated Enabled (VP output) Shadow bit-field. 0x0: Pixel Clock Gated Disabled 0x1: Pixel Clock Gated Enabled	RW	0x0
4	PIXELDATAGATED	Pixel Data Gated Enabled (VP output) Shadow bit-field. 0x0: Pixel Data Gated Disabled 0x1: Pixel Data Gated Enabled	RW	0x0
3	HDMIMODE	Configures the timing generator in HDMI compatible mode to generate same timings as HDMI wrapper timings. 0x0: Disable HDMI mode. 0x1: Enable HDMI mode timings: - vertical FSM starts with the VFP period (instead of VSYNC) - last frame ends with the last pixel of data state.	RW	0x0
2	GAMMAENABLE	Enable the gamma Shadow bit-field. 0x0: Gamma disabled 0x1: Gamma enabled	RW	0x0
1	DATAENABLEGATED	DE Gated Enable Shadow bit-field. 0x0: DE signal is not gated 0x1: DE signal is gated.	RW	0x0
0	PIXELGATED	Pixel Gated Enable Shadow bit-field. 0x0: Pixel clock always toggles (only in TFT mode) 0x1: Pixel clock only toggles when there is valid data to display. (only in TFT mode)	RW	0x0

Table 8-281. Register Call Summary for Register DISPC_VP1_CONFIG

Display Controller

- DISPC Clock Configuration: [0]
- DISPC VP1 Gamma Correction Unit: [1]
- DISPC VP1 Color Phase Rotation Unit: [2]
- DISPC VP1 Color Space Conversion: [3][4][5]
- DISPC VP1 BT.656 and BT.1120 Modes: [6]
- DISPC VP1 Timing Generator and Panel Settings: [7][8][9][10][11]
- DISPC_VP1 Register Summary: [12]

Table 8-282. DISPC_VP1_CONTROL

Address Offset	0x0000 0004		
Physical Address	0x5801 AC04	**Instance**	DISPC_VP1
Description	The control register configures the Display Controller module for the VP output.		
Type	RW		

31 30	29 28 27	26 25	24	23 22 21	20	19 18 17 16	15 14	13	12	11	10 9 8	7	6	5	4 3 2	1	0
SPATIALTEMPORALDITHERINGFRAMES	RESERVED	TDMUNUSEDBITS	TDMCYCLEFORMAT	TDMPARALLELMODE	TDMENABLE	RESERVED	HT	RESERVED	RESERVED	STALLMODE	DATALINES	STDITHERENABLE	RESERVED	GOBIT	RESERVED	VPPROGLINENUMBERMODULO	VPENABLE

Bits	Field Name	Description	Type	Reset
31:30	SPATIALTEMPORALDITHERINGFRAMES	Spatial/Temporal dithering number of frames for the VP output. Shadow bit-field. 0x0: Spatial only 0x1: Spatial and temporal over 2 frames 0x3: Reserved 0x2: Spatial and temporal over 4 frames	RW	0x0
29:27	RESERVED	Write 0's for future compatibility. Reads return 0	R	0x0
26:25	TDMUNUSEDBITS	State of unused bits (TDM mode only) for the VP output. Shadow bit-field. 0x0: low level (0) 0x1: high level (1) 0x3: reserved 0x2: unchanged from previous state	RW	0x0
24:23	TDMCYCLEFORMAT	Cycle format (TDM mode only) for the VP output. Shadow bit-field. 0x0: 1 cycle for 1 pixel 0x1: 2 cycles for 1 pixel 0x3: 3 cycles for 2 pixels 0x2: 3 cycles for 1 pixel	RW	0x0

Bits	Field Name	Description	Type	Reset
22:21	TDMPARALLELMODE	Output Interface width (TDM mode only) for the VP output. Shadow bit-field. 0x0: 8-bit parallel output interface selected 0x1: 9-bit parallel output interface selected 0x3: 16-bit parallel output interface selected 0x2: 12-bit parallel output interface selected	RW	0x0
20	TDMENABLE	Enable the multiple cycle format for the VP output. Shadow bit-field. 0x0: TDM disabled 0x1: TDM enabled	RW	0x0
19:17	RESERVED		R	0x0
16:14	HT	Hold Time for VP output. Shadow bit-field. Encoded value (from 1 to 8) to specify the number of external digital clock periods to hold the data (programmed value = value minus one)	RW	0x0
13	RESERVED		R	0x0
12	RESERVED		R	0x0
11	STALLMODE	STALL Mode for the VP output. Shadow bit-field. 0x0: Normal mode selected 0x1: STALL mode selected. The Display Controller sends the data without considering the VSYNC/HSYNC. The VP output is disabled at the end of the transfer of the frame. The S/W has to re-enable the VP output in order to generate a new frame.	RW	0x0
10:8	DATALINES	Width of the data bus on VP output. Shadow bit-field. 0x1: 16-bit output aligned on the LSB of the pixel data interface 0x0: 12-bit output aligned on the LSB of the pixel data interface 0x2: 18-bit output aligned on the LSB of the pixel data interface 0x4: 30-bit output aligned on the LSB of the pixel data interface 0x5: 36-bit output aligned on the LSB of the pixel data interface 0x3: 24-bit output aligned on the LSB of the pixel data interface	RW	0x0
7	STDITHERENABLE	Spatial Temporal dithering enable for the VP output Shadow bit-field. 0x0: Spatial/Temporal dithering logic disabled 0x1: Spatial/Temporal dithering logic enabled	RW	0x0
6	RESERVED		R	0x0
5	GOBIT	GO Command for the VP output. It is used to synchronized the pipelines associated with the VP output. wr:immediate 0x0: The hardware has finished updating the internal shadow registers of the pipeline(s) connected to the VP output using the user values. The hardware resets the bit when the update is completed. 0x1: The user has finished to program the shadow registers of the pipeline(s) associated with the VP output and the hardware can update the internal registers at the VFP start period	RW	0x0
4:2	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
1	VPPROGLINENUMBERMODULO	Enable the modulo of the line number interrupt generation 0x0: Disable modulo 0x1: Enable Modulo	RW	0x0
0	VPENABLE	Enable the video port output wr:immediate 0x0: VP output disabled (at the end of the frame when the bit is reset) 0x1: VP output enabled	RW	0x0

Table 8-283. Register Call Summary for Register DISPC_VP1_CONTROL

Display Controller
- DISPC VP1 Output and Data Formats: [0][1]
- DISPC Graphics Pipeline: [2][3]
- DISPC Video Pipelines: [4][5]
- Region-Based Mechanism for a Single Region Write-Back: [6][7]
- DISPC VP1 Spatial/Temporal Dithering: [8][9]
- DISPC VP1 Multiple Cycle Output Format (TDM): [10][11][12][13][14]
- DISPC VP1 Configuration for TV Support: [15][16]
- DISPC Shadow Registers: [17]
- DISPC_VP1 Register Summary: [18]

Table 8-284. DISPC_VP1_CPR_COEF_B

Address Offset	0x0000 0008		
Physical Address	0x5801 AC08	**Instance**	DISPC_VP1
Description	The register configures the color phase rotation matrix coefficients for the Blue component. Shadow register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BR										RESERVED	BG										RESERVED	BB									

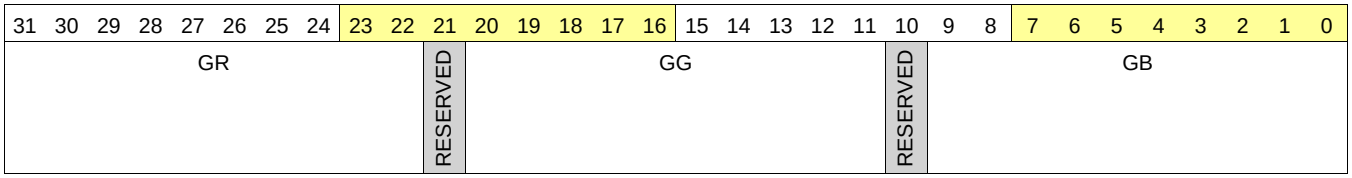
Bits	Field Name	Description	Type	Reset
31:22	BR	BR Coefficient Encoded signed value (from -512 to 511).	RW	0x0
21	RESERVED	Write 0's for future compatibility. Reads return 0	R	0x0
20:11	BG	BG Coefficient Encoded signed value (from -512 to 511).	RW	0x0
10	RESERVED	Write 0's for future compatibility. Reads return 0	R	0x0
9:0	BB	BB Coefficient Encoded signed value (from -512 to 511).	RW	0x0

Table 8-285. Register Call Summary for Register DISPC_VP1_CPR_COEF_B

Display Controller
- DISPC VP1 Color Phase Rotation Unit: [0][1]
- DISPC_VP1 Register Summary: [2]

Table 8-286. DISPC_VP1_CPR_COEF_G

Address Offset	0x0000 000C		
Physical Address	0x5801 AC0C	**Instance**	DISPC_VP1
Description	The register configures the color phase rotation matrix coefficients for the Green component. Shadow register.		
Type	RW		



Bits	Field Name	Description	Type	Reset
31:22	GR	GR Coefficient Encoded signed value (from -512 to 511).	RW	0x0
21	RESERVED	Write 0's for future compatibility. Reads return 0	R	0x0
20:11	GG	GG Coefficient Encoded signed value (from -512 to 511).	RW	0x0
10	RESERVED	Write 0's for future compatibility. Reads return 0	R	0x0
9:0	GB	GB Coefficient Encoded signed value (from -512 to 511).	RW	0x0

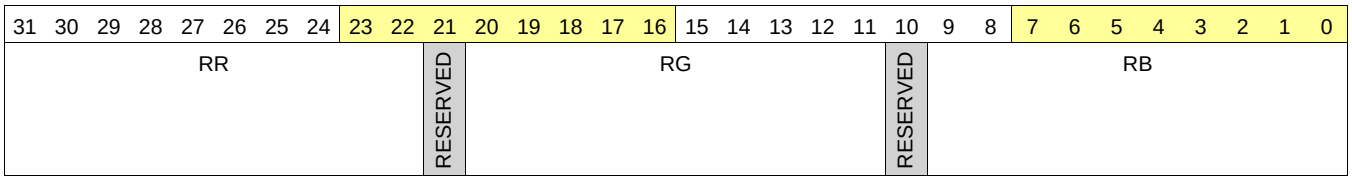
Table 8-287. Register Call Summary for Register DISPC_VP1_CPR_COEF_G

Display Controller

- DISPC VP1 Color Phase Rotation Unit: [0][1]
- DISPC_VP1 Register Summary: [2]

Table 8-288. DISPC_VP1_CPR_COEF_R

Address Offset	0x0000 0010		
Physical Address	0x5801 AC10	**Instance**	DISPC_VP1
Description	The register configures the color phase rotation matrix coefficients for the Red component. Shadow register.		
Type	RW		



Bits	Field Name	Description	Type	Reset
31:22	RR	RR Coefficient Encoded signed value (from -512 to 511).	RW	0x0
21	RESERVED	Write 0's for future compatibility. Reads return 0	R	0x0
20:11	RG	RG Coefficient Encoded signed value (from -512 to 511).	RW	0x0
10	RESERVED	Write 0's for future compatibility. Reads return 0	R	0x0
9:0	RB	RB Coefficient Encoded signed value (from -512 to 511).	RW	0x0

Table 8-289. Register Call Summary for Register DISPC_VP1_CPR_COEF_R

Display Controller

- DISPC VP1 Color Phase Rotation Unit: [0][1]
- DISPC_VP1 Register Summary: [2]

Table 8-290. DISPC_VP1_DATA_CYCLE_I

Address Offset	0x0000 0014		
Physical Address	0x5801 AC14 + (0x4 * I)	**Instance**	DISPC_VP1
Description	The control register configures the output data format over up to 3 cycles. Shadow register.		
Type	RW		

31 30 29 28	27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
RESERVED	BITALIGNMENTPIXEL2	RESERVED / NBBITSPIXEL2	RESERVED	BITALIGNMENTPIXEL1 / RESERVED / NBBITSPIXEL1

Bits	Field Name	Description	Type	Reset
31:28	RESERVED	Write 0's for future compatibility. Reads return 0	R	0x0
27:24	BITALIGNMENTPIXEL2	Bit alignment Alignment of the bits from pixel#2 on the output interface	RW	0x0
23:21	RESERVED	Write 0's for future compatibility. Reads return 0	R	0x0
20:16	NBBITSPIXEL2	Number of bits Number of bits from the pixel #2 (value from 0 to 16 bits). The values from 17 to 31 are invalid.	RW	0x0
15:12	RESERVED	Write 0's for future compatibility. Reads return 0	R	0x0
11:8	BITALIGNMENTPIXEL1	Bit alignment Alignment of the bits from pixel#1 on the output interface	RW	0x0
7:5	RESERVED	Write 0's for future compatibility. Reads return 0	R	0x0
4:0	NBBITSPIXEL1	Number of bits Number of bits from the pixel #1 (value from 0 to 16 bits). The values from 17 to 31 are invalid.	RW	0x0

Table 8-291. Register Call Summary for Register DISPC_VP1_DATA_CYCLE_I

Display Controller

- DISPC_VP1 Register Summary: [0]

Table 8-292. DISPC_VP1_GAMMA_TABLE

Address Offset	0x0000 0020		
Physical Address	0x5801 AC20	**Instance**	DISPC_VP1
Description	The register configures the gamma table on VP output.		
Type	W		

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
INDEX	VALUE_R	VALUE_G	VALUE_B

Bits	Field Name	Description	Type	Reset
31:24	INDEX	Defines the location in the table where the bit-field VALUE is stored	W	0x0
23:16	VALUE_R	8-bit value used to defined the value to be stored in the gamma table	W	0x0
15:8	VALUE_G	8-bit value used to defined the value to be stored in the gamma table	W	0x0
7:0	VALUE_B	8-bit value used to defined the value to be stored in the gamma table	W	0x0

Table 8-293. Register Call Summary for Register DISPC_VP1_GAMMA_TABLE

Display Controller

- DISPC VP1 Gamma Correction Unit: [0]
- DISPC_VP1 Register Summary: [1]

Table 8-294. DISPC_VP1_IRQENABLE

Address Offset	0x0000 003C		
Physical Address	0x5801 AC3C	**Instance**	DISPC_VP1
Description	This register allows to mask/unmask the module internal sources of interrupt, on an event-by-event basis		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																											VPSYNCLOST_EN	VPPROGRAMMEDLINENUMBER_EN	VPVSYNC_ODD_EN	VPVSYNC_EN	VPFRAMEDONE_EN

Bits	Field Name	Description	Type	Reset
31:5	RESERVED		R	0x0
4	VPSYNCLOST_EN	Synchronization Lost for Video Port. 0x0: SyncLost for the primary VP output is masked 0x1: SyncLost for the primary VP output generates an interrupt when it occurs	RW	0x0
3	VPPROGRAMMEDLINENUMBER_EN	Programmed Line Number. It indicates that the scan of the display has reached the programmed user line number. 0x0: ProgrammedLineNumber is masked 0x1: ProgrammedLineNumber generates an interrupt when it occurs	RW	0x0
2	VPVSYNC_ODD_EN	VSYNC for odd field from interlace mode only. 0x0: EVSYNC_ODD for the VP output is masked 0x1: EVSYNC_ODD for the VP output generates an interrupt when it occurs	RW	0x0
1	VPVSYNC_EN	Vertical Synchronization for VP. 0x0: VSYNC for the primary VP output is masked 0x1: VSYNC for the primary VP output generates an interrupt when it occurs	RW	0x0
0	VPFRAMEDONE_EN	Frame Done for Video Port. VP output has been disabled by user. All the data have been sent. 0x0: FrameDone for the primary VP output is masked 0x1: FrameDone for the primary VP output generates an interrupt when it occurs	RW	0x0

Table 8-295. Register Call Summary for Register DISPC_VP1_IRQENABLE

Display Controller

- DISPC Interrupt Requests: [0]
- DISPC_VP1 Register Summary: [1]

Table 8-296. DISPC_VP1_IRQSTATUS

Address Offset	0x0000 0040		
Physical Address	0x5801 AC40	**Instance**	DISPC_VP1
Description	This register regroups all the status of the module internal events that generate an interrupt. Write 1 to a given bit resets this bit		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																											VPSYNCLOST_IRQ	VPPROGRAMMEDLINENUMBER_IRQ	VPVSYNC_ODD_IRQ	VPVSYNC_IRQ	VPFRAMEDONE_IRQ

Bits	Field Name	Description	Type	Reset
31:5	RESERVED		R	0x0
4	VPSYNCLOST_IRQ	Synchronization Lost on VP output. The required data are not output at the correct time due to too short blanking periods or stall of at least one pipelines associated with VP output. 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW	0x0
3	VPPROGRAMMEDLINENUMBER_IRQ	Programmed Line Number. It indicates that the scan of the display has reached the programmed user line number. 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW	0x0
2	VPVSYNC_ODD_IRQ	VSYNC for odd field from interlace mode only. 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW	0x0
1	VPVSYNC_IRQ	Vertical Synchronization for VP output. It is used as VSYNC_EVEN in case of interlace mode. 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW	0x0

Bits	Field Name	Description	Type	Reset
0	VPFRAMEDONE_IRQ	Frame Done for VP. VP output has been disabled by user. All the data have been sent. 0x0: READS: Event is false. WRITES: Status bit unchanged. 0x1: READS: Event is true (pending). WRITES: Status bit is reset.	RW	0x0

Table 8-297. Register Call Summary for Register DISPC_VP1_IRQSTATUS

Display Controller
- DISPC Interrupt Requests: [0]
- DISPC_VP1 Register Summary: [1]

Table 8-298. DISPC_VP1_LINE_NUMBER

Address Offset	0x0000 0044		
Physical Address	0x5801 AC44	**Instance**	DISPC_VP1
Description	The control register indicates the panel display line number for the interrupt and the DMA request. Shadow register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								LINENUMBER							

Bits	Field Name	Description	Type	Reset
31:12	RESERVED		R	0x0
11:0	LINENUMBER	Display panel line number programming. Line number defines the line on which the programmable interrupt is generated and the DMA request occurs.	RW	0x0

Table 8-299. Register Call Summary for Register DISPC_VP1_LINE_NUMBER

Display Controller
- DISPC DMA Requests: [0]
- DISPC_VP1 Register Summary: [1]

Table 8-300. DISPC_VP1_LINE_STATUS

Address Offset	0x0000 0048		
Physical Address	0x5801 AC48	**Instance**	DISPC_VP1
Description	The control register indicates the panel display line number.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								LINENUMBER							

Bits	Field Name	Description	Type	Reset
31:12	RESERVED		R	0x0
11:0	LINENUMBER	Current display panel line number. Current display line number. The first active line has the value '0'. During blanking lines the line number is not incremented.	R	0x0

Table 8-301. Register Call Summary for Register DISPC_VP1_LINE_STATUS

Display Controller

- DISPC_VP1 Register Summary: [0]

Table 8-302. DISPC_VP1_POL_FREQ

Address Offset	0x0000 004C		
Physical Address	0x5801 AC4C	**Instance**	DISPC_VP1
Description	The register configures the signal configuration. Shadow register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED													ALIGN	ONOFF	RF	IEO	IPC	IHS	IVS	RESERVED											

Bits	Field Name	Description	Type	Reset
31:19	RESERVED	Write 0's for future compatibility. Reads return 0	R	0x0
18	ALIGN	Defines the alignment betwwen HSYNC and VSYNC assertion. 0x0: VSYNC and HSYNC are not aligned 0x1: VSYNC and HSYNC assertions are aligned.	RW	0x0
17	ONOFF	HSYNC/VSYNC Pixel clock Control On/Off 0x0: HSYNC and VSYNC are driven on opposite edges of pixel clock than pixel data 0x1: HSYNC and VSYNC are driven according to bit 16	RW	0x0
16	RF	Program HSYNC/VSYNC Rise or Fall 0x0: HSYNC and VSYNC are driven on falling edge of pixel clock (if bit 17 set to 1) 0x1: HSYNC and VSYNC are driven on rising edge of pixel clock (if bit 17 set to 1)	RW	0x0
15	IEO	Invert output enable 0x0: Ac-bias is active high (active display mode) 0x1: Ac-bias is active low (active display mode)	RW	0x0
14	IPC	Invert pixel clock 0x0: Data is driven on the VP data lines on the rising-edge of the pixel clock 0x1: Data is driven on the VP data lines on the falling-edge of the pixel clock	RW	0x0
13	IHS	Invert HSYNC 0x0: Line clock pin is active high and inactive low 0x1: Line clock pin is active low and inactive high	RW	0x0
12	IVS	Invert VSYNC 0x0: Frame clock pin is active high and inactive low 0x1: Frame clock pin is active low and inactive high	RW	0x0
11:0	RESERVED		R	0x0

Table 8-303. Register Call Summary for Register DISPC_VP1_POL_FREQ

Display Controller

- DISPC VP1 Active Marix Display Timing Diagrams: [0][1][2][3][4][5][6][7][8][9][10][11][12][13][14][15][16][17][18][19][20][21][22][23][24][25][26]
- DISPC VP1 Timing Generator and Panel Settings: [27][28][29][30][31]
- DISPC_VP1 Register Summary: [32]

Table 8-304. DISPC_VP1_SIZE_SCREEN

Address Offset	0x0000 0050		
Physical Address	0x5801 AC50	**Instance**	DISPC_VP1
Description	The register configures the panel size (horizontal and vertical). Shadow register. A delta value is used to indicate if the odd field has same vertical size as the even field or +/- one line.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								LPP								DELTA_LPP		RESERVED		PPL											

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:16	LPP	Lines per panel Encoded value (from 1 to 4096) to specify the number of lines per panel (program to value minus one).	RW	0x0
15:14	DELTA_LPP	Indicates the delta size value of the odd field compared to the even field 0x0: same size 0x1: odd size = even size +1 0x2: Odd size = even size -1	RW	0x0
13:12	RESERVED		R	0x0
11:0	PPL	Pixels per line Encoded value (from 1 to 4096) to specify the number of pixels contains within each line on the display (program to value minus one). In STALL mode, any value is valid. In non STALL mode, only values multiple of 8 pixels are valiid.	RW	0x0

Table 8-305. Register Call Summary for Register DISPC_VP1_SIZE_SCREEN

Display Controller

- DISPC VP1 Active Marix Display Timing Diagrams: [0][1]
- DISPC VP1 Timing Generator and Panel Settings: [2][3][4][5][6][7][8]
- DISPC VP1 Configuration for TV Support: [9][10][11][12][13][14]
- DISPC Extended 3D Support - Line Alternative Format: [15]
- DISPC_WB Register Description: [16]
- DISPC_VP1 Register Summary: [17]

Table 8-306. DISPC_VP1_TIMING_H

Address Offset	0x0000 0054		
Physical Address	0x5801 AC54	**Instance**	DISPC_VP1
Description	The register configures the timing logic for the HSYNC signal. Shadow register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
HBP																HFP								HSW							

Bits	Field Name	Description	Type	Reset
31:20	HBP	Horizontal Back Porch. Encoded value (from 1 to 4096) to specify the number of pixel clock periods to add to the beginning of a line transmission before the first set of pixels is output to the displaydisplay (program to value minus one). When in BT mode and interlaced, this field corresponds to the vertical field blanking No 2 for Even Field.	RW	0x0
19:8	HFP	Horizontal front porch. Encoded value (from 1 to 4096) to specify the number of pixel clock periods to add to the end of a line transmission before line clock is asserted display (program to value minus one). When in BT mode and interlaced, this field corresponds to the vertical field blanking No 1 for Even Field.	RW	0x0
7:0	HSW	Horizontal synchronization pulse width Encoded value (from 1 to 256) to specify the number of pixel clock periods to pulse the line clock at the end of each line display (program to value minus one). When in BT mode, this field corresponds to the LSB 8-bits of the 12-bit horizontal blanking(BT_HBLANK[11:0]= {VSW[3:0],HSW[7:0]}).	RW	0x0

Table 8-307. Register Call Summary for Register DISPC_VP1_TIMING_H

Display Controller

- DISPC VP1 Active Marix Display Timing Diagrams: [0][1][2]
- DISPC VP1 Timing Generator and Panel Settings: [3][4][5][6][7][8][9][10]
- DISPC_VP1 Register Summary: [11]

Table 8-308. DISPC_VP1_TIMING_V

Address Offset	0x0000 0058		
Physical Address	0x5801 AC58	**Instance**	DISPC_VP1
Description	The register configures the timing logic for the VSYNC signal. Shadow register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VBP								VFP																VSW							

Bits	Field Name	Description	Type	Reset
31:20	VBP	Vertical back porch Encoded value (from 0 to 4095) to specify the number of line clock periods to add to the beginning of a frame. When in BT mode and interlaced, this field corresponds to the vertical field blanking No 2 for Odd Field. When in BT and in progressive mode, this field corresponds to the Vertical frame blanking No 2 . before the first set of pixels is output to the display.	RW	0x0
19:8	VFP	Vertical front porch Encoded value (from 0 to 4095) to specify the number of line clock periods to add to the end of each frame. When in BT mode and interlaced, this field corresponds to the vertical field blanking No 1 for Odd Field. When in BT and in progressive mode, this field corresponds to the Vertical frame blanking No 2 .	RW	0x0
7:0	VSW	Vertical synchronization pulse width Encoded value (from 1 to 256) to specify the number of line clock periods (program to value minus one) to pulse the frame clock (VSYNC) pin at the end of each frame after the end of frame wait (VFP) period elapses. Frame clock uses as VSYNC signal in active mode. When in BT mode, the lsb 4-bits of this field (VSW[3:0]) corresponds to the MSB 4-bits of the 12-bit horizontal blanking(BT_HBLANK= {VSW[3:0],HSW[7:0]}).	RW	0x0

Table 8-309. Register Call Summary for Register DISPC_VP1_TIMING_V

Display Controller

- DISPC VP1 Active Marix Display Timing Diagrams: [0][1][2]
- DISPC VP1 Timing Generator and Panel Settings: [3][4][5][6][7][8][9][10][11][12]
- DISPC_VP1 Register Summary: [13]

8.3 Video Encoder

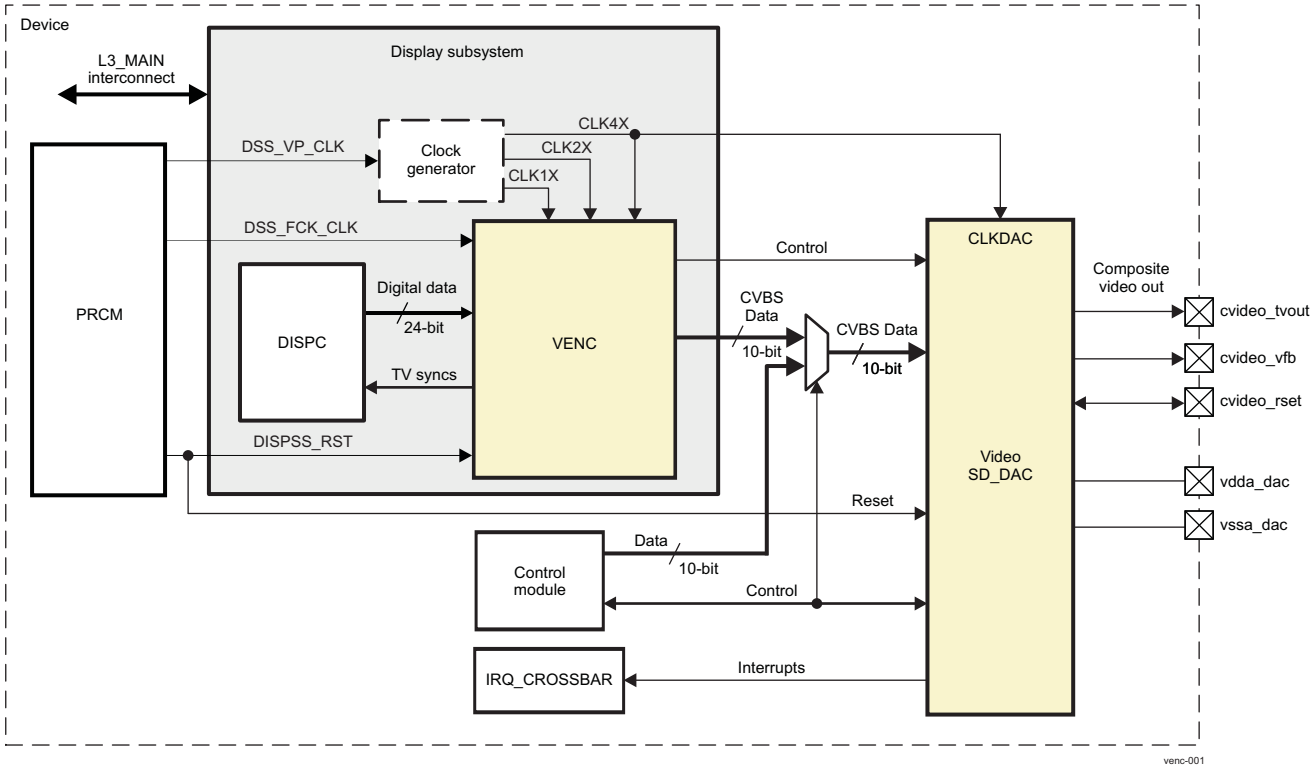
This section describes the video encoder (VENC) and associated video standard definition DAC (SD_DAC) modules for the device.

8.3.1 Video Encoder Overview

The VENC module provides the logic to display analog video on a TV set.

Figure 8-69 shows a block diagram of the video encoder and SD_DAC in the display subsystem.

Figure 8-69. Video Encoder Overview



The video encoder includes the following features:

- Supports the following standards for composite video (CVBS) :
  - NTSC-J, M
  - PAL-B, D, G, H, I
  - PAL-M
- Input data interface compatible with the following protocols:
  - 24-bit input bus compatible with external sync
  - RGB 4:4:4
- 10-bit interface for internal digital-to-analog converter (DAC) that supports:
  - 1.1V digital power supply, 1.8V analog power supply
  - 10-bit resolution
  - Sample rate of up to 60 megasamples per second (MSPS)
  - Supports Composite Video standart output
  - Supports TVOUT buffer bypass mode (DAC only mode)
  - Full-scale voltage output: 1.2 Vpp with a 75-Ω load
  - Internal TV-detect feature to detect both open and short conditions
  - Suitable for low-power wireless applications

- TV output data supports ITU-R BT 470-7 recommendation standard for consumer market
- Master clock input 13.5 MHz, 27 MHz (supports ITU-R 601 sampling for NTSC/PAL), and 54 MHz
- Programmable horizontal sync, vertical timing, and waveforms
- Programmable subcarrier frequency and SCH
- Internal test pattern generation (color bar, flat field, color burst)
- 2x/4x oversampling
- TV detection gating pulse generation
- Supports square pixel sampling (NTSC: 12.27 MHz, 24.54 MHz, 49.09 MHz PAL: 14.75 MHz, 29.5 MHz, 59 MHz)

CAUTION

In square pixel mode, an external clock generator is required to provide sampling frequencies.

8.3.2 Video Encoder Environment

To display analog video on a TV set, the path used is as follow:

- Display controller
- Video encoder
- 10-bit SD_DAC with video amplifier

The display controller module receives synchronization signals from the video encoder and synchronously sends pixel data to the video encoder with these signals. The digital output of the display controller is always a 24-bit RGB value based on a pixel request from the video encoder. For more information on the display controller settings and features, see Section 8.2, *Display Controller*.

The video encoder converts RGB video signals to conform to the NTSC/PAL standards analog video. The video encoder includes an integrated synchronization signal generator and a video digital-to-analog converter (SD_DAC) with video amplifiers, data manager, luma stage, chroma stage, modulator, and a control interface.

The video encoder also provides the synchronization signals to the display controller: VSYNC, active VIDeo (AVID), and field ID (FID).

NOTE: To enhance the TV color display, it is highly recommended to set the DSS.DSS_VENC_CTRL[1] DAC_DEMEN register bit. See Section 8.1.3, *Display Subsystem Register Manual*.

Figure 8-70 through Figure 8-72 show the TV display interface in composite mode.

Figure 8-70. Video Encoder Environment, Normal Mode DC-Coupling

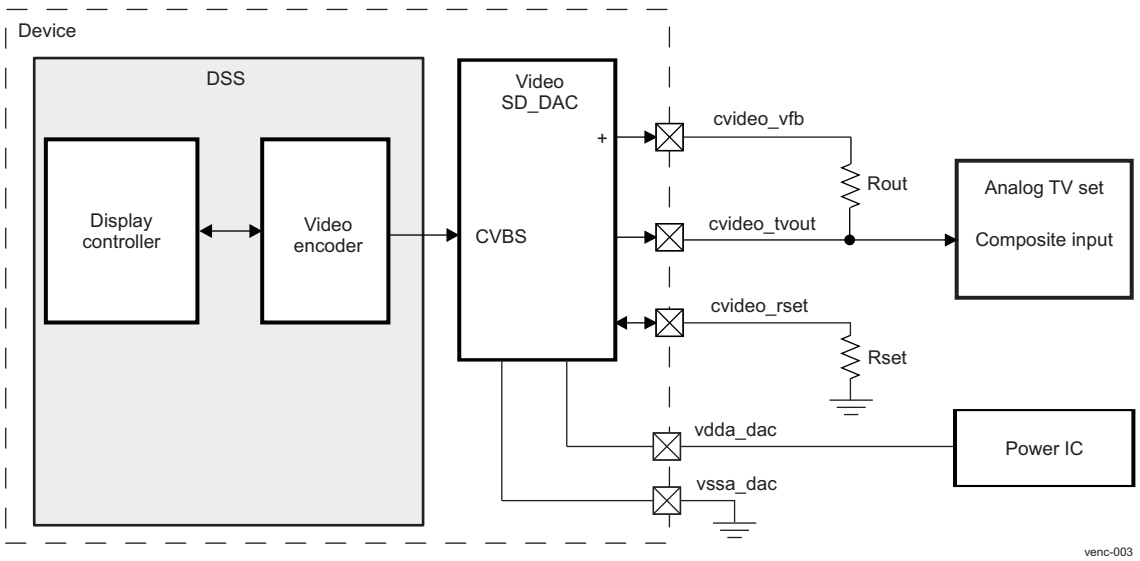


Figure 8-71. Video Encoder Environment, Normal Mode AC-Coupling

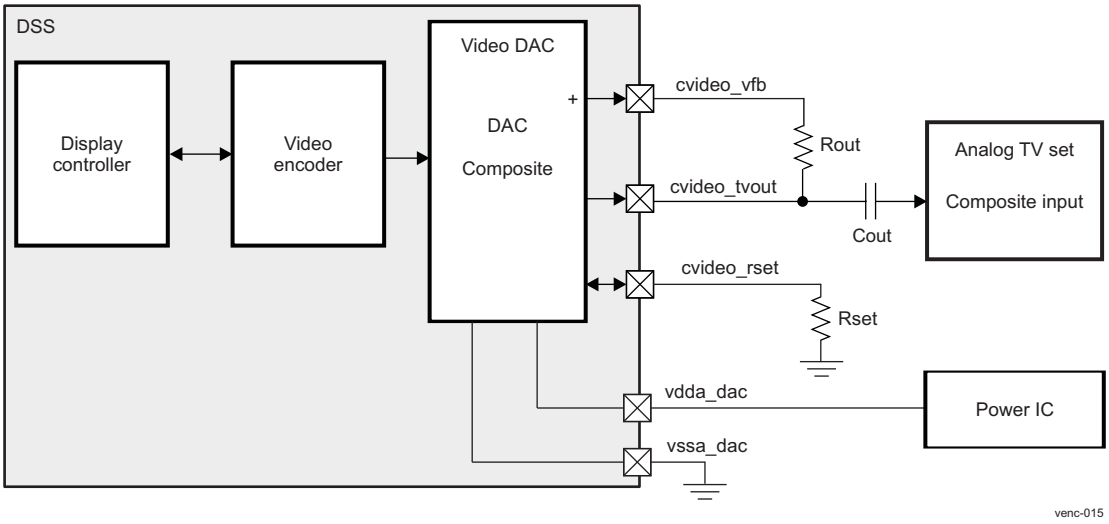


Figure 8-72. Video Encoder Environment, Bypass Mode

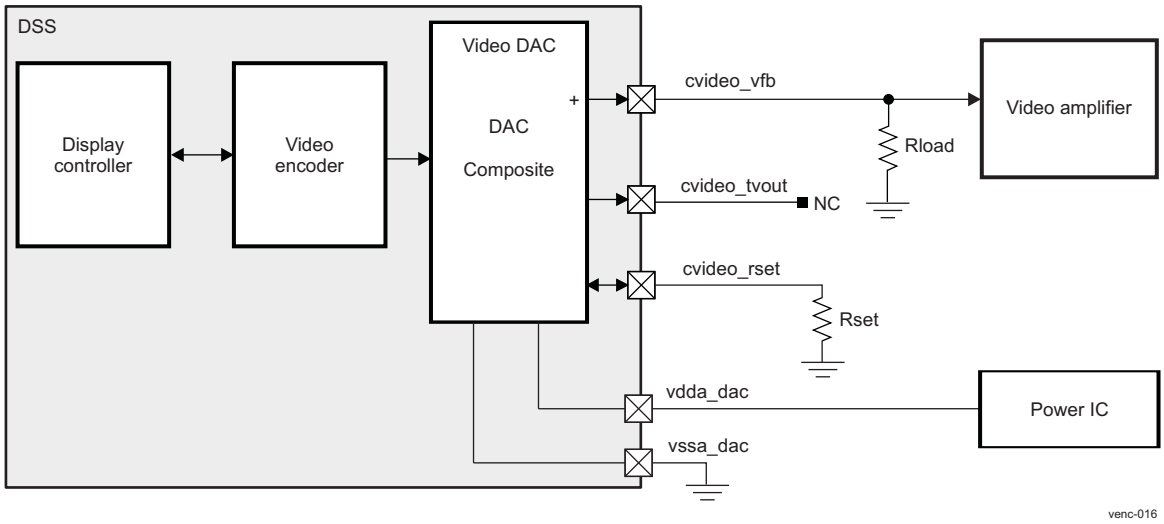


Table 8-310 describes the interface signals to/from the TV set for TV display support.

Table 8-310. TV Display Interface Pins

Pin Name	Type[1]	Description
cvideo_tvout	O	Composite video output. An external resistor (Rout) is connected between this node and the *cvideo_vfb* pin. The nominal value of Rout is 2700 Ω. This is the output node that drives the load (75 Ω). In AC-coupling mode, a series capacitor (Cout) must be connected. The nominal value for Cout is ≥ 200 µF.
cvideo_vfb	O	In Normal mode (video DAC+Buffer), it acts as the buffer feedback node. An external resistor (Rout) shall be connected between this node and *cvideo_tvout* pin. In TVOUT Bypass mode, it acts as the video DAC output. A load resistor (Rload) shall be connected to the pin to ground. The nominal value of Rload is 1500 Ω.

(1) I = input; O = output; power = power pin

Table 8-310. TV Display Interface Pins (continued)

Pin Name	Type[1]	Description
cvideo_rset	I/O	External resistor (Rset) shall be connected to this pin to set the reference current of the video DAC. The value of the resistor depends on the mode of operation (Normal or TVOUT Bypass). In Normal mode, the nominal value for Rset is 4700 Ω. In TVOUT Bypass mode, the nominal value for Rset is 10000 Ω.
vdda_dac	Power	Analog supply voltage for the video DAC
vssa_dac	Power	Analog ground for the video DAC

CAUTION

- *cvideo_tvout* is a very high-frequency analog signal and must be routed with extreme care. As a result, the path of this signal must be as short as possible, and as isolated as possible from other interfering signals.
- During board design, the onboard traces and parasites must be matched for the channel. *cvideo_vfb* pin is the most sensitive pins in the TV out system. Low onboard resistance and capacitance is required for the traces that connect the Rout to the *cvideo_vfb* and *cvideo_tvout* pins. The resistance on those trace affects output impedance matching. Therefore, Rout resistor is suggested to be placed as close as possible to the device pins. The onboard traces lead to the TV OUT pin must have a characteristic impedance of 75 Ohm starting from the closest possible place to the device pin output.
- If the TV output is not used, the following configuration for the SD_DAC pins must be applied:
  - vdda_dac: Must be grounded
  - vssa_dac: Must be grounded
  - cvideo_rset: Must be floating, left unconnected
  - cvideo_tvout: Must be floating, left unconnected
  - cvideo_vfb: Must be floating, left unconnected
- To avoid current leakage, the following register bits must be set to 0:
  - DSS.DSS_VENC_CTRL[2] DAC_POWERDN_BGZ
  - DSS.VENC_OUTPUT_CONTROL[1] COMPOSITE_ENABLE
  - CONTROL.CTRL_CORE_SD_DAC_CONTROL[3] TVOUTBYPASS

Table 8-311 lists the typical values of the external components for different modes of the TV display interface.

Table 8-311. Typical Values for Rout, Rset, Rload, and Cout

	Composite, DC-Coupled	Composite, AC-Coupled	Bypass Mode	Unit
Rout	2700	2700	N/A	Ω
Rset	4700	4700	10,000	Ω
Rload	N/A	N/A	1500	Ω
Cout	N/A	≥ 220	N/A	µF

8.3.3 Video Encoder Integration

This section describes the integration of the VENC and video SD_DAC modules, and details clocks, resets, hardware requests, and power modes.

Figure 8-73 shows the integration of the VENC and SD_DAC in the device.

Figure 8-73. Video Encoder Integration

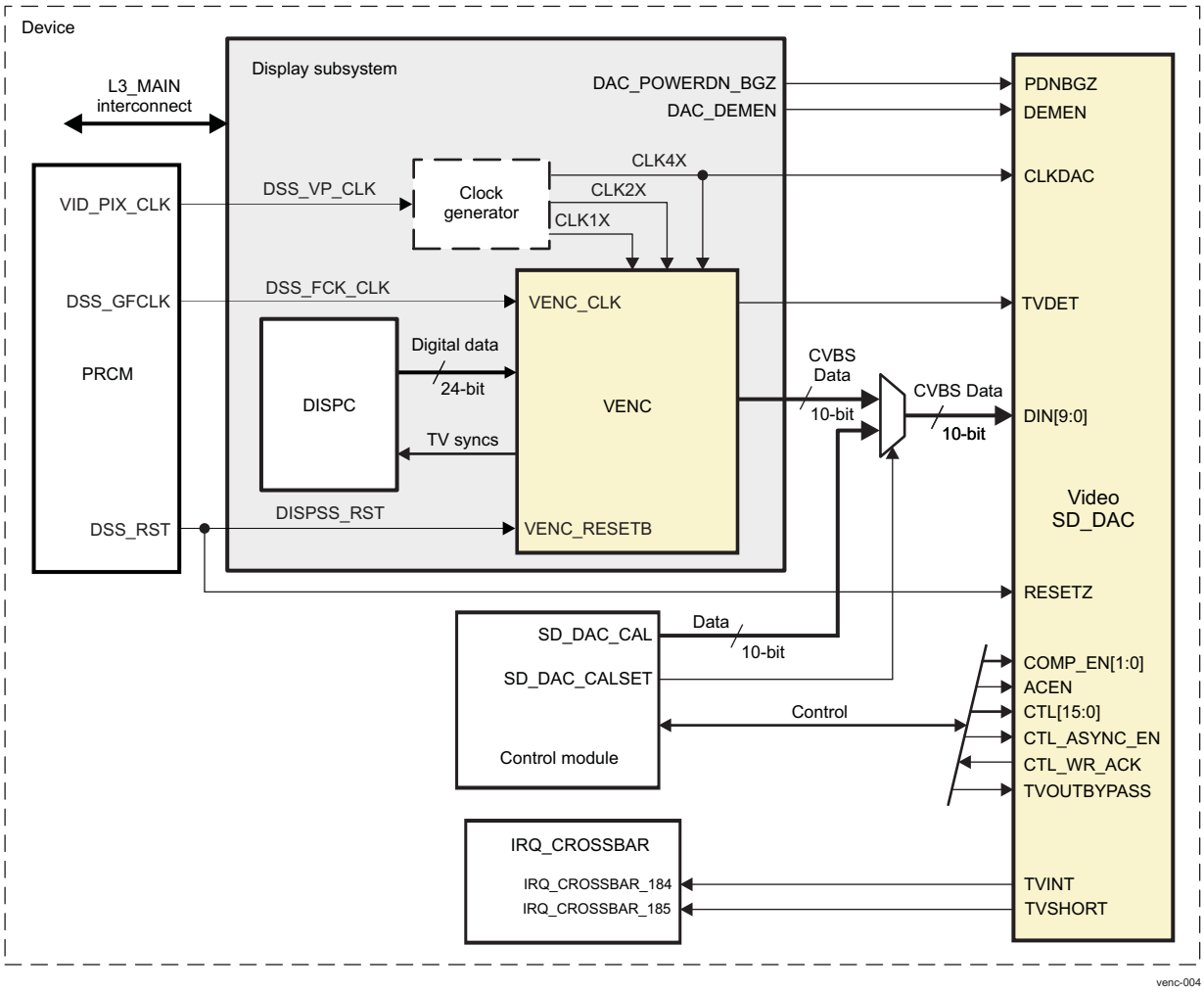


Table 8-312 shows the interrupt requests generated by the SD_DAC.

Table 8-312. Video SD_DAC Interrupt Requests

		Interrupt Requests		
Module Instance	**Source Signal Name**	**Destination IRQ_CROSSBAR Input**	**Default Mapping**	**Description**
SD_DAC	SD_DAC_IRQ_TVINT	IRQ_CROSSBAR_184	-	Interrupt for TV load detection from SD_DAC
	SD_DAC_IRQ_TVSHORT	IRQ_CROSSBAR_185	-	Interrupt for TV short detection from SD_DAC

NOTE: The "**Default Mapping**" column in Table 8-312 shows the default mapping of module IRQ source signals. These IRQ source signals can also be mapped to other lines of each device Interrupt controller through the IRQ_CROSSBAR module.
For more information about the IRQ_CROSSBAR module, see Section 13.4.6.4, *IRQ_CROSSBAR Module Functional Description*, in Chapter 13, *Control Module*.
For more information about the device interrupt controllers, see Chapter 12, *Interrupt Controllers*.

Table 8-313 describes the various Video encoder and SD_DAC clocks.

Table 8-313. Video Encoder and SD_DAC Clocks and Resets

Clocks				
Module Instance	**Destination Signal Name**	**Source Signal Name**	**Source**	**Description**
Video encoder	VENC_CLK	DSS_GFCLK	PRCM	VENC interface clock
	CLK1X CLK2X CLK4X	VID_PIX_CLK (indirect)	PRCM DSS clock generator	VENC functional clocks, generated on DSS level through division and gating. Refer to Section 8.1.2.2, *Display Subsystem Clocks*, for more details.
SD_DAC	CLKDAC	VID_PIX_CLK (indirect)	PRCM DSS clock generator	Functional clock for SD_DAC module. Gating control is available on DSS level. Refer to Section 8.1.2.2, *Display Subsystem Clocks*, for more details.
Resets				
Module Instance	**Destination Signal Name**	**Source Signal Name**	**Source**	**Description**
Video encoder SD_DAC	VENC_RESETB RESETZ	DSS_RST	PRCM	Reset signal from the PRCM module.

NOTE: For information about PRCM clock gating and management, see Chapter 3,*Power, Reset, and Clock Management*.

- **Video encoder functional clocks:**

  Three balanced clocks are provided to VENC module from DSS top level. Gating control for CLK4X clock is available through DSS.DSS_VENC_CTRL[0] VENC_CLOCK_4X_ENABLE bit (see Section 8.1.2.2, *Display Subsystem Clocks*).

Table 8-314. Digital Clock Division for the Video Encoder

VENC Clock Input	DSS Clock Generator Output
CLK4X	VID_PIX_CLK or 0 (gated)
CLK2X	VID_PIX_CLK/2
CLK1X	VID_PIX_CLK/4

NOTE: For proper video SD_DAC configuration, clock output CLK4X shall always be active by setting the DSS.DSS_VENC_CTRL[0] VENC_CLOCK_4X_ENABLE bit to 1 (see Section 8.1.3, *DSS Register Manual*).

CAUTION

DSS must be enabled first before VENC, in order the source clocks feeding VENC to be running before enabling the VENC.

VENC must be disabled before DSS, in order the source clocks feeding the VENC to be running before disabling the VENC. If DSS is disabled before VENC, the functional clocks will stop abruptly.

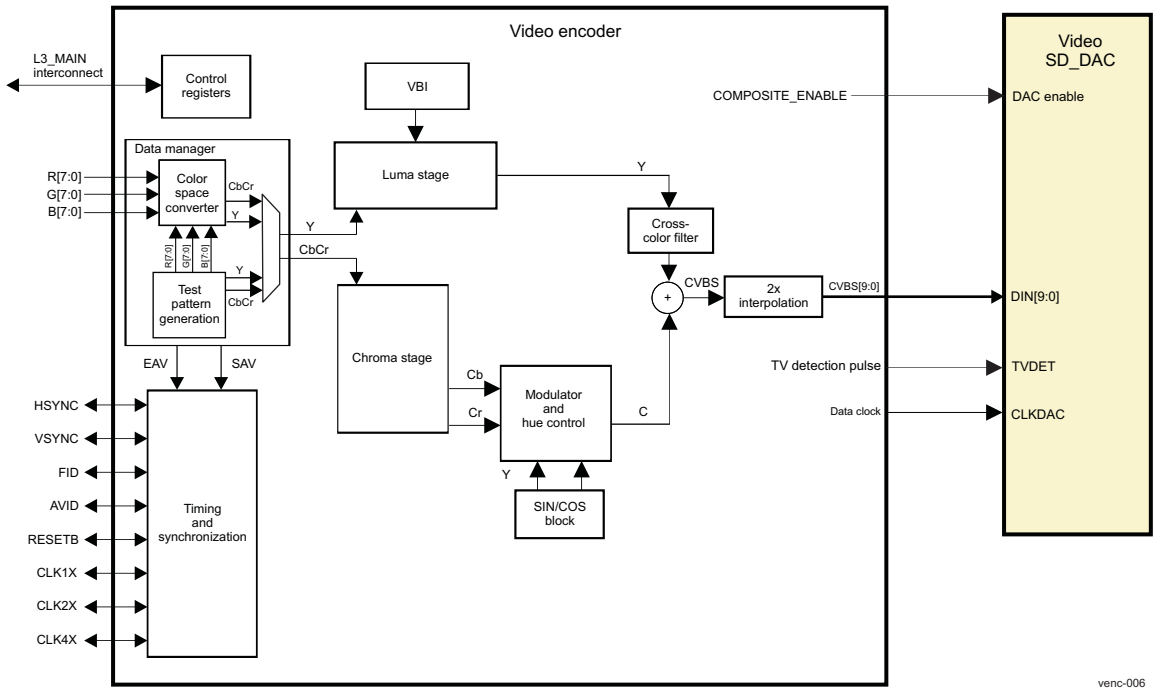
- **Video SD_DAC clock:**

  The video data sent from VENC to SD_DAC is latched on the positive edge of the CLKDAC clock.

8.3.4 Video Encoder Functional Description

Figure 8-74 shows an overview of the video encoder architecture.

Figure 8-74. Video Encoder Architecture Overview



8.3.4.1 Video Encoder Data Manager

The data manager allows user to select two sources of data inputs, by programming VENC_F_CONTROL[7:6] SVDS bitfield:

- Data coming from the Display Controller, when SVDS = 0x0.
- Data coming from the test pattern generation, when SVDS = 0x1 or 0x2 (default value).

The source is selected

8.3.4.1.1 Video Encoder Color Space Converter

In the display subsystem, the input format from the display controller is always 24-bit RGB coming from the display controller or the test pattern generation. The RGB-to-YCbCr color space converter converts the 24-bit RGB pixel data into 16-bit YCbCr 4:2:2 data.

8.3.4.1.2 Video Encoder Test Pattern Generation

For diagnostic purposes, the data manager can be forced to output 100/100 color bar RGB/YCbCr data. To use the test pattern generation, the VENC_F_CONTROL[7:6] SVDS bitfield shall be set to 0x1. If VENC_F_CONTROL[1:0] FMT bitfield is set to 0x0, then the 24-bits RGB test pattern data will be converted into 16-bit YCbCr 4:2:2 data by the Color Space Converter. Otherwise, if VENC_F_CONTROL[1:0] FMT bitfield is set to 0x2, then 16-bit YCbCr 4:2:2 test pattern data are outputed directly, the color space converted is bypassed. Table 8-315 resume the test paten generator data.

Table 8-315. Video Encoder 100/100 Color Bar Table

COLOR	R	G	B	Y	Cb	Cr
White	255	255	255	235	128	128
Yellow	255	255	0	210	16	146
Cyan	0	255	255	170	166	16

Table 8-315. Video Encoder 100/100 Color Bar Table (continued)

Green	0	255	0	145	54	34
Magenta	255	0	255	106	202	222
Red	255	0	0	81	90	240
Blue	0	0	255	41	240	110
Black	0	0	0	16	128	128

8.3.4.2 Video Encoder Luma Stage

The luma stage includes a luma pipeline delay, luma shaping, 2x interpolation filter, and luma variable delay. The luma pipeline delay block is used to match luma path length to chroma path length. In the luma gain shaper, a programmable gain is first applied to the luminance data output. The luminance gain is defined by the VENC_GAIN_Y register. Horizontal sync, vertical sync, and setup insertion are then performed.

Black level and blank level are programmable through the VENC_BLACK_LEVEL and VENC_BLANK_LEVEL registers. All the transition edges of the luminance signal, such as sync edges and active video edges, are properly shaped and filtered to keep the bandwidth within the standards.

After all required components of the luminance signal are added, the resulting signal is low-passed and interpolated to 2x-pixel rate. This 2x interpolation simplifies the external analog reconstruction filter design and improves the signal-to-noise ratio.

8.3.4.3 Video Encoder Chroma Stage

The chroma stage includes a low-pass filter, first-stage 2x interpolation, chroma gain shaper, and second-stage 2x interpolation. A pair of programmable gains adjusts the time-multiplexed U/V signal. The gains for U and V are independently controlled by the VENC_GAIN_U and VENC_GAIN_V register bits.

8.3.4.4 Video Encoder Subcarrier and Burst Generation

The encoder uses a 32-bit subcarrier increment to synthesize the subcarrier. The value of the subcarrier increments required to generate the desired subcarrier frequency for NTSC and PAL format is found by:

$S\_CARR = ROUND ([F_{sc}/F_{CLK2X}] \times 2^{32})$

where:

F_{sc} = Frequency of the subcarrier

F_{CLK2X} = Frequency of the internal video encoder

The VENC_S_CARR register controls the subcarrier frequency. The VENC_C_PHASE register controls the phase of the subcarrier. The phase of the color subcarrier is reset to VENC_C_PHASE. Table 8-316 presents the VENC_S_CARR register values depending the standard and pixel type used.

Table 8-316. VENC_S_CARR Register Recommended Values

Standard	Pixel Type	Subcarrier Frequency (Fsc) (MHz)	Fclkenc (MHz)	VENC_S_CARR Register Value (hexa)
NTSC-M, J	ITU-R601	3.579545	27	0x21F07C1F
PAL-M	ITU-R601	3.5756083125	27	0x21E6EFE3
PAL-B, D, G, H, I	ITU-R601	4.43361875	27	0x2A098ACB
NTSC-M, J	Square pixel	3.579545	24.5454(1)	0x25555555
PAL-M	Square pixel	3.579561149	24.5454(1)	0x1F15C01E
PAL-B, D, G, H, I	Square pixel	4.43361875	29.50(1)	0x26798C0C

(1) In square pixel mode, an external clock generator is needed to provide sampling frequencies (49.09 MHz for NTSC square pixel or 59 MHz for PAL square pixel).

The color subcarrier reset has four modes:

- No reset
- Reset every two lines
- Reset every two fields
- Reset every eight fields

The VENC_C_PHASE register can be used to adjust the SCH (subcarrier to horizontal sync phase). The VENC_BSTAMP_WSS_DATA[6:0] BSTAP bit field sets the amplitude of the color burst.

Phase alternation line refers to the encoding scheme in which the subcarrier alternates between two phases every scan line. Setting to 0x1, the VENC_M_CONTROL[1] PAL bit enables phase alternation line encoding. Otherwise, a normal subcarrier is generated. .Two possible alternation sequences are possible by setting the VENC_M_CONTROL[5] PALPHS bit.

8.3.4.5 Video Encoder Vertical Blanking Interval

8.3.4.5.1 Video Encoder Closed Caption Encoding

The encoder can be programmed to encode closed-caption (CC) data and extended data in the selected line. The data stream consists of 7-bit US-ASCII code and 1 odd-parity bit (see Video Encoder Closed-Caption Data Format).

Video Encoder Closed-Caption Data Format

MSB							LSB
Bit 6	Bit 5	Bit 4	Bit 3	Bit 2	Bit 1	Bit 0	Odd parity

The standard service encodes closed caption in both fields; the extended service encodes closed caption in even fields. When set to 1, the VENC_L21_WC_CTL[0] L21EN bit enables closed-caption encoding in odd fields. When set to 1, the VENC_L21_WC_CTL[1] L21EN bit enables closed-caption encoding in even fields.

To select the scan line where the CC data are encoded, program the VENC_LN_SEL[4:0] SLINE bit field.

CAUTION

The setting of the VENC_LN_SEL[4:0] SLINE bit field value depends on the video standard:

- PAL mode: Because there is a one-line offset, program the desired line number – 1. To activate the closed caption on line 21 (0x15), program the value 0x15 – 1 = 0x14. The default value is 0x15 + 1 = 0x16 (line 22).
- NTSC mode: Because there is a four-line offset, program the desired line number – 4. To activate the closed caption on line 21 (0x15), program the value 0x15 – 4 = 0x11. The default value is 0x15 + 4 = 0x19 (line 25).

The VENC_LN_SEL[25:16] LN21_RUNIN bit field should be kept at reset value (0x10B). Four closed-caption data registers contain the data to be encoded. The VENC_LINE21[15:8] L21O and VENC_LINE21[7:0] L21O bit fields contain the first and the second bytes, respectively, of closed-caption data to be encoded in the odd field. The VENC_LINE21[31:24] L21E and VENC_LINE21[23:16] L21E bit fields contain the first and the second bytes, respectively, of data to be encoded in the even field.

Immediately after the closed-caption data is written to the registers, in either the odd field or even field, the corresponding closed-caption status bit (VENC_STATUS[4] CCE or VENC_STATUS[3] CCO) is reset to 0 to indicate that the closed-caption data is available in the closed-caption data registers and yet to be encoded.

Immediately after the closed-caption data is encoded, the VENC_STATUS[4] CCE bit or the VENC_STATUS[3] CCO bit is set to 1 to indicate that the closed-caption data has been encoded and is ready to accept new data. As seen in Figure 8-75, a null character is automatically inserted if the closed-caption data is not written to the closed-caption data registers in time for encoding.

The running clock frequency is controlled by the VENC_CC_CARR_WSS_CARR[15:0] FCC bit field which should be kept at reset value (0x2631) to get 5034960.5Hz (32xfline) for NTSC-601. The closed-caption running clock common frequencies are detailed in Table 8-317.

Table 8-317. Video Encoder Closed-Caption RunClock Frequency Settings

	NTSC-601	PAL-601	NTSC Square Pixel	PAL Square Pixel
VENC_CC_CARR_WSS_CARR[15:0] FCC bit field value	0x2631	0x25ED	0x2A03	0x22B6

The closed-caption data is encoded in nonreturn-to-zero (NRZ) format. Additionally, the data translates to the IRE scale as follows:

- 0 = 0 IRE
- Sync pulse = -40 IRE
- 1 = 50 IRE

Figure 8-75 shows the parameters of closed-caption line data implemented in different standards.

Figure 8-75. Video Encoder Closed Captioning Timing

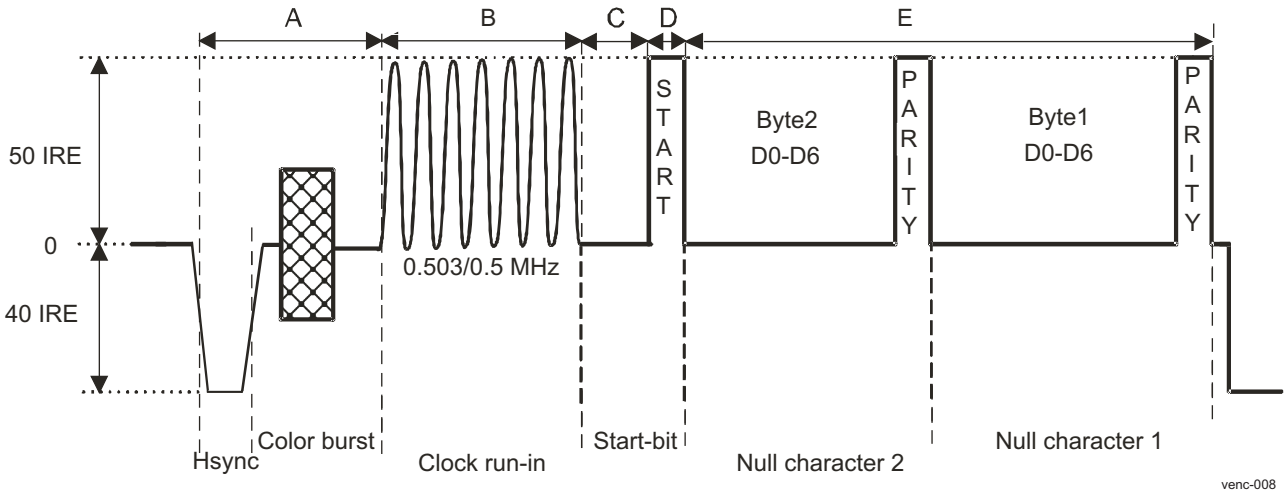


Table 8-318. Video Encoder Closed-Caption Standard Timing Values

Intervals	Description	Timing Values for Encoding			Timing Values for Decoding		
		Minimal	Nominal	Maximal	Lower Bound	Nominal	Upper Bound
A	HSYNC to clock running	10.250 µs	10.500 µs	10.750 µs	10.000 µs	10.500 µs	11.000 µs
B	Clock running		12.910 µs			12.910 µs	
C	Clock running to third start bit		3.972 µs			3.972 µs	
D	Start bit		1.986 µs			1.986 µs	
E	Data characters		31.778 µs			31.778 µs	

NOTE:

- The interval A is controlled by the VENC_LN_SEL[25:16] LN21_RUNIN bit field.
- The interval B is controlled by VENC_CC_CARR_WSS_CARR[15:0] FCC bit field.

All timing values listed in Table 8-318 are measured from the mid-point (half amplitude) on all edges.

For a complete description of closed-caption standard including CGMS-A copy protection, see the CEA-608-x standard.

8.3.4.5.2 Video Encoder Wide-Screen Signaling (WSS) Encoding

The encoder can embed data, encoded in accordance with the IEC61880 and ITU-R 1119 data insertion standard, within the vertical blanking interval.

The encoder supports WSS data insertion on line 20 of every frame in the NTSC format. WSS data insertion is enabled by activating the VENC_L21_WC_CTL[14:13] EVEN_ODD_EN bit and by programming the VENC_BSTAMP_WSS_DATA[27:8] WSS_DATA bit field.

The running clock frequency is controlled by the VENC_CC_CARR_WSS_CARR[31:16] FWSS bit field. The wide-screen signaling running clock common frequencies are detailed in Table 8-319

Table 8-319. Video Encoder Wide-Screen Signaling RunClock Frequency Settings

	NTSC-601	PAL-601	NTSC Square Pixel	PAL Square Pixel
VENC_CC_CARR_WSS_CARR[31:16] FWSS bit field value	0x043F	0x2F72	0x04AC	0x2B6D

To select the line where the WSS data are encoded, program the VENC_L21_WC_CTL[12:8] LINE bit field.

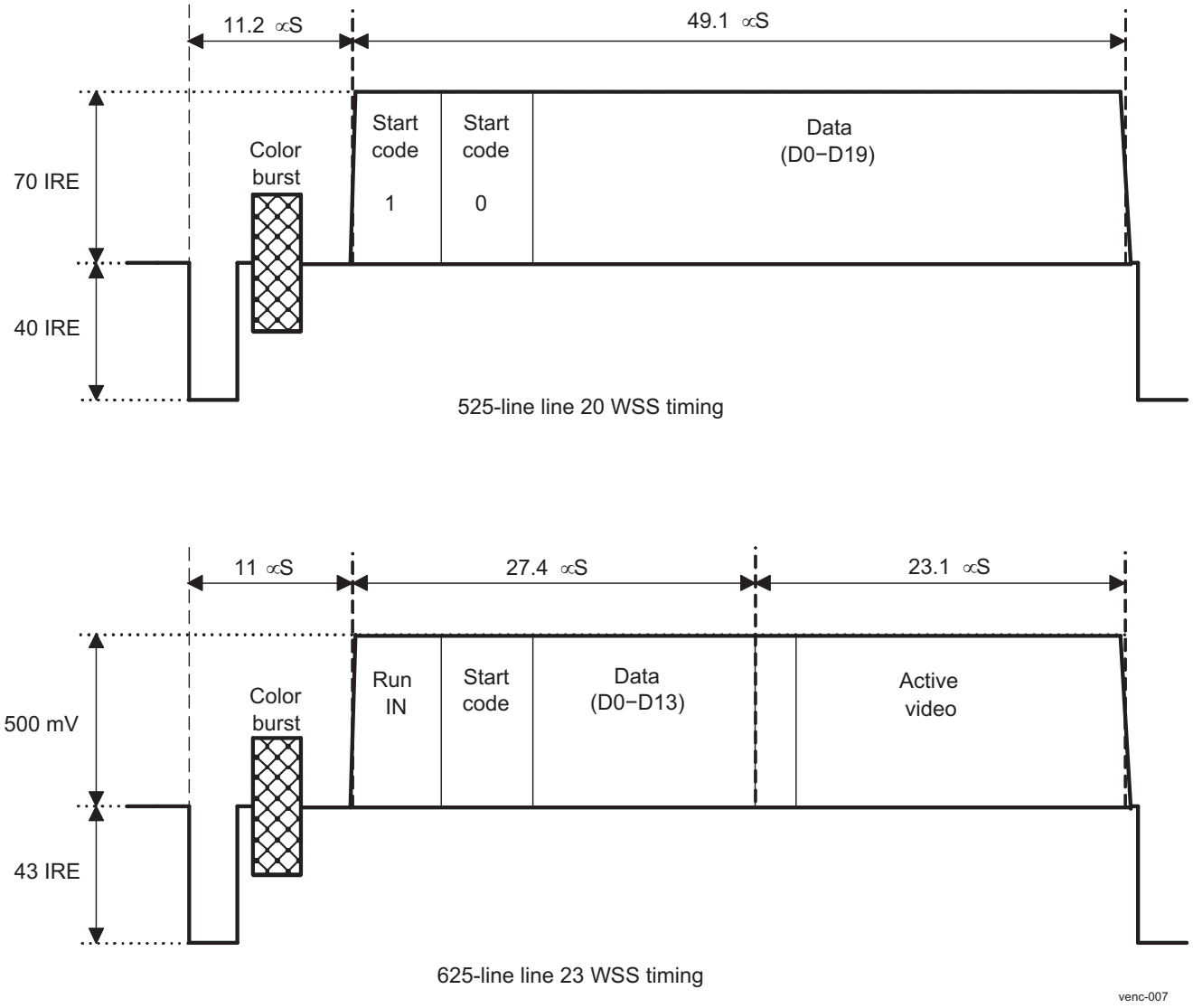
CAUTION

The setting of the VENC_L21_WC_CTL[12:8] LINE bit field value depends on the video standard:

- PAL mode: Because there is a one-line offset, program the desired line number – 1. The recommended value on line 23 is 0x16 + 1 = 0x17. The default value is 0x14 + 1 = 0x15 (line 21).
- NTSC mode: Because there is a four-line offset, program the desired line number – 4. The recommended value on line 20 is 0x10 + 4 = 0x14. The default value is 0x14 + 4 = 0x18 (line 24).

The WSS encoding block assumes that a full 10-bit video range is used to determine the 70 percent of peak-white amplitude of a logic-1 bit. The encoder also supports WSS data insertion on line 23 in the PAL format. Both waveforms are shown in Figure 8-76.

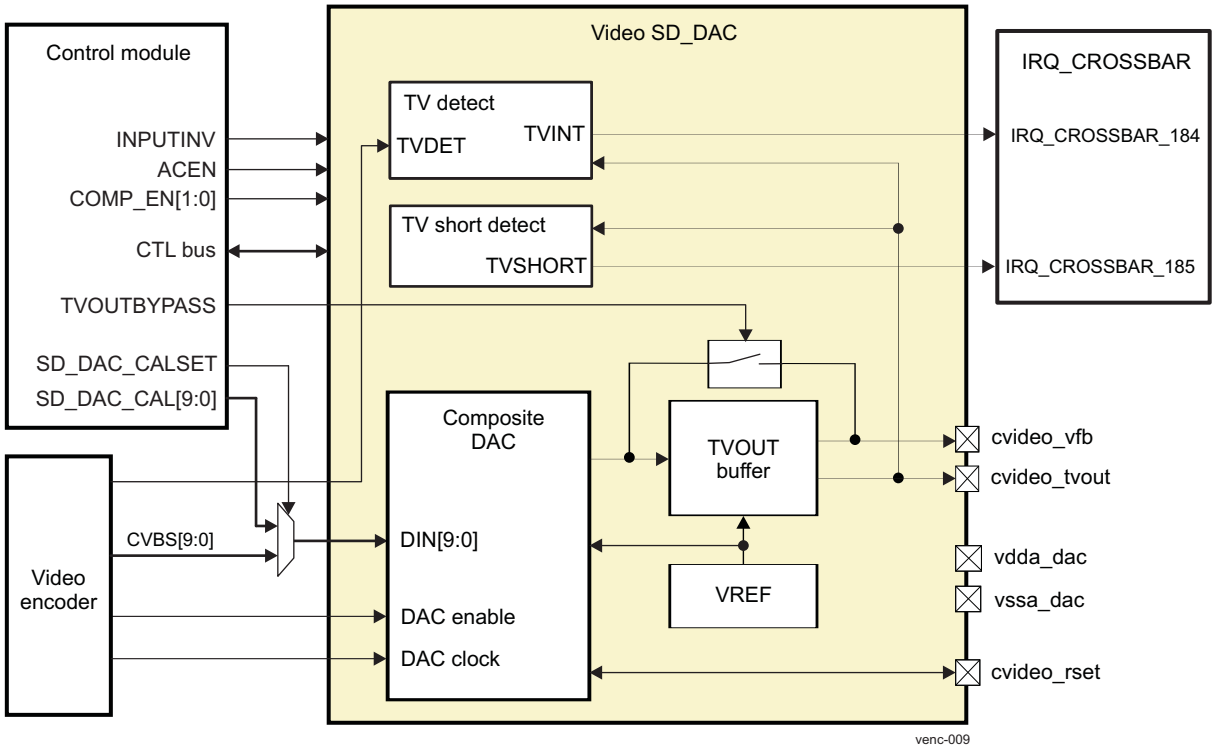
Figure 8-76. Video Encoder WSS Timing



8.3.4.6 Video Encoder SD_DAC

Figure 8-77 shows the architecture of the 10-bit video SD_DAC.

Figure 8-77. Video SD_DAC Architecture



The display subsystem provides the necessary control signals to interface the memory frame buffer directly to external displays. A 10-bit video SD_DAC is used to generate the video analog signal. The video SD_DAC also includes a TV detection/disconnection and power-down mode features.

8.3.4.6.1 Video SD_DAC DC/AC Coupled TV Load

The 10-bit video DAC supports both DC-coupled and AC-coupled TV loads. The CONTROL.CTRL_CORE_SD_DAC_CONTROL[2] ACEN bit is used to define which output coupling is used. This bit is the first one to be programmed according to the TV load on the PCB board.

NOTE: When DC coupling is used, there is a DC offset at the TVOUT output.

8.3.4.6.2 Video SD_DAC TV Detection/Disconnection Pulse Generation and Use

8.3.4.6.2.1 TV Detection/Disconnection Pulse Generation

The TV detect block is an integral part of the video SD_DAC module.

NOTE:

- The TV disconnection feature is recommended for power saving purpose. The TV detection/disconnection is only operational when video out is active. Therefore to detect cable connection automatically, it is necessary to periodically activate the video out to test for cable presence.

This block compares the output (*cvideo_tvout*) to reference voltage, to sense the condition of the load. To operate, the TV detect requires two digital signals, TVACEN and TVDET. The TVACEN signal indicates to both TVOUT buffer and the TV detect circuit, if the load is AC or DC coupling, to adjust accordingly. TVDET is a digital pulse at the frequency of the TV sync pulses. The operation of the circuit is based on the difference in voltage levels in the output of the buffer depending of the load status. The TV detect block compares the output against a couple of references and the result is latched at the start of every sync pulse. The status is read later in with the TVDET pulse rising edge.

The following registers are used to set the TV detection/disconnection pulse:

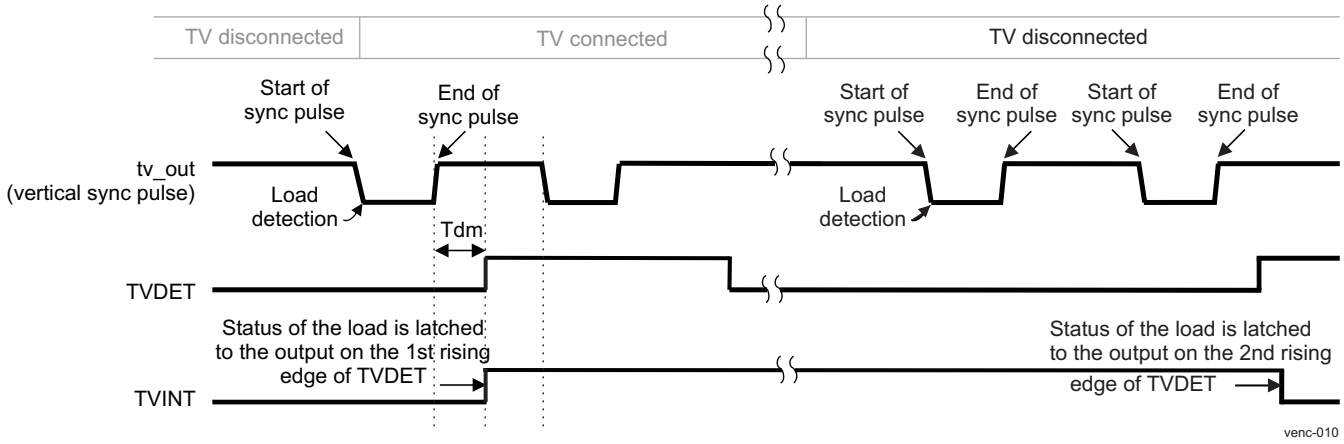
- The VENC_TVDETGP_INT_START_STOP_X register defines which pixels are used to start and stop the pulse inside their respective line.
- The VENC_TVDETGP_INT_START_STOP_Y register defines which lines are used to start and stop the pulse.
- The VENC_GEN_CTRL[0] EN bit enables or disables the TVDET pulse (0: Disable; 1: Enable).
- The VENC_GEN_CTRL[16] TVDP bit sets the TVDET pulse polarity (0: Active low; 1: Active high).

8.3.4.6.2.2 Recommended TV Detection/Disconnection Pulse Waveform

To enable the detection/disconnection of the load, the circuit requires that the TVDET pulse resembles the following waveform. By using the video encoder registers, the TVDET pulse polarity, start and stop is programmable. The only critical parameter is Tdm, which should be longer than the delay through the DAC and TVOUT buffer, which is at least 750 ns.

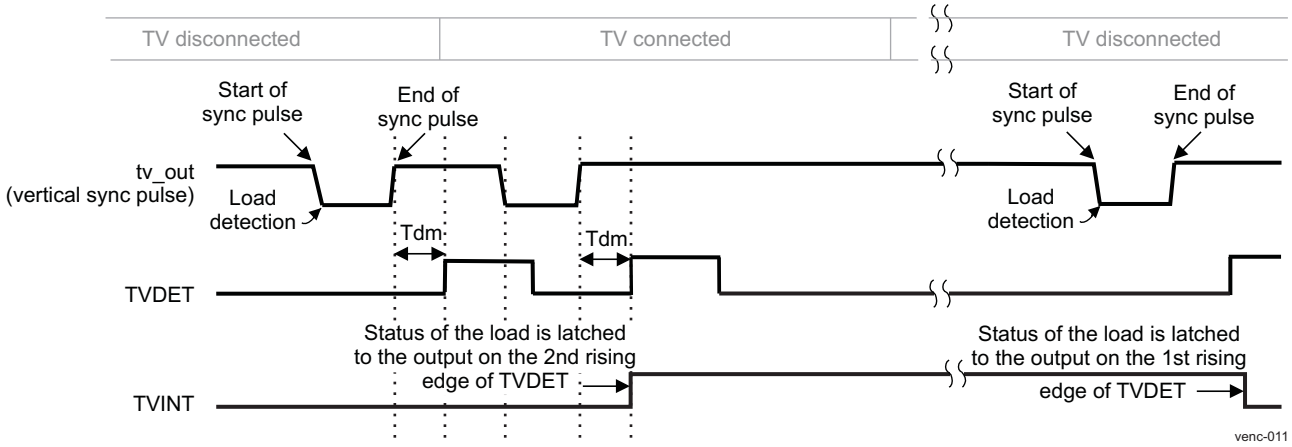
If DC-coupling is selected, the TVINT output signal for TV detection is latched at the rising edge of the first TVDET signal but the TVINT output signal for TV disconnection is latched after the next rising edge of the TVDET signal that happens during the next VSYNC timing. Figure 8-78 shows the waveforms for the DC-coupling TV detect pulse (TVDET) when load is connected and disconnected.

Figure 8-78. Video SD_DAC DC-Coupling TV Detect Waveforms for TV Connected and Disconnected



If AC-coupling is selected, the TVINT output signal for TV detection is latched after the next rising edge of the TVDET signal that happens during the next VSYNC timing but the TVINT output signal for TV disconnection is latched at the rising edge of the first TVDET signal. Figure 8-79 shows the waveforms for the ac-coupling TV detect pulse (TVDET) when load is connected and disconnected.

Figure 8-79. Video SD_DAC AC-Coupling TV Detect Waveforms for TV Connected and Disconnected



NOTE:

- When setting the VENC_TVDETGP_INT_START_STOP_X register, software users must ensure that the TVDET signal is in the active area. To avoid any problem, the TVDET signal must not be longer than one line.
- The activation of the TVDET signal will not have a visual impact on the *cvideo_tvout* output signal.

8.3.4.6.2.3 TV Detection/Disconnection Use

The TV-detection/TV-disconnection is based on the difference in voltage levels in the output of the TV buffer depending on the load status. The operation is slightly different for AC and for DC operation.

- For DC operation, the *cvideo_tvout* voltage is compared against a voltage reference that makes the comparator trigger in each sync pulse while the load is connected. For DC-coupling mode, see Figure 8-80.
- For AC operation, the *cvideo_tvout* voltage is compared against a voltage reference that makes the comparator trigger in each sync pulse while the load is disconnected. For AC-coupling mode, see Figure 8-81.

In both cases, the TVINT output signal produces a logic 1 when a load is connected, and a logic 0 when a load is disconnected.

NOTE: Because the video SD_DAC and the video encoder must be awake for connection detection, consider that the video SD_DAC can take up to 10 µs to wake up.

Figure 8-80. Video SD_DAC Signal Waveform Proposal for TV Detection/Disconnection in DC-Coupling Mode

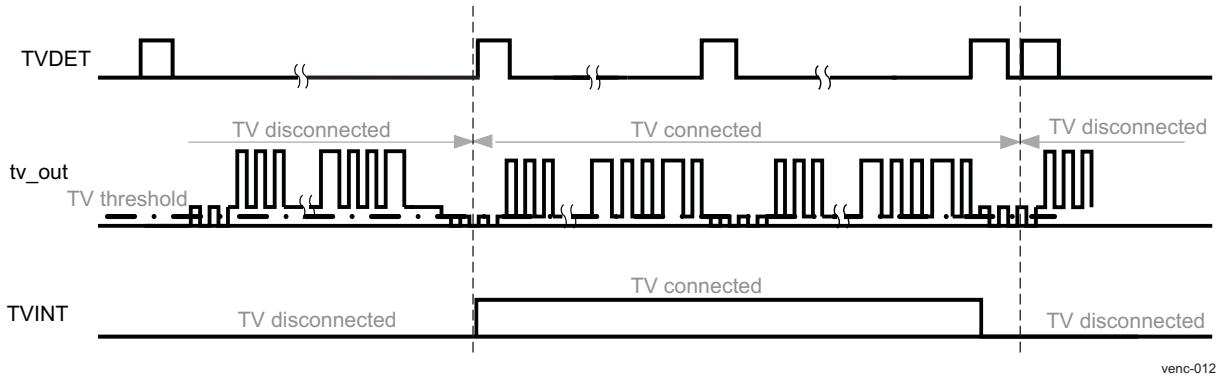
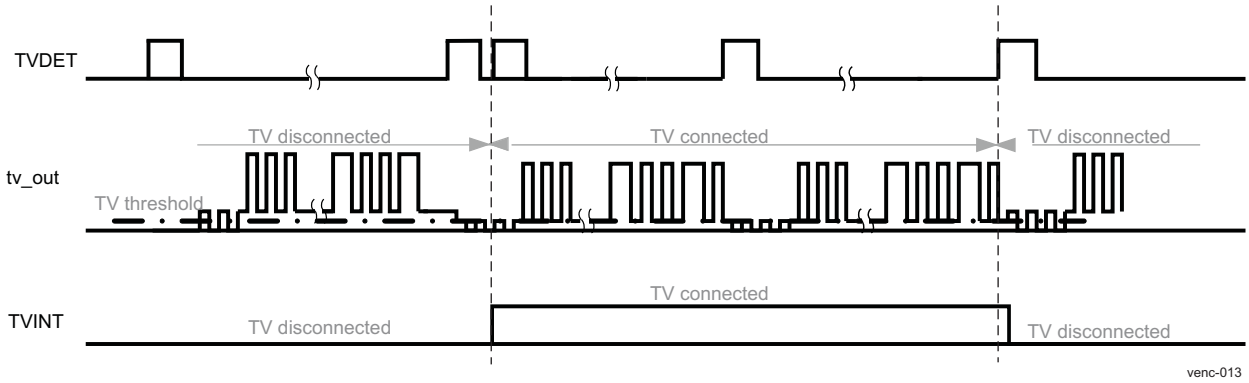


Figure 8-81. Video SD_DAC Signal Waveform Proposal for TV Detection/Disconnection in AC-Coupling Mode



8.3.4.6.3 Video SD_DAC TV Short Detection

A TV short-detection circuit, which can detect a short circuit condition at *cvideo_tvout* output during the vertical sync pulse, is integrated in the video SD_DAC module (see Figure 8-77).

When the output is shorted to ground, the TVSHORT signal turns to logic high. The TVSHORT signal is internally mapped in the device to allow interrupt generation (see Section 8.3.3, *Video Encoder Integration*). TV short detection is valid after two video frames pass with the short/no-short condition.

TV short detection has priority over TV load detection (see Section 8.3.4.6.2.2, *TV Detection/Disconnection Pulse Generation and Use*). Whenever a short is detected (with load or no-load), TVINT output is masked as long as the short remains. When released, the TV load detection resumes.

CAUTION

If TV short is detected, the video SD_DAC must be powered down by setting the DSS.DSS_VENC_CTRL[2] DAC_POWERDN_BGZ bit to 0.

8.3.4.6.4 Video SD_DAC Normal Mode

The normal mode of operation (DAC + TVOUT buffer) is the default mode of the video SD_DAC, with the CONTROL.CTRL_CORE_SD_DAC_CONTROL[3] TVOUTBYPASS bit set to 0 at device reset. The output of the DAC is internally connected to the TVOUT buffer, which can drive the composite video signal to 75 Ω load directly. For information about external connectivity, see Section 8.3.2, *Video Encoder Environment*.

In normal mode, higher input values of the DAC code result in lower output voltage levels by default, because of the inverting configuration of the TVOUT buffer.

8.3.4.6.5 Video SD_DAC Bypass Mode

The 10-bit video SD_DAC has a TVOUT buffer bypass mode that turns off the TVOUT buffer and redirects directly the DAC output to the *cvideo_vfb* pin.

This bypass mode is activated by setting the CONTROL.CTRL_CORE_SD_DAC_CONTROL[3] TVOUTBYPASS bit to 1. The reset value of the TVOUTBYPASS bit is 0 (that is, the TVOUT buffer is not bypassed).

NOTE: In bypass mode:

- *cvideo_rset* pin requires a Rset resistor connected to the ground. The typical value of the Rset resistor is 10000 Ω.
- *cvideo_vfb* pin requires a Rload resistor connected to the ground. The typical value of the Rload resistor is 1500 Ω.

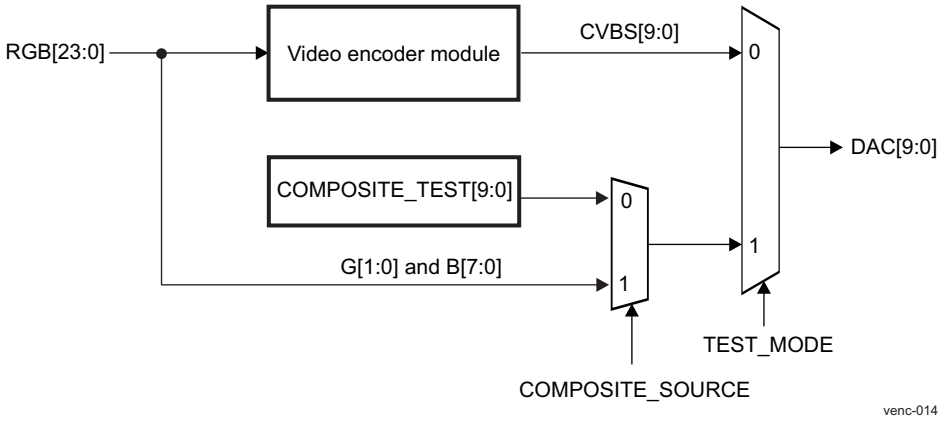
CAUTION

- The TV detect feature is not available in bypass mode.
- In bypass mode, an external video amplifier is needed on the *cvideo_vfb* pin.

8.3.4.6.6 Video SD_DAC Test Mode

10-bit external or internal datas values can be used to stimulate the DAC in debug mode, as shown in Figure 8-82.

Figure 8-82. Video SD_DAC Test Mode in Composite Video Mode



- Setting to 0x1 the VENC_OUTPUT_CONTROL[4] TEST_MODE bit will place the DAC under test.
- By setting the value of the VENC_OUTPUT_CONTROL[6] COMPOSITE_SOURCE bit to:
  - 0x0: the DAC input is coming from VENC_OUTPUT_TEST[9:0] COMPOSITE_TEST bit field
  - 0x1: the DAC input is coming from the display controller G[1:0], B[7:0]. To perform this configuration, the video encoder shall be configured to generate correct timing signals with the display controller, even if the video encoder is bypassed from data path perspective.

8.3.4.6.7 Video SD_DAC Power Management

After device reset, the DSS.DSS_VENC_CTRL[2] DAC_POWERDN_BGZ bit is set to 0, and the video SD_DAC is in power-down state.

Table 8-320 lists the possible SD_DAC power-management configurations and the corresponding register settings.

Table 8-320. Video SD_DAC Power Management Configurations

Power Management Controls				
DSS_VENC_CTRL [2] DAC_POWERDN_BGZ bit	**VENC_OUTPUT_CONTROL [1] COMPOSITE_ENABLE bit**	**CTRL_CORE_SD_DAC_CONTROL [3] TVOUTBYPASS bit**	**CTRL_CORE_SD_DAC_CONTROL [2] ACEN bit**	**Power State**
0	x	x	x	Total power down. Bandgap powered down.
1	0	x	x	Standby (analog in power down except bandgap/LDO)
1	1	0	0	Full power up in DC normal mode
1	1	0	1	Full power up in AC normal mode
1	1	1	x	Full power up in TVOUT bypass mode (DAC-only mode)

8.3.5 Video Encoder Programming Guide

8.3.5.1 Video Encoder Low-level Programming Models

8.3.5.1.1 Surrounding Modules Global Initialization

This section identifies the requirements of initializing the surrounding modules when the video emodule is to be used for the first time after a device reset. This initialization of the surrounding modules is based on the integration and environment of the video encoder (for more information, see Section 8.3.2 and Section 8.3.3).

CAUTION

DSS must be enabled first before VENC, in order the source clocks feeding VENC to be running before enabling the VENC.

VENC must be disabled before DSS, in order the source clocks feeding the VENC to be running before disabling the VENC. If DSS is disabled before VENC, the functional clocks will stop abruptly.

Global Initialization of Surrounding Modules

Surrounding Modules	Comments
PRCM	Module interface and functional clocks must be enabled. See Chapter 3, *Power, Reset, and Clock Management*
Control module	Module-specific pad muxing and configuration must be set in the control module. See Chapter 13, *Control Module*.
Display controller	See Section 8.2, *Display Controller,* for more information about the configuration.

8.3.5.1.2 Video Encoder Global Initialization

8.3.5.1.2.1 Main Sequence—Start Video Encoder

Start Video Encoder details the procedure to initialize the video encoder after a power-on or software reset.

Start Video Encoder

Step	Register/Bit Field/Programming Model	Value
Perform a video encoder reset.	See Section 8.3.5.1.2.3.	
Save the DSS VP1 IRQ enable register value (DSS VP interrupts context).	DISPC_VP1_IRQENABLE	
Reset all DSS VP1 IRQ.	DISPC_VP1_IRQENABLE	0x00
Configure the video encoder registers for PAL 601 or NTSC 601.	See Table 8-321.	0x--
Set the input video source and format.	VENC_F_CONTROL	See Table 8-321.
Set the synchronization modes.	VENC_SYNC_CTRL	See Table 8-321.
Configure the TV connection detection (optional).	See Section 8.3.5.1.2.4.	
Enable the composite video signal output.	See Section 8.3.5.1.2.6.	
Enable the display controller pipeline	See the display controller programming guide	
Wait until the first VSYNC pulse signal.		
Clear the SYNCLOST interrupt.	DISPC_VP1_IRQSTATUS[4] VPSYNCLOST_IRQ	0x1
Restore the DSS IRQ enable register.	DISPC_VP1_IRQENABLE	0x--

8.3.5.1.2.2 Main Sequence—Stop Video Encoder

Start Video Encoder details the procedure to stop the output of the composite signal.

Start Video Encoder

Step	Register/Bit Field/Programming Model	Value
Perform TV disconnection detection (optional).	See Section 8.3.5.1.2.5.	
Disable the composite video signal output.	See Section 8.3.5.1.2.7.	

8.3.5.1.2.3 Subsequence—Video Encoder Software Reset

Reset Video Encoder details the reset procedure.

Reset Video Encoder

Step	Register/Bit Field/Programming Model	Value
Activate reset.	VENC_F_CONTROL[8] RESET	0x1
Wait until reset is complete.	VENC_F_CONTROL[8] RESET	0x0

NOTE: Before changing the standard (NTSC or PAL) and related registers, a software reset is required to properly initialize the VENC module.

8.3.5.1.2.4 Subsequence—TV Connection Detection

TV Connection Detection Subsequence details the TV connection detection procedure.

TV Connection Detection Subsequence

Step	Register/Bit Field/Programming Model	Value
Set the pixel TV detection start and stop value.	VENC_TVDETGP_INT_START_STOP_X	0x--
Set the line TV detection start and stop value.	VENC_TVDETGP_INT_START_STOP_Y	0x--
Set the polarity of TV detection.	VENC_GEN_CTRL[16] TVDP	0x--
Enable TV detection.	VENC_GEN_CTRL[0] EN	0x1
Enable the composite video signal output.	See Section 8.3.5.1.2.6.	
Wait until TVINT = 1.		

CAUTION

- If ac coupling is selected, two TVDET pulses are required to set high the TVINT signal. Due to the internal logic of the video SD_DAC, the TVINT signal is generated after the next positive edge of the TVDET signal that happens during the next VSYNC timing.
- If dc coupling is selected, only one TVDET pulse is required to set high the TVINT signal.

8.3.5.1.2.5 Subsequence—TV Disconnection Detection

TV Disconnection Detection Subsequence details the TV disconnection detection procedure.

TV Disconnection Detection Subsequence

Step	Register/Bit Field/Programming Model	Value
Set the pixel TV detection start and stop value.	VENC_TVDETGP_INT_START_STOP_X	0x--
Set the line TV detection start and stop value.	VENC_TVDETGP_INT_START_STOP_Y	0x--
Set the polarity of TV detection.	VENC_GEN_CTRL[16] TVDP	0x--
Enable TV disconnection.	VENC_GEN_CTRL[0] EN	0x1
Wait until TVINT = 0.		

CAUTION

- If dc coupling is selected, two TVDET pulses are required to set low the TVINT signal. Because of the internal logic of the video SD_DAC, the TVINT signal is generated after the next positive edge of the TVDET signal that happens during the next VSYNC timing.
- If ac coupling is selected, only one TVDET pulse is required to set low the TVINT signal.

8.3.5.1.2.6 Subsequence—Enable Video Composite Signal Output

Enable Video Composite Signal Output details the procedure to enable the video composite signal output.

Enable Video Composite Signal Output

Step	Register/Bit Field/Programming Model	Value
Enable the composite video signal output.	VENC_OUTPUT_CONTROL	0x0000 000A
Power up the vdda_dac voltage for SD_DAC and the TVOUT buffer.		

8.3.5.1.2.7 Subsequence—Disable Video Composite Signal Output

Disable Video Composite Signal Output details the procedure to enable the video composite signal output.

Disable Video Composite Signal Output

Step	Register/Bit Field/Programming Model	Value
Disable the composite video signal output.	VENC_OUTPUT_CONTROL	0x00000000
Power down the vdda_dac voltage for SD_DAC and the TVOUT buffer.		

8.3.6 Video Encoder Use Case and Tips

8.3.6.1 Video Encoder Register Settings

Table 8-321 lists the register values to use in typical video encoder applications.

These values are validated programming values only for NTSC 601 and PAL 601 standards with a TV display resolution of 720×488 and 720×576, respectively.

Table 8-321. Video Encoder Register Programming Values

Register Name	NTSC 601	PAL 601
VENC_F_CONTROL	0x00000000	0x00000000
VENC_VIDOUT_CTRL	0x00000001	0x00000001

Table 8-321. Video Encoder Register Programming Values (continued)

Register Name	NTSC 601	PAL 601
VENC_SYNC_CTRL	0x00008040	0x00000040
VENC_LLEN	0x00000359	0x0000035F
VENC_FLENS	0x0000020C	0x00000270
VENC_HFLTR_CTRL	0x00000000	0x00000000
VENC_CC_CARR_WSS_CARR	0x043F2631	0x2F7225ED
VENC_C_PHASE	0x00000000	0x00000000
VENC_GAIN_U	0x00000102	0x00000111
VENC_GAIN_V	0x0000016C	0x00000181
VENC_GAIN_Y	0x0000012F	0x00000140
VENC_BLACK_LEVEL	0x00000043	0x0000003B
VENC_BLANK_LEVEL	0x00000038	0x0000003B
VENC_X_COLOR	0x00000007	0x00000007
VENC_M_CONTROL	0x00000001	0x00000002
VENC_BSTAMP_WSS_DATA	0x00000038	0x0000003F
VENC_S_CARR	0x21F07C1F	0x2A098ACB
VENC_LINE21	0x00000000	0x00000000
VENC_LN_SEL	0x01310011	0x01290015
VENC_L21_WC_CTL	0x0000F003	0x0000F603
VENC_HTRIGGER_VTRIGGER	0x00000000	0x00000000
VENC_SAVID_EAVID	0x069300F4	0x06A70108
VENC_FLEN_FAL	0x0016020C	0x00180270
VENC_LAL_PHASE_RESET	0x00060107	0x00040135
VENC_HS_INT_START_STOP_X	0x008E0350	0x00880358
VENC_HS_EXT_START_STOP_X	0x000F0359	0x000F035F
VENC_VS_INT_START_X	0x01A00000	0x01A70000
VENC_VS_INT_STOP_X_VS_INT_START_Y	0x020701A0	0x000001A7
VENC_VS_INT_STOP_Y_VS_EXT_START_X	0x01AC0024	0x01AF0000
VENC_VS_EXT_STOP_X_VS_EXT_START_Y	0x020D01AC	0x000101AF
VENC_VS_EXT_STOP_Y	0x00000006	0x00000025
VENC_AVID_START_STOP_X	0x03480078	0x03530083
VENC_AVID_START_STOP_Y	0x02060024	0x026C002E
VENC_FID_INT_START_X_FID_INT_START_Y	0x0001008A	0x0001008A
VENC_FID_INT_OFFSET_Y_FID_EXT_START_X	0x01AC0106	0x002E0138
VENC_FID_EXT_START_Y_FID_EXT_OFFSET_Y	0x01060006	0x01380001
VENC_TVDETGP_INT_START_STOP_X	0x00140001	0x00140001
VENC_TVDETGP_INT_START_STOP_Y	0x00010001	0x00010001
VENC_GEN_CTRL	0x00F90000	0x00FF0000
VENC_OUTPUT_CONTROL	0x0000000A	0x0000000A
VENC_OUTPUT_TEST	0x00000000	0x00000000

NOTE: The following display controller fields must be programmed to the NTSC 601 video standard:

- DISPC_VP1_SIZE_SCREEN[11:0] PPL = 720 - 1 = 719 = 0x2CF
- DISPC_VP1_SIZE_SCREEN[27:16] LPP = (488/2) - 1 = 243 = 0xF3

The following display controller fields must be programmed to the PAL 601 video standard:

- DISPC_VP1_SIZE_SCREEN[11:0] PPL = 720 - 1 = 719 = 0x2CF
- DISPC_VP1_SIZE_SCREEN[27:16] LPP = (576/2) - 1 = 287 = 0x11F

For both standards, the base addresses should be handle as follow:

- DISPC_GFX1_BA0 or DISPC_VID_BA0 = Base address of even bit field data
- DISPC_GFX1_BA1 or DISPC_VID_BA1 = Base address of odd bit field data

8.3.7 Video Encoder Register Manual

8.3.7.1 Video Encoder Instance Summary

Table 8-322. Video Encoder Instance Summary

Module Name	L3_MAIN Base Address	Size
VENC	0x5800 5000	4 KB

8.3.7.2 Video Encoder Registers

8.3.7.2.1 Video Encoder Register Summary

Table 8-323. Video Encoder Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	VENC L3_MAIN Physical Address
VENC_REV_ID	R	32	0x0000 0000	0x5800 5000
VENC_STATUS	R	32	0x0000 0004	0x5800 5004
VENC_F_CONTROL	RW	32	0x0000 0008	0x5800 5008
VENC_VIDOUT_CTRL	RW	32	0x0000 0010	0x5800 5010
VENC_SYNC_CTRL	RW	32	0x0000 0014	0x5800 5014
VENC_LLEN	RW	32	0x0000 001C	0x5800 501C
VENC_FLENS	RW	32	0x0000 0020	0x5800 5020
VENC_HFLTR_CTRL	RW	32	0x0000 0024	0x5800 5024
VENC_CC_CARR_WSS_CARR	RW	32	0x0000 0028	0x5800 5028
VENC_C_PHASE	RW	32	0x0000 002C	0x5800 502C
VENC_GAIN_U	RW	32	0x0000 0030	0x5800 5030
VENC_GAIN_V	RW	32	0x0000 0034	0x5800 5034
VENC_GAIN_Y	RW	32	0x0000 0038	0x5800 5038
VENC_BLACK_LEVEL	RW	32	0x0000 003C	0x5800 503C
VENC_BLANK_LEVEL	RW	32	0x0000 0040	0x5800 5040
VENC_X_COLOR	RW	32	0x0000 0044	0x5800 5044
VENC_M_CONTROL	RW	32	0x0000 0048	0x5800 5048
VENC_BSTAMP_WSS_DATA	RW	32	0x0000 004C	0x5800 504C
VENC_S_CARR	RW	32	0x0000 0050	0x5800 5050

Table 8-323. Video Encoder Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	VENC L3_MAIN Physical Address
VENC_LINE21	RW	32	0x0000 0054	0x5800 5054
VENC_LN_SEL	RW	32	0x0000 0058	0x5800 5058
VENC_L21_WC_CTL	RW	32	0x0000 005C	0x5800 505C
VENC_HTRIGGER_VTRIGGER	RW	32	0x0000 0060	0x5800 5060
VENC_SAVID_EAVID	RW	32	0x0000 0064	0x5800 5064
VENC_FLEN_FAL	RW	32	0x0000 0068	0x5800 5068
VENC_LAL_PHASE_RESET	RW	32	0x0000 006C	0x5800 506C
VENC_HS_INT_START_STOP_X	RW	32	0x0000 0070	0x5800 5070
VENC_HS_EXT_START_STOP_X	RW	32	0x0000 0074	0x5800 5074
VENC_VS_INT_START_X	RW	32	0x0000 0078	0x5800 5078
VENC_VS_INT_STOP_X_VS_INT_START_Y	RW	32	0x0000 007C	0x5800 507C
VENC_VS_INT_STOP_Y_VS_EXT_START_X	RW	32	0x0000 0080	0x5800 5080
VENC_VS_EXT_STOP_X_VS_EXT_START_Y	RW	32	0x0000 0084	0x5800 5084
VENC_VS_EXT_STOP_Y	RW	32	0x0000 0088	0x5800 5088
VENC_AVID_START_STOP_X	RW	32	0x0000 0090	0x5800 5090
VENC_AVID_START_STOP_Y	RW	32	0x0000 0094	0x5800 5094
VENC_FID_INT_START_X_FID_INT_START_Y	RW	32	0x0000 00A0	0x5800 50A0
VENC_FID_INT_OFFSET_Y_FID_EXT_START_X	RW	32	0x0000 00A4	0x5800 50A4
VENC_FID_EXT_START_Y_FID_EXT_OFFSET_Y	RW	32	0x0000 00A8	0x5800 50A8
VENC_TVDETGP_INT_START_STOP_X	RW	32	0x0000 00B0	0x5800 50B0
VENC_TVDETGP_INT_START_STOP_Y	RW	32	0x0000 00B4	0x5800 50B4
VENC_GEN_CTRL	RW	32	0x0000 00B8	0x5800 50B8
VENC_OUTPUT_CONTROL	RW	32	0x0000 00C4	0x5800 50C4
VENC_OUTPUT_TEST	RW	32	0x0000 00C8	0x5800 50C8

8.3.7.2.2 Video Encoder Register Description

Table 8-324. VENC_REV_ID

Address Offset	0x0000 0000		
Physical Address	0x5800 5000	**Instance**	VENC
Description	Revision ID for the encoder		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REVISION																															

Bits	Field Name	Description	Type	Reset
31:0	REVISION	IP Revision	R	See (1)

(1) TI internal data

Table 8-325. Register Call Summary for Register VENC_REV_ID

Video Encoder
- Video Encoder Register Summary: [0]

Table 8-326. VENC_STATUS

Address Offset	0x0000 0004		
Physical Address	0x5800 5004	**Instance**	VENC
Description	STATUS		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																											CCE	CCO	FSQ			

Bits	Field Name	Description	Type	Reset
31:5	RESERVED		R	0x0000000
4	CCE	Closed Caption Status for Even Field. This bit is set immediately after the data in registers LINE21_E0 and LINE21_E1 have been encoded to closed caption. This bit is reset when both of these registers are written.	R	0
3	CCO	Closed Caption Status for Odd Field. This bit is set immediately after the data in registers LINE21_O0 and LINE21_O1 have been encoded to closed caption. This bit is reset when both of these registers are written.	R	0
2:0	FSQ	Field Sequence ID. For PAL, all three FSQ[2:0] are used whereas for NTSC only FSQ[1:0] is meaningful. Furthermore, FSQ[0] represents ODD field when it is '0' and EVEN field when it is '1'. Read 0x0: ODD field Read 0x1: EVEN field	R	0x0

Table 8-327. Register Call Summary for Register VENC_STATUS

Video Encoder
- Video Encoder Closed Caption Encoding: [0][1][2][3]
- Video Encoder Register Summary: [4]

Table 8-328. VENC_F_CONTROL

Address Offset	0x0000 0008		
Physical Address	0x5800 5008	**Instance**	VENC
Description	This register specifies the input video source and format		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							RESET	SVDS		RGBF	BCOLOR			FMT	

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x000000
8	RESET	RESET the encoder 0x0: No effect 0x1: Reset the encoder, after reset, this bit is automatically set to zero.	RW	0
7:6	SVDS	Select Video Data Source. 0x0: Use external video source 0x1: Use internal Color BAR 0x2: Use background color 0x3: Reserved	RW	0x2
5	RGBF	RGB /YCrCb input coding range 0x0: The input RGB data are in binary format with coding range 0-255 The input YCrCb data are in binary format with coding range 0-255 0x1: The input RGB data are in binary format with coding range 16-235 The input YCrCb data are in binary format conforming to ITU-601 standard	RW	0
4:2	BCOLOR	Background color select 0x0: 0x1: 0x2: 0x3: 0x4: 0x5: 0x6: 0x7:	RW	0x1
1:0	FMT	These two bits specify the video input data stream format and timing 0x0: 24-bit 4:4:4 RGB 0x1: 24-bit 4:4:4 0x2: 16-bit 4:2:2 0x3: 8-bit ITU-R 656 4:2:2	RW	0x3

Table 8-329. Register Call Summary for Register VENC_F_CONTROL

Video Encoder

- Video Encoder Data Manager: [0]
- Video Encoder Test Pattern Generation: [1][2][3]
- Video Encoder Global Initialization: [4][5][6]
- Video Encoder Register Settings: [7]
- Video Encoder Register Summary: [8]

Table 8-330. VENC_VIDOUT_CTRL

Address Offset	0x0000 0010		
Physical Address	0x5800 5010	**Instance**	VENC
Description	Encoder output clock		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															MHZ_27_54

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	MHZ_27_54	Encoder output clock 0x0: 54 MHz, 4x oversampling 0x1: 27MHz, 2x oversampling, the last 2x oversampling filter bypassed	RW	0

Table 8-331. Register Call Summary for Register VENC_VIDOUT_CTRL

Video Encoder

- Video Encoder Register Settings: [0]
- Video Encoder Register Summary: [1]

Table 8-332. VENC_SYNC_CTRL

Address Offset	0x0000 0014		
Physical Address	0x5800 5014	**Instance**	VENC
Description	SYNC Control register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																FREE	ESAV	IGNP	NBLNKS	VBLKM		HBLKM		M_S	FID_POL	RESERVED		VS_POL	HS_POL	RESERVED	FHVMOD

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	reserved_5	R	0x0000
15	FREE	Free running 0x0: Free running disabled 0x1: Free running enabled. HSYNC and VSYNC are ignored	RW	1
14	ESAV	Enable to detect F and V bits only on EAV in ITU-R 656 input mode 0x0: Detection of F and V bits on both EAV and SAV 0x1: Detection of F and V bits only on EAV	RW	0
13	IGNP	Ignore protection bits in ITU-R 656 input mode 0x0: Protection bits are not ignored 0x1: Protection bits are ignored	RW	0

Bits	Field Name	Description	Type	Reset
12	NBLNKS	Blank shaping	RW	0
		0x0: Blank shaping enabled		
		0x1: Blank shaping disabled		
11:10	VBLKM	Vertical blanking mode	RW	0x0
		0x0: Internal default blanking		
		0x1: Internal default blanking AND internal programmable blanking defined by FAL and LAL		
		0x2: Reserved		
		0x3: Reserved		
9:8	HBLKM	Horizontal blanking mode	RW	0x0
		0x0: Internal default blanking		
		0x1: Internal programmable blanking defined by SAVID and EAVID		
		0x2: External blanking defined by AVID		
		0x3: Reserved		
7	M_S	Encoder is master or slave of external sync	RW	0
		0x0: Sync master. Encoder outputs Vsync, Hsync, FID and AVID.		
		0x1: Sync slave. Encoder is synchronized to external Vsync, Hsync, FID and AVID.		
6	FID_POL	FID output polarity	RW	0
		0x0: ODD field = '0' EVEN field = '1'		
		0x1: ODD field = '1' EVEN field = '0'		
5:4	RESERVED		R	0x0
3	VS_POL	VS input polarity	RW	0
		0x0: VS is active high		
		0x1: VS is active low		
2	HS_POL	HS input polarity	RW	0
		0x0: HS is active high		
		0x1: HS is active low		
1	RESERVED		R	0
0	FHVMOD	FID extracted from external FID or HSYNC and VSYNC	RW	0
		0x0: External FID		
		0x1: Extracted from HSYNC and VSYNC		

Table 8-333. Register Call Summary for Register VENC_SYNC_CTRL

Video Encoder

- Video Encoder Global Initialization: [0]
- Video Encoder Register Settings: [1]
- Video Encoder Register Summary: [2]

Table 8-334. VENC_LLEN

Address Offset	0x0000 001C		
Physical Address	0x5800 501C	**Instance**	VENC
Description	LLEN		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																LLEN_EN	RESERVED							LLEN							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15	LLEN_EN	LLEN_EN 0x0: disable 0x1: enable	RW	0
14:11	RESERVED		R	0x0
10:0	LLEN	LLEN[10:0] Line length or total number of pixels in a scan line including active video and blanking. Total number of pixels in a scan line = LLEN NOTE: A write to bit 11 of this bit field is illegal.	RW	0x359

Table 8-335. Register Call Summary for Register VENC_LLEN

Video Encoder
- Video Encoder Register Settings: [0]
- Video Encoder Register Summary: [1]

Table 8-336. VENC_FLENS

Address Offset	0x0000 0020		
Physical Address	0x5800 5020	**Instance**	VENC
Description	FLENS		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																					FLENS										

Bits	Field Name	Description	Type	Reset
31:11	RESERVED		R	0x000000
10:0	FLENS	The frame length or total number of lines in a frame including active video and blanking from the source image. Total number of lines in a frame from the source image = FLENS + 1	RW	0x20C

Table 8-337. Register Call Summary for Register VENC_FLENS

Video Encoder
- Video Encoder Register Settings: [0]
- Video Encoder Register Summary: [1]

Table 8-338. VENC_HFLTR_CTRL

Address Offset	0x0000 0024		
Physical Address	0x5800 5024	**Instance**	VENC
Description	HFLTR_CTRL		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																													CINTP		YINTP

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0000 0000
2:1	CINTP	Chrominance interpolation filter control 0x0: The chrominance interpolation filter is enabled 0x1: The first section of the chrominance interpolation filter is bypassed 0x2: The second section of the chrominance interpolation filter is bypassed 0x3: Both sections of the filter are bypassed	RW	0x0
0	YINTP	Luminance interpolation filter control 0x0: The luminance interpolation filter is enabled 0x1: The luminance interpolation filter is bypassed	RW	0

Table 8-339. Register Call Summary for Register VENC_HFLTR_CTRL

Video Encoder

- Video Encoder Register Settings: [0]
- Video Encoder Register Summary: [1]

Table 8-340. VENC_CC_CARR_WSS_CARR

Address Offset	0x0000 0028		
Physical Address	0x5800 5028	**Instance**	VENC
Description	Frequencie code control		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FWSS																FCC															

Bits	Field Name	Description	Type	Reset
31:16	FWSS	Wide screen signaling run-in code frequency control. For 50-Hz systems, FWSS = 2^{18} * 5 * 10^6/(LLEN * Fh), where LLEN = total number of pixels in a scan line Fh = line frequency.	RW	0x043F
15:0	FCC	Close caption run-in code frequency control. For 60-Hz system, FCC = 2^{18} * 0.5035 * 10^6/(LLEN * Fh) For 50-Hz systems, FCC = 2^{18} * 0.500 * 10^6/(LLEN * Fh), where LLEN = total number of pixels in a scan line Fh = line frequency.	RW	0x2631

Table 8-341. Register Call Summary for Register VENC_CC_CARR_WSS_CARR

Video Encoder

- Video Encoder Closed Caption Encoding: [0][1][2]
- Video Encoder Wide-Screen Signaling (WSS) Encoding: [3][4]
- Video Encoder Register Settings: [5]
- Video Encoder Register Summary: [6]

Table 8-342. VENC_C_PHASE

Address Offset	0x0000 002C		
Physical Address	0x5800 502C	**Instance**	VENC
Description	C_PHASE		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								CPHS							

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x000000
7:0	CPHS	Phase of the encoded video color subcarrier (including the color burst) relative to H-sync. The adjustable step is 360/256 degrees.	RW	0x00

Table 8-343. Register Call Summary for Register VENC_C_PHASE

Video Encoder

- Video Encoder Subcarrier and Burst Generation: [0][1][2]
- Video Encoder Register Settings: [3]
- Video Encoder Register Summary: [4]

Table 8-344. VENC_GAIN_U

Address Offset	0x0000 0030		
Physical Address	0x5800 5030	**Instance**	VENC
Description	Gain control for Cb signal		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								GU							

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x000000
8:0	GU	Gain control for Cb signal. Following are typical programming examples for NTSC and PAL standards. NTSC with 7.5 IRE pedestal: WHITE - BLACK = 92.5 IRE GU = 0x102 NTSC with no pedestal: WHITE - BLACK = 100 IRE GU = 0x117 PAL with no pedestal: WHITE - BLACK = 100 IRE GU = 0x111	RW	0x102

Table 8-345. Register Call Summary for Register VENC_GAIN_U

Video Encoder

- Video Encoder Chroma Stage: [0]
- Video Encoder Register Settings: [1]
- Video Encoder Register Summary: [2]

Table 8-346. VENC_GAIN_V

Address Offset	0x0000 0034		
Physical Address	0x5800 5034	**Instance**	VENC
Description	Gain control of Cr signal		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								GV							

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x000000
8:0	GV	Gain control of Cr signal. Following are typical programming examples for NTSC and PAL standards. NTSC with 7.5 IRE pedestal: WHITE - BLACK = 92.5 IRE GV = 0x16C NTSC with no pedestal: WHITE - BLACK = 100 IRE GV = 0x189 PAL with no pedestal: WHITE - BLACK = 100 IRE GV = 0x181	RW	0x16C

Table 8-347. Register Call Summary for Register VENC_GAIN_V

Video Encoder

- Video Encoder Chroma Stage: [0]
- Video Encoder Register Settings: [1]
- Video Encoder Register Summary: [2]

Table 8-348. VENC_GAIN_Y

Address Offset	0x0000 0038		
Physical Address	0x5800 5038	**Instance**	VENC
Description	Gain control of Y signal		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								GY							

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x000000
8:0	GY	Gain control of Y signal. Following are typical programming examples for NTSC/PAL standards. NTSC with 7.5 IRE pedestal: WHITE - BLACK = 92.5 IRE GY = 0x12F NTSC with no pedestal: WHITE - BLACK = 100 IRE GY = 0x147 PAL with no pedestal: WHITE - BLACK = 100 IRE GY = 0x140	RW	0x12F

Table 8-349. Register Call Summary for Register VENC_GAIN_Y

Video Encoder

- Video Encoder Luma Stage: [0]
- Video Encoder Register Settings: [1]
- Video Encoder Register Summary: [2]

Table 8-350. VENC_BLACK_LEVEL

Address Offset	0x0000 003C		
Physical Address	0x5800 503C	**Instance**	VENC
Description	BLACK LEVEL		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								BLACK							

Bits	Field Name	Description	Type	Reset
31:7	RESERVED		R	0x0000000
6:0	BLACK	Black level setting. Following are typical programming examples for NTSC/PAL standards. NTSC with 7.5 IRE pedestal: WHITE - BLACK = 92.5 IRE BLACK_LEVEL = 0x43 NTSC with no pedestal: WHITE - BLACK = 100 IRE BLACK_LEVEL = 0x38 PAL with no pedestal: WHITE - BLACK = 100 IRE BLACK_LEVEL = 0x3B	RW	0x43

Table 8-351. Register Call Summary for Register VENC_BLACK_LEVEL

Video Encoder

- Video Encoder Luma Stage: [0]
- Video Encoder Register Settings: [1]
- Video Encoder Register Summary: [2]

Table 8-352. VENC_BLANK_LEVEL

Address Offset	0x0000 0040		
Physical Address	0x5800 5040	**Instance**	VENC
Description	BLANK LEVEL		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								BLANK							

Bits	Field Name	Description	Type	Reset
31:7	RESERVED		R	0x0000000
6:0	BLANK	Blank level setting. Following are typical programming examples for NTSC/PAL standards. NTSC with 7.5 IRE pedestal: WHITE - BLACK = 92.5 IRE BLANK_LEVEL = 0x38 NTSC with no pedestal: WHITE - BLACK = 100 IRE BLANK_LEVEL = 0x38 PAL with no pedestal: WHITE - BLACK = 100 IRE BLANK_LEVEL = 0x3B	RW	0x38

Table 8-353. Register Call Summary for Register VENC_BLANK_LEVEL

Video Encoder

- Video Encoder Luma Stage: [0]
- Video Encoder Register Settings: [1]
- Video Encoder Register Summary: [2]

Table 8-354. VENC_X_COLOR

Address Offset	0x0000 0044		
Physical Address	0x5800 5044	**Instance**	VENC
Description	Cross-Colour Control register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								XCE	RESERVED		XCBW		LCD		

Bits	Field Name	Description	Type	Reset
31:7	RESERVED		R	0x0000000
6	XCE	Cross color reduction enable for composite video output. Cross color does not affect S-video output 0x0: Cross color reduction is disabled 0x1: Cross color is enabled	RW	0
5	RESERVED		R	0
4:3	XCBW	Cross color reduction filter selection 0x0: The notch is at 32.8 % of the frequency of the encoding pixel clock 0x1: The notch is at 26.5 % of the frequency of the encoding pixel clock 0x2: The notch is at 30.0 % of the frequency of the encoding pixel clock 0x3: The notch is at 29.2 % of the frequency of the encoding pixel clock	RW	0x0
2:0	LCD	These three bits can be used for chroma channel delay compensation. Delay on Luma channel. 0x0: 0 0x1: 0.5 pixel clock period 0x2: 1.0 pixel clock period 0x3: 1.5 pixel clock period 0x4: -2.0 pixel clock period 0x5: -1.5 pixel clock period 0x6: -1.0 pixel clock period 0x7: -0.5 pixel clock period	RW	0x0

Table 8-355. Register Call Summary for Register VENC_X_COLOR

Video Encoder

- Video Encoder Register Settings: [0]
- Video Encoder Register Summary: [1]

Table 8-356. VENC_M_CONTROL

Address Offset	0x0000 0048		
Physical Address	0x5800 5048	**Instance**	VENC
Description	M_CONTROL		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								PALI	PALN	PALPHS	CBW			PAL	FFRQ

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Reserved	R	0x000000
7	PALI	PAL I enable 0x0: Normal oeration 0x1: PAL I Enable	RW	0
6	PALN	PAL N Enable 0x0: Normal operation 0x1: PAL N enable	RW	0
5	PALPHS	PAL switch phase setting 0x0: PAL switch phase is nominal 0x1: PAL switch phase is inverted compared to nominal	RW	0
4:2	CBW	Chrominance lowpass filter bandwidth control 0x0: -6db at 21.8 % of encoding pixel clock frequency 0x1: -6db at 19.8 % of encoding pixel clock frequency 0x2: -6db at 18.0 % of encoding pixel clock frequency 0x3: Reserved 0x4: Reserved 0x5: -6db at 23.7 % of encoding pixel clock frequency 0x6: -6db at 26.8 % of encoding pixel clock frequency 0x7: Chrominance lowpass filter bypass	RW	0x0
1	PAL	Phase alternation line encoding selection 0x0: Phase alternation line encoding disabled 0x1: Phase alternation line encoding enabled	RW	0
0	FFRQ	The value of this field and the SQP bit in the BSTAMP_WSS_DATA register control the number of horizontal pixels displayed per scan line # OF MODE SQP FFRQ PIXEL PER LINE ITU-R 601 NTSC 0 1 858 Square pixel NTSC 1 1 780 ITU-R 601 PAL 0 0 864 Square pixel PAL 1 0 944	RW	1

Table 8-357. Register Call Summary for Register VENC_M_CONTROL

Video Encoder

- Video Encoder Subcarrier and Burst Generation: [0][1]
- Video Encoder Register Settings: [2]
- Video Encoder Register Summary: [3]

Table 8-358. VENC_BSTAMP_WSS_DATA

Address Offset	0x0000 004C		
Physical Address	0x5800 504C	**Instance**	VENC
Description	BSTAMP and WSS_DATA		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								WSS_D																SQP	BSTAP							

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:8	WSS_D	Wide Screen Signaling data NTSC: WORD 0 D0, D1 WORD 1 D2, D3, D4, D5 WORD 2 D6, D7, D8, D9, D10, D11, D12, D13 CRC D14, D15, D16, D17, D18, D19 PAL: GROUP A D0, D1, D2, D3 GROUP B D4, D5, D6, D7 GROUP C D8, D9 ,D10 GROUP D D11, D12, D13	RW	0x00000
7	SQP	Square-pixel sampling rate. See FFRQ in M_CONTROL register for programming information. 0x0: ITU-R 601 sampling rate 0x1: Square-pixel sampling rate	RW	0
6:0	BSTAP	Setting of amplitude of color burst.	RW	0x38

Table 8-359. Register Call Summary for Register VENC_BSTAMP_WSS_DATA

Video Encoder

- Video Encoder Subcarrier and Burst Generation: [0]
- Video Encoder Wide-Screen Signaling (WSS) Encoding: [1]
- Video Encoder Register Settings: [2]
- Video Encoder Register Summary: [3]

Table 8-360. VENC_S_CARR

Address Offset	0x0000 0050		
Physical Address	0x5800 5050	**Instance**	VENC
Description	Color Subcarrier Frequency Registers.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FSC																															

Bits	Field Name	Description	Type	Reset
31:0	FSC	These four bytes' data are used to program color subcarrier frequency. These four bytes are determined by the following formula. S_CARR = ROUND((Fsc/Fclkenc) * 232) Where Fsc = Frequency of the subcarrier Fclkenc = Frequency of the internal video encoding clock = 2*LLEN *Fh LLEN = Number of pixels in a scan line Fh = Line frequency See the description of LLEN registers (subaddresses 0x42 and 0x43).	RW	0x21F0 7C1F

Table 8-361. Register Call Summary for Register VENC_S_CARR

Video Encoder

- Video Encoder Subcarrier and Burst Generation: [0][1][2]
- Video Encoder Register Settings: [3]
- Video Encoder Register Summary: [4]

Table 8-362. VENC_LINE21

Address Offset	0x0000 0054		
Physical Address	0x5800 5054	**Instance**	VENC
Description	LINE 21		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
L21E																L21O															

Bits	Field Name	Description	Type	Reset
31:16	L21E	The two bytes of the closed caption data in the even field. For the data stream content, see Video Encoder Closed-Caption Data Format. [31:24] First byte of data [23:16] Second byte of data	RW	0x0000
15:0	L21O	The two bytes of the closed caption data in the odd field For the data stream content, see Video Encoder Closed-Caption Data Format. [15:8] First byte of data [7:0] Second byte of data	RW	0x0000

Table 8-363. Register Call Summary for Register VENC_LINE21

Video Encoder

- Video Encoder Closed Caption Encoding: [0][1][2][3]
- Video Encoder Register Settings: [4]
- Video Encoder Register Summary: [5]

Table 8-364. VENC_LN_SEL

Address Offset	0x0000 0058		
Physical Address	0x5800 5058	**Instance**	VENC
Description	LN_SEL		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								LN21_RUNIN								RESERVED														SLINE			

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x00
25:16	LN21_RUNIN	The two Bytes of the closed caption runin code position from the HSYNC	RW	0x10B
15:5	RESERVED		R	0x000
4:0	SLINE	Selects the line where closed caption or extended service data are encoded.	RW	0x15

Bits	Field Name	Description	Type	Reset
		PAL mode: Because there is a one-line offset, program the desired line number – 1. NTSC mode: Because there is a four-line offset, program the desired line number – 4.		

Table 8-365. Register Call Summary for Register VENC_LN_SEL

Video Encoder

- Video Encoder Closed Caption Encoding: [0][1][2][3]
- Video Encoder Register Settings: [4]
- Video Encoder Register Summary: [5]

Table 8-366. VENC_L21_WC_CTL

Address Offset	0x0000 005C		
Physical Address	0x5800 505C	**Instance**	VENC
Description	L21 and WC_CTL registers		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																INV	EVEN_ODD_EN		LINE					RESERVED						L21EN	

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15	INV	WSS inverter 0x0: no effect 0x1: invert WSS data	RW	0
14:13	EVEN_ODD_EN	This bit controls the WSS encoding. 0x0: WSS encoding OFF 0x1: Enables encoding in 2nd field (odd field) 0x2: Enables encoding in 1st field (even field) 0x3: Enables encoding in both fields	RW	0x0
12:8	LINE	Selects the line where WSS data are encoded. PAL mode: Because there is a one-line offset, program the desired line number – 1. NTSC mode: Because there is a four-line offset, program the desired line number – 4.	RW	0x14
7:2	RESERVED		R	0x00
1:0	L21EN	Those bits controls the Line21 closed caption encoding according to the mode. 0x0: Line21 encoding OFF 0x1: Enables encoding in 1st field (ODD field) 0x2: Enables encoding in 2d field (EVEN field) 0x3: Enables encoding in both fields	RW	0x0

Table 8-367. Register Call Summary for Register VENC_L21_WC_CTL

Video Encoder

- Video Encoder Closed Caption Encoding: [0][1]
- Video Encoder Wide-Screen Signaling (WSS) Encoding: [2][3][4]
- Video Encoder Register Settings: [5]
- Video Encoder Register Summary: [6]

Table 8-368. VENC_HTRIGGER_VTRIGGER

Address Offset	0x0000 0060		
Physical Address	0x5800 5060	**Instance**	VENC
Description	HTRIGGER and VTRIGGER		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								VTRIG								RESERVED								HTRIG							

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x00
25:16	VTRIG	Vertical trigger reference for VSYNC. These bits specify the phase between VSYNC input and the lines in a field. The VTRIG field is expressed in units of half-line.	RW	0x000
15:11	RESERVED		R	0x00
10:0	HTRIG	Horizontal trigger phase, which sets HSYNC. HTRIG is expressed in half-pixels or clk2x (27 MHz) periods	RW	0x000

Table 8-369. Register Call Summary for Register VENC_HTRIGGER_VTRIGGER

Video Encoder

- Video Encoder Register Settings: [0]
- Video Encoder Register Summary: [1]

Table 8-370. VENC_SAVID_EAVID

Address Offset	0x0000 0064		
Physical Address	0x5800 5064	**Instance**	VENC
Description	SAVID and EAVID		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								EAVID								RESERVED								SAVID							

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x00
26:16	EAVID	End of active video. These bits define the ending pixel position on a horizontal display line where active video will be displayed.	RW	0x693
15:11	RESERVED		R	0x00
10:0	SAVID	Start of active video. These bits define the starting pixel position on a horizontal line where active video will be displayed.	RW	0x0F4

Table 8-371. Register Call Summary for Register VENC_SAVID_EAVID

Video Encoder

- Video Encoder Register Settings: [0]
- Video Encoder Register Summary: [1]

Table 8-372. VENC_FLEN_FAL

Address Offset	0x0000 0068		
Physical Address	0x5800 5068	**Instance**	VENC
Description	FLEN and FAL		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								FAL								RESERVED								FLEN							

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x00
24:16	FAL	First Active Line of Field. These bits define the first active line of a field	RW	0x016
15:10	RESERVED		R	0x00
9:0	FLEN	Field length. These bits define the number of half_lines in each field. Length of field = (FLEN + 1) half_lines	RW	0x20C

Table 8-373. Register Call Summary for Register VENC_FLEN_FAL

Video Encoder

- Video Encoder Register Settings: [0]
- Video Encoder Register Summary: [1]

Table 8-374. VENC_LAL_PHASE_RESET

Address Offset	0x0000 006C		
Physical Address	0x5800 506C	**Instance**	VENC
Description	LAL and PHASE_RESET		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED													PRES		SBLANK	RESERVED								LAL							

Bits	Field Name	Description	Type	Reset
31:19	RESERVED		R	0x0000
18:17	PRES	Phase reset mode. 0x0: No reset 0x1: Reset every two lines 0x2: Reset every eight fields. Color subcarrier phase is reset to C_Phase (subaddress 5A) upon reset 0x3: Reset every four fields. Color subcarrier phase is reset to C_Phase (subaddress 5A) upon reset	RW	0x3

Bits	Field Name	Description	Type	Reset
16	SBLANK	Vertical blanking setting 0x0: Vertical blanking is defined by the setting of FAL and LAL registers. 0x1: Vertical blanking is forced automatically during field synchronization and equalization.	RW	0
15:9	RESERVED		R	0x00
8:0	LAL	Last Active Line of Field. These bits define the last active line of a field	RW	0x107

Table 8-375. Register Call Summary for Register VENC_LAL_PHASE_RESET

Video Encoder
- Video Encoder Register Settings: [0]
- Video Encoder Register Summary: [1]

Table 8-376. VENC_HS_INT_START_STOP_X

Address Offset	0x0000 0070		
Physical Address	0x5800 5070	**Instance**	VENC
Description	HS_INT_START_STOP_X		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								HS_INT_STOP_X								RESERVED								HS_INT_START_X							

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x00
25:16	HS_INT_STOP_X	HSYNC internal stop. These bits define HSYNC internal stop pixel value	RW	0x07E
15:10	RESERVED		R	0x00
9:0	HS_INT_START_X	HSYNC internal start. These bits define HSYNC internal start pixel value	RW	0x34E

Table 8-377. Register Call Summary for Register VENC_HS_INT_START_STOP_X

Video Encoder
- Video Encoder Register Settings: [0]
- Video Encoder Register Summary: [1]

Table 8-378. VENC_HS_EXT_START_STOP_X

Address Offset	0x0000 0074		
Physical Address	0x5800 5074	**Instance**	VENC
Description	HS_EXT_START_STOP_X		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								HS_EXT_STOP_X								RESERVED								HS_EXT_START_X							

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x00
25:16	HS_EXT_STOP_X	HSYNC external stop. These bits define HSYNC external stop pixel value	RW	0x00F

Bits	Field Name	Description	Type	Reset
15:10	RESERVED		R	0x00
9:0	HS_EXT_START_X	HSYNC external start. These bits define HSYNC external start pixel value	RW	0x359

Table 8-379. Register Call Summary for Register VENC_HS_EXT_START_STOP_X

Video Encoder
- Video Encoder Register Settings: [0]
- Video Encoder Register Summary: [1]

Table 8-380. VENC_VS_INT_START_X

Address Offset	0x0000 0078		
Physical Address	0x5800 5078	**Instance**	VENC
Description	VS_INT_START_X		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								VS_INT_START_X								RESERVED															

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x00
25:16	VS_INT_START_X	VSYNC internal start. These bits define VSYNC internal start pixel value.	RW	0x1A0
15:0	RESERVED		R	0x0000

Table 8-381. Register Call Summary for Register VENC_VS_INT_START_X

Video Encoder
- Video Encoder Register Settings: [0]
- Video Encoder Register Summary: [1]

Table 8-382. VENC_VS_INT_STOP_X_VS_INT_START_Y

Address Offset	0x0000 007C		
Physical Address	0x5800 507C	**Instance**	VENC
Description	VS_INT_STOP_X and VS_INT_START_Y		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								VS_INT_START_Y								RESERVED								VS_INT_STOP_X							

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x00
25:16	VS_INT_START_Y	VSYNC internal start. These bits define VSYNC internal start line value	RW	0x209
15:10	RESERVED		R	0x00
9:0	VS_INT_STOP_X	VSYNC internal stop. These bits define VSYNC internal stop pixel value	RW	0x1A0

Table 8-383. Register Call Summary for Register VENC_VS_INT_STOP_X_VS_INT_START_Y

Video Encoder
- Video Encoder Register Settings: [0]
- Video Encoder Register Summary: [1]

Table 8-384. VENC_VS_INT_STOP_Y_VS_EXT_START_X

Address Offset	0x0000 0080		
Physical Address	0x5800 5080	**Instance**	VENC
Description	VS_INT_STOP_Y and VS_EXT_START_X		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								VS_EXT_START_X								RESERVED								VS_INT_STOP_Y							

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x00
25:16	VS_EXT_START_X	VSYNC external start. These bits define VSYNC external start pixel value.	RW	0x1AC
15:10	RESERVED		R	0x00
9:0	VS_INT_STOP_Y	VSYNC internal stop. These bits define VSYNC internal stop line value.	RW	0x022

Table 8-385. Register Call Summary for Register VENC_VS_INT_STOP_Y_VS_EXT_START_X

Video Encoder
- Video Encoder Register Settings: [0]
- Video Encoder Register Summary: [1]

Table 8-386. VENC_VS_EXT_STOP_X_VS_EXT_START_Y

Address Offset	0x0000 0084		
Physical Address	0x5800 5084	**Instance**	VENC
Description	VS_EXT_STOP_X and VS_EXT_START_Y		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								VS_EXT_START_Y								RESERVED								VS_EXT_STOP_X							

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x00
25:16	VS_EXT_START_Y	VSYNC external start. These bits define VSYNC external start line value.	RW	0x20D
15:10	RESERVED		R	0x00
9:0	VS_EXT_STOP_X	VSYNC external stop. These bits define VSYNC external stop pixel value.	RW	0x1AC

Table 8-387. Register Call Summary for Register VENC_VS_EXT_STOP_X_VS_EXT_START_Y

Video Encoder
- Video Encoder Register Settings: [0]
- Video Encoder Register Summary: [1]

Table 8-388. VENC_VS_EXT_STOP_Y

Address Offset	0x0000 0088		
Physical Address	0x5800 5088	**Instance**	VENC
Description	VS_EXT_STOP_Y		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								VS_EXT_STOP_Y							

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x000000
9:0	VS_EXT_STOP_Y	VSYNC external stop. These bits define VSYNC external stop line value.	RW	0x006

Table 8-389. Register Call Summary for Register VENC_VS_EXT_STOP_Y

Video Encoder

- Video Encoder Register Settings: [0]
- Video Encoder Register Summary: [1]

Table 8-390. VENC_AVID_START_STOP_X

Address Offset	0x0000 0090		
Physical Address	0x5800 5090	**Instance**	VENC
Description	AVID_START_STOP_X		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								AVID_STOP_X								RESERVED								AVID_START_X							

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x00
25:16	AVID_STOP_X	AVID stop. These bits define AVID stop pixel value	RW	0x348
15:10	RESERVED		R	0x00
9:0	AVID_START_X	AVID start. These bits define AVID start pixel value	RW	0x078

Table 8-391. Register Call Summary for Register VENC_AVID_START_STOP_X

Video Encoder

- Video Encoder Register Settings: [0]
- Video Encoder Register Summary: [1]

Table 8-392. VENC_AVID_START_STOP_Y

Address Offset	0x0000 0094		
Physical Address	0x5800 5094	**Instance**	VENC
Description	AVID_START_STOP_Y		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								AVID_STOP_Y								RESERVED								AVID_START_Y							

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x00
25:16	AVID_STOP_Y	AVID stop. These bits define AVID stop line value.	RW	0x206
15:10	RESERVED		R	0x00
9:0	AVID_START_Y	AVID start. These bits define AVID start line value	RW	0x026

Table 8-393. Register Call Summary for Register VENC_AVID_START_STOP_Y

Video Encoder
- Video Encoder Register Settings: [0]
- Video Encoder Register Summary: [1]

Table 8-394. VENC_FID_INT_START_X_FID_INT_START_Y

Address Offset	0x0000 00A0		
Physical Address	0x5800 50A0	**Instance**	VENC
Description	FID_INT_START_X and FID_INT_START_Y		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								FID_INT_START_Y								RESERVED								FID_INT_START_X							

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x00
25:16	FID_INT_START_Y	FID internal stop. These bits define FID internal start line value	RW	0x001
15:10	RESERVED		R	0x00
9:0	FID_INT_START_X	FID internal start. These bits define FID internal start pixel value	RW	0x08A

Table 8-395. Register Call Summary for Register VENC_FID_INT_START_X_FID_INT_START_Y

Video Encoder
- Video Encoder Register Settings: [0]
- Video Encoder Register Summary: [1]

Table 8-396. VENC_FID_INT_OFFSET_Y_FID_EXT_START_X

Address Offset	0x0000 00A4		
Physical Address	0x5800 50A4	**Instance**	VENC
Description	FID_INT_OFFSET_Y and FID_EXT_START_X		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								FID_EXT_START_X								RESERVED								FID_INT_OFFSET_Y							

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x00
25:16	FID_EXT_START_X	FID external start. These bits define FID external start pixel value	RW	0x1AC
15:10	RESERVED		R	0x00
9:0	FID_INT_OFFSET_Y	FID internal offset. These bits define FID internal offset linel value	RW	0x106

Table 8-397. Register Call Summary for Register VENC_FID_INT_OFFSET_Y_FID_EXT_START_X

Video Encoder

- Video Encoder Register Settings: [0]
- Video Encoder Register Summary: [1]

Table 8-398. VENC_FID_EXT_START_Y_FID_EXT_OFFSET_Y

Address Offset	0x0000 00A8		
Physical Address	0x5800 50A8	**Instance**	VENC
Description	FID_EXT_START_Y and FID_EXT_OFFSET_Y		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								FID_EXT_OFFSET_Y								RESERVED								FID_EXT_START_Y							

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x00
25:16	FID_EXT_OFFSET_Y	FID external offset. These bits define FID external offset line value	RW	0x106
15:10	RESERVED		R	0x00
9:0	FID_EXT_START_Y	FID external start. These bits define FID external start line value.	RW	0x006

Table 8-399. Register Call Summary for Register VENC_FID_EXT_START_Y_FID_EXT_OFFSET_Y

Video Encoder

- Video Encoder Register Settings: [0]
- Video Encoder Register Summary: [1]

Table 8-400. VENC_TVDETGP_INT_START_STOP_X

Address Offset	0x0000 00B0		
Physical Address	0x5800 50B0	**Instance**	VENC
Description	TVDETGP_INT_START_STOP_X		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								TVDETGP_INT_STOP_X								RESERVED								TVDETGP_INT_START_X							

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x00
25:16	TVDETGP_INT_STOP_X	TVDETGP internal stop. These bits define TVDETGP internal stop pixel value.	RW	0x014
15:10	RESERVED		R	0x00
9:0	TVDETGP_INT_START_X	TVDETGP internal start. These bits define TVDETGP internal start pixel value	RW	0x001

Table 8-401. Register Call Summary for Register VENC_TVDETGP_INT_START_STOP_X

Video Encoder

- Video SD_DAC TV Detection/Disconnection Pulse Generation and Use: [0][1]
- Video Encoder Global Initialization: [2][3]
- Video Encoder Register Settings: [4]
- Video Encoder Register Summary: [5]

Table 8-402. VENC_TVDETGP_INT_START_STOP_Y

Address Offset	0x0000 00B4		
Physical Address	0x5800 50B4	**Instance**	VENC
Description	TVDETGP_INT_START_STOP_Y		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								TVDETGP_INT_STOP_Y								RESERVED								TVDETGP_INT_START_Y							

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x00
25:16	TVDETGP_INT_STOP_Y	TVDETGP internal stop. These bits define TVDETGP internal stop line value.	RW	0x001
15:10	RESERVED		R	0x00
9:0	TVDETGP_INT_START_Y	TVDETGP internal start. These bits define TVDETGP internal start line value	RW	0x001

Table 8-403. Register Call Summary for Register VENC_TVDETGP_INT_START_STOP_Y

Video Encoder

- Video SD_DAC TV Detection/Disconnection Pulse Generation and Use: [0]
- Video Encoder Global Initialization: [1][2]
- Video Encoder Register Settings: [3]
- Video Encoder Register Summary: [4]

Table 8-404. VENC_GEN_CTRL

Address Offset	0x0000 00B8		
Physical Address	0x5800 50B8	**Instance**	VENC
Description	TVDETGP enable and SYNC_POLARITY and UVPHASE_POL		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED					MS	UVPHASE_POL_656	CBAR	HIP	VIP	HEP	VEP	AVIDP	FIP	FEP	TVDP	RESERVED															EN

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x00
26	MS	UVPHASE_POL MS mode UV phase 0x0: CbCr 0x1: CrCb	RW	0
25	UVPHASE_POL_656	UVPHASE_POL 656 input mode UV phase 0x0: CbCr 0x1: CrCb	RW	0
24	CBAR	UVPHASE_POL CBAR mode UV phase 0x0: CbCr 0x1: CrCb	RW	0
23	HIP	HSYNC internal polarity 0x0: Active Low 0x1: Active High	RW	1
22	VIP	VSYNC internal polarity 0x0: Active Low 0x1: Active High	RW	1
21	HEP	HSYNC external polarity 0x0: Active Low 0x1: Active High	RW	1
20	VEP	VSYNC externall polarity 0x0: Active Low 0x1: Active High	RW	1
19	AVIDP	AVID polarity 0x0: Active Low 0x1: Active High	RW	1
18	FIP	FID internal polarity 0x0: Active Low 0x1: Active High	RW	1
17	FEP	FID external polarity 0x0: Active Low 0x1: Active High	RW	1
16	TVDP	TVDETGP polarity 0x0: Active Low 0x1: Active High	RW	1

Bits	Field Name	Description	Type	Reset
15:1	RESERVED		R	0x0000
0	EN	TVDETGP generation enable 0x0: Disabled 0x1: Enabled	RW	0

Table 8-405. Register Call Summary for Register VENC_GEN_CTRL

Video Encoder

- Video SD_DAC TV Detection/Disconnection Pulse Generation and Use: [0][1]
- Video Encoder Global Initialization: [2][3][4][5]
- Video Encoder Register Settings: [6]
- Video Encoder Register Summary: [7]

Table 8-406. VENC_OUTPUT_CONTROL

Address Offset	0x0000 00C4		
Physical Address	0x5800 50C4	**Instance**	VENC
Description	Output channel control register Also contains some test control features		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								COMPOSITE_SOURCE	RESERVED	TEST_MODE	VIDEO_INVERT	RESERVED	COMPOSITE_ENABLE	RESERVED	

Bits	Field Name	Description	Type	Reset
31:7	RESERVED		R	0x00
6	COMPOSITE_SOURCE	Source of composite video data in test mode 0x0: Composite test data comes from internal register OUTPUT_TEST[9:0] 0x1: Composite test data comes from display controller video port G[1:0], B[7:0]	RW	0
5	RESERVED		RW	0x00
4	TEST_MODE	This enables the video DACs to be tested. The values sent to the DACs comes from a register for each output channel (Luma, Composite or Chroma) or from the display controller video port bits G[1:0], B[7:0], depending on the setting of the Source bits 0x0: Video outputs are in normal operation 0x1: Test mode. Video outputs are directly connected to either internal registers or the display controller video port.	RW	0
3	VIDEO_INVERT	Controls the video output polarity. This may be used to correct for inversion in an external video amplifier. 0x0: Video outputs are inverted 0x1: Video outputs are normal polarity	RW	1
2	RESERVED		RW	0x00

Bits	Field Name	Description	Type	Reset
1	COMPOSITE_ENABLE	Enable the Composite output channel 0x0: Composite output is disabled 0x1: Composite output is enabled	RW	0
0	RESERVED		RW	0x00

Table 8-407. Register Call Summary for Register VENC_OUTPUT_CONTROL

Video Encoder

- Video Encoder Environment: [0]
- Video SD_DAC Test Mode: [1][2]
- Video SD_DAC Power Management: [3]
- Video Encoder Global Initialization: [4][5]
- Video Encoder Register Settings: [6]
- Video Encoder Register Summary: [7]

Table 8-408. VENC_OUTPUT_TEST

Address Offset	0x0000 00C8		
Physical Address	0x5800 50C8	**Instance**	VENC
Description	Test values for the Luma/Composite Video DAC		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								COMPOSITE_TEST							

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x00
9:0	COMPOSITE_TEST	In test mode, DAC input value (if composite video is selected)	RW	0x000

Table 8-409. Register Call Summary for Register VENC_OUTPUT_TEST

Video Encoder

- Video SD_DAC Test Mode: [0]
- Video Encoder Register Settings: [1]
- Video Encoder Register Summary: [2]

Interconnect

This chapter describes the device interconnect.

NOTE: The level 3 (L3) interconnect is an instantiation of the NoC interconnect from Arteris, Inc. Arteris is a registered trademark of Arteris, Inc.

This document contains materials that are ©Arteris, Inc., and that constitute proprietary information of Arteris, Inc.

All materials and trademarks are used under license from Arteris, Inc. For additional information, see the Arteris Reference manuals, or contact Arteris, Inc.

NoC is an abbreviation for Network On Chip.

NOTE: The level 4 (L4) interconnects are instantiations of the Sonics3220™ interconnect from Sonics, Inc.

This document contains materials that are ©2003-2010 Sonics, Inc., and that constitute proprietary information of Sonics, Inc.

SonicsLX and Sonics3220, are trademarks or registered trademarks of Sonics, Inc. All such materials and trademarks are used under license from Sonics, Inc. For additional information, see the SonicsLX or Sonics3220 Reference manuals, or contact Sonics, Inc.

SLX is an abbreviation for SonicsLX®.

Topic **Page**

9.1 Interconnect Overview

9.1.1 Terminology

The following terminology is critical to understanding the interconnect:

- Initiator: Module able to initiate read and write requests to the chip interconnect (typically: processors, DMA, etc.).
- Target: Unlike an initiator, a target module cannot generate read/write requests to the chip interconnect, but it can respond to these requests. However, it may generate interrupts or a DMA request to the system (typically: peripherals, memory controllers).

NOTE: A module can have several separate ports; therefore, a module can be an initiator and a target.

- Agent: Each connection of one module to one interconnect is done using an agent, which is an adaptation (sometimes configurable) between the module and the interconnect. A target module is connected by a target agent (TA), and an initiator module is connected by an initiator agent (IA).
- Interconnect: The decoding, routing, and arbitration logic that enables the connection between multiple initiator modules and multiple target modules connected on it. Quality of service (QoS) is guaranteed.
- Register target (RT): Special TA used to access the interconnect internal configuration registers
- Data-flow signal: Any signal that is part of a clearly identified transfer or data flow (typically: command, address, byte enables, etc.). Signal behaviour is defined by the protocol semantics.
- Sideband signal: Any signal whose behaviour is not associated to a precise transaction or data flow.
- Out-of-band error: Any signal whose behaviour is associated to a device error-reporting scheme, as opposed to in-band errors.

NOTE: Interrupt requests and DMA requests are not routed by the interconnect in the device.

- Firewall: A programmable feature integrated in a target agent or L4 interconnect to prevent unauthorized access to or from a module. A firewall can be configured using three criteria:
  - Initiator requesting access
  - Address space access
  - Type of access
- ConnID: Any transaction in the system interconnect is tagged by an in-band qualifier ConnID, which uniquely identifies the initiator at a given interconnect point. A ConnID is transmitted in band with the request and is used for firewall and error-logging mechanism.
- Firewall comparison mechanism: A comparison made in the firewall between access in-band qualifiers and access permissions that are programmed in the firewall configuration registers. If the comparison is successful, access is allowed; otherwise, access is denied.
- MCmd qualifier: Command bus that indicates the type of transfer requested. Table 9-1 lists the commands encoded. For information specific to L3 Interconnect error logging, see *L3 Firewall Error-Logging Registers*

Table 9-1. MCmd Qualifier Description

MCmd[2:0]	Transaction Type
0 0 0	Idle
0 0 1	Write
0 1 0	Read
0 1 1	ReadEx
1 0 0	Read link
1 0 1	Write nonposted
1 1 0	Write conditional

Table 9-1. MCmd Qualifier Description (continued)

MCmd[2:0]	Transaction Type
1 1 1	Write broadcast

- MReqInfo qualifier: Three MReqInfo qualifiers describe the access during the use of the firewall comparison mechanism, as described in Table 9-2.

Table 9-2. MReqInfo Qualifier Description

Qualifiers	Description
MReqType	0: Data access
	1: Opcode fetch
MReqSupervisor	0: User mode
	1: Supervisor mode
MReqDebug	0: Functional access
	1: Debug access

- Register that configures the combination of the MReqInfo, allowing access permission to the target module (TM) based on the MReqInfo in-band qualifier values.
- SError: Target that indicates an error condition to the initiator.
- SResp qualifier: Response from the target to the initiator concerning the transaction, as described in Table 9-3.

Table 9-3. SResp Qualifier Description

SResp[1:0]	Description
0 0	No response
0 1	Data valid/accept
1 0	Not used
1 1	Error

- MTagID: Interconnect qualifier generated by the L3_MAIN masters which purpose is to identify whether reordering is allowed or not relative to other transactions. Strong ordering is ensured by using same MTagID values between transactions. Reordering is allowed by using different MTagID values between transactions.
  The MTagID values may or may not be changed by the interconnect, but the intended reordering restriction must match what came from the master. In other words, the interconnect allocates dynamically MTagID values in such a way that the intended reordering restrictions from each master are honored.

9.1.2 Architecture Overview

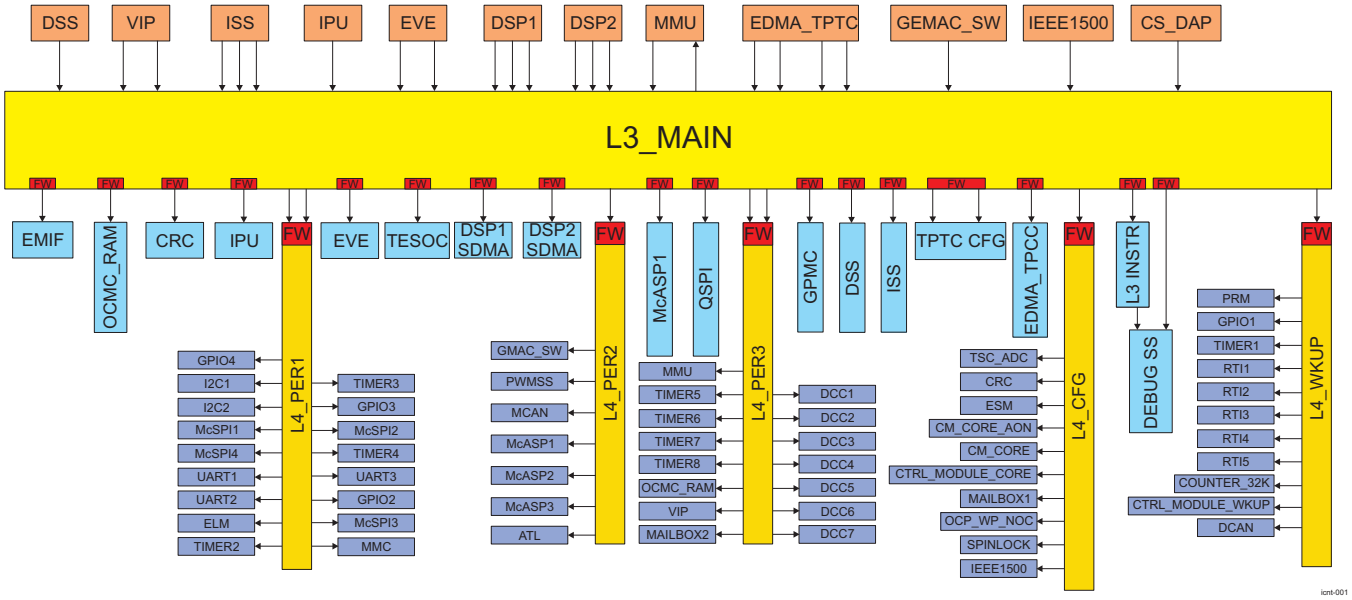
The device memory hierarchy includes four levels:

- L1 and L2 memory spaces are associated with the IPU and DSP subsystems.
- The chip-level interconnect consists of one L3 interconnect and five L4 interconnects. It enables communication among the modules and subsystems in the device.

Figure 9-2 shows an overview of the L3 and L4 interconnect architecture.

NOTE: ISS, CRC, TESOC, ESM and DCC1 to DCC7 are not supported on the DRA78x family of devices.

Figure 9-1. Interconnect Overview



- L3 handles many types of data transfers, especially exchanges with system-on-chip/external memories. L3 transfers data with a maximum width of 128 bits from the initiator to the target. The L3 interconnect is a little-endian platform
- The L4 is composed of the following:
  - L4_CFG: Includes the majority of the firewall configuration interface for level 3 (L3) system modules and peripheral interconnect;
  - L4_PER: Includes the main peripherals that require direct memory access access. L4_PER has three instances L4_PER1, L4_PER2 and L4_PER3. These three instances has the following ports connecting them to the L3_MAIN interconnect:
    - L4_PER1_P1, L4_PER1_P2
    - L4_PER2_P1
    - L4_PER3_P1, L4_PER3_P2

    Through the L3_MAIN interconnect, different initiators can access each of these ports. For information about which port can be accessed by each initiator, see Figure 9-3.
  - L4_WKUP: Includes peripherals attached to the WKUP power domain.

Modules are connected to the interconnect through an IA for the initiator module and a TA for target modules. Each module/subsystem connection is statically configured to tune the access, depending on the characteristics of the module.

To unauthorize a module or L4 interconnect access, some TAs include configurable firewalls (FWs). A firewall restricts or filters the accesses allowed to an initiator according to different access criteria. The firewalls can usually be configured by software.

The L3 and L4 interconnect default settings are fully functional; they enable all possible functional data paths and a minimal default protection setting.

9.2 L3_MAIN Interconnect

This section describes the L3_MAIN interconnect and its components. With the exception of register points, each component includes functions for the request and response networks.

9.2.1 L3_MAIN Interconnect Overview

The L3_MAIN interconnect links cores in a flexible topology that couples low power with high performance. Innovative physical structures and advanced protocols ensure bandwidth and latency to individual IP cores, providing dedicated connections between IP cores and logical connections over a shared interconnect.

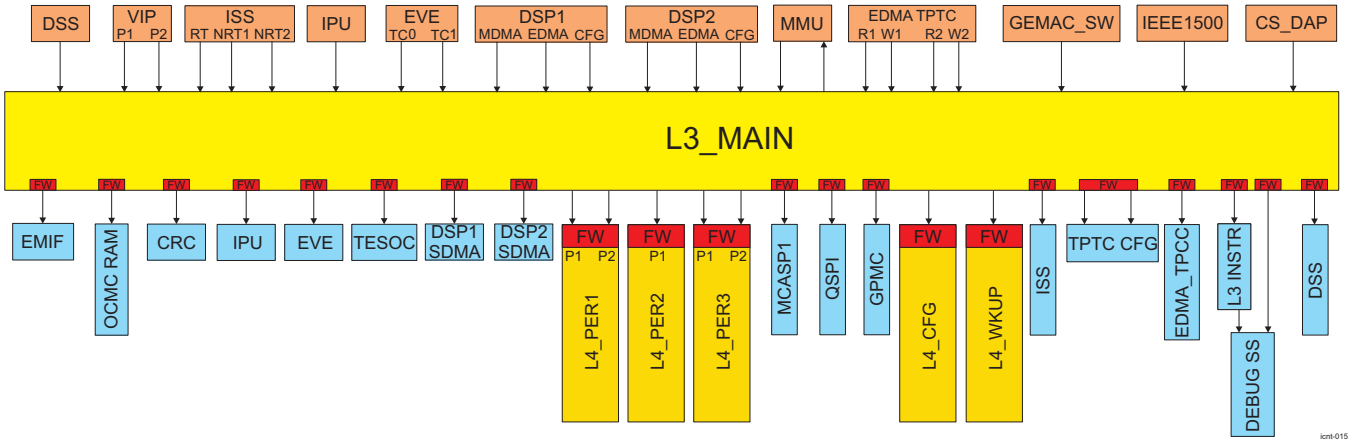
The main features of the L3_MAIN interconnects are:

- NIUs: Master NIUs for the IAs and slave NIUs for the TAs
- A partially depleted cross-bar exchange network
- A special internal slave NIU for accessing L3_MAIN interconnect configuration registers
- QoS management for real-time hardware operators, while maintaining optimal memory latency for CPU access to memory resources
- True little-endian platform
- Transaction errors tracking and logging registers
- All signaling support for chip-level power-management infrastructure
- One interrupt line signaling transaction error
- One interrupt line for reporting satistical events on the L3_MAIN interconnect

Figure 9-2 shows an overview of the L3 interconnect and the peripherals attached to it.

NOTE: ISS, CRC and TESOC are not supported on the DRA78x family of devices.

Figure 9-2. L3_MAIN Interconnect Overview



9.2.2 L3_MAIN Interconnect Integration

Table 9-4 through Table 9-6 summarize the integration of the module in the device.

Table 9-4. Integration Attributes

Module Instance	Attributes
	Power Domain
L3_MAIN	PD_COREAON

Table 9-5. Clocks and Resets

Clocks				
Module Instance	**Destination Signal Name**	**Source Signal Name**	**Source**	**Description**
L3_MAIN	L3_CLK1	L3MAIN1_L3_GICLK	PRCM	Functional and interface clock
	L3_CLK2	L3INSTR_L3_GICLK	PRCM	Functional and interface clock
Resets				
Module Instance	**Destination Signal Name**	**Source Signal Name**	**Source**	**Description**
L3_MAIN_MAIN	L3_CORE_RET_RST	CORE_PWRON_RET_RST	PRCM	Reset of L3_MAIN interconnect registers
	L3_CORE_RST	CORE_RST	PRCM	Reset of L3_MAIN interconnect

Table 9-6. Hardware Requests

Interrupt Requests				
Module Instance	**Source Signal Name**	**Destination IRQ_CROSSBAR Input**	**Defalut Mapping**	**Description**
L3_MAIN	L3_MAIN_IRQ_DBG_ERR	IRQ_CROSSBAR_4	DSP1_IRQ_35 DSP2_IRQ_35 EVE_IRQ_3	Interrupt indicating debug error occurrence.
	L3_MAIN_IRQ_APP_ERR	IRQ_CROSSBAR_5	DSP1_IRQ_36 DSP2_IRQ_36 EVE_IRQ_4	Interrupt indicating application error occurrence.
	L3_MAIN_IRQ_STAT_ALARM	IRQ_CROSSBAR_11	DSP1_IRQ_42 DSP2_IRQ_42	Statistic collector alarm interrupt.

NOTE: The "**Default Mapping**" column in Table 9-6 *Hardware Requests* shows the default mapping of module IRQ source signals. These IRQ source signals can also be mapped to other lines of each device Interrupt controller through the IRQ_CROSSBAR module.
For more information about the IRQ_CROSSBAR module, see Section 13.4.6.4, *IRQ_CROSSBAR Module Functional Description*, in Chapter 13, *Control Module*.
For more information about the device interrupt controllers, see Chapter 12, *Interrupt Controllers*.

9.2.3 L3_MAIN Interconnect Functional Description

9.2.3.1 Module Use in L3_MAIN Interconnect

The L3_MAIN interconnect network components have ConnID values for each master NIU or slave NIU. The ID uniquely identifies the master NIU or the slave NIU for an interconnect transfer. The interconnect uses ConnIDs for a number of purposes, including the following:

- Slave NIUs for error logging
- Power disconnect slave NIU for error logging
- FLAGMUXes for masking interrupts
- Statistic Collectors (STATCOLL) for configuring and monitoring: The STATCOLL components compute the traffic statistics within a user-defined window and periodically report to the user through the debug interface.
- Bandwidth regulator for configuration

9.2.3.2 Module Distribution

Master NIUs and slave NIUs provide the interface to connect the different modules to their associated interconnect.

Table 9-7 and Table 9-8 list all modules and subsystems with their associated agents. The agents are listed for each L3_MAIN interconnect domain.

9.2.3.2.1 L3_MAIN Interconnect Agents

Any initiator or target core is connected to the L3_MAIN interconnect through an NIU. NIUs act as entry points to the L3_MAIN interconnect, and also include various programming registers. Table 9-7 lists the supported master NIU ports.

Table 9-7. Master NIUs

Master NIU	Description
DSS1_INIT	Display SubSystem initiator port. One 128b initiator port
VIP_P1_INIT	Port 1 of VIP-128b initiator port.
VIP_P2_INIT	Port 2 of VIP-128b initiator port.
ISS_RT_INIT	ISS real time (RT) initiator port. One 128b initiator port. Note: ISS is not supported on the DRA78x family of devices.
ISS_NRT1_INIT	ISS non real time (NRT) initiator port 1. One 128b initiator port.
ISS_NRT2_INIT	ISS non real time (NRT) initiator port 2. One 128b initiator port.
IPU_INIT	IPU initiator port. One 64b initiator port
EVE_TC0_INIT	Embedded Vision Engine 1 (EVE) initiator port 1 (P1). One 128b initiator port.
EVE_TC1_INIT	Embedded Vision Engine 1 (EVE) initiator port 1 (P2). One 128b initiator port.
DSP1_MDMA_INIT	DSP1 Initiator port. One 128b MDMA interconnect initiator port (used also for cache requests)
DSP1_EDMA_INIT	DSP1 initiator port. One 128b EDMA interconnect initiator port.
DSP1_CFG_INIT	One 32b configuration initiator port.
DSP2_MDMA_INIT	DSP2 Initiator port. One 128b MDMA interconnect initiator port (used also for cache requests)
DSP2_EDMA_INIT	DSP2 initiator port. One 128b EDMA interconnect initiator port.
DSP2_CFG_INIT	One 32b configuration initiator port.
MMU_INIT	MMU initiator port.One 128b initiator port.
TPTC1_WR_INIT	EDMA TPTC initiator write port 1. One 128b initiator port.
TPTC1_RD_INIT	EDMA TPTC initiator read port 1. One 128b initiator port.
TPTC2_WR_INIT	EDMA TPTC initiator write port 2. One 128b initiator port.

Table 9-7. Master NIUs (continued)

Master NIU	Description
TPTC2_RD_INIT	EDMA TPTC initiator read port 2. One 128b initiator port.
GMAC_SW_INIT	GMAC_SW 32b initiator port.
IEEE1500_2_OCP_INIT	IEEE1500 32b initiator port.
DEBUGSS_CS_DAP_INIT	Debug subsystem CS_DAP 32b initiator port.

Table 9-8 lists the supported slave NIU ports.

Table 9-8. Slave NIUs

Slave NIU	Description
EMIF_P1_TARG	EMIF 128b target port.
OCMC_RAM_TARG	On-chip memory controller 128b target port.
CRC_TARG	CRC 64b target port. Note: CRC is not supported on the DRA78x family of devices.
IPU_TARG	IPU 64b target port.
EVE_TARG	EVE 128b target port.
TESOC_TARG	Tester-On-Chip 128b target port. Note: TESOC is not supported on the DRA78x family of devices.
DSP1_SDMA_TARG	128b target port for system requests to DSP1 memory or DSP1 address space.
DSP2_SDMA_TARG	128b target port for system requests to DSP2 memory or DSP2 address space.
L4_PER1_P1_TARG (L4_PER_P0_TARG)	L4_PER1 32b target port 1 (P1).
L4_PER1_P2_TARG (L4_PER_P1_TARG)	L4_PER1 32b target port 2 (P2).
L4_PER2_P1_TARG (L4_PER_P3_TARG)	L4_PER2 32b target port 1 (P1).
L4_PER3_P1_TARG (L4_PER_P6_TARG)	L4_PER3 32b target port 1 (P1).
L4_PER3_P2_TARG (L4_PER_P7_TARG)	L4_PER3 32b target port 2 (P2).
MCASP1_TARG	McASP 32b target port.
QSPI_TARG	QSPI 32b target port.
GPMC_TARG	GPMC 32b target port.
L4_CFG_TARG	L4_CFG 32b target port.
L4_WKUP_TARG	L4_WKUP 32b target port.
ISS_TARG	ISS 32b target port. Note: ISS is not supported on the DRA78x family of devices.
TPTC1_TARG	EDMA TPTC 32b target port 1.
TPTC2_TARG	EDMA TPTC 32b target port 2.
EDMA_TPCC_TARG	EDMA TPCC 32b target port.
L3_INSTR_TARG	L3 instrumentation 32b target port.
DEBUGSS_CT_TBR_TARG	Debug subsystem CT_TBR 64b target port.
DSS_TARG	Display subsystem 32b target port.
MMU_TARG	Memory management unit 128b target port.

9.2.3.2.2 L3_MAIN Connectivity Matrix

The L3 interconnect is divided into two clock domains L3_CLK1 and L3_CLK2. L3_CLK1 domain is further splitted into two sub groups:

- L3_CLK1_1: Low-power domain
- L3_CLK1_2: Peripherals and multimedia

L3_CLK2 is instrumentation (debug) associated. The two clock elements (L3_CLK1 and L3_CLK2) are implemented in different clock domains.

9.2.3.2.2.1 Clock Domain Mapping of the L3_MAIN Interconnect Modules

Each clock domain (L3_CLK1 and L3_CLK2) has it own host, flag mux, slave NIUs, and bandwidth regulators. Table 9-9 lists the relationships between these domains and these elements.

Table 9-9. L3_MAIN Clock Domains and Elements

Clock Domain	Elements
L3_CLK1_1	HOST_CLK1_1
	GPMC_TARG
	EMIF_P1_TARG
	DSP1_SDMA_TARG
	L4_CFG_TARG
	DSP2_SDMA_TARG
	CRC_TARG Note: CRC is not supported on the DRA78x family of devices.
	ISS_TARG Note: ISS is not supported on the DRA78x family of devices.
	EVE_TARG
	OCMC_RAM_TARG
	IPU_TARG
	TESOC_TARG Note: TESOC is not supported on the DRA78x family of devices.
	L4_PER1_P1_TARG
	L4_WKUP_TARG
	L4_PER1_P2_TARG
	EDMA_TPCC_TARG
	L4_PER2_P1_TARG
	MMU_TARG
	L4_PER3_P1_TARG
	L4_PER3_P2_TARG
	DSS_TARG
	TPTC2_TARG
	TPTC1_TARG
	MCASP1_TARG
	QSPI_TARG
	HOST_CLK1_2
	CLK1_FLAGMUX_CLK1MERGE
	CLK1_FLAGMUX_CLK1_1
	CLK1_FLAGMUX_CLK1_2
	MMU_BW_LIMITER
	IPU_BW_LIMITER
	EVE_TC0_BW_REGULATOR
	EVE_TC1_BW_REGULATOR
	DSP2_EDMA_BW_LIMITER
	DSP1_EDMA_BW_LIMITER
	DSP1_MDMA_BW_REGULATOR

Table 9-9. L3_MAIN Clock Domains and Elements (continued)

Clock Domain	Elements
L3_CLK1_2	IPU_BW_REGULATOR
	DSP2_MDMA_BW_REGULATOR
	GMAC_SW_BW_REGULATOR
	FLAGMUX_CLK1_TIMEOUT1
	FLAGMUX_CLK1_TIMEOUT2
	DSP1_MDMA_BW_LIMITER
	DSP2_MDMA_BW_LIMITER
	EVE_TC0_BW_LIMITER
	EVE_TC1_BW_LIMITER
	ISS_NRT2_BW_LIMITER Note: ISS is not supported on the DRA78x family of devices.
	ISS_NRT1_BW_LIMITER
	TPTC1_RD_BW_LIMITER
	TPTC1_WR_BW_LIMITER
	TPTC2_RD_BW_LIMITER
	TPTC2_WR_BW_LIMITER
L3_CLK2_1	HOST_CLK2_1
	L3_INSTR_TARG
	CLK2_FLAGMUX_CLK2_1
	DEBUGSS_CT_TBR_TARG
	FLAGMUX_CLK2_TIMEOUT
	FLAGMUX_STATCOLLS
	STATCOLL1
	STATCOLL2
	STATCOLL3
	STATCOLL4

Figure 9-3shows the functional paths between the L3_MAIN master NIUs and the L3_MAIN and L4 slave NIU agents. The functional paths in L3_MAIN are indicated by the following:

- A cell contains the letter C when a functional path exists.
- A cell is empty when a functional path does not exist.

NOTE: ISS, CRC and TESOC are not supported on the DRA78x family of devices.

Figure 9-3. Connectivity Matrix

		TARGETS																											
		EMIF_P1_TARG	OCMC_RAM_TARG	DSP1_SDMA_TARG	DSP2_SDMA_TARG	EVE_TARG	CRC_TARG	TESOC_TARG	IPU_TARG	GPMC_TARG	QSPI_TARG	TPTC1_TARG	TPTC2_TARG	EDMA_TPCC_TARG	MCASP1_TARG	DSS_TARG	ISS_TARG	L4_PER1_P1_TARG	L4_PER1_P2_TARG	L4_PER2_P1_TARG	L4_PER3_P1_TARG	L4_PER3_P2_TARG	L4_WKUP_TARG	L4_CFG_TARG	L3_MAIN_SN	L3_INSTR_TARG	DEBUGSS_CT_TBR_TARG	MMU_TARG
INITIATORS	DSP1_MDMA_INIT	C	C		C	C	C	C	C	C	C	C	C	C	C	C	C		C	C		C	C	C	C	C	C	
	DSP1_EDMA_INIT	C	C		C	C	C	C	C	C	C	C	C	C	C	C	C	C		C	C		C	C	C	C	C	C	
	DSP1_CFG_INIT					C						C	C	C														
	DSP2_MDMA_INIT	C	C	C		C	C	C	C	C	C	C	C	C	C	C	C		C	C		C	C	C	C	C	C	
	DSP2_EDMA_INIT	C	C	C		C	C	C	C	C	C	C	C	C	C	C	C	C		C	C		C	C	C	C	C	C	
	DSP2_CFG_INIT					C						C	C	C														
	EVE_TC0_INIT	C	C	C	C		C	C	C	C	C	C	C	C	C	C	C	C		C	C		C	C	C	C	C	
	EVE_TC1_INIT	C	C	C	C		C	C	C	C	C	C	C	C	C	C	C	C		C	C		C	C	C	C	C	
	IPU_INIT	C	C	C	C	C	C	C		C	C	C	C	C	C	C	C		C	C		C	C	C	C	C	C	
	VIP_P1_INIT	C	C	C	C	C	C	C	C	C	C														C	C	C	
	VIP_P2_INIT	C	C	C	C	C	C	C	C	C	C														C	C	C	
	DSS1_INIT	C	C	C	C	C	C	C	C	C	C														C	C	C	
	ISS_RT_INIT	C	C	C	C	C	C	C	C	C	C														C	C	C	
	ISS_NRT1_INIT	C	C	C	C	C	C	C	C	C	C														C	C	C	
	ISS_NRT2_INIT	C	C	C	C	C	C	C	C	C	C														C	C	C	
	TPTC1_RD_INIT	C	C	C	C	C	C	C	C	C	C		C	C	C	C	C	C		C	C		C	C	C	C	C	C
	TPTC1_WR_INIT	C	C	C	C	C	C	C	C	C	C		C	C	C	C	C	C		C	C		C	C	C	C		C
	TPTC2_RD_INIT	C	C	C	C	C	C	C	C	C	C	C		C	C	C	C	C		C	C		C	C	C	C	C	C
	TPTC2_WR_INIT	C	C	C	C	C	C	C	C	C	C	C		C	C	C	C	C		C	C		C	C	C	C		C
	MMU_INIT	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C		C	C		C	C	C	C	C	
	GMAC_SW_INIT	C	C	C	C	C	C	C	C	C	C																	
	DEBUGSS_CS_DAP_INIT	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C		C	C		C	C	C	C	C	C	
	IEEE1500_2_OCP_INIT	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C	C		C	C		C	C	C	C	C	C	

icnt-002

9.2.3.2.3 Master NIU Identification

A master NIU ID (ConnID) is assigned to every module in the device. The ID uniquely identifies the master NIU for an interconnect transfer (see Table 9-10). The interconnect uses ConnID values for a number of purposes, including:

- Master source identification for the protection mechanism
- Response route generation
- Firewall error logging
- L3_MAIN interconnect error logging

All L3_MAIN firewalls use 4-bit ConnID values from several registers in the control module instead of the L3_MAIN ConnIDs. Thus appropriate grouping of initiators based on a configurable ConnID can be done. The 4-bit ConnID is equal to bits[7:4] of the L3_MAIN 8-bit ConnID. The reset values of these control module registers mach the value of bits[7:4]. The registers are the following:

- CTRL_CORE_FIREWALL_CONNID_CONTROL_0

- CTRL_CORE_FIREWALL_CONNID_CONTROL_1
- CTRL_CORE_FIREWALL_CONNID_CONTROL_2
- CTRL_CORE_FIREWALL_CONNID_CONTROL_3

Table 9-10. ConnID Values

8-bit ConnID (hex)	4-bit ConnID (hex)	Master NIU
10	1	DEBUGSS_CS_DAP_INIT
14	1	IEEE1500_2_OCP_INIT
20	2	DSP1_MDMA_INIT
24	2	DSP1_CFG_INIT
28	2	DSP1_EDMA_INIT
2C	2	DSP2_EDMA_INIT
30	3	DSP2_CFG_INIT
34	3	DSP2_MDMA_INIT
42	4	EVE_TC0_INIT
60	6	IPU_INIT
70	7	TPTC1_WR_INIT
72	7	TPTC1_RD_INIT
74	7	TPTC2_WR_INIT
76	7	TPTC2_RD_INIT
80	8	DSS1_INIT
86	8	MMU_INIT
90	9	VIP_P1_INIT
92	9	VIP_P2_INIT
AC	A	GMAC_SW_INIT
C0	C	ISS_RT_INIT Note: ISS is not supported on the DRA78x family of devices.
C4	C	ISS_NRT1_INIT
C8	C	ISS_NRT2_INIT
D2	D	EVE_TC1_INIT
E0	E	STATCOLL1
E0	E	STATCOLL2
E0	E	STATCOLL3
E0	E	STATCOLL4

The 8-bit ConnID values are used by error decoding to distinguish the different initiators, see Section 9.2.3.8.4, *Example for Decoding Standard/Custom Errors Logged in L3_MAIN*. They are also used by the EMIF controller. The 4-bit ConnID values are used by the firewalls to allow or not an access to a slave NIU, see Table 9-16.

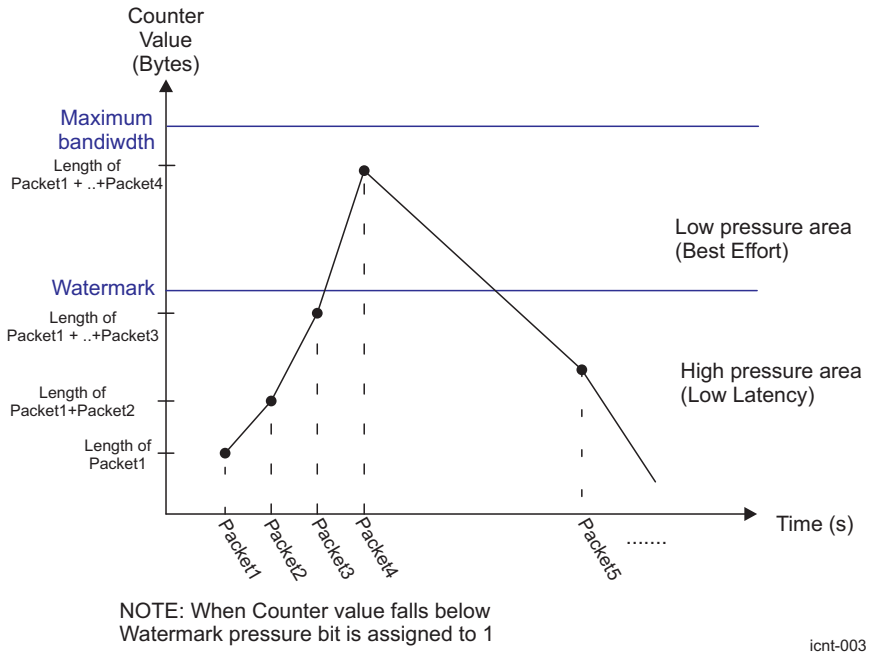
9.2.3.3 Bandwidth Regulators

The bandwidth regulators prevent master NIUs from consuming too much bandwidth of a link, or a slave NIU that is shared between several data flows: packets are then transported at a slower rate. The value of a bandwidth can be programmed in the bandwidth regulator. When the bandwidth is below the programmed value, the pressure bit is set to 1, giving priority to this master. When the bandwidth is above the programmed value, the pressure bit is set to 0 and the concerned master has the same weight as others.

A counter is used to store the sum of data lengths (in bytes) of each packet passing through the bandwidth regulator, and a value equal to the expected bandwidth is subtracted from the counter at each clock cycle. The value of the counter is compared to a programmable threshold (called Watermark), and this comparison determines whether the packet is processed with high pressure for minimum latency or low pressure for best effort processing.

The bandwidth regulator monitors the traffic using open connections between the initiators and the targets. If there is insufficient bandwidth allocated to the connection, the bandwidth regulator can increase the pressure on connections. Generally, the connection is a dataflow between master and slave NIUs. In some cases, the bandwidth regulator is attached to the master and monitors single dataflow to a target (single connection).

Figure 9-4. Bandwidth Regulator Pressure Settings



When Counter Value (bytes) falls below Watermark, the pressure bit is assigned to 1

The bandwidth regulator effective resolution is set to 8.3125 MBps. The maximum average bandwidth (max watermark value) computed in a moving window is set to up to 4096 bytes of payload. These settings are implemented by hardware.

The following is an example of bandwidth regulator settings:

Suppose the bandwidth regulator is set to run at 200 MHz and the application requires an expected bandwidth of 165.888 MBps (±5 MBps), computed through a moving window of 5 µs (1000 cycles). To attribute high pressure on all packet requests, the watermark could be set to the maximum bandwidth needed in the 5-µs window.

Considering the example, the settings of the bandwidth regulator are:

- L3_BW_REGULATOR_WATERMARK[11:0] WATERMARK = moving window × expected bandwidth = 829.44 bytes = 0x33D
- L3_BW_REGULATOR_BANDWIDTH[15:0] BANDWIDTH = ceil(165.888/8.3125) = ceil(19.956) = 20d = 0x14

The bandwidth registers regulate the packet flow by applying flow control on the RX port, thus ensuring that the traffic does not exceed the allocated bandwidth. The next packet is sent only when an internal timer expires. The registers in this group are:

- L3_BW_REGULATOR_WATERMARK: Gives the amount of data allowed to exceed the average bandwidth during a short time period
- L3_BW_REGULATOR_PRESS: Describes the pressure applied to outgoing packets
- L3_BW_REGULATOR_CLEARHISTORY: Resets the traffic counter when set to 1. This register is

used after an update in the L3_BW_REGULATOR_BANDWIDTH and L3_BW_REGULATOR_WATERMARK registers (see Section 9.2.5.1.7, *L3 BW Regulator Register Summary and Description*).

Bandwidth regulators are mainly used to give priority to the following master NIUs:

- IPU_INIT
- EVE_TC0_INIT
- EVE_TC1_INIT
- DSP1_MDMA_INIT
- DSP2_MDMA_INIT
- GMAC_SW_INIT

Priority to the following master NIUs is given by setting their internal MFlag signals:

- DSS1_INIT
- VIP_P1_INIT
- VIP_P2_INIT
- ISS_RT_INIT

NOTE: ISS is not supported on the DRA78x family of devices.

9.2.3.4 Bandwidth Limiters

Bandwidth limiters are added to control the bandwidth of:

- ISS_NRT1_INIT
- ISS_NRT2_INIT
- *IPU_INIT*
- *EVE_TC0_INIT*
- *EVE_TC1_INIT*
- *DSP1_MDMA_INIT*
- DSP1_EDMA_INIT
- *DSP2_MDMA_INIT*
- DSP2_EDMA_INIT
- MMU_INIT
- TPTC1_WR_INIT
- TPTC1_RD_INIT
- TPTC2_WR_INIT
- TPTC2_RD_INIT

This prevents a large number of RD requests being processed together, thus avoiding a large number of RD responses.

NOTE: The following initiator NIUs have also bandwith regulators:

- *IPU_INIT*
- *EVE_TC0_INIT*
- *EVE_TC1_INIT*
- *DSP1_MDMA_INIT*
- *DSP2_MDMA_INIT*

The bandwidth limiter regulates the packet flow in the L3_MAIN interconnect by applying flow control when a user-defined bandwidth limit is reached. The next packet is served only after an internal timer expires, thus ensuring that traffic does not exceed the allocated bandwidth. Bandwidth limiter can be used with a watermark mechanism that allows traffic to temporarily exceeds the peak bandwidth.

The registers in this group are:

- L3_BW_LIMITER_WATERMARK_0: Gives the amount of data allowed to exceed the average bandwidth during a short time period. To set the actual watermark to n bytes, the register must be set to n + 1.
- L3_BW_LIMITER_CLEARHISTORY: Resets the traffic counter when set to 1.
- L3_BW_LIMITER_BANDWIDTH_FRACTIONAL and L3_BW_LIMITER_BANDWIDTH_INTEGER: These two registers are used to set the average payload bandwidth.

Example of setting the Bandwidth Limiters:

L3_BW_LIMITER_BANDWIDTH_FRACTIONAL must be set to BandwidthConf [4 down to 0] and L3_BW_LIMITER_BANDWIDTH_INTEGER contains the remaining BandiwdthConf bits, shifted to the right. BandwidthConf = AverageBandwidth / EffectiveResolution, where EffectiveResolution parameter is set to 8.3125 MHz at design time

AverageBandwidth is a parameter representing the value to which the payload bandwidth must be limited in average; this parameter depends on the use case and is the reason that makes the content of the registers variable. In addition the maximum packet length is 8 cells, that is 32 bytes.

Assuming that the use case requires that in average the payload bandwidth is limited to 200 MB/s, then registers must be set to the following values:

BandwidthConf = 200 MBps / 8.3125 MHz = 38 (0x26)

L3_BW_LIMITER_BANDWIDTH_FRACTIONAL = LSBs [4 down to 0] of BandwidthConf = 0x6.

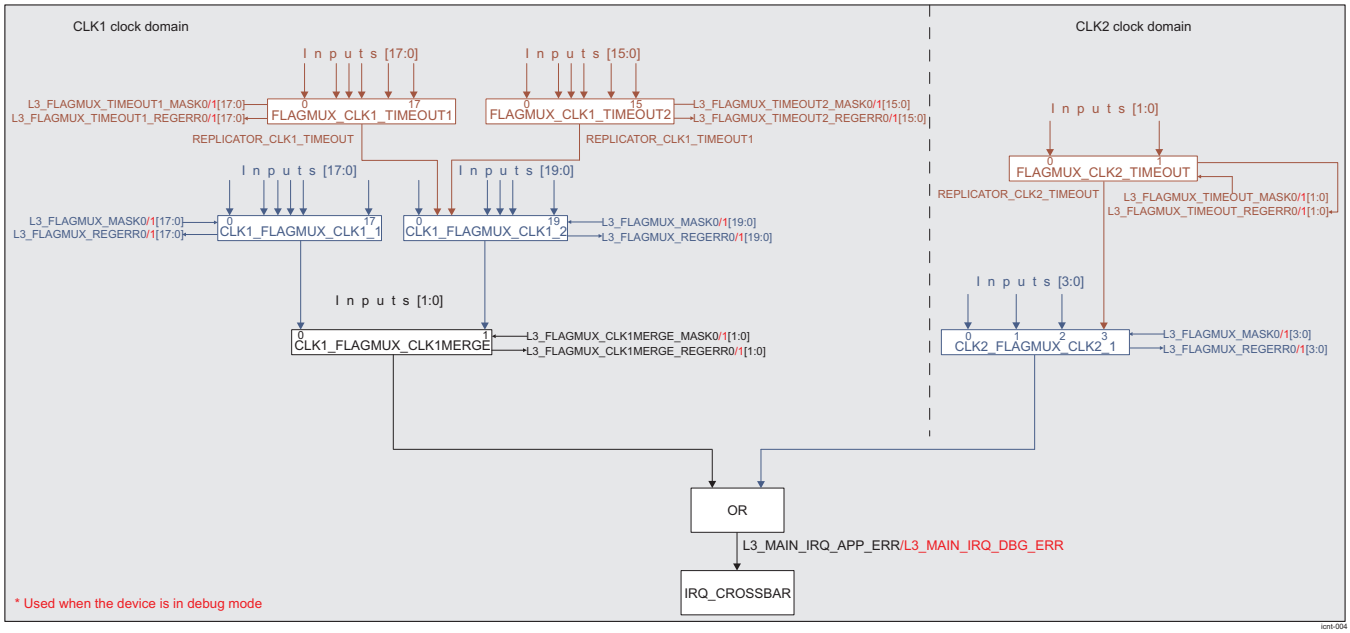
L3_BW_LIMITER_BANDWIDTH_INTEGER = the remaining bits of BandwidthConf shifted to the right = 0x1.

By setting L3_BW_LIMITER_WATERMARK to 65 (L3_BW_LIMITER_WATERMARK_0 = 0x41), the bandwidth limiter is able to send three packets consecutively during peaks, therefore exceeding the peak traffic with two packets (64 bytes)

9.2.3.5 Flag Muxing

Flag muxing is used to collect information such as errors and interrupts from slave NIUs and interconnect firewalls. The result signals are then sent to the IRQ_CROSSBAR module without interfering with the interconnect traffic. Figure 9-5 shows the flag muxing scheme.

Figure 9-5. Flag Muxing Scheme



The following three flag muxes collect information from targets in L3_CLK1 and L3_CLK2 clock domains:

- CLK1_FLAGMUX_CLK1_1
- CLK1_FLAGMUX_CLK1_2
- CLK2_FLAGMUX_CLK2_1

There are three L3_FLAGMUX_MASK0 registers each associated with a single flag mux of the previously listed. Each bit in these registers is associated with only one source. If a bit is set to 0x0, the corresponding flag is masked which means that the input source is disabled, and if set to 0x1 the flag is unmasked. There are also three L3_FLAGMUX_REGERR0 status registers for checking the source of error. A value of 0x1 indicates that the corresponding flag is set, that is, an error is generated. Table 9-11 shows the mapping between the input sources and the bits in L3_FLAGMUX_MASK0 and L3_FLAGMUX_REGERR0 registers. The missing values are reserved and writing to these bits has no effect.

Table 9-11. Flag Mux Input Mapping

Flag Mux	Corresponding bit in registers L3_FLAGMUX_MASK0 and L3_FLAGMUX_REGERR0 (Flag Mux Input)	Flag Mux Input Sources
CLK1_FLAGMUX_CLK1_1	1	EMIF_P1
	2	DSP2_SDMA
	3	TESOC Note: TESOC is not supported on the DRA78x family of devices.
	5	DSP1_SDMA
	6	EVE
	7	DSS
	8	GPMC
	10	L4_CFG
	11	L4_WKUP
	13	IPU
	14	EDMA_TPCC
	15	TPTC1 (EDMA)

Table 9-11. Flag Mux Input Mapping (continued)

Flag Mux	Corresponding bit in registers L3_FLAGMUX_MASK0 and L3_FLAGMUX_REGERR0 (Flag Mux Input)	Flag Mux Input Sources
	16	TPTC2 (EDMA)
	17	MMU
CLK1_FLAGMUX_CLK1_2	0	HOST_CLK1_1
	1	HOST_CLK1_2
	2	REPLICATOR_CLK1_TIMEOUT
	4	REPLICATOR_CLK1_TIMEOUT1
	5	L4_PER2_P1 (L4_PER_P3)
	6	L4_PER1_P1 (L4_PER_P0)
	7	L4_PER1_P2 (L4_PER_P1)
	8	L4_PER3_P2 (L4_PER_P7)
	9	L4_PER3_P1 (L4_PER_P6)
	10	MCASP1
	12	OCMC_RAM
	16	QSPI
	18	CRC Note: CRC is not supported on the DRA78x family of devices.
	19	ISS Note: ISS is not supported on the DRA78x family of devices.
CLK2_FLAGMUX_CLK2_1	0	L3_INSTR
	1	DEBUGSS_CT_TBR
	2	HOST_CLK2_1
	3	REPLICATOR_CLK2_TIMEOUT

Each of the three flag muxes has also the following two registers:

- L3_FLAGMUX_STDHOSTHDR_COREREG register for identification of the attached core type
- L3_FLAGMUX_STDHOSTHDR_VERSIONREG register for identification of the characteristics of the attached core

There is also a separate mux (CLK1_FLAGMUX_CLK1MERGE) that merges the outputs of the CLK1_FLAGMUX_CLK1_1 and CLK1_FLAGMUX_CLK1_2 flag muxes. Its functionality follows the same logic as the functionality of the flag muxes previously described. The following registers are available:

- L3_FLAGMUX_CLK1MERGE_MASK0 register for enabling/disabling the corresponding input sources
- L3_FLAGMUX_CLK1MERGE_REGERR0 status register for checking the source of error
- L3_FLAGMUX_CLK1MERGE_STDHOSTHDR_COREREG register for identification of the attached core type
- L3_FLAGMUX_CLK1MERGE_STDHOSTHDR_VERSIONREG register for identification of the characteristics of the attached core

In addition, it must be taken into account that the following registers (already described) with suffix "0" are used when the device is in application mode:

- L3_FLAGMUX_MASK0
- L3_FLAGMUX_REGERR0
- L3_FLAGMUX_CLK1MERGE_MASK0
- L3_FLAGMUX_CLK1MERGE_REGERR0

The following registers with suffix "1" are used when the device is in debug mode:

- L3_FLAGMUX_MASK1

- L3_FLAGMUX_REGERR1
- L3_FLAGMUX_CLK1MERGE_MASK1
- L3_FLAGMUX_CLK1MERGE_REGERR1

Both register groups have identical bits and follow same logic. The only difference is that registers with suffix "0" are used in application mode and registers with suffix "1" are used in debug mode.

9.2.3.5.1 Time-out Flag Muxing

If a target does not respond after a fixed number of clock cycles, a time-out error flag is generated if enabled. There are three time-out flag muxes:

- FLAGMUX_CLK1_TIMEOUT1 with registers L3_FLAGMUX_TIMEOUT1_MASK0 and L3_FLAGMUX_TIMEOUT1_REGERR0
- FLAGMUX_CLK1_TIMEOUT2 with registers L3_FLAGMUX_TIMEOUT2_MASK0 and L3_FLAGMUX_TIMEOUT2_REGERR0
- FLAGMUX_CLK2_TIMEOUT with registers L3_FLAGMUX_TIMEOUT_MASK0 and L3_FLAGMUX_TIMEOUT_REGERR0

Each bit in the _MASK0 registers previously listed is associated with only one time-out source. If a bit is set to 0x0, the corresponding time-out flag is masked, and if set to 0x1 the flag is unmasked. To check the source of time-out error the corresponding bit in one of the three previously listed _REGERR0 registers should be read. A value of 0x1 indicates that a time-out error has occurred.

Table 9-12 shows the mapping between the time-out input sources and the bits in _MASK0 and _REGERR0 registers. The missing values are reserved and writing to these bits has no effect.

Table 9-12. L3 Time-out Flag Mapping

Time-out Flag Mux	Flag Mux Input (Corresponding bit in registers _MASK0 and _REGERR0)	Flag Mux Time-out Input Sources
FLAGMUX_CLK1_TIMEOUT1	2	DSP1_SDMA
	3	DSP2_SDMA
	4	DSS
	5	EVE
	6	IPU
	7	MMU
	8	EMIF_P1
	9	TESOC Note: TESOC is not supported on the DRA78x family of devices.
	11	L4_PER_P0
	12	L4_PER_P1
	13	L4_PER_P3
	14	L4_PER_P7
	15	L4_PER_P6
	16	L4_CFG
	17	L4_WKUP
FLAGMUX_CLK1_TIMEOUT2	2	ISS Note: ISS is not supported on the DRA78x family of devices.
	4	QSPI
	6	EDMA_TPCC
	7	TPTC1
	8	TPTC2
	11	MCASP1
	12	OCMC_RAM

Table 9-12. L3 Time-out Flag Mapping (continued)

Time-out Flag Mux	Flag Mux Input (Corresponding bit in registers _MASK0 and _REGERR0)	Flag Mux Time-out Input Sources
	14	GPMC
	15	CRC Note: CRC is not supported on the DRA78x family of devices.
FLAGMUX_CLK2_TIMEOUT	0	L3_INSTR
	1	DEBUGSS_CT_TBR

For example, to enable IPU time-out error the L3_FLAGMUX_TIMEOUT1_MASK0[6] bit must be set to 0x1. To check whether IPU time-out error has occurred the L3_FLAGMUX_TIMEOUT1_REGERR0[6] bit must be read.

9.2.3.6 Statistic Collectors Group

Statistic collectors are internal masters that share the same master address as the master NIUs. These components compute the traffic statistics within a defined window and periodically report through the DEBUG interface. The key features of the statistic collector are:

- Nonintrusive monitoring
- Programmable filters and counters
- Collects results at a programmable time interval

Event detectors are programmed through the L3_STCOL_REQEVT and L3_STCOL_RSPEVT configuration registers for request and response ports, respectively. The following events can be identified:

- Word transfer
- WAIT cycles
- Flow control
- Payload transfers
- Latency measurements

Performance monitoring is enabled through the L3_STCOL_EN register. The L3_STCOL_SOFTEN register enables software to monitor the performance. Event muxes are programmed through the L3_STCOL_EVTMUX_SEL0 configuration register, which determines which port will be monitored by a filter configured by the filter registers (see Section 9.2.5.1.9).

Filters are programmed through the L3_STCOL_FILTER_i_GLOBALEN configuration register, along with additional selection criteria programmed through the mask/match registers (see Table 9-221). A filter can be configured to accept or reject:

- Read operations
- Write operations
- Errors
- Addresses

Filter operation is programmed through the L3_STCOL_OP registers (see Table 9-221).

There are four statistic collectors used for traffic monitoring on device ports. For more detailed description regarding statistic collectors, see *L3 Target Load Monitoring* and *L3 Master Latency Monitoring* in Chapter 26 *On-Chip Debug Support*.

9.2.3.7 L3_MAIN Protection and Firewalls

Device protection relies on L3 firewalls and their configuration.

9.2.3.7.1 L3_MAIN Firewall Reset

The values of the L3_MAIN firewall registers upon reset are tied in hardware or exported from the control module registers.

Values exported from the control module are intended to give defined rights to the firewalls at reset and thus ensure the content after going out of reset.

The L3_MAIN firewall registers are located in the CORE_AON power domain and thus no retention capability is needed. The control module registers are reset by a cold reset only, whereas the L3_MAIN firewall registers are reset by clearing the REGUPDATE_CONTROL[1] FW_LOAD_REQ bit. When the REGUPDATE_CONTROL[1] FW_LOAD_REQ bit comes back automatically to 1, the exported values are loaded.

CAUTION

Before reprogramming the firewall registers and/or before using the FW_LOAD_REQ mechanism, the request must be asserted by configuring the REGUPDATE_CONTROL[0] BUSY_REQ bit.

To load the exported values at run time:

1. Set the REGUPDATE_CONTROL[0] BUSY_REQ bit to 0x1 to ensure that no transaction can reach the slave NIU (suspend).
2. Clear the REGUPDATE_CONTROL[1] FW_LOAD_REQ bit by writing 0x1 to it.
3. Wait until the REGUPDATE_CONTROL[1] FW_LOAD_REQ bit is reset to 0x1 by hardware.
4. Set the REGUPDATE_CONTROL[0] BUSY_REQ bit to 0x0 to allow transactions to reach the slave NIU (resume).

To reprogram the firewall registers at run time:

1. Write 0x1 to the REGUPDATE_CONTROL register.
2. Update the firewall registers.
3. Write 0x0 to the REGUPDATE_CONTROL register.

NOTE: While reprogramming the firewall registers at run time it must be taken into account that the REGUPDATE_CONTROL[1] FW_LOAD_REQ bit is written as '0' because a value of '1' reloads the firewall default values.

NOTE: At reset, exported values from the control module can modify hardware reset values.

9.2.3.7.2 Power Management

As part of the system-wide power-management scheme, the L3_MAIN interconnect goes into IDLE state after receiving a request from the power, reset, and clock management (PRCM) module after all commands are serviced. This function is handled by hardware.

To reduce power consumption, the L3_MAIN interconnect automatically performs internal clock autogating. This is managed by hardware; no software configurations or settings are required.

L3_MAIN supports a partial retention scheme. Retention is performed on the follwoing registers:

- Statistic collectors
- Bandwidth regulators
- Bandwidth limiters
- Firewalls
- Flag muxes

This process prevents reconfiguration after a clock domain switches off.

9.2.3.7.3 L3_MAIN Firewall Functionality

The access to the slave NIUs is granted only to master NIUs according to in-band attributes sent in each transaction crossing the L3_MAIN interconnect, such as:

- MCMD: Specifies the type of access (read or write) required by the master NIU
- ConnID: Used to determine the permission of the master NIU
- MReqInfo: Transaction attribute adding information about the access type

Table 9-13 lists the MReqInfo values.

Table 9-13. MReqInfo Values

Qualifier	Access Definition	Access Description
MReqDebug	0: Functional 1: Debug	When set, indicates that the request has been issued by a master NIU in DEBUG state
MReqType	00: Processor data access 01: Processor instruction access 10: DMA access 11: Other	Indicates whether the request is for processor instruction fetch, processor data access or DMA access
MReqSupervisor	0: User 1: Privilege	When set, indicates that the request is qualified with the supervisor attribute. It can be provided by a processor running in supervisor mode or by a module that inherited this attribute from the processor (DMA channel with a supervisor attribute).

The firewall comparison mechanism enables access to a protected slave NIU only when a correct combination of three MReqInfo in-band parameters is transmitted.

MReqInfo is a combination of a fixed 4-bit pattern that corresponds to a combination of the parameters MReqDebug, MReqType, and MReqSupervisor. See Table 9-14.

Table 9-14. L3_MAIN ReqInfo Mapping

ReqInfo Name		MReqDebug	MReqType	MReqSupervisor
Master NIUs	MMU_INIT	x		x
	TPTC1_RD_INIT			x
	TPTC1_WR_INIT			x
	TPTC2_RD_INIT			x
	TPTC2_WR_INIT			x
	ISS_RT_INIT Note: ISS is not supported on the DRA78x family of devices.	x		
	ISS_NRT1_INIT	x		
	ISS_NRT2_INIT	x		
	VIP_P1_INIT			
	VIP_P2_INIT			
	EVE_TC0_INIT	x	x	
	EVE_TC1_INIT	x	x	
	DSP1_EDMA_INIT	x	x	x
	DSP1_MDMA_INIT	x	x	x
	DSP2_EDMA_INIT	x	x	x
	DSP2_MDMA_INIT	x	x	x
	DSS1_INIT			
	IPU_INIT	x	x	x
	DSP1_CFG_INIT	x	x	x
	DSP2_CFG_INIT	x	x	x
	GMAC_SW_INIT			

Table 9-14. L3_MAIN ReqInfo Mapping (continued)

ReqInfo Name		MReqDebug	MReqType	MReqSupervisor
	DEBUGSS_CS_DAP_INIT	x		x
Slave NIUs	EMIF_P1_TARG	x	x	x
	CRC_TARG Note: CRC is not supported on the DRA78x family of devices.	x		
	DSP1_SDMA_TARG	x	x	x
	DSP2_SDMA_TARG	x	x	x
	EVE_TARG	x		
	TESOC_TARG Note: TESOC is not supported on the DRA78x family of devices.			
	L4_CFG_TARG	x		x
	L4_WKUP_TARG	x		x
	ISS_TARG			
	TPTC1_TARG	x		x
	TPTC2_TARG	x		x
	EDMA_TPCC_TARG	x	x	x
	L3_INSTR_TARG	x		
	DEBUGSS_CT_TBR_TARG	x		
	OCMC_RAM_TARG			
	IPU_TARG			
	GPMC_TARG			
	L4_PER1_P1_TARG	x		x
	L4_PER1_P2_TARG	x		x
	L4_PER2_P1_TARG	x		x
	L4_PER3_P1_TARG	x		x
	L4_PER3_P2_TARG	x		x
	QSPI_TARG			
	MCASP1_TARG			
	DSS_TARG			x
	MMU_TARG	x		x

9.2.3.7.3.1 Protection Regions

Each slave NIU address space is subdivided into protection regions. The regions are configurable with a size of 4-KiB granularity.

Table 9-15 lists the number of protected regions for each slave NIU.

Table 9-15. Slave NIU Firewall and Region Configuration

Firewall name	Slave NIU	Number of Firewall Regions
EMIF_FW	EMIF_P1_TARG	24
OCMC_RAM_FW	OCMC_RAM_TARG	24
CRC_FW Note: CRC is not supported on the DRA78x family of devices.	CRC_TARG	4
IPU_FW	IPU_TARG	4
EVE_FW	EVE_TARG	1

Table 9-15. Slave NIU Firewall and Region Configuration (continued)

Firewall name	Slave NIU	Number of Firewall Regions
TESOC_FW Note: TESOC is not supported on the DRA78x family of devices.	TESOC_TARG	4
DSP1_SDMA_FW	DSP1_SDMA_TARG	1
DSP2_SDMA_FW	DSP2_SDMA_TARG	1
MCASP1_FW	MCASP1_TARG	1
QSPI_FW	QSPI_TARG	4
GPMC_FW	GPMC_TARG	8
ISS_FW Note: ISS is not supported on the DRA78x family of devices.	ISS_TARG	4
TPTC_FW	TPTC1_TARG and TPTC2_TARG	2
EDMA_TPCC_FW	EDMA_TPCC_TARG	1
L3_INSTR_FW	L3_INSTR_TARG	1
DEBUGSS_CT_TBR_FW	DEBUGSS_CT_TBR_TARG	2
DSS_FW	DSS_TARG	8

Two types of regions are distinguished in a slave NIU firewall:

- Default region: Available in all slave NIUs. The default region covers the entire slave NIU address range. Other firewall-configured regions must reset or overlay the default region, because it always has the lowest priority.
- Normal region: The number of normal regions varies in a slave NIU; they have identical capabilities (see Table 9-15).

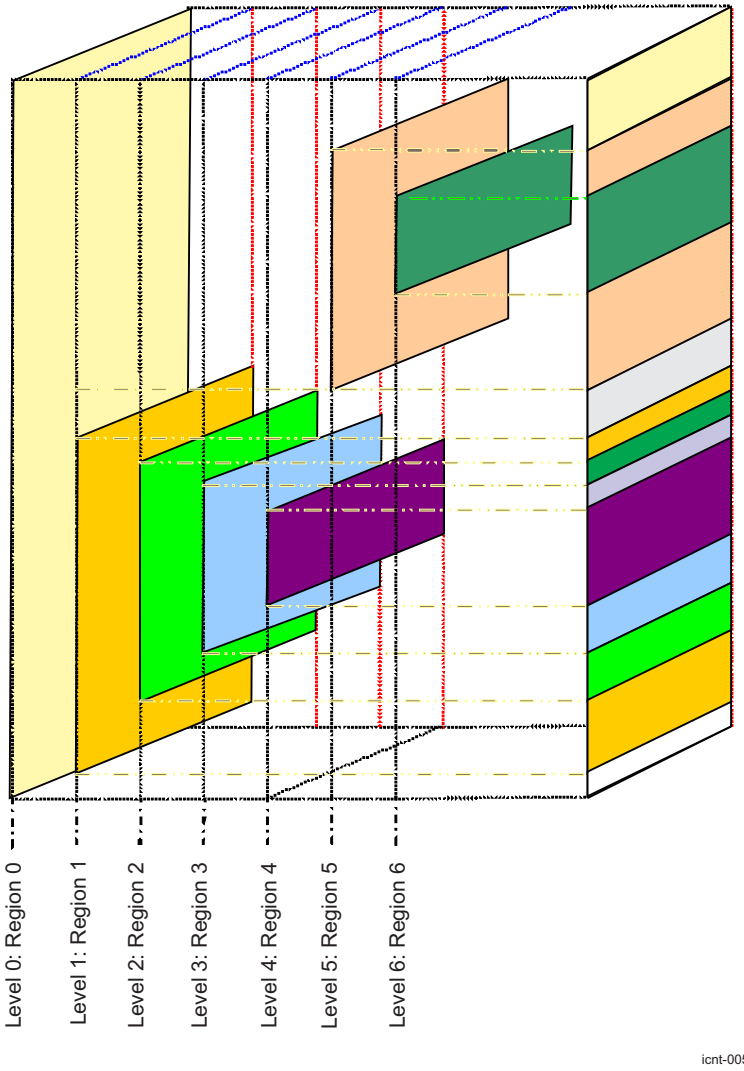
Each region has the following characteristics:

- Start address: Physical slave NIU start address
- End address: Physical slave NIU end address
- Specific access rights (see Section 9.2.3.7.3.3, *Protection Mechanism per Region Examples*)
- Priority level

Depending on its priority level, a region can override the settings of another region; the access rights of the region with the highest priority apply. All regions have a fixed (not configurable) priority level that corresponds to their number: Region 0 has priority level 0 (lowest), region 1 has priority level 1, and so on.

Figure 9-6 shows the priority level with associated regions. This priority level scheme allows multiplying the flexibility and capability of the firewall. Figure 9-6 shows a 7-region firewall setting that creates 16 regions (twice the number of regions created than originally available).

Figure 9-6. L3 Interconnect Region Overlay and Priority Level Overview



The address range covered by the regions is defined in the START_REGION_i and END_REGION_i registers. The boundary checks are done on a minimum size of 4-KiB pages; thus, bits [11:0] of those 32-bit registers are not checked.

The address space size of the slave NIUs ([bits [31:12]) depends on the size of the slave NIU to protect (that is, if a memory is only 48KiB, then the size is defined through bits [16:12] of the slave NIU start and end address registers of the firewall region (START_REGION_i[16:12] and END_REGION_i[16:12]).

To enable/disable regions the END_REGION_i[0] END_REGION_i_ENABLE bit should be set/cleared.

There is only one multiport firewall in this device. This is the TPTC_FW firewall which has two ports. The firewall configuration applies to all ports but there is one set of error logging registers per port. The ERROR_LOG_0 register logs errors for traffic going to the TPTC1_TARG and the ERROR_LOG_1 register logs errors for traffic going to the TPTC2_TARG.

The EMIF firewall (EMIF_FW) has 4 GiB address space and protects both the SDRAM and the EMIF configuration registers. Although the SDRAM is accessible at address space 0x8000_0000 through 0xFFFF_FFFF (total of 2GiB) the EMIF_FW sees that range in its address space at addresses 0x0000_0000 through 0x7FFF_FFFF. This means, to protect the SDRAM the start address in the START_REGION_i register should be 0x0000_0000, and not 0x8000_0000. The EMIF configuration registers are accessible through system base address 0x4C00_0000. But in the EMIF_FW address space that address is mapped to 0x8000_0000. In other words, EMIF_FW sees the EMIF configuration registers at 0x8000_0000. This means, to protect the EMIF configuration registers the start address in the START_REGION_i register should be 0x8000_0000, and not 0x4C00_0000.

9.2.3.7.3.2 L3_MAIN Firewall Registers Overview

Table 9-16 and Table 9-17 list the L3_MAIN firewall permission-setting registers.

Table 9-16. L3_MAIN Firewall Read/Write Permission-Setting Register

Register Name	Bits	Field Name	4-bit ConnID Value (hex) (see Table 9-10)	Field Modifiability
MRM_PERMISSION_REGION_HIGH_j	31	W15	ConnID = F write permission	RW
	30	R15	ConnID = F read permission	RW
	29	W14	ConnID = E write permission	RW
	28	R14	ConnID = E read permission	RW
	27	W13	ConnID = D write permission	RW
	26	R13	ConnID = D read permission	RW
	25	W12	ConnID = C write permission	RW
	24	R12	ConnID = C read permission	RW
	23	W11	ConnID = B write permission	RW
	22	R11	ConnID = B read permission	RW
	21	W10	ConnID = A write permission	RW
	20	R10	ConnID = A read permission	RW
	19	W9	ConnID = 9 write permission	RW
	18	R9	ConnID = 9 read permission	RW
	17	W8	ConnID = 8 write permission	RW
	16	R8	ConnID = 8 read permission	RW
	15	W7	ConnID = 7 write permission	RW
	14	R7	ConnID = 7 read permission	RW
	13	W6	ConnID = 6 write permission	RW
	12	R6	ConnID = 6 read permission	RW
	11	W5	ConnID = 5 write permission	RW
	10	R5	ConnID = 5 read permission	RW
	9	W4	ConnID = 4 write permission	RW
	8	R4	ConnID = 4 read permission	RW
	7	W3	ConnID = 3 write permission	RW
	6	R3	ConnID = 3 read permission	RW
	5	W2	ConnID = 2 write permission	RW
	4	R2	ConnID = 2 read permission	RW
	3	W1	ConnID = 1 write permission	RW
	2	R1	ConnID = 1 read permission	RW
	1	W0	ConnID = 0 write permission	RW
	0	R0	ConnID = 0 read permission	RW

Table 9-17. L3_MAIN Firewall Permission-Setting Register

Register name	Type of Permission	Bits	Field Name	Description	Field Modifiability
MRM_PERMISSION_REGION_LOW_j	Reserved	31:16	Reserved		
	Debug	15	PUB_PRV_DEBUG	PUBLIC PRIVILEGE DOMAIN DEBUG ALLOWED	RW
		14	PUB_USR_DEBUG	PUBLIC USER DOMAIN DEBUG ALLOWED	RW
	Reserved	13:12	Reserved		
	Access	11	PUB_PRV_READ	PUBLIC PRIVILEGE READ ACCESS ALLOWED	RW
		10	PUB_PRV_WRITE	PUBLIC PRIVILEGE WRITE ACCESS ALLOWED	RW
		9	PUB_PRV_EXE	PUBLIC PRIVILEGE EXE ACCESS ALLOWED	RW
		8	PUB_USR_READ	PUBLIC USER READ ACCESS ALLOWED	RW
		7	PUB_USR_WRITE	PUBLIC USER WRITE ACCESS ALLOWED	RW
		6	PUB_USR_EXE	PUBLIC USER EXE ACCESS ALLOWED	RW

9.2.3.7.3.3 Protection Mechanism per Region Examples

The access permission of each region is configurable and defined through the MRM_PERMISSION_REGION_HIGH_j and MRM_PERMISSION_REGION_LOW_j registers (see Section 9.2.3.7.3.2, *L3_MAIN Firewall Registers Overview*).

Master NIU permissions:

1. To give read access to the master NIU with 4-bit ConnID = n(1), set the MRM_PERMISSION_REGION_HIGH_j[n × 2] Rn bit.
2. To give write access to the master NIU with 4-bit ConnID = n(1), set the MRM_PERMISSION_REGION_HIGH_j[n × 2 + 1] Wn bit.

(1) - n should be first converted from hex to decimal value

Debug permissions:

1. To give privilege debug access, set the MRM_PERMISSION_REGION_LOW_j[15] PUB_PRV_DEBUG bit.
2. To give user debug access, set the MRM_PERMISSION_REGION_LOW_j[14] PUB_USR_DEBUG bit.

User, read, write, and executable permissions:

1. To give privileged read access, set the MRM_PERMISSION_REGION_LOW_j[11] PUB_PRV_READ bit.
2. To give privileged write access, set the MRM_PERMISSION_REGION_LOW_j[10] PUB_PRV_WRITE bit.
3. To give privileged executable access, set the MRM_PERMISSION_REGION_LOW_j[9] PUB_PRV_EXE bit.
4. To give user read access, set the MRM_PERMISSION_REGION_LOW_j[8] PUB_USR_READ bit.
5. To give user write access, set the MRM_PERMISSION_REGION_LOW_j[7] PUB_USR_WRITE bit.
6. To give user executable access, set the MRM_PERMISSION_REGION_LOW_j[6] PUB_USR_EXE bit.

Example: To provide debug write privilege access to the master NIU with 4-bit ConnID = 0x8, set to 0x1 the following bits:

- MRM_PERMISSION_REGION_HIGH_j[17] W8
- MRM_PERMISSION_REGION_LOW_j[15] PUB_PRV_DEBUG

9.2.3.7.3.4 L3_MAIN Firewall Error Logging

If a protection violation error is detected, the following signals are generated:

- An in-band error (SRESP = ERROR) is generated to the master NIU of the access.
- An out-band error is sent to the control module
- An interrupt is generated to the IRQ_CROSSBAR.

The L3_MAIN interconnect does not differentiate errors generated by firewalls from all other supported types of errors.

An in-band error is generated by modules each time an access is not allowed. When an in-band error is sent back into the transaction it is seen as an external prefetch or data abort by the initiator, depending on whether the transaction was an instruction fetch or a data access.

Information about in-band errors is logged into two registers:

- ERROR_LOG_k: Logs the information about the start/end address of the hit region and the qualifiers of the transaction
- LOGICAL_ADDR_ERRLOG_k[31:12]: Logs the address of the failed access

Table 9-18 lists the L3_MAIN firewall error-logging registers.

Table 9-18. L3 Firewall Error-Logging Registers

Register Name	Register Field Name	Field Modifiability	Parameter Comments
ERROR_LOG_k	RESERVED[31:24]	Read only	Reads return 0s.
	BLK_BURST_VIOLATION[23]	Read/write	Read 0x1: 2D burst not allowed or exceeds allowed size. Write to clear the ERROR_LOG_k and LOGICAL_ADDR_ERRLOG_k registers.
	RESERVED[22]	Read only	Read return 0s.
	REGION_START_ERRLOG[21:17]	Read/write	Read: Wrong access hit this region number. Write to clear the ERROR_LOG_k and LOGICAL_ADDR_ERRLOG_k registers.
	REGION_END_ERRLOG[16:12]	Read/write	Read: Wrong access hit this region number. Write to clear the ERROR_LOG_k and LOGICAL_ADDR_ERRLOG_k registers.
	REQINFO_ERRLOG[11:0]	Read/write	Mapping of the error according to the reqinfo vector: ConnID [3:0] MCMD [0] [3] MReqDebug [1] MReqSupervisor [0] MReqType

L3 firewall errors can be cleared by writing to the ERROR_LOG_k register in the firewall that recorded the error. Clearing the ERROR_LOG_k register deasserts the corresponding error if it exists.

The L3 firewall register ERROR_LOG_k must be cleared before clearing the CTRL_CORE_SEC_ERR_STATUS_FUNC_i (i = 1, 2 and 3) and CTRL_CORE_SEC_ERR_STATUS_DEBUG_i (i = 1, 2 and 3) registers in the control module.

When a protection violation occurs, an interrupt is sent to the IRQ_CROSSBAR and depending on the functional mode an error is logged in:

- the CTRL_CORE_SEC_ERR_STATUS_FUNC_i registers when the device is in application mode
- the CTRL_CORE_SEC_ERR_STATUS_DEBUG_i registers when the device is in debug mode

For more information, see Chapter 13, *Control Module*.

9.2.3.8 L3_MAIN Interconnect Error Handling

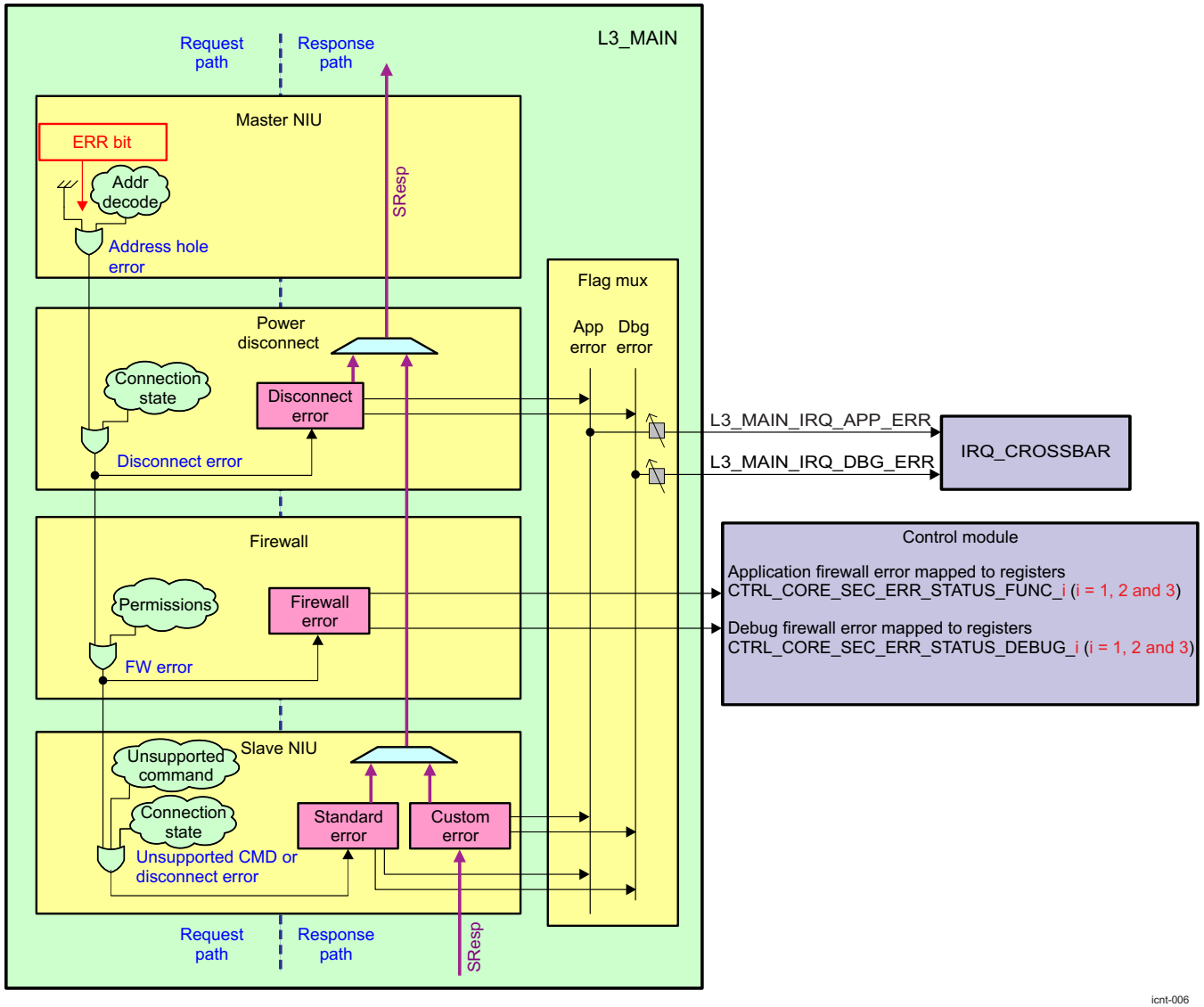
Error logging is enabled in the L3_MAIN interconnect. The three major types of errors are:

- Slave NIU errors
- Firewall errors (see Section 9.2.3.7.3.4, *L3_MAIN Firewall Error Logging*)
- Flag mux errors

9.2.3.8.1 Global Error-Routing Scheme

Figure 9-7 shows the L3_MAIN global error-routing scheme.

Figure 9-7. L3_MAIN Global Error-Routing Scheme



9.2.3.8.2 Slave NIU Error Logging

Error logging is implemented only at slave NIUs. Because the interconnect does not support master NIU error logging, an erroneous packet must be created and sent to one of the slave NIUs. The slave NIU that receives an erroneous packet is predictable but can change per master (see Table 9-19).

Table 9-19. L3_MAIN Connectivity and Holes Error Routing

Master	Connectivity and Hole Errors Logged Into Slave NIUs
All initiators	GPMC_TARG

The slave NIU can:

- be configured to report standard errors (errors generated within the interconnect):
  - Firewall error (protection violation):
    this error indicates that a request was rejected by a firewall and is reported to the control module. For more information, see Section 9.2.3.7.3.4, *L3_MAIN Firewall Error Logging.*
  - Address hole:
    this error reports an unknown address for a request. The address map is local to each master NIU; therefore, an address hole error is reported each time a master NIU requests an access to a slave NIU to which it is not logically connected, even if this address exists in the global L3_MAIN address map. This error is detected only once per burst.
  - Unsupported commands:
    this error reports that the master NIU sent a command that cannot be processed, because the slave NIU cannot accept it and no conversion to another command is possible. This error is detected only once per burst.
- be configured to report custom errors: basically, when the slave answer is SResp = ERR
- be configured to report severity level, for standard error and custom errors:
  - None: error logging for this type of error is disabled.
  - Error: error is logged for this type of error.
  - Fault: error is logged and interrupt is generated for this type of error.
- generate interrupt on two IRQ lines depending on the MReqDebug qualifier:
  - Application error - on L3_MAIN_IRQ_APP_ERR line
  - Debug error - on L3_MAIN_IRQ_DBG_ERR line

  By default, all slave NIUs are configured with standard and custom error levels set to Fault. The errors are reported on the flag muxes (see Figure 9-7), depending on the access type, application or debug. For more information, see Section 9.2.3.5, *Flag Muxing*.

  The slave NIU power-disconnect component also has error logging enabled, because in this case the slave NIU is in a clock domain that is switched off and therefore cannot catch the error. By nature, this component can generate only standard errors. By default, it is configured with the error level set to Fault.

  Wake up on demand: If an error packet reaches a slave NIU that is set with MDiscBehave = 1 (wake up on demand), then the active signal is asserted and L3 processes the error generation when the slave is awake. Although this is inefficient, it simplifies NIU implementation and should not be a problem because errors are supposed to occur only during software debug.

9.2.3.8.3 Severity Level of Standard and Custom Errors

The slave NIU registers are important for error logging.

- The L3_TARG_STDERRLOG_SVRTSTDLVL register shows the severity level for standard errors. According to the severity level, error logging is disabled, enabled with level ERROR, or enabled with level ERROR and flag FAULT.
- The L3_TARG_STDERRLOG_SVRTCUSTOMLVL register shows the severity level for custom errors.
- The L3_TARG_STDERRLOG_MAIN register (the main register for error-logging management) shows the validity of the logged information, standard or custom.
- The L3_TARG_STDERRLOG_HDR register stores packets in case of a standard error.
- The L3_TARG_STDERRLOG_MSTADDR register returns the MSTADDR field of the logged packet.
- The L3_TARG_STDERRLOG_SLVADDR register returns the SLVADDR field of the stored packet.
- The L3_TARG_STDERRLOG_INFO register saves the information field of the logged packet.

9.2.3.8.4 Example for Decoding Standard/Custom Errors Logged in L3_MAIN

Following is a procedure that must be followed for identifying Standard/Custom errors logged in L3_MAIN:

- Read registers L3_HOST_STDERRLOG_MAIN and L3_TARG_STDERRLOG_MAIN:
  - bit[0]STDERRLOG_MAIN_ERRLOGVLD = 0 -> No error
  - bit[0]STDERRLOG_MAIN_ERRLOGVLD = 1 -> Error
  - Error type is visible through:
    - bit[1]STDERRLOG_MAIN_ERRTYPE = 0 -> Standard Error (FW error, Address Hole, Unsupported command)
    - bit[1]STDERRLOG_MAIN_ERRTYPE = 1 -> Custom Error
- Standard Error:
  - Read L3_TARG_STDERRLOG_HDR[3:0] STDERRLOG_HDR_OPCODE
  - Read L3_TARG_STDERRLOG_MSTADDR[7:0] STDERRLOG_MSTADDR - > 8-bit NTTP master address used to distinguish the different initiators (see Table 9-10). That master address is also referred to as ConnID or MConnID.
  - Read L3_TARG_STDERRLOG_SLVADDR[4:0] STDERRLOG_SLVADDR -> NTTP Slave Address of the logged packet
  - Read L3_TARG_STDERRLOG_INFO[7:0]STDERRLOG_INFO -> see , *Terminology* for details.

The procedure for Custom error is the same, but the *_STDERRLOG_CUSTOMINFO_* registers should be read.

9.2.4 L3_MAIN Interconnect Programming Guide

9.2.4.1 L3 _MAIN Interconnect Low-Level Programming Models

This section describes the low-level hardware programming sequences for configuring and using the L3_MAIN interconnect module.

9.2.4.1.1 Global Initialization

9.2.4.1.1.1 Global Initialization of Surrounding Modules

This section identifies the requirements for initializing the surrounding modules when the L3_MAIN interconnect module is to be used for the first time after a device reset. The initialization of surrounding modules is based on the integration of the L3_MAIN interconnect. For more information, see Section 9.2.2, *L3_MAIN Interconnect Integration* .

Table 9-20. Global Initialization of Surrounding Modules

Surrounding Modules	Comments
PRCM	For information about the configuration of the PRCM module, see Chapter 3, *Power, Reset, and Clock Management*.
Control module	For information about the configuration of the control module, see Chapter 13, *Control Module*.
IRQ_CROSSBAR	IRQ_CROSSBAR configuration must be done to allow module IRQs to be mapped to certain device INTC line. For more information see Section 13.4.6.4, *IRQ_CROSSBAR Module Functional Description*, in Chapter 13, *Control Module*
Device INTCs	Device INTCs must be configured to enable the interrupt request generation. For more information see Chapter 12 *Interrupt Controllers*

9.2.4.2 Operational Modes Configuration

9.2.4.2.1 L3_MAIN Interconnect Error Analysis Mode

9.2.4.2.1.1 Main Sequence: L3_MAIN Interconnect Error Analysis Mode

The information required to analyze an error source is logged in several registers. The number of registers to access depends on the error source.

Figure 9-8 shows the software sequence required in most cases.

Figure 9-8. Typical Error Analysis Sequence

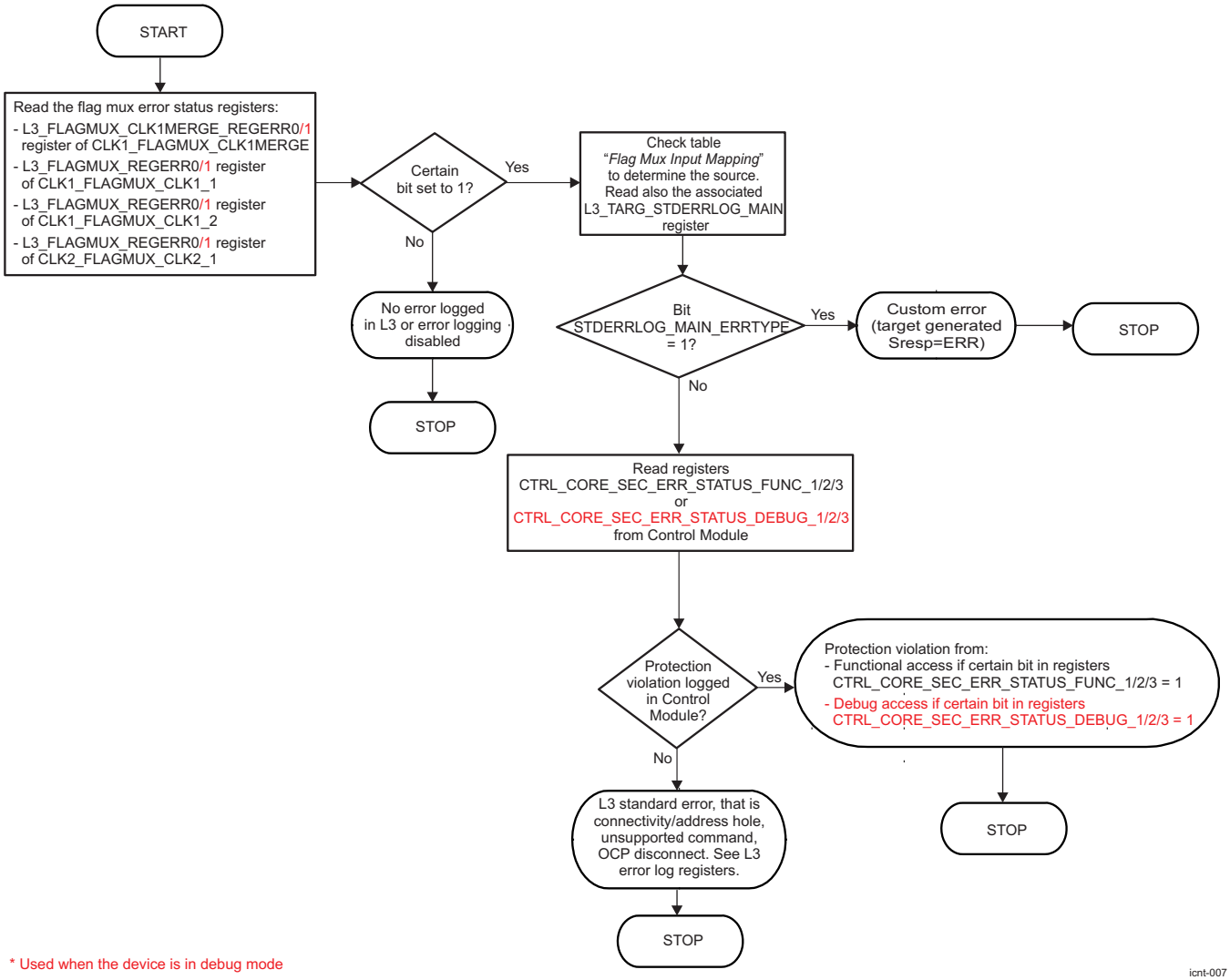


Table 9-21 lists the subprocess call summary for error analysis mode in the main sequence.

Table 9-21. Subprocess Call Summary for Main Sequence – Error Analysis Mode

Subprocess	Cross-Reference
L3 interconnect error analysis	See Section 9.2.4.2.1, *L3_MAIN Interconnect Error Analysis Mode.*
L3_MAIN interconnect protection violation error identification	See Section 9.2.4.2.1.1.2, *Subsequence: L3_MAIN Interconnect Protection Violation Error Identification.*

Table 9-21. Subprocess Call Summary for Main Sequence – Error Analysis Mode (continued)

Subprocess	Cross-Reference
L3_MAIN interconnect unsupported command/address hole error identification	See Section 9.2.4.2.1.1.3, *Subsequence: L3_MAIN Interconnect Standard Error Identification.*
L3_MAIN interconnect reset FLAGMUX and module	See Section 9.2.4.2.1.1.4, *Subsequence: L3_MAIN Interconnect FLAGMUX Configuration.*

9.2.4.2.1.1.1 Subsequence: L3_MAIN Custom Error Identification

The procedure listed in Table 9-22 describes custom error identification.

Table 9-22. Custom Error Identification

Step	Register/Bit Field/Programming Model	Value
IF: Is custom error detected?	L3_TARG_STDERRLOG_MAIN[1] STDERRLOG_MAIN_ERRTYPE	=0x1
Read information field of the response packet.	L3_TARG_STDERRLOG_CUSTOMINFO_INFO[7:0] STDERRLOG_CUSTOMINFO_INFO	xxx
Read the address of the initiator that caused error.	L3_TARG_STDERRLOG_CUSTOMINFO_MSTADDR[7:0] STDERRLOG_CUSTOMINFO_MSTADDR	xxx
Read the type of operation (read/write).	L3_TARG_STDERRLOG_CUSTOMINFO_OPCODE[1:0] STDERRLOG_CUSTOMINFO_OPCODE	xxx
ENDIF		

9.2.4.2.1.1.2 Subsequence: L3_MAIN Interconnect Protection Violation Error Identification

The procedure listed in Table 9-23 describes protection violation error identification and where it is logged in the control module registers. Two types of errors are logged: application errors and debug errors.

Table 9-23. L3_MAIN Protection Violation Error Identification

Step	Register/Bit Field/Programming Model	Value
Read the burst violation.	ERROR_LOG_k[23] BLK_BURST_VIOLATION	xxx
Read the initiator ConnID [3:0].	ERROR_LOG_k[11:8]	xxx
Read the command (MCMD [0]) that caused the error.	ERROR_LOG_k[7]	xxx
Read the address of the request that caused the error.	LOGICAL_ADDR_ERRLOG_k[31:0] SLVOFS_LOGICAL	xxx
IF: Is it an application error?	L3_FLAGMUX_REGERR0 [REGERROR0]	0x0
Read the status bits to see which module firewall has worked.	CTRL_CORE_SEC_ERR_STATUS_FUNC_i	xxx
Clear the status bits.	CTRL_CORE_SEC_ERR_STATUS_FUNC_i	xxx
Clear the status bit.	L3_TARG_STDERRLOG_MAIN [31] STDERRLOG_MAIN_CLRLOG	0x0
ELSE IF	L3_FLAGMUX_REGERR1[7:0] REGERROR1	= 0x1
Read the status bits to see the module.	CTRL_CORE_SEC_ERR_STATUS_DEBUG_i	xxx
Clear the status bits.	CTRL_CORE_SEC_ERR_STATUS_DEBUG_i	xxx
Clear the status bit.	L3_TARG_STDERRLOG_MAIN[31] STDERRLOG_MAIN_CLRLOG	0x0
ENDIF		
Clear the burst violation.	ERROR_LOG_k[23] BLK_BURST_VIOLATION	0x0
Clear the error status.	ERROR_LOG_k[21:17] REGION_START_ERRLOG	0x00

9.2.4.2.1.1.3 Subsequence: L3_MAIN Interconnect Standard Error Identification

The procedure listed in Table 9-24 describes the identification of standard errors inside the L3_MAIN interconnect. The standard errors are: unsupported command, address hole, and disconnect.

Table 9-24. L3_MAIN Standard Error Identification

Step	Register/Bit Field/Programming Model	Value
IF: Is an error detected?	L3_TARG_STDERRLOG_MAIN[18] STDERRLOG_MAIN_ERRCNT	= 0x1
Read the corresponding flag.	L3_FLAGMUX_REGERR0[REGERROR0]	xxx
Read the corresponding flag.	L3_FLAGMUX_REGERR1[REGERROR1]	xxx
Localize the slave NIU that generated the error.	See Table 9-11.	
ELSE		
Clear the error log.	L3_TARG_STDERRLOG_MAIN[31] STDERRLOG_MAIN_CLRLOG	0x0
Clear the severity error status.	L3_TARG_STDERRLOG_SVRTSTDLVL[1:0] STDERRLOG_SVRTSTDLVL_0	0x2
ENDIF		

9.2.4.2.1.1.4 Subsequence: L3_MAIN Interconnect FLAGMUX Configuration

The procedures listed in Table 9-25 and Table 9-26 give information about the configuration of FLAGMUX masks.

Table 9-25. FLAGMUX Configuration

Step	Register/Bit Field/Programming Model	Value
Set the FLAGMUX masks to mask an event.	L3_FLAGMUX_MASK0[MASK0] L3_FLAGMUX_MASK1[MASK1]	xxx
Read the REGERR bits to see if an error is recorded.	L3_FLAGMUX_REGERR0[REGERR0] L3_FLAGMUX_REGERR1[REGERR1]	xxx
Clear the slave NIU error log and the FLAGMUX error.	L3_TARG_STDERRLOG_MAIN[31] STDERRLOG_SVRTSTDLVL_0	0x1

Table 9-26. FLAGMUX_CLK1_TIMEOUT1 and FLAGMUX_CLK1_TIMEOUT2 Configuration

Step	Register/Bit Field/Programming Model	Value
Enable time-out for target.	L3_FLAGMUX_TIMEOUTx_MASK0[MASK0]	xxx
Read the REGERR bits to see if an error is recorded.	L3_FLAGMUX_TIMEOUTx_REGERR0[REGERR0]	xxx

Table 9-27. FLAGMUX_CLK2_TIMEOUT Configuration

Step	Register/Bit Field/Programming Model	Value
Enable time-out for target.	L3_FLAGMUX_TIMEOUT_MASK0[1:0] MASK0	xxx
Read the REGERR bits to see if an error is recorded.	L3_FLAGMUX_TIMEOUT_REGERR0[1:0] REGERR0	xxx

9.2.5 L3_MAIN Interconnect Register Manual

9.2.5.1 L3_MAIN Register Group Summary

The registers in the L3 interconnect are divided into the following groups:

- Firewall registers (see Table 9-29)
- HOST registers (see Table 9-52)
- TARG registers (see Table 9-88)
- Flag Muxing registers (see Section 9.2.5.1.4 through Section 9.2.5.1.6)
- Bandwidth Regulator registers (see Table 9-186)
- Bandwidth Limiter registers (see Table 9-202)
- STATCOLL registers (see Table 9-221)

9.2.5.1.1 L3_MAIN Firewall Registers Summary and Description

NOTE: ISS, CRC and TESOC are not supported on the DRA78x family of devices.

Table 9-28. L3_MAIN Firewall Instance Summary

Module Name	Base Address	Size
DEBUGSS_CT_TBR_FW	0x4A22 4000	4KiB
DSP1_SDMA_FW	0x4A17 1000	4KiB
DSP2_SDMA_FW	0x4A17 3000	4KiB
DSS_FW	0x4A21 C000	4KiB
EVE_FW	0x4A15 1000	4KiB
GPMC_FW	0x4A21 0000	4KiB
IPU_FW	0x4A15 B000	4KiB
L3_INSTR_FW	0x4A22 6000	4KiB
OCMC_RAM_FW	0x4A21 2000	4KiB
EMIF_FW	0x4A20 C000	4KiB
QSPI_FW	0x4A17 9000	4KiB
EDMA_TPCC_FW	0x4A16 1000	4KiB
TPTC_FW	0x4A16 3000	4KiB
CRC_FW	0x4A24 4000	4KiB
TESOC_FW	0x4A24 0000	4KiB
ISS_FW	0x4A23 E000	4KiB
MCASP1_FW	0x4A16 7000	4KiB

9.2.5.1.1.1 L3_MAIN Firewall Registers Summary

Table 9-29. L3_MAIN Firewall Registers Summary

Register Name	Type	Register Width (Bits)	Address Offset	DEBUGSS_CT_TBR_FW L3_MAIN Physical Address	DSP1_SDMA_FW L3_MAIN Physical Address	DSP2_SDMA_FW L3_MAIN Physical Address
ERROR_LOG_k [1]	RW	32	0x0000 0000+(0x10*k)	0x4A22 4000 + (0x10*k)	0x4A17 1000 + (0x10*k)	0x4A17 3000 + (0x10*k)
LOGICAL_ADDR_ERRLOG_k [1]	RO	32	0x0000 0004+(0x10*k)	0x4A22 4004 + (0x10*k)	0x4A17 1004 + (0x10*k)	0x4A17 3004 + (0x10*k)

(1) k = 0 for DEBUGSS_CT_TBR_FW
k = 0 for DSP1_SDMA_FW
k = 0 for DSP2_SDMA_FW

Table 9-29. L3_MAIN Firewall Registers Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	DEBUGSS_CT_TBR_FW L3_MAIN Physical Address	DSP1_SDMA_FW L3_MAIN Physical Address	DSP2_SDMA_FW L3_MAIN Physical Address
REGUPDATE_CONTROL	RW	32	0x0000 0040	0x4A22 4040	0x4A17 1040	0x4A17 3040
START_REGION_i (2)	RW	32	0x0000 0080+(0x10*i)	0x4A22 4080 + (0x10*i)	-	-
END_REGION_i (2)	RW	32	0x0000 0084+(0x10*i)	0x4A22 4084 + (0x10*i)	-	-
MRM_PERMISSION_REGION_HIGH_j (3)	RW	32	0x0000 008C+(0x10*j)	0x4A22 408C + (0x10*j)	0x4A17 108C + (0x10*j)	0x4A17 308C + (0x10*j)
MRM_PERMISSION_REGION_LOW_j (3)	RW	32	0x0000 0088+(0x10*j)	0x4A22 4088 + (0x10*j)	0x4A17 1088 + (0x10*j)	0x4A17 3088 + (0x10*j)

(2) i = 1 for DEBUGSS_CT_TBR_FW
(3) j = 0 to 1 for DEBUGSS_CT_TBR_FW
j = 0 for DSP1_SDMA_FW
j = 0 for DSP2_SDMA_FW

Table 9-30. L3_MAIN Firewall Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	DSS_FW L3_MAIN Physical Address	EVE_FW L3_MAIN Physical Address	GPMC_FW L3_MAIN Physical Address
ERROR_LOG_k (1)	RW	32	0x0000 0000+(0x10*k)	0x4A21 C000 + (0x10*k)	0x4A15 1000 + (0x10*k)	0x4A21 0000 + (0x10*k)
LOGICAL_ADDR_ERRLOG_k (1)	RO	32	0x0000 0004+(0x10*k)	0x4A21 C004 + (0x10*k)	0x4A15 1004 + (0x10*k)	0x4A21 0004 + (0x10*k)
REGUPDATE_CONTROL	RW	32	0x0000 0040	0x4A21 C040	0x4A15 1040	0x4A21 0040
START_REGION_i (2)	RW	32	0x0000 0080+(0x10*i)	0x4A21 C080 + (0x10*i)	-	0x4A21 0080 + (0x10*i)
END_REGION_i (2)	RW	32	0x0000 0084+(0x10*i)	0x4A21 C084 + (0x10*i)	-	0x4A21 0084 + (0x10*i)
MRM_PERMISSION_REGION_HIGH_j (3)	RW	32	0x0000 008C+(0x10*j)	0x4A21 C08C + (0x10*j)	0x4A15 108C + (0x10*j)	0x4A21 008C + (0x10*j)
MRM_PERMISSION_REGION_LOW_j (3)	RW	32	0x0000 0088+(0x10*j)	0x4A21 C088 + (0x10*j)	0x4A15 1088 + (0x10*j)	0x4A21 0088 + (0x10*j)

(1) k = 0 for DSS_FW
k = 0 for EVE_FW
k = 0 for GPMC_FW
(2) i = 1 to 7 for DSS_FW
i = 1 to 7 for GPMC_FW
(3) j = 0 to 7 for DSS_FW
j = 0 for EVE_FW
j = 0 to 7 for GPMC_FW

Table 9-31. L3_MAIN Firewall Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	IPU_FW L3_MAIN Physical Address	L3_INSTR_FW L3_MAIN Physical Address	OCMC_RAM_FW L3_MAIN Physical Address
ERROR_LOG_k (1)	RW	32	0x0000 0000+(0x10*k)	0x4A15 B000 + (0x10*k)	0x4A22 6000 + (0x10*k)	0x4A21 2000 + (0x10*k)
LOGICAL_ADDR_ERRLOG_k (1)	RO	32	0x0000 0004+(0x10*k)	0x4A15 B004 + (0x10*k)	0x4A22 6004 + (0x10*k)	0x4A21 2004 + (0x10*k)
REGUPDATE_CONTROL	RW	32	0x0000 0040	0x4A15 B040	0x4A22 6040	0x4A21 2040

(1) k = 0 for IPU_FW
k = 0 for L3_INSTR_FW
k = 0 for OCMC_RAM_FW

Table 9-31. L3_MAIN Firewall Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	IPU_FW L3_MAIN Physical Address	L3_INSTR_FW L3_MAIN Physical Address	OCMC_RAM_FW L3_MAIN Physical Address
START_REGION_i (2)	RW	32	0x0000 0080+(0x10*i)	0x4A15 B080 + (0x10*i)	-	0x4A21 2080 + (0x10*i)
END_REGION_i (2)	RW	32	0x0000 0084+(0x10*i)	0x4A15 B084 + (0x10*i)	-	0x4A21 2084 + (0x10*i)
MRM_PERMISSION_REGION_HIGH_j (3)	RW	32	0x0000 008C+(0x10*j)	0x4A15 B08C + (0x10*j)	0x4A22 608C + (0x10*j)	0x4A21 208C + (0x10*j)
MRM_PERMISSION_REGION_LOW_j (3)	RW	32	0x0000 0088+(0x10*j)	0x4A15 B088 + (0x10*j)	0x4A22 6088 + (0x10*j)	0x4A21 2088 + (0x10*j)

(2) i = 1 to 3 for IPU_FW
i =1 to 23 for OCMC_RAM_FW
(3) j = 0 to 3 for IPU_FW
j = 0 for L3_INSTR_FW
j = 0 to 23 for OCMC_RAM_FW

Table 9-32. L3_MAIN Firewall Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	EMIF_FW L3_MAIN Physical Address	QSPI_FW L3_MAIN Physical Address	EDMA_TPCC_FW L3_MAIN Physical Address
ERROR_LOG_k (1)	RW	32	0x0000 0000+(0x10*k)	0x4A20 C000 + (0x10*k)	0x4A17 9000 + (0x10*k)	0x4A16 1000 + (0x10*k)
LOGICAL_ADDR_ERRLOG_k (1)	RO	32	0x0000 0004+(0x10*k)	0x4A20 C004 + (0x10*k)	0x4A17 9004 + (0x10*k)	0x4A16 1004 + (0x10*k)
REGUPDATE_CONTROL	RW	32	0x0000 0040	0x4A20 C040	0x4A17 9040	0x4A16 1040
START_REGION_i (2)	RW	32	0x0000 0080+(0x10*i)	0x4A20 C080 + (0x10*i)	0x4A17 9080 + (0x10*i)	-
END_REGION_i (2)	RW	32	0x0000 0084+(0x10*i)	0x4A20 C084 + (0x10*i)	0x4A17 9084 + (0x10*i)	-
MRM_PERMISSION_REGION_HIGH_j (3)	RW	32	0x0000 008C+(0x10*j)	0x4A20 C08C + (0x10*j)	0x4A17 908C + (0x10*j)	0x4A16 108C + (0x10*j)
MRM_PERMISSION_REGION_LOW_j (3)	RW	32	0x0000 0088+(0x10*j)	0x4A20 C088 + (0x10*j)	0x4A17 9088 + (0x10*j)	0x4A16 1088 + (0x10*j)

(1) k = 0 for EMIF_FW
k = 0 for QSPI_FW
k = 0 for EDMA_TPCC_FW
(2) i = 1 to 23 for EMIF_FW
i = 1 to 3 for QSPI_FW
(3) j = 0 to 23 for EMIF_FW
j = 0 to 3 for QSPI_FW
j = 0 for EDMA_TPCC_FW

Table 9-33. L3_MAIN Firewall Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	TPTC_FW L3_MAIN Physical Address	CRC_FW L3_MAIN Physical Address	TESOC_FW L3_MAIN Physical Address
ERROR_LOG_k (1)	RW	32	0x0000 0000+(0x10*k)	0x4A16 3000+(0x10*k)	0x4A24 4000+(0x10*k)	0x4A24 0000+(0x10*k)
LOGICAL_ADDR_ERRLOG_k (1)	RO	32	0x0000 0004+(0x10*k)	0x4A16 3004+(0x10*k)	0x4A24 4004+(0x10*k)	0x4A24 0004+(0x10*k)
REGUPDATE_CONTROL	RW	32	0x0000 0040	0x4A16 3040	0x4A24 4040	0x4A24 0040

(1) k = 0 to 1 for TPTC_FW
k = 0 for CRC_FW
k = 0 for TESOC_FW

Table 9-33. L3_MAIN Firewall Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	TPTC_FW L3_MAIN Physical Address	CRC_FW L3_MAIN Physical Address	TESOC_FW L3_MAIN Physical Address
START_REGION_i [2]	RW	32	0x0000 0080+(0x10*i)	0x4A16 3080+(0x10*i)	0x4A24 4080+(0x10*i)	0x4A24 0080+(0x10*i)
END_REGION_i [2]	RW	32	0x0000 0084+(0x10*i)	0x4A16 3084+(0x10*i)	0x4A24 4084+(0x10*i)	0x4A24 0084+(0x10*i)
MRM_PERMISSION_REGION_HIGH_j [3]	RW	32	0x0000 008C+(0x10*j)	0x4A16 308C+(0x10*j)	0x4A24 408C+(0x10*j)	0x4A24 008C+(0x10*j)
MRM_PERMISSION_REGION_LOW_j [3]	RW	32	0x0000 0088+(0x10*j)	0x4A16 3088+(0x10*j)	0x4A24 4088+(0x10*j)	0x4A24 0088+(0x10*j)

(2) i = 1 for TPTC_FW
i = 1 to 3 for CRC_FW
i = 1 to 3 for TESOC_FW

(3) j = 0 to 1 for TPTC_FW
j = 0 to 3 for CRC_FW
j = 0 to 3 for TESOC_FW

Table 9-34. L3_MAIN Firewall Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	ISS_FW L3_MAIN Physical Address	MCASP1_FW L3_MAIN Physical Address
ERROR_LOG_k [1]	RW	32	0x0000 0000+(0x10*k)	0x4A23 E000+(0x10*k)	0x4A16 7000+(0x10*k)
LOGICAL_ADDR_ERRLOG_k [1]	RO	32	0x0000 0004+(0x10*k)	0x4A23 E004+(0x10*k)	0x4A16 7004+(0x10*k)
REGUPDATE_CONTROL	RW	32	0x0000 0040	0x4A23 E040	0x4A16 7040
START_REGION_i [2]	RW	32	0x0000 0080+(0x10*i)	0x4A23 E080+(0x10*i)	-
END_REGION_i [2]	RW	32	0x0000 0084+(0x10*i)	0x4A23 E084+(0x10*i)	-
MRM_PERMISSION_REGION_HIGH_j [3]	RW	32	0x0000 008C+(0x10*j)	0x4A23 E08C+(0x10*j)	0x4A16 708C+(0x10*j)
MRM_PERMISSION_REGION_LOW_j [3]	RW	32	0x0000 0088+(0x10*j)	0x4A23 E088+(0x10*j)	0x4A16 7088+(0x10*j)

(1) k = 0 for ISS_FW
k = 0 for MCASP1_FW

(2) i = 1 to 3 for ISS_FW

(3) j = 0 to 3 for ISS_FW
j = 0 for MCASP1_FW

9.2.5.1.1.2 L3_MAIN Firewall Registers Description

NOTE: Hardware reset values can be modified by exported values from the control module at reset.

Table 9-35. ERROR_LOG_k

Address Offset	0x0000 0000+(0x10*k)	**Index**	See Table 9-29 to Table 9-34.
Physical Address	0x4A22 4000 + (0x10*k) 0x4A17 1000 + (0x10*k) 0x4A17 3000 + (0x10*k) 0x4A21 C000 + (0x10*k) 0x4A15 1000 + (0x10*k) 0x4A21 0000 + (0x10*k) 0x4A15 B000 + (0x10*k) 0x4A22 6000 + (0x10*k) 0x4A21 2000 + (0x10*k) 0x4A20 C000 + (0x10*k) 0x4A17 9000 + (0x10*k) 0x4A16 1000 + (0x10*k) 0x4A16 3000+(0x10*k) 0x4A24 4000+(0x10*k) 0x4A24 0000+(0x10*k) 0x4A23 E000+(0x10*k) 0x4A16 7000+(0x10*k)	**Instance**	DEBUGSS_CT_TBR_FW DSP1_SDMA_FW DSP2_SDMA_FW DSS_FW EVE_FW GPMC_FW IPU_FW L3_INSTR_FW OCMC_RAM_FW EMIF_FW QSPI_FW EDMA_TPCC_FW TPTC_FW CRC_FW TESOC_FW ISS_FW MCASP1_FW
Description	Error log register for port k		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								BLK_BURST_VIOLATION	RESERVED	REGION_START_ERRLOG						REGION_END_ERRLOG								REQINFO_ERRLOG							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED	Reads return 0s.	R	0x00
23	BLK_BURST_VIOLATION	Read 0x1: 2D burst not allowed or exceeding allowed size Write to clear ERROR_LOG_k and LOGICAL_ADDR_ERRLOG_k registers	RW	0
22	RESERVED	Reads return 0s.	R	0
21:17	REGION_START_ERRLOG	Read: Wrong access hit this region number Write to clear ERROR_LOG_k and LOGICAL_ADDR_ERRLOG_k registers	RW	0x00
16:12	REGION_END_ERRLOG	Read: Wrong access hit this region number Write to clear ERROR_LOG_k and LOGICAL_ADDR_ERRLOG_k registers	RW	0x00
11:0	REQINFO_ERRLOG	Read: Error in reqinfo vector mapped as follows: [11: 8] ConnID [3:0] [7] MCMD [0] [6:4] Reserved [3] MReqDebug [2] Reserved [1] MReqSupervisor [0] MReqType Write to clear ERROR_LOG_k and LOGICAL_ADDR_ERRLOG_k registers	RW	0x000

Table 9-36. Register Call Summary for Register ERROR_LOG_k

L3_MAIN Interconnect
- L3_MAIN Firewall Functionality: [0][1][2][3][4][5][6][7]
- L3_MAIN Interconnect Error Analysis Mode: [8][9][10][11][12]
- L3_MAIN Firewall Registers Summary and Description: [13][14][15][16][17][18][19][20][21][22]

Table 9-37. LOGICAL_ADDR_ERRLOG_k

Address Offset	0x0000 0004+(0x10*k)	**Index**	See Table 9-29 to Table 9-34.
Physical Address	0x4A22 4004 + (0x10*k) 0x4A17 1004 + (0x10*k) 0x4A17 3004 + (0x10*k) 0x4A21 C004 + (0x10*k) 0x4A15 1004 + (0x10*k) 0x4A21 0004 + (0x10*k) 0x4A15 B004 + (0x10*k) 0x4A22 6004 + (0x10*k) 0x4A21 2004 + (0x10*k) 0x4A20 C004 + (0x10*k) 0x4A17 9004 + (0x10*k) 0x4A16 1004 + (0x10*k) 0x4A16 3004+(0x10*k) 0x4A24 4004+(0x10*k) 0x4A24 0004+(0x10*k) 0x4A23 E004+(0x10*k) 0x4A16 7004+(0x10*k)	**Instance**	DEBUGSS_CT_TBR_FW DSP1_SDMA_FW DSP2_SDMA_FW DSS_FW EVE_FW GPMC_FW IPU_FW L3_INSTR_FW OCMC_RAM_FW EMIF_FW QSPI_FW EDMA_TPCC_FW TPTC_FW CRC_FW TESOC_FW ISS_FW MCASP1_FW
Description	Logical Physical Address Error log register for port k		
Type	RO		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SLVOFS_LOGICAL																															

Bits	Field Name	Description	Type	Reset
31:0	SLVOFS_LOGICAL	Address generated by the Initiator before being translated	R	0x00000

Table 9-38. Register Call Summary for Register LOGICAL_ADDR_ERRLOG_k

L3_MAIN Interconnect
- L3_MAIN Firewall Functionality: [0][1][2][3]
- L3_MAIN Interconnect Error Analysis Mode: [4]
- L3_MAIN Firewall Registers Summary and Description: [5][6][7][8][9][10][11][12][13][14]

Table 9-39. REGUPDATE_CONTROL

Address Offset	0x0000 0040		
Physical Address	0x4A22 4040	**Instance**	DEBUGSS_CT_TBR_FW
	0x4A17 1040		DSP1_SDMA_FW
	0x4A17 3040		DSP2_SDMA_FW
	0x4A21 C040		DSS_FW
	0x4A15 1040		EVE_FW
	0x4A21 0040		GPMC_FW
	0x4A15 B040		IPU_FW
	0x4A22 6040		L3_INSTR_FW
	0x4A21 2040		OCMC_RAM_FW
	0x4A20 C040		EMIF_FW
	0x4A17 9040		QSPI_FW
	0x4A16 1040		EDMA_TPCC_FW
	0x4A16 3040		TPTC_FW
	0x4A24 4040		CRC_FW
	0x4A24 0040		TESOC_FW
	0x4A23 E040		ISS_FW
	0x4A16 7040		MCASP1_FW
Description	Register update control register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												FW_ADDR_SPACE_MSB				RESERVED														FW_LOAD_REQ	BUSY_REQ

Bits	Field Name	Description	Type	Reset
31:20	RESERVED	Reads return 0s.	R	0x0000 0000
19:16	FW_ADDR_SPACE_MSB	Address space size	R	0x0
15:2	RESERVED	Reserved	R	0x0000 0000
1	FW_LOAD_REQ	Writing '1' to this bit causes the bit to self-clear and triggers the reload of L3 firewall default values. This bit will subsequently self-set when the reload procedure is complete. Writing '0' has no effect.	RW W1toClr	0x1
0	BUSY_REQ	Busy request 0x0: Allow transactions to reach the slave NIU (resume) 0x1: No transaction can reach the slave NIU (suspend)	RW	0x0

Table 9-40. Register Call Summary for Register REGUPDATE_CONTROL

L3_MAIN Interconnect

- L3_MAIN Firewall Reset: [0][1][2][3][4][5][6][7][8][9]
- L3_MAIN Firewall Registers Summary and Description: [10][11][12][13][14][15]

Table 9-41. START_REGION_i

Address Offset	0x0000 0080+(0x10*i)	**Index**	See Table 9-29 to Table 9-34.
Physical Address	0x4A22 4080 + (0x10*i) 0x4A21 C080 + (0x10*i) 0x4A21 0080 + (0x10*i) 0x4A15 B080 + (0x10*i) 0x4A21 2080 + (0x10*i) 0x4A20 C080 + (0x10*i) 0x4A17 9080 + (0x10*i) 0x4A16 3080+(0x10*i) 0x4A24 4080+(0x10*i) 0x4A24 0080+(0x10*i) 0x4A23 E080+(0x10*i)	**Instance**	DEBUGSS_CT_TBR_FW DSS_FW GPMC_FW IPU_FW OCMC_RAM_FW EMIF_FW QSPI_FW TPTC_FW CRC_FW TESOC_FW ISS_FW
Description	Start physical address of region i		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																				RESERVED											

Bits	Field Name	Description	Type	Reset
31:12	START_REGION_i	Physical target start address of firewall region i. The size of this bit field depends on target addressable space, the maximum is [31:12]. See Table 9-43. Each of the LSbits is assumed to be 0.The programmed address is included in the region i boundary.	RW	0x00000
11:0	RESERVED	Reads return 0s.	R	0x0000

Table 9-42. Register Call Summary for Register START_REGION_i

L3_MAIN Interconnect

- L3_MAIN Firewall Functionality: [0][1][2][3]
- L3_MAIN Firewall Registers Summary and Description: [4][5][6][7][8][9][10][11][12][13][14][15][16][17][18][19][20][21][22][23][24][25][26][27][28][29][30][31][32]

Table 9-43. Size of START_REGION_i[] START_REGION_i Bit Field

Firewall	Bit Field
DSS_FW	START_REGION_i[22:12] START_REGION_i
GPMC_FW	START_REGION_i[28:12] START_REGION_i
IPU_FW	START_REGION_i[22:12] START_REGION_i
OCMC_RAM_FW	START_REGION_i[22:12] START_REGION_i
EMIF_FW	START_REGION_i[30:12] START_REGION_i
TPTC_FW	START_REGION_i[19:12] START_REGION_i
DEBUGSS_CT_TBR_FW	START_REGION_i[22:12] START_REGION_i
QSPI_FW	START_REGION_i[25:12] START_REGION_i
CRC_FW	START_REGION_i[19:12] START_REGION_i
TESOC_FW	START_REGION_i[21:12] START_REGION_i
ISS_FW	START_REGION_i[24:12] START_REGION_i

Table 9-44. END_REGION_i

Address Offset	0x0000 0084+(0x10*i)	**Index**	See Table 9-29 to Table 9-34.
Physical Address	0x4A22 4084 + (0x10*i) 0x4A21 C084 + (0x10*i) 0x4A21 0084 + (0x10*i) 0x4A15 B084 + (0x10*i) 0x4A21 2084 + (0x10*i) 0x4A20 C084 + (0x10*i) 0x4A17 9084 + (0x10*i) 0x4A16 3084+(0x10*i) 0x4A24 4084+(0x10*i) 0x4A24 0084+(0x10*i) 0x4A23 E084+(0x10*i)	**Instance**	DEBUGSS_CT_TBR_FW DSS_FW GPMC_FW IPU_FW OCMC_RAM_FW EMIF_FW QSPI_FW TPTC_FW CRC_FW TESOC_FW ISS_FW
Description	End physical address of region i		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
																				RESERVED											END_REGION_i_ENABLE

Bits	Field Name	Description	Type	Reset
31:12	END_REGION_i	Physical target end address of firewall region i. The size of this bit field depends on target addressable space, the maximum is [31:12]. See Table 9-46. Each of the LSbits is assumed to be 1.The programmed address is included in the region i boundary.	RW	0x00000
11:1	RESERVED	Reads return 0s.	R	0x0000
0	END_REGION_i_ENABLE	Enable this region.	RW	0x0

Table 9-45. Register Call Summary for Register END_REGION_i

L3_MAIN Interconnect

- L3_MAIN Firewall Functionality: [0][1][2][3]
- L3_MAIN Firewall Registers Summary and Description: [4][5][6][7][8][9][10][11][12][13][14][15][16][17][18][19][20][21][22][23][24][25][26][27][28][29][30][31][32]

Table 9-46. Size of END_REGION_i[] END_REGION_i Bit Field

Firewall	Bit Field
DSS_FW	END_REGION_i[22:12] END_REGION_i
GPMC_FW	END_REGION_i[28:12] END_REGION_i
IPU_FW	END_REGION_i[22:12] END_REGION_i
OCMC_RAM_FW	END_REGION_i[22:12] END_REGION_i
EMIF_FW	END_REGION_i[30:12] END_REGION_i
TPTC_FW	END_REGION_i[19:12] END_REGION_i
DEBUGSS_CT_TBR_FW	END_REGION_i[22:12] END_REGION_i
QSPI_FW	END_REGION_i[25:12] END_REGION_i
CRC_FW	END_REGION_i[19:12] END_REGION_i
TESOC_FW	END_REGION_i[21:12] END_REGION_i
ISS_FW	END_REGION_i[24:12] END_REGION_i

Table 9-47. MRM_PERMISSION_REGION_HIGH_j

Address Offset	0x0000 008C+(0x10*j)	**Index**	See Table 9-29 to Table 9-34.
Physical Address	0x4A22 408C + (0x10*j) 0x4A17 108C + (0x10*j) 0x4A17 308C + (0x10*j) 0x4A21 C08C + (0x10*j) 0x4A15 108C + (0x10*j) 0x4A21 008C + (0x10*j) 0x4A15 B08C + (0x10*j) 0x4A22 608C + (0x10*j) 0x4A21 208C + (0x10*j) 0x4A20 C08C + (0x10*j) 0x4A17 908C + (0x10*j) 0x4A16 108C + (0x10*j) 0x4A16 308C+(0x10*j) 0x4A24 408C+(0x10*j) 0x4A24 008C+(0x10*j) 0x4A23 E08C+(0x10*j) 0x4A16 708C+(0x10*j)	**Instance**	DEBUGSS_CT_TBR_FW DSP1_SDMA_FW DSP2_SDMA_FW DSS_FW EVE_FW GPMC_FW IPU_FW L3_INSTR_FW OCMC_RAM_FW EMIF_FW QSPI_FW EDMA_TPCC_FW TPTC_FW CRC_FW TESOC_FW ISS_FW MCASP1_FW
Description	Region j Permission High		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
W15	R15	W14	R14	W13	R13	W12	R12	W11	R11	W10	R10	W9	R9	W8	R8	W7	R7	W6	R6	W5	R5	W4	R4	W3	R3	W2	R2	W1	R1	W0	R0

Bits	Field Name	Description	Type	Reset
31	W15	Master NIU ConnID = 115 write permission	RW	0x1
30	R15	Master NIU ConnID = 115 read permission	RW	0x1
29	W14	Master NIU ConnID = 14 write permission	RW	0x1
28	R14	Master NIU ConnID = 14 read permission	RW	0x1
27	W13	Master NIU ConnID = 13 write permission	RW	0x1
26	R13	Master NIU ConnID = 13 read permission	RW	0x1
25	W12	Master NIU ConnID = 12 write permission	RW	0x1
24	R12	Master NIU ConnID = 12 read permission	RW	0x1
23	W11	Master NIU ConnID = 11 write permission	RW	0x1
22	R11	Master NIU ConnID = 11 read permission	RW	0x1
21	W10	Master NIU ConnID = 10 write permission	RW	0x1
20	R10	Master NIU ConnID = 10 read permission	RW	0x1
19	W9	Master NIU ConnID = 9 write permission	RW	0x1
18	R9	Master NIU ConnID = 9 read permission	RW	0x1
17	W8	Master NIU ConnID = 8 write permission	RW	0x1
16	R8	Master NIU ConnID = 8 read permission	RW	0x1
15	W7	Master NIU ConnID = 7 write permission	RW	0x1
14	R7	Master NIU ConnID = 7 read permission	RW	0x1
13	W6	Master NIU ConnID = 6 write permission	RW	0x1
12	R6	Master NIU ConnID = 6 read permission	RW	0x1
11	W5	Master NIU ConnID = 5 write permission	RW	0x1
10	R5	Master NIU ConnID = 5 read permission	RW	0x1
9	W4	Master NIU ConnID = 4 write permission	RW	0x1
8	R4	Master NIU ConnID = 4 read permission	RW	0x1
7	W3	Master NIU ConnID = 3 write permission	RW	0x1
6	R3	Master NIU ConnID = 3 read permission	RW	0x1
5	W2	Master NIU ConnID = 2 write permission	RW	0x1
4	R2	Master NIU ConnID = 2 read permission	RW	0x1

Bits	Field Name	Description	Type	Reset
3	W1	Master NIU ConnID = 1 write permission	RW	0x1
2	R1	Master NIU ConnID = 1 read permission	RW	0x1
1	W0	Master NIU ConnID = 0 write permission	RW	0x1
0	R0	Master NIU ConnID = 0 read permission	RW	0x1

Table 9-48. Register Call Summary for Register MRM_PERMISSION_REGION_HIGH_j

L3_MAIN Interconnect

- L3_MAIN Firewall Functionality: [0][1][2][3][4]
- L3_MAIN Firewall Registers Summary and Description: [5][6][7][8][9][10]

Table 9-49. MRM_PERMISSION_REGION_LOW_j

Address Offset	0x0000 0088+(0x10*j)	**Index**	See Table 9-29 to Table 9-34.
Physical Address	0x4A22 4088 + (0x10*j) 0x4A17 1088 + (0x10*j) 0x4A17 3088 + (0x10*j) 0x4A21 C088 + (0x10*j) 0x4A15 1088 + (0x10*j) 0x4A21 0088 + (0x10*j) 0x4A15 B088 + (0x10*j) 0x4A22 6088 + (0x10*j) 0x4A21 2088 + (0x10*j) 0x4A20 C088 + (0x10*j) 0x4A17 9088 + (0x10*j) 0x4A16 1088 + (0x10*j) 0x4A16 3088+(0x10*j) 0x4A24 4088+(0x10*j) 0x4A24 0088+(0x10*j) 0x4A23 E088+(0x10*j) 0x4A16 7088+(0x10*j)	**Instance**	DEBUGSS_CT_TBR_FW DSP1_SDMA_FW DSP2_SDMA_FW DSS_FW EVE_FW GPMC_FW IPU_FW L3_INSTR_FW OCMC_RAM_FW EMIF_FW QSPI_FW EDMA_TPCC_FW TPTC_FW CRC_FW TESOC_FW ISS_FW MCASP1_FW
Description	Region j Permission Low		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																PUB_PRV_DEBUG	PUB_USR_DEBUG	RESERVED		PUB_PRV_READ	PUB_PRV_WRITE	PUB_PRV_EXE	PUB_USR_READ	PUB_USR_WRITE	PUB_USR_EXE	RESERVED					

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	RESERVED	R	0xFFFF
15	PUB_PRV_DEBUG	Public Privilege Debug Allowed	RW	0x1
14	PUB_USR_DEBUG	Public User Debug Allowed	RW	0x1
13:12	RESERVED	RESERVED	R	0x3
11	PUB_PRV_READ	Public Privilege Read Allowed	RW	0x1
10	PUB_PRV_WRITE	Public Privilege Write Allowed	RW	0x1
9	PUB_PRV_EXE	Public Privilege Exe Allowed	RW	0x1
8	PUB_USR_READ	Public User Read Access Allowed	RW	0x1
7	PUB_USR_WRITE	Public User Write Access Allowed	RW	0x1
6	PUB_USR_EXE	Public User Exe Access Allowed	RW	0x1
5:0	RESERVED	RESERVED	R	0x3F

Table 9-50. Register Call Summary for Register MRM_PERMISSION_REGION_LOW_j

L3_MAIN Interconnect

- L3_MAIN Firewall Functionality: [5][6][7][8][9][10][11][12][13][14][15]
- L3_MAIN Firewall Registers Summary and Description: [16][17][18][19][20][21]

9.2.5.1.2 L3_MAIN Host Register Summary and Description

Table 9-51. HOST Instance Summary

Module Name	Base Address	Size
CLK1_HOST_CLK1_1	0x4400 0000	8MiB
CLK1_HOST_CLK1_2	0x4480 0000	8MiB
CLK2_HOST_CLK2_1	0x4500 0000	8MiB

9.2.5.1.2.1 L3_MAIN HOST Register Summary

Table 9-52. HOST Registers Summary

Register Name	Type	Register Width (Bits)	Address Offset	CLK1_HOST_CLK1_1 L3_MAIN Physical Address	CLK1_HOST_CLK1_2 L3_MAIN Physical Address	CLK2_HOST_CLK2_1 L3_MAIN Physical Address
L3_HOST_STDHOSTHDR_COREREG	R	32	0x0000 0000	0x4400 0000	0x4480 0000	0x4500 0000
L3_HOST_STDHOSTHDR_VERSIONREG	R	32	0x0000 0004	0x4400 0004	0x4480 0004	0x4500 0004
L3_HOST_STDHOSTHDR_MAINCTLREG	R	32	0x0000 0008	0x4400 0008	0x4480 0008	0x4500 0008
L3_HOST_STDERRLOG_SVRTSTDLVL	RW	32	0x0000 0040	0x4400 0040	0x4480 0040	0x4500 0040
L3_HOST_STDERRLOG_SVRTCUSTOMLVL	RW	32	0x0000 0044	0x4400 0044	0x4480 0044	0x4500 0044
L3_HOST_STDERRLOG_MAIN	RW	32	0x0000 0048	0x4400 0048	0x4480 0048	0x4500 0048
L3_HOST_STDERRLOG_HDR	R	32	0x0000 004C	0x4400 004C	0x4480 004C	0x4500 004C
L3_HOST_STDERRLOG_MSTADDR	R	32	0x0000 0050	0x4400 0050	0x4480 0050	0x4500 0050
L3_HOST_STDERRLOG_SLVADDR	R	32	0x0000 0054	0x4400 0054	0x4480 0054	0x4500 0054
L3_HOST_STDERRLOG_INFO	R	32	0x0000 0058	0x4400 0058	0x4480 0058	0x4500 0058
L3_HOST_STDERRLOG_SLVOFSLSB	R	32	0x0000 005C	0x4400 005C	0x4480 005C	0x4500 005C
L3_HOST_STDERRLOG_SLVOFSMSB	R	32	0x0000 0060	0x4400 0060	0x4480 0060	0x4500 0060
L3_HOST_STDERRLOG_CUSTOMINFO_MSTADDR	R	32	0x0000 0064	0x4400 0064	0x4480 0064	0x4500 0064
L3_HOST_STDERRLOG_CUSTOMINFO_INFO	R	32	0x0000 0068	0x4400 0068	0x4480 0068	0x4500 0068
L3_HOST_STDERRLOG_CUSTOMINFO_WR	R	32	0x0000 006C	0x4400 006C	0x4480 006C	0x4500 006C
L3_HOST_STDERRLOG_CUSTOMINFO_ADDR	R	32	0x0000 0070	0x4400 0070	0x4480 0070	0x4500 0070
L3_HOST_STDERRLOG_CUSTOMINFO_DECERR	R	32	0x0000 0074	0x4400 0074	0x4480 0074	0x4500 0074

9.2.5.1.2.2 L3_MAIN HOST Register Description

Table 9-53. L3_HOST_STDHOSTHDR_COREREG

Address Offset	See Table 9-52.		
Physical Address	0x4400 0000 0x4480 0000 0x4500 0000	**Instance**	CLK1_HOST_CLK1_1 CLK1_HOST_CLK1_2 CLK2_HOST_CLK2_1
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								STDHOSTHDR_COREREG_CORECODE								RESERVED								STDHOSTHDR_COREREG_VENDORCODE							

Bits	Field Name	Description	Type	Reset
31:22	RESERVED		R	0x000
21:16	STDHOSTHDR_COREREG_CORECODE	The Core Code field is a constant reporting a vendor-specific core generator code. Type: Constant. Reset value: 0x1A.	R	0x1A
15:1	RESERVED		R	0x0000
0	STDHOSTHDR_COREREG_VENDORCODE	The Vendor Code field is a constant reporting the core generator vendor code. Type: Constant. Reset value: 0x1. Read 0x0: Third-party vendor. Read 0x1:	R	1

Table 9-54. Register Call Summary for Register L3_HOST_STDHOSTHDR_COREREG

L3_MAIN Interconnect
- L3_MAIN Host Register Summary and Description: [0]

Table 9-55. L3_HOST_STDHOSTHDR_VERSIONREG

Address Offset	See Table 9-52.		
Physical Address	0x4400 0004 0x4480 0004 0x4500 0004	**Instance**	CLK1_HOST_CLK1_1 CLK1_HOST_CLK1_2 CLK2_HOST_CLK2_1
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STDHOSTHDR_VERSIONREG_REVISIONID								STDHOSTHDR_VERSIONREG_COREPARAMSCHECKSUM																							

Bits	Field Name	Description	Type	Reset
31:24	STDHOSTHDR_VERSIONREG_REVISIONID	The Revision Identifier field is a constant reporting the core generator revision number. Type: Constant. Reset value: 0x0.	R	0x00
23:0	STDHOSTHDR_VERSIONREG_COREPARAMSCHECKSUM	Reserved. Type: Reserved. Reset value: Reserved.	R	0x000000

Table 9-56. Register Call Summary for Register L3_HOST_STDHOSTHDR_VERSIONREG

L3_MAIN Interconnect

- L3_MAIN Host Register Summary and Description: [0]

Table 9-57. L3_HOST_STDHOSTHDR_MAINCTLREG

Address Offset	See Table 9-52.		
Physical Address	0x4400 0008 0x4480 0008 0x4500 0008	**Instance**	CLK1_HOST_CLK1_1 CLK1_HOST_CLK1_2 CLK2_HOST_CLK2_1
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																													STDHOSTHDR_MAINCTLREG_FLT	RESERVED	

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0000 0000
2	STDHOSTHDR_MAINCTLREG_FLT	Asserted when a Fault condition is detected: if the unit includes Error Logging, Fault is asserted when the Fault Control register field indicates a Fault, and de-asserted when FltCnt is reset. If no Error Logging is implemented, this bit becomes unit-dependent. In all cases, Flt bit and Flt pin (service network) have the same logical level. Type: Status. Reset value: X.	R	0
1:0	RESERVED		R	0x0

Table 9-58. Register Call Summary for Register L3_HOST_STDHOSTHDR_MAINCTLREG

L3_MAIN Interconnect

- L3_MAIN Host Register Summary and Description: [0]

Table 9-59. L3_HOST_STDERRLOG_SVRTSTDLVL

Address Offset	See Table 9-52.		
Physical Address	0x4400 0040 0x4480 0040 0x4500 0040	**Instance**	CLK1_HOST_CLK1_1 CLK1_HOST_CLK1_2 CLK2_HOST_CLK2_1
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														STDERRLOG_SVRTSTDLVL_0	

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0000 0000
1:0	STDERRLOG_SVRTSTDLVL_0	Severity level parameters Type: Control. Reset value: 0x2. 0x0: Error logging is disabled. 0x1: Errors are logged with severity level Error. 0x2: Errors are logged with severity level Fault.	RW	0x2

Table 9-60. Register Call Summary for Register L3_HOST_STDERRLOG_SVRTSTDLVL

L3_MAIN Interconnect
- L3_MAIN Host Register Summary and Description: [0]

Table 9-61. L3_HOST_STDERRLOG_SVRTCUSTOMLVL

Address Offset	See Table 9-52.		
Physical Address	0x4400 0044 0x4480 0044 0x4500 0044	**Instance**	CLK1_HOST_CLK1_1 CLK1_HOST_CLK1_2 CLK2_HOST_CLK2_1
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														STDERRLOG_SVRTCUSTOMLVL_0	

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0000 0000
1:0	STDERRLOG_SVRTCUSTOMLVL_0	Severity level parameters Type: Control. Reset value: 0x2. 0x0: Error logging is disabled. 0x1: Errors are logged with severity level Error. 0x2: Errors are logged with severity level Fault.	RW	0x2

Table 9-62. Register Call Summary for Register L3_HOST_STDERRLOG_SVRTCUSTOMLVL

L3_MAIN Interconnect

- L3_MAIN Host Register Summary and Description: [0]

Table 9-63. L3_HOST_STDERRLOG_MAIN

Address Offset	See Table 9-52.		
Physical Address	0x4400 0048 0x4480 0048 0x4500 0048	**Instance**	CLK1_HOST_CLK1_1 CLK1_HOST_CLK1_2 CLK2_HOST_CLK2_1
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STDERRLOG_MAIN_CLRLOG	RESERVED											STDERRLOG_MAIN_FLTCNT	STDERRLOG_MAIN_ERRCNT	RESERVED																	STDERRLOG_MAIN_ERRTYPE	STDERRLOG_MAIN_ERRLOGVLD

Bits	Field Name	Description	Type	Reset
31	STDERRLOG_MAIN_CLRLOG	Clears "Error Logging Valid" bit when written to 1. Type: Give_AutoCleared. Reset value: 0x0.	RW	0
30:20	RESERVED		R	0x000
19	STDERRLOG_MAIN_FLTCNT	Asserted when at least one error with severity level FAULT is detected. Reset when written to 1. Type: Take. Reset value: 0x0.	RW	0
18	STDERRLOG_MAIN_ERRCNT	Asserted when at least one error with severity level ERROR is detected. Reset when written to 1. Type: Take. Reset value: 0x0.	RW	0
17:2	RESERVED		R	0x0000
1	STDERRLOG_MAIN_ERRTYPE	Indicates logging type. Type: Status. Reset value: X. Read 0x0: Logged Error format is standard (header and necker recorded). Read 0x1: Logged Error format is module dependent. CustomInfo register(s) may be implemented to store additional information.	R	0
0	STDERRLOG_MAIN_ERRLOGVLD	Error Logging Valid. Asserted when logging information is valid(indicates that an error has been logged). Type: Status. Reset value: X.	R	0

Table 9-64. Register Call Summary for Register L3_HOST_STDERRLOG_MAIN

L3_MAIN Interconnect

- Example for Decoding Standard/Custom Errors Logged in L3_MAIN: [0]
- L3_MAIN Host Register Summary and Description: [1]

Table 9-65. L3_HOST_STDERRLOG_HDR

Address Offset	See Table 9-52.		
Physical Address	0x4400 004C 0x4480 004C 0x4500 004C	**Instance**	CLK1_HOST_CLK1_1 CLK1_HOST_CLK1_2 CLK2_HOST_CLK2_1
Description			
Type	R		

31 30 29 28	27 26 25 24	23 22 21 20 19 18	17 16	15 14 13 12	11	10 9 8	7 6	5 4	3 2 1 0
RESERVED	STDERRLOG_HDR_LEN1		RESERVED	STDERRLOG_HDR_STOPOFSWRPSZ	STDERRLOG_HDR_ERR	RESERVED	STDERRLOG_HDR_PRESSURE	RESERVED	STDERRLOG_HDR_OPCODE

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x00
27:18	STDERRLOG_HDR_LEN1	This field contains the number of payload cell(s) minus one of the logged packet. Type: Status. Reset value: X.	R	0x00
17:16	RESERVED		R	0x0
15:12	STDERRLOG_HDR_STOPOFSWRPSZ	StopOfs or WrapSize field of the logged packet (meaning depends on Wrp bit of logged opcode). Type: Status. Reset value: X.	R	0x0
11	STDERRLOG_HDR_ERR	Err bit of the logged packet. Type: Status. Reset value: X.	R	0
10:8	RESERVED		R	0x0
7:6	STDERRLOG_HDR_PRESSURE	Pressure field of the logged packet. Type: Status. Reset value: X.	R	0
5:4	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
3:0	STDERRLOG_HDR_OPCODE	Opcode of the logged packet. Type: Status. Reset value: X. 0x0: Store without acknowledge, incrementing burst, non-atomic request 0x1: Store without acknowledge, wrapping burst, non-atomic request 0x2: Store with acknowledge, incrementing burst, non-atomic request 0x3: Store with acknowledge, wrapping burst, non-atomic request 0x4: Load, incrementing burst, non-atomic request 0x5: Load, wrapping burst, non-atomic request 0x6: Control packet 0x7: Flush 0x8: Store without acknowledge, incrementing burst, atomic request 0x9: Store without acknowledge, wrapping burst, atomic request 0xA: Store with acknowledge, incrementing burst, atomic request 0xB: Store with acknowledge, wrapping burst, atomic request 0xC: Load, incrementing burst, atomic request 0xD: Load, wrapping burst, atomic request 0xE: Reserved 0xF: Reserved	R	0x0

Table 9-66. Register Call Summary for Register L3_HOST_STDERRLOG_HDR

L3_MAIN Interconnect
- L3_MAIN Host Register Summary and Description: [0]

Table 9-67. L3_HOST_STDERRLOG_MSTADDR

Address Offset	See Table 9-52.		
Physical Address	0x4400 0050 0x4480 0050 0x4500 0050	**Instance**	CLK1_HOST_CLK1_1 CLK1_HOST_CLK1_2 CLK2_HOST_CLK2_1
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								STDERRLOG_MSTADDR							

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x000000
7:0	STDERRLOG_MSTADDR	Master Address field of the logged packet. Type: Status. Reset value: X.	R	0x00

Table 9-68. Register Call Summary for Register L3_HOST_STDERRLOG_MSTADDR

L3_MAIN Interconnect
- L3_MAIN Host Register Summary and Description: [0]

Table 9-69. L3_HOST_STDERRLOG_SLVADDR

Address Offset	See Table 9-52.		
Physical Address	0x4400 0054 0x4480 0054 0x4500 0054	**Instance**	CLK1_HOST_CLK1_1 CLK1_HOST_CLK1_2 CLK2_HOST_CLK2_1
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								STDERRLOG_SLVADDR							

Bits	Field Name	Description	Type	Reset
31:7	RESERVED		R	0x0000000
6:0	STDERRLOG_SLVADDR	Slave Address field of the logged packet. Type: Status. Reset value: X.	R	0x00

Table 9-70. Register Call Summary for Register L3_HOST_STDERRLOG_SLVADDR

L3_MAIN Interconnect

- L3_MAIN Host Register Summary and Description: [0]

Table 9-71. L3_HOST_STDERRLOG_INFO

Address Offset	See Table 9-52.		
Physical Address	0x4400 0058 0x4480 0058 0x4500 0058	**Instance**	CLK1_HOST_CLK1_1 CLK1_HOST_CLK1_2 CLK2_HOST_CLK2_1
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								STDERRLOG_INFO							

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x000000
7:0	STDERRLOG_INFO	Info field of the logged packet. Type: Status. Reset value: X.	R	0x00

Table 9-72. Register Call Summary for Register L3_HOST_STDERRLOG_INFO

L3_MAIN Interconnect

- L3_MAIN Host Register Summary and Description: [0]

Table 9-73. L3_HOST_STDERRLOG_SLVOFSLSB

Address Offset	See Table 9-52.		
Physical Address	0x4400 005C 0x4480 005C 0x4500 005C	**Instance**	CLK1_HOST_CLK1_1 CLK1_HOST_CLK1_2 CLK2_HOST_CLK2_1
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STDERRLOG_SLVOFSLSB																															

Bits	Field Name	Description	Type	Reset
31:0	STDERRLOG_SLVOFSLSB	LSB of the "slave offset" field, concatenated with "start offset" field of the logged packet. Type: Status. Reset value: X.	R	0x0000 0000

Table 9-74. Register Call Summary for Register L3_HOST_STDERRLOG_SLVOFSLSB

L3_MAIN Interconnect

- L3_MAIN Host Register Summary and Description: [0]

Table 9-75. L3_HOST_STDERRLOG_SLVOFSMSB

Address Offset	See Table 9-52.		
Physical Address	0x4400 0060 0x4480 0060 0x4500 0060	**Instance**	CLK1_HOST_CLK1_1 CLK1_HOST_CLK1_2 CLK2_HOST_CLK2_1
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															STDERRLOG_SLVOFSMSB

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	STDERRLOG_SLVOFSMSB	MSB of the "slave offset" field of the logged packet (according to NTTP packet format, this register field may exceed the actual "slave offset" size. Unused bits are stuck at 0, if any). Type: Status. Reset value: X.	R	0

Table 9-76. Register Call Summary for Register L3_HOST_STDERRLOG_SLVOFSMSB

L3_MAIN Interconnect

- L3_MAIN Host Register Summary and Description: [0]

Table 9-77. L3_HOST_STDERRLOG_CUSTOMINFO_MSTADDR

Address Offset	See Table 9-52.		
Physical Address	0x4400 0064 0x4480 0064 0x4500 0064	**Instance**	CLK1_HOST_CLK1_1 CLK1_HOST_CLK1_2 CLK2_HOST_CLK2_1
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								STDERRLOG_CUSTOMINFO_MSTADDR							

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x000000
7:0	STDERRLOG_CUSTOMINFO_MSTADDR	Type: Status. Reset value: X.	R	0x00

Table 9-78. Register Call Summary for Register L3_HOST_STDERRLOG_CUSTOMINFO_MSTADDR

L3_MAIN Interconnect

- L3_MAIN Host Register Summary and Description: [0]

Table 9-79. L3_HOST_STDERRLOG_CUSTOMINFO_INFO

Address Offset	See Table 9-52.		
Physical Address	0x4400 0068 0x4480 0068 0x4500 0068	**Instance**	CLK1_HOST_CLK1_1 CLK1_HOST_CLK1_2 CLK2_HOST_CLK2_1
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								STDERRLOG_CUSTOMINFO_INFO							

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x000000
7:0	STDERRLOG_CUSTOMINFO_INFO	Type: Status. Reset value: X.	R	0x00

Table 9-80. Register Call Summary for Register L3_HOST_STDERRLOG_CUSTOMINFO_INFO

L3_MAIN Interconnect
- L3_MAIN Host Register Summary and Description: [0]

Table 9-81. L3_HOST_STDERRLOG_CUSTOMINFO_WR

Address Offset	See Table 9-52.		
Physical Address	0x4400 006C 0x4480 006C 0x4500 006C	**Instance**	CLK1_HOST_CLK1_1 CLK1_HOST_CLK1_2 CLK2_HOST_CLK2_1
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															STDERRLOG_CUSTOMINFO_WR

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	STDERRLOG_CUSTOMINFO_WR	Type: Status. Reset value: X.	R	0

Table 9-82. Register Call Summary for Register L3_HOST_STDERRLOG_CUSTOMINFO_WR

L3_MAIN Interconnect
- L3_MAIN Host Register Summary and Description: [0]

Table 9-83. L3_HOST_STDERRLOG_CUSTOMINFO_ADDR

Address Offset	See Table 9-52.		
Physical Address	0x4400 0070 0x4480 0070 0x4500 0070	**Instance**	CLK1_HOST_CLK1_1 CLK1_HOST_CLK1_2 CLK2_HOST_CLK2_1
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								STDERRLOG_CUSTOMINFO_ADDR																									

Bits	Field Name	Description	Type	Reset
31:21	RESERVED		R	0x000
20:0	STDERRLOG_CUSTOMINFO_ADDR	Type: Status. Reset value: X.	R	0x000000

Table 9-84. Register Call Summary for Register L3_HOST_STDERRLOG_CUSTOMINFO_ADDR

L3_MAIN Interconnect
- L3_MAIN Host Register Summary and Description: [0]

Table 9-85. L3_HOST_STDERRLOG_CUSTOMINFO_DECERR

Address Offset	See Table 9-52.		
Physical Address	0x4400 0074 0x4480 0074 0x4500 0074	**Instance**	CLK1_HOST_CLK1_1 CLK1_HOST_CLK1_2 CLK2_HOST_CLK2_1
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															STDERRLOG_CUSTOMINFO_DECERR

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	STDERRLOG_CUSTOMINFO_DECERR	Type: Status. Reset value: X.	R	0

Table 9-86. Register Call Summary for Register L3_HOST_STDERRLOG_CUSTOMINFO_DECERR

L3_MAIN Interconnect
- L3_MAIN Host Register Summary and Description: [0]

9.2.5.1.3 L3_MAIN TARG Register Summary and Description

NOTE: ISS, CRC and TESOC are not supported on the DRA78x family of devices.

Table 9-87. L3_MAIN TARG Instance Summary

Module Name	Base Address	Size
GPMC_TARG	0x4400 0100	4KiB
EMIF_P1_TARG	0x4400 0200	4KiB
DSP1_SDMA_TARG	0x4400 0300	4KiB

Table 9-87. L3_MAIN TARG Instance Summary (continued)

Module Name	Base Address	Size
L4_CFG_TARG	0x4400 0500	4KiB
DSP2_SDMA_TARG	0x4400 0600	4KiB
CRC_TARG	0x4400 0700	4KiB
ISS_TARG	0x4400 0800	4KiB
EVE_TARG	0x4400 0A00	4KiB
OCMC_RAM_TARG	0x4400 0F00	4KiB
IPU_TARG	0x4400 1000	4KiB
TESOC_TARG	0x4400 1300	4KiB
L4_PER1_P1_TARG	0x4400 1C00	4KiB
L4_WKUP_TARG	0x4400 1D00	4KiB
L4_PER1_P2_TARG	0x4400 1F00	4KiB
EDMA_TPCC_TARG	0x4400 2000	4KiB
MMU_TARG	0x4400 2200	4KiB
L4_PER2_P1_TARG	0x4400 2300	4KiB
L4_PER3_P1_TARG	0x4400 2600	4KiB
L4_PER3_P2_TARG	0x4400 2700	4KiB
DSS_TARG	0x4400 2900	4KiB
TPTC2_TARG	0x4400 2B00	4KiB
TPTC1_TARG	0x4400 2E00	4KiB
MCASP1_TARG	0x4400 2F00	4KiB
QSPI_TARG	0x4400 3900	4KiB
L3_INSTR	0x4500 0100	4KiB
DEBUGSS_CT_TBR_TARG	0x4500 0300	4KiB

9.2.5.1.3.1 L3_MAIN TARG Register Summary

Table 9-88. L3_MAIN TARG Register Summary

Register Name	Type	Register Width (Bits)	Address Offset	GPMC_TARG L3_MAIN Physical Address	EMIF_P1_TARG L3_MAIN Physical Address	DSP1_SDMA_TARG L3_MAIN Physical Address
L3_TARG_STDHOSTHDR_COREREG	R	32	0x0000 0000	0x4400 0100	0x4400 0200	0x4400 0300
L3_TARG_STDHOSTHDR_VERSIONREG	R	32	0x0000 0004	0x4400 0104	0x4400 0204	0x4400 0304
L3_TARG_STDHOSTHDR_MAINCTLREG	RW	32	0x0000 0008	0x4400 0108	0x4400 0208	0x4400 0308
L3_TARG_STDHOSTHDR_NTTPADDR_0	R	32	0x0000 0010	0x4400 0110	0x4400 0210	0x4400 0310
L3_TARG_STDERRLOG_SVRTSTDLVL	RW	32	0x0000 0040	0x4400 0140	0x4400 0240	0x4400 0340
L3_TARG_STDERRLOG_SVRTCUSTOMLVL	RW	32	0x0000 0044	0x4400 0144	0x4400 0244	0x4400 0344
L3_TARG_STDERRLOG_MAIN	RW	32	0x0000 0048	0x4400 0148	0x4400 0248	0x4400 0348
L3_TARG_STDERRLOG_HDR	R	32	0x0000 004C	0x4400 014C	0x4400 024C	0x4400 034C
L3_TARG_STDERRLOG_MSTADDR	R	32	0x0000 0050	0x4400 0150	0x4400 0250	0x4400 0350
L3_TARG_STDERRLOG_SLVADDR	R	32	0x0000 0054	0x4400 0154	0x4400 0254	0x4400 0354
L3_TARG_STDERRLOG_INFO	R	32	0x0000 0058	0x4400 0158	0x4400 0258	0x4400 0358
L3_TARG_STDERRLOG_SLVOFSLSB	R	32	0x0000 005C	0x4400 015C	0x4400 025C	0x4400 035C
L3_TARG_STDERRLOG_SLVOFSMSB	R	32	0x0000 0060	0x4400 0160	0x4400 0260	0x4400 0360
L3_TARG_STDERRLOG_CUSTOMINFO_INFO	R	32	0x0000 0064	0x4400 0164	0x4400 0264	0x4400 0364
L3_TARG_STDERRLOG_CUSTOMINFO_MSTADDR	R	32	0x0000 0068	0x4400 0168	0x4400 0268	0x4400 0368
L3_TARG_STDERRLOG_CUSTOMINFO_OPCODE	R	32	0x0000 006C	0x4400 016C	0x4400 026C	0x4400 036C
L3_TARG_ADDRSPACESIZELOG	RW	32	0x0000 0080	0x4400 0180	0x4400 0280	0x4400 0380

Table 9-89. L3_MAIN TARG Register Summary

Register Name	Type	Register Width (Bits)	Address Offset	L4_CFG_TARG L3_MAIN Physical Address	DSP2_SDMA_TARG L3_MAIN Physical Address	CRC_TARG L3_MAIN Physical Address
L3_TARG_STDHOSTHDR_COREREG	R	32	0x0000 0000	0x4400 0500	0x4400 0600	0x4400 0700
L3_TARG_STDHOSTHDR_VERSIONREG	R	32	0x0000 0004	0x4400 0504	0x4400 0604	0x4400 0704
L3_TARG_STDHOSTHDR_MAINCTLREG	RW	32	0x0000 0008	0x4400 0508	0x4400 0608	0x4400 0708
L3_TARG_STDHOSTHDR_NTTPADDR_0	R	32	0x0000 0010	0x4400 0510	0x4400 0610	0x4400 0710
L3_TARG_STDERRLOG_SVRTSTDLVL	RW	32	0x0000 0040	0x4400 0540	0x4400 0640	0x4400 0740
L3_TARG_STDERRLOG_SVRTCUSTOMLVL	RW	32	0x0000 0044	0x4400 0544	0x4400 0644	0x4400 0744
L3_TARG_STDERRLOG_MAIN	RW	32	0x0000 0048	0x4400 0548	0x4400 0648	0x4400 0748
L3_TARG_STDERRLOG_HDR	R	32	0x0000 004C	0x4400 054C	0x4400 064C	0x4400 074C
L3_TARG_STDERRLOG_MSTADDR	R	32	0x0000 0050	0x4400 0550	0x4400 0650	0x4400 0750

Table 9-89. L3_MAIN TARG Register Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	L4_CFG_TARG L3_MAIN Physical Address	DSP2_SDMA_TARG L3_MAIN Physical Address	CRC_TARG L3_MAIN Physical Address
L3_TARG_STDERRLOG_SLVADDR	R	32	0x0000 0054	0x4400 0554	0x4400 0654	0x4400 0754
L3_TARG_STDERRLOG_INFO	R	32	0x0000 0058	0x4400 0558	0x4400 0658	0x4400 0758
L3_TARG_STDERRLOG_SLVOFSLSB	R	32	0x0000 005C	0x4400 055C	0x4400 065C	0x4400 075C
L3_TARG_STDERRLOG_SLVOFSMSB	R	32	0x0000 0060	0x4400 0560	0x4400 0660	0x4400 0760
L3_TARG_STDERRLOG_CUSTOMINFO_INFO	R	32	0x0000 0064	0x4400 0564	0x4400 0664	0x4400 0764
L3_TARG_STDERRLOG_CUSTOMINFO_MSTADDR	R	32	0x0000 0068	0x4400 0568	0x4400 0668	0x4400 0768
L3_TARG_STDERRLOG_CUSTOMINFO_OPCODE	R	32	0x0000 006C	0x4400 056C	0x4400 066C	0x4400 076C
L3_TARG_ADDRSPACESIZELOG	RW	32	0x0000 0080	0x4400 0580	0x4400 0680	0x4400 0780

Table 9-90. L3_MAIN TARG Register Summary

Register Name	Type	Register Width (Bits)	Address Offset	ISS_TARG L3_MAIN Physical Address	MCASP1_TARG L3_MAIN Physical Address	EVE_TARG L3_MAIN Physical Address
L3_TARG_STDHOSTHDR_COREREG	R	32	0x0000 0000	0x4400 0800	0x4400 2F00	0x4400 0A00
L3_TARG_STDHOSTHDR_VERSIONREG	R	32	0x0000 0004	0x4400 0804	0x4400 2F04	0x4400 0A04
L3_TARG_STDHOSTHDR_MAINCTLREG	RW	32	0x0000 0008	0x4400 0808	0x4400 2F08	0x4400 0A08
L3_TARG_STDHOSTHDR_NTTPADDR_0	R	32	0x0000 0010	0x4400 0810	0x4400 2F10	0x4400 0A10
L3_TARG_STDERRLOG_SVRTSTDLVL	RW	32	0x0000 0040	0x4400 0840	0x4400 2F40	0x4400 0A40
L3_TARG_STDERRLOG_SVRTCUSTOMLVL	RW	32	0x0000 0044	0x4400 0844	0x4400 2F44	0x4400 0A44
L3_TARG_STDERRLOG_MAIN	RW	32	0x0000 0048	0x4400 0848	0x4400 2F48	0x4400 0A48
L3_TARG_STDERRLOG_HDR	R	32	0x0000 004C	0x4400 084C	0x4400 2F4C	0x4400 0A4C
L3_TARG_STDERRLOG_MSTADDR	R	32	0x0000 0050	0x4400 0850	0x4400 2F50	0x4400 0A50
L3_TARG_STDERRLOG_SLVADDR	R	32	0x0000 0054	0x4400 0854	0x4400 2F54	0x4400 0A54
L3_TARG_STDERRLOG_INFO	R	32	0x0000 0058	0x4400 0858	0x4400 2F58	0x4400 0A58
L3_TARG_STDERRLOG_SLVOFSLSB	R	32	0x0000 005C	0x4400 085C	0x4400 2F5C	0x4400 0A5C
L3_TARG_STDERRLOG_SLVOFSMSB	R	32	0x0000 0060	0x4400 0860	0x4400 2F60	0x4400 0A60
L3_TARG_STDERRLOG_CUSTOMINFO_INFO	R	32	0x0000 0064	0x4400 0864	0x4400 2F64	0x4400 0A64
L3_TARG_STDERRLOG_CUSTOMINFO_MSTADDR	R	32	0x0000 0068	0x4400 0868	0x4400 2F68	0x4400 0A68
L3_TARG_STDERRLOG_CUSTOMINFO_OPCODE	R	32	0x0000 006C	0x4400 086C	0x4400 2F6C	0x4400 0A6C
L3_TARG_ADDRSPACESIZELOG	RW	32	0x0000 0080	0x4400 0880	0x4400 2F80	0x4400 0A80

Table 9-91. L3_MAIN TARG Register Summary

Register Name	Type	Register Width (Bits)	Address Offset	OCMC_RAM_TARG L3_MAIN Physical Address	IPU_TARG L3_MAIN Physical Address	TESOC_TARG L3_MAIN Physical Address
L3_TARG_STDHOSTHDR_COREREG	R	32	0x0000 0000	0x4400 0F00	0x4400 1000	0x4400 1300
L3_TARG_STDHOSTHDR_VERSIONREG	R	32	0x0000 0004	0x4400 0F04	0x4400 1004	0x4400 1304
L3_TARG_STDHOSTHDR_MAINCTLREG	RW	32	0x0000 0008	0x4400 0F08	0x4400 1008	0x4400 1308
L3_TARG_STDHOSTHDR_NTTPADDR_0	R	32	0x0000 0010	0x4400 0F10	0x4400 1010	0x4400 1310
L3_TARG_STDERRLOG_SVRTSTDLVL	RW	32	0x0000 0040	0x4400 0F40	0x4400 1040	0x4400 1340
L3_TARG_STDERRLOG_SVRTCUSTOMLVL	RW	32	0x0000 0044	0x4400 0F44	0x4400 1044	0x4400 1344
L3_TARG_STDERRLOG_MAIN	RW	32	0x0000 0048	0x4400 0F48	0x4400 1048	0x4400 1348
L3_TARG_STDERRLOG_HDR	R	32	0x0000 004C	0x4400 0F4C	0x4400 104C	0x4400 134C
L3_TARG_STDERRLOG_MSTADDR	R	32	0x0000 0050	0x4400 0F50	0x4400 1050	0x4400 1350
L3_TARG_STDERRLOG_SLVADDR	R	32	0x0000 0054	0x4400 0F54	0x4400 1054	0x4400 1354
L3_TARG_STDERRLOG_INFO	R	32	0x0000 0058	0x4400 0F58	0x4400 1058	0x4400 1358
L3_TARG_STDERRLOG_SLVOFSLSB	R	32	0x0000 005C	0x4400 0F5C	0x4400 105C	0x4400 135C
L3_TARG_STDERRLOG_SLVOFSMSB	R	32	0x0000 0060	0x4400 0F60	0x4400 1060	0x4400 1360
L3_TARG_STDERRLOG_CUSTOMINFO_INFO	R	32	0x0000 0064	0x4400 0F64	0x4400 1064	0x4400 1364
L3_TARG_STDERRLOG_CUSTOMINFO_MSTADDR	R	32	0x0000 0068	0x4400 0F68	0x4400 1068	0x4400 1368
L3_TARG_STDERRLOG_CUSTOMINFO_OPCODE	R	32	0x0000 006C	0x4400 0F6C	0x4400 106C	0x4400 136C
L3_TARG_ADDRSPACESIZELOG	RW	32	0x0000 0080	0x4400 0F80	0x4400 1080	0x4400 1380

Table 9-92. L3_MAIN TARG Register Summary

Register Name	Type	Register Width (Bits)	Address Offset	L4_PER1_P1_TARG L3_MAIN Physical Address	L4_WKUP_TARG L3_MAIN Physical Address	L4_PER1_P2_TARG L3_MAIN Physical Address
L3_TARG_STDHOSTHDR_COREREG	R	32	0x0000 0000	0x4400 1C00	0x4400 1D00	0x4400 1F00
L3_TARG_STDHOSTHDR_VERSIONREG	R	32	0x0000 0004	0x4400 1C04	0x4400 1D04	0x4400 1F04
L3_TARG_STDHOSTHDR_MAINCTLREG	RW	32	0x0000 0008	0x4400 1C08	0x4400 1D08	0x4400 1F08
L3_TARG_STDHOSTHDR_NTTPADDR_0	R	32	0x0000 0010	0x4400 1C10	0x4400 1D10	0x4400 1F10
L3_TARG_STDERRLOG_SVRTSTDLVL	RW	32	0x0000 0040	0x4400 1C40	0x4400 1D40	0x4400 1F40
L3_TARG_STDERRLOG_SVRTCUSTOMLVL	RW	32	0x0000 0044	0x4400 1C44	0x4400 1D44	0x4400 1F44
L3_TARG_STDERRLOG_MAIN	RW	32	0x0000 0048	0x4400 1C48	0x4400 1D48	0x4400 1F48
L3_TARG_STDERRLOG_HDR	R	32	0x0000 004C	0x4400 1C4C	0x4400 1D4C	0x4400 1F4C
L3_TARG_STDERRLOG_MSTADDR	R	32	0x0000 0050	0x4400 1C50	0x4400 1D50	0x4400 1F50
L3_TARG_STDERRLOG_SLVADDR	R	32	0x0000 0054	0x4400 1C54	0x4400 1D54	0x4400 1F54
L3_TARG_STDERRLOG_INFO	R	32	0x0000 0058	0x4400 1C58	0x4400 1D58	0x4400 1F58

Table 9-92. L3_MAIN TARG Register Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	L4_PER1_P1_TARG L3_MAIN Physical Address	L4_WKUP_TARG L3_MAIN Physical Address	L4_PER1_P2_TARG L3_MAIN Physical Address
L3_TARG_STDERRLOG_SLVOFSLSB	R	32	0x0000 005C	0x4400 1C5C	0x4400 1D5C	0x4400 1F5C
L3_TARG_STDERRLOG_SLVOFSMSB	R	32	0x0000 0060	0x4400 1C60	0x4400 1D60	0x4400 1F60
L3_TARG_STDERRLOG_CUSTOMINFO_INFO	R	32	0x0000 0064	0x4400 1C64	0x4400 1D64	0x4400 1F64
L3_TARG_STDERRLOG_CUSTOMINFO_MSTADDR	R	32	0x0000 0068	0x4400 1C68	0x4400 1D68	0x4400 1F68
L3_TARG_STDERRLOG_CUSTOMINFO_OPCODE	R	32	0x0000 006C	0x4400 1C6C	0x4400 1D6C	0x4400 1F6C
L3_TARG_ADDRSPACESIZELOG	RW	32	0x0000 0080	0x4400 1C80	0x4400 1D80	0x4400 1F80

Table 9-93. L3_MAIN TARG Register Summary

Register Name	Type	Register Width (Bits)	Address Offset	EDMA_TPCC_TARG L3_MAIN Physical Address	MMU_TARG L3_MAIN Physical Address	L4_PER2_P1_TARG L3_MAIN Physical Address
L3_TARG_STDHOSTHDR_COREREG	R	32	0x0000 0000	0x4400 2000	0x4400 2200	0x4400 2300
L3_TARG_STDHOSTHDR_VERSIONREG	R	32	0x0000 0004	0x4400 2004	0x4400 2204	0x4400 2304
L3_TARG_STDHOSTHDR_MAINCTLREG	RW	32	0x0000 0008	0x4400 2008	0x4400 2208	0x4400 2308
L3_TARG_STDHOSTHDR_NTTPADDR_0	R	32	0x0000 0010	0x4400 2010	0x4400 2210	0x4400 2310
L3_TARG_STDERRLOG_SVRTSTDLVL	RW	32	0x0000 0040	0x4400 2040	0x4400 2240	0x4400 2340
L3_TARG_STDERRLOG_SVRTCUSTOMLVL	RW	32	0x0000 0044	0x4400 2044	0x4400 2244	0x4400 2344
L3_TARG_STDERRLOG_MAIN	RW	32	0x0000 0048	0x4400 2048	0x4400 2248	0x4400 2348
L3_TARG_STDERRLOG_HDR	R	32	0x0000 004C	0x4400 204C	0x4400 224C	0x4400 234C
L3_TARG_STDERRLOG_MSTADDR	R	32	0x0000 0050	0x4400 2050	0x4400 2250	0x4400 2350
L3_TARG_STDERRLOG_SLVADDR	R	32	0x0000 0054	0x4400 2054	0x4400 2254	0x4400 2354
L3_TARG_STDERRLOG_INFO	R	32	0x0000 0058	0x4400 2058	0x4400 2258	0x4400 2358
L3_TARG_STDERRLOG_SLVOFSLSB	R	32	0x0000 005C	0x4400 205C	0x4400 225C	0x4400 235C
L3_TARG_STDERRLOG_SLVOFSMSB	R	32	0x0000 0060	0x4400 2060	0x4400 2260	0x4400 2360
L3_TARG_STDERRLOG_CUSTOMINFO_INFO	R	32	0x0000 0064	0x4400 2064	0x4400 2264	0x4400 2364
L3_TARG_STDERRLOG_CUSTOMINFO_MSTADDR	R	32	0x0000 0068	0x4400 2068	0x4400 2268	0x4400 2368
L3_TARG_STDERRLOG_CUSTOMINFO_OPCODE	R	32	0x0000 006C	0x4400 206C	0x4400 226C	0x4400 236C
L3_TARG_ADDRSPACESIZELOG	RW	32	0x0000 0080	0x4400 2080	0x4400 2280	0x4400 2380

Table 9-94. L3_MAIN TARG Register Summary

Register Name	Type	Register Width (Bits)	Address Offset	L4_PER3_P1_TARG L3_MAIN Physical Address	L4_PER3_P2_TARG L3_MAIN Physical Address	DSS_TARG L3_MAIN Physical Address	TPTC2_TARGL3_MAIN Physical Address
L3_TARG_STDHOSTHDR_COREREG	R	32	0x0000 0000	0x4400 2600	0x4400 2700	0x4400 2900	0x4400 2B00
L3_TARG_STDHOSTHDR_VERSIONREG	R	32	0x0000 0004	0x4400 2604	0x4400 2704	0x4400 2904	0x4400 2B04
L3_TARG_STDHOSTHDR_MAINCTLREG	RW	32	0x0000 0008	0x4400 2608	0x4400 2708	0x4400 2908	0x4400 2B08
L3_TARG_STDHOSTHDR_NTTPADDR_0	R	32	0x0000 0010	0x4400 2610	0x4400 2710	0x4400 2910	0x4400 2B10
L3_TARG_STDERRLOG_SVRTSTDLVL	RW	32	0x0000 0040	0x4400 2640	0x4400 2740	0x4400 2940	0x4400 2B40
L3_TARG_STDERRLOG_SVRTCUSTOMLVL	RW	32	0x0000 0044	0x4400 2644	0x4400 2744	0x4400 2944	0x4400 2B44
L3_TARG_STDERRLOG_MAIN	RW	32	0x0000 0048	0x4400 2648	0x4400 2748	0x4400 2948	0x4400 2B48
L3_TARG_STDERRLOG_HDR	R	32	0x0000 004C	0x4400 264C	0x4400 274C	0x4400 294C	0x4400 2B4C
L3_TARG_STDERRLOG_MSTADDR	R	32	0x0000 0050	0x4400 2650	0x4400 2750	0x4400 2950	0x4400 2B50
L3_TARG_STDERRLOG_SLVADDR	R	32	0x0000 0054	0x4400 2654	0x4400 2754	0x4400 2954	0x4400 2B54
L3_TARG_STDERRLOG_INFO	R	32	0x0000 0058	0x4400 2658	0x4400 2758	0x4400 2958	0x4400 2B58
L3_TARG_STDERRLOG_SLVOFSLSB	R	32	0x0000 005C	0x4400 265C	0x4400 275C	0x4400 295C	0x4400 2B5C
L3_TARG_STDERRLOG_SLVOFSMSB	R	32	0x0000 0060	0x4400 2660	0x4400 2760	0x4400 2960	0x4400 2B60
L3_TARG_STDERRLOG_CUSTOMINFO_INFO	R	32	0x0000 0064	0x4400 2664	0x4400 2764	0x4400 2964	0x4400 2B64
L3_TARG_STDERRLOG_CUSTOMINFO_MSTADDR	R	32	0x0000 0068	0x4400 2668	0x4400 2768	0x4400 2968	0x4400 2B68
L3_TARG_STDERRLOG_CUSTOMINFO_OPCODE	R	32	0x0000 006C	0x4400 266C	0x4400 276C	0x4400 296C	0x4400 2B6C
L3_TARG_ADDRSPACESIZELOG	RW	32	0x0000 0080	0x4400 2680	0x4400 2780	0x4400 2980	0x4400 2B80

Table 9-95. L3_MAIN TARG Register Summary

Register Name	Type	Register Width (Bits)	Address Offset	TPTC1_TARG L3_MAIN Physical Address
L3_TARG_STDHOSTHDR_COREREG	R	32	0x0000 0000	0x4400 2E00
L3_TARG_STDHOSTHDR_VERSIONREG	R	32	0x0000 0004	0x4400 2E04
L3_TARG_STDHOSTHDR_MAINCTLREG	RW	32	0x0000 0008	0x4400 2E08
L3_TARG_STDHOSTHDR_NTTPADDR_0	R	32	0x0000 0010	0x4400 2E10
L3_TARG_STDERRLOG_SVRTSTDLVL	RW	32	0x0000 0040	0x4400 2E40
L3_TARG_STDERRLOG_SVRTCUSTOMLVL	RW	32	0x0000 0044	0x4400 2E44
L3_TARG_STDERRLOG_MAIN	RW	32	0x0000 0048	0x4400 2E48
L3_TARG_STDERRLOG_HDR	R	32	0x0000 004C	0x4400 2E4C
L3_TARG_STDERRLOG_MSTADDR	R	32	0x0000 0050	0x4400 2E50
L3_TARG_STDERRLOG_SLVADDR	R	32	0x0000 0054	0x4400 2E54
L3_TARG_STDERRLOG_INFO	R	32	0x0000 0058	0x4400 2E58
L3_TARG_STDERRLOG_SLVOFSLSB	R	32	0x0000 005C	0x4400 2E5C

Table 9-95. L3_MAIN TARG Register Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	TPTC1_TARG L3_MAIN Physical Address
L3_TARG_STDERRLOG_SLVOFSMSB	R	32	0x0000 0060	0x4400 2E60
L3_TARG_STDERRLOG_CUSTOMINFO_INFO	R	32	0x0000 0064	0x4400 2E64
L3_TARG_STDERRLOG_CUSTOMINFO_MSTADDR	R	32	0x0000 0068	0x4400 2E68
L3_TARG_STDERRLOG_CUSTOMINFO_OPCODE	R	32	0x0000 006C	0x4400 2E6C
L3_TARG_ADDRSPACESIZELOG	RW	32	0x0000 0080	0x4400 2E80

Table 9-96. L3_MAIN TARG Register Summary

Register Name	Type	Register Width (Bits)	Address Offset	QSPI_TARG L3_MAIN Physical Address	L3_INSTR_TARG L3_MAIN Physical Address	DEBUGSS_CT_TBR_TARG L3_MAIN Physical Address
L3_TARG_STDHOSTHDR_COREREG	R	32	0x0000 0000	0x4400 3900	0x4500 0100	0x4500 0300
L3_TARG_STDHOSTHDR_VERSIONREG	R	32	0x0000 0004	0x4400 3904	0x4500 0104	0x4500 0304
L3_TARG_STDHOSTHDR_MAINCTLREG	RW	32	0x0000 0008	0x4400 3908	0x4500 0108	0x4500 0308
L3_TARG_STDHOSTHDR_NTTPADDR_0	R	32	0x0000 0010	0x4400 3910	0x4500 0110	0x4500 0310
L3_TARG_STDERRLOG_SVRTSTDLVL	RW	32	0x0000 0040	0x4400 3940	0x4500 0140	0x4500 0340
L3_TARG_STDERRLOG_SVRTCUSTOMLVL	RW	32	0x0000 0044	0x4400 3944	0x4500 0144	0x4500 0344
L3_TARG_STDERRLOG_MAIN	RW	32	0x0000 0048	0x4400 3948	0x4500 0148	0x4500 0348
L3_TARG_STDERRLOG_HDR	R	32	0x0000 004C	0x4400 394C	0x4500 014C	0x4500 034C
L3_TARG_STDERRLOG_MSTADDR	R	32	0x0000 0050	0x4400 3950	0x4500 0150	0x4500 0350
L3_TARG_STDERRLOG_SLVADDR	R	32	0x0000 0054	0x4400 3954	0x4500 0154	0x4500 0354
L3_TARG_STDERRLOG_INFO	R	32	0x0000 0058	0x4400 3958	0x4500 0158	0x4500 0358
L3_TARG_STDERRLOG_SLVOFSLSB	R	32	0x0000 005C	0x4400 395C	0x4500 015C	0x4500 035C
L3_TARG_STDERRLOG_SLVOFSMSB	R	32	0x0000 0060	0x4400 3960	0x4500 0160	0x4500 0360
L3_TARG_STDERRLOG_CUSTOMINFO_INFO	R	32	0x0000 0064	0x4400 3964	0x4500 0164	0x4500 0364
L3_TARG_STDERRLOG_CUSTOMINFO_MSTADDR	R	32	0x0000 0068	0x4400 3968	0x4500 0168	0x4500 0368
L3_TARG_STDERRLOG_CUSTOMINFO_OPCODE	R	32	0x0000 006C	0x4400 396C	0x4500 016C	0x4500 036C
L3_TARG_ADDRSPACESIZELOG	RW	32	0x0000 0080	0x4400 3980	0x4500 0180	0x4500 0380

9.2.5.1.3.2 L3_MAIN TARG Register Description

Table 9-97. L3_TARG_STDHOSTHDR_COREREG

Address Offset	See Table 9-88 to Table 9-96.		
Physical Address	0x4400 0100	**Instance**	GPMC_TARG
	0x4400 0200		EMIF_P1_TARG
	0x4400 0300		DSP1_SDMA_TARG
	0x4400 0500		L4_CFG_TARG
	0x4400 0600		DSP2_SDMA_TARG
	0x4400 0700		CRC_TARG
	0x4400 0800		ISS_TARG
	0x4400 2F00		MCASP1_TARG
	0x4400 0A00		EVE_TARG
	0x4400 0F00		OCMC_RAM_TARG
	0x4400 1000		IPU_TARG
	0x4400 1300		TESOC_TARG
	0x4400 1D00		L4_WKUP_TARG
	0x4400 2000		EDMA_TPCC_TARG
	0x4400 2200		MMU_TARG
	0x4400 2900		DSS_TARG
	0x4400 2E00		TPTC1_TARG
	0x4400 3900		QSPI_TARG
	0x4500 0300		DEBUGSS_CT_TBR_TARG
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								STDHOSTHDR_COREREG_ CORECODE								RESERVED								STDHOSTHDR_COREREG_ VENDORCODE							

Bits	Field Name	Description	Type	Reset
31:22	RESERVED		R	0x000
21:16	STDHOSTHDR_COREREG_ CORECODE	The Core Code field is a constant reporting a vendor-specific core generator code. Type: Constant. Reset value: 0x13.	R	0x13
15:1	RESERVED		R	0x0000
0	STDHOSTHDR_COREREG_ VENDORCODE	The Vendor Code field is a constant reporting the core generator vendor code. Type: Constant. Reset value: 0x1. Read 0x0: Third-party vendor. Read 0x1:	R	1

Table 9-98. Register Call Summary for Register L3_TARG_STDHOSTHDR_COREREG

L3_MAIN Interconnect
- L3_MAIN TARG Register Summary and Description: [0][1][2][3][4][5][6][7][8]

Table 9-99. L3_TARG_STDHOSTHDR_VERSIONREG

Address Offset	See Table 9-88 to Table 9-96.		
Physical Address	0x4400 0104	**Instance**	GPMC_TARG
	0x4400 0204		EMIF_P1_TARG
	0x4400 0304		DSP1_SDMA_TARG
	0x4400 0504		L4_CFG_TARG
	0x4400 0604		DSP2_SDMA_TARG
	0x4400 0704		CRC_TARG
	0x4400 0804		ISS_TARG
	0x4400 2F04		MCASP1_TARG
	0x4400 0A04		EVE_TARG
	0x4400 0F04		OCMC_RAM_TARG
	0x4400 1004		IPU_TARG
	0x4400 1304		TESOC_TARG
	0x4400 1D04		L4_WKUP_TARG
	0x4400 2004		EDMA_TPCC_TARG
	0x4400 2204		MMU_TARG
	0x4400 2904		DSS_TARG
	0x4400 2E04		TPTC1_TARG
	0x4400 3904		QSPI_TARG
	0x4500 0304		DEBUGSS_CT_TBR_TARG
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STDHOSTHDR_VERSIONREG_REVISIONID								STDHOSTHDR_VERSIONREG_COREPARAMSCHECKSUM																							
R-0x00								R-0x000000																							

Bits	Field Name	Description	Type	Reset
31:24	STDHOSTHDR_VERSIONREG_REVISIONID	The Revision Identifier field is a constant reporting the core generator revision number. Type: Constant. Reset value: 0x0.	R	0x00
23:0	STDHOSTHDR_VERSIONREG_COREPARAMSCHECKSUM	Reserved. Type: Reserved. Reset value: Reserved.	R	0x000000

Table 9-100. Register Call Summary for Register L3_TARG_STDHOSTHDR_VERSIONREG

L3_MAIN Interconnect

- L3_MAIN TARG Register Summary and Description: [0][1][2][3][4][5][6][7][8]

Table 9-101. L3_TARG_STDHOSTHDR_MAINCTLREG

Address Offset	See Table 9-88 to Table 9-96.		
Physical Address	0x4400 0108	**Instance**	GPMC_TARG
	0x4400 0208		EMIF_P1_TARG
	0x4400 0308		DSP1_SDMA_TARG
	0x4400 0508		L4_CFG_TARG
	0x4400 0608		DSP2_SDMA_TARG
	0x4400 0708		CRC_TARG
	0x4400 0808		ISS_TARG
	0x4400 2F08		MCASP1_TARG
	0x4400 0A08		EVE_TARG
	0x4400 0F08		OCMC_RAM_TARG
	0x4400 1008		IPU_TARG
	0x4400 1308		TESOC_TARG
	0x4400 1D08		L4_WKUP_TARG
	0x4400 2008		EDMA_TPCC_TARG
	0x4400 2208		MMU_TARG
	0x4400 2908		DSS_TARG
	0x4400 2E08		TPTC1_TARG
	0x4400 3908		QSPI_TARG
	0x4500 0308		DEBUGSS_CT_TBR_TARG
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												STDHOSTHDR_MAINCTLREG_CM	STDHOSTHDR_MAINCTLREG_FLT	RESERVED	STDHOSTHDR_MAINCTLREG_EN

Bits	Field Name	Description	Type	Reset
31:4	RESERVED	Reserved	R	0x0000000
3	STDHOSTHDR_MAINCTLREG_CM	Reserved for internal testing. Must be set to 0. Type: Control. Reset value: 0x0.	R	0
2	STDHOSTHDR_MAINCTLREG_FLT	Asserted when a Fault condition is detected: if the unit includes Error Logging, Flt is asserted when the FltCnt register field indicates a Fault, and deasserted when FltCnt is reset. If no Error Logging is implemented, this bit becomes unit-dependent. In all cases, Flt bit and Flt pin (service network) have the same logical level. Type: Status. Reset value: X.	R	0
1	RESERVED	Reserved	R	0
0	STDHOSTHDR_MAINCTLREG_EN	Sets the global core enable. Note: A disabled master does not generate any NTTP requests, and a disabled slave replies with an error packet to any request it receives. Type: Control. Reset value: 0x1.	RW	1

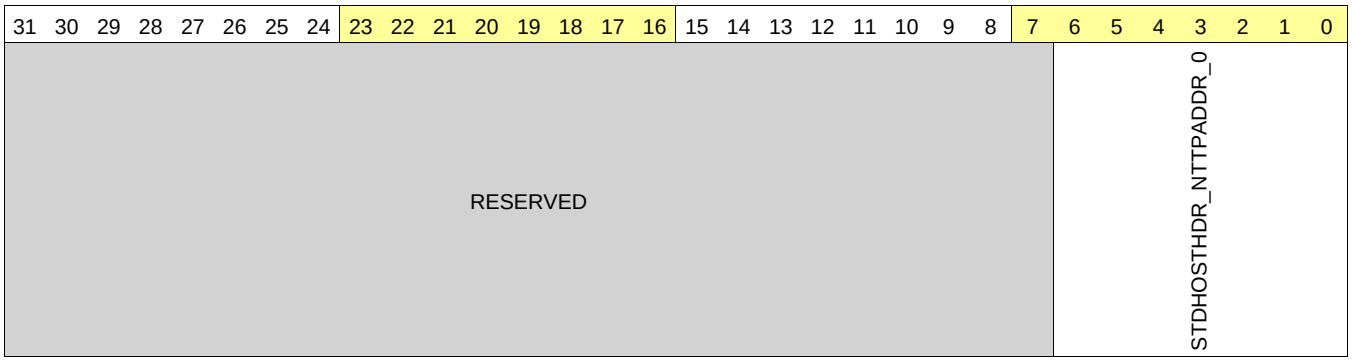
Table 9-102. Register Call Summary for Register L3_TARG_STDHOSTHDR_MAINCTLREG

L3_MAIN Interconnect

- L3_MAIN TARG Register Summary and Description: [0][1][2][3][4][5][6][7][8]

Table 9-103. L3_TARG_STDHOSTHDR_NTTPADDR_0

Address Offset	See Table 9-88 to Table 9-96.		
Physical Address	0x4400 0110	**Instance**	GPMC_TARG
	0x4400 0210		EMIF_P1_TARG
	0x4400 0310		DSP1_SDMA_TARG
	0x4400 0510		L4_CFG_TARG
	0x4400 0610		DSP2_SDMA_TARG
	0x4400 0710		CRC_TARG
	0x4400 0810		ISS_TARG
	0x4400 2F10		MCASP1_TARG
	0x4400 0A10		EVE_TARG
	0x4400 0F10		OCMC_RAM_TARG
	0x4400 1010		IPU_TARG
	0x4400 1310		TESOC_TARG
	0x4400 1D10		L4_WKUP_TARG
	0x4400 2010		EDMA_TPCC_TARG
	0x4400 2210		MMU_TARG
	0x4400 2910		DSS_TARG
	0x4400 2E10		TPTC1_TARG
	0x4400 3910		QSPI_TARG
	0x4500 0310		DEBUGSS_CT_TBR_TARG
Description			
Type	R		



Bits	Field Name	Description	Type	Reset
31:7	RESERVED	Reserved	R	0x0000000
6:0	STDHOSTHDR_NTTPADDR_0	Shows the Rx port address. Type: Control. Reset value: 0x15.	R	See (1)

(1) GPMC- 0xC; DSP1_SDMA - 0x4; L4_CFG - 0x14; DSP2_SDMA - 0x5; MCASP1 - 0x1F; EVE - 0x7; OCMC_RAM - 0x24; IPU - 0x10; L4_PER1_P1 - 0x16; L4_WKUP - 0x1E; L4_PER1_P2 - 0x17; EDMA_TPCC - 0x30; MMU - 0x22; L4_PER2_P1 - 0x18; L4_PER3_P1 - 0x1B; L4_PER3_P2 - 0x1C; DSS - 0x6; TPTC2 - 0x32; TPTC1 - 0x31; QSPI - 0x39; L3_INSTR- 0x19; DEBUGSS - 0x41; EMIF_P1 - 0x2; CRC - 0x8; ISS - 0x42; TESOC - 0x3.

Table 9-104. Register Call Summary for Register L3_TARG_STDHOSTHDR_NTTPADDR_0

L3_MAIN Interconnect

- L3_MAIN TARG Register Summary and Description: [0][1][2][3][4][5][6][7][8]

Table 9-105. L3_TARG_STDERRLOG_SVRTSTDLVL

Address Offset	See Table 9-88 to Table 9-96.		
Physical Address	0x4400 0140	**Instance**	GPMC_TARG
	0x4400 0240		EMIF_P1_TARG
	0x4400 0340		DSP1_SDMA_TARG
	0x4400 0540		L4_CFG_TARG
	0x4400 0640		DSP2_SDMA_TARG
	0x4400 0740		CRC_TARG
	0x4400 0840		ISS_TARG
	0x4400 2F40		MCASP1_TARG
	0x4400 0A40		EVE_TARG
	0x4400 0F40		OCMC_RAM_TARG
	0x4400 1040		IPU_TARG
	0x4400 1340		TESOC_TARG
	0x4400 1D40		L4_WKUP_TARG
	0x4400 2040		EDMA_TPCC_TARG
	0x4400 2240		MMU_TARG
	0x4400 2940		DSS_TARG
	0x4400 2E40		TPTC1_TARG
	0x4400 3940		QSPI_TARG
	0x4500 0340		DEBUGSS_CT_TBR_TARG
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														STDERRLOG_SVRTSTDLVL_0	

Bits	Field Name	Description	Type	Reset
31:2	RESERVED	Reserved	R	0x0000 0000
1:0	STDERRLOG_SVRTSTDLVL_0	Severity level parameters Type: Control. Reset value: 0x2.	RW	0x2
		0x0: Error logging is disabled.		
		0x1: Errors are logged with severity level Error.		
		0x2: Errors are logged with severity level Fault.		

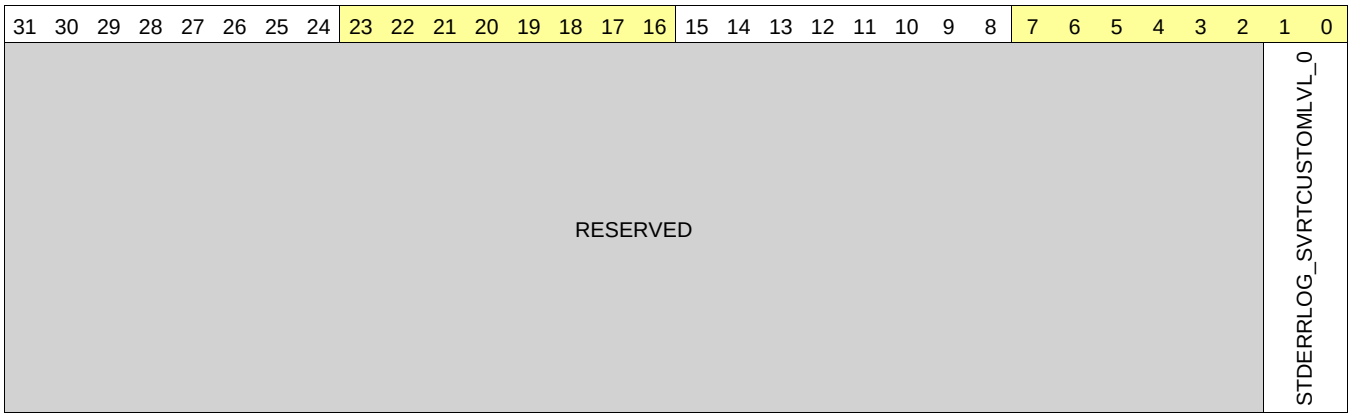
Table 9-106. Register Call Summary for Register L3_TARG_STDERRLOG_SVRTSTDLVL

L3_MAIN Interconnect

- Severity Level of Standard and Custom Errors: [0]
- L3_MAIN Interconnect Error Analysis Mode: [1]
- L3_MAIN TARG Register Summary and Description: [2][3][4][5][6][7][8][9][10]

Table 9-107. L3_TARG_STDERRLOG_SVRTCUSTOMLVL

Address Offset	See Table 9-88 to Table 9-96.		
Physical Address	0x4400 0144	**Instance**	GPMC_TARG
	0x4400 0244		EMIF_P1_TARG
	0x4400 0344		DSP1_SDMA_TARG
	0x4400 0544		L4_CFG_TARG
	0x4400 0644		DSP2_SDMA_TARG
	0x4400 0744		CRC_TARG
	0x4400 0844		ISS_TARG
	0x4400 2F44		MCASP1_TARG
	0x4400 0A44		EVE_TARG
	0x4400 0F44		OCMC_RAM_TARG
	0x4400 1044		IPU_TARG
	0x4400 1344		TESOC_TARG
	0x4400 1D44		L4_WKUP_TARG
	0x4400 2044		EDMA_TPCC_TARG
	0x4400 2244		MMU_TARG
	0x4400 2944		DSS_TARG
	0x4400 2E44		TPTC1_TARG
	0x4400 3944		QSPI_TARG
	0x4500 0344		DEBUGSS_CT_TBR_TARG
Description			
Type	RW		



Bits	Field Name	Description	Type	Reset
31:2	RESERVED	Reserved	R	0x0000 0000
1:0	STDERRLOG_SVRTCUSTOMLVL_0	Severity level parameters Type: Control. Reset value: 0x2.	RW	0x2
		0x0: Error logging is disabled.		
		0x1: Errors are logged with severity level Error.		
		0x2: Errors are logged with severity level Fault.		

Table 9-108. Register Call Summary for Register L3_TARG_STDERRLOG_SVRTCUSTOMLVL

L3_MAIN Interconnect

- Severity Level of Standard and Custom Errors: [0]
- L3_MAIN TARG Register Summary and Description: [1][2][3][4][5][6][7][8][9]

Table 9-109. L3_TARG_STDERRLOG_MAIN

Address Offset	See Table 9-88 to Table 9-96.		
Physical Address	0x4400 0148	**Instance**	GPMC_TARG
	0x4400 0248		EMIF_P1_TARG
	0x4400 0348		DSP1_SDMA_TARG
	0x4400 0548		L4_CFG_TARG
	0x4400 0648		DSP2_SDMA_TARG
	0x4400 0748		CRC_TARG
	0x4400 0848		ISS_TARG
	0x4400 2F48		MCASP1_TARG
	0x4400 0A48		EVE_TARG
	0x4400 0F48		OCMC_RAM_TARG
	0x4400 1048		IPU_TARG
	0x4400 1348		TESOC_TARG
	0x4400 1D48		L4_WKUP_TARG
	0x4400 2048		EDMA_TPCC_TARG
	0x4400 2248		MMU_TARG
	0x4400 2948		DSS_TARG
	0x4400 2E48		TPTC1_TARG
	0x4400 3948		QSPI_TARG
	0x4500 0348		DEBUGSS_CT_TBR_TARG
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STDERRLOG_MAIN_CLRLOG	RESERVED											STDERRLOG_MAIN_FLTCNT	STDERRLOG_MAIN_ERRCNT	RESERVED																	STDERRLOG_MAIN_ERRTYPE	STDERRLOG_MAIN_ERRLOGVLD
W	R																														
0	0x000											0	0	0x0000																	

Bits	Field Name	Description	Type	Reset
31	STDERRLOG_MAIN_CLRLOG	Clears "Error Logging Valid" bit when written to 1. Type: Give_AutoCleared. Reset value: 0x0.	RW	0
30:20	RESERVED	Reserved	R	0x000
19	STDERRLOG_MAIN_FLTCNT	Asserted when at least one error with severity level FAULT is detected. Reset when written to 1. Type: Take. Reset value: 0x0.	RW	0
18	STDERRLOG_MAIN_ERRCNT	Asserted when at least one error with severity level ERROR is detected. Reset when written to 1. Type: Take. Reset value: 0x0.	RW	0
17:2	RESERVED	Reserved	R	0x0000
1	STDERRLOG_MAIN_ERRTYPE	Indicates logging type. Type: Status. Reset value: X. Read 0x0: Logged Error format is standard (header and necker recorded). Read 0x1: Logged Error format is module dependent. CustomInfo register(s) may be implemented to store additional information.	R	0
0	STDERRLOG_MAIN_ERRLOGVLD	Error Logging Valid. Asserted when logging information is valid(indicates that an error has been logged). Type: Status. Reset value: X.	R	0

Table 9-110. Register Call Summary for Register L3_TARG_STDERRLOG_MAIN

L3_MAIN Interconnect

- Severity Level of Standard and Custom Errors: [0]
- Example for Decoding Standard/Custom Errors Logged in L3_MAIN: [1]
- L3_MAIN Interconnect Error Analysis Mode: [2][3][4][5][6][7]
- L3_MAIN TARG Register Summary and Description: [8][9][10][11][12][13][14][15][16]

Table 9-111. L3_TARG_STDERRLOG_HDR

Address Offset	See Table 9-88 to Table 9-96.		
Physical Address	0x4400 014C	**Instance**	GPMC_TARG
	0x4400 024C		EMIF_P1_TARG
	0x4400 034C		DSP1_SDMA_TARG
	0x4400 054C		L4_CFG_TARG
	0x4400 064C		DSP2_SDMA_TARG
	0x4400 074C		CRC_TARG
	0x4400 084C		ISS_TARG
	0x4400 2F4C		MCASP1_TARG
	0x4400 0A4C		EVE_TARG
	0x4400 0F4C		OCMC_RAM_TARG
	0x4400 104C		IPU_TARG
	0x4400 134C		TESOC_TARG
	0x4400 1D4C		L4_WKUP_TARG
	0x4400 204C		EDMA_TPCC_TARG
	0x4400 224C		MMU_TARG
	0x4400 294C		DSS_TARG
	0x4400 2E4C		TPTC1_TARG
	0x4400 394C		QSPI_TARG
	0x4500 034C		DEBUGSS_CT_TBR_TARG
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				STDERRLOG_HDR_LEN1										RESERVED		STDERRLOG_HDR_STOPOFSWRPSZ				STDERRLOG_HDR_ERR	RESERVED			STDERRLOG_HDR_PRESSURE		RESERVED		STDERRLOG_HDR_OPCODE			

Bits	Field Name	Description	Type	Reset
31:28	RESERVED	Reserved	R	0x00
27:18	STDERRLOG_HDR_LEN1	This field contains the number of payload cell(s) minus one of the logged packet. Type: Status. Reset value: X.	R	0x00
17:16	RESERVED	Reserved	R	0x0
15:12	STDERRLOG_HDR_STOPOFSWRPSZ	StopOfs or WrapSize field of the logged packet (meaning depends on Wrp bit of logged opcode). Type: Status. Reset value: X.	R	0x0
11	STDERRLOG_HDR_ERR	Err bit of the logged packet. Type: Status. Reset value: X.	R	0
10:8	RESERVED	Reserved	R	0x0
7:6	STDERRLOG_HDR_PRESSURE	Pressure field of the logged packet. Type: Status. Reset value: X.	R	0

Bits	Field Name	Description	Type	Reset
5:4	RESERVED	Reserved	R	0x0
3:0	STDERRLOG_HDR_OPCODE	Opcode of the logged packet. Type: Status. Reset value: X. 0x0: Store without acknowledge, incrementing burst, non-atomic request 0x1: Store without acknowledge, wrapping burst, non-atomic request 0x2: Store with acknowledge, incrementing burst, non-atomic request 0x3: Store with acknowledge, wrapping burst, non-atomic request 0x4: Load, incrementing burst, non-atomic request 0x5: Load, wrapping burst, non-atomic request 0x6: Control packet 0x7: Flush 0x8: Store without acknowledge, incrementing burst, atomic request 0x9: Store without acknowledge, wrapping burst, atomic request 0xA: Store with acknowledge, incrementing burst, atomic request 0xB: Store with acknowledge, wrapping burst, atomic request 0xC: Load, incrementing burst, atomic request 0xD: Load, wrapping burst, atomic request 0xE: Reserved 0xF: Reserved	R	0x0

Table 9-112. Register Call Summary for Register L3_TARG_STDERRLOG_HDR

L3_MAIN Interconnect

- Severity Level of Standard and Custom Errors: [0]
- Example for Decoding Standard/Custom Errors Logged in L3_MAIN: [1]
- L3_MAIN TARG Register Summary and Description: [2][3][4][5][6][7][8][9][10]

Table 9-113. L3_TARG_STDERRLOG_MSTADDR

Address Offset	See Table 9-88 to Table 9-96.		
Physical Address	0x4400 0150	**Instance**	GPMC_TARG
	0x4400 0250		EMIF_P1_TARG
	0x4400 0350		DSP1_SDMA_TARG
	0x4400 0550		L4_CFG_TARG
	0x4400 0650		DSP2_SDMA_TARG
	0x4400 0750		CRC_TARG
	0x4400 0850		ISS_TARG
	0x4400 2F50		MCASP1_TARG
	0x4400 0A50		EVE_TARG
	0x4400 0F50		OCMC_RAM_TARG
	0x4400 1050		IPU_TARG
	0x4400 1350		TESOC_TARG
	0x4400 1D50		L4_WKUP_TARG
	0x4400 2050		EDMA_TPCC_TARG
	0x4400 2250		MMU_TARG
	0x4400 2950		DSS_TARG
	0x4400 2E50		TPTC1_TARG
	0x4400 3950		QSPI_TARG
	0x4500 0350		DEBUGSS_CT_TBR_TARG
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								STDERRLOG_MSTADDR							

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Reserved	R	0x000000
7:0	STDERRLOG_MSTADDR	Master Address field of the logged packet. Type: Status. Reset value: X.	R	0x00

Table 9-114. Register Call Summary for Register L3_TARG_STDERRLOG_MSTADDR

L3_MAIN Interconnect

- Severity Level of Standard and Custom Errors: [0]
- Example for Decoding Standard/Custom Errors Logged in L3_MAIN: [1]
- L3_MAIN TARG Register Summary and Description: [2][3][4][5][6][7][8][9][10]

Table 9-115. L3_TARG_STDERRLOG_SLVADDR

Address Offset	See Table 9-88 to Table 9-96.		
Physical Address	0x4400 0154	**Instance**	GPMC_TARG
	0x4400 0254		EMIF_P1_TARG
	0x4400 0354		DSP1_SDMA_TARG
	0x4400 0554		L4_CFG_TARG
	0x4400 0654		DSP2_SDMA_TARG
	0x4400 0754		CRC_TARG
	0x4400 0854		ISS_TARG
	0x4400 2F54		MCASP1_TARG
	0x4400 0A54		EVE_TARG
	0x4400 0F54		OCMC_RAM_TARG
	0x4400 1054		IPU_TARG
	0x4400 1354		TESOC_TARG
	0x4400 1D54		L4_WKUP_TARG
	0x4400 2054		EDMA_TPCC_TARG
	0x4400 2254		MMU_TARG
	0x4400 2954		DSS_TARG
	0x4400 2E54		TPTC1_TARG
	0x4400 3954		QSPI_TARG
	0x4500 0354		DEBUGSS_CT_TBR_TARG
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								STDERRLOG_SLVADDR							

Bits	Field Name	Description	Type	Reset
31:7	RESERVED	Reserved	R	0x0000000
6:0	STDERRLOG_SLVADDR	Slave Address field of the logged packet. Type: Status. Reset value: X.	R	0x00

Table 9-116. Register Call Summary for Register L3_TARG_STDERRLOG_SLVADDR

L3_MAIN Interconnect

- Severity Level of Standard and Custom Errors: [0]
- Example for Decoding Standard/Custom Errors Logged in L3_MAIN: [1]
- L3_MAIN TARG Register Summary and Description: [2][3][4][5][6][7][8][9][10]

Table 9-117. L3_TARG_STDERRLOG_INFO

Address Offset	See Table 9-88 to Table 9-96.		
Physical Address	0x4400 0158	**Instance**	GPMC_TARG
	0x4400 0258		EMIF_P1_TARG
	0x4400 0358		DSP1_SDMA_TARG
	0x4400 0558		L4_CFG_TARG
	0x4400 0658		DSP2_SDMA_TARG
	0x4400 0758		CRC_TARG
	0x4400 0858		ISS_TARG
	0x4400 2F58		MCASP1_TARG
	0x4400 0A58		EVE_TARG
	0x4400 0F58		OCMC_RAM_TARG
	0x4400 1058		IPU_TARG
	0x4400 1358		TESOC_TARG
	0x4400 1D58		L4_WKUP_TARG
	0x4400 2058		EDMA_TPCC_TARG
	0x4400 2258		MMU_TARG
	0x4400 2958		DSS_TARG
	0x4400 2E58		TPTC1_TARG
	0x4400 3958		QSPI_TARG
	0x4500 0358		DEBUGSS_CT_TBR_TARG
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								STDERRLOG_INFO							

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Reserved	R	0x000000
7:0	STDERRLOG_INFO	Info field of the logged packet. Type: Status. Reset value: X.	R	0x00

Table 9-118. Register Call Summary for Register L3_TARG_STDERRLOG_INFO

L3_MAIN Interconnect

- Severity Level of Standard and Custom Errors: [0]
- Example for Decoding Standard/Custom Errors Logged in L3_MAIN: [1]
- L3_MAIN TARG Register Summary and Description: [2][3][4][5][6][7][8][9][10]

Table 9-119. L3_TARG_STDERRLOG_SLVOFSLSB

Address Offset	See Table 9-88 to Table 9-96.		
Physical Address	0x4400 015C	**Instance**	GPMC_TARG
	0x4400 025C		EMIF_P1_TARG
	0x4400 035C		DSP1_SDMA_TARG
	0x4400 055C		L4_CFG_TARG
	0x4400 065C		DSP2_SDMA_TARG
	0x4400 075C		CRC_TARG
	0x4400 085C		ISS_TARG
	0x4400 2F5C		MCASP1_TARG
	0x4400 0A5C		EVE_TARG
	0x4400 0F5C		OCMC_RAM_TARG
	0x4400 105C		IPU_TARG
	0x4400 135C		TESOC_TARG
	0x4400 1D5C		L4_WKUP_TARG
	0x4400 205C		EDMA_TPCC_TARG
	0x4400 225C		MMU_TARG
	0x4400 295C		DSS_TARG
	0x4400 2E5C		TPTC1_TARG
	0x4400 395C		QSPI_TARG
	0x4500 035C		DEBUGSS_CT_TBR_TARG
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STDERRLOG_SLVOFSLSB																															

Bits	Field Name	Description	Type	Reset
31:0	STDERRLOG_SLVOFSLSB	LSB of the "slave offset" field, concatenated with "start offset" field of the logged packet. Type: Status. Reset value: X.	R	0x0000 0000

Table 9-120. Register Call Summary for Register L3_TARG_STDERRLOG_SLVOFSLSB

L3_MAIN Interconnect

- L3_MAIN TARG Register Summary and Description: [0][1][2][3][4][5][6][7][8]

Table 9-121. L3_TARG_STDERRLOG_SLVOFSMSB

Address Offset	See Table 9-88 to Table 9-96.		
Physical Address	0x4400 0160	**Instance**	GPMC_TARG
	0x4400 0260		EMIF_P1_TARG
	0x4400 0360		DSP1_SDMA_TARG
	0x4400 0560		L4_CFG_TARG
	0x4400 0660		DSP2_SDMA_TARG
	0x4400 0760		CRC_TARG
	0x4400 0860		ISS_TARG
	0x4400 2F60		MCASP1_TARG
	0x4400 0A60		EVE_TARG
	0x4400 0F60		OCMC_RAM_TARG
	0x4400 1060		IPU_TARG
	0x4400 1360		TESOC_TARG
	0x4400 1D60		L4_WKUP_TARG
	0x4400 2060		EDMA_TPCC_TARG
	0x4400 2260		MMU_TARG
	0x4400 2960		DSS_TARG
	0x4400 2E60		TPTC1_TARG
	0x4400 3960		QSPI_TARG
	0x4500 0360		DEBUGSS_CT_TBR_TARG
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															STDERRLOG_SLVOFSMSB

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved	R	0x0000 0000
0	STDERRLOG_SLVOFSMSB	MSB of the "slave offset" field of the logged packet (according to NTTP packet format, this register field may exceed the actual "slave offset" size. Unused bits are stuck at 0, if any). Type: Status. Reset value: X.	R	0

Table 9-122. Register Call Summary for Register L3_TARG_STDERRLOG_SLVOFSMSB

L3_MAIN Interconnect

- L3_MAIN TARG Register Summary and Description: [0][1][2][3][4][5][6][7][8]

Table 9-123. L3_TARG_STDERRLOG_CUSTOMINFO_INFO

Address Offset	See Table 9-88 to Table 9-96.		
Physical Address	0x4400 0164	**Instance**	GPMC_TARG
	0x4400 0264		EMIF_P1_TARG
	0x4400 0364		DSP1_SDMA_TARG
	0x4400 0564		L4_CFG_TARG
	0x4400 0664		DSP2_SDMA_TARG
	0x4400 0764		CRC_TARG
	0x4400 0864		ISS_TARG
	0x4400 2F64		MCASP1_TARG
	0x4400 0A64		EVE_TARG
	0x4400 0F64		OCMC_RAM_TARG
	0x4400 1064		IPU_TARG
	0x4400 1364		TESOC_TARG
	0x4400 1D64		L4_WKUP_TARG
	0x4400 2064		EDMA_TPCC_TARG
	0x4400 2264		MMU_TARG
	0x4400 2964		DSS_TARG
	0x4400 2E64		TPTC1_TARG
	0x4400 3964		QSPI_TARG
	0x4500 0364		DEBUGSS_CT_TBR_TARG
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								STDERRLOG_CUSTOMINFO_INFO							

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Reserved	R	0x000000
7:0	STDERRLOG_CUSTOMINFO_INFO	Info field of the response packet. Type: Status. Reset value: X.	R	0x00

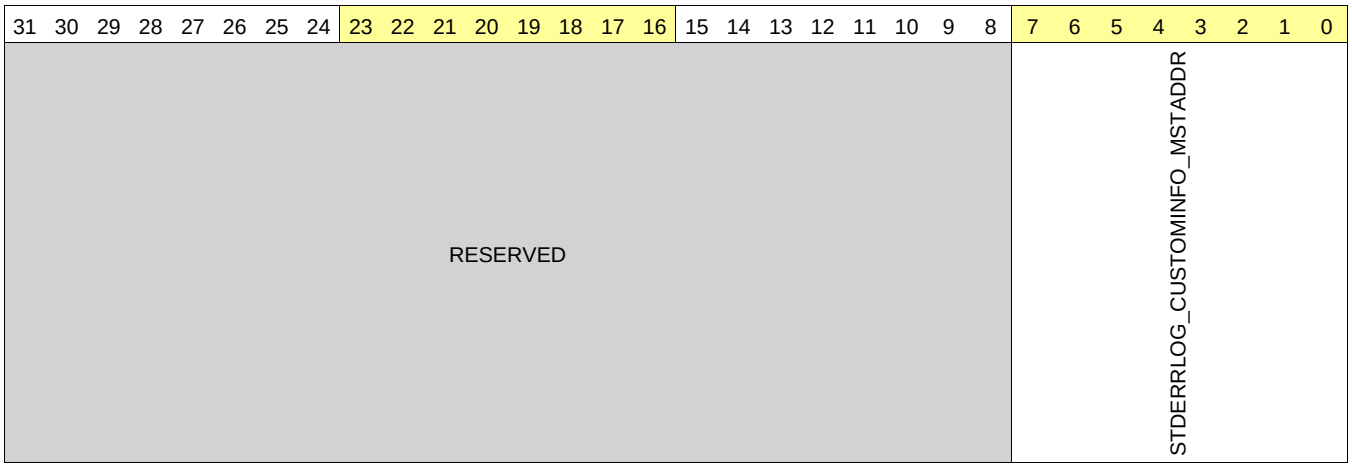
Table 9-124. Register Call Summary for Register L3_TARG_STDERRLOG_CUSTOMINFO_INFO

L3_MAIN Interconnect

- L3_MAIN Interconnect Error Analysis Mode: [0]
- L3_MAIN TARG Register Summary and Description: [1][2][3][4][5][6][7][8][9]

Table 9-125. L3_TARG_STDERRLOG_CUSTOMINFO_MSTADDR

Address Offset	See Table 9-88 to Table 9-96.		
Physical Address	0x4400 0168	**Instance**	GPMC_TARG
	0x4400 0268		EMIF_P1_TARG
	0x4400 0368		DSP1_SDMA_TARG
	0x4400 0568		L4_CFG_TARG
	0x4400 0668		DSP2_SDMA_TARG
	0x4400 0768		CRC_TARG
	0x4400 0868		ISS_TARG
	0x4400 2F68		MCASP1_TARG
	0x4400 0A68		EVE_TARG
	0x4400 0F68		OCMC_RAM_TARG
	0x4400 1068		IPU_TARG
	0x4400 1368		TESOC_TARG
	0x4400 1D68		L4_WKUP_TARG
	0x4400 2068		EDMA_TPCC_TARG
	0x4400 2268		MMU_TARG
	0x4400 2968		DSS_TARG
	0x4400 2E68		TPTC1_TARG
	0x4400 3968		QSPI_TARG
	0x4500 0368		DEBUGSS_CT_TBR_TARG
Description			
Type	R		



Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Reserved	R	0x000000
7:0	STDERRLOG_CUSTOMINFO_MSTADDR	MstAddr field of the response packet. Type: Status. Reset value: X.	R	0x00

Table 9-126. Register Call Summary for Register L3_TARG_STDERRLOG_CUSTOMINFO_MSTADDR

L3_MAIN Interconnect

- L3_MAIN Interconnect Error Analysis Mode: [0]
- L3_MAIN TARG Register Summary and Description: [1][2][3][4][5][6][7][8][9]

Table 9-127. L3_TARG_STDERRLOG_CUSTOMINFO_OPCODE

Address Offset	See Table 9-88 to Table 9-96.		
Physical Address	0x4400 016C	**Instance**	GPMC_TARG
	0x4400 026C		EMIF_P1_TARG
	0x4400 036C		DSP1_SDMA_TARG
	0x4400 056C		L4_CFG_TARG
	0x4400 066C		DSP2_SDMA_TARG
	0x4400 076C		CRC_TARG
	0x4400 086C		ISS_TARG
	0x4400 2F6C		MCASP1_TARG
	0x4400 0A6C		EVE_TARG
	0x4400 0F6C		OCMC_RAM_TARG
	0x4400 106C		IPU_TARG
	0x4400 136C		TESOC_TARG
	0x4400 1D6C		L4_WKUP_TARG
	0x4400 206C		EDMA_TPCC_TARG
	0x4400 226C		MMU_TARG
	0x4400 296C		DSS_TARG
	0x4400 2E6C		TPTC1_TARG
	0x4400 396C		QSPI_TARG
	0x4500 036C		DEBUGSS_CT_TBR_TARG
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														STDERRLOG_CUSTOMINFO_OPCODE	

Bits	Field Name	Description	Type	Reset
31:2	RESERVED	Reserved	R	0x0000 0000
1:0	STDERRLOG_CUSTOMINFO_OPCODE	Opcode of the response packet. Type: Status. Reset value: X. 0x0: Logged request is *Store without acknowledge*. 0x1: Logged request is *Store with acknowledge*. 0x2: Logged request is *Load*. 0x3: Reserved	R	0x0

Table 9-128. Register Call Summary for Register L3_TARG_STDERRLOG_CUSTOMINFO_OPCODE

L3_MAIN Interconnect

- L3_MAIN Interconnect Error Analysis Mode: [0]
- L3_MAIN TARG Register Summary and Description: [1][2][3][4][5][6][7][8][9]

Table 9-129. L3_TARG_ADDRSPACESIZELOG

Address Offset	See Table 9-88 to Table 9-96.		
Physical Address	0x4400 0180	**Instance**	GPMC_TARG
	0x4400 0280		EMIF_P1_TARG
	0x4400 0380		DSP1_SDMA_TARG
	0x4400 0580		L4_CFG_TARG
	0x4400 0680		DSP2_SDMA_TARG
	0x4400 0780		CRC_TARG
	0x4400 0880		ISS_TARG
	0x4400 2F80		MCASP1_TARG
	0x4400 0A80		EVE_TARG
	0x4400 0F80		OCMC_RAM_TARG
	0x4400 1080		IPU_TARG
	0x4400 1380		TESOC_TARG
	0x4400 1D80		L4_WKUP_TARG
	0x4400 2080		EDMA_TPCC_TARG
	0x4400 2280		MMU_TARG
	0x4400 2980		DSS_TARG
	0x4400 2E80		TPTC1_TARG
	0x4400 3980		QSPI_TARG
	0x4500 0380		DEBUGSS_CT_TBR_TARG
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								ADDRSPACESIZELOG							

Bits	Field Name	Description	Type	Reset
31:5	RESERVED	Reserved	R	0x0000000
4:0	ADDRSPACESIZELOG	The address space size is equal to 2**AddrSpaceSizeLog * 4K in bytes. Type: Control. Reset value: 0x1F.	RW	0x1F

Table 9-130. Register Call Summary for Register L3_TARG_ADDRSPACESIZELOG

L3_MAIN Interconnect

- L3_MAIN TARG Register Summary and Description: [0][1][2][3][4][5][6][7][8]

9.2.5.1.4 L3_MAIN Flag Muxing Registers Summary and Description

Table 9-131. Flag Muxing Instance Summary

Module Name	Base Address	Size
CLK1_FLAGMUX_CLK1_1	0x4480 3500	4KiB
CLK1_FLAGMUX_CLK1_2	0x4480 3600	4KiB
CLK2_FLAGMUX_CLK2_1	0x4500 0200	4KiB

9.2.5.1.4.1 L3_MAIN Flag Muxing Registers Summary

Table 9-132. Flag Muxing Registers Summary

Register Name	Type	Register Width (Bits)	Address Offset	CLK1_FLAGMUX_CLK1_1 L3_MAIN Physical Address	CLK1_FLAGMUX_CLK1_2 L3_MAIN Physical Address	CLK2_FLAGMUX_CLK2_1 L3_MAIN Physical Address
L3_FLAGMUX_STDHOSTHDR_COREREG	R	32	0x0000 0000	0x4480 3500	0x4480 3600	0x4500 0200
L3_FLAGMUX_STDHOSTHDR_VERSIONREG	R	32	0x0000 0004	0x4480 3504	0x4480 3604	0x4500 0204
L3_FLAGMUX_MASK0	RW	32	0x0000 0008	0x4480 3508	0x4480 3608	0x4500 0208
L3_FLAGMUX_REGERR0	R	32	0x0000 000C	0x4480 350C	0x4480 360C	0x4500 020C
L3_FLAGMUX_MASK1	RW	32	0x0000 0010	0x4480 3510	0x4480 3610	0x4500 0210
L3_FLAGMUX_REGERR1	R	32	0x0000 0014	0x4480 3514	0x4480 3614	0x4500 0214

9.2.5.1.4.2 L3_MAIN Flag Muxing Registers Description

Table 9-133. L3_FLAGMUX_STDHOSTHDR_COREREG

Address Offset	See Table 9-132		
Physical Address	0x4480 3500 0x4480 3600 0x4500 0200	**Instance**	CLK1_FLAGMUX_CLK1_1 CLK1_FLAGMUX_CLK1_2 CLK2_FLAGMUX_CLK2_1
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								STDHOSTHDR_COREREG_CORECODE								RESERVED								STDHOSTHDR_COREREG_VENDORCODE							

Bits	Field Name	Description	Type	Reset
31:22	RESERVED	Reserved	R	0x000
21:16	STDHOSTHDR_COREREG_CORECODE	The Core Code field is a constant reporting a vendor-specific core generator code. Type: Constant. Reset value: 0x37.	R	0x37
15:1	RESERVED	Reserved	R	0x0000

Bits	Field Name	Description	Type	Reset
0	STDHOSTHDR_COREREG_VENDORCODE	The Vendor Code field is a constant reporting the core generator vendor code. Type: Constant. Reset value: 0x1. Read 0x0: Third-party vendor. Read 0x1:	R	1

Table 9-134. Register Call Summary for Register L3_FLAGMUX_STDHOSTHDR_COREREG

L3_MAIN Interconnect
- Flag Muxing: [0]
- L3_MAIN Flag Muxing Registers Summary and Description: [1]

Table 9-135. L3_FLAGMUX_STDHOSTHDR_VERSIONREG

Address Offset	See Table 9-132		
Physical Address	0x4480 3504 0x4480 3604 0x4500 0204	**Instance**	CLK1_FLAGMUX_CLK1_1 CLK1_FLAGMUX_CLK1_2 CLK2_FLAGMUX_CLK2_1
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STDHOSTHDR_VERSIONREG_REVISIONID								STDHOSTHDR_VERSIONREG_COREPARAMSCHECKSUM																							

Bits	Field Name	Description	Type	Reset
31:24	STDHOSTHDR_VERSIONREG_REVISIONID	The Revision Identifier field is a constant reporting the core generator revision number. Type: Constant. Reset value: 0x0.	R	0x00
23:0	STDHOSTHDR_VERSIONREG_COREPARAMSCHECKSUM	Reserved. Type: Reserved. Reset value: Reserved.	R	0x000000

Table 9-136. Register Call Summary for Register L3_FLAGMUX_STDHOSTHDR_VERSIONREG

L3_MAIN Interconnect
- Flag Muxing: [0]
- L3_MAIN Flag Muxing Registers Summary and Description: [1]

Table 9-137. L3_FLAGMUX_MASK0

Address Offset	See Table 9-132		
Physical Address	0x4480 3508 0x4480 3608 0x4500 0208	**Instance**	CLK1_FLAGMUX_CLK1_1 CLK1_FLAGMUX_CLK1_2 CLK2_FLAGMUX_CLK2_1
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								MASK0																							

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	See (1)
19:0 (2)	MASK0	Mask flag inputs 0 Type: Control.	RW	See (1)

(1) For CLK1_1 reset is 0x3FFFF; for CLK1_2 reset is 0xFFFFF; for CLK2 reset is 0xF

(2) For CLK1_1 bit field is [17:0]; for CLK1_2 bit field is [19:0]; for CLK2 bit field is [3:0]

Table 9-138. Register Call Summary for Register L3_FLAGMUX_MASK0

L3_MAIN Interconnect

- Flag Muxing: [0][1][2][3]
- L3_MAIN Interconnect Error Analysis Mode: [4]
- L3_MAIN Flag Muxing Registers Summary and Description: [5]

Table 9-139. L3_FLAGMUX_REGERR0

Address Offset	See Table 9-132		
Physical Address	0x4480 350C 0x4480 360C 0x4500 020C	**Instance**	CLK1_FLAGMUX_CLK1_1 CLK1_FLAGMUX_CLK1_2 CLK2_FLAGMUX_CLK2_1
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REGERR0																															

Bits	Field Name	Description	Type	Reset
31:0	REGERR0	Flag inputs 0 Type: Status. Reset value: X.	R	0x00000

Table 9-140. Register Call Summary for Register L3_FLAGMUX_REGERR0

L3_MAIN Interconnect

- Flag Muxing: [0][1][2][3]
- L3_MAIN Interconnect Error Analysis Mode: [4][5][6]
- L3_MAIN Flag Muxing Registers Summary and Description: [7]

Table 9-141. L3_FLAGMUX_MASK1

Address Offset	See Table 9-132		
Physical Address	0x4480 3510 0x4480 3610 0x4500 0210	**Instance**	CLK1_FLAGMUX_CLK1_1 CLK1_FLAGMUX_CLK1_2 CLK2_FLAGMUX_CLK2_1
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								MASK1																							

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	See (1)
19:0 (2)	MASK1	Mask flag inputs 1 Type: Control. Reset value: 0x7FFFFF.	RW	See (1)

(1) For CLK1_1 reset is 0x3FFFF; for CLK1_2 reset is 0xFFFFF; for CLK2 reset is 0xF
(2) For CLK1_1 bit field is [17:0]; for CLK1_2 bit field is [19:0]; for CLK2 bit field is [3:0]

Table 9-142. Register Call Summary for Register L3_FLAGMUX_MASK1

L3_MAIN Interconnect
- Flag Muxing: [0]
- L3_MAIN Interconnect Error Analysis Mode: [1]
- L3_MAIN Flag Muxing Registers Summary and Description: [2]

Table 9-143. L3_FLAGMUX_REGERR1

Address Offset	See Table 9-132		
Physical Address	0x4480 3514 0x4480 3614 0x4500 0214	**Instance**	CLK1_FLAGMUX_CLK1_1 CLK1_FLAGMUX_CLK1_2 CLK2_FLAGMUX_CLK2_1
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REGERR1																															

Bits	Field Name	Description	Type	Reset
31:0	REGERR1	Flag inputs 1 Type: Status. Reset value: X.	R	0x00000

Table 9-144. Register Call Summary for Register L3_FLAGMUX_REGERR1

L3_MAIN Interconnect
- Flag Muxing: [0]
- L3_MAIN Interconnect Error Analysis Mode: [1][2][3]
- L3_MAIN Flag Muxing Registers Summary and Description: [4]

9.2.5.1.5 L3_MAIN FLAGMUX CLK1MERGE Registers Summary and Description

Table 9-145. FLAGMUX CLK1MERGE Instance Summary

Module Name	Base Address	Size
CLK1_FLAGMUX_CLK1MERGE	0x4480 0400	4KiB

9.2.5.1.5.1 L3_MAIN FLAGMUX CLK1MERGE Registers Summary

Table 9-146. FLAGMUX CLK1MERGE Registers Summary

Register Name	Type	Register Width (Bits)	Address Offset	CLK1_FLAGMUX_CLK1MERGE L3_MAIN Physical Address
L3_FLAGMUX_CLK1MERGE_STDHOSTHDR_COREREG	R	32	0x0080 0000	0x4480 0400
L3_FLAGMUX_CLK1MERGE_STDHOSTHDR_VERSIONREG	R	32	0x0080 0004	0x4480 0404
L3_FLAGMUX_CLK1MERGE_MASK0	RW	32	0x0080 0008	0x4480 0408
L3_FLAGMUX_CLK1MERGE_REGERR0	R	32	0x0080 000C	0x4480 040C
L3_FLAGMUX_CLK1MERGE_MASK1	RW	32	0x0080 0010	0x4480 0410
L3_FLAGMUX_CLK1MERGE_REGERR1	R	32	0x0080 0014	0x4480 0414

9.2.5.1.5.2 L3_MAIN FLAGMUX CLK1MERGE Registers Description

Table 9-147. L3_FLAGMUX_CLK1MERGE_STDHOSTHDR_COREREG

Address Offset	See Table 9-146.		
Physical Address	0x4480 0400	**Instance**	CLK1_FLAGMUX_CLK1MERGE
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								STDHOSTHDR_COREREG_CORECODE								RESERVED								STDHOSTHDR_COREREG_VENDORCODE							

Bits	Field Name	Description	Type	Reset
31:22	RESERVED	Reserved	R	0x000
21:16	STDHOSTHDR_COREREG_CORECODE	The Core Code field is a constant reporting a vendor-specific core generator code. Type: Constant. Reset value: 0x37.	R	0x37
15:1	RESERVED	Reserved	R	0x0000
0	STDHOSTHDR_COREREG_VENDORCODE	The Vendor Code field is a constant reporting the core generator vendor code. Type: Constant. Reset value: 0x1. Read 0x0: Third-party vendor. Read 0x1:	R	1

Table 9-148. Register Call Summary for Register L3_FLAGMUX_CLK1MERGE_STDHOSTHDR_COREREG

L3_MAIN Interconnect

- Flag Muxing: [0]
- L3_MAIN FLAGMUX CLK1MERGE Registers Summary and Description: [1]

Table 9-149. L3_FLAGMUX_CLK1MERGE_STDHOSTHDR_VERSIONREG

Address Offset	See Table 9-146.		
Physical Address	0x4480 0404	**Instance**	CLK1_FLAGMUX_CLK1MERGE
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STDHOSTHDR_VERSIONREG_REVISIONID								STDHOSTHDR_VERSIONREG_COREPARAMSCHECKSUM																									

Bits	Field Name	Description	Type	Reset
31:24	STDHOSTHDR_VERSIONREG_REVISIONID	The Revision Identifier field is a constant reporting the core generator revision number. Type: Constant. Reset value: 0x0.	R	0x00
23:0	STDHOSTHDR_VERSIONREG_COREPARAMSCHECKSUM	Reserved. Type: Reserved. Reset value: Reserved.	R	0x000000

Table 9-150. Register Call Summary for Register L3_FLAGMUX_CLK1MERGE_STDHOSTHDR_VERSIONREG

L3_MAIN Interconnect

- Flag Muxing: [0]
- L3_MAIN FLAGMUX CLK1MERGE Registers Summary and Description: [1]

Table 9-151. L3_FLAGMUX_CLK1MERGE_MASK0

Address Offset	See Table 9-146.		
Physical Address	0x4480 0408	**Instance**	CLK1_FLAGMUX_CLK1MERGE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														MASK0	

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1:0	MASK0	Mask flag inputs 0 Type: Control. Reset value: 0x3	RW	0x3

Table 9-152. Register Call Summary for Register L3_FLAGMUX_CLK1MERGE_MASK0

L3_MAIN Interconnect
- Flag Muxing: [0][1]
- L3_MAIN FLAGMUX CLK1MERGE Registers Summary and Description: [2]

Table 9-153. L3_FLAGMUX_CLK1MERGE_REGERR0

Address Offset	See Table 9-146.		
Physical Address	0x4480 040C	**Instance**	CLK1_FLAGMUX_CLK1MERGE
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														REGERR0	

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1:0	REGERR0	Flag inputs 0 Type: Control. Reset value: X	RW	0x0

Table 9-154. Register Call Summary for Register L3_FLAGMUX_CLK1MERGE_REGERR0

L3_MAIN Interconnect
- Flag Muxing: [0][1]
- L3_MAIN FLAGMUX CLK1MERGE Registers Summary and Description: [2]

Table 9-155. L3_FLAGMUX_CLK1MERGE_MASK1

Address Offset	See Table 9-146.		
Physical Address	0x4480 0410	**Instance**	CLK1_FLAGMUX_CLK1MERGE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														MASK1	

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1:0	MASK1	Mask flag inputs 0 Type: Control. Reset value: 0x3	RW	0x3

Table 9-156. Register Call Summary for Register L3_FLAGMUX_CLK1MERGE_MASK1

L3_MAIN Interconnect
- Flag Muxing: [0]
- L3_MAIN FLAGMUX CLK1MERGE Registers Summary and Description: [1]

Table 9-157. L3_FLAGMUX_CLK1MERGE_REGERR1

Address Offset	See Table 9-146.		
Physical Address	0x4480 0414	**Instance**	CLK1_FLAGMUX_CLK1MERGE
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														REGERR1	

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1:0	REGERR1	Flag inputs 0 Type: Control. Reset value: X	RW	0x0

Table 9-158. Register Call Summary for Register L3_FLAGMUX_CLK1MERGE_REGERR1

L3_MAIN Interconnect
- Flag Muxing: [0]
- L3_MAIN FLAGMUX CLK1MERGE Registers Summary and Description: [1]

9.2.5.1.6 L3_MAIN Time-out Flag Muxing Registers Summary and Description

Table 9-159. Time-out Flag Muxing Instance Summary

Module Name	Base Address	Size
CLK1_FLAGMUX_CLK1	0x4480 5700	4KiB
CLK2_FLAGMUX_CLK2	0x4500 0400	4KiB

9.2.5.1.6.1 L3_MAIN Time-out Flag Muxing Registers Summary

Table 9-160. Time-out Flag Muxing Registers Summary

Register Name	Type	Register Width (Bits)	Address Offset	CLK1_FLAGMUX_CLK1 L3_MAIN Physical Address	CLK2_FLAGMUX_CLK2 L3_MAIN Physical Address
L3_FLAGMUX_TIMEOUT_STDHOSTHDR_COREREG	R	32	0x0000 0000	N/A	0x4500 0400
L3_FLAGMUX_TIMEOUT_STDHOSTHDR_VERSIONREG	R	32	0x0000 0004	N/A	0x4500 0404
L3_FLAGMUX_TIMEOUT_MASK0	RW	32	0x0000 0008	N/A	0x4500 0408
L3_FLAGMUX_TIMEOUT_REGERR0	R	32	0x0000 000C	N/A	0x4500 040C
L3_FLAGMUX_TIMEOUT1_STDHOSTHDR_COREREG	R	32	0x0000 0000	0x4480 5700	N/A
L3_FLAGMUX_TIMEOUT1_STDHOSTHDR_VERSIONREG	R	32	0x0000 0004	0x4480 5704	N/A
L3_FLAGMUX_TIMEOUT1_MASK0	RW	32	0x0000 0008	0x4480 5708	N/A
L3_FLAGMUX_TIMEOUT1_REGERR0	R	32	0x0000 000C	0x4480 570C	N/A
L3_FLAGMUX_TIMEOUT2_STDHOSTHDR_COREREG	R	32	0x0000 0100	0x4480 5800	N/A
L3_FLAGMUX_TIMEOUT2_STDHOSTHDR_VERSIONREG	R	32	0x0000 0104	0x4480 5804	N/A
L3_FLAGMUX_TIMEOUT2_MASK0	RW	32	0x0000 0108	0x4480 5808	N/A
L3_FLAGMUX_TIMEOUT2_REGERR0	R	32	0x0000 010C	0x4480 580C	N/A

9.2.5.1.6.2 L3_MAIN Time-out Flag Muxing Registers Description

Table 9-161. L3_FLAGMUX_TIMEOUT_STDHOSTHDR_COREREG

Address Offset	See Table 9-160.		
Physical Address	0x4500 0400	**Instance**	CLK2_FLAGMUX_CLK2
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								STDHOSTHDR_COREREG_CORECODE								RESERVED								STDHOSTHDR_COREREG_VENDORCODE							

Bits	Field Name	Description	Type	Reset
31:22	RESERVED		R	0x-
21:16	STDHOSTHDR_COREREG_CORECODE	The Core Code field is a constantreporting a vendor-specific core generator code. Type: Constant. Reset value: 0x37.	R	0x37
15:1	RESERVED		R	0x-
0	STDHOSTHDR_COREREG_VENDORCODE	The Vendor Code field is a constantreporting the core generator vendor code. Type: Constant. Reset value: 0x1. Read 0x1: Read 0x0: Third-party vendor.	R	1

Table 9-162. Register Call Summary for Register L3_FLAGMUX_TIMEOUT_STDHOSTHDR_COREREG

L3_MAIN Interconnect

- L3_MAIN Time-out Flag Muxing Registers Summary and Description: [0]

Table 9-163. L3_FLAGMUX_TIMEOUT_STDHOSTHDR_VERSIONREG

Address Offset	See Table 9-160.		
Physical Address	0x4500 0404	**Instance**	CLK2_FLAGMUX_CLK2
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STDHOSTHDR_VERSIONREG_REVISIONID								STDHOSTHDR_VERSIONREG_COREPARAMSCHECKSUM																							

Bits	Field Name	Description	Type	Reset
31:24	STDHOSTHDR_VERSIONREG_REVISIONID	The Revision Identifier field is a constantreporting the core generator revision number. Type: Constant. Reset value: 0x0.	R	0x0
23:0	STDHOSTHDR_VERSIONREG_COREPARAMSCHECKSUM	Reserved. Type: Reserved. Reset value: Reserved.	R	0x-

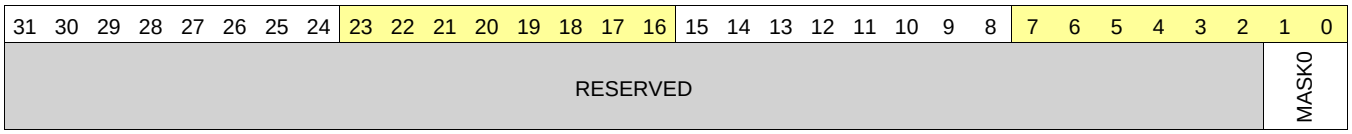
Table 9-164. Register Call Summary for Register L3_FLAGMUX_TIMEOUT_STDHOSTHDR_VERSIONREG

L3_MAIN Interconnect

- L3_MAIN Time-out Flag Muxing Registers Summary and Description: [0]

Table 9-165. L3_FLAGMUX_TIMEOUT_MASK0

Address Offset	See Table 9-160.		
Physical Address	0x4500 0408	**Instance**	CLK2_FLAGMUX_CLK2
Description			
Type	RW		



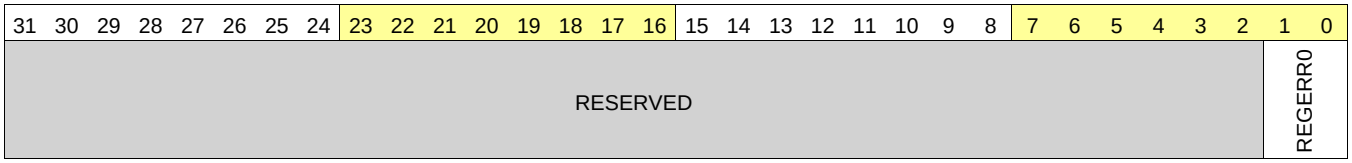
Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x-
1:0	MASK0	mask flag inputs 0 Type: Control.	RW	0x3

Table 9-166. Register Call Summary for Register L3_FLAGMUX_TIMEOUT_MASK0

L3_MAIN Interconnect
- Time-out Flag Muxing: [0]
- L3_MAIN Interconnect Error Analysis Mode: [1]
- L3_MAIN Time-out Flag Muxing Registers Summary and Description: [2]

Table 9-167. L3_FLAGMUX_TIMEOUT_REGERR0

Address Offset	See Table 9-160.		
Physical Address	0x4500 040C	**Instance**	CLK2_FLAGMUX_CLK2
Description			
Type	R		



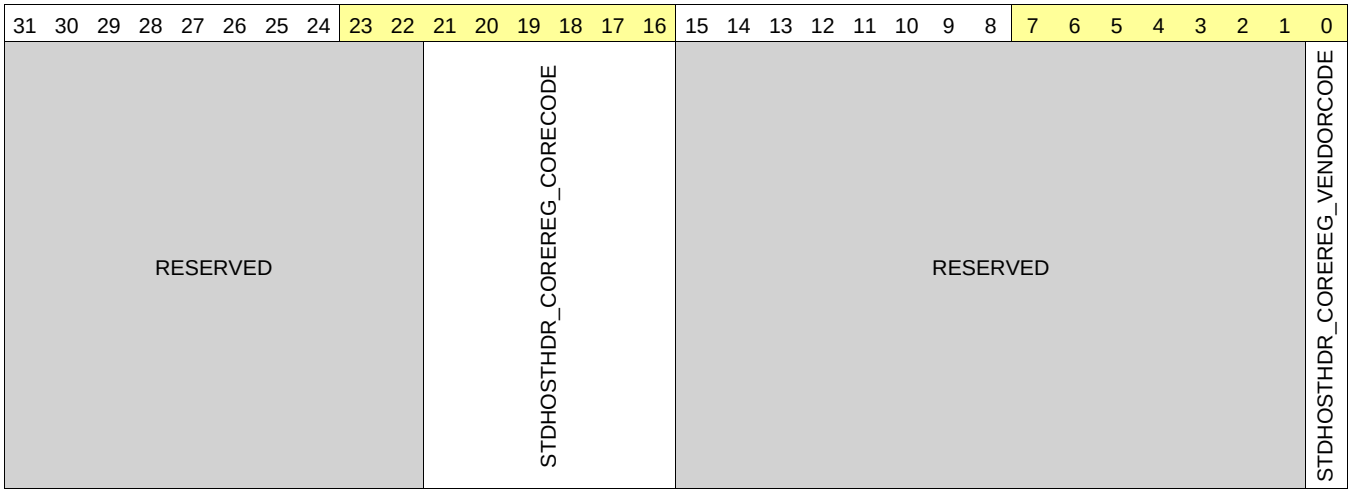
Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x-
1:0	REGERR0	flag inputs 0 Type: Status.	R	0x-

Table 9-168. Register Call Summary for Register L3_FLAGMUX_TIMEOUT_REGERR0

L3_MAIN Interconnect
- Time-out Flag Muxing: [0]
- L3_MAIN Interconnect Error Analysis Mode: [1]
- L3_MAIN Time-out Flag Muxing Registers Summary and Description: [2]

Table 9-169. L3_FLAGMUX_TIMEOUT1_STDHOSTHDR_COREREG

Address Offset	See Table 9-160.		
Physical Address	0x4480 5700	**Instance**	CLK1_FLAGMUX_CLK1
Description			
Type	R		



Bits	Field Name	Description	Type	Reset
31:22	RESERVED		R	0x-
21:16	STDHOSTHDR_COREREG_CORECODE	The Core Code field is a constantreporting a vendor-specific core generator code. Type: Constant. Reset value: 0x37.	R	0x37
15:1	RESERVED		R	0x-
0	STDHOSTHDR_COREREG_VENDORCODE	The Vendor Code field is a constantreporting the core generator vendor code. Type: Constant. Reset value: 0x1. Read 0x1: Read 0x0: Third-party vendor.	R	1

Table 9-170. Register Call Summary for Register L3_FLAGMUX_TIMEOUT1_STDHOSTHDR_COREREG

L3_MAIN Interconnect

- L3_MAIN Time-out Flag Muxing Registers Summary and Description: [0]

Table 9-171. L3_FLAGMUX_TIMEOUT1_STDHOSTHDR_VERSIONREG

Address Offset	See Table 9-160.		
Physical Address	0x4480 5704	**Instance**	CLK1_FLAGMUX_CLK1
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STDHOSTHDR_VERSIONREG_REVISIONID								STDHOSTHDR_VERSIONREG_COREPARAMSCHECKSUM																									

Bits	Field Name	Description	Type	Reset
31:24	STDHOSTHDR_VERSIONREG_REVISIONID	The Revision Identifier field is a constantreporting the core generator revision number. Type: Constant. Reset value: 0x0.	R	0x0
23:0	STDHOSTHDR_VERSIONREG_COREPARAMSCHECKSUM	Reserved. Type: Reserved. Reset value: Reserved.	R	0x-

Table 9-172. Register Call Summary for Register L3_FLAGMUX_TIMEOUT1_STDHOSTHDR_VERSIONREG

L3_MAIN Interconnect

- L3_MAIN Time-out Flag Muxing Registers Summary and Description: [0]

Table 9-173. L3_FLAGMUX_TIMEOUT1_MASK0

Address Offset	See Table 9-160.		
Physical Address	0x4480 5708	**Instance**	CLK1_FLAGMUX_CLK1
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																MASK0															

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x-
17:0	MASK0	mask flag inputs 0 Type: Control.	RW	0x3FFFF

Table 9-174. Register Call Summary for Register L3_FLAGMUX_TIMEOUT1_MASK0

L3_MAIN Interconnect

- Time-out Flag Muxing: [0][1]
- L3_MAIN Time-out Flag Muxing Registers Summary and Description: [2]

Table 9-175. L3_FLAGMUX_TIMEOUT1_REGERR0

Address Offset	See Table 9-160.		
Physical Address	0x4480 570C	**Instance**	CLK1_FLAGMUX_CLK1
Description			

Table 9-175. L3_FLAGMUX_TIMEOUT1_REGERR0 (continued)

Type	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																REGERR0															

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x-
17:0	REGERR0	flag inputs 0 Type: Status.	R	0x-

Table 9-176. Register Call Summary for Register L3_FLAGMUX_TIMEOUT1_REGERR0

L3_MAIN Interconnect

- Time-out Flag Muxing: [0][1]
- L3_MAIN Time-out Flag Muxing Registers Summary and Description: [2]

Table 9-177. L3_FLAGMUX_TIMEOUT2_STDHOSTHDR_COREREG

Address Offset	See Table 9-160.		
Physical Address	0x4480 5800	**Instance**	CLK1_FLAGMUX_CLK1
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								STDHOSTHDR_COREREG_CORECODE								RESERVED															STDHOSTHDR_COREREG_VENDORCODE

Bits	Field Name	Description	Type	Reset
31:22	RESERVED		R	0x-
21:16	STDHOSTHDR_COREREG_CORECODE	The Core Code field is a constantreporting a vendor-specific core generator code. Type: Constant. Reset value: 0x37.	R	0x37
15:1	RESERVED		R	0x-
0	STDHOSTHDR_COREREG_VENDORCODE	The Vendor Code field is a constantreporting the core generator vendor code. Type: Constant. Reset value: 0x1. Read 0x1: Read 0x0: Third-party vendor.	R	1

Table 9-178. Register Call Summary for Register L3_FLAGMUX_TIMEOUT2_STDHOSTHDR_COREREG

L3_MAIN Interconnect

- L3_MAIN Time-out Flag Muxing Registers Summary and Description: [0]

Table 9-179. L3_FLAGMUX_TIMEOUT2_STDHOSTHDR_VERSIONREG

Address Offset	See Table 9-160.		
Physical Address	0x4480 5804	**Instance**	CLK1_FLAGMUX_CLK1
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STDHOSTHDR_VERSIONREG_REVISIONID								STDHOSTHDR_VERSIONREG_COREPARAMSCHECKSUM																							

Bits	Field Name	Description	Type	Reset
31:24	STDHOSTHDR_VERSIONREG_REVISIONID	The Revision Identifier field is a constantreporting the core generator revision number. Type: Constant. Reset value: 0x0.	R	0x0
23:0	STDHOSTHDR_VERSIONREG_COREPARAMSCHECKSUM	Reserved. Type: Reserved. Reset value: Reserved.	R	0x-

Table 9-180. Register Call Summary for Register L3_FLAGMUX_TIMEOUT2_STDHOSTHDR_VERSIONREG

L3_MAIN Interconnect

- L3_MAIN Time-out Flag Muxing Registers Summary and Description: [0]

Table 9-181. L3_FLAGMUX_TIMEOUT2_MASK0

Address Offset	See Table 9-160.		
Physical Address	0x4480 5808	**Instance**	CLK1_FLAGMUX_CLK1
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																MASK0															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x-
15:0	MASK0	mask flag inputs 0 Type: Control.	RW	0xFFFF

Table 9-182. Register Call Summary for Register L3_FLAGMUX_TIMEOUT2_MASK0

L3_MAIN Interconnect
- Time-out Flag Muxing: [0]
- L3_MAIN Time-out Flag Muxing Registers Summary and Description: [1]

Table 9-183. L3_FLAGMUX_TIMEOUT2_REGERR0

Address Offset	See Table 9-160.		
Physical Address	0x4480 580C	**Instance**	CLK1_FLAGMUX_CLK1
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																REGERR0															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x-
15:0	REGERR0	flag inputs 0 Type: Status.	R	0x-

Table 9-184. Register Call Summary for Register L3_FLAGMUX_TIMEOUT2_REGERR0

L3_MAIN Interconnect
- Time-out Flag Muxing: [0]
- L3_MAIN Time-out Flag Muxing Registers Summary and Description: [1]

9.2.5.1.7 L3_MAIN Bandwidth Regulator Register Summary and Description

Table 9-185. Bandwidth Regulator Instance Summary

Module Name	Base Address	Size
CLK1_EVE_TC0_BW_REGULATOR	0x4480 4200	4KiB
CLK1_EVE_TC1_BW_REGULATOR	0x4480 4600	4KiB
CLK1_DSP1_MDMA_BW_REGULATOR	0x4480 4C00	4KiB
CLK1_DSP2_MDMA_BW_REGULATOR	0x4480 4D00	4KiB
CLK1_IPU_BW_REGULATOR	0x4480 4E00	4KiB
CLK1_GMAC_SW_BW_REGULATOR	0x4480 5600	4KiB

9.2.5.1.7.1 L3_MAIN Bandwidth Regulator Register Summary

Table 9-186. Bandwidth Regulator Register Summary

Register Name	Type	Register Width (Bits)	Address Offset	CLK1_EVE_TC0_BW_REGULATOR L3_MAIN Physical Address	CLK1_EVE_TC1_BW_REGULATOR L3_MAIN Physical Address
L3_BW_REGULATOR_STDHOSTHDR_COREREG	R	32	0x0000 0000	0x4480 4200	0x4480 4600

Table 9-186. Bandwidth Regulator Register Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	CLK1_EVE_TC0_BW_REGULATOR L3_MAIN Physical Address	CLK1_EVE_TC1_BW_REGULATOR L3_MAIN Physical Address
L3_BW_REGULATOR_STDHOSTHDR_VERSIONREG	R	32	0x0000 0004	0x4480 4204	0x4480 4604
L3_BW_REGULATOR_BANDWIDTH	RW	32	0x0000 0008	0x4480 4208	0x4480 4608
L3_BW_REGULATOR_WATERMARK	RW	32	0x0000 000C	0x4480 420C	0x4480 460C
L3_BW_REGULATOR_PRESS	R	32	0x0000 0010	0x4480 4210	0x4480 4610
L3_BW_REGULATOR_CLEARHISTORY	RW	32	0x0000 0014	0x4480 4214	0x4480 4614

Table 9-187. Bandwidth Regulator Register Summary

Register Name	Type	Register Width (Bits)	Address Offset	CLK1_DSP1_MDMA_BW_REGULATOR L3_MAIN Physical Address	CLK1_DSP2_MDMA_BW_REGULATOR L3_MAIN Physical Address
L3_BW_REGULATOR_STDHOSTHDR_COREREG	R	32	0x0000 0000	0x4480 4C00	0x4480 4D00
L3_BW_REGULATOR_STDHOSTHDR_VERSIONREG	R	32	0x0000 0004	0x4480 4C04	0x4480 4D04
L3_BW_REGULATOR_BANDWIDTH	RW	32	0x0000 0008	0x4480 4C08	0x4480 4D08
L3_BW_REGULATOR_WATERMARK	RW	32	0x0000 000C	0x4480 4C0C	0x4480 4D0C
L3_BW_REGULATOR_PRESS	R	32	0x0000 0010	0x4480 4C10	0x4480 4D10
L3_BW_REGULATOR_CLEARHISTORY	RW	32	0x0000 0014	0x4480 4C14	0x4480 4D14

Table 9-188. Bandwidth Regulator Register Summary

Register Name	Type	Register Width (Bits)	Address Offset	CLK1_IPU_BW_REGULATOR L3_MAIN Physical Address	CLK1_GMAC_SW_BW_REGULATOR L3_MAIN Physical Address
L3_BW_REGULATOR_STDHOSTHDR_COREREG	R	32	0x0000 0000	0x4480 4E00	0x4480 5600
L3_BW_REGULATOR_STDHOSTHDR_VERSIONREG	R	32	0x0000 0004	0x4480 4E04	0x4480 5604
L3_BW_REGULATOR_BANDWIDTH	RW	32	0x0000 0008	0x4480 4E08	0x4480 5608
L3_BW_REGULATOR_WATERMARK	RW	32	0x0000 000C	0x4480 4E0C	0x4480 560C
L3_BW_REGULATOR_PRESS	R	32	0x0000 0010	0x4480 4E10	0x4480 5610
L3_BW_REGULATOR_CLEARHISTORY	RW	32	0x0000 0014	0x4480 4E14	0x4480 5614

9.2.5.1.7.2 L3_MAIN Bandwidth Regulator Register Description

Table 9-189. L3_BW_REGULATOR_STDHOSTHDR_COREREG

Address Offset	See Table 9-186 to Table 9-188		
Physical Address	0x4480 4200 0x4480 4600 0x4480 4C00 0x4480 4D00 0x4480 4E00 0x4480 5600	**Instance**	CLK1_EVE_TC0_BW_REGULATOR CLK1_EVE_TC1_BW_REGULATOR CLK1_DSP1_MDMA_BW_REGULATOR CLK1_DSP2_MDMA_BW_REGULATOR CLK1_IPU_BW_REGULATOR CLK1_GMAC_SW_BW_REGULATOR
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								STDHOSTHDR_COREREG_CORECODE								RESERVED								STDHOSTHDR_COREREG_VENDORCODE							

Bits	Field Name	Description	Type	Reset
31:22	RESERVED	Reserved	R	0x000
21:16	STDHOSTHDR_COREREG_CORECODE	The Core Code field is a constant reporting a vendor-specific core generator code. Type: Constant. Reset value: 0x31.	R	0x31
15:1	RESERVED	Reserved	R	0x0000
0	STDHOSTHDR_COREREG_VENDORCODE	The Vendor Code field is a constant reporting the core generator vendor code. Type: Constant. Reset value: 0x1. Read 0x0: Third-party vendor. Read 0x1:	R	1

Table 9-190. Register Call Summary for Register L3_BW_REGULATOR_STDHOSTHDR_COREREG

L3_MAIN Interconnect
- L3_MAIN Bandwidth Regulator Register Summary and Description: [0][1][2]

Table 9-191. L3_BW_REGULATOR_STDHOSTHDR_VERSIONREG

Address Offset	See Table 9-186 to Table 9-188		
Physical Address	0x4480 4204 0x4480 4604 0x4480 4C04 0x4480 4D04 0x4480 4E04 0x4480 5604	**Instance**	CLK1_EVE_TC0_BW_REGULATOR CLK1_EVE_TC1_BW_REGULATOR CLK1_DSP1_MDMA_BW_REGULATOR CLK1_DSP2_MDMA_BW_REGULATOR CLK1_IPU_BW_REGULATOR CLK1_GMAC_SW_BW_REGULATOR
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STDHOSTHDR_VERSIONREG_REVISIONID								STDHOSTHDR_VERSIONREG_COREPARAMSCHECKSUM																							

Bits	Field Name	Description	Type	Reset
31:24	STDHOSTHDR_VERSIONREG_REVISIONID	The Revision Identifier field is a constant reporting the core generator revision number. Type: Constant. Reset value: 0x0.	R	0x00
23:0	STDHOSTHDR_VERSIONREG_COREPARAMSCHECKSUM	Reserved. Type: Reserved. Reset value: Reserved.	R	0x000000

Table 9-192. Register Call Summary for Register L3_BW_REGULATOR_STDHOSTHDR_VERSIONREG

L3_MAIN Interconnect

- L3_MAIN Bandwidth Regulator Register Summary and Description: [0][1][2]

Table 9-193. L3_BW_REGULATOR_BANDWIDTH

Address Offset	See Table 9-186 to Table 9-188		
Physical Address	0x4480 4208 0x4480 4608 0x4480 4C08 0x4480 4D08 0x4480 4E08 0x4480 5608	**Instance**	CLK1_EVE_TC0_BW_REGULATOR CLK1_EVE_TC1_BW_REGULATOR CLK1_DSP1_MDMA_BW_REGULATOR CLK1_DSP2_MDMA_BW_REGULATOR CLK1_IPU_BW_REGULATOR CLK1_GMAC_SW_BW_REGULATOR
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																BANDWIDTH															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Reserved	R	0x0000
15:0	BANDWIDTH	Bandwidth, in bytes per second. Type: Control. Reset value: 0x0.	RW	0x0000

Table 9-194. Register Call Summary for Register L3_BW_REGULATOR_BANDWIDTH

L3_MAIN Interconnect

- Bandwidth Regulators: [0][1]
- L3_MAIN Bandwidth Regulator Register Summary and Description: [2][3][4]

Table 9-195. L3_BW_REGULATOR_WATERMARK

Address Offset	See Table 9-186 to Table 9-188		
Physical Address	0x4480 420C 0x4480 460C 0x4480 4C0C 0x4480 4D0C 0x4480 4E0C 0x4480 560C	**Instance**	CLK1_EVE_TC0_BW_REGULATOR CLK1_EVE_TC1_BW_REGULATOR CLK1_DSP1_MDMA_BW_REGULATOR CLK1_DSP2_MDMA_BW_REGULATOR CLK1_IPU_BW_REGULATOR CLK1_GMAC_SW_BW_REGULATOR
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																				WATERMARK											

Bits	Field Name	Description	Type	Reset
31:12	RESERVED	Reserved	R	0x00000
11:0	WATERMARK	Peak permissible bandwidth, in bytes. Type: Control. Reset value: 0x1.	RW	0x001

Table 9-196. Register Call Summary for Register L3_BW_REGULATOR_WATERMARK

L3_MAIN Interconnect

- Bandwidth Regulators: [0][1][2]
- L3_MAIN Bandwidth Regulator Register Summary and Description: [3][4][5]

Table 9-197. L3_BW_REGULATOR_PRESS

Address Offset	See Table 9-186 to Table 9-188		
Physical Address	0x4480 4210 0x4480 4610 0x4480 4C10 0x4480 4D10 0x4480 4E10 0x4480 5610	**Instance**	CLK1_EVE_TC0_BW_REGULATOR CLK1_EVE_TC1_BW_REGULATOR CLK1_DSP1_MDMA_BW_REGULATOR CLK1_DSP2_MDMA_BW_REGULATOR CLK1_IPU_BW_REGULATOR CLK1_GMAC_SW_BW_REGULATOR
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												PRESS_LOW		PRESS_HIGH	

Bits	Field Name	Description	Type	Reset
31:4	RESERVED	Reserved	R	0x0000 0000
3:2	PRESS_LOW	Pressure value inserted if the measured bandwidth is over the watermark. The pressure is bar graph encoded. Type: Control. Reset value: 0x0.	R	0
1:0	PRESS_HIGH	Pressure value inserted if the measured bandwidth is under the watermark. The pressure is bar graph encoded. Type: Control. Reset value: 0x1.	R	0x3

Table 9-198. Register Call Summary for Register L3_BW_REGULATOR_PRESS

L3_MAIN Interconnect

- Bandwidth Regulators: [0]
- L3_MAIN Bandwidth Regulator Register Summary and Description: [1][2][3]

Table 9-199. L3_BW_REGULATOR_CLEARHISTORY

Address Offset	See Table 9-186 to Table 9-188		
Physical Address	0x4480 4214 0x4480 4614 0x4480 4C14 0x4480 4D14 0x4480 4E14 0x4480 5614	**Instance**	CLK1_EVE_TC0_BW_REGULATOR CLK1_EVE_TC1_BW_REGULATOR CLK1_DSP1_MDMA_BW_REGULATOR CLK1_DSP2_MDMA_BW_REGULATOR CLK1_IPU_BW_REGULATOR CLK1_GMAC_SW_BW_REGULATOR
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															CLEARHISTORY

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved	R	0x0000 0000
0	CLEARHISTORY	Write a 1 clear the traffic counter Type: Give_AutoCleared. Reset value: 0x0.	RW	0

Table 9-200. Register Call Summary for Register L3_BW_REGULATOR_CLEARHISTORY

L3_MAIN Interconnect

- Bandwidth Regulators: [0]
- L3_MAIN Bandwidth Regulator Register Summary and Description: [1][2][3]

9.2.5.1.8 L3_MAIN Bandwidth Limiter Register Summary and Description

NOTE: ISS is not supported on the DRA78x family of devices.

Table 9-201. BW_LIMITER Instance Summary

Module Name	Base Address	Size
CLK1_TPTC1_RD_BW_LIMITER	0x4480 7000	256B
CLK1_TPTC1_WR_BW_LIMITER	0x4480 7100	256B
CLK1_TPTC2_RD_BW_LIMITER	0x4480 7200	256B
CLK1_TPTC2_WR_BW_LIMITER	0x4480 7300	256B
CLK1_MMU_BW_LIMITER	0x4480 3A00	256B
CLK1_IPU_BW_LIMITER	0x4480 3C00	256B
CLK1_DSP2_EDMA_BW_LIMITER	0x4480 4A00	256B
CLK1_DSP1_EDMA_BW_LIMITER	0x4480 4B00	256B
CLK1_DSP1_MDMA_BW_LIMITER	0x4480 6A00	256B
CLK1_DSP2_MDMA_BW_LIMITER	0x4480 6B00	256B
CLK1_EVE_TC0_BW_LIMITER	0x4480 6C00	256B
CLK1_EVE_TC1_BW_LIMITER	0x4480 6D00	256B

Table 9-201. BW_LIMITER Instance Summary (continued)

Module Name	Base Address	Size
CLK1_ISS_NRT2_BW_LIMITER	0x4480 6E00	256B
CLK1_ISS_NRT1_BW_LIMITER	0x4480 6F00	256B

9.2.5.1.8.1 L3_MAIN BW Limiter Register Summary

Table 9-202. BW_LIMITER Register Summary

Register Name	Type	Register Width (Bits)	Address offset	CLK1_TPTC1_RD_BW_LIMITER L3_MAIN Physical Address	CLK1_TPTC1_WR_BW_LIMITER L3_MAIN Physical Address	CLK1_TPTC2_RD_BW_LIMITER L3_MAIN Physical Address
L3_BW_LIMITER_STDHOSTHDR_COREREG	R	32	0x0000 0000	0x4480 7000	0x4480 7100	0x4480 7200
L3_BW_LIMITER_STDHOSTHDR_VERSIONREG	R	32	0x0000 0004	0x4480 7004	0x4480 7104	0x4480 7204
L3_BW_LIMITER_BANDWIDTH_FRACTIONAL	RW	32	0x0000 0008	0x4480 7008	0x4480 7108	0x4480 7208
L3_BW_LIMITER_BANDWIDTH_INTEGER	RW	32	0x0000 000C	0x4480 700C	0x4480 710C	0x4480 720C
L3_BW_LIMITER_WATERMARK_0	RW	32	0x0000 0010	0x4480 7010	0x4480 7110	0x4480 7210
L3_BW_LIMITER_CLEARHISTORY	RW	32	0x0000 0014	0x4480 7014	0x4480 7114	0x4480 7214

Table 9-203. BW_LIMITER Register Summary

Register Name	Type	Register Width (Bits)	Address offset	CLK1_TPTC2_WR_BW_LIMITER L3_MAIN Physical Address	CLK1_MMU_BW_LIMITER L3_MAIN Physical Address	CLK1_IPU_BW_LIMITER L3_MAIN Physical Address
L3_BW_LIMITER_STDHOSTHDR_COREREG	R	32	0x0000 0000	0x4480 7300	0x4480 3A00	0x4480 3C00
L3_BW_LIMITER_STDHOSTHDR_VERSIONREG	R	32	0x0000 0004	0x4480 7304	0x4480 3A04	0x4480 3C04
L3_BW_LIMITER_BANDWIDTH_FRACTIONAL	RW	32	0x0000 0008	0x4480 7308	0x4480 3A08	0x4480 3C08
L3_BW_LIMITER_BANDWIDTH_INTEGER	RW	32	0x0000 000C	0x4480 730C	0x4480 3A0C	0x4480 3C0C
L3_BW_LIMITER_WATERMARK_0	RW	32	0x0000 0010	0x4480 7310	0x4480 3A10	0x4480 3C10
L3_BW_LIMITER_CLEARHISTORY	RW	32	0x0000 0014	0x4480 7314	0x4480 3A14	0x4480 3C14

Table 9-204. BW_LIMITER Register Summary

Register Name	Type	Register Width (Bits)	Address offset	CLK1_DSP2_EDMA_BW_LIMITER L3_MAIN Physical Address	CLK1_DSP1_EDMA_BW_LIMITER L3_MAIN Physical Address	CLK1_DSP1_MDMA_BW_LIMITER L3_MAIN Physical Address
L3_BW_LIMITER_STDHOSTHDR_COREREG	R	32	0x0000 0000	0x4480 4A00	0x4480 4B00	0x4480 6A00
L3_BW_LIMITER_STDHOSTHDR_VERSIONREG	R	32	0x0000 0004	0x4480 4A04	0x4480 4B04	0x4480 6A04

Table 9-204. BW_LIMITER Register Summary (continued)

Register Name	Type	Register Width (Bits)	Address offset	CLK1_DSP2_EDMA_BW_LIMITER L3_MAIN Physical Address	CLK1_DSP1_EDMA_BW_LIMITER L3_MAIN Physical Address	CLK1_DSP1_MDMA_BW_LIMITER L3_MAIN Physical Address
L3_BW_LIMITER_BANDWIDTH_FRACTIONAL	RW	32	0x0000 0008	0x4480 4A08	0x4480 4B08	0x4480 6A08
L3_BW_LIMITER_BANDWIDTH_INTEGER	RW	32	0x0000 000C	0x4480 4A0C	0x4480 4B0C	0x4480 6A0C
L3_BW_LIMITER_WATERMARK_0	RW	32	0x0000 0010	0x4480 4A10	0x4480 4B10	0x4480 6A10
L3_BW_LIMITER_CLEARHISTORY	RW	32	0x0000 0014	0x4480 4A14	0x4480 4B14	0x4480 6A14

Table 9-205. BW_LIMITER Register Summary

Register Name	Type	Register Width (Bits)	Address offset	CLK1_DSP2_MDMA_BW_LIMITER L3_MAIN Physical Address	CLK1_EVE_TC0_BW_LIMITER L3_MAIN Physical Address	CLK1_EVE_TC1_BW_LIMITER L3_MAIN Physical Address
L3_BW_LIMITER_STDHOSTHDR_COREREG	R	32	0x0000 0000	0x4480 6B00	0x4480 6C00	0x4480 6D00
L3_BW_LIMITER_STDHOSTHDR_VERSIONREG	R	32	0x0000 0004	0x4480 6B04	0x4480 6C04	0x4480 6D04
L3_BW_LIMITER_BANDWIDTH_FRACTIONAL	RW	32	0x0000 0008	0x4480 6B08	0x4480 6C08	0x4480 6D08
L3_BW_LIMITER_BANDWIDTH_INTEGER	RW	32	0x0000 000C	0x4480 6B0C	0x4480 6C0C	0x4480 6D0C
L3_BW_LIMITER_WATERMARK_0	RW	32	0x0000 0010	0x4480 6B10	0x4480 6C10	0x4480 6D10
L3_BW_LIMITER_CLEARHISTORY	RW	32	0x0000 0014	0x4480 6B14	0x4480 6C14	0x4480 6D14

Table 9-206. BW_LIMITER Register Summary

Register Name	Type	Register Width (Bits)	Address offset	CLK1_ISS_NRT2_BW_LIMITER L3_MAIN Physical Address	CLK1_ISS_NRT1_BW_LIMITER L3_MAIN Physical Address
L3_BW_LIMITER_STDHOSTHDR_COREREG	R	32	0x0000 0000	0x4480 6E00	0x4480 6F00
L3_BW_LIMITER_STDHOSTHDR_VERSIONREG	R	32	0x0000 0004	0x4480 6E04	0x4480 6F04
L3_BW_LIMITER_BANDWIDTH_FRACTIONAL	RW	32	0x0000 0008	0x4480 6E08	0x4480 6F08
L3_BW_LIMITER_BANDWIDTH_INTEGER	RW	32	0x0000 000C	0x4480 6E0C	0x4480 6F0C
L3_BW_LIMITER_WATERMARK_0	RW	32	0x0000 0010	0x4480 6E10	0x4480 6F10
L3_BW_LIMITER_CLEARHISTORY	RW	32	0x0000 0014	0x4480 6E14	0x4480 6F14

9.2.5.1.8.2 L3_MAIN BW Limiter Register Description

Table 9-207. L3_BW_LIMITER_STDHOSTHDR_COREREG

Address Offset	See Table 9-202		
Physical Address	0x4480 7000 0x4480 7100 0x4480 7200 0x4480 7300 0x4480 3A00 0x4480 3C00 0x4480 4A00 0x4480 4B00 0x4480 6A00 0x4480 6B00 0x4480 6C00 0x4480 6D00 0x4480 6E00 0x4480 6F00	**Instance**	CLK1_TPTC1_RD_BW_LIMITER CLK1_TPTC1_WR_BW_LIMITER CLK1_TPTC2_RD_BW_LIMITER CLK1_TPTC2_WR_BW_LIMITER CLK1_MMU_BW_LIMITER CLK1_IPU_BW_LIMITER CLK1_DSP2_EDMA_BW_LIMITER CLK1_DSP1_EDMA_BW_LIMITER CLK1_DSP1_MDMA_BW_LIMITER CLK1_DSP2_MDMA_BW_LIMITER CLK1_EVE_TC0_BW_LIMITER CLK1_EVE_TC1_BW_LIMITER CLK1_ISS_NRT2_BW_LIMITER CLK1_ISS_NRT1_BW_LIMITER
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								STDHOSTHDR_COREREG_CORECODE								RESERVED								STDHOSTHDR_COREREG_VENDORCODE							

Bits	Field Name	Description	Type	Reset
31:22	RESERVED		R	0x0
21:16	STDHOSTHDR_COREREG_CORECODE	The Core Code field is a constantreporting a vendor-specific core generator code. Type: Constant. Reset value: 0x2C.	R	0x2C
15:1	RESERVED		R	0x0
0	STDHOSTHDR_COREREG_VENDORCODE	The Vendor Code field is a constantreporting the core generator vendor code. Type: Constant. Reset value: 0x1. Read 0x1: Read 0x0: Third-party vendor.	R	1

Table 9-208. Register Call Summary for Register L3_BW_LIMITER_STDHOSTHDR_COREREG

L3_MAIN Interconnect

- L3_MAIN Bandwidth Limiter Register Summary and Description: [0][1][2][3][4]

Table 9-209. L3_BW_LIMITER_STDHOSTHDR_VERSIONREG

Address Offset	See Table 9-202		
Physical Address	0x4480 7004 0x4480 7104 0x4480 7204 0x4480 7304 0x4480 3A04 0x4480 3C04 0x4480 4A04 0x4480 4B04 0x4480 6A04 0x4480 6B04 0x4480 6C04 0x4480 6D04 0x4480 6E04 0x4480 6F04	**Instance**	CLK1_TPTC1_RD_BW_LIMITER CLK1_TPTC1_WR_BW_LIMITER CLK1_TPTC2_RD_BW_LIMITER CLK1_TPTC2_WR_BW_LIMITER CLK1_MMU_BW_LIMITER CLK1_IPU_BW_LIMITER CLK1_DSP2_EDMA_BW_LIMITER CLK1_DSP1_EDMA_BW_LIMITER CLK1_DSP1_MDMA_BW_LIMITER CLK1_DSP2_MDMA_BW_LIMITER CLK1_EVE_TC0_BW_LIMITER CLK1_EVE_TC1_BW_LIMITER CLK1_ISS_NRT2_BW_LIMITER CLK1_ISS_NRT1_BW_LIMITER
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STDHOSTHDR_VERSIONREG_REVISIONID								STDHOSTHDR_VERSIONREG_COREPARAMSCHECKSUM																									

Bits	Field Name	Description	Type	Reset
31:24	STDHOSTHDR_VERSIONREG_REVISIONID	The Revision Identifier field is a constantreporting the core generator revision number. Type: Constant. Reset value: 0x0.	R	0x00
23:0	STDHOSTHDR_VERSIONREG_COREPARAMSCHECKSUM	Reserved. Type: Reserved. Reset value: Reserved.	R	0x0

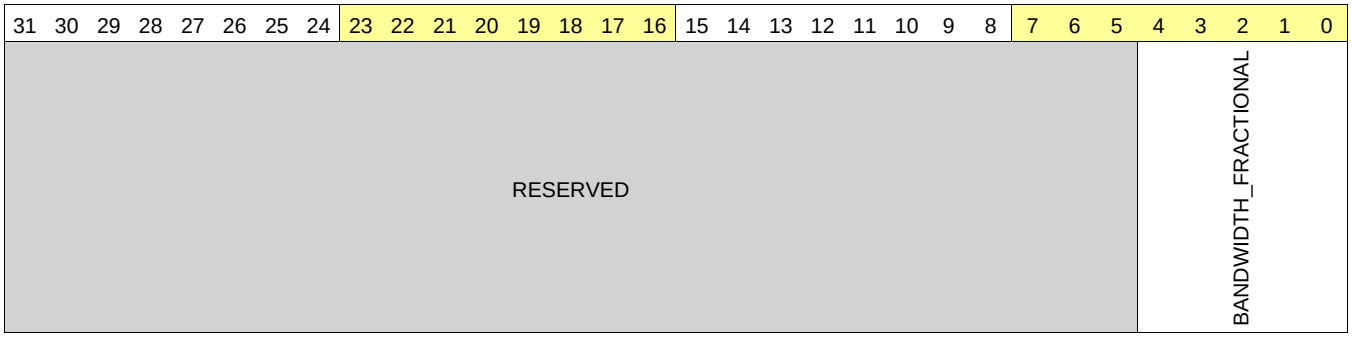
Table 9-210. Register Call Summary for Register L3_BW_LIMITER_STDHOSTHDR_VERSIONREG

L3_MAIN Interconnect

- L3_MAIN Bandwidth Limiter Register Summary and Description: [0][1][2][3][4]

Table 9-211. L3_BW_LIMITER_BANDWIDTH_FRACTIONAL

Address Offset	See Table 9-202		
Physical Address	0x4480 7008 0x4480 7108 0x4480 7208 0x4480 7308 0x4480 3A08 0x4480 3C08 0x4480 4A08 0x4480 4B08 0x4480 6A08 0x4480 6B08 0x4480 6C08 0x4480 6D08 0x4480 6E08 0x4480 6F08	**Instance**	CLK1_TPTC1_RD_BW_LIMITER CLK1_TPTC1_WR_BW_LIMITER CLK1_TPTC2_RD_BW_LIMITER CLK1_TPTC2_WR_BW_LIMITER CLK1_MMU_BW_LIMITER CLK1_IPU_BW_LIMITER CLK1_DSP2_EDMA_BW_LIMITER CLK1_DSP1_EDMA_BW_LIMITER CLK1_DSP1_MDMA_BW_LIMITER CLK1_DSP2_MDMA_BW_LIMITER CLK1_EVE_TC0_BW_LIMITER CLK1_EVE_TC1_BW_LIMITER CLK1_ISS_NRT2_BW_LIMITER CLK1_ISS_NRT1_BW_LIMITER
Description			
Type	RW		



Bits	Field Name	Description	Type	Reset
31:5	RESERVED		R	0x0
4:0	BANDWIDTH_FRACTIONAL	Fractional part of bandwitdh in terms of bytes per second Type: Control. Reset value: 0x0.	RW	0x0

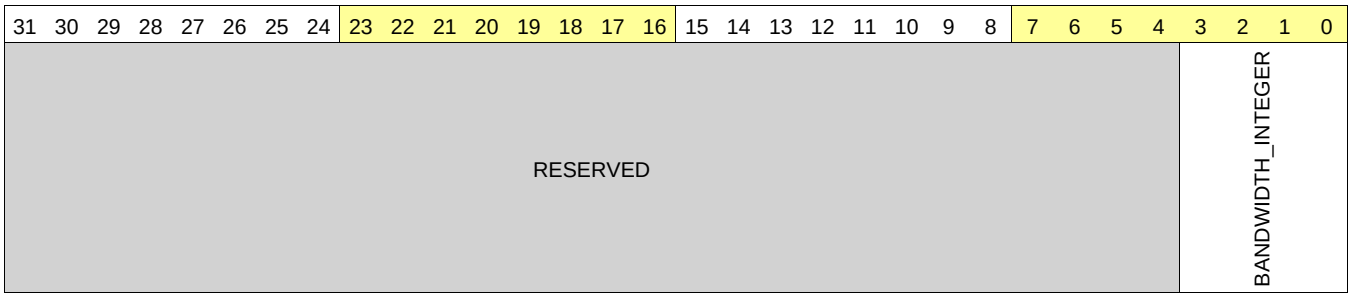
Table 9-212. Register Call Summary for Register L3_BW_LIMITER_BANDWIDTH_FRACTIONAL

L3_MAIN Interconnect

- Bandwidth Limiters: [0][1][2]
- L3_MAIN Bandwidth Limiter Register Summary and Description: [3][4][5][6][7]

Table 9-213. L3_BW_LIMITER_BANDWIDTH_INTEGER

Address Offset	See Table 9-202		
Physical Address	0x4480 700C 0x4480 710C 0x4480 720C 0x4480 730C 0x4480 3A0C 0x4480 3C0C 0x4480 4A0C 0x4480 4B0C 0x4480 6A0C 0x4480 6B0C 0x4480 6C0C 0x4480 6D0C 0x4480 6E0C 0x4480 6F0C	**Instance**	CLK1_TPTC1_RD_BW_LIMITER CLK1_TPTC1_WR_BW_LIMITER CLK1_TPTC2_RD_BW_LIMITER CLK1_TPTC2_WR_BW_LIMITER CLK1_MMU_BW_LIMITER CLK1_IPU_BW_LIMITER CLK1_DSP2_EDMA_BW_LIMITER CLK1_DSP1_EDMA_BW_LIMITER CLK1_DSP1_MDMA_BW_LIMITER CLK1_DSP2_MDMA_BW_LIMITER CLK1_EVE_TC0_BW_LIMITER CLK1_EVE_TC1_BW_LIMITER CLK1_ISS_NRT2_BW_LIMITER CLK1_ISS_NRT1_BW_LIMITER
Description			
Type	RW		



Bits	Field Name	Description	Type	Reset
31:4	RESERVED		R	0x0
3:0	BANDWIDTH_INTEGER	Integer part of bandwitdh in terms of bytes per second Type: Control. Reset value: 0x0.	RW	0x0

Table 9-214. Register Call Summary for Register L3_BW_LIMITER_BANDWIDTH_INTEGER

L3_MAIN Interconnect

- Bandwidth Limiters: [0][1][2]
- L3_MAIN Bandwidth Limiter Register Summary and Description: [3][4][5][6][7]

Table 9-215. L3_BW_LIMITER_WATERMARK_0

Address Offset	See Table 9-202		
Physical Address	0x4480 7010 0x4480 7110 0x4480 7210 0x4480 7310 0x4480 3A10 0x4480 3C10 0x4480 4A10 0x4480 4B10 0x4480 6A10 0x4480 6B10 0x4480 6C10 0x4480 6D10 0x4480 6E10 0x4480 6F10	**Instance**	CLK1_TPTC1_RD_BW_LIMITER CLK1_TPTC1_WR_BW_LIMITER CLK1_TPTC2_RD_BW_LIMITER CLK1_TPTC2_WR_BW_LIMITER CLK1_MMU_BW_LIMITER CLK1_IPU_BW_LIMITER CLK1_DSP2_EDMA_BW_LIMITER CLK1_DSP1_EDMA_BW_LIMITER CLK1_DSP1_MDMA_BW_LIMITER CLK1_DSP2_MDMA_BW_LIMITER CLK1_EVE_TC0_BW_LIMITER CLK1_EVE_TC1_BW_LIMITER CLK1_ISS_NRT2_BW_LIMITER CLK1_ISS_NRT1_BW_LIMITER
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																WATERMARK_0															

Bits	Field Name	Description	Type	Reset
31:14	RESERVED		R	0x0
13:0	WATERMARK_0	Peak bandwidth allowed Type: Control. Reset value: 0x3FF.	RW	0x3FFF

Table 9-216. Register Call Summary for Register L3_BW_LIMITER_WATERMARK_0

L3_MAIN Interconnect

- Bandwidth Limiters: [0][1]
- L3_MAIN Bandwidth Limiter Register Summary and Description: [2][3][4][5][6]

Table 9-217. L3_BW_LIMITER_CLEARHISTORY

Address Offset	See Table 9-202		
Physical Address	0x4480 7014 0x4480 7114 0x4480 7214 0x4480 7314 0x4480 3A14 0x4480 3C14 0x4480 4A14 0x4480 4B14 0x4480 6A14 0x4480 6B14 0x4480 6C14 0x4480 6D14 0x4480 6E14 0x4480 6F14	**Instance**	CLK1_TPTC1_RD_BW_LIMITER CLK1_TPTC1_WR_BW_LIMITER CLK1_TPTC2_RD_BW_LIMITER CLK1_TPTC2_WR_BW_LIMITER CLK1_MMU_BW_LIMITER CLK1_IPU_BW_LIMITER CLK1_DSP2_EDMA_BW_LIMITER CLK1_DSP1_EDMA_BW_LIMITER CLK1_DSP1_MDMA_BW_LIMITER CLK1_DSP2_MDMA_BW_LIMITER CLK1_EVE_TC0_BW_LIMITER CLK1_EVE_TC1_BW_LIMITER CLK1_ISS_NRT2_BW_LIMITER CLK1_ISS_NRT1_BW_LIMITER
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															CLEARHISTORY

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	CLEARHISTORY	Write a 1 clear the traffic counter Type: Give_AutoCleared. Reset value: 0x0.	RW	0

Table 9-218. Register Call Summary for Register L3_BW_LIMITER_CLEARHISTORY

L3_MAIN Interconnect

- Bandwidth Limiters: [0]
- L3_MAIN Bandwidth Limiter Register Summary and Description: [1][2][3][4][5]

9.2.5.1.9 L3_MAIN STATCOLL Register Summary and Description

Table 9-219. STATCOLL Instance Summary

Module Name	Base Address	Size
CLK2_FLAGMUX_STATCOLL	0x4500 0500	512 bytes
CLK2_STATCOLL1	0x4500 2000	512 bytes
CLK2_STATCOLL3	0x4500 3000	512 bytes
CLK2_STATCOLL2	0x4500 4000	512 bytes

Table 9-219. STATCOLL Instance Summary (continued)

Module Name	Base Address	Size
CLK2_STATCOLL4	0x4500 5000	512 bytes

9.2.5.1.9.1 L3_MAIN STATCOLL Register Summary

Table 9-220. STATCOLL Register Summary

Register Name	Type	Register Width (bits)	Address offset for FlAGMUX	CLK2_FLAGMUX_STATCOLL L3_MAIN Physical Address
L3_STCOL_STDHOSTHDR_COREREG	R	32	0x0100 0500	0x4500 0500
L3_STCOL_STDHOSTHDR_VERSIONREG	R	32	0x0100 0504	0x4500 0504
L3_STCOL_MASK0	RW	32	0x0100 0508	0x4500 0508
L3_STCOL_REGERR0	R	32	0x0100 050C	0x4500 050C

Table 9-221. STATCOLL Register Summary

Register Name	Type	Register Width (Bits)	Address Offset for SDRAM	CLK2_STATCOLL1 L3_MAIN Physical Address	CLK2_STATCOLL3 L3_MAIN Physical Address	CLK2_STATCOLL2 L3_MAIN Physical Address	CLK2_STATCOLL4 L3_MAIN Physical Address
L3_STCOL_STDHOSTHDR_COREREG	R	32	0x0000 0000	0x4500 2000	0x4500 3000	0x4500 4000	0x4500 5000
L3_STCOL_STDHOSTHDR_VERSIONREG	R	32	0x0000 0004	0x4500 2004	0x4500 3004	0x4500 4004	0x4500 5004
L3_STCOL_EN	RW	32	0x0000 0008	0x4500 2008	0x4500 3008	0x4500 4008	0x4500 5008
L3_STCOL_SOFTEN	RW	32	0x0000 000C	0x4500 200C	0x4500 300C	0x4500 400C	0x4500 500C
L3_STCOL_IGNORESUSPEND	RW	32	0x0000 0010	0x4500 2010	0x4500 3010	0x4500 4010	0x4500 5010
L3_STCOL_TRIGEN	RW	32	0x0000 0014	0x4500 2014	0x4500 3014	0x4500 4014	0x4500 5014
L3_STCOL_REQEVT	RW	32	0x0000 0018	0x4500 2018	0x4500 3018	0x4500 4018	0x4500 5018
L3_STCOL_RSPEVT	RW	32	0x0000 001C	0x4500 201C	0x4500 301C	0x4500 401C	0x4500 501C
L3_STCOL_EVTMUX_SEL0	RW	32	0x0000 0020	0x4500 2020	0x4500 3020	0x4500 4020	0x4500 5020
L3_STCOL_EVTMUX_SEL1	RW	32	0x0000 0024	0x4500 2024	0x4500 3024	0x4500 4024	0x4500 5024
L3_STCOL_EVTMUX_SEL2	RW	32	0x0000 0028	0x4500 2028	0x4500 3028	0x4500 4028	0x4500 5028
L3_STCOL_EVTMUX_SEL3	RW	32	0x0000 002C	0x4500 202C	0x4500 302C	0x4500 402C	0x4500 502C
L3_STCOL_DUMP_IDENTIFIER	R	32	0x0000 0040	0x4500 2040	0x4500 3040	0x4500 4040	0x4500 5040
L3_STCOL_DUMP_COLLECTTIME	RW	32	0x0000 0044	0x4500 2044	0x4500 3044	0x4500 4044	0x4500 5044
L3_STCOL_DUMP_SLVADDR	R	32	0x0000 0048	0x4500 2048	0x4500 3048	0x4500 4048	0x4500 5048
L3_STCOL_DUMP_MSTADDR	R	32	0x0000 004C	0x4500 204C	0x4500 304C	0x4500 404C	0x4500 504C
L3_STCOL_DUMP_SLVOFS	RW	32	0x0000 0050	0x4500 2050	0x4500 3050	0x4500 4050	0x4500 5050
L3_STCOL_DUMP_MODE	RW	32	0x0000 0054	0x4500 2054	0x4500 3054	0x4500 4054	0x4500 5054
L3_STCOL_DUMP_SEND	RW	32	0x0000 0058	0x4500 2058	0x4500 3058	0x4500 4058	0x4500 5058
L3_STCOL_DUMP_DISABLE	RW	32	0x0000 005C	0x4500 205C	0x4500 305C	0x4500 405C	0x4500 505C
L3_STCOL_DUMP_ALARM_TRIG	RW	32	0x0000 0060	0x4500 2060	0x4500 3060	0x4500 4060	0x4500 5060
L3_STCOL_DUMP_ALARM_MINVAL	RW	32	0x0000 0064	0x4500 2064	0x4500 3064	0x4500 4064	0x4500 5064
L3_STCOL_DUMP_ALARM_MAXVAL	RW	32	0x0000 0068	0x4500 2068	0x4500 3068	0x4500 4068	0x4500 5068
L3_STCOL_DUMP_ALARM_MODE0	RW	32	0x0000 006C	0x4500 206C	0x4500 306C	0x4500 406C	0x4500 506C
L3_STCOL_DUMP_ALARM_MODE1	RW	32	0x0000 0070	0x4500 2070	0x4500 3070	0x4500 4070	0x4500 5070
L3_STCOL_DUMP_ALARM_MODE2	RW	32	0x0000 0074	0x4500 2074	0x4500 3074	0x4500 4074	0x4500 5074
L3_STCOL_DUMP_ALARM_MODE3	RW	32	0x0000 0078	0x4500 2078	0x4500 3078	0x4500 4078	0x4500 5078
L3_STCOL_DUMP_CNT0	R	32	0x0000 008C	0x4500 208C	0x4500 308C	0x4500 408C	0x4500 508C
L3_STCOL_DUMP_CNT1	R	32	0x0000 0090	0x4500 2090	0x4500 3090	0x4500 4090	0x4500 5090
L3_STCOL_DUMP_CNT2	R	32	0x0000 0094	0x4500 2094	0x4500 3094	0x4500 4094	0x4500 5094
L3_STCOL_DUMP_CNT3	R	32	0x0000 0098	0x4500 2098	0x4500 3098	0x4500 4098	0x4500 5098

Table 9-221. STATCOLL Register Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset for SDRAM	CLK2_STATCOLL1 L3_MAIN Physical Address	CLK2_STATCOLL3 L3_MAIN Physical Address	CLK2_STATCOLL2 L3_MAIN Physical Address	CLK2_STATCOLL4 L3_MAIN Physical Address
L3_STCOL_FILTER_i_GLOBALEN[1]	RW	32	0x0000 00AC + (0x158*i)	0x4500 20AC + (0x158*i)	0x4500 30AC + (0x158*i)	0x4500 40AC + (0x158*i)	0x4500 50AC + (0x158*i)
L3_STCOL_FILTER_i_ADDRMIN[1]	RW	32	0x0000 00B0 + (0x158*i)	0x4500 20B0 + (0x158*i)	0x4500 30B0 + (0x158*i)	0x4500 40B0 + (0x158*i)	0x4500 50B0 + (0x158*i)
L3_STCOL_FILTER_i_ADDRMAX[1]	RW	32	0x0000 00B4 + (0x158*i)	0x4500 20B4 + (0x158*i)	0x4500 30B4 + (0x158*i)	0x4500 40B4 + (0x158*i)	0x4500 50B4 + (0x158*i)
L3_STCOL_FILTER_i_ADDREN[1]	RW	32	0x0000 00B8 + (0x158*i)	0x4500 20B8 + (0x158*i)	0x4500 30B8 + (0x158*i)	0x4500 40B8 + (0x158*i)	0x4500 50B8 + (0x158*i)
L3_STCOL_FILTER_i_EN0[1]	RW	32	0x0000 00BC + (0x158*i)	0x4500 20BC + (0x158*i)	0x4500 30BC + (0x158*i)	0x4500 40BC + (0x158*i)	0x4500 50BC + (0x158*i)
L3_STCOL_FILTER_i_MASK0_RD [1]	RW	32	0x0000 00C0 + (0x158*i)	0x4500 20C0 + (0x158*i)	0x4500 30C0 + (0x158*i)	0x4500 40C0 + (0x158*i)	0x4500 50C0 + (0x158*i)
L3_STCOL_FILTER_i_MASK0_WR [1]	RW	32	0x0000 00C4 + (0x158*i)	0x4500 20C4 + (0x158*i)	0x4500 30C4 + (0x158*i)	0x4500 40C4 + (0x158*i)	0x4500 50C4 + (0x158*i)
L3_STCOL_FILTER_i_MASK0_MSTADDR [1]	RW	32	0x0000 00C8 + (0x158*i)	0x4500 20C8 + (0x158*i)	0x4500 30C8 + (0x158*i)	0x4500 40C8 + (0x158*i)	0x4500 50C8 + (0x158*i)
L3_STCOL_FILTER_i_MASK0_SLVADDR[1]	RW	32	0x0000 00CC + (0x158*i)	0x4500 20CC + (0x158*i)	0x4500 30CC + (0x158*i)	0x4500 40CC + (0x158*i)	0x4500 50CC + (0x158*i)
L3_STCOL_FILTER_i_MASK0_ERR [1]	RW	32	0x0000 00D0 + (0x158*i)	0x4500 20D0 + (0x158*i)	0x4500 30D0 + (0x158*i)	0x4500 40D0 + (0x158*i)	0x4500 50D0 + (0x158*i)
L3_STCOL_FILTER_i_MASK0_REQUSERINFO[1]	RW	32	0x0000 00D4 + (0x158*i)	0x4500 20D4 + (0x158*i)	0x4500 30D4 + (0x158*i)	0x4500 40D4 + (0x158*i)	0x4500 50D4 + (0x158*i)
L3_STCOL_FILTER_i_MASK0_RSPUSERINFO[1]	RW	32	0x0000 00D8 + (0x158*i)	0x4500 20D8 + (0x158*i)	0x4500 30D8 + (0x158*i)	0x4500 40D8 + (0x158*i)	0x4500 50D8 + (0x158*i)
L3_STCOL_FILTER_i_MATCH0_RD [1]	RW	32	0x0000 00E0 + (0x158*i)	0x4500 20E0 + (0x158*i)	0x4500 30E0 + (0x158*i)	0x4500 40E0 + (0x158*i)	0x4500 50E0 + (0x158*i)
L3_STCOL_FILTER_i_MATCH0_WR[1]	RW	32	0x0000 00E4 + (0x158*i)	0x4500 20E4 + (0x158*i)	0x4500 30E4 + (0x158*i)	0x4500 40E4 + (0x158*i)	0x4500 50E4 + (0x158*i)
L3_STCOL_FILTER_i_MATCH0_MSTADDR [1]	RW	32	0x0000 00E8 + (0x158*i)	0x4500 20E8 + (0x158*i)	0x4500 30E8 + (0x158*i)	0x4500 40E8 + (0x158*i)	0x4500 50E8 + (0x158*i)
L3_STCOL_FILTER_i_MATCH0_SLVADDR[1]	RW	32	0x0000 00EC + (0x158*i)	0x4500 20EC + (0x158*i)	0x4500 30EC + (0x158*i)	0x4500 40EC + (0x158*i)	0x4500 50EC + (0x158*i)
L3_STCOL_FILTER_i_MATCH0_ERR [1]	RW	32	0x0000 00F0 + (0x158*i)	0x4500 20F0 + (0x158*i)	0x4500 30F0 + (0x158*i)	0x4500 40F0 + (0x158*i)	0x4500 50F0 + (0x158*i)
L3_STCOL_FILTER_i_MATCH0_REQUSERINFO[1]	RW	32	0x0000 00F4 + (0x158*i)	0x4500 20F4 + (0x158*i)	0x4500 30F4 + (0x158*i)	0x4500 40F4 + (0x158*i)	0x4500 50F4 + (0x158*i)
L3_STCOL_FILTER_i_MATCH0_RSPUSERINFO[1]	RW	32	0x0000 00F8 + (0x158*i)	0x4500 20F8 + (0x158*i)	0x4500 30F8 + (0x158*i)	0x4500 40F8 + (0x158*i)	0x4500 50F8 + (0x158*i)

(1) i = 0 to 3

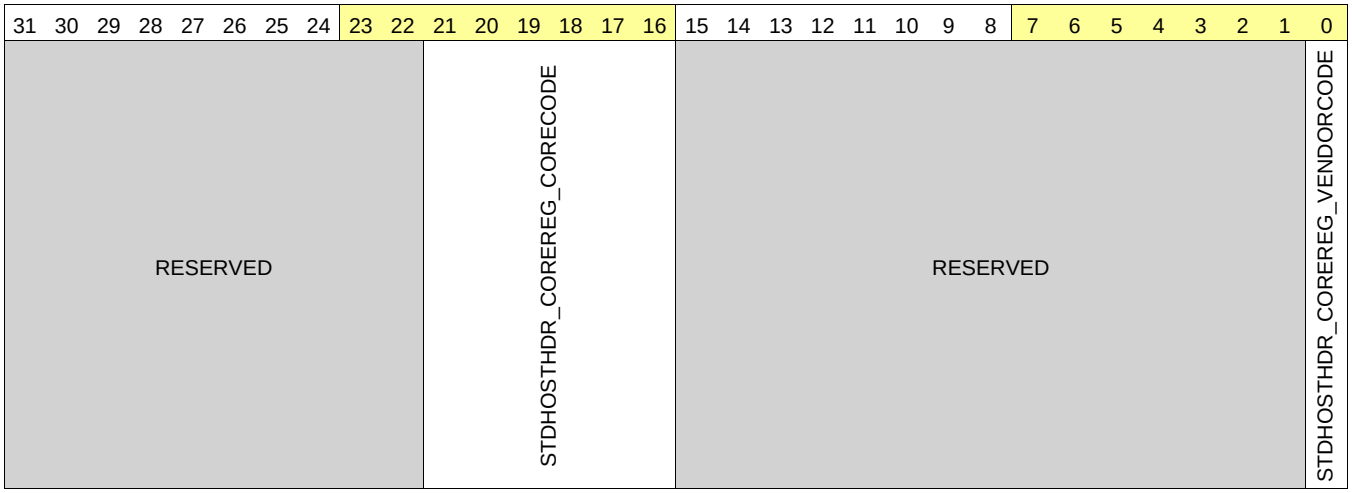
Table 9-221. STATCOLL Register Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset for SDRAM	CLK2_STATCOLL1 L3_MAIN Physical Address	CLK2_STATCOLL3 L3_MAIN Physical Address	CLK2_STATCOLL2 L3_MAIN Physical Address	CLK2_STATCOLL4 L3_MAIN Physical Address
L3_STCOL_OP_i_THRESHOLD_MINVAL (1)	RW	32	0x0000 01F0 + (0x158*i)	0x4500 21F0 + (0x158*i)	0x4500 31F0 + (0x158*i)	0x4500 41F0 + (0x158*i)	0x4500 51F0 + (0x158*i)
L3_STCOL_OP_i_THRESHOLD_MAXVAL (1)	RW	32	0x0000 01F4 + (0x158*i)	0x4500 21F4 + (0x158*i)	0x4500 31F4 + (0x158*i)	0x4500 41F4 + (0x158*i)	0x4500 51F4 + (0x158*i)
L3_STCOL_OP_i_EVTINFOSEL (1)	RW	32	0x0000 01F8 + (0x158*i)	0x4500 21F8 + (0x158*i)	0x4500 31F8 + (0x158*i)	0x4500 41F8 + (0x158*i)	0x4500 51F8 + (0x158*i)
L3_STCOL_OP_i_SEL (1)	RW	32	0x0000 01FC + (0x158*i)	0x4500 21FC + (0x158*i)	0x4500 31FC + (0x158*i)	0x4500 41FC + (0x158*i)	0x4500 51FC + (0x158*i)

9.2.5.1.9.2 L3_MAIN STATCOLL Register Description

Table 9-222. L3_STCOL_STDHOSTHDR_COREREG

Address Offset	See Table 9-221.		
Physical Address	0x4500 0500 0x4500 2000 0x4500 3000 0x4500 4000 0x4500 5000	**Instance**	CLK2_FLAGMUX_STATCOLL CLK2_STATCOLL1 CLK2_STATCOLL3 CLK2_STATCOLL2 CLK2_STATCOLL4
Description			
Type	R		



Bits	Field Name	Description	Type	Reset
31:22	RESERVED	Reserved	R	0x000
21:16	STDHOSTHDR_COREREG_CORECODE	The Core Code field is a constant reporting a vendor-specific core generator code. Type: Constant. Reset value: 0x3A. (When the instance is CLK2_FLAGMUX_STATCOLL reset value is 0x37)	R	0x3A
15:1	RESERVED	Reserved	R	0x0000
0	STDHOSTHDR_COREREG_VENDORCODE	The Vendor Code field is a constant reporting the core generator vendor code. Type: Constant. Reset value: 0x1. Read 0x0: Third-party vendor. Read 0x1:	R1	1

Table 9-223. Register Call Summary for Register L3_STCOL_STDHOSTHDR_COREREG

L3_MAIN Interconnect

- L3_MAIN STATCOLL Register Summary and Description: [0][1]

Table 9-224. L3_STCOL_STDHOSTHDR_VERSIONREG

Address Offset	See Table 9-221.		
Physical Address	0x4500 0504 0x4500 2004 0x4500 3004 0x4500 4004 0x4500 5004	**Instance**	CLK2_FLAGMUX_STATCOLL CLK2_STATCOLL1 CLK2_STATCOLL3 CLK2_STATCOLL2 CLK2_STATCOLL4
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STDHOSTHDR_VERSIONREG_REVISIONID								STDHOSTHDR_VERSIONREG_COREPARAMSCHECKSUM																							

Bits	Field Name	Description	Type	Reset
31:24	STDHOSTHDR_VERSIONREG_REVISIONID	The Revision Identifier field is a constant reporting the core generator revision number. Type: Constant. Reset value: 0x1.	R	0x1
23:0	STDHOSTHDR_VERSIONREG_COREPARAMSCHECKSUM	Reserved. Type: Reserved. Reset value: Reserved.	R	0x000000

Table 9-225. Register Call Summary for Register L3_STCOL_STDHOSTHDR_VERSIONREG

L3_MAIN Interconnect

- L3_MAIN STATCOLL Register Summary and Description: [0][1]

Table 9-226. L3_STCOL_MASK0

Address Offset	See Table 9-220.		
Physical Address	0x4500 0508	**Instance**	CLK2_FLAGMUX_STATCOLL
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								MASK0							

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0000 0000
9:0	MASK0	mask flag inputs 0 Type: Control. Reset value: 0x7.	RW	0x3ff

Table 9-227. Register Call Summary for Register L3_STCOL_MASK0

L3_MAIN Interconnect

- L3_MAIN STATCOLL Register Summary and Description: [0]

Table 9-228. L3_STCOL_REGERR0

Address Offset	See Table 9-220.		
Physical Address	0x4500 050C	**Instance**	CLK2_FLAGMUX_STATCOLL
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								REGERR0							

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0000 0000
9:0	REGERR0	flag inputs 0 Type: Status. Reset value: X.	R	0x0000 0000

Table 9-229. Register Call Summary for Register L3_STCOL_REGERR0

L3_MAIN Interconnect

- L3_MAIN STATCOLL Register Summary and Description: [0]

Table 9-230. L3_STCOL_EN

Address Offset	See Table 9-221.		
Physical Address	0x4500 2008 0x4500 3008 0x4500 4008 0x4500 5008	**Instance**	CLK2_STATCOLL1 CLK2_STATCOLL3 CLK2_STATCOLL2 CLK2_STATCOLL4
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															EN

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved	R	0x0000 0000
0	EN	Enable performance monitoring, this will also shut down the clock if En = 0 Type: Control. Reset value: 0x0.	RW	0

Table 9-231. Register Call Summary for Register L3_STCOL_EN

L3_MAIN Interconnect

- Statistic Collectors Group: [0]
- L3_MAIN STATCOLL Register Summary and Description: [1]

Table 9-232. L3_STCOL_SOFTEN

Address Offset	See Table 9-221.		
Physical Address	0x4500 200C 0x4500 300C 0x4500 400C 0x4500 500C	**Instance**	CLK2_STATCOLL1 CLK2_STATCOLL3 CLK2_STATCOLL2 CLK2_STATCOLL4
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															SOFTEN

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved	R	0x0000 0000
0	SOFTEN	Software enable for performance monitoring Type: Control. Reset value: 0x0.	RW	0

Table 9-233. Register Call Summary for Register L3_STCOL_SOFTEN

L3_MAIN Interconnect

- Statistic Collectors Group: [0]
- L3_MAIN STATCOLL Register Summary and Description: [1]

Table 9-234. L3_STCOL_IGNORESUSPEND

Address Offset	See Table 9-221.		
Physical Address	0x4500 2010 0x4500 3010 0x4500 4010 0x4500 5010	**Instance**	CLK2_STATCOLL1 CLK2_STATCOLL3 CLK2_STATCOLL2 CLK2_STATCOLL4
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															IGNORESUSPEND

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	IGNORESUSPEND	Ignore suspend if set to one for suspend mechanism Type: Control. Reset value: 0x0.	RW	0

Table 9-235. Register Call Summary for Register L3_STCOL_IGNORESUSPEND

L3_MAIN Interconnect

- L3_MAIN STATCOLL Register Summary and Description: [0]

Table 9-236. L3_STCOL_TRIGEN

Address Offset	See Table 9-221.		
Physical Address	0x4500 2014 0x4500 3014 0x4500 4014 0x4500 5014	**Instance**	CLK2_STATCOLL1 CLK2_STATCOLL3 CLK2_STATCOLL2 CLK2_STATCOLL4
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															TRIGEN

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved	R	0x0000 0000
0	TRIGEN	TrigEn when set, it enable the external trigger start and stop Type: Control. Reset value: 0x0.	RW	0

Table 9-237. Register Call Summary for Register L3_STCOL_TRIGEN

L3_MAIN Interconnect

- L3_MAIN STATCOLL Register Summary and Description: [0]

Table 9-238. L3_STCOL_REQEVT

Address Offset	See Table 9-221.		
Physical Address	0x4500 2018 0x4500 3018 0x4500 4018 0x4500 5018	**Instance**	CLK2_STATCOLL1 CLK2_STATCOLL3 CLK2_STATCOLL2 CLK2_STATCOLL4
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												REQEVT			

Bits	Field Name	Description	Type	Reset
31:4	RESERVED	Reserved	R	0x0000000
3:0	REQEVT	Req event select Type: Control. Reset value: 0x0. 0x0: Collect is disabled default value 0x1: Collect all event: hit always (cycle) 0x2: Collect transfers: actually used cycle for transferring aN NTTP word 0x3: Collect wait cycle: transfer has been delayed by source 0x4: Collect busy: transfer has been delayed by destination 0x5: Collect packet: new packet start 0x6: Collect data: data cycle transfer, write for requests, read for responses 0x7: Collect idles: transfer is not initiated by source 0x8: Collect latency: hit when actually detecting debug bit on response links	RW	0x0

Table 9-239. Register Call Summary for Register L3_STCOL_REQEVT

L3_MAIN Interconnect

- Statistic Collectors Group: [0]
- L3_MAIN STATCOLL Register Summary and Description: [1]

Table 9-240. L3_STCOL_RSPEVT

Address Offset	See Table 9-221.		
Physical Address	0x4500 201C 0x4500 301C 0x4500 401C 0x4500 501C	**Instance**	CLK2_STATCOLL1 CLK2_STATCOLL3 CLK2_STATCOLL2 CLK2_STATCOLL4
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												RSPEVT			

Bits	Field Name	Description	Type	Reset
31:4	RESERVED	Reserved	R	0x0000000
3:0	RSPEVT	Rsp event select Type: Control. Reset value: 0x0. 0x0: Collect is disabled default value 0x1: Collect all event: hit always (cycle) 0x2: Collect transfers: actually used cycle for transferring a NTTP word 0x3: Collect wait cycle: transfer has been delayed by source 0x4: Collect busy: transfer has been delayed by destination 0x5: Collect packet: new packet start 0x6: Collect data: data cycle transfer, write for requests, read for responses 0x7: Collect idles: transfer is not initiated by source 0x8: Collect latency: hit when actually detecting debug bit on response links	RW	0x0

Table 9-241. Register Call Summary for Register L3_STCOL_RSPEVT

L3_MAIN Interconnect

- Statistic Collectors Group: [0]
- L3_MAIN STATCOLL Register Summary and Description: [1]

Table 9-242. L3_STCOL_EVTMUX_SEL0

Address Offset	See Table 9-221.		
Physical Address	0x4500 2020 0x4500 3020 0x4500 4020 0x4500 5020	**Instance**	CLK2_STATCOLL1 CLK2_STATCOLL3 CLK2_STATCOLL2 CLK2_STATCOLL4
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														EVTMUX_SEL0	

Bits	Field Name	Description	Type	Reset
31:3	RESERVED	Reserved	R	0x0000 0000
2:0	EVTMUX_SEL0	The select of the mux 0 Type: Control. Reset value: 0x0.	RW	0x0

Table 9-243. Register Call Summary for Register L3_STCOL_EVTMUX_SEL0

L3_MAIN Interconnect

- Statistic Collectors Group: [0]
- L3_MAIN STATCOLL Register Summary and Description: [1]

Table 9-244. L3_STCOL_EVTMUX_SEL1

Address Offset	See Table 9-221.		
Physical Address	0x4500 2024 0x4500 3024 0x4500 4024 0x4500 5024	**Instance**	CLK2_STATCOLL1 CLK2_STATCOLL3 CLK2_STATCOLL2 CLK2_STATCOLL4
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														EVTMUX_SEL1	

Bits	Field Name	Description	Type	Reset
31:3	RESERVED	Reserved	R	0x0000 0000
2:0	EVTMUX_SEL1	The select of the mux 1 Type: Control. Reset value: 0x0.	RW	0x0

Table 9-245. Register Call Summary for Register L3_STCOL_EVTMUX_SEL1

L3_MAIN Interconnect

- L3_MAIN STATCOLL Register Summary and Description: [0]

Table 9-246. L3_STCOL_EVTMUX_SEL2

Address Offset	See Table 9-221.		
Physical Address	0x4500 2028 0x4500 3028 0x4500 4028 0x4500 5028	**Instance**	CLK2_STATCOLL1 CLK2_STATCOLL3 CLK2_STATCOLL2 CLK2_STATCOLL4
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														EVTMUX_SEL2	

Bits	Field Name	Description	Type	Reset
31:3	RESERVED	Reserved	R	0x0000 0000
2:0	EVTMUX_SEL2	The select of the mux 2 Type: Control. Reset value: 0x0.	RW	0x0

Table 9-247. Register Call Summary for Register L3_STCOL_EVTMUX_SEL2

L3_MAIN Interconnect
- L3_MAIN STATCOLL Register Summary and Description: [0]

Table 9-248. L3_STCOL_EVTMUX_SEL3

Address Offset	See Table 9-221.		
Physical Address	0x4500 202C 0x4500 302C 0x4500 402C 0x4500 502C	**Instance**	CLK2_STATCOLL1 CLK2_STATCOLL3 CLK2_STATCOLL2 CLK2_STATCOLL4
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														EVTMUX_SEL3	

Bits	Field Name	Description	Type	Reset
31:3	RESERVED	Reserved	R	0x0000 0000
2:0	EVTMUX_SEL3	The select of the mux 3 Type: Control. Reset value: 0x0.	RW	0x0

Table 9-249. Register Call Summary for Register L3_STCOL_EVTMUX_SEL3

L3_MAIN Interconnect
- L3_MAIN STATCOLL Register Summary and Description: [0]

Table 9-250. L3_STCOL_DUMP_IDENTIFIER

Address Offset	See Table 9-221.		
Physical Address	0x4500 2040 0x4500 3040 0x4500 4040 0x4500 5040	**Instance**	CLK2_STATCOLL1 CLK2_STATCOLL3 CLK2_STATCOLL2 CLK2_STATCOLL4
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								DUMP_IDENTIFIER							

Bits	Field Name	Description	Type	Reset
31:4	RESERVED	Reserved	R	0x0000 0000
3:0	DUMP_IDENTIFIER	Probe identifier Type: Control. Reset value: 0x0.	R	0x0

Table 9-251. Register Call Summary for Register L3_STCOL_DUMP_IDENTIFIER

L3_MAIN Interconnect

- L3_MAIN STATCOLL Register Summary and Description: [0]

Table 9-252. L3_STCOL_DUMP_COLLECTTIME

Address Offset	See Table 9-221.		
Physical Address	0x4500 2044 0x4500 3044 0x4500 4044 0x4500 5044	**Instance**	CLK2_STATCOLL1 CLK2_STATCOLL3 CLK2_STATCOLL2 CLK2_STATCOLL4
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DUMP_COLLECTTIME																															

Bits	Field Name	Description	Type	Reset
31:0	DUMP_COLLECTTIME	Number of cycle to wait between two statistics frame Type: Control. Reset value: 0x0.	RW	0x0000

Table 9-253. Register Call Summary for Register L3_STCOL_DUMP_COLLECTTIME

L3_MAIN Interconnect

- L3_MAIN STATCOLL Register Summary and Description: [0]

Table 9-254. L3_STCOL_DUMP_SLVADDR

Address Offset	See Table 9-221.		
Physical Address	0x4500 2048 0x4500 3048 0x4500 4048 0x4500 5048	**Instance**	CLK2_STATCOLL1 CLK2_STATCOLL3 CLK2_STATCOLL2 CLK2_STATCOLL4
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								DUMP_SLVADDR							

Bits	Field Name	Description	Type	Reset
31:7	RESERVED	Reserved	R	0x0000000
6:0	DUMP_SLVADDR	Dump slave address Type: Control. Reset value: 0x19.	R	0x19

Table 9-255. Register Call Summary for Register L3_STCOL_DUMP_SLVADDR

L3_MAIN Interconnect

- L3_MAIN STATCOLL Register Summary and Description: [0]

Table 9-256. L3_STCOL_DUMP_MSTADDR

Address Offset	See Table 9-221.		
Physical Address	0x4500 204C 0x4500 304C 0x4500 404C 0x4500 504C	**Instance**	CLK2_STATCOLL1 CLK2_STATCOLL3 CLK2_STATCOLL2 CLK2_STATCOLL4
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								DUMP_MSTADDR							

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Reserved	R	0x000000
7:0	DUMP_MSTADDR	Dump master address Type: Control. Reset value: 0xE0.	R	0x380

Table 9-257. Register Call Summary for Register L3_STCOL_DUMP_MSTADDR

L3_MAIN Interconnect

- L3_MAIN STATCOLL Register Summary and Description: [0]

Table 9-258. L3_STCOL_DUMP_SLVOFS

Address Offset	See Table 9-221.		
Physical Address	0x4500 2050 0x4500 3050 0x4500 4050 0x4500 5050	**Instance**	CLK2_STATCOLL1 CLK2_STATCOLL3 CLK2_STATCOLL2 CLK2_STATCOLL4
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DUMP_SLVOFS																															

Bits	Field Name	Description	Type	Reset
31:0	DUMP_SLVOFS	Dump slave offset Type: Control. Reset value: 0x800.	RW	0x0000 0800

Table 9-259. Register Call Summary for Register L3_STCOL_DUMP_SLVOFS

L3_MAIN Interconnect
- L3_MAIN STATCOLL Register Summary and Description: [0]

Table 9-260. L3_STCOL_DUMP_MODE

Address Offset	See Table 9-221.		
Physical Address	0x4500 2054 0x4500 3054 0x4500 4054 0x4500 5054	**Instance**	CLK2_STATCOLL1 CLK2_STATCOLL3 CLK2_STATCOLL2 CLK2_STATCOLL4
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														DUMP_MODE_CONDITIONAL	DUMP_MODE_MANUAL

Bits	Field Name	Description	Type	Reset
31:2	RESERVED	Reserved	R	0x0000 0000
1	DUMP_MODE_CONDITIONAL	Define the stat conditional dump, if one a dump will be generated when alarm is trigged Type: Control. Reset value: 0x0.	RW	0
0	DUMP_MODE_MANUAL	Define the dump mode: if != 0 the dump is controlled by the Send register. Type: Control. Reset value: 0x0.	RW	0

Table 9-261. Register Call Summary for Register L3_STCOL_DUMP_MODE

L3_MAIN Interconnect
- L3_MAIN STATCOLL Register Summary and Description: [0]

Table 9-262. L3_STCOL_DUMP_SEND

Address Offset	See Table 9-221.		
Physical Address	0x4500 2058 0x4500 3058 0x4500 4058 0x4500 5058	**Instance**	CLK2_STATCOLL1 CLK2_STATCOLL3 CLK2_STATCOLL2 CLK2_STATCOLL4
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															DUMP_SEND

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved	R	0x0000 0000
0	DUMP_SEND	In manual mode, is used to send the dump content and initialize the counters. Type: Give_AutoCleared. Reset value: 0x0. • Dumping can be performed only if monitoring is enabled • For “one shot metrics dump” the DUMP_SEND command has to be issued before disabling monitoring	RW	0

Table 9-263. Register Call Summary for Register L3_STCOL_DUMP_SEND

L3_MAIN Interconnect
- L3_MAIN STATCOLL Register Summary and Description: [0]

Table 9-264. L3_STCOL_DUMP_DISABLE

Address Offset	See Table 9-221.		
Physical Address	0x4500 205C 0x4500 305C 0x4500 405C 0x4500 505C	**Instance**	CLK2_STATCOLL1 CLK2_STATCOLL3 CLK2_STATCOLL2 CLK2_STATCOLL4
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															DUMP_DISABLE

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	DUMP_DISABLE	If 1, the dump frame will be disabled, but counters still active.This is typically used when counters monitoring is enabled Type: Control. Reset value: 0x0.	RW	0

Table 9-265. Register Call Summary for Register L3_STCOL_DUMP_DISABLE

L3_MAIN Interconnect • L3_MAIN STATCOLL Register Summary and Description: [0]

Table 9-266. L3_STCOL_DUMP_ALARM_TRIG

Address Offset	See Table 9-221.		
Physical Address	0x4500 2060 0x4500 3060 0x4500 4060 0x4500 5060	**Instance**	CLK2_STATCOLL1 CLK2_STATCOLL3 CLK2_STATCOLL2 CLK2_STATCOLL4
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															DUMP_ALARM_TRIG

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0bxxx xxxx xxxx xxxx xxxx xxxx xxxx xxxx
0	DUMP_ALARM_TRIG	In Alarm Mode, is used to reset Alarm Type: Take. Reset value: 0x0.	RW	0

Table 9-267. Register Call Summary for Register L3_STCOL_DUMP_ALARM_TRIG

L3_MAIN Interconnect • L3_MAIN STATCOLL Register Summary and Description: [0]

Table 9-268. L3_STCOL_DUMP_ALARM_MINVAL

Address Offset	See Table 9-221.		
Physical Address	0x4500 2064 0x4500 3064 0x4500 4064 0x4500 5064	**Instance**	CLK2_STATCOLL1 CLK2_STATCOLL3 CLK2_STATCOLL2 CLK2_STATCOLL4
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DUMP_ALARM_MINVAL																															

Bits	Field Name	Description	Type	Reset
31:0	DUMP_ALARM_MINVAL	In Alarm Mode, used to trig an alert if any of counter value is less than AlarmMinVal Type: Control. Reset value: 0x0.	RW	0x0000 0000

Table 9-269. Register Call Summary for Register L3_STCOL_DUMP_ALARM_MINVAL

L3_MAIN Interconnect

- L3_MAIN STATCOLL Register Summary and Description: [0]

Table 9-270. L3_STCOL_DUMP_ALARM_MAXVAL

Address Offset	See Table 9-221.		
Physical Address	0x4500 2068 0x4500 3068 0x4500 4068 0x4500 5068	**Instance**	CLK2_STATCOLL1 CLK2_STATCOLL3 CLK2_STATCOLL2 CLK2_STATCOLL4
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DUMP_ALARM_MAXVAL																															

Bits	Field Name	Description	Type	Reset
31:0	DUMP_ALARM_MAXVAL	In Alarm Mode, used to trig an alert if any of counter value is larger or equal to AlarmMaxVal Type: Control. Reset value: 0x0.	RW	0x0000 0000

Table 9-271. Register Call Summary for Register L3_STCOL_DUMP_ALARM_MAXVAL

L3_MAIN Interconnect

- L3_MAIN STATCOLL Register Summary and Description: [0]

Table 9-272. L3_STCOL_DUMP_ALARM_MODE0

Address Offset	See Table 9-221.		
Physical Address	0x4500 206C 0x4500 306C 0x4500 406C 0x4500 506C	**Instance**	CLK2_STATCOLL1 CLK2_STATCOLL3 CLK2_STATCOLL2 CLK2_STATCOLL4
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														DUMP_ALARM_MODE0	

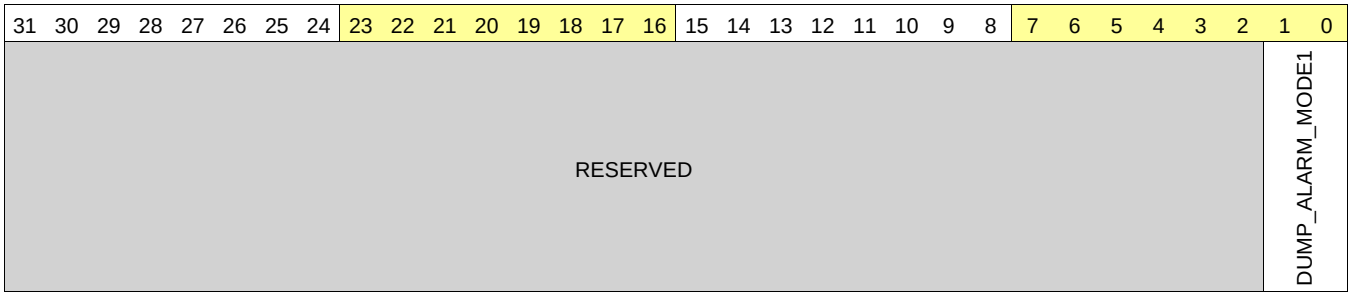
Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0000 0000
1:0	DUMP_ALARM_MODE0	Alarm Mode off/min/max/both Type: Control. Reset value: 0x0. 0x0: OFF 0x1: MIN 0x3: MAX 0x2: BOTH	RW	0x0

Table 9-273. Register Call Summary for Register L3_STCOL_DUMP_ALARM_MODE0

L3_MAIN Interconnect
- L3_MAIN STATCOLL Register Summary and Description: [0]

Table 9-274. L3_STCOL_DUMP_ALARM_MODE1

Address Offset	See Table 9-221.		
Physical Address	0x4500 2070 0x4500 3070 0x4500 4070 0x4500 5070	**Instance**	CLK2_STATCOLL1 CLK2_STATCOLL3 CLK2_STATCOLL2 CLK2_STATCOLL4
Description			
Type	RW		



Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0000 0000
1:0	DUMP_ALARM_MODE1	Alarm Mode off/min/max/both Type: Control. Reset value: 0x0. 0x0: OFF 0x1: MIN 0x3: MAX 0x2: BOTH	RW	0x0

Table 9-275. Register Call Summary for Register L3_STCOL_DUMP_ALARM_MODE1

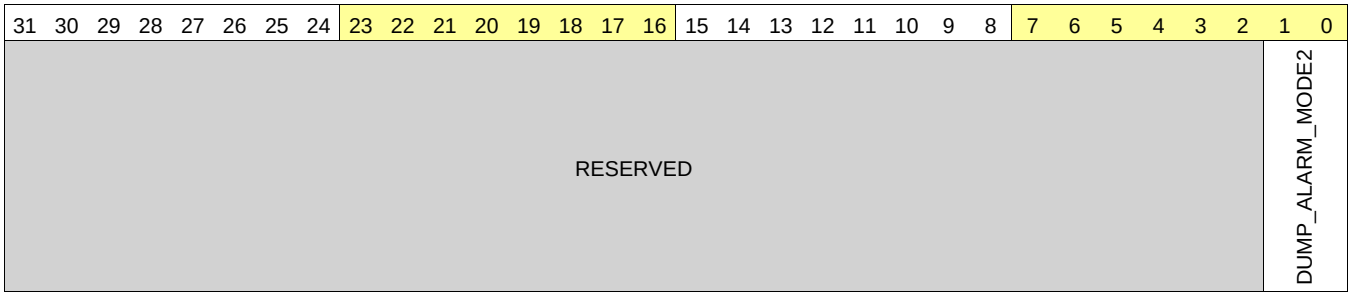
L3_MAIN Interconnect
- L3_MAIN STATCOLL Register Summary and Description: [0]

Table 9-276. L3_STCOL_DUMP_ALARM_MODE2

Address Offset	See Table 9-221.		
Physical Address	0x4500 2074 0x4500 3074 0x4500 4074 0x4500 5074	**Instance**	CLK2_STATCOLL1 CLK2_STATCOLL3 CLK2_STATCOLL2 CLK2_STATCOLL4

Table 9-276. L3_STCOL_DUMP_ALARM_MODE2 (continued)

Description	
Type	RW



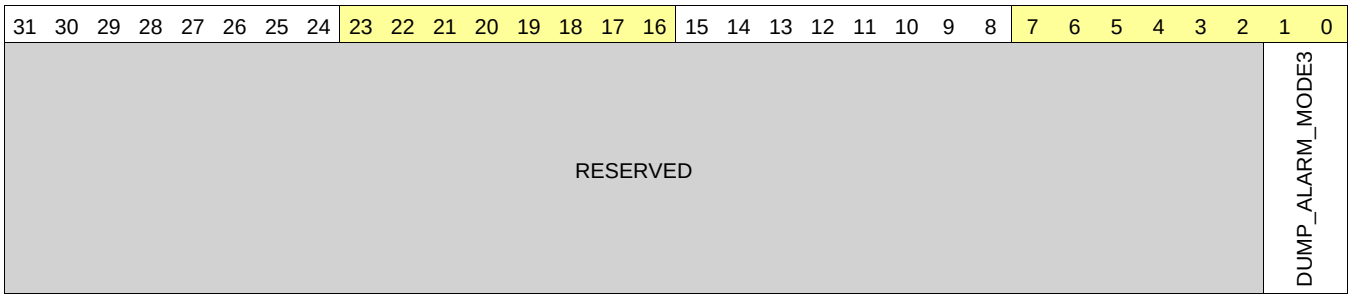
Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0000 0000
1:0	DUMP_ALARM_MODE2	Alarm Mode off/min/max/both Type: Control. Reset value: 0x0. 0x0: OFF 0x1: MIN 0x3: MAX 0x2: BOTH	RW	0x0

Table 9-277. Register Call Summary for Register L3_STCOL_DUMP_ALARM_MODE2

L3_MAIN Interconnect
- L3_MAIN STATCOLL Register Summary and Description: [0]

Table 9-278. L3_STCOL_DUMP_ALARM_MODE3

Address Offset	See Table 9-221.		
Physical Address	0x4500 2078 0x4500 3078 0x4500 4078 0x4500 5078	**Instance**	CLK2_STATCOLL1 CLK2_STATCOLL3 CLK2_STATCOLL2 CLK2_STATCOLL4
Description			
Type	RW		



Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0000 0000
1:0	DUMP_ALARM_MODE3	Alarm Mode off/min/max/both Type: Control. Reset value: 0x0. 0x0: OFF 0x1: MIN 0x3: MAX 0x2: BOTH	RW	0x0

Table 9-279. Register Call Summary for Register L3_STCOL_DUMP_ALARM_MODE3

L3_MAIN Interconnect
- L3_MAIN STATCOLL Register Summary and Description: [0]

Table 9-280. L3_STCOL_DUMP_CNT0

Address Offset	See Table 9-221.		
Physical Address	0x4500 208C 0x4500 308C 0x4500 408C 0x4500 508C	**Instance**	CLK2_STATCOLL1 CLK2_STATCOLL3 CLK2_STATCOLL2 CLK2_STATCOLL4
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DUMP_CNT0																															

Bits	Field Name	Description	Type	Reset
31:0	DUMP_CNT0	Dump counter value Type: Status. Reset value: X.	R	0x0

Table 9-281. Register Call Summary for Register L3_STCOL_DUMP_CNT0

L3_MAIN Interconnect
- L3_MAIN STATCOLL Register Summary and Description: [0]

Table 9-282. L3_STCOL_DUMP_CNT1

Address Offset	See Table 9-221.		
Physical Address	0x4500 2090 0x4500 3090 0x4500 4090 0x4500 5090	**Instance**	CLK2_STATCOLL1 CLK2_STATCOLL3 CLK2_STATCOLL2 CLK2_STATCOLL4
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DUMP_CNT1																															

Bits	Field Name	Description	Type	Reset
31:0	DUMP_CNT1	Dump counter value Type: Status. Reset value: X.	R	0x0

Table 9-283. Register Call Summary for Register L3_STCOL_DUMP_CNT1

L3_MAIN Interconnect

- L3_MAIN STATCOLL Register Summary and Description: [0]

Table 9-284. L3_STCOL_DUMP_CNT2

Address Offset	See Table 9-221.		
Physical Address	0x4500 2094 0x4500 3094 0x4500 4094 0x4500 5094	**Instance**	CLK2_STATCOLL1 CLK2_STATCOLL3 CLK2_STATCOLL2 CLK2_STATCOLL4
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DUMP_CNT2																															

Bits	Field Name	Description	Type	Reset
31:0	DUMP_CNT2	Dump counter value Type: Status. Reset value: X.	R	0x0

Table 9-285. Register Call Summary for Register L3_STCOL_DUMP_CNT2

L3_MAIN Interconnect

- L3_MAIN STATCOLL Register Summary and Description: [0]

Table 9-286. L3_STCOL_DUMP_CNT3

Address Offset	See Table 9-221.		
Physical Address	0x4500 2098 0x4500 3098 0x4500 4098 0x4500 5098	**Instance**	CLK2_STATCOLL1 CLK2_STATCOLL3 CLK2_STATCOLL2 CLK2_STATCOLL4
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DUMP_CNT3																															

Bits	Field Name	Description	Type	Reset
31:0	DUMP_CNT3	Dump counter value Type: Status. Reset value: X.	R	0x0

Table 9-287. Register Call Summary for Register L3_STCOL_DUMP_CNT3

L3_MAIN Interconnect

- L3_MAIN STATCOLL Register Summary and Description: [0]

Table 9-288. L3_STCOL_FILTER_i_GLOBALEN

Address Offset	See Table 9-221.		
Physical Address	0x4500 20AC + (0x158*i) 0x4500 30AC + (0x158*i) 0x4500 40AC + (0x158*i) 0x4500 50AC + (0x158*i)	**Instance**	CLK2_STATCOLL1 CLK2_STATCOLL3 CLK2_STATCOLL2 CLK2_STATCOLL4
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															FILTER_i_GLOBALEN

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved	R	0x0000 0000
0	FILTER_i_GLOBALEN	Filter global enable Type: Control. Reset value: 0x0.	RW	0

Table 9-289. Register Call Summary for Register L3_STCOL_FILTER_i_GLOBALEN

L3_MAIN Interconnect

- Statistic Collectors Group: [0]
- L3_MAIN STATCOLL Register Summary and Description: [1]

Table 9-290. L3_STCOL_FILTER_i_ADDRMIN

Address Offset	See Table 9-221.		
Physical Address	0x4500 20B0 + (0x158*i) 0x4500 30B0 + (0x158*i) 0x4500 40B0 + (0x158*i) 0x4500 50B0 + (0x158*i)	**Instance**	CLK2_STATCOLL1 CLK2_STATCOLL3 CLK2_STATCOLL2 CLK2_STATCOLL4
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								FILTER0_ADDRMIN																							

Bits	Field Name	Description	Type	Reset
31:23	RESERVED		R	0x0000 0000
22:0	FILTER0_ADDRMIN	Min addr range Type: Control. Reset value: 0x0.	RW	0x00 0000

Table 9-291. Register Call Summary for Register L3_STCOL_FILTER_i_ADDRMIN

L3_MAIN Interconnect

- L3_MAIN STATCOLL Register Summary and Description: [0]

Table 9-292. L3_STCOL_FILTER_i_ADDRMAX

Address Offset	See Table 9-221.		
Physical Address	0x4500 20B4 + (0x158*i) 0x4500 30B4 + (0x158*i) 0x4500 40B4 + (0x158*i) 0x4500 50B4 + (0x158*i)	**Instance**	CLK2_STATCOLL1 CLK2_STATCOLL3 CLK2_STATCOLL2 CLK2_STATCOLL4
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								FILTER0_ADDRMAX																							

Bits	Field Name	Description	Type	Reset
31:23	RESERVED		R	0x0000 0000
22:0	FILTER0_ADDRMAX	Max addr range Type: Control. Reset value: 0x0.	RW	0x00 0000

Table 9-293. Register Call Summary for Register L3_STCOL_FILTER_i_ADDRMAX

L3_MAIN Interconnect

- L3_MAIN STATCOLL Register Summary and Description: [0]

Table 9-294. L3_STCOL_FILTER_i_ADDREN

Address Offset	See Table 9-221.		
Physical Address	0x4500 20B8 + (0x158*i) 0x4500 30B8 + (0x158*i) 0x4500 40B8 + (0x158*i) 0x4500 50B8 + (0x158*i)	**Instance**	CLK2_STATCOLL1 CLK2_STATCOLL3 CLK2_STATCOLL2 CLK2_STATCOLL4
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															FILTER0_ADDREN

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0bxxx xxxx xxxx xxxx xxxx xxxx xxxx xxxx
0	FILTER0_ADDREN	max filtering enable Type: Control. Reset value: 0x0.	RW	0

Table 9-295. Register Call Summary for Register L3_STCOL_FILTER_i_ADDREN

L3_MAIN Interconnect

- L3_MAIN STATCOLL Register Summary and Description: [0]

Table 9-296. L3_STCOL_FILTER_i_EN0

Address Offset	See Table 9-221.		
Physical Address	0x4500 20BC + (0x158*i) 0x4500 30BC + (0x158*i) 0x4500 40BC + (0x158*i) 0x4500 50BC + (0x158*i)	**Instance**	CLK2_STATCOLL1 CLK2_STATCOLL3 CLK2_STATCOLL2 CLK2_STATCOLL4
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															FILTER_i_EN0

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved	R	0x0000 0000
0	FILTER_i_EN0	Enable filter stage 0 Type: Control. Reset value: 0x0.	RW	0

Table 9-297. Register Call Summary for Register L3_STCOL_FILTER_i_EN0

L3_MAIN Interconnect

- L3_MAIN STATCOLL Register Summary and Description: [0]

Table 9-298. L3_STCOL_FILTER_i_MASK0_MSTADDR

Address Offset	See Table 9-221.		
Physical Address	0x4500 20C8 + (0x158*i) 0x4500 30C8 + (0x158*i) 0x4500 40C8 + (0x158*i) 0x4500 50C8 + (0x158*i)	**Instance**	CLK2_STATCOLL1 CLK2_STATCOLL3 CLK2_STATCOLL2 CLK2_STATCOLL4
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								FILTER_i_MASK0_MSTADDR							

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Reserved	R	0x000000
7:0	FILTER_i_MASK0_MSTADDR	Mask/Match of MstAddr Type: Control. Reset value: 0x0.	RW	0x00

Table 9-299. Register Call Summary for Register L3_STCOL_FILTER_i_MASK0_MSTADDR

L3_MAIN Interconnect

- L3_MAIN STATCOLL Register Summary and Description: [0]

Table 9-300. L3_STCOL_FILTER_i_MASK0_RD

Address Offset	See Table 9-221.		
Physical Address	0x4500 20C0 + (0x158*i) 0x4500 30C0 + (0x158*i) 0x4500 40C0 + (0x158*i) 0x4500 50C0 + (0x158*i)	**Instance**	CLK2_STATCOLL1 CLK2_STATCOLL3 CLK2_STATCOLL2 CLK2_STATCOLL4
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															FILTER_i_MASK0_RD

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved	R	0x0000 0000
0	FILTER_i_MASK0_RD	Mask/Match of Rd Type: Control. Reset value: 0x0.	RW	0

Table 9-301. Register Call Summary for Register L3_STCOL_FILTER_i_MASK0_RD

L3_MAIN Interconnect

- L3_MAIN STATCOLL Register Summary and Description: [0]

Table 9-302. L3_STCOL_FILTER_i_MASK0_WR

Address Offset	See Table 9-221.		
Physical Address	0x4500 20C4 + (0x158*i) 0x4500 30C4 + (0x158*i) 0x4500 40C4 + (0x158*i) 0x4500 50C4 + (0x158*i)	**Instance**	CLK2_STATCOLL1 CLK2_STATCOLL3 CLK2_STATCOLL2 CLK2_STATCOLL4
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															FILTER_i_MASK0_WR

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved	R	0x0000 0000
0	FILTER_i_MASK0_WR	Mask/Match of Wr Type: Control. Reset value: 0x0.	RW	0

Table 9-303. Register Call Summary for Register L3_STCOL_FILTER_i_MASK0_WR

L3_MAIN Interconnect

- L3_MAIN STATCOLL Register Summary and Description: [0]

Table 9-304. L3_STCOL_FILTER_i_MASK0_ERR

Address Offset	See Table 9-221.		
Physical Address	0x4500 20D0 + (0x158*i) 0x4500 30D0 + (0x158*i) 0x4500 40D0 + (0x158*i) 0x4500 50D0 + (0x158*i)	**Instance**	CLK2_STATCOLL1 CLK2_STATCOLL3 CLK2_STATCOLL2 CLK2_STATCOLL4
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															FILTER_i_MASK0_ERR

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved	R	0x0000 0000
0	FILTER_i_MASK0_ERR	Mask/Match of Err Type: Control. Reset value: 0x0.	RW	0

Table 9-305. Register Call Summary for Register L3_STCOL_FILTER_i_MASK0_ERR

L3_MAIN Interconnect

- L3_MAIN STATCOLL Register Summary and Description: [0]

Table 9-306. L3_STCOL_FILTER_i_MASK0_SLVADDR

Address Offset	See Table 9-221.		
Physical Address	0x4500 20CC + (0x158*i) 0x4500 30CC + (0x158*i) 0x4500 40CC + (0x158*i) 0x4500 50CC + (0x158*i)	**Instance**	CLK2_STATCOLL1 CLK2_STATCOLL3 CLK2_STATCOLL2 CLK2_STATCOLL4
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															FILTER_i_MASK0_SLVADDR

Bits	Field Name	Description	Type	Reset
31:7	RESERVED		R	0x0000 0000
6:0	FILTER_i_MASK0_SLVADDR	Mask/Match of SlvAddr Type: Control. Reset value: 0x0.	RW	0x00

Table 9-307. Register Call Summary for Register L3_STCOL_FILTER_i_MASK0_SLVADDR

L3_MAIN Interconnect

- L3_MAIN STATCOLL Register Summary and Description: [0]

Table 9-308. L3_STCOL_FILTER_i_MASK0_REQUSERINFO

Address Offset	See Table 9-221.		
Physical Address	0x4500 20D4 + (0x158*i) 0x4500 30D4 + (0x158*i) 0x4500 40D4 + (0x158*i) 0x4500 50D4 + (0x158*i)	**Instance**	CLK2_STATCOLL1 CLK2_STATCOLL3 CLK2_STATCOLL2 CLK2_STATCOLL4
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								FILTER_i_MASK0_REQUSERINFO																									

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0000 0000
27:0	FILTER_i_MASK0_REQUSERINFO	Mask/Match of ReqUserInfo Type: Control. Reset value: 0x0.	RW	0x00

Table 9-309. Register Call Summary for Register L3_STCOL_FILTER_i_MASK0_REQUSERINFO

L3_MAIN Interconnect

- L3_MAIN STATCOLL Register Summary and Description: [0]

Table 9-310. L3_STCOL_FILTER_i_MASK0_RSPUSERINFO

Address Offset	See Table 9-221.		
Physical Address	0x4500 20D8 + (0x158*i) 0x4500 30D8 + (0x158*i) 0x4500 40D8 + (0x158*i) 0x4500 50D8 + (0x158*i)	**Instance**	CLK2_STATCOLL1 CLK2_STATCOLL3 CLK2_STATCOLL2 CLK2_STATCOLL4
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														FILTER_i_MASK0_RSPUSERINFO	

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0000 0000
2:0	FILTER_i_MASK0_RSPUSERINFO	Mask/Match of RspUserInfo Type: Control. Reset value: 0x0.	RW	0x0

Table 9-311. Register Call Summary for Register L3_STCOL_FILTER_i_MASK0_RSPUSERINFO

L3_MAIN Interconnect
- L3_MAIN STATCOLL Register Summary and Description: [0]

Table 9-312. L3_STCOL_FILTER_i_MATCH0_MSTADDR

Address Offset	See Table 9-221.		
Physical Address	0x4500 20E8 + (0x158*i) 0x4500 30E8 + (0x158*i) 0x4500 40E8 + (0x158*i) 0x4500 50E8 + (0x158*i)	**Instance**	CLK2_STATCOLL1 CLK2_STATCOLL3 CLK2_STATCOLL2 CLK2_STATCOLL4
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								FILTER_i_MATCH0_MSTADDR							

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Reserved	R	0x000000
7:0	FILTER_i_MATCH0_MSTADDR	Mask/Match of MstAddr Type: Control. Reset value: 0x0.	RW	0x00

Table 9-313. Register Call Summary for Register L3_STCOL_FILTER_i_MATCH0_MSTADDR

L3_MAIN Interconnect
- L3_MAIN STATCOLL Register Summary and Description: [0]

Table 9-314. L3_STCOL_FILTER_i_MATCH0_SLVADDR

Address Offset	See Table 9-221.		
Physical Address	0x4500 20EC + (0x158*i) 0x4500 30EC + (0x158*i) 0x4500 40EC + (0x158*i) 0x4500 50EC + (0x158*i)	**Instance**	CLK2_STATCOLL1 CLK2_STATCOLL3 CLK2_STATCOLL2 CLK2_STATCOLL4
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																													FILTER0_MATCH0_SLVADDR		

Bits	Field Name	Description	Type	Reset
31:5	RESERVED		R	0x0000 0000
4:0	FILTER0_MATCH0_SLVADDR	Mask/Match of SlvAddr Type: Control. Reset value: 0x0.	RW	0x00

Table 9-315. Register Call Summary for Register L3_STCOL_FILTER_i_MATCH0_SLVADDR

L3_MAIN Interconnect

- L3_MAIN STATCOLL Register Summary and Description: [0]

Table 9-316. L3_STCOL_FILTER_i_MATCH0_RD

Address Offset	See Table 9-221.		
Physical Address	0x4500 20E0 + (0x158*i) 0x4500 30E0 + (0x158*i) 0x4500 40E0 + (0x158*i) 0x4500 50E0 + (0x158*i)	**Instance**	CLK2_STATCOLL1 CLK2_STATCOLL3 CLK2_STATCOLL2 CLK2_STATCOLL4
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															FILTER_i_MATCH0_RD

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved	R	0x0000 0000
0	FILTER_i_MATCH0_RD	Mask/Match of Rd Type: Control. Reset value: 0x0.	RW	0

Table 9-317. Register Call Summary for Register L3_STCOL_FILTER_i_MATCH0_RD

L3_MAIN Interconnect

- L3_MAIN STATCOLL Register Summary and Description: [0]

Table 9-318. L3_STCOL_FILTER_i_MATCH0_WR

Address Offset	See Table 9-221.		
Physical Address	0x4500 20E4 + (0x158*i) 0x4500 30E4 + (0x158*i) 0x4500 40E4 + (0x158*i) 0x4500 50E4 + (0x158*i)	**Instance**	CLK2_STATCOLL1 CLK2_STATCOLL3 CLK2_STATCOLL2 CLK2_STATCOLL4
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															FILTER_i_MATCH0_WR

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved	R	0x0000 0000
0	FILTER_i_MATCH0_WR	Mask/Match of Wr Type: Control. Reset value: 0x0.	RW	0

Table 9-319. Register Call Summary for Register L3_STCOL_FILTER_i_MATCH0_WR

L3_MAIN Interconnect

- L3_MAIN STATCOLL Register Summary and Description: [0]

Table 9-320. L3_STCOL_FILTER_i_MATCH0_ERR

Address Offset	See Table 9-221.		
Physical Address	0x4500 20F0 + (0x158*i) 0x4500 30F0 + (0x158*i) 0x4500 40F0 + (0x158*i) 0x4500 50F0 + (0x158*i)	**Instance**	CLK2_STATCOLL1 CLK2_STATCOLL3 CLK2_STATCOLL2 CLK2_STATCOLL4
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															FILTER_i_MATCH0_ERR

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved	R	0x0000 0000
0	FILTER_i_MATCH0_ERR	Mask/Match of Err Type: Control. Reset value: 0x0.	RW	0

Table 9-321. Register Call Summary for Register L3_STCOL_FILTER_i_MATCH0_ERR

L3_MAIN Interconnect

- L3_MAIN STATCOLL Register Summary and Description: [0]

Table 9-322. L3_STCOL_FILTER_i_MATCH0_REQUSERINFO

Address Offset	See Table 9-221.		
Physical Address	0x4500 20F4 + (0x158*i) 0x4500 30F4 + (0x158*i) 0x4500 40F4 + (0x158*i) 0x4500 50F4 + (0x158*i)	**Instance**	CLK2_STATCOLL1 CLK2_STATCOLL3 CLK2_STATCOLL2 CLK2_STATCOLL4
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								FILTER_i_MASK0_REQUSERINFO																									

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0000 0000
27:0	FILTER_i_MASK0_REQUSERINFO	Mask/Match of ReqUserInfo Type: Control. Reset value: 0x0.	RW	0x00

Table 9-323. Register Call Summary for Register L3_STCOL_FILTER_i_MATCH0_REQUSERINFO

L3_MAIN Interconnect

- L3_MAIN STATCOLL Register Summary and Description: [0]

Table 9-324. L3_STCOL_FILTER_i_MATCH0_RSPUSERINFO

Address Offset	See Table 9-221.		
Physical Address	0x4500 20F8 + (0x158*i) 0x4500 30F8 + (0x158*i) 0x4500 40F8 + (0x158*i) 0x4500 50F8 + (0x158*i)	**Instance**	CLK2_STATCOLL1 CLK2_STATCOLL3 CLK2_STATCOLL2 CLK2_STATCOLL4
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														FILTER_i_MASK0_RSPUSERINFO	

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0000 0000
2:0	FILTER_i_MASK0_RSPUSERINFO	Mask/Match of RspUserInfo Type: Control. Reset value: 0x0.	RW	0x0

Table 9-325. Register Call Summary for Register L3_STCOL_FILTER_i_MATCH0_RSPUSERINFO

L3_MAIN Interconnect
- L3_MAIN STATCOLL Register Summary and Description: [0]

Table 9-326. L3_STCOL_OP_i_THRESHOLD_MINVAL

Address Offset See Table 9-221.

Physical Address 0x4500 21F0 + (0x158*i) — **Instance** CLK2_STATCOLL1
0x4500 31F0 + (0x158*i) — CLK2_STATCOLL3
0x4500 41F0 + (0x158*i) — CLK2_STATCOLL2
0x4500 51F0 + (0x158*i) — CLK2_STATCOLL4

Description

Type RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																OP_i_THRESHOLD_MINVAL															

Bits	Field Name	Description	Type	Reset
31:13	RESERVED	Reserved	R	0x00000
12:0	OP_i_THRESHOLD_MINVAL	Min value Type: Control. Reset value: 0x0.	RW	0x000

Table 9-327. Register Call Summary for Register L3_STCOL_OP_i_THRESHOLD_MINVAL

L3_MAIN Interconnect
- L3_MAIN STATCOLL Register Summary and Description: [0]

Table 9-328. L3_STCOL_OP_i_THRESHOLD_MAXVAL

Address Offset See Table 9-221.

Physical Address 0x4500 21F4 + (0x158*i) — **Instance** CLK2_STATCOLL1
0x4500 31F4 + (0x158*i) — CLK2_STATCOLL3
0x4500 41F4 + (0x158*i) — CLK2_STATCOLL2
0x4500 51F4 + (0x158*i) — CLK2_STATCOLL4

Description

Type RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																OP_i_THRESHOLD_MAXVAL															

Bits	Field Name	Description	Type	Reset
31:13	RESERVED	Reserved	R	0x00000
12:0	OP_i_THRESHOLD_MAXVAL	Max value Type: Control. Reset value: 0x0.	RW	0x000

Table 9-329. Register Call Summary for Register L3_STCOL_OP_i_THRESHOLD_MAXVAL

L3_MAIN Interconnect
- L3_MAIN STATCOLL Register Summary and Description: [0]

Table 9-330. L3_STCOL_OP_i_EVTINFOSEL

Address Offset	See Table 9-221.		
Physical Address	0x4500 21F8 + (0x158*i) 0x4500 31F8 + (0x158*i) 0x4500 41F8 + (0x158*i) 0x4500 51F8 + (0x158*i)	**Instance**	CLK2_STATCOLL1 CLK2_STATCOLL3 CLK2_STATCOLL2 CLK2_STATCOLL4
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														OP_i_EVTINFOSEL	

Bits	Field Name	Description	Type	Reset
31:2	RESERVED	Reserved	R	0x0000 0000
1:0	OP_i_EVTINFOSEL	Select event info data to add to counter (len/press or latency) Type: Control. Reset value: 0x0. 0x0: Select len from event info list 0x1: Select pressure if available from event info list 0x2: Select latency if available from event info list	RW	0x0

Table 9-331. Register Call Summary for Register L3_STCOL_OP_i_EVTINFOSEL

L3_MAIN Interconnect
- L3_MAIN STATCOLL Register Summary and Description: [0]

Table 9-332. L3_STCOL_OP_i_SEL

Address Offset	See Table 9-221.		
Physical Address	0x4500 21FC + (0x158*i) 0x4500 31FC + (0x158*i) 0x4500 41FC + (0x158*i) 0x4500 51FC + (0x158*i)	**Instance**	CLK2_STATCOLL1 CLK2_STATCOLL3 CLK2_STATCOLL2 CLK2_STATCOLL4
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												OP_i_SEL			

Bits	Field Name	Description	Type	Reset
31:4	RESERVED	Reserved	R	0x0000000
3:0	OP_i_SEL	Select logical operation Type: Control. Reset value: 0x0. 0x0: Increment counter on each mask/match filter hit 0x1: Increment counter on each min/max level hit 0x2: Add to counter the selected event info value (len/press or latency) 0x3: increment counter when all filter event hits (And(Fi)) 0x4: Increment counter if any of filter event hits (Or(Fi)) 0x5: Add to counter the number of current request event that hit 0x6: Add to counter the number of current response event that hit 0x7: Add to counter the number of all event that hit 0x8:Increment counter on each selected external event hit	RW	0x0

Table 9-333. Register Call Summary for Register L3_STCOL_OP_i_SEL

L3_MAIN Interconnect
- L3_MAIN STATCOLL Register Summary and Description: [0]

9.3 L4 Interconnects

This section details the device L4 interconnects.

9.3.1 L4 Interconnect Overview

The device uses three separate L4 interconnect structures to connect peripheral modules. All L4s handle transfers with peripherals but are located in distinct power domains.

Figure 9-9 is an overview of the L4 interconnects and the peripherals attached to them.

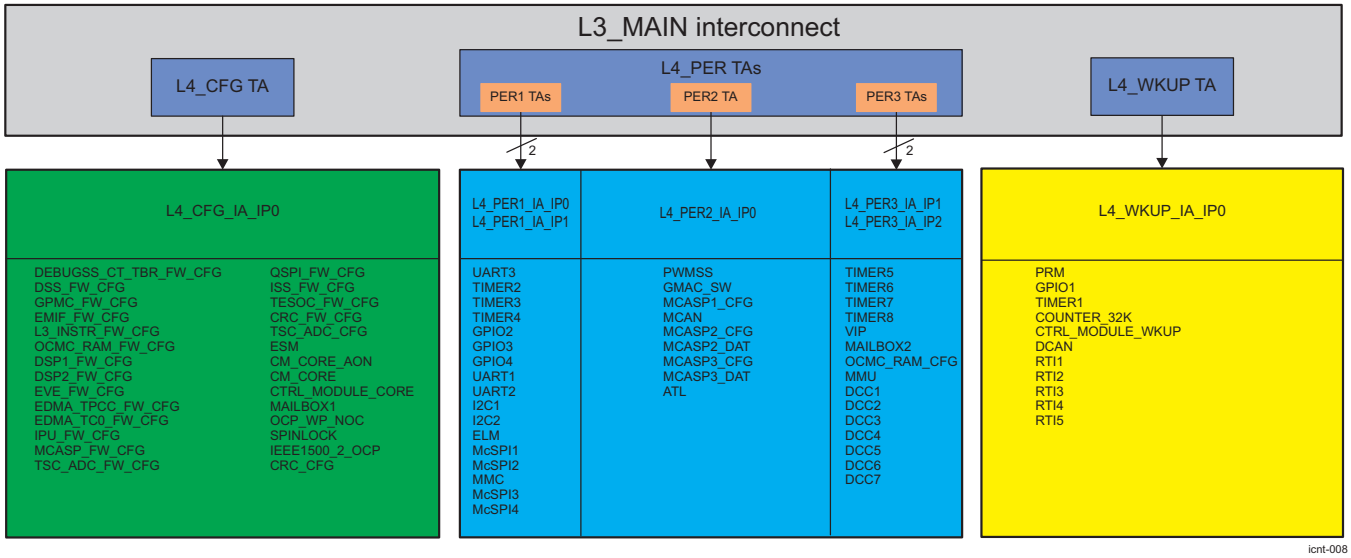
The L4 interconnect is composed of the following interconnects:

- L4_CFG: Includes the majority of the configuration interface for L3_MAIN system modules and peripheral interconnect
- L4 PER: Includes the main peripherals in the device. L4_PER is further divided into three sub-interconnects: L4_PER1, L4_PER2 and L4_PER3. Each of these L4_PERx sub-interconnects is connected to L3_MAIN initiators via the following ports:
  - L4_PER1_P1, L4_PER1_P2
  - L4_PER2_P1
  - L4_PER3_P1, L4_PER3_P2

  L3_MAIN initiators access the L4_PERx peripherals through these ports. All peripherals attached to L4_PER1and L4_PER3 are visible through the two L4_PER1 and L4_PER3 ports. For information about which initiator can access which L4_PERx port, see Figure 9-3.
- L4_WKUP: Includes peripherals attached to the WKUP power domain

NOTE: ISS, CRC, TESOC, ESM and DCC1 to DCC7 are not supported on the DRA78x family of devices.

Figure 9-9. L4 Interconnect Overview



The main features of the L4 interconnects are:

- Seven ports from L3_MAIN interconnect onto 5 parallel L4 interconnects
- One or two 32-bit initiator ports for each L4 interconnect instance (7 in total)
- 8-, 16-, or 32-bit data, single, or burst transactions
- Little-endian platform
- Non-blocking architecture with fair arbitration between masters
- Target interfaces: Fully synchronous or divided synchronous clock frequencies

- L4_CFG and L4_PER1, L4_PER2, L4_PER3 frequency equals half of L3 frequency
- Protection logic that provides user-configurable access control to targets by each initiator

9.3.2 L4 Interconnect Integration

Table 9-334 and Table 9-335 summarize the integration of the module in the device.

Table 9-334. Integration Attributes

Module Instance	Attributes
	Power Domain
L4_PER1, L4_PER2, L4_PER3	PD_COREAON
L4_CFG	PD_COREAON
L4_WKUP	PD_WKUPAON

Table 9-335. Clocks and Resets

Clocks				
Module Instance	**Destination Signal Name**	**Source Signal Name**	**Source**	**Description**
L4_PER1	L4_PER1_CLK	L4PER_L3_GICLK	PRCM module	Functional and interface clock
L4_PER2	L4_PER2_CLK	L4PER_L3_GICLK	PRCM module	Functional and interface clock
L4_PER3	L4_PER3_CLK	L4PER_L3_GICLK	PRCM module	Functional and interface clock
L4_CFG	L4_CFG_CLK	L4CFG_L3_GICLK	PRCM module	Functional and interface clock
L4_WKUP	L4_WKUP_CLK	WKUPAON_GICLK	PRCM module	Functional and interface clock
Resets				
Module Instance	**Destination Signal Name**	**Source Signal Name**	**Source**	**Description**
L4_PER1, L4_PER2, L4_PER3	L4_PER_RST	L4_PER_RST	PRCM module	Reset of L4_PERx interconnect
	L4_PER_RET_RST	L4_PER_PWRON_RET_RST	PRCM module	Reset of L4_PERx interconnect retention registers. For information about retention reset, see Chapter 3, *Power, Reset, and Clock Management*.
L4_CFG	L4_CFG_RST	CORE_RST	PRCM module	Reset of L4_CFG interconnect.
	L4_CFG_RET_RST	CORE_PWRON_RET_RST	PRCM module	Reset of L4_CFG interconnect retention registers. For information about retention reset, see Chapter 3, *Power, Reset, and Clock Management*.
L4_WKUP	L4_WKUP_RST	WKUPAON_RST	PRCM module	Reset of L4_WKUP interconnect
	L4_WKUP_RET_RST	L4_WKUP_RET_RST	PRCM module	Reset of L4_WKUP interconnect. For information about retention reset, see Chapter 3, *Power, Reset, and Clock Management*.

9.3.3 L4 Interconnect Functional Description

9.3.3.1 Module Distribution

IAs and TAs provide the interface to connect the different modules to their associated interconnect.

Table 9-336 through Table 9-345 list all the modules and subsystems with their associated agents. The agents are listed for each L4 interconnect domain.

9.3.3.1.1 L4_PER1 Interconnect Agents

The L4_PER1 interconnect handles transfers only to peripherals in the PER power domain. Table 9-336 lists the L4_PER1 TAs.

Table 9-336. L4_PER1 TAs

Module Target Name	Description
UART3_TARG	Universal Asynchronous Receiver/Transmitter target port
TIMER2_TARG	General Purpose Timer 2 target port
TIMER3_TARG	TIMER3 target port
TIMER4_TARG	TIMER4 target port
GPIO2_TARG	GPIO2 target port
GPIO3_TARG	GPIO3 target port
GPIO4_TARG	GPIO4 target port
UART1_TARG	UART1 target port
UART2_TARG	UART2 target port
I2C1_TARG	I2C1 target port
I2C2_TARG	I2C2 target port
ELM_TARG	Error Location Module target port
MCSPI1_TARG	Multi-channel Serial Peripheral Interface 1 target port
MCSPI2_TARG	MCSPI2 target port
MMC_TARG	MMC target port
MCSPI3_TARG	MCSPI3 target port
MCSPI4_TARG	MCSPI4 target port

Two ports communicate between the L3_MAIN interconnect and the L4_PER1 interconnect to allow the L3_MAIN initiators to access the L4_PER1 targets. Table 9-337 lists the L4_PER1 IAs.

For the list of initiators authorized to access the L4_PER1 peripherals, see Figure 9-3.

Table 9-337. L4_PER1 IAs

Module Iniator Name	Description
L4_PER1_IA_IP0	L4_PER1 Initiator Port 0 (IP0)
L4_PER1_IA_IP1	L4_PER1 Initiator Port 1 (IP1)

9.3.3.1.2 L4_PER2 Interconnect Agents

The L4_PER2 interconnect handles transfers only to peripherals in the PER power domain. Table 9-338 lists the L4_PER2 TAs.

Table 9-338. L4_PER2 TAs

Module Target Name	Description
MCASP1_CFG_TARG	MCASP_CFG target port
PWMSS_TARG	Pulse-Width Modulation target port

Table 9-338. L4_PER2 TAs (continued)

Module Target Name	Description
MCAN_TARG	MCAN target port
GMAC_SW_TARG	Ethernet Controller target port
ATL_TARG	Audio Tracking Logic target port
MCASP2_DAT_TARG	MCASP2_DAT target port
MCASP2_CFG_TARG	MCASP2_CFG target port
MCASP3_DAT_TARG	MCASP3_DAT target port
MCASP3_CFG_TARG	MCASP3_CFG target port

One port communicates between the L3_MAIN interconnect and the L4_PER2 interconnect to allow the L3_MAIN initiators to access the L4_PER2 targets. Table 9-339 lists the L4_PER2 IA.

For the list of initiators authorized to access the L4_PER2 peripherals, see Figure 9-3.

Table 9-339. L4_PER2 IA

Module Iniator Name	Description
L4_PER2_IA_IP0	L4_PER2 Initiator Port 0 (IP0)

9.3.3.1.3 L4_PER3 Interconnect Agents

The L4_PER3 interconnect handles transfers only to peripherals in the PER power domain. Table 9-340 lists the L4_PER3 TAs.

Table 9-340. L4_PER3 TAs

Module Target Name	Description
TIMER5_TARG	TIMER5 target port
TIMER6_TARG	TIMER6 target port
TIMER7_TARG	TIMER7 target port
TIMER8_TARG	TIMER8 target port
VIP_TARG	VIP target port
MAILBOX2_TARG	Mailbox 2 target port
OCMC_RAM_CFG_TARG	OCMC_RAM target port
MMU_TARG	MMU target port
DCC1_TARG	DCC1 target port Note: DCC1 to DCC7 are not supported on the DRA78x family of devices.
DCC2_TARG	DCC2 target port
DCC3_TARG	DCC3 target port
DCC4_TARG	DCC4 target port
DCC5_TARG	DCC5 target port
DCC6_TARG	DCC6 target port
DCC7_TARG	DCC7 target port

Two ports communicate between the L3_MAIN interconnect and the L4_PER3 interconnect to allow the L3_MAIN initiators to access the L4_PER3 targets. Table 9-341 lists the L4_PER3 IAs.

For the list of initiators authorized to access the L4_PER3 peripherals, see Figure 9-3.

Table 9-341. L4_PER3 IAs

Module Iniator Name	Description
L4_PER3_IA_IP1	L4_PER3 Initiator Port 1 (IP1)
L4_PER3_IA_IP2	L4_PER3 Initiator Port 2 (IP2)

9.3.3.1.4 L4_CFG Interconnect Agents

The L4_CFG interconnect handles only transfers to peripherals in the CORE power domain. Table 9-342 lists the L4_CFG TAs.

Table 9-342. L4_CFG TAs

Module Target Name	Description
DEBUGSS_CT_TBR_FW_CFG_TARG	Debug subsystem firewall
DSS_FW_CFG_TARG	DSS firewall
GPMC_FW_CFG_TARG	GPMC firewall
EMIF_FW_CFG_TARG	EMIF firewall
L3_INSTR_FW_CFG_TARG	L3 Instrumentation firewall
OCMC_RAM_FW_CFG_TARG	OCMC_RAM firewall
DSP1_FW_CFG_TARG	DSP1 firewall
DSP2_FW_CFG_TARG	DSP2 firewall
EVE_FW_CFG_TARG	EVE firewall
EDMA_TPCC_FW_CFG_TARG	EDMA Channel Controller firewall
EDMA_TC0_FW_CFG_TARG	EDMA Transfer Controller firewall
IPU_FW_CFG_TARG	IPU firewall
MCASP1_FW_CFG_TARG	McASP firewall
TSC_ADC_FW_CFG_TARG	ADC firewall
QSPI_FW_CFG_TARG	QSPI firewall
ISS_FW_CFG_TARG	ISS firewall Note: ISS is not supported on the DRA78x family of devices.
TESOC_FW_CFG_TARG	TESOC firewall Note: TESOC is not supported on the DRA78x family of devices.
CRC_FW_CFG_TARG	CRC firewall Note: CRC is not supported on the DRA78x family of devices.
TSC_ADC_CFG_TARG	ADC configuration target port
ESM_TARG	ESM target port Note: ESM is not supported on the DRA78x family of devices.
CM_CORE_AON_TARG	CM_CORE_AON target port
CM_CORE_TARG	CM_CORE target port
CTRL_MODULE_CORE_TARG	CTRL_MODULE_CORE target port
MAILBOX1_TARG	Mailbox1 target port
OCP_WP_NOC_TARG	OCP watchpoint target port
SPINLOCK_TARG	Spinlock target port
IEEE1500_2_OCP_TARG	IEEE1500 target port
CRC_CFG_TARG	CRC configuration target port

One port communicates between the L3_MAIN interconnect and the L4_CFG interconnect to allow the L3_MAIN initiators to access the L4_CFG targets. Table 9-343 lists the L4_CFG IA.

For the list of initiators authorized to access the L4_CFG peripherals, see Figure 9-3.

Table 9-343. L4_CFG IA

Module Iniator Name	Description
L4_CFG_IA_IP0	L4_CFG Initiator Port (IP0)

9.3.3.1.5 L4_WKUP Interconnect Agents

The L4-WKUP interconnect handles transfers only to peripherals in the WKUP power domain. Table 9-344 lists the TAs. Table 9-345 lists the L4 WKUP IA.

Table 9-344. L4_WKUP TAs

Module Target Name	Description
PRM_TARG	PRM target port
GPIO1_TARG	GPIO1 target port
TIMER1_TARG	Timer 1 target port
COUNTER_32K_TARG	Counter 32k timer
CTRL_MODULE_WKUP_TARG	CTRL_MODULE_WKUP target port
DCAN_TARG	DCAN target port
RTI1_TARG	RTI1 target port
RTI2_TARG	RTI2 target port
RTI3_TARG	RTI3 target port
RTI4_TARG	RTI4 target port
RTI5_TARG	RTI5 target port

Table 9-345. L4_WKUP IA

Module Iniator Name	Description
L4_WKUP_IA_IP0	L4_WKUP Initiator Port (IP0)

9.3.3.2 Power Management

As part of the system-wide power-management scheme, the L4 interconnects go into IDLE state after receiving a request from the PRCM module after all commands are serviced. This function is handled by hardware. For more information, see Chapter 3, *Power, Reset, and Clock Management*.

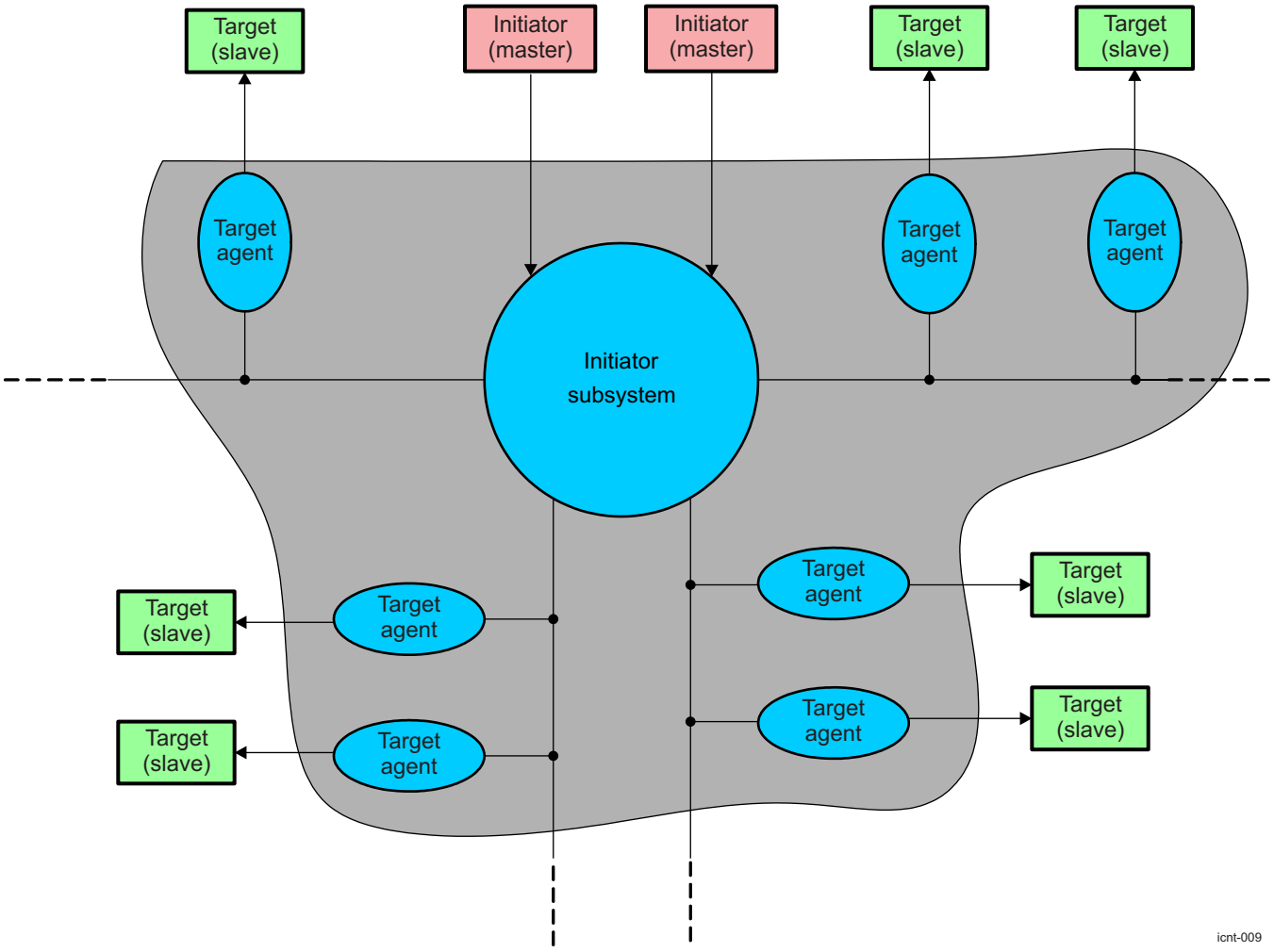
To reduce power consumption, each L4 interconnect automatically performs internal clock autogating. This is managed by hardware; no software configurations or settings are required.

L4_CFG, L4_PER1, L4_PER2, L4_PER3, and L4_WKUP are located in the always-on power domain and no retention is needed for these L4 interconnects.

9.3.3.3 L4 Firewalls

Figure 9-10 is an internal view of the L4 interconnects in the overall interconnect. This architecture, with one initiator subsystem centralizing all initiator master requests and distributing them to all target modules (peripherals), enables the L4 interconnect firewall functions to be centralized at the L4 initiator subsystem level. The L4 firewall filters the accesses based on the configurable protection groups defined in the L4 address protection (AP) registers. Each module or TA is assigned to a protection group. The configuration is also defined in the L4 AP and is programmable on a module-per-module basis.

Figure 9-10. L4 Initiator-Target Connectivity



NOTE: As Figure 9-10 shows, targets are attached to branches. Branches do not impact the function of the L4 interconnect but are present to simplify timing closure and reduce active power consumption.

Because of the large address spaces and the number of peripherals connected to an L4 interconnect, two parameters are used to set up access permission:

- Programmable groups for initiators:
  - Eight protection groups for the L4 interconnect
- Segments divided into regions
  - 78 regions for the L4_PER1 interconnect
  - 33 regions for the L4_PER2 interconnect
  - 93 regions for the L4_PER3 interconnect
  - 111 regions for the L4_CFG interconnect
  - 44 regions for the L4_WKUP interconnect

NOTE: There might be regions which are not used so the actual region count may not be the same as the listed above.

Protection group members are TAs with the same protection settings. A region is programmed to allow access to an unique selectable protection group. For better protection, different regions are grouped into protection group regions and associated with a protection group member.

9.3.3.3.1 Protection Group

The following registers are used to define a protection group:

- The CONNID_BIT_VECTOR field of register L4_AP_PROT_GROUP_MEMBERS_k_L defines which initiator belongs to a group. A protection group is accessible by an initiator when the bit position corresponding to its ConnID is set to 1 in the CONNID_BIT_VECTOR field. Table 9-346 lists the L3 ConnIDs available at L4 level. They are 4-bit ConnIDs and are equal to bits [5:2] of the 6-bit L3_MAIN ConnIDs.
- The L4_AP_PROT_GROUP_ROLES_k_L register defines the initiator requests accepted or not by the L4 firewalls depending on the combination of MReqDebug and MReqSupervisor qualifiers. Each bit of this register corresponds to an unique combination of these two qualifiers. To accept a request with combination n, bit n of the L4_AP_PROT_GROUP_ROLES_k_L register must be set to 1. Table 9-347 lists all possible combinations and the corresponding bits which have to be set to 1 for accepting particular combination.

NOTE: Permissions are identical for read and write accesses in L4 interconnect.

k indicates the protection group number.

L indicates the region number.

Table 9-346. L4 ConnID Definition

ConnID	L3 Initiator
0x1	DEBUGSS_CS_DAP_INIT, IEEE1500_2_OCP_INIT
0x2	DSP1_MDMA_INIT, DSP1_EDMA_INIT, DSP1_CFG_INIT, DSP2_EDMA_INIT
0x3	DSP2_MDMA_INIT, DSP2_CFG_INIT
0x4	EVE_TC0_INIT
0x6	IPU_INIT
0x7	TPTC1_WR_INIT, TPTC1_RD_INIT, TPTC2_WR_INIT, TPTC2_RD_INIT
0x8	DSS1_INIT, MMU_INIT
0x9	VIP_P1_INIT, VIP_P2_INIT
0xA	GMAC_SW_INIT
0xC	ISS_RT_INIT, ISS_NRT1_INIT, ISS_NRT2_INIT Note: ISS is not supported on the DRA78x family of devices.
0xD	EVE_TC1_INIT

Table 9-347. L4 Qualifiers Combination

MReqSupervisor	MReqDebug	Bit Positions In Register L4_AP_PROT_GROUP_ROLES_k_L Which Have To Be Set To 1
0	0	2
0	1	3
1	0	6
1	1	7

Table 9-348 shows an example how to set the CONNID_BIT_VECTOR field.

Table 9-348. Example For Setting The CONNID_BIT_VECTOR Field Of Register L4_AP_PROT_GROUP_MEMBERS_k_L

Initiator	Access Allowed?	CONNID_BIT_VECTOR	Bits
Reserved	-	0	0
DEBUGSS_CS_DAP_INIT, IEEE1500_2_OCP_INIT	N	0	1
DSP1_MDMA_INIT, DSP1_EDMA_INIT, DSP1_CFG_INIT, DSP2_EDMA_INIT	N	0	2
DSP2_MDMA_INIT, DSP2_CFG_INIT	N	0	3
EVE_TC0_INIT	Y	1	4
Reserved	-	0	5
IPU_INIT	Y	1	6
TPTC1_WR_INIT, TPTC1_RD_INIT, TPTC2_WR_INIT, TPTC2_RD_INIT	N	0	7
DSS1_INIT, MMU_INIT	N	0	8
VIP_P1_INIT, VIP_P2_INIT	N	0	9
GMAC_SW_INIT	N	0	10
Reserved	-	0	11
ISS_RT_INIT, ISS_NRT1_INIT, ISS_NRT2_INIT Note: ISS is not supported on the DRA78x family of devices.	N	0	12
EVE_TC1_INIT	N	0	13
Reserved	-	0	14
Reserved	-	0	15

Setting to 0x1 bits 4 and 6 in the L4_AP_PROT_GROUP_MEMBERS_k_L (for example, k = 1) register defines that group 1 can be accessed by the following initiators:

- EVE_TC0_INIT
- IPU_INIT

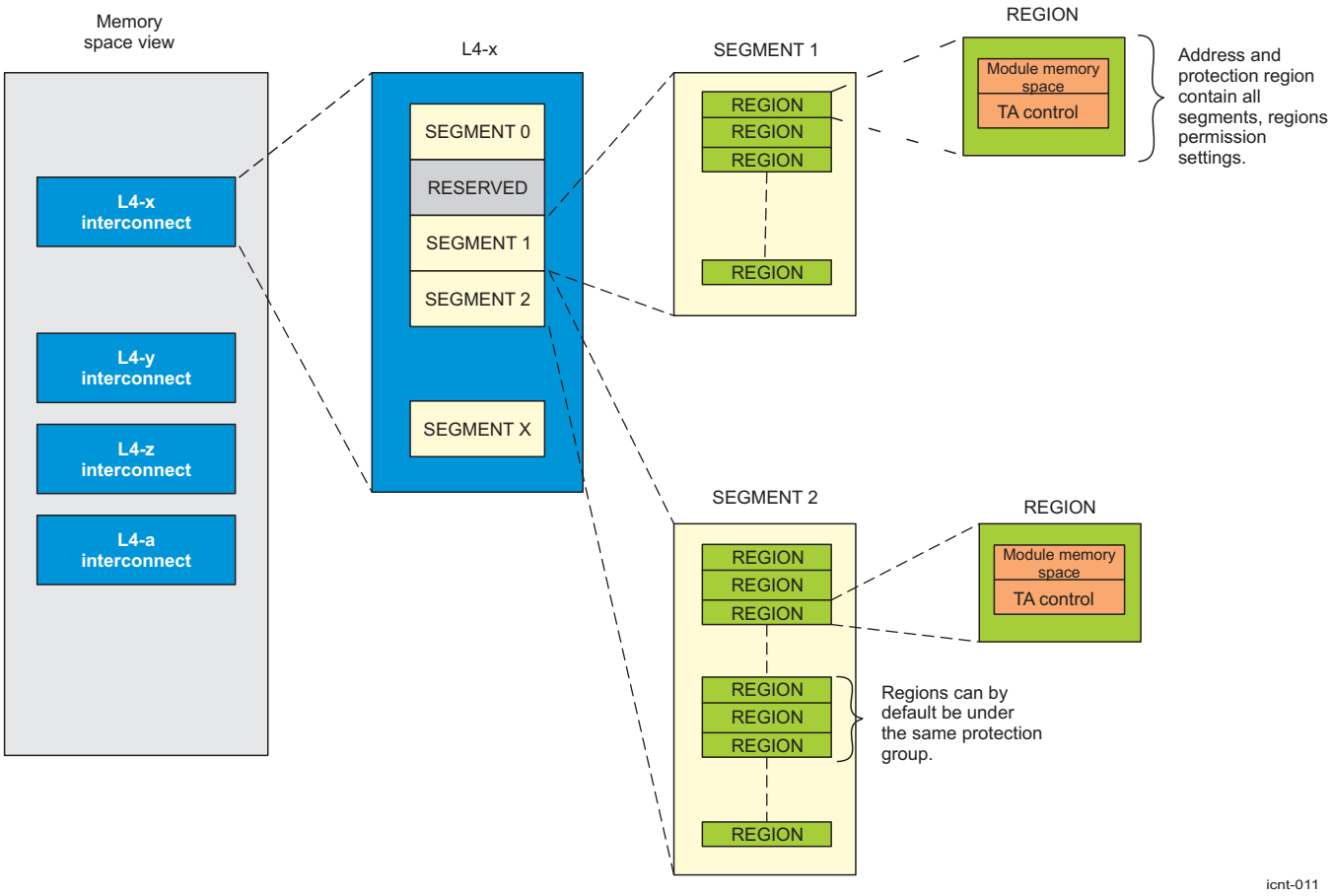
Protection group 1 can be applied to multiple protection regions with no limitation. Each protection region that is configured with protection group 1 enables permission access only to the two initiators.

The L4_AP_REGION_l_H[22:20] PROT_GROUP_ID bit field determines with which protection group the region l is associated.

9.3.3.3.2 Segments and Regions

The protection mechanism for L4 interconnects is based on a hierarchical segmentation, as shown in Figure 9-11. By default, some regions are attached to specific protection group members. This specificity lets users set up the permission access to certain types of modules requiring the same access protection without managing region allocation.

Figure 9-11. L4 Segmentation



All interconnect address spaces are covered by regions. Table 9-349 through Table 9-353 list the module mapping with their addresses, region numbers, and default protection group allocated to them.

NOTE: Module refers to the configuration registers of the module.

TA (Target Agent) refers to the interconnect configuration registers of the TA associated with the module.

Table 9-349. Region Allocations for L4_PER1 Interconnect

Module	Region	Description
L4_PER1_AP	0	Address Protection
L4_PER1_IA_IP0	1	Initiator Port 0 (IP0)
L4_PER1_LA	2	Link Agent
UART3_TARG	3	Module
	4	TA
TIMER2_TARG	5	Module
	6	TA
TIMER3_TARG	7	Module
	8	TA
TIMER4_TARG	9	Module
	10	TA

Table 9-349. Region Allocations for L4_PER1 Interconnect (continued)

Module	Region	Description
GPIO2_TARG	13	Module
	14	TA
GPIO3_TARG	15	Module
	16	TA
GPIO4_TARG	17	Module
	18	TA
UART1_TARG	24	Module
	25	TA
UART2_TARG	26	Module
	27	TA
I2C1_TARG	30	Module
	31	TA
I2C2_TARG	32	Module
	33	TA
ELM_TARG	39	Module
	40	TA
MCSPI1_TARG	47	Module
	48	TA
MCSPI2_TARG	49	Module
	50	TA
MCSPI3_TARG	67	Module
	68	TA
MCSPI4_TARG	69	Module
	70	TA
MMC_TARG	71	Module
	72	TA
L4_PER1_IA_IP1	77	L4_PER1 Initiator Port 1 (IP1)

Table 9-350. Region Allocations for L4_PER2 Interconnect

Module	Region	Description
L4_PER2_AP	0	Address Protection
L4_PER2_IA_IP0	1	Initiator Port 0 (IP0)
L4_PER2_LA	2	Link Agent
GMAC_SW_TARG	3	Module
	6	TA
MCASP1_CFG_TARG	9	Module
	10	TA
MCASP2_CFG_TARG	15	Target port
	16	Module
MCASP2_DAT_TARG	17	Target port
	18	Module
MCASP3_CFG_TARG	19	Target port
	20	Module
MCASP3_DAT_TARG	21	Target port
	22	Module

Table 9-350. Region Allocations for L4_PER2 Interconnect (continued)

Module	Region	Description
ATL_TARG	23	Target port
	24	Module
PWMSS_TARG	25	Module
	26	TA
MCAN_TARG	31	Module
	32	TA

Table 9-351. Region Allocations for L4_PER3 Interconnect

Module	Region	Description
L4_PER3_AP	0	Address Protection
L4_PER3_LA	1	Link Agent
L4_PER3_IA_IP1	3	Initiator Port 1 (IP1)
L4_PER3_IA_IP2	4	Initiator Port 2 (IP2)
TIMER5_TARG	5	Module
	6	TA
TIMER6_TARG	7	Module
	8	TA
TIMER7_TARG	9	Module
	10	TA
TIMER8_TARG	11	Module
	12	TA
VIP_TARG	21	Module
	22	TA
DCC4_TARG	31	Module
	32	TA
MAILBOX2_TARG	33	Module
	34	TA
DCC6_TARG Note: DCC1 to DCC7 are not supported on the DRA78x family of devices.	47	Module
	48	TA
DCC7_TARG	49	Module
	50	TA
DCC5_TARG	51	Module
	52	TA
DCC1_TARG	61	Module
	62	TA
DCC2_TARG	63	Module
	64	TA
DCC3_TARG	65	Module
	66	TA
OCMC_RAM_CFG_TARG	81	Module
	82	TA
MMU_TARG	91	Module
	92	TA

Table 9-352. Region Allocations for L4_CFG Interconnect

Module	Region	Description
L4_CFG_AP	0	Address Protection
L4_CFG_LA	1	Link Agent
L4_CFG_IA_IP0	2	Initiator Port (IP0)
CTRL_MODULE_CORE_TARG	3	Module
	4	TA
CM_CORE_AON_TARG	5	Module
	6	TA
CM_CORE_TARG	7	Module
	8	TA
MAILBOX1_TARG	17	Module
	18	TA
SPINLOCK_TARG	19	Module
	20	TA
OCP_WP_NOC_TARG	21	Module
	22	TA
IEEE1500_2_OCP_TARG	23	Module
	24	TA
EVE_FW_CFG_TARG	25	Module
	26	TA
IPU_FW_CFG_TARG	27	Module
	28	TA
EDMA_TPCC_FW_CFG_TARG	29	Module
	30	TA
EDMA_TC0_FW_CFG_TARG	31	Module
	32	TA
MCASP1_FW_CFG_TARG	33	Module
	34	TA
TSC_ADC_FW_CFG_TARG	35	Module
	36	TA
DSP1_FW_CFG_TARG	37	Module
	38	TA
DSP2_FW_CFG_TARG	39	Module
	40	TA
EMIF_FW_CFG_TARG	43	Module
	44	TA
GPMC_FW_CFG_TARG	47	Module
	48	TA
OCMC_RAM_FW_CFG_TARG	49	Module
	50	TA
DSS_FW_CFG_TARG	53	Module
	54	TA
DEBUGSS_CT_TBR_FW_CFG_TARG	55	Module
	56	TA
L3_INSTR_FW_CFG_TARG	57	Module
	58	TA
QSPI_FW_CFG_TARG	59	Module
	60	TA

Table 9-352. Region Allocations for L4_CFG Interconnect (continued)

Module	Region	Description
CRC_FW_CFG_TARG Note: CRC is not supported on the DRA78x family of devices.	71	Module
	72	TA
TSC_ADC_CFG_TARG	89	Module
	90	TA
ISS_FW_CFG_TARG Note: ISS is not supported on the DRA78x family of devices.	105	Module
	106	TA
TESOC_FW_CFG_TARG Note: TESOC is not supported on the DRA78x family of devices.	107	Module
	108	TA
CRC_CFG_TARG	109	Module
	110	TA

Table 9-353. Region Allocations for L4_WKUP Interconnect

Module	Region	Description
L4_WKUP_AP	0	Address Protection
L4_WKUP_IA_IP0	1	Initiator Port (IP0)
L4_WKUP_LA	2	Link Agent
PRM_TARG	3	Module
	4	TA
GPIO1_TARG	5	Module
	6	TA
TIMER1_TARG	9	Module
	10	TA
COUNTER_32K_TARG	15	Module
	16	TA
CTRL_MODULE_WKUP_TARG	17	Module
	18	TA
DCAN_TARG	30	Module
	31	TA
RTI1_TARG	34	Module
	35	TA
RTI2_TARG	36	Module
	37	TA
RTI3_TARG	38	Module
	39	TA
RTI4_TARG	40	Module
	41	TA
RTI5_TARG	42	Module
	43	TA

9.3.3.3.3 L4 Firewall Address and Protection Register Settings

Table 9-354 lists the settings of the AP registers relative to an L4 interconnect firewall. These values are computed based on the physical implementation of each L4 interconnect.

Table 9-354. L4 Firewall Register Description Overview

Register Type	Register Name	Bits	Field	Description
Segment	L4_AP_SEGMENT_i_L (1)	31:0	BASE	Segment base address
	L4_AP_SEGMENT_i_H (1)	5:0	SIZE	Segment size equals to 2^{SIZE}
Protection groups	L4_AP_PROT_GROUP_MEMBERS_k_L (2)	15:0	CONNID_BIT_VECTOR	For L4 ConnID, see Table 9-346.
	L4_AP_PROT_GROUP_ROLES_k_L (2)	31:0	ENABLE	Defines the initiator requests accepted or not by the L4 firewalls depending on combination of L4 qualifiers.
Region setting	L4_AP_REGION_l_L (3)	20:0	BASE	Defines the base address of a region with respect to its segment base address
	L4_AP_REGION_l_H (3)	31:28	MADDRSPACE	Target interconnect MAddrSpace
		26:24	SEGMENT_ID	Segment ID number of the region
		22:20	PROT_GROUP_ID	Protection group member attached to the region
		18:17	BYTE_DATA_WIDTH_EXP	Determines the number of bytes in an access
		14:8	PHY_TARGET_ID	Physical target ID
		6:1	SIZE	Size of the region equals to 2^{SIZE}
		0	ENABLE	Enables the region protection

(1) i = 0 to 1 for L4_PER1_AP
i = 0 for L4_PER2_AP
i = 0 for L4_PER3_AP
i = 0 to 2 for L4_CFG_AP
i = 0 to 3 for L4_WKUP_AP

(2) k = 0 to 7 for L4_PER1_AP
k = 0 to 7 for L4_PER2_AP
k = 0 to 7 for L4_PER3_AP
k = 0 to 7 for L4_CFG_AP
k = 0 to 7 for L4_WKUP_AP

(3) l = 0 to 77 for L4_PER1_AP
l = 0 to 32 for L4_PER2_AP
l = 0 to 92 for L4_PER3_AP
l = 0 to 110 for L4_CFG_AP
l = 0 to 43 for L4_WKUP_AP

9.3.3.4 L4 Error Detection and Reporting

9.3.3.4.1 IA and TA Error Detection and Logging

The L4 interconnect provides mechanisms for handling internally detected errors or errors reported by modules attached to the L4 target ports.

NOTE: L4_IA denotes the IA for all L4 interconnects: L4_PER1, L4_PER2, L4_PER3, L4_CFG, and L4_WKUP.

L4_TA denotes the TA for all L4 interconnects: L4_PER1, L4_PER2, L4_PER3, L4_CFG, and L4_WKUP.

The L4 interconnects handle four types of errors:

- No target core found or address hole, detected and logged at IA
- Unsupported command, detected and logged at IA
- Protection violation, detected and logged at IA (see Section 9.3.3.3, *L4 Firewalls*)
- Target does not service a request before a time-out expires. The error is detected and logged at TA (see Section 9.3.3.4.2, *Time-Out*).

Table 9-355 lists the value of the L4_IA_ERROR_LOG_L[25:24] CODE bit field stored when an error occurs.

Table 9-355. L4 CODE Bit Field Definition

CODE (bits 1:0)	Error Type	REQ_INFO	Secondary	ConnID	CMD
0	No error				
1	Unsupported command	x	x	x	x
2	Address hole	x	x	x	x
3	Protection violation	x		x	x

- No target core found/address hole: This error indicates that a request was addressed to a hole in the L4 address map. When this error occurs, an in-band error response is returned to the L3 level. The error is also logged into the L4_IA_AGENT_STATUS_L[27] INBAND_ERROR bit. Additionally, an address hole error code is logged to the L4_IA_ERROR_LOG_L[25:24] CODE bit field.
- Unsupported command: This error indicates that the command type of the request is not supported by the accessed target register. The error is logged into the L4_IA_AGENT_STATUS_L[27] INBAND_ERROR bit. An unsupported command error code is written to the L4_IA_ERROR_LOG_L[25:24] CODE bit field for the initiator interface.
- Protection violation: This error indicates that a request is not issued from an allowed initiator member or is issued with the inappropriate ReqInfo qualifiers associated with the target region. This error is reported using an in-band error and is written to the L4_IA_AGENT_STATUS_L[27] INBAND_ERROR bit. A protection violation error code is saved into the L4_IA_ERROR_LOG_L[25:24] CODE bit field for the same initiator interface. A protection violation is also logged in the L4_IA_AGENT_STATUS_L[31] PROT_ERROR_SECONDARY bit when the device is in debug mode or in the L4_IA_AGENT_STATUS_L[30] PROT_ERROR_PRIMARY bit when the device is in application mode.

  The L4_IA_ERROR_LOG_L[30] SECONDARY bit indicates whether the error occurred in application or debug.

  The L4_IA_ERROR_LOG_H[15:0] REQ_INFO bit field returns the type of access (MreqInfo qualifier) that has caused the error.

  The L4_IA_ERROR_LOG_L[13:8] CONNID bit field returns the ConnID of the initiator that has caused the error.

  The L4_IA_ERROR_LOG_ADDR_L[31:0] ADDR register logs the address for error conditions.

9.3.3.4.2 Time-Out

A time-out mechanism can be enabled at the interconnect level and on a per-target basis. If the mechanism is enabled for a TA and interconnect and commands are not accepted or responses are not returned within the expected delay, the L4 interconnect generates an error event.

The error is logged in the L4_TA_AGENT_STATUS_L[8] REQ_TIMEOUT bit. The affected TA enters an error state that causes it to send an error response to any new request targeted at it. To recover from this state, system software must reset the TA. The time-out is counted starting from the moment a command is presented to the target, regardless of the target response to this command.

The L4 interconnect implements a centralized time-base circuit that broadcasts a set of four periodic pulse signals to all connected TAs. The time-base circuit offers four possible sets of four time-base signals. The time-base signals are selected by programming the L4_LA_NETWORK_CONTROL_L[10:8] TIMEOUT_BASE bit field.

The selected time-base signals are available at any TA. Each TA can be programmed to refer to one of these four time-base signals, using the L4_TA_AGENT_CONTROL_L[10:8] REQ_TIMEOUT bit field. These four signals are referred to as 1X time-base, 4X time-base, 16X time-base, and 64X time-base.

Table 9-356 lists all values in number of L4 clock cycles.

Table 9-356. L4 Time-out Link and TA Programming

	L4_TA_AGENT_CONTROL_L [10:8] REQ_TIMEOUT				
L4_LA_NETWORK_CONTROL_L[10:8] TIMEOUT_BASE	0	1	2	3	4
0	All L4 time-out features are disabled.				
1	Locally disabled	64	256	1024	4096
2		256	1024	4096	16,384
3		1024	4096	16,384	65,536
4		4096	16,384	65,536	262,144

The default reset value is 0x2 for REQ_TIMEOUT and 0x4 for TIMEOUT_BASE, implying 16,384 clock cycles.

A time-out condition is detected when the command acceptance or the response is not received after a delay of from one to three time-base periods.

Example:

- L4 frequency = 65MHz, 15.3-µs period
- TIMEOUT_BASE = 4 in the L4_LA_NETWORK_CONTROL_L register
- REQ_TIMEOUT = 2 in the L4_TA_AGENT_CONTROL_L for TA A
- REQ_TIMEOUT = 4 in the L4_TA_AGENT_CONTROL_L for TA B

At agent A, the time-base unit is 16,384 cycles. A time-out is issued when a request to the attached module is not accepted, or no response is sent after a delay of 252 µs to 756 µs.

At agent B, the time-base unit is 262,144 cycles. A time-out is issued when a request to the attached module is not accepted or no response is sent after a delay of 4 ms to 12 ms.

When a time-out condition is detected, the TA logs the error in the L4_TA_AGENT_STATUS_L[8] REQ_TIMEOUT bit, and it also reports the error to the IA, which forwards it to the L3 interconnects.

After the time-out is detected and logged, the behavior of the attached module is ignored. A new request targeting the module arriving at the timed-out TA receives an error response. If the request is addressed to the agent internal registers, it is processed normally.

To recover from a time-out error, software is assumed to first reset the faulty module and then the TA using the L4_TA_AGENT_CONTROL_L[0] OCP_RESET bit.

9.3.3.4.3 Error Reporting

Figure 9-12 shows the error-reporting scheme used in the L4 interconnects. All L4 in-band errors are reported to their respective L3 TA, where errors are converted in an out-of band error signals (L3_MAIN_IRQ_APP_ERR and L3_MAIN_IRQ_DBG_ERR) going to the IRQ_CROSSBAR.

To enable interrupt reporting in case of error occurrence, one or all of the following bits (depending on user's desire) must be set to 0x1:

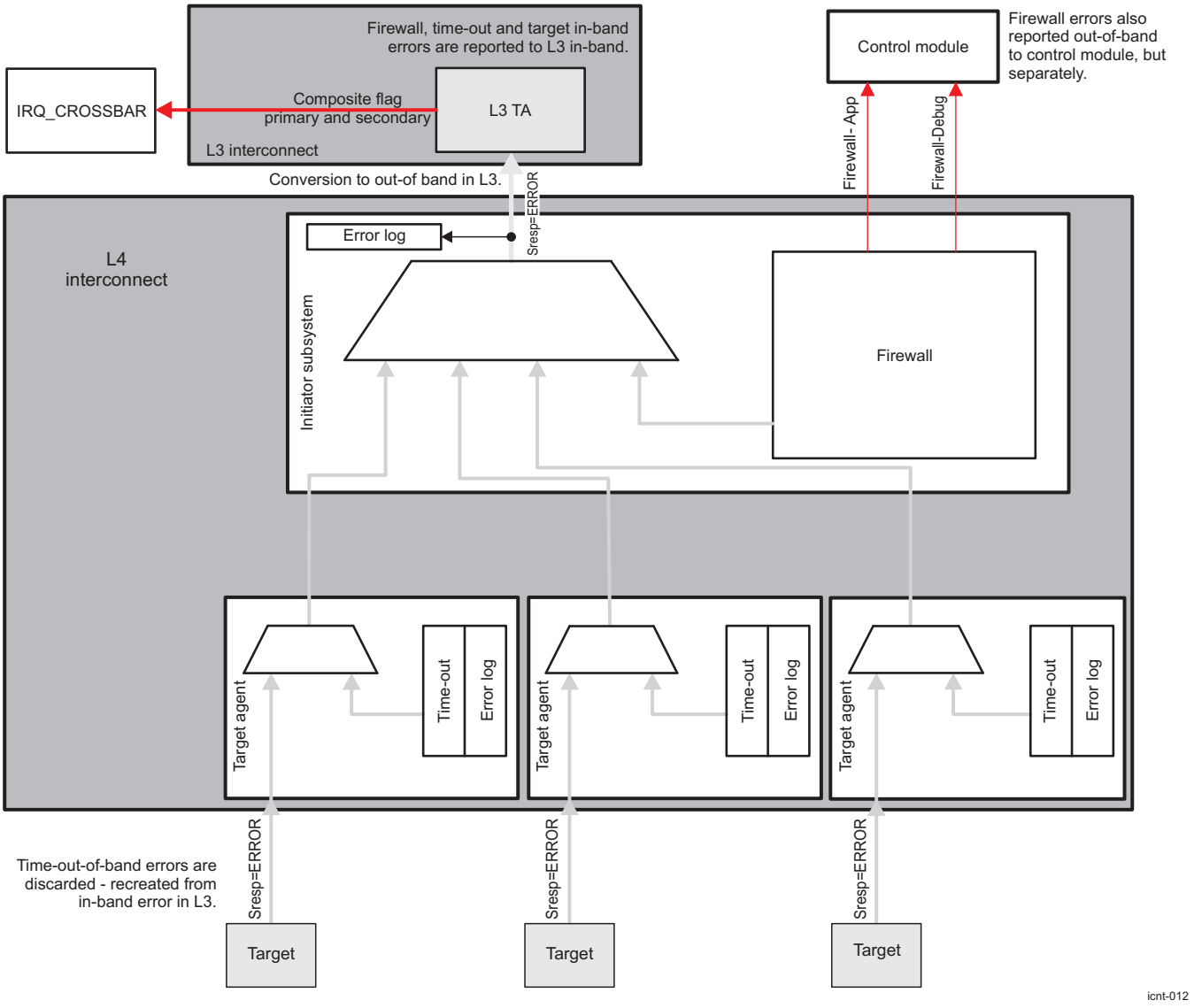
- CLK1_FLAGMUX_CLK1_1.L3_FLAGMUX_MASK0[10] for L4_CFG
- CLK1_FLAGMUX_CLK1_1.L3_FLAGMUX_MASK0[11] for L4_WKUP
- CLK1_FLAGMUX_CLK1_2.L3_FLAGMUX_MASK0[5] for L4_PER2_P1
- CLK1_FLAGMUX_CLK1_2.L3_FLAGMUX_MASK0[6] for L4_PER1_P1
- CLK1_FLAGMUX_CLK1_2.L3_FLAGMUX_MASK0[7] for L4_PER1_P2
- CLK1_FLAGMUX_CLK1_2.L3_FLAGMUX_MASK0[8] for L4_PER3_P2
- CLK1_FLAGMUX_CLK1_2.L3_FLAGMUX_MASK0[9] for L4_PER3_P1

The following bits must also be set to 0x1:

- L3_FLAGMUX_CLK1MERGE_MASK0[1]

- L3_FLAGMUX_CLK1MERGE_MASK0[0]

Figure 9-12. L4 Error Reporting



9.3.3.4.4 Error Recovery

Setting the L4_TA_AGENT_CONTROL_L[0] OCP_RESET bit to 1 initiates the software reset period. Software reset must be asserted for at least 16 cycles of the target module interface clock, which can be a divided clock with respect to the L4 clock.

During the software reset period:

- Requests sent to the target module receive error responses. Therefore, if the faulty request is part of a DMA transfer, it is necessary to stop the DMA to prevent unwanted errors.
- Requests sent to the TA register block are processed as usual.
- The L4_TA_AGENT_STATUS_L[8] REQ_TIMEOUT status bit is cleared.

Setting the L4_TA_AGENT_CONTROL_L[0] OCP_RESET bit to 0 terminates the software reset period.

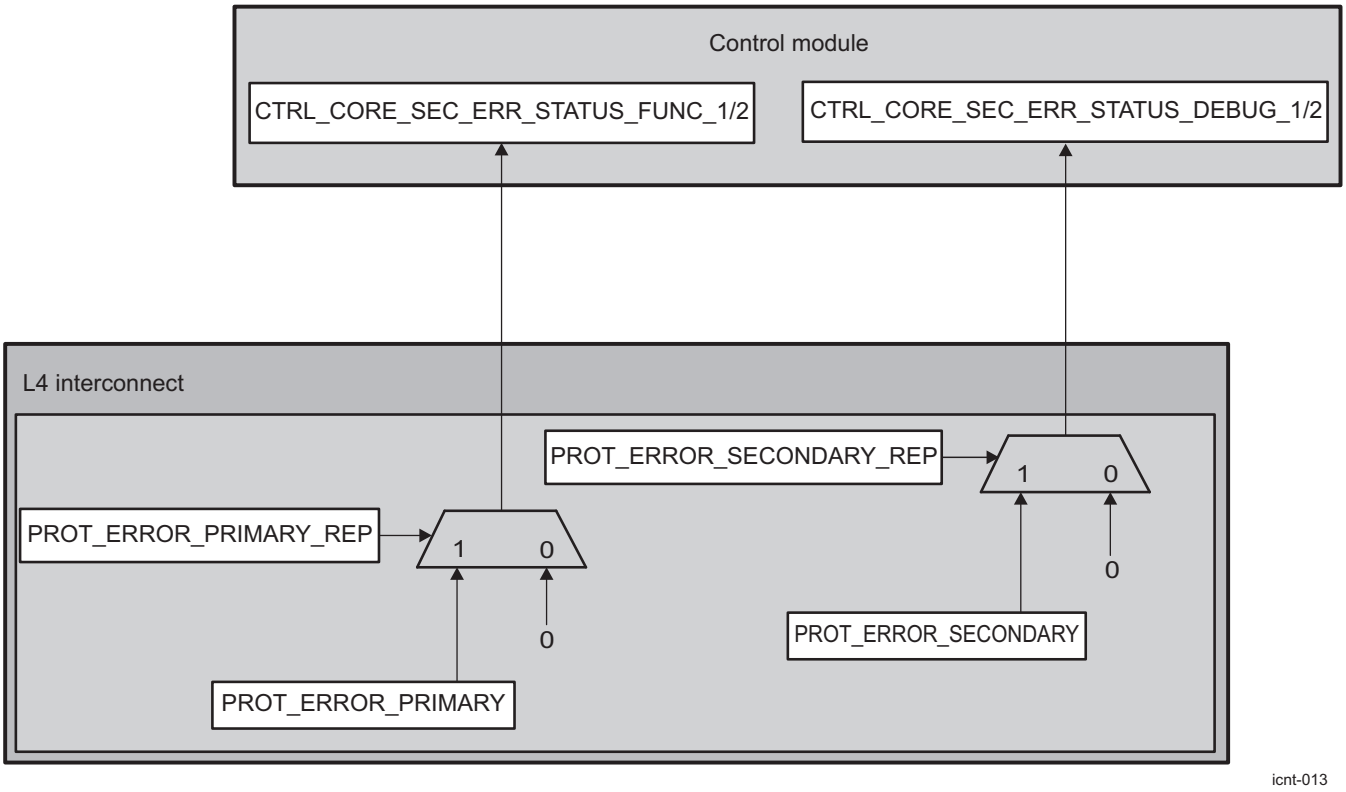
Reset the attached module to complete the recovery.

9.3.3.4.5 Firewall Error Logging in the Control Module

When a protection violation occurs, an interrupt is sent to the IRQ_CROSSBAR. An in-band error is sent back to L3 IA and an out-of-band error is logged in the CTRL_CORE_SEC_ERR_STATUS_FUNC_1/2 register (used in application mode) or in the CTRL_CORE_SEC_ERR_STATUS_DEBUG_1/2 register (used in debug mode). These out-of-band errors are enabled at the L4 IA level by setting to 0x1 the L4_IA_AGENT_CONTROL_L[31] PROT_ERROR_SECONDARY_REP bit when the device is in debug mode or the L4_IA_AGENT_CONTROL_L[30] PROT_ERROR_PRIMARY_REP bit when the device is in application mode.

Figure 9-13 shows the global protection error reporting to the control module.

Figure 9-13. Protection Violation Out-of-Band Error Reporting



9.3.4 L4 Interconnect Programming Guide

9.3.4.1 L4 Interconnect Low-level Programming Models

This section describes the low-level hardware programming sequences for configuring and using the L4 interconnect module.

9.3.4.1.1 Global Initialization

9.3.4.1.1.1 Surrounding Modules Global Initialization

This section identifies the requirements of initializing the surrounding modules when the L4 interconnects are used for the first time after a device reset. This initialization of surrounding modules is based on the integration of the L4 interconnects. For more information, see Section 9.3.2, *L4 Interconnect Integration*.

Table 9-357 lists the surrounding modules.

Table 9-357. Global Initialization of Surrounding Modules

Surrounding Modules	Comments
PRCM	For information about the configuration of the PRCM module, see Chapter 3, *Power, Reset, and Clock Management*.
Control module	For information about the configuration of the control module, see Chapter 13, *Control Module*.
IRQ_CROSSBAR	IRQ_CROSSBAR configuration must be done to allow module IRQs to be mapped to certain device INTC line. For more information see Section 13.4.6.4, *IRQ_CROSSBAR Module Functional Description*, in Chapter 13, *Control Module*
Device INTCs	Device INTCs must be configured to enable the interrupt request generation. For more information see Chapter 12 *Interrupt Controllers*
L3_MAIN interconnect	For more information about the L3_MAIN interconnect configuration, see Section 9.2, *L3_MAIN Interconnect*.

9.3.4.1.2 Operational Modes Configuration

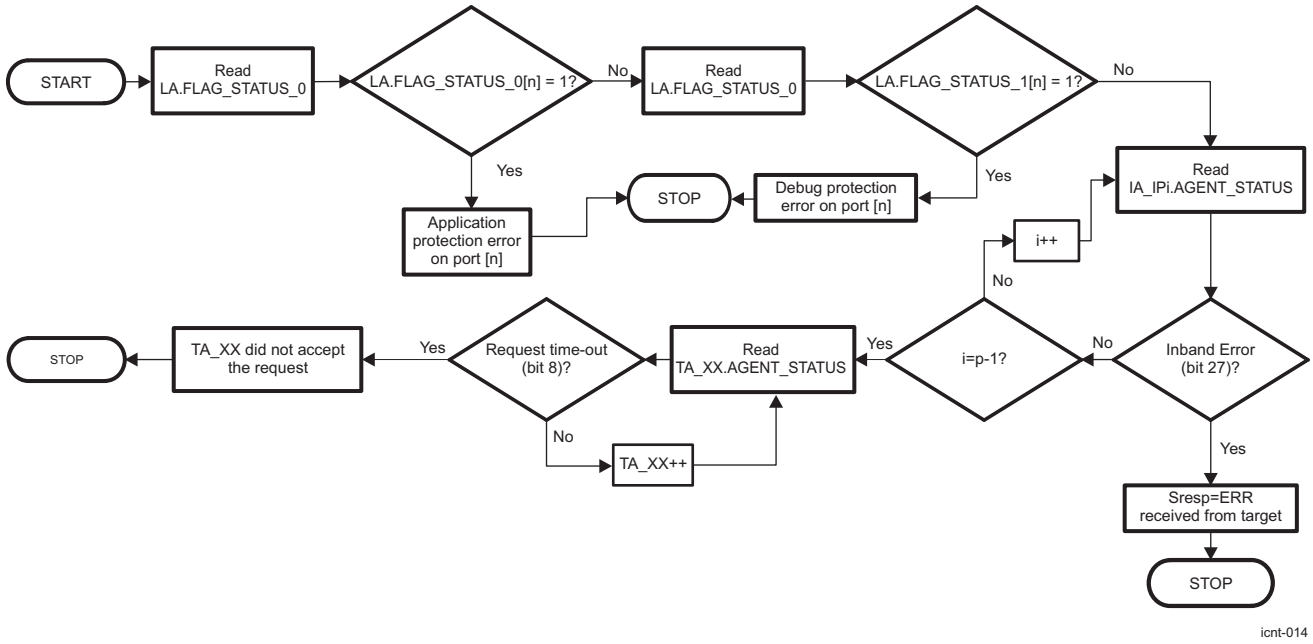
9.3.4.1.2.1 L4 Interconnect Error Analysis Mode

9.3.4.1.2.1.1 Main Sequence: L4 Interconnect Error Analysis Mode

The information required to analyze an error source is logged in several registers. The number of registers to access depends on the error source.

Figure 9-14 shows the software sequence required in most cases.

Figure 9-14. Typical Error Analysis Sequence



NOTE: L4 interconnects don't log any address or other specific information for a custom error returned from any target IP. They rather pass an error response up to the master supposed to analyze it.

In case of posted writes, the master access completes before it actually completed at the end slave level, this way no error response is sent back to the master, making it impossible to have a direct way of uderstanding the origin of L4 error during posted writes. However, even though no address is logged, an error flag is generated and needs to be processed.

9.3.4.1.2.1.2 Subsequence: L4 Interconnect Protection Violation Error Identification

This procedure describes the protection violation error identification (see Table 9-358).

Table 9-358. Protection Violation Error Identification

Step	Register/Bit Field/Programming Model	Value
Read multiple errors detection.	L4_IA_ERROR_LOG_L[31] MULTI	0x-
Read the initiator ConnID.	L4_IA_ERROR_LOG_L[13:8] CONNID	0x-
Read the command that has caused the error.	L4_IA_ERROR_LOG_L[2:0] CMD	0x-
Read address of request that has caused the error.	L4_IA_ERROR_LOG_ADDR_L[31:0] ADDR	0x-
IF: Is it a primary error?	L4_IA_AGENT_STATUS_L[30] PROT_ERROR_PRIMARY	0x1
Read status bits.	CTRL_CORE_SEC_ERR_STATUS_FUNC_1/2	0x-
Write 1 to clear status bits.	CTRL_CORE_SEC_ERR_STATUS_FUNC_1[16] L4_PERIPH1_FW_ERROR	0x1
	CTRL_CORE_SEC_ERR_STATUS_FUNC_1[17] L4_CONFIG_FW_ERROR	0x1
	CTRL_CORE_SEC_ERR_STATUS_FUNC_1[22] L4_WAKEUP_FW_ERROR	0x1
	CTRL_CORE_SEC_ERR_STATUS_FUNC_2[4] L4_PERIPH2_FW_ERROR	0x1
	CTRL_CORE_SEC_ERR_STATUS_FUNC_2[5] L4_PERIPH3_FW_ERROR	0x1

Table 9-358. Protection Violation Error Identification (continued)

Step	Register/Bit Field/Programming Model	Value
Write 1 to clear IA status bit.	L4_IA_AGENT_STATUS_L[30] PROT_ERROR_PRIMARY	0x1
ELSE		
Read status bits.	CTRL_CORE_SEC_ERR_STATUS_DEBUG_1/2	0x-
Write 1 to clear status bits.	CTRL_CORE_SEC_ERR_STATUS_DEBUG_1[16] L4_PERIPH1_DBGFW_ERROR	0x1
	CTRL_CORE_SEC_ERR_STATUS_DEBUG_1[17] L4_CONFIG_DBGFW_ERROR	0x1
	CTRL_CORE_SEC_ERR_STATUS_DEBUG_1[22] L4_WAKEUP_DBGFW_ERROR	0x1
	CTRL_CORE_SEC_ERR_STATUS_DEBUG_2[4] L4_PERIPH2_DBGFW_ERROR	0x1
	CTRL_CORE_SEC_ERR_STATUS_DEBUG_2[5] L4_PERIPH3_DBGFW_ERROR	0x1
Write 1 to clear IA status bit	L4_IA_AGENT_STATUS_L[31] PROT_ERROR_SECONDARY	0x1
ENDIF		
Write 1 to clear multiple errors detection.	L4_IA_ERROR_LOG_L[31] MULTI	0x1
Write 1 to clear in-band error status.	L4_IA_AGENT_STATUS_L[24] INBAND_ERROR	0x1

9.3.4.1.2.1.3 Subsequence: L4 Interconnect Unsupported Command/Address Hole Error Identification

This procedure describes the identification of unsupported command/address hole error (see Table 9-359).

Table 9-359. Unsupported Command/Address Hole Error Identification

Step	Register/Bit Field/Programming Model	Value
Read multiple errors detection.	L4_IA_ERROR_LOG_L[31] MULTI	0x-
Read the initiator ConnID.	L4_IA_ERROR_LOG_L[11:8] CONNID	0x-
Read command that has caused the error.	L4_IA_ERROR_LOG_L[2:0] CMD	0x-
Read the address of the request that has caused the error.	L4_IA_ERROR_LOG_ADDR_L[31:0] ADDR	0x-
Read secondary status.	L4_IA_ERROR_LOG_L[30] SECONDARY	0x-
Write 1 to clear secondary status.	L4_IA_ERROR_LOG_L[30] SECONDARY	0x1
Write 1 to clear multiple errors detection.	L4_IA_ERROR_LOG_L[31] MULTI	0x1
Write 1 to clear inband error status.	L4_IA_AGENT_STATUS_L[24] INBAND_ERROR	0x1

9.3.4.1.2.1.4 Subsequence: L4 Interconnect Reset TA and Module

This procedure describes the reset TA and module (see Table 9-360).

Table 9-360. Reset TA and Module

Step	Register/Bit Field/Programming Model	Value
Reset TA.	L4_TA_AGENT_CONTROL_L[0] OCP_RESET	0x1
Wait until target module clock = 16 cycles.		
Write 0 to clear TA time-out status.	L4_TA_AGENT_CONTROL_L[10:8] REQ_TIMEOUT	0x0
Write 0 to clear TA reset.	L4_TA_AGENT_CONTROL_L[0] OCP_RESET	0x0
Reset the attached module.	For more information, see the respective module chapter.	

9.3.4.1.2.2 L4 Interconnect Time-Out Configuration Mode

9.3.4.1.2.2.1 Main Sequence: L4 Interconnect Time-Out Configuration Mode

This procedure describes the time-out configuration sequence (see Table 9-361).

Table 9-361. Time-Out Configuration

Step	Register/Bit Field/Programming Model	Value
Disable time-out.	L4_LA_NETWORK_CONTROL_L[10:8] TIMEOUT_BASE	0x0
Clear TA time-out error status.(1)	L4_TA_AGENT_STATUS_L[8] REQ_TIMEOUT	0x1
Set time-out at TA level.(1)	L4_TA_AGENT_CONTROL_L[10:8] REQ_TIMEOUT	0x-
Set time-out base.	L4_LA_NETWORK_CONTROL_L[10:8] TIMEOUT_BASE	0x-

(1) Required for each TA.

9.3.4.1.2.3 L4 Interconnect Firewall Configuration Mode

9.3.4.1.2.3.1 Main Sequence: L4 Interconnect Firewall Configuration Mode

This procedure describes the firewall configuration sequence (see Table 9-362).

Table 9-362. Firewall Configuration

Step	Register/Bit Field/Programming Model	Value
Define the members of protection group k. In other words, define which initiators can access the protection group k. For ConnID values, see Table 9-346 .	L4_AP_PROT_GROUP_MEMBERS_k_L[15:0] CONNID_BIT_VECTOR(1)	0x-
For the protection group k define the initiator requests accepted or not by the L4 firewalls depending on combination of L4 qualifiers. For more details, see Table 9-347	L4_AP_PROT_GROUP_ROLES_k_L[31:0] ENABLE(1)	0x-
Assign the region l to a protection group.	L4_AP_REGION_l_H[22:20] PROT_GROUP_ID(2)	0x-

(1) Required for each protection group.
(2) Required for each region.

9.3.5 L4 Interconnects Register Manual

Table 9-363 through Table 9-367 list all L4 register blocks for IA, TA, AP, and LA. Each module instance is shown with the module register mapping and bit and bit field definitions.

9.3.5.1 L4 Interconnects Instance Summary

NOTE: ISS, CRC, TESOC, ESM and DCC1 to DCC7 are not supported on the DRA78x family of devices.

Table 9-363. L4_PER1 Instance Summary

Module Name	L3_MAIN Base Address	Size
L4_PER1_AP	0x4800 0000	2KB
L4_PER1_LA	0x4800 0800	2KB
L4_PER1_IA_IP0	0x4800 1000	1KB
L4_PER1_IA_IP1	0x4800 1400	1KB
UART3_TARG	0x4802 1000	4KB
TIMER2_TARG	0x4803 3000	4KB
TIMER3_TARG	0x4803 5000	4KB
TIMER4_TARG	0x4803 7000	4KB
GPIO2_TARG	0x4805 6000	4KB
GPIO3_TARG	0x4805 8000	4KB
GPIO4_TARG	0x4805 A000	4KB
UART1_TARG	0x4806 B000	4KB
UART2_TARG	0x4806 D000	4KB
I2C1_TARG	0x4807 1000	4KB
I2C2_TARG	0x4807 3000	4KB
ELM_TARG	0x4807 9000	4KB
MCSPI1_TARG	0x4809 9000	4KB
MCSPI2_TARG	0x4809 B000	4KB
MCSPI3_TARG	0x480B 9000	4KB
MCSPI4_TARG	0x480B B000	4KB
MMC_TARG	0x480D 2000	4KB

Table 9-364. L4_PER2 Instance Summary

Module Name	L3_MAIN Base Address	Size
L4_PER2_AP	0x4840 0000	2KB
L4_PER2_LA	0x4840 0800	2KB
L4_PER2_IA_IP0	0x4840 1000	1KB
GMAC_SW_TARG	0x4848 8000	4KB
MCASP1_CFG_TARG	0x4846 2000	4KB
PWMSS_TARG	0x4843 F000	4KB
MCAN_TARG	0x4848 2000	4KB
ATL_TARG	0x4843 D000	4KB
MCASP2_DAT_TARG	0x4843 7000	4KB
MCASP2_CFG_TARG	0x4846 E000	4KB
MCASP3_DAT_TARG	0x4843 B000	4KB
MCASP3_CFG_TARG	0x4847 2000	4KB

Table 9-365. L4_PER3 Instance Summary

Module Name	L3_MAIN Base Address	Size
L4_PER3_AP	0x4880 0000	2KB
L4_PER3_LA	0x4880 0800	2KB
L4_PER3_IA_IP1	0x4880 1400	1KB
L4_PER3_IA_IP2	0x4880 1800	1KB
TIMER5_TARG	0x4882 1000	4KB
TIMER6_TARG	0x4882 3000	4KB
TIMER7_TARG	0x4882 5000	4KB
TIMER8_TARG	0x4882 7000	4KB
VIP_TARG	0x4898 0000	4KB
MAILBOX2_TARG	0x4883 B000	4KB
DCC1_TARG	0x4882 9000	4KB
DCC2_TARG	0x4882 B000	4KB
DCC3_TARG	0x4882 D000	4KB
DCC4_TARG	0x4882 F000	4KB
DCC5_TARG	0x4884 F000	4KB
DCC6_TARG	0x4885 3000	4KB
DCC7_TARG	0x4885 5000	4KB
OCMC_RAM_CFG_TARG	0x4880 5000	4KB
MMU_TARG	0x4881 D000	4KB

Table 9-366. L4_CFG Instance Summary

Module Name	L3_MAIN Base Address	Size
L4_CFG_AP	0x4A00 0000	2KB
L4_CFG_LA	0x4A00 0800	2KB
L4_CFG_IA_IP0	0x4A00 1000	4KB
CTRL_MODULE_CORE_TARG	0x4A00 4000	4KB
CM_CORE_AON_TARG	0x4A00 6000	4KB
CM_CORE_TARG	0x4A00 A000	4KB
MAILBOX1_TARG	0x4A0F 5000	4KB
SPINLOCK_TARG	0x4A0F 7000	4KB
OCP_WP_NOC_TARG	0x4A10 3000	4KB
IEEE1500_2_OCP_TARG	0x4A10 9000	4KB
EVE_FW_CFG_TARG	0x4A15 2000	4KB
IPU_FW_CFG_TARG	0x4A15 C000	4KB
EDMA_TPCC_FW_CFG_TARG	0x4A16 2000	4KB
EDMA_TC0_FW_CFG_TARG	0x4A16 4000	4KB
MCASP1_FW_CFG_TARG	0x4A16 8000	4KB
TSC_ADC_FW_CFG_TARG	0x4A16 E000	4KB
DSP1_FW_CFG_TARG	0x4A17 2000	4KB
DSP2_FW_CFG_TARG	0x4A17 4000	4KB
QSPI_FW_CFG_TARG	0x4A17 A000	4KB
EMIF_FW_CFG_TARG	0x4A20 D000	4KB
GPMC_FW_CFG_TARG	0x4A21 1000	4KB
OCMC_RAM_FW_CFG_TARG	0x4A21 3000	4KB
DSS_FW_CFG_TARG	0x4A21 D000	4KB
DEBUGSS_CT_TBR_FW_CFG_TARG	0x4A22 5000	4KB
L3_INSTR_FW_CFG_TARG	0x4A22 7000	4KB

Table 9-366. L4_CFG Instance Summary (continued)

Module Name	L3_MAIN Base Address	Size
ESM_TARG	0x4A23 D000	4KB
ISS_FW_CFG_TARG	0x4A23 F000	4KB
TESOC_FW_CFG_TARG	0x4A24 1000	4KB
CRC_FW_CFG_TARG	0x4A24 5000	4KB
CRC_CFG_TARG	0x4A26 1000	4KB
TSC_ADC_CFG_TARG	0x4A26 5000	4KB

Table 9-367. L4_WKUP Instance Summary

Module Name	L3_MAIN Base Address	Size
L4_WKUP_AP	0x4AE0 0000	2KB
L4_WKUP_LA	0x4AE0 0800	2KB
L4_WKUP_IA_IP0	0x4AE0 1000	4KB
PRM_TARG	0x4AE0 8000	4KB
GPIO1_TARG	0x4AE1 1000	4KB
TIMER1_TARG	0x4AE1 9000	4KB
COUNTER_32K_TARG	0x4AE0 5000	4KB
CTRL_MODULE_WKUP_TARG	0x4AE0 D000	4KB
DCAN_TARG	0x4AE3 E000	4KB
RTI1_TARG	0x4AE3 2000	4KB
RTI2_TARG	0x4AE3 4000	4KB
RTI3_TARG	0x4AE3 6000	4KB
RTI4_TARG	0x4AE3 8000	4KB
RTI5_TARG	0x4AE3 A000	4KB

9.3.5.2 L4 Initiator Agent (L4 IA)

9.3.5.2.1 L4 Initiator Agent (L4 IA) Register Summary

Table 9-368 summarizes the L4 IA register mapping.

Table 9-368. IA Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	L4_PER1_IA_IP0 L3_MAIN Physical Address	L4_PER1_IA_IP1 L3_MAIN Physical Address
L4_IA_COMPONENT_L	R	32	0x0000 0000	0x4800 1000	0x4800 1400
L4_IA_COMPONENT_H	R	32	0x0000 0004	0x4800 1004	0x4800 1404
L4_IA_CORE_L	R	32	0x0000 0018	0x4800 1018	0x4800 1418
L4_IA_CORE_H	R	32	0x0000 001C	0x4800 101C	0x4800 141C
L4_IA_AGENT_CONTROL_L	RW	32	0x0000 0020	0x4800 1020	0x4800 1420
L4_IA_AGENT_CONTROL_H	R	32	0x0000 0024	0x4800 1024	0x4800 1424
L4_IA_AGENT_STATUS_L	RW	32	0x0000 0028	0x4800 1028	0x4800 1428
L4_IA_AGENT_STATUS_H	R	32	0x0000 002C	0x4800 102C	0x4800 142C
L4_IA_ERROR_LOG_L	RW	32	0x0000 0058	0x4800 1058	0x4800 1458
L4_IA_ERROR_LOG_H	R	32	0x0000 005C	0x4800 105C	0x4800 145C

Table 9-368. IA Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	L4_PER1_IA_IP0 L3_MAIN Physical Address	L4_PER1_IA_IP1 L3_MAIN Physical Address
L4_IA_ERROR_LOG_ADDR_L	R	32	0x0000 0060	0x4800 1060	0x4800 1460
L4_IA_ERROR_LOG_ADDR_H	R	32	0x0000 0064	0x4800 1064	0x4800 1464

Table 9-369. IA Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	L4_PER2_IA_IP0 L3_MAIN Physical Address
L4_IA_COMPONENT_L	R	32	0x0000 0000	0x4840 1000
L4_IA_COMPONENT_H	R	32	0x0000 0004	0x4840 1004
L4_IA_CORE_L	R	32	0x0000 0018	0x4840 1018
L4_IA_CORE_H	R	32	0x0000 001C	0x4840 101C
L4_IA_AGENT_CONTROL_L	RW	32	0x0000 0020	0x4840 1020
L4_IA_AGENT_CONTROL_H	R	32	0x0000 0024	0x4840 1024
L4_IA_AGENT_STATUS_L	RW	32	0x0000 0028	0x4840 1028
L4_IA_AGENT_STATUS_H	R	32	0x0000 002C	0x4840 102C
L4_IA_ERROR_LOG_L	RW	32	0x0000 0058	0x4840 1058
L4_IA_ERROR_LOG_H	R	32	0x0000 005C	0x4840 105C
L4_IA_ERROR_LOG_ADDR_L	R	32	0x0000 0060	0x4840 1060
L4_IA_ERROR_LOG_ADDR_H	R	32	0x0000 0064	0x4840 1064

Table 9-370. IA Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	L4_PER3_IA_IP1 L3_MAIN Physical Address	L4_PER3_IA_IP2 L3_MAIN Physical Address
L4_IA_COMPONENT_L	R	32	0x0000 0000	0x4880 1400	0x4880 1800
L4_IA_COMPONENT_H	R	32	0x0000 0004	0x4880 1404	0x4880 1804
L4_IA_CORE_L	R	32	0x0000 0018	0x4880 1418	0x4880 1818
L4_IA_CORE_H	R	32	0x0000 001C	0x4880 141C	0x4880 181C
L4_IA_AGENT_CONTROL_L	RW	32	0x0000 0020	0x4880 1420	0x4880 1820
L4_IA_AGENT_CONTROL_H	R	32	0x0000 0024	0x4880 1424	0x4880 1824
L4_IA_AGENT_STATUS_L	RW	32	0x0000 0028	0x4880 1428	0x4880 1828
L4_IA_AGENT_STATUS_H	R	32	0x0000 002C	0x4880 142C	0x4880 182C
L4_IA_ERROR_LOG_L	RW	32	0x0000 0058	0x4880 1458	0x4880 1858
L4_IA_ERROR_LOG_H	R	32	0x0000 005C	0x4880 145C	0x4880 185C
L4_IA_ERROR_LOG_ADDR_L	R	32	0x0000 0060	0x4880 1460	0x4880 1860
L4_IA_ERROR_LOG_ADDR_H	R	32	0x0000 0064	0x4880 1464	0x4880 1864

Table 9-371. IA Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	L4_CFG_IA_IP0 L3_MAIN Physical Address	L4_WKUP_IA_IP0 L3_MAIN Physical Address
L4_IA_COMPONENT_L	R	32	0x0000 0000	0x4A00 1000	0x4AE0 1000
L4_IA_COMPONENT_H	R	32	0x0000 0004	0x4A00 1004	0x4AE0 1004
L4_IA_CORE_L	R	32	0x0000 0018	0x4A00 1018	0x4AE0 1018
L4_IA_CORE_H	R	32	0x0000 001C	0x4A00 101C	0x4AE0 101C
L4_IA_AGENT_CONTROL_L	RW	32	0x0000 0020	0x4A00 1020	0x4AE0 1020
L4_IA_AGENT_CONTROL_H	R	32	0x0000 0024	0x4A00 1024	0x4AE0 1024
L4_IA_AGENT_STATUS_L	RW	32	0x0000 0028	0x4A00 1028	0x4AE0 1028
L4_IA_AGENT_STATUS_H	R	32	0x0000 002C	0x4A00 102C	0x4AE0 102C
L4_IA_ERROR_LOG_L	RW	32	0x0000 0058	0x4A00 1058	0x4AE0 1058
L4_IA_ERROR_LOG_H	R	32	0x0000 005C	0x4A00 105C	0x4AE0 105C
L4_IA_ERROR_LOG_ADDR_L	R	32	0x0000 0060	0x4A00 1060	0x4AE0 1060
L4_IA_ERROR_LOG_ADDR_H	R	32	0x0000 0064	0x4A00 1064	0x4AE0 1064

9.3.5.2.2 L4 Initiator Agent (L4 IA) Register Description

Table 9-372 through Table 9-394 describe the L4 IA registers.

Table 9-372. L4_IA_COMPONENT_L

Address Offset	0x0000 0000		
Physical Address	0x4800 1000 0x4800 1400 0x4840 1000 0x4880 1400 0x4880 1800 0x4A00 1000 0x4AE0 1000	**Instance**	L4_PER1_IA_IP0 L4_PER1_IA_IP1 L4_PER2_IA_IP0 L4_PER3_IA_IP1 L4_PER3_IA_IP2 L4_CFG_IA_IP0 L4_WKUP_IA_IP0
Description	COMPONENT register identifies the component to which this register block belongs. The register contains a component code and revision, which are used to identify the hardware of the component. The COMPONENT register is read-only.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CODE																REV															

Bits	Field Name	Description	Type	Reset
31:16	CODE	Interconnect code	R	See (1).
15:0	REV	Component revision code	R	See(1).

(1) TI Internal Data

Table 9-373. Register Call Summary for Register L4_IA_COMPONENT_L

L4 Interconnects

- L4 Initiator Agent (L4 IA) Register Summary: [0][1][2][3]

Table 9-374. L4_IA_COMPONENT_H

Address Offset	0x0000 0004		
Physical Address	0x4800 1004 0x4800 1404 0x4840 1004 0x4880 1404 0x4880 1804 0x4A00 1004 0x4AE0 1004	**Instance**	L4_PER1_IA_IP0 L4_PER1_IA_IP1 L4_PER2_IA_IP0 L4_PER3_IA_IP1 L4_PER3_IA_IP2 L4_CFG_IA_IP0 L4_WKUP_IA_IP0
Description	COMPONENT register identifies the component to which this register block belongs. The register contains a component code and revision, which are used to identify the hardware of the component. The COMPONENT register is read-only.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															

Bits	Field Name	Description	Type	Reset
31:0	RESERVED	Read returns 0.	R	0x0000

Table 9-375. Register Call Summary for Register L4_IA_COMPONENT_H

L4 Interconnects
- L4 Initiator Agent (L4 IA) Register Summary: [0][1][2][3]

Table 9-376. L4_IA_CORE_L

Address Offset	0x0000 0018		
Physical Address	0x4800 1018 0x4800 1418 0x4840 1018 0x4880 1418 0x4880 1818 0x4A00 1018 0x4AE0 1018	**Instance**	L4_PER1_IA_IP0 L4_PER1_IA_IP1 L4_PER2_IA_IP0 L4_PER3_IA_IP1 L4_PER3_IA_IP2 L4_CFG_IA_IP0 L4_WKUP_IA_IP0
Description	Provide information about the core initiator		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CORE_CODE																CORE_REV															

Bits	Field Name	Description	Type	Reset
31:16	CORE_CODE	Interconnect core code	R	See (1).
15:0	CORE_REV	Component revision code code	R	See(1).

(1) TI Internal Data

Table 9-377. Register Call Summary for Register L4_IA_CORE_L

L4 Interconnects
- L4 Initiator Agent (L4 IA) Register Summary: [0][1][2][3]

Table 9-378. L4_IA_CORE_H

Address Offset	0x0000 001C		
Physical Address	0x4800 101C 0x4800 141C 0x4840 101C 0x4880 141C 0x4880 181C 0x4A00 101C 0x4AE0 101C	**Instance**	L4_PER1_IA_IP0 L4_PER1_IA_IP1 L4_PER2_IA_IP0 L4_PER3_IA_IP1 L4_PER3_IA_IP2 L4_CFG_IA_IP0 L4_WKUP_IA_IP0
Description	Provide information about the core initiator		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																VENDOR_CODE															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Reserved	R	0x0000
15:0	VENDOR_CODE	Vendor revision core code	R	See (1).

(1) TI Internal Data

Table 9-379. Register Call Summary for Register L4_IA_CORE_H

L4 Interconnects

- L4 Initiator Agent (L4 IA) Register Summary: [0][1][2][3]

Table 9-380. L4_IA_AGENT_CONTROL_L

Address Offset	0x0000 0020		
Physical Address	0x4800 1020 0x4800 1420 0x4840 1020 0x4880 1420 0x4880 1820 0x4A00 1020 0x4AE0 1020	**Instance**	L4_PER1_IA_IP0 L4_PER1_IA_IP1 L4_PER2_IA_IP0 L4_PER3_IA_IP1 L4_PER3_IA_IP2 L4_CFG_IA_IP0 L4_WKUP_IA_IP0
Description	Core control for an initiator OCP interface		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PROT_ERROR_SECONDARY_REP	PROT_ERROR_PRIMARY_REP	RESERVED		INBAND_ERROR_REP	RESERVED		MERROR_REP	RESERVED																							

Bits	Field Name	Description	Type	Reset
31	PROT_ERROR_SECONDARY_REP	Out-of-band reporting of protection mechanism secondary errors	RW	1
30	PROT_ERROR_PRIMARY_REP	Out-of-band reporting of protection mechanism primary errors	RW	1

Bits	Field Name	Description	Type	Reset
29:28	RESERVED	Read returns 0.	R	0x0
27	INBAND_ERROR_REP	Setting this field to 1 reports on in-band errors using the INBAND_ERROR log bit of IA.AGENT_STATUS register. The error reporting mechanism is enabled when the INBAND_ERROR_REP bit field is set to 1.	RW	1
26:25	RESERVED	Read returns 0.	R	0x0
24	MERROR_REP	OCP MError reporting control. The out-of-band OCP MError reporting mechanism is enabled when the MERROR_REP bit field is set to 1.	R	0
23:0	RESERVED		R	0x0

Table 9-381. Register Call Summary for Register L4_IA_AGENT_CONTROL_L

L4 Interconnects

- Firewall Error Logging in the Control Module: [0][1]
- L4 Initiator Agent (L4 IA) Register Summary: [2][3][4][5]

Table 9-382. L4_IA_AGENT_CONTROL_H

Address Offset	0x0000 0024		
Physical Address	0x4800 1024 0x4800 1424 0x4840 1024 0x4880 1424 0x4880 1824 0x4A00 1024 0x4AE0 1024	**Instance**	L4_PER1_IA_IP0 L4_PER1_IA_IP1 L4_PER2_IA_IP0 L4_PER3_IA_IP1 L4_PER3_IA_IP2 L4_CFG_IA_IP0 L4_WKUP_IA_IP0
Description	Enable error reporting on an initiator interface.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															

Bits	Field Name	Description	Type	Reset
31:0	RESERVED	Read returns 0.	R	0x0000 0000

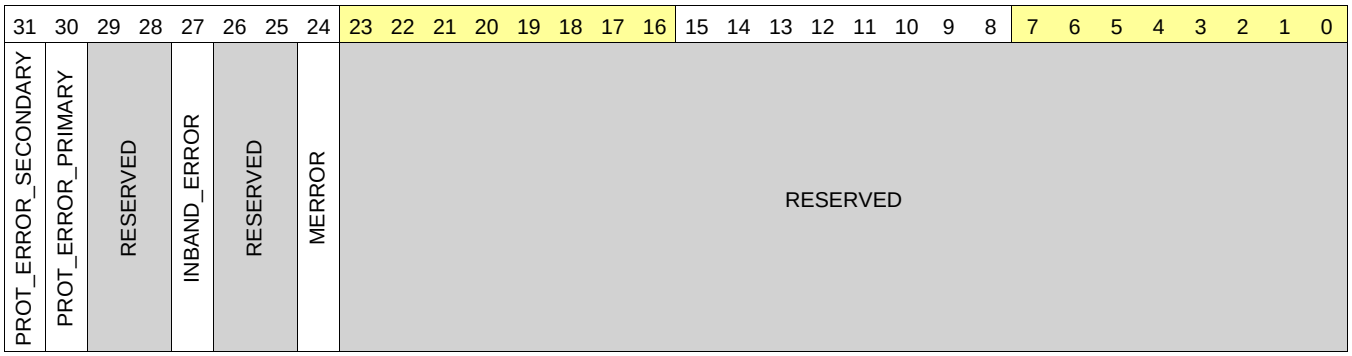
Table 9-383. Register Call Summary for Register L4_IA_AGENT_CONTROL_H

L4 Interconnects

- L4 Initiator Agent (L4 IA) Register Summary: [0][1][2][3]

Table 9-384. L4_IA_AGENT_STATUS_L

Address Offset	0x0000 0028		
Physical Address	0x4800 1028	**Instance**	L4_PER1_IA_IP0
	0x4800 1428		L4_PER1_IA_IP1
	0x4840 1028		L4_PER2_IA_IP0
	0x4880 1428		L4_PER3_IA_IP1
	0x4880 1828		L4_PER3_IA_IP2
	0x4A00 1028		L4_CFG_IA_IP0
	0x4AE0 1028		L4_WKUP_IA_IP0
Description	Stores status information for an initiator. The INBAND_ERROR and MERROR fields are read/write and are implemented as log bits.		
Type	RW		



Bits	Field Name	Description	Type	Reset
31	PROT_ERROR_SECONDARY	0x0: Secondary Protection error not present.0x1: Secondary Protection error present	RW W1toClr	0
30	PROT_ERROR_PRIMARY	0x0: Primary Protection error not present.0x1: Primary Protection error present	RW W1toClr	0
29:28	RESERVED	Read returns 0.	R	0x0
27	INBAND_ERROR	0x0 No In-Band error present.0x1 In-Band error present.	RW W1toClr	0
26:25	RESERVED	Read returns 0.	R	0x0
24	MERROR	Value of the OCP MError signal	R	0
23:0	RESERVED	Read returns 0	R	0X0

Table 9-385. Register Call Summary for Register L4_IA_AGENT_STATUS_L

L4 Interconnects

- IA and TA Error Detection and Logging: [0][1][2][3][4]
- Operational Modes Configuration: [5][6][7][8][9]
- L4 Initiator Agent (L4 IA) Register Summary: [10][11][12][13]

Table 9-386. L4_IA_AGENT_STATUS_H

Address Offset	0x0000 002C		
Physical Address	0x4800 102C 0x4800 142C 0x4840 102C 0x4880 142C 0x4880 182C 0x4A00 102C 0x4AE0 102C	**Instance**	L4_PER1_IA_IP0 L4_PER1_IA_IP1 L4_PER2_IA_IP0 L4_PER3_IA_IP1 L4_PER3_IA_IP2 L4_CFG_IA_IP0 L4_WKUP_IA_IP0
Description	Stores status information for an initiator.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															

Bits	Field Name	Description	Type	Reset
31:0	RESERVED	Read returns 0.	R	0x0000 0000

Table 9-387. Register Call Summary for Register L4_IA_AGENT_STATUS_H

L4 Interconnects

- L4 Initiator Agent (L4 IA) Register Summary: [0][1][2][3]

Table 9-388. L4_IA_ERROR_LOG_L

Address Offset	0x0000 0058		
Physical Address	0x4800 1058 0x4800 1458 0x4840 1058 0x4880 1458 0x4880 1858 0x4A00 1058 0x4AE0 1058	**Instance**	L4_PER1_IA_IP0 L4_PER1_IA_IP1 L4_PER2_IA_IP0 L4_PER3_IA_IP1 L4_PER3_IA_IP2 L4_CFG_IA_IP0 L4_WKUP_IA_IP0
Description	Log information about error conditions. The CODE field logs any protection violation or address hole errors detected by the initiator subsystem while decoding a request.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MULTI	SECONDARY	RESERVED				CODE		RESERVED								CONNID								RESERVED						CMD	

Bits	Field Name	Description	Type	Reset
31	MULTI	Multiple errors detected	RW W1toClr	0
30	SECONDARY	Indicates whether protection violation was a primary or secondary error	RW W1toClr	0
29:26	RESERVED	Read returns 0.	R	0x0
25:24	CODE	The error code of an initiator request. 0x0: No errors 0x1: Unsupported command 0x2: Address hole 0x3: Protection violation	RW W1toClr	0x0
23:14	RESERVED	Read returns 0.	R	0x000
13:8	CONNID	ConnID of request causing the error, refer to Table 9-346	R	0x00

Bits	Field Name	Description	Type	Reset
7:3	RESERVED	Read returns 0.	R	0x00
2:0	CMD	Command that has caused an error.	R	0x0

Table 9-389. Register Call Summary for Register L4_IA_ERROR_LOG_L

L4 Interconnects
- IA and TA Error Detection and Logging: [0][1][2][3][4][5]
- Operational Modes Configuration: [6][7][8][9][10][11][12][13][14][15]
- L4 Initiator Agent (L4 IA) Register Summary: [16][17][18][19]

Table 9-390. L4_IA_ERROR_LOG_H

Address Offset	0x0000 005C		
Physical Address	0x4800 105C 0x4800 145C 0x4840 105C 0x4880 145C 0x4880 185C 0x4A00 105C 0x4AE0 105C	**Instance**	L4_PER1_IA_IP0 L4_PER1_IA_IP1 L4_PER2_IA_IP0 L4_PER3_IA_IP1 L4_PER3_IA_IP2 L4_CFG_IA_IP0 L4_WKUP_IA_IP0
Description	Log information about error conditions.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																REQ_INFO															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Read returns 0.	R	0x0000
15:0	REQ_INFO	MReqInfo bits of request that has caused the error REQ_INFO[0] = supervisor, REQ_INFO[1] = Debug	R	0x0000

Table 9-391. Register Call Summary for Register L4_IA_ERROR_LOG_H

L4 Interconnects
- IA and TA Error Detection and Logging: [0]
- L4 Initiator Agent (L4 IA) Register Summary: [1][2][3][4]

Table 9-392. L4_IA_ERROR_LOG_ADDR_L

Address Offset	0x0000 0060		
Physical Address	0x4800 1060 0x4800 1460 0x4840 1060 0x4880 1460 0x4880 1860 0x4A00 1060 0x4AE0 1060	**Instance**	L4_PER1_IA_IP0 L4_PER1_IA_IP1 L4_PER2_IA_IP0 L4_PER3_IA_IP1 L4_PER3_IA_IP2 L4_CFG_IA_IP0 L4_WKUP_IA_IP0
Description	Extended error log (address information)		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR																															

Bits	Field Name	Description	Type	Reset
31:0	ADDR	Address of request that has caused the error. N is the number MAddr bits.	R	0x0000 0000

Table 9-393. Register Call Summary for Register L4_IA_ERROR_LOG_ADDR_L

L4 Interconnects
- IA and TA Error Detection and Logging: [0]
- Operational Modes Configuration: [1][2]
- L4 Initiator Agent (L4 IA) Register Summary: [3][4][5][6]

Table 9-394. L4_IA_ERROR_LOG_ADDR_H

Address Offset	0x0000 0064		
Physical Address	0x4800 1064 0x4800 1464 0x4840 1064 0x4880 1464 0x4880 1864 0x4A00 1064 0x4AE0 1064	**Instance**	L4_PER1_IA_IP0 L4_PER1_IA_IP1 L4_PER2_IA_IP0 L4_PER3_IA_IP1 L4_PER3_IA_IP2 L4_CFG_IA_IP0 L4_WKUP_IA_IP0
Description	Extended error log (address information)		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															

Bits	Field Name	Description	Type	Reset
31:0	RESERVED	Read returns 0.	R	0x0000 0000

Table 9-395. Register Call Summary for Register L4_IA_ERROR_LOG_ADDR_H

L4 Interconnects
- L4 Initiator Agent (L4 IA) Register Summary: [0][1][2][3]

9.3.5.3 L4 Target Agent (L4 TA)

9.3.5.3.1 L4 Target Agent (L4 TA) Register Summary

Table 9-396 through Table 9-428 summarizes the L4 TA registers.

Table 9-396. L4_CFG TA Register Mapping Summary 1

Register Name	Type	Register Width (Bits)	Address Offset	CTRL_MODULE_CORE_TARG L3_MAIN Physical Address	CM_CORE_AON_TARG L3_MAIN Physical Address	CM_CORE_TARG L3_MAIN Physical Address
L4_TA_COMPONENT_L	R	32	0x0000 0000	0x4A00 4000	0x4A00 6000	0x4A00 A000
L4_TA_COMPONENT_H	R	32	0x0000 0004	0x4A00 4004	0x4A00 6004	0x4A00 A004
L4_TA_CORE_L	R	32	0x0000 0018	0x4A00 4018	0x4A00 6018	0x4A00 A018
L4_TA_CORE_H	R	32	0x0000 001C	0x4A00 401C	0x4A00 601C	0x4A00 A01C
L4_TA_AGENT_CONTROL_L	RW	32	0x0000 0020	0x4A00 4020	0x4A00 6020	0x4A00 A020
L4_TA_AGENT_CONTROL_H	R	32	0x0000 0024	0x4A00 4024	0x4A00 6024	0x4A00 A024

Table 9-396. L4_CFG TA Register Mapping Summary 1 (continued)

Register Name	Type	Register Width (Bits)	Address Offset	CTRL_MODULE_CORE_TARG L3_MAIN Physical Address	CM_CORE_AON_TARG L3_MAIN Physical Address	CM_CORE_TARG L3_MAIN Physical Address
L4_TA_AGENT_STATUS_L	R	32	0x0000 0028	0x4A00 4028	0x4A00 6028	0x4A00 A028
L4_TA_AGENT_STATUS_H	R	32	0x0000 002C	0x4A00 402C	0x4A00 602C	0x4A00 A02C

Table 9-397. L4_CFG TA Register Mapping Summary 2

Register Name	Type	Register Width (Bits)	Address Offset	MAILBOX1_TARG L3_MAIN Physical Address	SPINLOCK_TARG L3_MAIN Physical Address	OCP_WP_NOC_TARG L3_MAIN Physical Address
L4_TA_COMPONENT_L	R	32	0x0000 0000	0x4A0F 5000	0x4A0F 7000	0x4A10 3000
L4_TA_COMPONENT_H	R	32	0x0000 0004	0x4A0F 5004	0x4A0F 7004	0x4A10 3004
L4_TA_CORE_L	R	32	0x0000 0018	0x4A0F 5018	0x4A0F 7018	0x4A10 3018
L4_TA_CORE_H	R	32	0x0000 001C	0x4A0F 501C	0x4A0F 701C	0x4A10 301C
L4_TA_AGENT_CONTROL_L	RW	32	0x0000 0020	0x4A0F 5020	0x4A0F 7020	0x4A10 3020
L4_TA_AGENT_CONTROL_H	R	32	0x0000 0024	0x4A0F 5024	0x4A0F 7024	0x4A10 3024
L4_TA_AGENT_STATUS_H	R	32	0x0000 0028	0x4A0F 5028	0x4A0F 7028	0x4A10 3028
L4_TA_AGENT_STATUS_L	R	32	0x0000 002C	0x4A0F 502C	0x4A0F 702C	0x4A10 302C

Table 9-398. L4_CFG TA Register Mapping Summary 3

Register Name	Type	Register Width (Bits)	Address Offset	EVE_FW_CFG_TARG L3_MAIN Physical Address	IPU_FW_CFG_TARG L3_MAIN Physical Address
L4_TA_COMPONENT_L	R	32	0x0000 0000	0x4A15 2000	0x4A15 C000
L4_TA_COMPONENT_H	R	32	0x0000 0004	0x4A15 2004	0x4A15 C004
L4_TA_CORE_L	R	32	0x0000 0018	0x4A15 2018	0x4A15 C018
L4_TA_CORE_H	R	32	0x0000 001C	0x4A15 201C	0x4A15 C01C
L4_TA_AGENT_CONTROL_L	RW	32	0x0000 0020	0x4A15 2020	0x4A15 C020
L4_TA_AGENT_CONTROL_H	R	32	0x0000 0024	0x4A15 2024	0x4A15 C024
L4_TA_AGENT_STATUS_L	R	32	0x0000 0028	0x4A15 2028	0x4A15 C028
L4_TA_AGENT_STATUS_H	R	32	0x0000 002C	0x4A15 202C	0x4A15 C02C

Table 9-399. L4_CFG TA Register Mapping Summary 4

Register Name	Type	Register Width (Bits)	Address Offset	EDMA_TPCC_FW_CFG_TARG L3_MAIN Physical Address	EDMA_TC0_FW_CFG_TARG L3_MAIN Physical Address
L4_TA_COMPONENT_L	R	32	0x0000 0000	0x4A16 2000	0x4A16 4000

Table 9-399. L4_CFG TA Register Mapping Summary 4 (continued)

Register Name	Type	Register Width (Bits)	Address Offset	EDMA_TPCC_FW_CFG_TARG L3_MAIN Physical Address	EDMA_TC0_FW_CFG_TARG L3_MAIN Physical Address
L4_TA_COMPONENT_H	R	32	0x0000 0004	0x4A16 2004	0x4A16 4004
L4_TA_CORE_L	R	32	0x0000 0018	0x4A16 2018	0x4A16 4018
L4_TA_CORE_H	R	32	0x0000 001C	0x4A16 201C	0x4A16 401C
L4_TA_AGENT_CONTROL_L	RW	32	0x0000 0020	0x4A16 2020	0x4A16 4020
L4_TA_AGENT_CONTROL_H	R	32	0x0000 0024	0x4A16 2024	0x4A16 4024
L4_TA_AGENT_STATUS_L	R	32	0x0000 0028	0x4A16 2028	0x4A16 4028
L4_TA_AGENT_STATUS_H	R	32	0x0000 002C	0x4A16 202C	0x4A16 402C

Table 9-400. L4_CFG TA Register Mapping Summary 5

Register Name	Type	Register Width (Bits)	Address Offset	DSP1_FW_CFG_TARG L3_MAIN Physical Address	DSP2_FW_CFG_TARG L3_MAIN Physical Address	QSPI_FW_CFG_TARG L3_MAIN Physical Address
L4_TA_COMPONENT_L	R	32	0x0000 0000	0x4A17 2000	0x4A17 4000	0x4A17 A000
L4_TA_COMPONENT_H	R	32	0x0000 0004	0x4A17 2004	0x4A17 4004	0x4A17 A004
L4_TA_CORE_L	R	32	0x0000 0018	0x4A17 2018	0x4A17 4018	0x4A17 A018
L4_TA_CORE_H	R	32	0x0000 001C	0x4A17 201C	0x4A17 401C	0x4A17 A01C
L4_TA_AGENT_CONTROL_L	RW	32	0x0000 0020	0x4A17 2020	0x4A17 4020	0x4A17 A020
L4_TA_AGENT_CONTROL_H	R	32	0x0000 0024	0x4A17 2024	0x4A17 4024	0x4A17 A024
L4_TA_AGENT_STATUS_L	R	32	0x0000 0028	0x4A17 2028	0x4A17 4028	0x4A17 A028
L4_TA_AGENT_STATUS_H	R	32	0x0000 002C	0x4A17 202C	0x4A17 402C	0x4A17 A02C

Table 9-401. L4_CFG TA Register Mapping Summary 6

Register Name	Type	Register Width (Bits)	Address Offset	DSS_FW_CFG_TARG L3_MAIN Physical Address	EMIF_FW_CFG_TARG L3_MAIN Physical Address
L4_TA_COMPONENT_L	R	32	0x0000 0000	0x4A21 D000	0x4A20 D000
L4_TA_COMPONENT_H	R	32	0x0000 0004	0x4A21 D004	0x4A20 D004
L4_TA_CORE_L	R	32	0x0000 0018	0x4A21 D018	0x4A20 D018
L4_TA_CORE_H	R	32	0x0000 001C	0x4A21 D01C	0x4A20 D01C
L4_TA_AGENT_CONTROL_L	RW	32	0x0000 0020	0x4A21 D020	0x4A20 D020
L4_TA_AGENT_CONTROL_H	R	32	0x0000 0024	0x4A21 D024	0x4A20 D024
L4_TA_AGENT_STATUS_L	R	32	0x0000 0028	0x4A21 D028	0x4A20 D028
L4_TA_AGENT_STATUS_H	R	32	0x0000 002C	0x4A21 D02C	0x4A20 D02C

Table 9-402. L4_CFG TA Register Mapping Summary 7

Register Name	Type	Register Width (Bits)	Address Offset	GPMC_FW_CFG_TARG L3_MAIN Physical Address	OCMC_RAM_FW_CFG_TARG L3_MAIN Physical Address
L4_TA_COMPONENT_L	R	32	0x0000 0000	0x4A21 1000	0x4A21 3000
L4_TA_COMPONENT_H	R	32	0x0000 0004	0x4A21 1004	0x4A21 3004
L4_TA_CORE_L	R	32	0x0000 0018	0x4A21 1018	0x4A21 3018
L4_TA_CORE_H	R	32	0x0000 001C	0x4A21 101C	0x4A21 301C
L4_TA_AGENT_CONTROL_L	RW	32	0x0000 0020	0x4A21 1020	0x4A21 3020
L4_TA_AGENT_CONTROL_H	R	32	0x0000 0024	0x4A21 1024	0x4A21 3024
L4_TA_AGENT_STATUS_L	R	32	0x0000 0028	0x4A21 1028	0x4A21 3028
L4_TA_AGENT_STATUS_H	R	32	0x0000 002C	0x4A21 102C	0x4A21 302C

Table 9-403. L4_CFG TA Register Mapping Summary 8

Register Name	Type	Register Width (Bits)	Address Offset	DEBUGSS_CT_TBR_FW_CFG_TARG L3_MAIN Physical Address	L3_INSTR_FW_CFG_TARG L3_MAIN Physical Address
L4_TA_COMPONENT_L	R	32	0x0000 0000	0x4A22 5000	0x4A22 7000
L4_TA_COMPONENT_H	R	32	0x0000 0004	0x4A22 5004	0x4A22 7004
L4_TA_CORE_L	R	32	0x0000 0018	0x4A22 5018	0x4A22 7018
L4_TA_CORE_H	R	32	0x0000 001C	0x4A22 501C	0x4A22 701C
L4_TA_AGENT_CONTROL_L	RW	32	0x0000 0020	0x4A22 5020	0x4A22 7020
L4_TA_AGENT_CONTROL_H	R	32	0x0000 0024	0x4A22 5024	0x4A22 7024
L4_TA_AGENT_STATUS_L	R	32	0x0000 0028	0x4A22 5028	0x4A22 7028
L4_TA_AGENT_STATUS_H	R	32	0x0000 002C	0x4A22 502C	0x4A22 702C

Table 9-404. L4_CFG TA Register Mapping Summary 9

Register Name	Type	Register Width (Bits)	Address Offset	IEEE1500_2_OCP_TARG L3_MAIN Physical Address	MCASP1_FW_CFG_TARG L3_MAIN Physical Address	TSC_ADC_FW_CFG_TARG L3_MAIN Physical Address
L4_TA_COMPONENT_L	R	32	0x0000 0000	0x4A10 9000	0x4A16 8000	0x4A16 E000
L4_TA_COMPONENT_H	R	32	0x0000 0004	0x4A10 9004	0x4A16 8004	0x4A16 E004
L4_TA_CORE_L	R	32	0x0000 0018	0x4A10 9018	0x4A16 8018	0x4A16 E018
L4_TA_CORE_H	R	32	0x0000 001C	0x4A10 901C	0x4A16 801C	0x4A16 E01C
L4_TA_AGENT_CONTROL_L	RW	32	0x0000 0020	0x4A10 9020	0x4A16 8020	0x4A16 E020
L4_TA_AGENT_CONTROL_H	R	32	0x0000 0024	0x4A10 9024	0x4A16 8024	0x4A16 E024
L4_TA_AGENT_STATUS_L	R	32	0x0000 0028	0x4A10 9028	0x4A16 8028	0x4A16 E028
L4_TA_AGENT_STATUS_H	R	32	0x0000 002C	0x4A10 902C	0x4A16 802C	0x4A16 E02C

Table 9-405. L4_CFG TA Register Mapping Summary 10

Register Name	Type	Register Width (Bits)	Address Offset	ESM_TARG L3_MAIN Physical Address	ISS_FW_CFG_TARG L3_MAIN Physical Address	TESOC_FW_CFG_TARG L3_MAIN Physical Address
L4_TA_COMPONENT_L	R	32	0x0000 0000	0x4A23 D000	0x4A23 F000	0x4A24 1000
L4_TA_COMPONENT_H	R	32	0x0000 0004	0x4A23 D004	0x4A23 F004	0x4A24 1004
L4_TA_CORE_L	R	32	0x0000 0018	0x4A23 D018	0x4A23 F018	0x4A24 1018
L4_TA_CORE_H	R	32	0x0000 001C	0x4A23 D01C	0x4A23 F01C	0x4A24 101C
L4_TA_AGENT_CONTROL_L	RW	32	0x0000 0020	0x4A23 D020	0x4A23 F020	0x4A24 1020
L4_TA_AGENT_CONTROL_H	R	32	0x0000 0024	0x4A23 D024	0x4A23 F024	0x4A24 1024
L4_TA_AGENT_STATUS_L	R	32	0x0000 0028	0x4A23 D028	0x4A23 F028	0x4A24 1028
L4_TA_AGENT_STATUS_H	R	32	0x0000 002C	0x4A23 D02C	0x4A23 F02C	0x4A24 102C

Table 9-406. L4_CFG TA Register Mapping Summary 11

Register Name	Type	Register Width (Bits)	Address Offset	CRC_FW_CFG_TARG L3_MAIN Physical Address	CRC_CFG_TARG L3_MAIN Physical Address	TSC_ADC_CFG_TARG L3_MAIN Physical Address
L4_TA_COMPONENT_L	R	32	0x0000 0000	0x4A24 5000	0x4A26 1000	0x4A26 5000
L4_TA_COMPONENT_H	R	32	0x0000 0004	0x4A24 5004	0x4A26 1004	0x4A26 5004
L4_TA_CORE_L	R	32	0x0000 0018	0x4A24 5018	0x4A26 1018	0x4A26 5018
L4_TA_CORE_H	R	32	0x0000 001C	0x4A24 501C	0x4A26 101C	0x4A26 501C
L4_TA_AGENT_CONTROL_L	RW	32	0x0000 0020	0x4A24 5020	0x4A26 1020	0x4A26 5020
L4_TA_AGENT_CONTROL_H	R	32	0x0000 0024	0x4A24 5024	0x4A26 1024	0x4A26 5024
L4_TA_AGENT_STATUS_L	R	32	0x0000 0028	0x4A24 5028	0x4A26 1028	0x4A26 5028
L4_TA_AGENT_STATUS_H	R	32	0x0000 002C	0x4A24 502C	0x4A26 102C	0x4A26 502C

Table 9-407. L4_PER1 TA Register Mapping Summary 1

Register Name	Type	Register Width (Bits)	Address Offset	UART3_TARG L3_MAIN Physical Address	TIMER2_TARG L3_MAIN Physical Address	TIMER3_TARG L3_MAIN Physical Address
L4_TA_COMPONENT_L	R	32	0x0000 0000	0x4802 1000	0x4803 3000	0x4803 5000
L4_TA_COMPONENT_H	R	32	0x0000 0004	0x4802 1004	0x4803 3004	0x4803 5004
L4_TA_CORE_L	R	32	0x0000 0018	0x4802 1018	0x4803 3018	0x4803 5018
L4_TA_CORE_H	R	32	0x0000 001C	0x4802 101C	0x4803 301C	0x4803 501C
L4_TA_AGENT_CONTROL_L	RW	32	0x0000 0020	0x4802 1020	0x4803 3020	0x4803 5020
L4_TA_AGENT_CONTROL_H	R	32	0x0000 0024	0x4802 1024	0x4803 3024	0x4803 5024
L4_TA_AGENT_STATUS_L	R	32	0x0000 0028	0x4802 1028	0x4803 3028	0x4803 5028
L4_TA_AGENT_STATUS_H	R	32	0x0000 002C	0x4802 102C	0x4803 302C	0x4803 502C

Table 9-408. L4_PER1 TA Register Mapping Summary 2

Register Name	Type	Register Width (Bits)	Address Offset	TIMER4_TARG L3_MAIN Physical Address	GPIO2_TARG L3_MAIN Physical Address
L4_TA_COMPONENT_L	R	32	0x0000 0000	0x4803 7000	0x4805 6000
L4_TA_COMPONENT_H	R	32	0x0000 0004	0x4803 7004	0x4805 6004
L4_TA_CORE_L	R	32	0x0000 0018	0x4803 7018	0x4805 6018
L4_TA_CORE_H	R	32	0x0000 001C	0x4803 701C	0x4805 601C
L4_TA_AGENT_CONTROL_L	RW	32	0x0000 0020	0x4803 7020	0x4805 6020
L4_TA_AGENT_CONTROL_H	R	32	0x0000 0024	0x4803 7024	0x4805 6024
L4_TA_AGENT_STATUS_L	R	32	0x0000 0028	0x4803 7028	0x4805 6028
L4_TA_AGENT_STATUS_H	R	32	0x0000 002C	0x4803 702C	0x4805 602C

Table 9-409. L4_PER1 TA Register Mapping Summary 3

Register Name	Type	Register Width (Bits)	Address Offset	GPIO3_TARG L3_MAIN Physical Address	GPIO4_TARG L3_MAIN Physical Address
L4_TA_COMPONENT_L	R	32	0x0000 0000	0x4805 8000	0x4805 A000
L4_TA_COMPONENT_H	R	32	0x0000 0004	0x4805 8004	0x4805 A004
L4_TA_CORE_L	R	32	0x0000 0018	0x4805 8018	0x4805 A018
L4_TA_CORE_H	R	32	0x0000 001C	0x4805 801C	0x4805 A01C
L4_TA_AGENT_CONTROL_L	RW	32	0x0000 0020	0x4805 8020	0x4805 A020
L4_TA_AGENT_CONTROL_H	R	32	0x0000 0024	0x4805 8024	0x4805 A024
L4_TA_AGENT_STATUS_L	R	32	0x0000 0028	0x4805 8028	0x4805 A028
L4_TA_AGENT_STATUS_H	R	32	0x0000 002C	0x4805 802C	0x4805 A02C

Table 9-410. L4_PER1 TA Register Mapping Summary 4

Register Name	Type	Register Width (Bits)	Address Offset	UART1_TARG L3_MAIN Physical Address	UART2_TARG L3_MAIN Physical Address
L4_TA_COMPONENT_L	R	32	0x0000 0000	0x4806 B000	0x4806 D000
L4_TA_COMPONENT_H	R	32	0x0000 0004	0x4806 B004	0x4806 D004
L4_TA_CORE_L	R	32	0x0000 0018	0x4806 B018	0x4806 D018
L4_TA_CORE_H	R	32	0x0000 001C	0x4806 B01C	0x4806 D01C
L4_TA_AGENT_CONTROL_L	RW	32	0x0000 0020	0x4806 B020	0x4806 D020
L4_TA_AGENT_CONTROL_H	R	32	0x0000 0024	0x4806 B024	0x4806 D024
L4_TA_AGENT_STATUS_L	R	32	0x0000 0028	0x4806 B028	0x4806 D028
L4_TA_AGENT_STATUS_H	R	32	0x0000 002C	0x4806 B02C	0x4806 D02C

Table 9-411. L4_PER1 TA Register Mapping Summary 5

Register Name	Type	Register Width (Bits)	Address Offset	I2C1_TARG L3_MAIN Physical Address	I2C2_TARG L3_MAIN Physical Address
L4_TA_COMPONENT_L	R	32	0x0000 0000	0x4807 1000	0x4807 3000
L4_TA_COMPONENT_H	R	32	0x0000 0004	0x4807 1004	0x4807 3004
L4_TA_CORE_L	R	32	0x0000 0018	0x4807 1018	0x4807 3018
L4_TA_CORE_H	R	32	0x0000 001C	0x4807 101C	0x4807 301C
L4_TA_AGENT_CONTROL_L	RW	32	0x0000 0020	0x4807 1020	0x4807 3020
L4_TA_AGENT_CONTROL_H	R	32	0x0000 0024	0x4807 1024	0x4807 3024
L4_TA_AGENT_STATUS_L	R	32	0x0000 0028	0x4807 1028	0x4807 3028
L4_TA_AGENT_STATUS_H	R	32	0x0000 002C	0x4807 102C	0x4807 302C

Table 9-412. L4_PER1 TA Register Mapping Summary 6

Register Name	Type	Register Width (Bits)	Address Offset	ELM_TARG L3_MAIN Physical Address	MCSPI1_TARG L3_MAIN Physical Address	MCSPI2_TARG L3_MAIN Physical Address
L4_TA_COMPONENT_L	R	32	0x0000 0000	0x4807 9000	0x4809 9000	0x4809 B000
L4_TA_COMPONENT_H	R	32	0x0000 0004	0x4807 9004	0x4809 9004	0x4809 B004
L4_TA_CORE_L	R	32	0x0000 0018	0x4807 9018	0x4809 9018	0x4809 B018
L4_TA_CORE_H	R	32	0x0000 001C	0x4807 901C	0x4809 901C	0x4809 B01C
L4_TA_AGENT_CONTROL_L	RW	32	0x0000 0020	0x4807 9020	0x4809 9020	0x4809 B020
L4_TA_AGENT_CONTROL_H	R	32	0x0000 0024	0x4807 9024	0x4809 9024	0x4809 B024
L4_TA_AGENT_STATUS_L	R	32	0x0000 0028	0x4807 9028	0x4809 9028	0x4809 B028
L4_TA_AGENT_STATUS_H	R	32	0x0000 002C	0x4807 902C	0x4809 902C	0x4809 B02C

Table 9-413. L4_PER1 TA Register Mapping Summary 7

Register Name	Type	Register Width (Bits)	Address Offset	MCSPI3_TARG L3_MAIN Physical Address	MCSPI4_TARG L3_MAIN Physical Address	MMC_TARG L3_MAIN Physical Address
L4_TA_COMPONENT_L	R	32	0x0000 0000	0x480B 9000	0x480B B000	0x480D 2000
L4_TA_COMPONENT_H	R	32	0x0000 0004	0x480B 9004	0x480B B004	0x480D 2004
L4_TA_CORE_L	R	32	0x0000 0018	0x480B 9018	0x480B B018	0x480D 2018
L4_TA_CORE_H	R	32	0x0000 001C	0x480B 901C	0x480B B01C	0x480D 201C
L4_TA_AGENT_CONTROL_L	RW	32	0x0000 0020	0x480B 9020	0x480B B020	0x480D 2020
L4_TA_AGENT_CONTROL_H	R	32	0x0000 0024	0x480B 9024	0x480B B024	0x480D 2024
L4_TA_AGENT_STATUS_L	R	32	0x0000 0028	0x480B 9028	0x480B B028	0x480D 2028
L4_TA_AGENT_STATUS_H	R	32	0x0000 002C	0x480B 902C	0x480B B02C	0x480D 202C

Table 9-414. L4_PER2 TA Register Mapping Summary 1

Register Name	Type	Register Width (Bits)	Address Offset	GMAC_SW_TARG L3_MAIN Physical Address	MCASP1_CFG_TARG L3_MAIN Physical Address
L4_TA_COMPONENT_L	R	32	0x0000 0000	0x4848 8000	0x4846 2000
L4_TA_COMPONENT_H	R	32	0x0000 0004	0x4848 8004	0x4846 2004
L4_TA_CORE_L	R	32	0x0000 0018	0x4848 8018	0x4846 2018
L4_TA_CORE_H	R	32	0x0000 001C	0x4848 801C	0x4846 201C
L4_TA_AGENT_CONTROL_L	RW	32	0x0000 0020	0x4848 8020	0x4846 2020
L4_TA_AGENT_CONTROL_H	R	32	0x0000 0024	0x4848 8024	0x4846 2024
L4_TA_AGENT_STATUS_L	R	32	0x0000 0028	0x4848 8028	0x4846 2028
L4_TA_AGENT_STATUS_H	R	32	0x0000 002C	0x4848 802C	0x4846 202C

Table 9-415. L4_PER2 TA Register Mapping Summary 2

Register Name	Type	Register Width (Bits)	Address Offset	PWMSS_TARG L3_MAIN Physical Address	MCAN_TARG L3_MAIN Physical Address	ATL_TARG L3_MAIN Physical Address
L4_TA_COMPONENT_L	R	32	0x0000 0000	0x4843 F000	0x4848 2000	0x4843 D000
L4_TA_COMPONENT_H	R	32	0x0000 0004	0x4843 F004	0x4848 2004	0x4843 D004
L4_TA_CORE_L	R	32	0x0000 0018	0x4843 F018	0x4848 2018	0x4843 D018
L4_TA_CORE_H	R	32	0x0000 001C	0x4843 F01C	0x4848 201C	0x4843 D01C
L4_TA_AGENT_CONTROL_L	RW	32	0x0000 0020	0x4843 F020	0x4848 2020	0x4843 D020
L4_TA_AGENT_CONTROL_H	R	32	0x0000 0024	0x4843 F024	0x4848 2024	0x4843 D024
L4_TA_AGENT_STATUS_L	R	32	0x0000 0028	0x4843 F028	0x4848 2028	0x4843 D028
L4_TA_AGENT_STATUS_H	R	32	0x0000 002C	0x4843 F02C	0x4848 202C	0x4843 D02C

Table 9-416. L4_PER2 TA Register Mapping Summary 3

Register Name	Type	Register Width (Bits)	Address Offset	MCASP2_DAT_TARG L3_MAIN Physical Address	MCASP2_CFG_TARG L3_MAIN Physical Address
L4_TA_COMPONENT_L	R	32	0x0000 0000	0x4843 7000	0x4846 E000
L4_TA_COMPONENT_H	R	32	0x0000 0004	0x4843 7004	0x4846 E004
L4_TA_CORE_L	R	32	0x0000 0018	0x4843 7018	0x4846 E018
L4_TA_CORE_H	R	32	0x0000 001C	0x4843 701C	0x4846 E01C
L4_TA_AGENT_CONTROL_L	RW	32	0x0000 0020	0x4843 7020	0x4846 E020
L4_TA_AGENT_CONTROL_H	R	32	0x0000 0024	0x4843 7024	0x4846 E024
L4_TA_AGENT_STATUS_L	R	32	0x0000 0028	0x4843 7028	0x4846 E028
L4_TA_AGENT_STATUS_H	R	32	0x0000 002C	0x4843 702C	0x4846 E02C

Table 9-417. L4_PER2 TA Register Mapping Summary 4

Register Name	Type	Register Width (Bits)	Address Offset	MCASP3_DAT_TARG L3_MAIN Physical Address	MCASP3_CFG_TARG L3_MAIN Physical Address
L4_TA_COMPONENT_L	R	32	0x0000 0000	0x4843 B000	0x4847 2000
L4_TA_COMPONENT_H	R	32	0x0000 0004	0x4843 B004	0x4847 2004
L4_TA_CORE_L	R	32	0x0000 0018	0x4843 B018	0x4847 2018
L4_TA_CORE_H	R	32	0x0000 001C	0x4843 B01C	0x4847 201C
L4_TA_AGENT_CONTROL_L	RW	32	0x0000 0020	0x4843 B020	0x4847 2020
L4_TA_AGENT_CONTROL_H	R	32	0x0000 0024	0x4843 B024	0x4847 2024
L4_TA_AGENT_STATUS_L	R	32	0x0000 0028	0x4843 B028	0x4847 2028
L4_TA_AGENT_STATUS_H	R	32	0x0000 002C	0x4843 B02C	0x4847 202C

Table 9-418. L4_PER3 TA Register Mapping Summary 1

Register Name	Type	Register Width (Bits)	Address Offset	OCMC_RAM_CFG_TARG L3_MAIN Physical Address	MMU_TARG L3_MAIN Physical Address
L4_TA_COMPONENT_L	R	32	0x0000 0000	0x4880 5000	0x4881 D000
L4_TA_COMPONENT_H	R	32	0x0000 0004	0x4880 5004	0x4881 D004

Table 9-418. L4_PER3 TA Register Mapping Summary 1 (continued)

Register Name	Type	Register Width (Bits)	Address Offset	OCMC_RAM_CFG_TARG L3_MAIN Physical Address	MMU_TARG L3_MAIN Physical Address
L4_TA_CORE_L	R	32	0x0000 0018	0x4880 5018	0x4881 D018
L4_TA_CORE_H	R	32	0x0000 001C	0x4880 501C	0x4881 D01C
L4_TA_AGENT_CONTROL_L	RW	32	0x0000 0020	0x4880 5020	0x4881 D020
L4_TA_AGENT_CONTROL_H	R	32	0x0000 0024	0x4880 5024	0x4881 D024
L4_TA_AGENT_STATUS_L	R	32	0x0000 0028	0x4880 5028	0x4881 D028
L4_TA_AGENT_STATUS_H	R	32	0x0000 002C	0x4880 502C	0x4881 D02C

Table 9-419. L4_PER3 TA Register Mapping Summary 2

Register Name	Type	Register Width (Bits)	Address Offset	TIMER5_TARG L3_MAIN Physical Address	TIMER6_TARG L3_MAIN Physical Address
L4_TA_COMPONENT_L	R	32	0x0000 0000	0x4882 1000	0x4882 3000
L4_TA_COMPONENT_H	R	32	0x0000 0004	0x4882 1004	0x4882 3004
L4_TA_CORE_L	R	32	0x0000 0018	0x4882 1018	0x4882 3018
L4_TA_CORE_H	R	32	0x0000 001C	0x4882 101C	0x4882 301C
L4_TA_AGENT_CONTROL_L	RW	32	0x0000 0020	0x4882 1020	0x4882 3020
L4_TA_AGENT_CONTROL_H	R	32	0x0000 0024	0x4882 1024	0x4882 3024
L4_TA_AGENT_STATUS_L	R	32	0x0000 0028	0x4882 1028	0x4882 3028
L4_TA_AGENT_STATUS_H	R	32	0x0000 002C	0x4882 102C	0x4882 302C

Table 9-420. L4_PER3 TA Register Mapping Summary 3

Register Name	Type	Register Width (Bits)	Address Offset	TIMER7_TARG L3_MAIN Physical Address	TIMER8_TARG L3_MAIN Physical Address
L4_TA_COMPONENT_L	R	32	0x0000 0000	0x4882 5000	0x4882 7000
L4_TA_COMPONENT_H	R	32	0x0000 0004	0x4882 5004	0x4882 7004
L4_TA_CORE_L	R	32	0x0000 0018	0x4882 5018	0x4882 7018
L4_TA_CORE_H	R	32	0x0000 001C	0x4882 501C	0x4882 701C
L4_TA_AGENT_CONTROL_L	RW	32	0x0000 0020	0x4882 5020	0x4882 7020
L4_TA_AGENT_CONTROL_H	R	32	0x0000 0024	0x4882 5024	0x4882 7024
L4_TA_AGENT_STATUS_L	R	32	0x0000 0028	0x4882 5028	0x4882 7028
L4_TA_AGENT_STATUS_H	R	32	0x0000 002C	0x4882 502C	0x4882 702C

Table 9-421. L4_PER3 TA Register Mapping Summary 4

Register Name	Type	Register Width (Bits)	Address Offset	MAILBOX2_TARG L3_MAIN Physical Address	VIP_TARG L3_MAIN Physical Address
L4_TA_COMPONENT_L	R	32	0x0000 0000	0x4883 B000	0x4898 0000
L4_TA_COMPONENT_H	R	32	0x0000 0004	0x4883 B004	0x4898 0004
L4_TA_CORE_L	R	32	0x0000 0018	0x4883 B018	0x4898 0018
L4_TA_CORE_H	R	32	0x0000 001C	0x4883 B01C	0x4898 001C
L4_TA_AGENT_CONTROL_L	RW	32	0x0000 0020	0x4883 B020	0x4898 0020
L4_TA_AGENT_CONTROL_H	R	32	0x0000 0024	0x4883 B024	0x4898 0024
L4_TA_AGENT_STATUS_L	R	32	0x0000 0028	0x4883 B028	0x4898 0028
L4_TA_AGENT_STATUS_H	R	32	0x0000 002C	0x4883 B02C	0x4898 002C

Table 9-422. L4_PER3 TA Register Mapping Summary 5

Register Name	Type	Register Width (Bits)	Address Offset	DCC1_TARG L3_MAIN Physical Address	DCC2_TARG L3_MAIN Physical Address
L4_TA_COMPONENT_L	R	32	0x0000 0000	0x4882 9000	0x4882 B000
L4_TA_COMPONENT_H	R	32	0x0000 0004	0x4882 9004	0x4882 B004
L4_TA_CORE_L	R	32	0x0000 0018	0x4882 9018	0x4882 B018
L4_TA_CORE_H	R	32	0x0000 001C	0x4882 901C	0x4882 B01C
L4_TA_AGENT_CONTROL_L	RW	32	0x0000 0020	0x4882 9020	0x4882 B020
L4_TA_AGENT_CONTROL_H	R	32	0x0000 0024	0x4882 9024	0x4882 B024
L4_TA_AGENT_STATUS_L	R	32	0x0000 0028	0x4882 9028	0x4882 B028
L4_TA_AGENT_STATUS_H	R	32	0x0000 002C	0x4882 902C	0x4882 B02C

Table 9-423. L4_PER3 TA Register Mapping Summary 6

Register Name	Type	Register Width (Bits)	Address Offset	DCC3_TARG L3_MAIN Physical Address	DCC4_TARG L3_MAIN Physical Address
L4_TA_COMPONENT_L	R	32	0x0000 0000	0x4882 D000	0x4882 F000
L4_TA_COMPONENT_H	R	32	0x0000 0004	0x4882 D004	0x4882 F004
L4_TA_CORE_L	R	32	0x0000 0018	0x4882 D018	0x4882 F018
L4_TA_CORE_H	R	32	0x0000 001C	0x4882 D01C	0x4882 F01C
L4_TA_AGENT_CONTROL_L	RW	32	0x0000 0020	0x4882 D020	0x4882 F020
L4_TA_AGENT_CONTROL_H	R	32	0x0000 0024	0x4882 D024	0x4882 F024
L4_TA_AGENT_STATUS_L	R	32	0x0000 0028	0x4882 D028	0x4882 F028
L4_TA_AGENT_STATUS_H	R	32	0x0000 002C	0x4882 D02C	0x4882 F02C

Table 9-424. L4_PER3 TA Register Mapping Summary 7

Register Name	Type	Register Width (Bits)	Address Offset	DCC5_TARG L3_MAIN Physical Address	DCC6_TARG L3_MAIN Physical Address	DCC7_TARG L3_MAIN Physical Address
L4_TA_COMPONENT_L	R	32	0x0000 0000	0x4884 F000	0x4885 3000	0x4885 5000
L4_TA_COMPONENT_H	R	32	0x0000 0004	0x4884 F004	0x4885 3004	0x4885 5004
L4_TA_CORE_L	R	32	0x0000 0018	0x4884 F018	0x4885 3018	0x4885 5018
L4_TA_CORE_H	R	32	0x0000 001C	0x4884 F01C	0x4885 301C	0x4885 501C
L4_TA_AGENT_CONTROL_L	RW	32	0x0000 0020	0x4884 F020	0x4885 3020	0x4885 5020
L4_TA_AGENT_CONTROL_H	R	32	0x0000 0024	0x4884 F024	0x4885 3024	0x4885 5024
L4_TA_AGENT_STATUS_L	R	32	0x0000 0028	0x4884 F028	0x4885 3028	0x4885 5028
L4_TA_AGENT_STATUS_H	R	32	0x0000 002C	0x4884 F02C	0x4885 302C	0x4885 502C

Table 9-425. L4_WKUP TA Register Mapping Summary 1

Register Name	Type	Register Width (Bits)	Address Offset	COUNTER_32K_TARG L3_MAIN Physical Address	PRM_TARG L3_MAIN Physical Address	CTRL_MODULE_WKUP_TARG L3_MAIN Physical Address
L4_TA_COMPONENT_L	R	32	0x0000 0000	0x4AE0 5000	0x4AE0 8000	0x4AE0 D000
L4_TA_COMPONENT_H	R	32	0x0000 0004	0x4AE0 5004	0x4AE0 8004	0x4AE0 D004
L4_TA_CORE_L	R	32	0x0000 0018	0x4AE0 5018	0x4AE0 8018	0x4AE0 D018

Table 9-425. L4_WKUP TA Register Mapping Summary 1 (continued)

Register Name	Type	Register Width (Bits)	Address Offset	COUNTER_32K_TARG L3_MAIN Physical Address	PRM_TARG L3_MAIN Physical Address	CTRL_MODULE_WKUP_TARG L3_MAIN Physical Address
L4_TA_CORE_H	R	32	0x0000 001C	0x4AE0 501C	0x4AE0 801C	0x4AE0 D01C
L4_TA_AGENT_CONTROL_L	RW	32	0x0000 0020	0x4AE0 5020	0x4AE0 8020	0x4AE0 D020
L4_TA_AGENT_CONTROL_H	R	32	0x0000 0024	0x4AE0 5024	0x4AE0 8024	0x4AE0 D024
L4_TA_AGENT_STATUS_L	R	32	0x0000 0028	0x4AE0 5028	0x4AE0 8028	0x4AE0 D028
L4_TA_AGENT_STATUS_H	R	32	0x0000 002C	0x4AE0 502C	0x4AE0 802C	0x4AE0 D02C

Table 9-426. L4_WKUP TA Register Mapping Summary 2

Register Name	Type	Register Width (Bits)	Address Offset	GPIO1_TARG L3_MAIN Physical Address	TIMER1_TARG L3_MAIN Physical Address	DCAN_TARG L3_MAIN Physical Address
L4_TA_COMPONENT_L	R	32	0x0000 0000	0x4AE1 1000	0x4AE1 9000	0x4AE3 E000
L4_TA_COMPONENT_H	R	32	0x0000 0004	0x4AE1 1004	0x4AE1 9004	0x4AE3 E004
L4_TA_CORE_L	R	32	0x0000 0018	0x4AE1 1018	0x4AE1 9018	0x4AE3 E018
L4_TA_CORE_H	R	32	0x0000 001C	0x4AE1 101C	0x4AE1 901C	0x4AE3 E01C
L4_TA_AGENT_CONTROL_L	RW	32	0x0000 0020	0x4AE1 1020	0x4AE1 9020	0x4AE3 E020
L4_TA_AGENT_CONTROL_H	R	32	0x0000 0024	0x4AE1 1024	0x4AE1 9024	0x4AE3 E024
L4_TA_AGENT_STATUS_L	R	32	0x0000 0028	0x4AE1 1028	0x4AE1 9028	0x4AE3 E028
L4_TA_AGENT_STATUS_H	R	32	0x0000 002C	0x4AE1 102C	0x4AE1 902C	0x4AE3 E02C

Table 9-427. L4_WKUP TA Register Mapping Summary 3

Register Name	Type	Register Width (Bits)	Address Offset	RTI1_TARG L3_MAIN Physical Address	RTI2_TARG L3_MAIN Physical Address	RTI3_TARG L3_MAIN Physical Address
L4_TA_COMPONENT_L	R	32	0x0000 0000	0x4AE3 2000	0x4AE3 4000	0x4AE3 6000
L4_TA_COMPONENT_H	R	32	0x0000 0004	0x4AE3 2004	0x4AE3 4004	0x4AE3 6004
L4_TA_CORE_L	R	32	0x0000 0018	0x4AE3 2018	0x4AE3 4018	0x4AE3 6018
L4_TA_CORE_H	R	32	0x0000 001C	0x4AE3 201C	0x4AE3 401C	0x4AE3 601C
L4_TA_AGENT_CONTROL_L	RW	32	0x0000 0020	0x4AE3 2020	0x4AE3 4020	0x4AE3 6020
L4_TA_AGENT_CONTROL_H	R	32	0x0000 0024	0x4AE3 2024	0x4AE3 4024	0x4AE3 6024
L4_TA_AGENT_STATUS_L	R	32	0x0000 0028	0x4AE3 2028	0x4AE3 4028	0x4AE3 6028
L4_TA_AGENT_STATUS_H	R	32	0x0000 002C	0x4AE3 202C	0x4AE3 402C	0x4AE3 602C

Table 9-428. L4_WKUP TA Register Mapping Summary 4

Register Name	Type	Register Width (Bits)	Address Offset	RTI4_TARG L3_MAIN Physical Address	RTI5_TARG L3_MAIN Physical Address
L4_TA_COMPONENT_L	R	32	0x0000 0000	0x4AE3 8000	0x4AE3 A000
L4_TA_COMPONENT_H	R	32	0x0000 0004	0x4AE3 8004	0x4AE3 A004
L4_TA_CORE_L	R	32	0x0000 0018	0x4AE3 8018	0x4AE3 A018
L4_TA_CORE_H	R	32	0x0000 001C	0x4AE3 801C	0x4AE3 A01C
L4_TA_AGENT_CONTROL_L	RW	32	0x0000 0020	0x4AE3 8020	0x4AE3 A020
L4_TA_AGENT_CONTROL_H	R	32	0x0000 0024	0x4AE3 8024	0x4AE3 A024
L4_TA_AGENT_STATUS_L	R	32	0x0000 0028	0x4AE3 8028	0x4AE3 A028
L4_TA_AGENT_STATUS_H	R	32	0x0000 002C	0x4AE3 802C	0x4AE3 A02C

9.3.5.3.2 L4 Target Agent (L4 TA) Register Description

Table 9-429 through Table 9-443 describe the L4 TA registers.

Table 9-429. L4_TA_COMPONENT_H

Address Offset	0x0000 0004		
Physical Address	See Table 9-396 to Table 9-428	**Instance**	See Table 9-396 to Table 9-428
Description	Contains a component code and revision.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															

Bits	Field Name	Description	Type	Reset
31:0	RESERVED	Read returns 0	R	0x0000 000

Table 9-430. Register Call Summary for Register L4_TA_COMPONENT_H

L4 Interconnects

- L4 Target Agent (L4 TA) Register Summary: [0][1][2][3][4][5][6][7][8][9][10][11][12][13][14][15][16][17][18][19][20][21][22][23][24][25][26][27][28][29][30]

Table 9-431. L4_TA_COMPONENT_L

Address Offset	0x0000 0000		
Physical Address	See Table 9-396 to Table 9-428	**Instance**	See Table 9-396 to Table 9-428
Description	Contains a component code and revision.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CODE																REV															

Bits	Field Name	Description	Type	Reset
31:16	CODE	Interconnect code.	R	See (1).
15:0	REV	Component revision code.	R	See (1).

(1) TI Internal Data

Table 9-432. Register Call Summary for Register L4_TA_COMPONENT_L

L4 Interconnects

- L4 Target Agent (L4 TA) Register Summary: [0][1][2][3][4][5][6][7][8][9][10][11][12][13][14][15][16][17][18][19][20][21][22][23][24][25][26][27][28][29][30]

Table 9-433. L4_TA_CORE_L

Address Offset	0x0000 0018		
Physical Address	See Table 9-396 to Table 9-428	**Instance**	See Table 9-396 to Table 9-428
Description	Contains a component code and revision.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CORE_CODE																CORE_REV															

Bits	Field Name	Description	Type	Reset
31:16	CORE_CODE	Interconnect core code	R	See (1).
15:0	CORE_REV	Component revision code code	R	See (1).

(1) TI Internal Data

Table 9-434. Register Call Summary for Register L4_TA_CORE_L

L4 Interconnects

- L4 Target Agent (L4 TA) Register Summary: [0][1][2][3][4][5][6][7][8][9][10][11][12][13][14][15][16][17][18][19][20][21][22][23][24][25][26][27][28][29][30]

Table 9-435. L4_TA_CORE_H

Address Offset	0x0000 001C		
Physical Address	See Table 9-396 to Table 9-428	**Instance**	See Table 9-396 to Table 9-428
Description	Contains a component code and revision.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																VENDOR_CODE															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Reserved	R	0x0000
15:0	VENDOR_CODE	Vendor revision core code	R	See (1).

(1) TI Internal Data

Table 9-436. Register Call Summary for Register L4_TA_CORE_H

L4 Interconnects

- L4 Target Agent (L4 TA) Register Summary: [0][1][2][3][4][5][6][7][8][9][10][11][12][13][14][15][16][17][18][19][20][21][22][23][24][25][26][27][28][29][30]

Table 9-437. L4_TA_AGENT_CONTROL_L

Address Offset	0x0000 0020		
Physical Address	See Table 9-396 to Table 9-428	**Instance**	See Table 9-396 to Table 9-428
Description	Enable error reporting		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED							SERROR_REP	RESERVED													REQ_TIMEOUT			RESERVED							OCP_RESET

Bits	Field Name	Description	Type	Reset
31:25	RESERVED	Read returns 0.	R	0x00
24	SERROR_REP	Enable logging of error	R	0x0
23:11	RESERVED	Read returns 0.	R	0x0
10:8	REQ_TIMEOUT	Time-out Bound. Values are: 0 - No time-out 1 - 1x base cycles. 2 - 4x base cycles. 3 - 16x base cycles. 4 - 64x base cycles.	RW	0x2
7:1	RESERVED	Read returns 0.	R	0x00
0	OCP_RESET	The OCP_RESET field controls the OCP reset signal to the attached core. Setting this bit clears any pending transfers and resets the OCP interface. The bit must be cleared to deassert the OCP reset signal. When the software reset feature is available on a target agent, the target agent OCP must also have a reset signal directed to the target core.	RW	0

Table 9-438. Register Call Summary for Register L4_TA_AGENT_CONTROL_L

L4 Interconnects

- Time-Out: [0][1][2][3][4]
- Error Recovery: [5][6]
- Operational Modes Configuration: [7][8][9][10]
- L4 Target Agent (L4 TA) Register Summary: [11][12][13][14][15][16][17][18][19][20][21][22][23][24][25][26][27][28][29][30][31][32][33][34][35][36][37][38][39][40][41]

Table 9-439. L4_TA_AGENT_CONTROL_H

Address Offset	0x0000 0024		
Physical Address	See Table 9-396 to Table 9-428	**Instance**	See Table 9-396 to Table 9-428
Description	Enable clock power management		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						AUTO_WAKEUP_RESP_CODE	EXT_CLOCK	RESERVED							

Bits	Field Name	Description	Type	Reset
31:10	RESERVED	Read returns 0.	R	0x000000
9	AUTO_WAKEUP_RESP_CODE		R	0
8	EXT_CLOCK	When set to 1, the ext_clk_off_i signal on a target agent indicates when the target agent should shut off.	R	0
7:0	RESERVED	Read returns 0.	R	0x00

Table 9-440. Register Call Summary for Register L4_TA_AGENT_CONTROL_H

L4 Interconnects

- L4 Target Agent (L4 TA) Register Summary: [0][1][2][3][4][5][6][7][8][9][10][11][12][13][14][15][16][17][18][19][20][21][22][23][24][25][26][27][28][29][30]

Table 9-441. L4_TA_AGENT_STATUS_L

Address Offset	0x0000 0028		
Physical Address	See Table 9-396 to Table 9-428	**Instance**	See Table 9-396 to Table 9-428
Description	Error reporting		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED							SERROR	RESERVED															REQ_TIMEOUT	RESERVED							OCP_RESET

Bits	Field Name	Description	Type	Reset
31:25	RESERVED	Read returns 0.	R	0x00
24	SERROR	Value of OCP SError signal	R	0
23:9	RESERVED	Read returns 0.	R	0x0000
8	REQ_TIMEOUT	Time-out status: 0x0: No request time-out 0x1: A request time-out has occurred	R 1toCLR	0
7:1	RESERVED	Read returns 0.	R	0x00

Bits	Field Name	Description	Type	Reset
0	OCP_RESET	L3 Reset	R	0

Table 9-442. Register Call Summary for Register L4_TA_AGENT_STATUS_L

L4 Interconnects
- Time-Out: [0][1]
- Error Recovery: [2]
- Operational Modes Configuration: [3]
- L4 Target Agent (L4 TA) Register Summary: [4][5][6][7][8][9][10][11][12][13][14][15][16][17][18][19][20][21][22][23][24][25][26][27][28][29][30][31][32][33][34]

Table 9-443. L4_TA_AGENT_STATUS_H

Address Offset	0x0000 002C		
Physical Address	See Table 9-396 to Table 9-428	**Instance**	See Table 9-396 to Table 9-428
Description	Error reporting		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															

Bits	Field Name	Description	Type	Reset
31:0	RESERVED	Read returns 0	R	0x0000 000

Table 9-444. Register Call Summary for Register L4_TA_AGENT_STATUS_H

L4 Interconnects
- L4 Target Agent (L4 TA) Register Summary: [0][1][2][3][4][5][6][7][8][9][10][11][12][13][14][15][16][17][18][19][20][21][22][23][24][25][26][27][28][29][30]

9.3.5.4 L4 Link Agent (L4 LA)

9.3.5.4.1 L4 Link Agent (L4 LA) Register Summary

Table 9-445 summarizes the L4 LA register mapping.

Table 9-445. LA Register Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	L4_PER1_LA L3_MAIN Physical Address	L4_PER2_LA L3_MAIN Physical Address	L4_PER3_LA L3_MAIN Physical Address
L4_LA_COMPONENT_L	R	32	0x0000 0000	0x4800 0800	0x4840 0800	0x4880 0800
L4_LA_COMPONENT_H	R	32	0x0000 0004	0x4800 0804	0x4840 0804	0x4880 0804
L4_LA_NETWORK_L	R	32	0x0000 0010	0x4800 0810	0x4840 0810	0x4880 0810
L4_LA_NETWORK_H	R	32	0x0000 0014	0x4800 0814	0x4840 0814	0x4880 0814
L4_LA_INITIATOR_INFO_L	R	32	0x0000 0018	0x4800 0818	0x4840 0818	0x4880 0818
L4_LA_INITIATOR_INFO_H	R	32	0x0000 001C	0x4800 081C	0x4840 081C	0x4880 081C
L4_LA_NETWORK_CONTROL_L	RW	32	0x0000 0020	0x4800 0820	0x4840 0820	0x4880 0820
L4_LA_NETWORK_CONTROL_H	RW	32	0x0000 0024	0x4800 0824	0x4840 0824	0x4880 0824

Table 9-445. LA Register Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	L4_PER1_LA L3_MAIN Physical Address	L4_PER2_LA L3_MAIN Physical Address	L4_PER3_LA L3_MAIN Physical Address
L4_LA_FLAG_MASK_j_L[1]	RW	32	0x0000 0100 + (0x20*j)	0x4800 0900 + (0x20*j)	0x4840 0900 + (0x20*j)	0x4880 0900 + (0x20*j)
L4_LA_FLAG_MASK_j_H[1]	RW	32	0x0000 0104 + (0x20*j)	0x4800 0904 + (0x20*j)	0x4840 0904 + (0x20*j)	0x4880 0904 + (0x20*j)
L4_LA_FLAG_STATUS_j_L[1]	R	32	0x0000 0110 + (0x20*j)	0x4800 0910 + (0x20*j)	0x4840 0910 + (0x20*j)	0x4880 0910 + (0x20*j)
L4_LA_FLAG_STATUS_j_H[1]	R	32	0x0000 0114 + (0x20*j)	0x4800 0914 + (0x20*j)	0x4840 0914 + (0x20*j)	0x4880 0914 + (0x20*j)

(1) j = 0 to 1 for L4_PER1_LA
j = 0 to 1 for L4_PER2_LA
j = 0 to 1 for L4_PER3_LA

Table 9-446. LA Register Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	L4_CFG_LA L3_MAIN Physical Address	L4_WKUP_LA L3_MAIN Physical Address
L4_LA_COMPONENT_L	R	32	0x0000 0000	0x4A00 0800	0x4AE0 0800
L4_LA_COMPONENT_H	R	32	0x0000 0004	0x4A00 0804	0x4AE0 0804
L4_LA_NETWORK_L	R	32	0x0000 0010	0x4A00 0810	0x4AE0 0810
L4_LA_NETWORK_H	R	32	0x0000 0014	0x4A00 0814	0x4AE0 0814
L4_LA_INITIATOR_INFO_L	R	32	0x0000 0018	0x4A00 0818	0x4AE0 0818
L4_LA_INITIATOR_INFO_H	R	32	0x0000 001C	0x4A00 081C	0x4AE0 081C
L4_LA_NETWORK_CONTROL_L	RW	32	0x0000 0020	0x4A00 0820	0x4AE0 0820
L4_LA_NETWORK_CONTROL_H	RW	32	0x0000 0024	0x4A00 0824	0x4AE0 0824

9.3.5.4.2 L4 Link Agent (L4 LA) Register Description

Table 9-447 through Table 9-472 describe the L4 LA registers.

Table 9-447. L4_LA_COMPONENT_L

Address Offset	0x0000 0000		
Physical Address	0x4800 0800 0x4840 0800 0x4880 0800 0x4A00 0800 0x4AE0 0800	**Instance**	L4_PER1_LA L4_PER2_LA L4_PER3_LA L4_CFG_LA L4_WKUP_LA
Description	Contain a component code and revision, which are used to identify the hardware of the component.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CODE																REV															

Bits	Field Name	Description	Type	Reset
31:16	CODE	Interconnect code.	R	See (1).
15:0	REV	Component revision code.	R	See (1).

(1) TI Internal Data

Table 9-448. Register Call Summary for Register L4_LA_COMPONENT_L

L4 Interconnects

- L4 Link Agent (L4 LA) Register Summary: [0][1]

Table 9-449. L4_LA_COMPONENT_H

Address Offset	0x0000 0004		
Physical Address	0x4800 0804 0x4840 0804 0x4880 0804 0x4A00 0804 0x4AE0 0804	**Instance**	L4_PER1_LA L4_PER2_LA L4_PER3_LA L4_CFG_LA L4_WKUP_LA
Description	Contain a component code and revision, which are used to identify the hardware of the component.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															

Bits	Field Name	Description	Type	Reset
31:0	RESERVED	Read returns 0.	R	0x0000 0000

Table 9-450. Register Call Summary for Register L4_LA_COMPONENT_H

L4 Interconnects

- L4 Link Agent (L4 LA) Register Summary: [0][1]

Table 9-451. L4_LA_NETWORK_L

Address Offset	0x0000 0010		
Physical Address	0x4800 0810 0x4840 0810 0x4880 0810 0x4A00 0810 0x4AE0 0810	**Instance**	L4_PER1_LA L4_PER2_LA L4_PER3_LA L4_CFG_LA L4_WKUP_LA
Description	Identify the interconnect		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															

Bits	Field Name	Description	Type	Reset
31:0	RESERVED	Read returns 0	R	0x0000 0000

Table 9-452. Register Call Summary for Register L4_LA_NETWORK_L

L4 Interconnects

- L4 Link Agent (L4 LA) Register Summary: [0][1]

Table 9-453. L4_LA_NETWORK_H

Address Offset	0x0000 0014		
Physical Address	0x4800 0814 0x4840 0814 0x4880 0814 0x4A00 0814 0x4AE0 0814	**Instance**	L4_PER1_LA L4_PER2_LA L4_PER3_LA L4_CFG_LA L4_WKUP_LA
Description	Identify the interconnect		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID																															

Bits	Field Name	Description	Type	Reset
31:0	ID	The ID field uniquely identifies this interconnect.	R	0x00000000

Table 9-454. Register Call Summary for Register L4_LA_NETWORK_H

L4 Interconnects

- L4 Link Agent (L4 LA) Register Summary: [0][1]

Table 9-455. L4_LA_INITIATOR_INFO_L

Address Offset	0x0000 0018		
Physical Address	0x4800 0818 0x4840 0818 0x4880 0818 0x4A00 0818 0x4AE0 0818	**Instance**	L4_PER1_LA L4_PER2_LA L4_PER3_LA L4_CFG_LA L4_WKUP_LA
Description	Contain initiator subsystem information.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				PROT_GROUPS				NUMBER_REGIONS								RESERVED												SEGMENTS			

Bits	Field Name	Description	Type	Reset
31:28	RESERVED	Read returns 0.	R	0x0
27:24	PROT_GROUPS	Number of protection groups in the current L4 0x0: No protection group 0x1: 1 protection group 0x2: 2 protection groups .... 0x8: 8 protection groups 0x9 to 0xF: Reserved	R	see Table 9-457
23:16	NUMBER_REGIONS	Number of regions in the current L4 0x0: Reserved 0x1: 1 region 0x2: 2 regions .... Max regions +1 to 0xFF: Reserved, "Max regions" is listed in Table 9-457 on row "NUMBER_REGIONS".	R	see Table 9-457
15:4	RESERVED	Read returns 0.	R	0x000

Bits	Field Name	Description	Type	Reset
3:0	SEGMENTS	Number of segments in the current L4 0x0: Reserved 0x1: 1 segment 0x2: 2 segments .... 0x8: 8 segments	R	see Table 9-457

Table 9-456. Register Call Summary for Register L4_LA_INITIATOR_INFO_L

L4 Interconnects

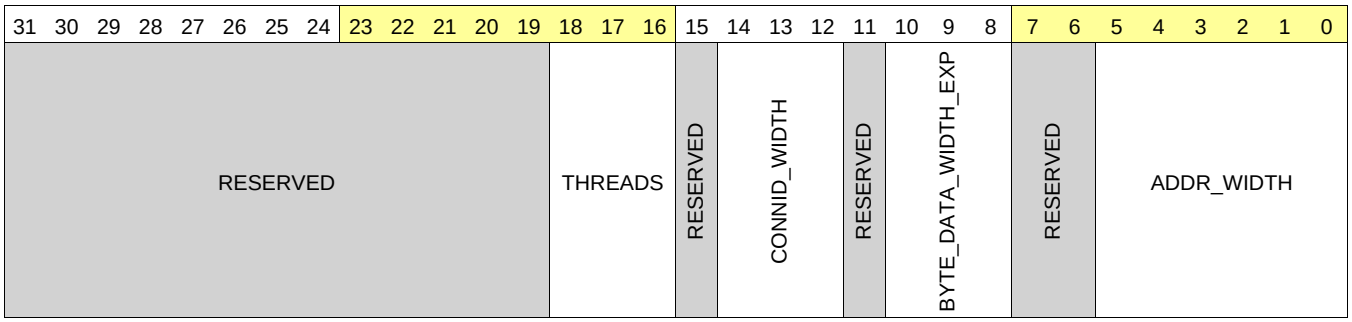
- L4 Link Agent (L4 LA) Register Summary: [0][1]

Table 9-457. Reset value for L4_LA_INITIATOR_INFO_L

Field Name	L4_PER1	L4_PER2	L4_PER3	L4_CFG	L4_WKUP
PROT_GROUPS	0x8	0x8	0x8	0x8	0x8
NUMBER_REGIONS	0x55	0x3F	0x61	0x6F	0x2C
SEGMENTS	0x2	0x1	0x1	0x3	0x4

Table 9-458. L4_LA_INITIATOR_INFO_H

Address Offset	0x0000 001C		
Physical Address	0x4800 081C 0x4840 081C 0x4880 081C 0x4A00 081C 0x4AE0 081C	**Instance**	L4_PER1_LA L4_PER2_LA L4_PER3_LA L4_CFG_LA L4_WKUP_LA
Description	Contain initiator subsystem information.		
Type	R		



Bits	Field Name	Description	Type	Reset
31:19	RESERVED	Read returns 0.	R	0x0000
18:16	THREADS	The THREADS field specifies the number of initiator threads connected to the interconnect. The field contains read-only configuration information for the initiator subsystem.	R	see Table 9-460
15	RESERVED	Read returns 0.	R	0
14:12	CONNID_WIDTH	The initiator subsystem ConnID width. The CONNID_WIDTH field contains read-only configuration information for the initiator subsystem.	R	see Table 9-460
11	RESERVED	Read returns 0.	R	0

Bits	Field Name	Description	Type	Reset
10:8	BYTE_DATA_WIDTH_EXP	This field specifies the initiator subsystem data width.The BYTE_DATA_WIDTH_EXP field contains read-only configuration information for the initiator subsystem. 0x1: 16-bit data width is specified 0x2: 32-bit data width is specified	R	see Table 9-460
7:6	RESERVED	Read returns 0.	R	0x0
5:0	ADDR_WIDTH	This field specifies the initiator subsystem address width. The ADDR_WIDTH field contains read-only configuration information for the initiator subsystem.	R	see Table 9-460

Table 9-459. Register Call Summary for Register L4_LA_INITIATOR_INFO_H

L4 Interconnects
- L4 Link Agent (L4 LA) Register Summary: [0][1]

Table 9-460. Reset value for L4_LA_INITIATOR_INFO_H

Field Name	L4_PER1	L4_PER2	L4_PER3	L4_CFG	L4_WKUP
THREADS	0x4	0x3	0x3	0x1	0x1
CONNID_WIDTH	0x4	0x4	0x5	0x4	0x4
BYTE_DATA_WIDTH_EXP	0x2	0x2	0x2	0x2	0x2
ADDR_WIDTH	0x18	0x18	0x18	0x18	0x15

Table 9-461. L4_LA_NETWORK_CONTROL_L

Address Offset	0x0000 0020		
Physical Address	0x4800 0820 0x4840 0820 0x4880 0820 0x4A00 0820 0x4AE0 0820	**Instance**	L4_PER1_LA L4_PER2_LA L4_PER3_LA L4_CFG_LA L4_WKUP_LA
Description	Control interconnect minimum timeout values.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																					TIMEOUT_BASE			RESERVED							

Bits	Field Name	Description	Type	Reset
31:11	RESERVED	Read returns 0.	R	0x000000
10:8	TIMEOUT_BASE	The TIMEOUT_BASE field indicates the time-out period (that is, base cycles) for the highest frequency time-base signal sent from the L4 initiator subsystem to all target agents that have time-out enabled. Values for the field are: 0 - Time-out disabled 1 - L4 interconnect clock cycles divided by 64 2 - L4 interconnect clock cycles divided by 256 3 - L4 interconnect clock cycles divided by 1024 4 - L4 interconnect clock cycles divided by 4096	RW	0x4
7:0	RESERVED	Read returns 0.	R	0x00

Table 9-462. Register Call Summary for Register L4_LA_NETWORK_CONTROL_L

L4 Interconnects

- Time-Out: [0][1][2]
- Operational Modes Configuration: [3][4]
- L4 Link Agent (L4 LA) Register Summary: [5][6]

Table 9-463. L4_LA_NETWORK_CONTROL_H

Address Offset	0x0000 0024		
Physical Address	0x4800 0824 0x4840 0824 0x4880 0824 0x4A00 0824 0x4AE0 0824	**Instance**	L4_PER1_LA L4_PER2_LA L4_PER3_LA L4_CFG_LA L4_WKUP_LA
Description	Control interconnect global power control		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED							CLOCK_GATE_DISABLE	RESERVED			THREAD0_PRI	RESERVED											EXT_CLOCK	RESERVED							

Bits	Field Name	Description	Type	Reset
31:25	RESERVED	Read returns 0.	R	0x00
24	CLOCK_GATE_DISABLE	When set to 1 this field disables all clock gating.	RW	0
23:21	RESERVED	Read returns 0.	R	0x0
20	THREAD0_PRI	Sets thread priority. If the field is set to 0, the default, all initiator threads are treated the same. Setting the THREAD0_PRI field to 1 assigns a higher arbitration priority to thread 0 of the first initiator OCP interface. To avoid starvation, arbitration is imposed by the initiator subsystem. When multiple requests from different initiator threads are dispatched to targets simultaneously, the oldest request is dispatched first. If thread 0 is assigned a higher priority, a request on thread 0 always wins arbitration. Assigning thread 0 of the first initiator OCP the highest priority on a request or response can result in the starvation of other threads.	R	1
19:9	RESERVED	Read returns 0.	R	0x000
8	EXT_CLOCK	Global external clock control. When set to 1, the ext_clk_off_i signal on the initiator subsystem instructs the entire L4 to shut off.	R	1
7:0	RESERVED	Read returns 0.	R	0x00

Table 9-464. Register Call Summary for Register L4_LA_NETWORK_CONTROL_H

L4 Interconnects

- L4 Link Agent (L4 LA) Register Summary: [0][1]

Table 9-465. L4_LA_FLAG_MASK_j_L

Address Offset	0x0000 0100 + (0x20*j)		
Physical Address	0x4800 0900 + (0x20*j) 0x4840 0900 + (0x20*j) 0x4880 0900 + (0x20*j)	**Instance**	L4_PER1_LA L4_PER2_LA L4_PER3_LA
Description	Mask of composite sideband flag(0)		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												MASK			

Bits	Field Name	Description	Type	Reset
31:4	RESERVED	Read returns 0	R	0x0000 0000
3:0	MASK	Number of input sideband signals	RW	0xF

Table 9-466. Register Call Summary for Register L4_LA_FLAG_MASK_j_L

L4 Interconnects

- L4 Link Agent (L4 LA) Register Summary: [0]

Table 9-467. Reset Value for L4_LA_FLAG_MASK_j_L

Initiator	Bit Field (MASK)	Reset
L4_PER1	[3:0]	0xF
L4_PER2	[2:0]	0x7
L4_PER3	[2:0]	0x7

Table 9-468. L4_LA_FLAG_MASK_j_H

Address Offset	0x0000 0104 + (0x20*j)		
Physical Address	0x4800 0904 + (0x20*j) 0x4840 0904 + (0x20*j) 0x4880 0904 + (0x20*j)	**Instance**	L4_PER1_LA L4_PER2_LA L4_PER3_LA
Description	Status of composite sideband flag(0)		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															

Bits	Field Name	Description	Type	Reset
31:0	RESERVED	Read returns 0	R	0x0000 0000

Table 9-469. Register Call Summary for Register L4_LA_FLAG_MASK_j_H

L4 Interconnects

- L4 Link Agent (L4 LA) Register Summary: [0]

Table 9-470. L4_LA_FLAG_STATUS_j_L

Address Offset	0x0000 0110 + (0x20*j)		
Physical Address	0x4800 0910 + (0x20*j) 0x4840 0910 + (0x20*j) 0x4880 0910 + (0x20*j)	**Instance**	L4_PER1_LA L4_PER2_LA L4_PER3_LA

Table 9-470. L4_LA_FLAG_STATUS_j_L (continued)

Description	Mask of composite sideband flag(1)
Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												STATUS			

Bits	Field Name	Description	Type	Reset
31:4	RESERVED	Read returns 0	R	0x0000 0000
3:0	STATUS	Status of input sideband signals	RW	0x0

Table 9-471. Register Call Summary for Register L4_LA_FLAG_STATUS_j_L

L4 Interconnects
- L4 Link Agent (L4 LA) Register Summary: [0]

Table 9-472. L4_LA_FLAG_STATUS_j_H

Address Offset	0x0000 0114 + (0x20*j)		
Physical Address	0x4800 0914 + (0x20*j) 0x4840 0914 + (0x20*j) 0x4880 0914 + (0x20*j)	**Instance**	L4_PER1_LA L4_PER2_LA L4_PER3_LA
Description	Status of composite sideband flag(1)		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															

Bits	Field Name	Description	Type	Reset
31:0	RESERVED	Read returns 0	R	0x000 0000 0000 0000

Table 9-473. Register Call Summary for Register L4_LA_FLAG_STATUS_j_H

L4 Interconnects
- L4 Link Agent (L4 LA) Register Summary: [0]

9.3.5.5 L4 Address Protection (L4 AP)

9.3.5.5.1 L4 Address Protection (L4 AP) Register Summary

Table 9-474 and Table 9-475 summarizes the L4 AP register mapping.

Table 9-474. L4 AP Register Summary

Register Name	Type	Register Width (Bits)	Address Offset	L4_PER1_AP Physical Address	L4_PER2_AP Physical Address	L4_PER3_AP Physical Address
L4_AP_COMPONENT_L	R	32	0x0000 0000	0x4800 0000	0x4840 0000	0x4880 0000
L4_AP_COMPONENT_H	R	32	0x0000 0004	0x4800 0004	0x4840 0004	0x4880 0004

Table 9-474. L4 AP Register Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	L4_PER1_AP Physical Address	L4_PER2_AP Physical Address	L4_PER3_AP Physical Address
L4_AP_SEGMENT_i_L (1)	R	32	0x0000 0100 + (0x08*i)	0x4800 0100 + (0x08*i)	0x4840 0100 + (0x08*i)	0x4880 0100 + (0x08*i)
L4_AP_SEGMENT_i_H (1)	R	32	0x0000 0104 + (0x08*i)	0x4800 0104 + (0x08*i)	0x4840 0104 + (0x08*i)	0x4880 0104 + (0x08*i)
L4_AP_PROT_GROUP_MEMBERS_k_L (2)	RW	32	0x0000 0200 + (0x08*k)	0x4800 0200 + (0x08*k)	0x4840 0200 + (0x08*k)	0x4880 0200 + (0x08*k)
L4_AP_PROT_GROUP_ROLES_k_L (2)	RW	32	0x0000 0280 + (0x08*k)	0x4800 0280 + (0x08*k)	0x4840 0280 + (0x08*k)	0x4880 0280 + (0x08*k)
L4_AP_PROT_GROUP_ROLES_k_H (2)	R	32	0x0000 0284 + (0x08*k)	0x4800 0284 + (0x08*k)	0x4840 0284 + (0x08*k)	0x4880 0284 + (0x08*k)
L4_AP_REGION_l_L (3)	R	32	0x0000 0300 + (0x08*l)	0x4800 0300 + (0x08*l)	0x4840 0300 + (0x08*l)	0x4880 0300 + (0x08*l)
L4_AP_REGION_l_H (3)	RW	32	0x0000 0304 + (0x08*l)	0x4800 0304 + (0x08*l)	0x4840 0304 + (0x08*l)	0x4880 0304 + (0x08*l)

(1) i = 0 to 1 for L4_PER1_AP
i = 0 for L4_PER2_AP
i = 0 for L4_PER3_AP
(2) k = 0 to 7 for L4_PER1_AP
k = 0 to 7 for L4_PER2_AP
k = 0 to 7 for L4_PER3_AP
(3) l = 0 to 77 for L4_PER1_AP
l = 0 to 32 for L4_PER2_AP
l = 0 to 92 for L4_PER3_AP

Table 9-475. L4 AP Register Summary

Register Name	Type	Register Width (Bits)	Address Offset	L4_CFG_AP Physical Address	L4_WKUP_AP Physical Address
L4_AP_COMPONENT_L	R	32	0x0000 0000	0x4A00 0000	0x4AE0 0000
L4_AP_COMPONENT_H	R	32	0x0000 0004	0x4A00 0004	0x4AE0 0004
L4_AP_SEGMENT_i_L (1)	R	32	0x0000 0100 + (0x08*i)	0x4A00 0100 + (0x08*i)	0x4AE0 0100 + (0x08*i)
L4_AP_SEGMENT_i_H (1)	R	32	0x0000 0104 + (0x08*i)	0x4A00 0104 + (0x08*i)	0x4AE0 0104 + (0x08*i)
L4_AP_PROT_GROUP_MEMBERS_k_L (2)	RW	32	0x0000 0200 + (0x08*k)	0x4A00 0200 + (0x08*k)	0x4AE0 0200 + (0x08*k)
L4_AP_PROT_GROUP_ROLES_k_L (2)	RW	32	0x0000 0280 + (0x08*k)	0x4A00 0280 + (0x08*k)	0x4AE0 0280 + (0x08*k)
L4_AP_PROT_GROUP_ROLES_k_H (2)	R	32	0x0000 0284 + (0x08*k)	0x4A00 0284 + (0x08*k)	0x4AE0 0284 + (0x08*k)
L4_AP_REGION_l_L (3)	R	32	0x0000 0300 + (0x08*l)	0x4A00 0300 + (0x08*l)	0x4AE0 0300 + (0x08*l)
L4_AP_REGION_l_H (3)	RW	32	0x0000 0304 + (0x08*l)	0x4A00 0304 + (0x08*l)	0x4AE0 0304 + (0x08*l)

(1) i = 0 to 1 for L4_PER1_AP
i = 0 for L4_PER2_AP
i = 0 for L4_PER3_AP
(2) k = 0 to 7 for L4_PER1_AP
k = 0 to 7 for L4_PER2_AP
k = 0 to 7 for L4_PER3_AP
(3) l = 0 to 77 for L4_PER1_AP
l = 0 to 32 for L4_PER2_AP
l = 0 to 92 for L4_PER3_AP

9.3.5.5.2 L4 Address Protection (L4 AP) Register Description

Table 9-476 through Table 9-498 describe the L4 AP registers.

Table 9-476. L4_AP_COMPONENT_L

Address Offset	0x000		
Physical Address	0x4800 0000 0x4840 0000 0x4880 0000 0x4A00 0000 0x4AE0 0000	**Instance**	L4_PER1_AP L4_PER2_AP L4_PER3_AP L4_CFG_AP L4_WKUP_AP
Description	Contains a component code and revision, which are used to identify the hardware of the component.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CODE																REV															

Bits	Field Name	Description	Type	Reset
31:16	CODE	Interconnect code	R	See (1).
15:0	REV	Component revision code	R	See (1).

(1) TI Internal Data

Table 9-477. Register Call Summary for Register L4_AP_COMPONENT_L

L4 Interconnects
- L4 Address Protection (L4 AP) Register Summary: [0][1]

Table 9-478. L4_AP_COMPONENT_H

Address Offset	0x004		
Physical Address	0x4800 0004 0x4840 0004 0x4880 0004 0x4A00 0004 0x4AE0 0004	**Instance**	L4_PER1_AP L4_PER2_AP L4_PER3_AP L4_CFG_AP L4_WKUP_AP
Description	Contains a component code and revision, which are used to identify the hardware of the component.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															

Bits	Field Name	Description	Type	Reset
31:0	RESERVED	Read returns 0	R	0x0000 0000

Table 9-479. Register Call Summary for Register L4_AP_COMPONENT_H

L4 Interconnects
- L4 Address Protection (L4 AP) Register Summary: [0][1]

Table 9-480. L4_AP_SEGMENT_i_L

Address Offset	0x100 + (0x08*i)	**Index**	
Physical Address	0x4800 0100 + (0x08*i) 0x4840 0100 + (0x08*i) 0x4880 0100 + (0x08*i) 0x4A00 0100 + (0x08*i) 0x4AE0 0100 + (0x08*i)	**Instance**	L4_PER1_AP L4_PER2_AP L4_PER3_AP L4_CFG_AP L4_WKUP_AP
Description	Defines the base address of each segment		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BASE																															

Bits	Field Name	Description	Type	Reset
31:0	BASE	The base address of the segment (with 0s from bit 0 to bit SIZE-1).	R	see Table 9-484

Table 9-481. Register Call Summary for Register L4_AP_SEGMENT_i_L

L4 Interconnects

- L4 Firewall Address and Protection Register Settings: [0]
- L4 Address Protection (L4 AP) Register Summary: [1][2]

Table 9-482. L4_AP_SEGMENT_i_H

Address Offset	0x104 + (0x08*i)	**Index**	
Physical Address	0x4800 0104 + (0x08*i) 0x4840 0104 + (0x08*i) 0x4880 0104 + (0x08*i) 0x4A00 0104 + (0x08*i) 0x4AE0 0104 + (0x08*i)	**Instance**	L4_PER1_AP L4_PER2_AP L4_PER3_AP L4_CFG_AP L4_WKUP_AP
Description	Defines the size of each segment		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												SIZE			

Bits	Field Name	Description	Type	Reset
31:6	RESERVED	Read returns 0.	R	0x0000000
5:0	SIZE	Segment size is a power of 2, where 2^{SIZE} is the byte size of a segment (all segment registers use the same size).	R	see Table 9-484

Table 9-483. Register Call Summary for Register L4_AP_SEGMENT_i_H

L4 Interconnects

- L4 Firewall Address and Protection Register Settings: [0]
- L4 Address Protection (L4 AP) Register Summary: [1][2]

Table 9-484. Reset Value for L4_AP_SEGMENT_i

i	L4_PER1		L4_PER2		L4_PER3		L4_CFG		L4_WKUP	
	BASE	SIZE	BASE	SIZE	BASE	SIZE	BASE	SIZE	BASE	SIZE
0	0x0000 0000	0x15	0x0000 0000	0x16	0x0000 0000	0x15	0x0000 0000	0x14	0x0000 0000	0x10
1	0x0020 0000	0x15	-	-	-	-	0x0010 0000	0x14	0x0001 0000	0x10

Table 9-484. Reset Value for L4_AP_SEGMENT_i (continued)

i	L4_PER1		L4_PER2		L4_PER3		L4_CFG		L4_WKUP	
	BASE	SIZE	BASE	SIZE	BASE	SIZE	BASE	SIZE	BASE	SIZE
2	-	-	-	-	-	-	0x0020 0000	0x14	0x0002 0000	0x10
3	-	-	-	-	-	-	-	-	0x0003 0000	0x10

Table 9-485. L4_AP_PROT_GROUP_MEMBERS_k_L

Address Offset	0x200 + (0x08*k)	**Index**	
Physical Address	0x4800 0200 + (0x08*k) 0x4840 0200 + (0x08*k) 0x4880 0200 + (0x08*k) 0x4A00 0200 + (0x08*k) 0x4AE0 0200 + (0x08*k)	**Instance**	L4_PER1_AP L4_PER2_AP L4_PER3_AP L4_CFG_AP L4_WKUP_AP
Description	Defines which initiator can access the protection group *k*.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																CONNID_BIT_VECTOR															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Reserved	R	0x000000000000
15:0	CONNID_BIT_VECTOR	The protection group *k* is accessible by an initiator with ConnID *n* when bit *n* of this field is set to 1.	RW[(1)]	0xFFFF

(1) For L4_PER1 when k = 0 and 1 the access type of CONNID_BIT_VECTOR is R.
For L4_PER2 when k = 0 the access type of CONNID_BIT_VECTOR is R.
For L4_PER3 when k = 0 the access type of CONNID_BIT_VECTOR is R.
For L4_CFG when k = 0, 1 and 5 the access type of CONNID_BIT_VECTOR is R.
For L4_WKUP when k = 0, 1, 3 and 4 the access type of CONNID_BIT_VECTOR is R.

Table 9-486. Register Call Summary for Register L4_AP_PROT_GROUP_MEMBERS_k_L

L4 Interconnects
- Protection Group: [0][1]
- L4 Firewall Address and Protection Register Settings: [10]
- Operational Modes Configuration: [11]
- L4 Address Protection (L4 AP) Register Summary: [12][13]

Table 9-487. L4_AP_PROT_GROUP_ROLES_k_L

Address Offset	0x200 + (0x08*k)	**Index**	
Physical Address	0x4800 0280 + (0x08*k) 0x4840 0280 + (0x08*k) 0x4880 0280 + (0x08*k) 0x4A00 0280 + (0x08*k) 0x4AE0 0280 + (0x08*k)	**Instance**	L4_PER1_AP L4_PER2_AP L4_PER3_AP L4_CFG_AP L4_WKUP_AP
Description	Defines the initiator requests accepted or not by the L4 firewalls depending on combination of L4 qualifiers.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ENABLE																															

Bits	Field Name	Description	Type	Reset
31:0	ENABLE	Each bit of this field corresponds to an unique combination of L4 qualifiers. To accept a request with combination *n*, bit *n* must be set to 1. For the list of combinations see Table 9-347.	RW[1]	0xFFFFFFFF

[1] For L4_PER1 when k = 0 and 1 the access type is R.
For L4_PER2 when k = 0 the access type is R.
For L4_PER3 when k = 0 the access type is R.
For L4_CFG when k = 0, 1 and 5 the access type is R.
For L4_WKUP when k = 0, 1, 3 and 4 the access type is R.

Table 9-488. Register Call Summary for Register L4_AP_PROT_GROUP_ROLES_k_L

L4 Interconnects
- Protection Group: [0][1][2]
- L4 Firewall Address and Protection Register Settings: [13]
- Operational Modes Configuration: [14]
- L4 Address Protection (L4 AP) Register Summary: [15][16]

Table 9-489. L4_AP_PROT_GROUP_ROLES_k_H

Address Offset	0x204 + (0x08*k)	**Index**	
Physical Address	0x4800 0284 + (0x08*k) 0x4840 0284 + (0x08*k) 0x4880 0284 + (0x08*k) 0x4A00 0284 + (0x08*k) 0x4AE0 0284 + (0x08*k)	**Instance**	L4_PER1_AP L4_PER2_AP L4_PER3_AP L4_CFG_AP L4_WKUP_AP
Description	Reserved.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ENABLE																															

Bits	Field Name	Description	Type	Reset
31:0	ENABLE	Reserved.	R	0xFFFFFFFF

Table 9-490. Register Call Summary for Register L4_AP_PROT_GROUP_ROLES_k_H

L4 Interconnects
- L4 Firewall Address and Protection Register Settings:
- L4 Address Protection (L4 AP) Register Summary: [7][8]

Table 9-491. L4_AP_REGION_l_L

Address Offset	0x300 + (0x08*l)	**Index**	
Physical Address	0x4800 0300 + (0x08*l) 0x4840 0300 + (0x08*l) 0x4880 0300 + (0x08*l) 0x4A00 0300 + (0x08*l) 0x4AE0 0300 + (0x08*l)	**Instance**	L4_PER1_AP L4_PER2_AP L4_PER3_AP L4_CFG_AP L4_WKUP_AP
Description	Defines the base address of the region in respect to the segment it belongs to.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																BASE															

Bits	Field Name	Description	Type	Reset
31:21	RESERVED	Read returns 0.	R	0x00
20:0	BASE	Sets the base address of the region relative to its segment base.	R	See Table 9-495 to Table 9-499

Table 9-492. Register Call Summary for Register L4_AP_REGION_I_L

L4 Interconnects

- L4 Firewall Address and Protection Register Settings: [0]
- L4 Address Protection (L4 AP) Register Summary: [1][2]

Table 9-493. L4_AP_REGION_I_H

Address Offset	0x304 + (0x08*I)	**Index**	
Physical Address	0x4800 0304 + (0x08*I) 0x4840 0304 + (0x08*I) 0x4880 0304 + (0x08*I) 0x4A00 0304 + (0x08*I) 0x4AE0 0304 + (0x08*I)	**Instance**	L4_PER1_AP L4_PER2_AP L4_PER3_AP L4_CFG_AP L4_WKUP_AP
Description	Define the size, protection group and segment ID of the region		
Type	RW		

31 30 29 28	27	26 25 24	23	22 21 20	19	18 17	16	15 14 13 12 11 10 9 8	7	6 5 4 3 2 1	0
MADDRSPACE	RESERVED	SEGMENT_ID	RESERVED	PROT_GROUP_ID	RESERVED	BYTE_DATA_WIDTH_EXP	RESERVED	PHY_TARGET_ID	RESERVED	SIZE	ENABLE

Bits	Field Name	Description	Type	Reset
31:28	MADDRSPACE	Target interconnect MAddrSpace	R	See Table 9-495 to Table 9-499
27	RESERVED	Read returns 0	R	0x0
26:24	SEGMENT_ID	Segment ID of the region	R	See Table 9-495 to Table 9-499
23	RESERVED	Read returns 0	R	0x0
22:20	PROT_GROUP_ID	Protection group ID	RW	See Table 9-495 to Table 9-499
19	RESERVED	Read returns 0	R	0x0
18:17	BYTE_DATA_WIDTH_EXP	Target data byte width	R	See Table 9-495 to Table 9-499
16:15	RESERVED	Read returns 0	R	0x0
14:8	PHY_TARGET_ID	Physical target ID	R	See Table 9-495 to Table 9-499
7	RESERVED	Read returns 0.	R	0x0
6:1	SIZE	Define the size of the region in bytes. 2^{SIZE} equals the region.	R	See Table 9-495 to Table 9-499
0	ENABLE	0x0: Disable the region. No access allowed 0x1: Enable the region.	R	0x1

Table 9-494. Register Call Summary for Register L4_AP_REGION_l_H

L4 Interconnects

- Protection Group: [0]
- L4 Firewall Address and Protection Register Settings: [1]
- Operational Modes Configuration: [2]
- L4 Address Protection (L4 AP) Register Summary: [3][4]

Table 9-495. L4_AP_REGION_l Reset Values for L4_PER1

Region	MADDRSPACE	SEGMENT_ID	PROT_GROUP_ID	BYTE_DATA_WIDTH_EXP	PHY_TARGET_ID	SIZE	ENABLE	BASE
0	0x0	0x0	0x0	0x2	0x00	0x0B	0x1	0x00 0000
1	0x1	0x0	0x7	0x2	0x00	0x0A	0x1	0x00 1000
2	0x5	0x0	0x0	0x2	0x01	0x0B	0x1	0x00 0800
3	0x0	0x0	0x7	0x2	0x04	0x0C	0x1	0x02 0000
4	0x0	0x0	0x7	0x2	0x05	0x0C	0x1	0x02 1000
5	0x0	0x0	0x7	0x2	0x3E	0x0C	0x1	0x03 2000
6	0x0	0x0	0x7	0x2	0x3F	0x0C	0x1	0x03 3000
7	0x0	0x0	0x7	0x2	0x46	0x0C	0x1	0x03 4000
8	0x0	0x0	0x7	0x2	0x47	0x0C	0x1	0x03 5000
9	0x0	0x0	0x7	0x2	0x4E	0x0C	0x1	0x03 6000
10	0x0	0x0	0x7	0x2	0x4F	0x0C	0x1	0x03 7000
13	0x0	0x0	0x7	0x2	0x0E	0x0C	0x1	0x05 5000
14	0x0	0x0	0x7	0x2	0x0F	0x0C	0x1	0x05 6000
15	0x0	0x0	0x7	0x2	0x06	0x0C	0x1	0x05 7000
16	0x0	0x0	0x7	0x2	0x07	0x0C	0x1	0x05 8000
17	0x0	0x0	0x7	0x2	0x16	0x0C	0x1	0x05 9000
18	0x0	0x0	0x7	0x2	0x17	0x0C	0x1	0x05 A000
24	0x0	0x0	0x7	0x2	0x24	0x0C	0x1	0x06 A000
25	0x0	0x0	0x7	0x2	0x25	0x0C	0x1	0x06 B000
26	0x0	0x0	0x7	0x2	0x2C	0x0C	0x1	0x06 C000
27	0x0	0x0	0x7	0x2	0x2D	0x0C	0x1	0x06 D000
30	0x0	0x0	0x7	0x2	0x22	0x0C	0x1	0x07 0000
31	0x0	0x0	0x7	0x2	0x23	0x0C	0x1	0x07 1000
32	0x0	0x0	0x7	0x2	0x2A	0x0C	0x1	0x07 2000
33	0x0	0x0	0x7	0x2	0x2B	0x0C	0x1	0x07 3000
39	0x0	0x0	0x7	0x2	0x0A	0x0C	0x1	0x07 8000
40	0x0	0x0	0x7	0x2	0x0B	0x0C	0x1	0x07 9000
47	0x0	0x0	0x7	0x2	0x08	0x0C	0x1	0x09 8000
48	0x0	0x0	0x7	0x2	0x09	0x0C	0x1	0x09 9000
49	0x0	0x0	0x7	0x2	0x10	0x0C	0x1	0x09 A000
50	0x0	0x0	0x7	0x2	0x11	0x0C	0x1	0x09 B000
67	0x0	0x0	0x7	0x2	0x48	0x0C	0x1	0x0B 8000
68	0x0	0x0	0x7	0x2	0x49	0x0C	0x1	0x0B 9000
69	0x0	0x0	0x7	0x2	0x18	0x0C	0x1	0x0B A000
70	0x0	0x0	0x7	0x2	0x19	0x0C	0x1	0x0B B000
71	0x0	0x0	0x7	0x2	0x28	0x0C	0x1	0x0D 1000
72	0x0	0x0	0x7	0x2	0x29	0x0C	0x1	0x0D 2000
77	0x2	0x0	0x7	0x2	0x00	0x0A	0x1	0x00 1400

Table 9-496. L4_AP_REGION_I Reset Values for L4_PER2

Region	MADDRSPACE	SEGMENT_ID	PROT_GROUP_ID	BYTE_DATA_WIDTH_EXP	PHY_TARGET_ID	SIZE	ENABLE	BASE
0	0x0	0x0	0x0	0x2	0x00	0x0B	0x1	0x00 0000
1	0x1	0x0	0x7	0x2	0x00	0x0A	0x1	0x00 1000
2	0x5	0x0	0x0	0x2	0x01	0x0B	0x1	0x00 0800
3	0x0	0x0	0x7	0x2	0x0E	0x0C	0x1	0x08 4000
6	0x0	0x0	0x7	0x2	0x11	0x0C	0x1	0x08 8000
9	0x0	0x0	0x7	0x2	0x0E	0x0D	0x1	0x06 0000
10	0x0	0x0	0x7	0x2	0x0F	0x0C	0x1	0x06 2000
25	0x0	0x0	0x7	0x2	0x30	0x0C	0x1	0x03 E000
26	0x0	0x0	0x7	0x2	0x31	0x0C	0x1	0x03 F000
31	0x0	0x0	0x7	0x2	0x16	0x0D	0x1	0x08 0000
32	0x0	0x0	0x7	0x2	0x17	0x0C	0x1	0x08 2000

Table 9-497. L4_AP_REGION_I Reset Values for L4_PER3

Region	MADDRSPACE	SEGMENT_ID	PROT_GROUP_ID	BYTE_DATA_WIDTH_EXP	PHY_TARGET_ID	SIZE	ENABLE	BASE
0	0x0	0x0	0x0	0x2	0x00	0x0B	0x1	0x00 0000
1	0x5	0x0	0x0	0x2	0x01	0x0B	0x1	0x00 0800
3	0x2	0x0	0x7	0x2	0x00	0x0A	0x1	0x00 1400
4	0x3	0x0	0x7	0x2	0x0A	0x0C	0x1	0x00 1800
5	0x0	0x0	0x7	0x2	0x08	0x0C	0x1	0x02 0000
6	0x0	0x0	0x7	0x2	0x09	0x0C	0x1	0x02 1000
7	0x0	0x0	0x7	0x2	0x24	0x0C	0x1	0x02 2000
8	0x0	0x0	0x7	0x2	0x25	0x0C	0x1	0x02 3000
9	0x0	0x0	0x7	0x2	0x26	0x0C	0x1	0x02 4000
10	0x0	0x0	0x7	0x2	0x27	0x0C	0x1	0x02 5000
11	0x0	0x0	0x7	0x2	0x0C	0x0C	0x1	0x02 6000
12	0x0	0x0	0x7	0x2	0x0D	0x0C	0x1	0x02 7000
21	0x0	0x0	0x7	0x2	0x0A	0x10	0x1	0x17 0000
22	0x0	0x0	0x7	0x2	0x0B	0x0C	0x1	0x18 0000
31	0x0	0x0	0x7	0x2	0x32	0x0D	0x1	0x05 C000
32	0x0	0x0	0x7	0x2	0x33	0x0C	0x1	0x05 D000
33	0x0	0x0	0x7	0x2	0x3E	0x0C	0x1	0x03 A000
34	0x0	0x0	0x7	0x2	0x3F	0x0C	0x1	0x03 B000
47	0x0	0x0	0x7	0x2	0x46	0x0C	0x1	0x04 8000
48	0x0	0x0	0x7	0x2	0x37	0x0C	0x1	0x04 9000
49	0x0	0x0	0x7	0x2	0x38	0x0C	0x1	0x04 A000
50	0x0	0x0	0x7	0x2	0x39	0x0C	0x1	0x04 B000
51	0x0	0x0	0x7	0x2	0x44	0x0C	0x1	0x04 C000
52	0x0	0x0	0x7	0x2	0x45	0x0C	0x1	0x04 D000
61	0x0	0x0	0x7	0x2	0x22	0x0C	0x1	0x05 6000
62	0x0	0x0	0x7	0x2	0x23	0x0C	0x1	0x05 7000
63	0x0	0x0	0x7	0x2	0x2A	0x0C	0x1	0x05 8000
64	0x0	0x0	0x7	0x2	0x2B	0x0C	0x1	0x05 9000
65	0x0	0x0	0x7	0x2	0x5C	0x0C	0x1	0x05 A000
66	0x0	0x0	0x7	0x2	0x5D	0x0C	0x1	0x05 B000
81	0x0	0x0	0x7	0x2	0x20	0x0C	0x1	0x00 4000

Table 9-497. L4_AP_REGION_l Reset Values for L4_PER3 (continued)

Region	MADDRSPACE	SEGMENT_ID	PROT_GROUP_ID	BYTE_DATA_WIDTH_EXP	PHY_TARGET_ID	SIZE	ENABLE	BASE
82	0x0	0x0	0x7	0x2	0x21	0x0C	0x1	0x00 5000
91	0x0	0x0	0x7	0x2	0x1C	0x0C	0x1	0x01 C000
92	0x0	0x0	0x7	0x2	0x1D	0x0C	0x1	0x01 D000

Table 9-498. L4_AP_REGION_l Reset Values for L4_CFG

Region	MADDRSPACE	SEGMENT_ID	PROT_GROUP_ID	BYTE_DATA_WIDTH_EXP	PHY_TARGET_ID	SIZE	ENABLE	BASE
0	0x0	0x0	0x0	0x2	0x00	0x0B	0x1	0x0 0000
1	0x5	0x0	0x0	0x2	0x01	0x0B	0x1	0x0 0800
2	0x1	0x0	0x7	0x2	0x00	0x0C	0x1	0x0 1000
3	0x0	0x0	0x7	0x2	0x08	0x0D	0x1	0x0 2000
4	0x0	0x0	0x7	0x2	0x09	0x0C	0x1	0x0 4000
5	0x0	0x0	0x7	0x2	0x10	0x0C	0x1	0x0 5000
6	0x0	0x0	0x7	0x2	0x11	0x0C	0x1	0x0 6000
7	0x0	0x0	0x7	0x2	0x0E	0x0D	0x1	0x0 8000
8	0x0	0x0	0x7	0x2	0x0F	0x0C	0x1	0x0 A000
17	0x0	0x0	0x7	0x2	0x72	0x0C	0x1	0xD 9000
18	0x0	0x0	0x7	0x2	0x73	0x0C	0x1	0xD A000
19	0x0	0x0	0x7	0x2	0x18	0x0C	0x1	0xD D000
20	0x0	0x0	0x7	0x2	0x19	0x0C	0x1	0xD E000
21	0x0	0x0	0x5	0x2	0x28	0x0C	0x1	0xE 0000
22	0x0	0x0	0x7	0x2	0x29	0x0C	0x1	0xE 1000
23	0x0	0x0	0x7	0x2	0x04	0x0C	0x1	0xF 4000
24	0x0	0x0	0x7	0x2	0x05	0x0C	0x1	0xF 5000
25	0x0	0x0	0x7	0x2	0x78	0x0C	0x1	0xF 6000
26	0x0	0x0	0x7	0x2	0x79	0x0C	0x1	0xF 7000
27	0x0	0x1	0x7	0x2	0x3C	0x0C	0x1	0x0 2000
28	0x0	0x1	0x7	0x2	0x3D	0x0C	0x1	0x0 3000
29	0x0	0x1	0x7	0x2	0x1E	0x0C	0x1	0x0 8000
30	0x0	0x1	0x7	0x2	0x1F	0x0C	0x1	0x0 9000
31	0x0	0x1	0x7	0x2	0x06	0x10	0x1	0x4 0000
32	0x0	0x1	0x7	0x2	0x07	0x0C	0x1	0x5 0000
33	0x0	0x1	0x1	0x2	0x50	0x0C	0x1	0x5 1000
34	0x0	0x1	0x7	0x2	0x51	0x0C	0x1	0x5 2000
35	0x0	0x1	0x1	0x2	0x54	0x0C	0x1	0x5 3000
36	0x0	0x1	0x7	0x2	0x55	0x0C	0x1	0x5 4000
37	0x0	0x1	0x1	0x2	0x46	0x0C	0x1	0x5 5000
38	0x0	0x1	0x7	0x2	0x47	0x0C	0x1	0x5 6000
39	0x0	0x1	0x1	0x2	0x58	0x0C	0x1	0x5 7000
40	0x0	0x1	0x7	0x2	0x59	0x0C	0x1	0x5 8000
43	0x0	0x2	0x1	0x2	0x12	0x0C	0x1	0x1 8000
44	0x0	0x2	0x7	0x2	0x13	0x0C	0x1	0x1 9000
47	0x0	0x1	0x1	0x2	0x56	0x0C	0X1	0x5 F000
48	0x0	0x1	0x7	0x2	0x57	0x0C	0x1	0x6 0000
49	0x0	0x1	0x1	0x2	0x32	0x0C	0x1	0x6 1000
50	0x0	0x1	0x7	0x2	0x33	0x0C	0x1	0x6 2000

Table 9-498. L4_AP_REGION_I Reset Values for L4_CFG (continued)

Region	MADDRSPACE	SEGMENT_ID	PROT_GROUP_ID	BYTE_DATA_WIDTH_EXP	PHY_TARGET_ID	SIZE	ENABLE	BASE
53	0x0	0x1	0x1	0x2	0x4E	0x0C	0x01	0x6 5000
54	0x0	0x1	0x7	0x2	0x4F	0x0C	0x01	0x6 6000
55	0x0	0x1	0x1	0x2	0x5E	0x0C	0X01	0x6 7000
56	0x0	0x1	0x7	0x2	0x5F	0x0C	0x01	0x6 8000
57	0x0	0x1	0x1	0x2	0x68	0x0C	0x01	0x6 D000
58	0x0	0x1	0x7	0x2	0x69	0x0C	0x01	0x6 E000
59	0x0	0x0	0x7	0x2	0x42	0x0F	0x01	0x9 0000
60	0x0	0x0	0x7	0x2	0x43	0x0C	0x01	0x9 8000
71	0x0	0x1	0x7	0x2	0x2E	0x0C	0x01	0x8 3000
72	0x0	0x1	0x7	0x2	0x2F	0x0C	0x01	0x8 4000
89	0x0	0x2	0x1	0x2	0x1C	0x0C	0x01	0x1 4000
90	0x0	0x2	0x7	0x2	0x1D	0x0C	0x01	0x1 5000
105	0x0	0x1	0x1	0x2	0x34	0x0C	0x01	0x7 9000
106	0x0	0x1	0x7	0x2	0x35	0x0C	0x01	0x7 A000
107	0x0	0x1	0x1	0x2	0x52	0x0C	0x01	0x6 B000
108	0x0	0x1	0x7	0x2	0x53	0x0C	0x01	0x6 C000
109	0x0	0x2	0x1	0x2	0x6C	0x0C	0x01	0x2 C000
110	0x0	0x2	0x7	0x2	0x6D	0x0C	0x01	0x2 D000

Table 9-499. L4_AP_REGION_I Reset Values for L4_WKUP

Region	MADDRSPACE	SEGMENT_ID	PROT_GROUP_ID	BYTE_DATA_WIDTH_EXP	PHY_TARGET_ID	SIZE	ENABLE	BASE
0	0x0	0x0	0x0	0x2	0x0	0x0B	0x01	0x0000
1	0x1	0x0	0x7	0x2	0x0	0x0C	0x01	0x1000
2	0x5	0x0	0x0	0x2	0x01	0x0B	0x01	0x0800
3	0x0	0x0	0x7	0x2	0x10	0x0D	0x01	0x6000
4	0x0	0x0	0x7	0x2	0x11	0x0C	0x01	0x8000
5	0x0	0x1	0x7	0x2	0x20	0x0C	0x01	0x0000
6	0x0	0x1	0x7	0x2	0x21	0x0C	0x01	0x1000
9	0x0	0x1	0x7	0x2	0x30	0x0C	0x01	0x8000
10	0x0	0x1	0x7	0x2	0x31	0x0C	0x01	0x9000
15	0x0	0x0	0x7	0x2	0x40	0x0C	0x01	0x4000
16	0x0	0x0	0x7	0x2	0x41	0x0C	0x01	0x5000
17	0x0	0x0	0x7	0x2	0x50	0x0C	0x01	0xC000
18	0x0	0x0	0x7	0x2	0x51	0x0C	0x01	0xD000
30	0x0	0x3	0x7	0x2	0x04	0x0D	0x01	0xC000
31	0x0	0x3	0x7	0x2	0x05	0x0C	0x01	0xE000
34	0x0	0x3	0x7	0x2	0x60	0x0C	0x01	0x1000
35	0x0	0x3	0x7	0x2	0x61	0x0C	0x01	0x2000
36	0x0	0x3	0x7	0x2	0x0A	0x0C	0x01	0x3000
37	0x0	0x3	0x7	0x2	0x0B	0x0C	0x01	0x4000
38	0x0	0x3	0x7	0x2	0x0C	0x0C	0x01	0x5000
39	0x0	0x3	0x7	0x2	0x0D	0x0C	0x01	0x6000
40	0x0	0x3	0x7	0x2	0x68	0x0C	0x01	0x7000
41	0x0	0x3	0x7	0x2	0x69	0x0C	0x01	0x8000
42	0x0	0x3	0x7	0x2	0x70	0x0C	0x01	0x9000

Table 9-499. L4_AP_REGION_I Reset Values for L4_WKUP (continued)

Region	MADDRSPACE	SEGMENT_ID	PROT_GROUP_ID	BYTE_DATA_WIDTH_EXP	PHY_TARGET_ID	SIZE	ENABLE	BASE
43	0x0	0x3	0x7	0x2	0x71	0x0C	0x01	0xA000

Memory Subsystem

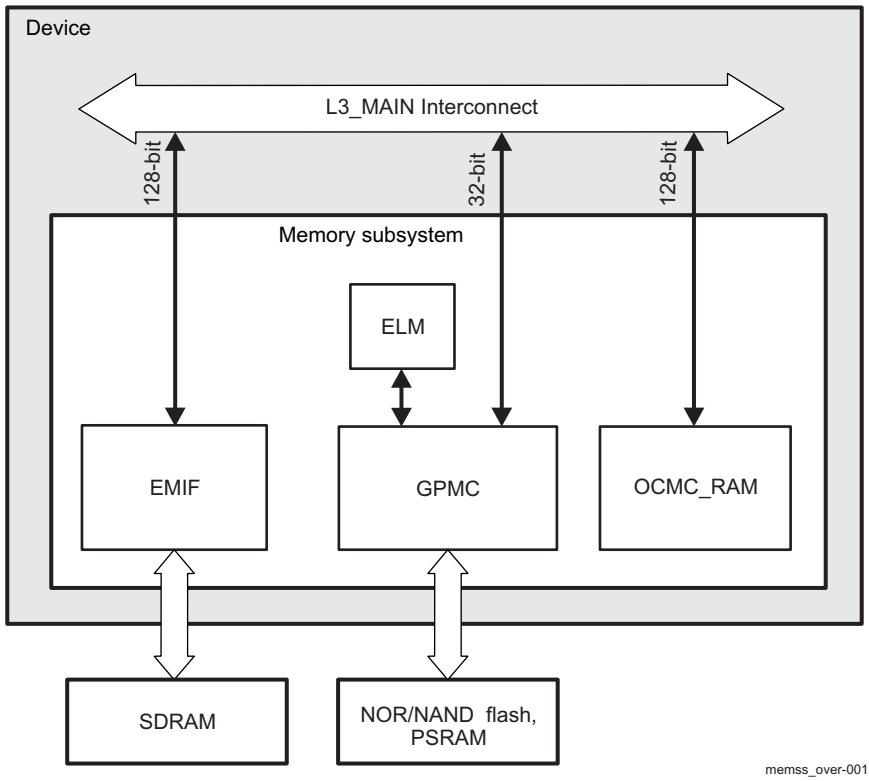
This chapter describes the Memory Subsystem for the device.

Topic **Page**

10.1 Memory Subsystem Overview

Figure 10-1 shows a functional diagram of all memory subsystems in the device.

Figure 10-1. Memory Subsystem Functional Diagram



10.1.1 EMIF Overview

NOTE: LPDDR2 is not supported on the DRA78x family of devices.

The EMIF module provides connectivity between DDR memory types and manages data bus read/write accesses between external memory and device subsystems which have master access to the L3_MAIN interconnect and DMA capability.

The EMIF module has the following capabilities:

- Supports JEDEC standard-compliant LPDDR2/DDR2-SDRAM and DDR3/DDR3L-SDRAM memory types
- 2-GiB SDRAM address range over one chip-select.
- Supports SDRAM devices with one, two, four or eight internal banks
- Data bus widths:
  - 128-bit L3_MAIN (system) interconnect data bus width
  - 32-bit SDRAM data bus width
  - 16-bit SDRAM data bus width used in narrow mode
- Supported CAS latencies:
  - DDR3: 5, 6, 7, 8, 9, 10 and 11
  - DDR2: 2, 3, 4, 5, 6 and 7
  - LPDDR2: 3, 4, 5, 6, 7, and 8
- Supports 256-, 512-, 1024-, and 2048-word page sizes

- Supported burst length: 8
- Supports sequential burst type
- SDRAM auto initialization from reset or configuration change
- Supports self refresh and power-down modes for low power
- Partial array self-refresh mode for low power when DDR3 is used
- Output impedance (ZQ) calibration for DDR3
- Supports on-die termination (ODT) for DDR2 and DDR3
- Supports prioritized refresh
- Programmable SDRAM refresh rate and backlog counter
- Programmable SDRAM timing parameters
- Write and read leveling/calibration and data eye training for DDR3.
- ECC on the SDRAM data bus:
  - 7-bit ECC over 32-bit data
  - 6-bit ECC over 16-bit data when narrow mode is used
  - 1-bit error correction and 2-bit error detection
  - Programmable address ranges to define ECC protected region
  - ECC calculated and stored on all writes to ECC protected address region
  - ECC verified on all reads from ECC protected address region
  - Statistics for 1-bit ECC and 2-bit ECC errors
  - The total width of the ECC DDR data bus is 8 bits

The EMIF module does not support:

- Burst chop for DDR3
- Interleave burst type
- Auto precharge because of better Bank Interleaving performance
- OCD calibration for DDR2
- CAS Read Latency of 2 and CAS Write Latency of 1 for DDR2
- DLL disabling from EMIF side

10.1.2 GPMC Overview

The General Purpose Memory Controller (GPMC) is an external memory controller of the device. Its data access engine provides a flexible programming model for communication with all standard memories.

The GPMC supports the following various access types:

- Asynchronous read/write access
- Asynchronous read page access (4, 8, and 16 Word16)
- Synchronous read/write access
- Synchronous read/write burst access without wrap capability (4, 8 and 16 Word16)
- Synchronous read/write burst access with wrap capability (4, 8 and 16 Word16)
- Address-data-multiplexed (AD) access
- Address-address-data (AAD) multiplexed access
- Little- and big-endian access

The GPMC can communicate with a wide range of external devices:

- External asynchronous or synchronous 8-bit wide memory or device (non burst device)
- External asynchronous or synchronous 16-bit wide memory or device
- External 16-bit non-multiplexed NOR flash device
- External 16-bit address and data multiplexed NOR Flash device

- External 8-bit and 16-bit NAND flash device
- External 16-bit pseudo-SRAM (pSRAM) device

The main features of the GPMC are:

- 8- or 16-bit-wide data path to external memory device
- Supports up to eight CS regions of programmable size and programmable base addresses in a total address space of 1 GiB
- Supports transactions controlled by a firewall
- On-the-fly error code detection using the Bose-ChaudhurI-Hocquenghem (BCH) (t = 4, 8, or 16) or Hamming code to improve the reliability of NAND with a minimum effect on software (NAND flash with 512-byte page size or greater)
- Fully pipelined operation for optimal memory bandwidth use
- The clock to the external memory is provided from GPMC functional clock divided by 1, 2, 3, or 4
- Supports programmable autoclock gating when no access is detected
- Independent and programmable control signal timing parameters for setup and hold time on a per-chip basis. Parameters are set according to the memory device timing parameters, with a timing granularity of one GPMC functional clock cycle.
- Flexible internal access time control (WAIT state) and flexible handshake mode using external WAIT pin monitoring
- Support bus keeping
- Support bus turnaround
- Prefetch and write posting engine associated with a device DMA controller to achieve full performance from the NAND device with minimum effect on NOR/SRAM concurrent access

NOTE: Page mode is available only in nonmultiplexed mode.

10.1.3 ELM Overview

In the case of NAND modules with no internal correction capability, sometimes referred to as bare NAND, the correction process can be delegated to the error location module (ELM) used in conjunction with the GPMC.

The ELM supports the following features:

- 4, 8, and 16 bits per 512-byte block error location based on BCH algorithm
- Eight simultaneous processing contexts
- Page-based and continuous modes
- Interrupt generation when error location process completes:
  - When the full page has been processed in page mode
  - For each syndrome polynomial (checksum-like information) in continuous mode

10.1.4 OCM Overview

There is one on-chip memory controller (OCMC) in the device.

The OCM Controller supports the following features:

- L3_MAIN data interface:
  - Used for maximum throughput performance
  - 128-bit data bus width
  - Burst supported
- L4 interface:
  - Used for access to configuration registers
  - 32-bit data bus width

  - Only single accesses supported
  - The L4 associated OCMC clock is two times lower than the L3 associated OCMC clock
- Error correction and detection:
  - Single error correction and dual error detection
  - 9-bit Hamming error correction code (ECC) calculated on 128-bit data word which is concatenated with memory address bits
  - Hamming distance of 4
  - Enable/Disable mode control through a dedicated register
  - Single bit error correction on a read transaction
  - Exclusion of repeated addresses from correctable error address trace history
  - ECC valid for all write transactions to an enabled region
  - Sub-128-bit writes supported via read modify write
- ECC Error Status Reporting:
  - Trace history buffer (FIFO) with depth of 4 for corrected error address
  - Trace history buffer with depth of 4 for non correctable error address and also including double error detection
  - Interrupt generation for correctable and uncorrectable detected errors
- ECC Diagnostics Configuration:
  - Counters for single error correction (SEC), double error detection (DED) and address error events (AEE)
  - Programmable threshold registers for exceptions associated with SEC, DED and AEE counters
  - Register control for enabling and disabling of diagnostics
  - Configuration registers and ECC status accessible through L4 interconnect
- Circular buffer for sliced based VIP frame transfers:
  - Up to 12 programmable circular buffers mapped with unique virtual frame addresses
  - On the fly (with no additional latency) address translation from virtual to OCMC circular buffer memory space
  - Virtual frame size up to 8 MiB and circular buffer size up to 1 MiB
  - Error handling and reporting of illegal CBUF addressing
  - Underflow and Overflow status reporting and error handling
  - Last access read/write address history
- Two Interrupt outputs configured independently to service either ECC or CBUF interrupt events

The OCM controller does not have a memory protection logic and does not support endianism conversion.

10.2 EMIF Controller

This section describes mainly the features and functions of the external memory interface (EMIF) controller and also its associated PHYs.

10.2.1 EMIF Controller Overview

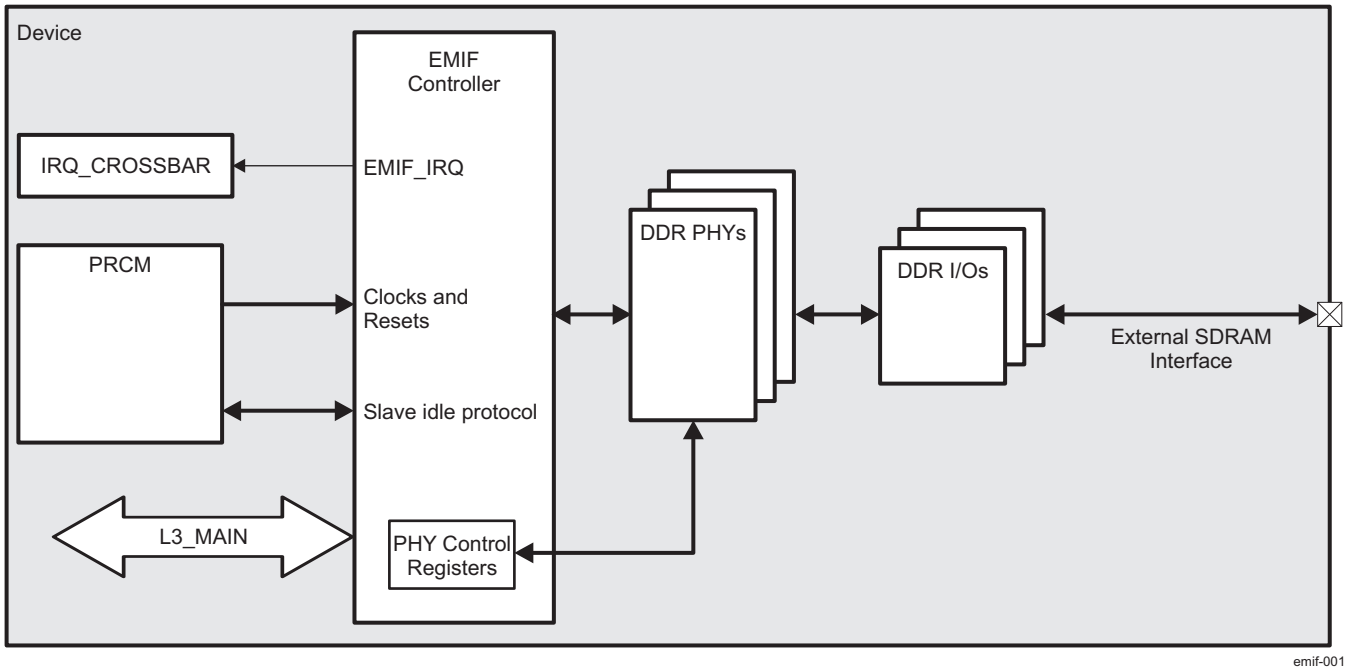
The EMIF controller provides connectivity between the device and LPDDR2/DDR2/DDR3/DDR3L types of memories and manages data bus read/write accesses between external memories and the device subsystems which have access to the L3_MAIN interconnect and DMA capability too.

NOTE: LPDDR2 is not supported on the DRA78x family of devices.

The EMIF features are introduced in Section 10.1.1 *EMIF Overview* of Section 10.1 *Memory Subsystem Overview*.

The device includes one EMIF controller. Figure 10-2 shows an overview of this EMIF controller and also the connections to the other surrounding modules. As can be seen on Figure 10-2 the EMIF is not directly available on device pads. That is, it is not directly connected to the external SDRAM. There are DDR PHYs and then DDR I/Os between the EMIF controller and external SDRAM. The EMIF controller, the DDR PHYs and the DDR I/Os work like a single unit to manage data exchanges to and from external memories. To achieve successful data transaction between an internal device initiator and external SDRAM all these three components must be used.

Figure 10-2. EMIF Controller Overview



10.2.2 EMIF Module Environment

This section describes the external connections of the EMIF module.

Figure 10-3 shows an example EMIF DDR2/DDR3/DDR3L configuration without ECC memory connected.

Figure 10-4 shows an example EMIF DDR2/DDR3/DDR3L configuration with ECC memory connected.

Figure 10-5 shows an example EMIF LPDDR2 configuration without ECC memory connected.

Figure 10-6 shows an example EMIF LPDDR2 configuration with ECC memory connected.

For simplification the DDR PHYs and DDR I/Os are not shown. Only the I/O signals and their corresponding EMIF pins are shown.

NOTE: LPDDR2 is not supported on the DRA78x family of devices.

Figure 10-3. EMIF DDR2/DDR3/DDR3L Configuration Without ECC

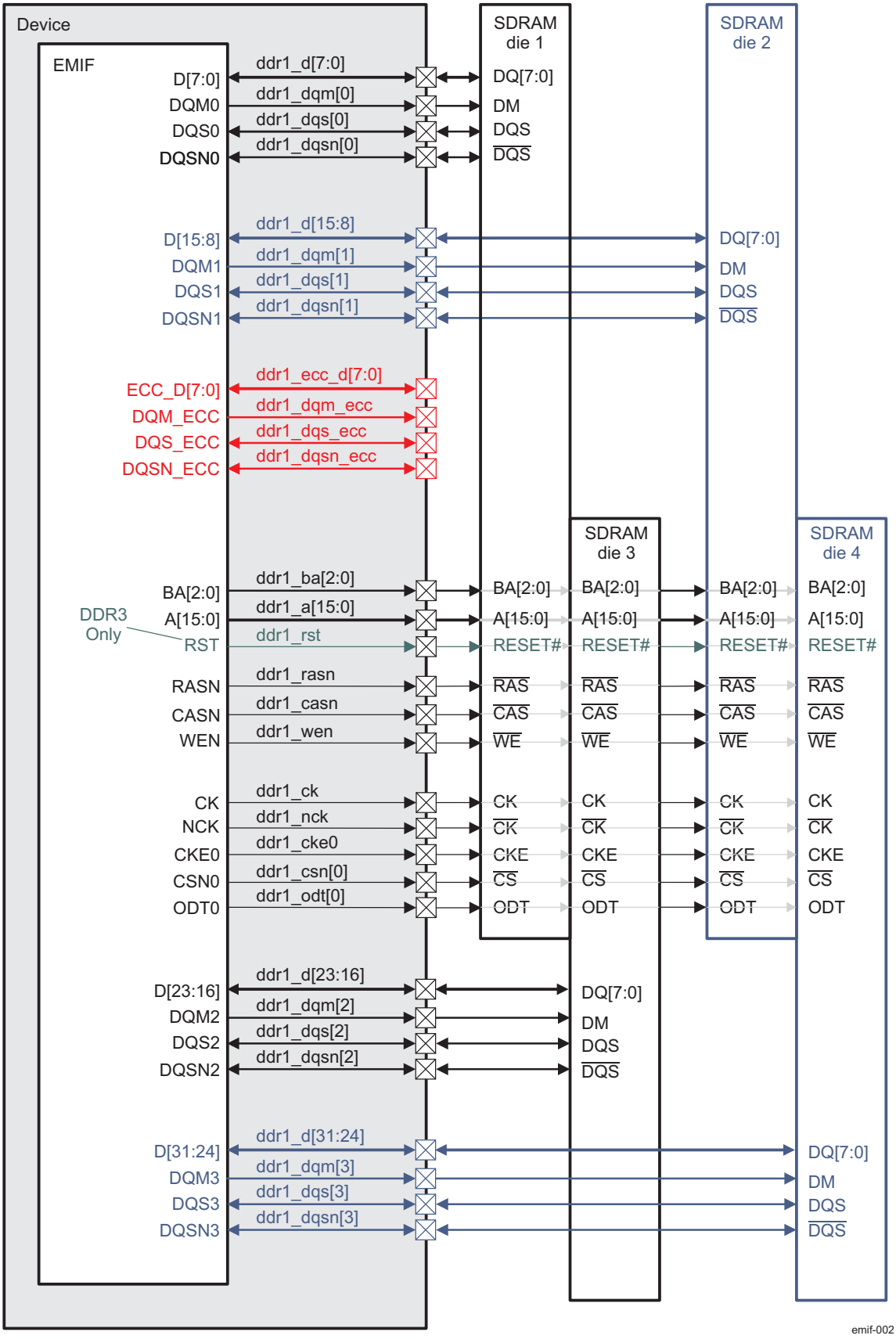


Figure 10-4. EMIF DDR2/DDR3/DDR3L Configuration With ECC

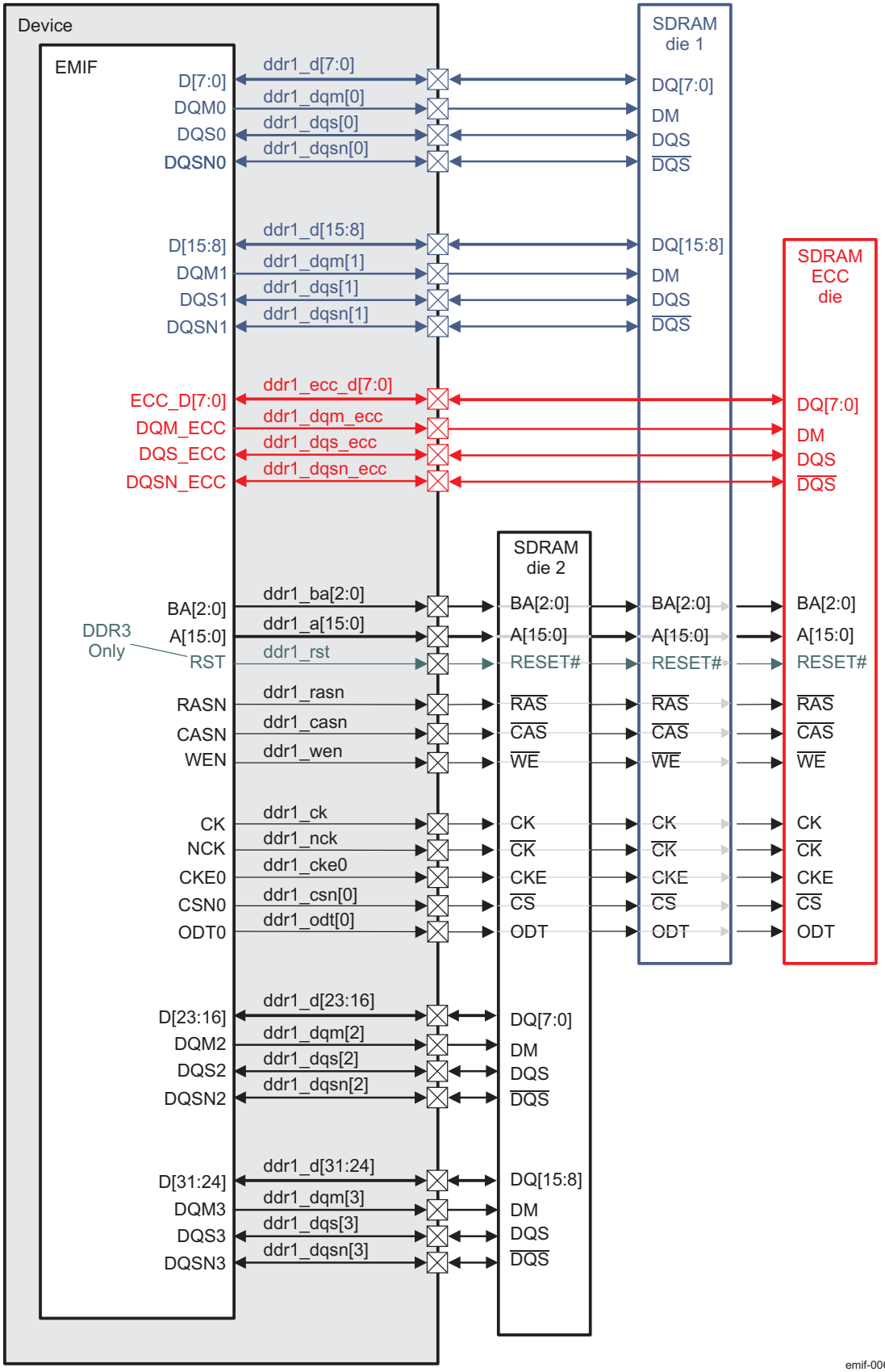


Figure 10-5. EMIF LPDDR2 Configuration Without ECC

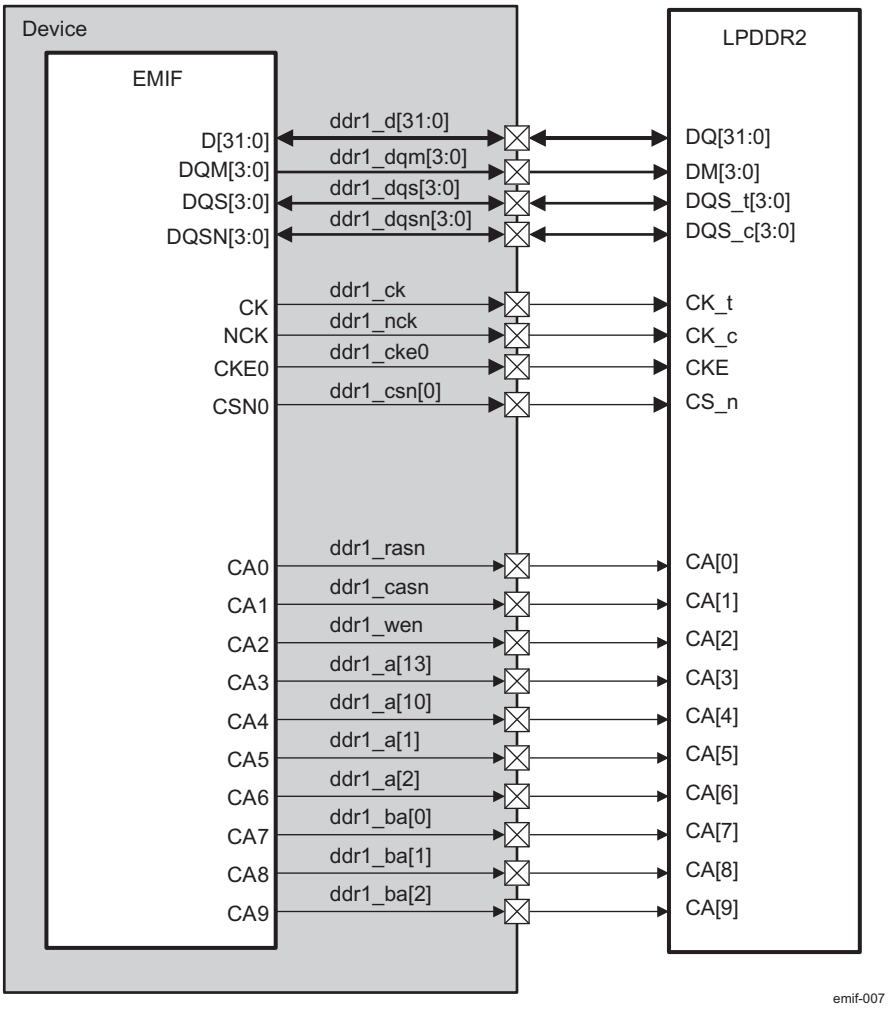


Figure 10-6. EMIF LPDDR2 Configuration With ECC

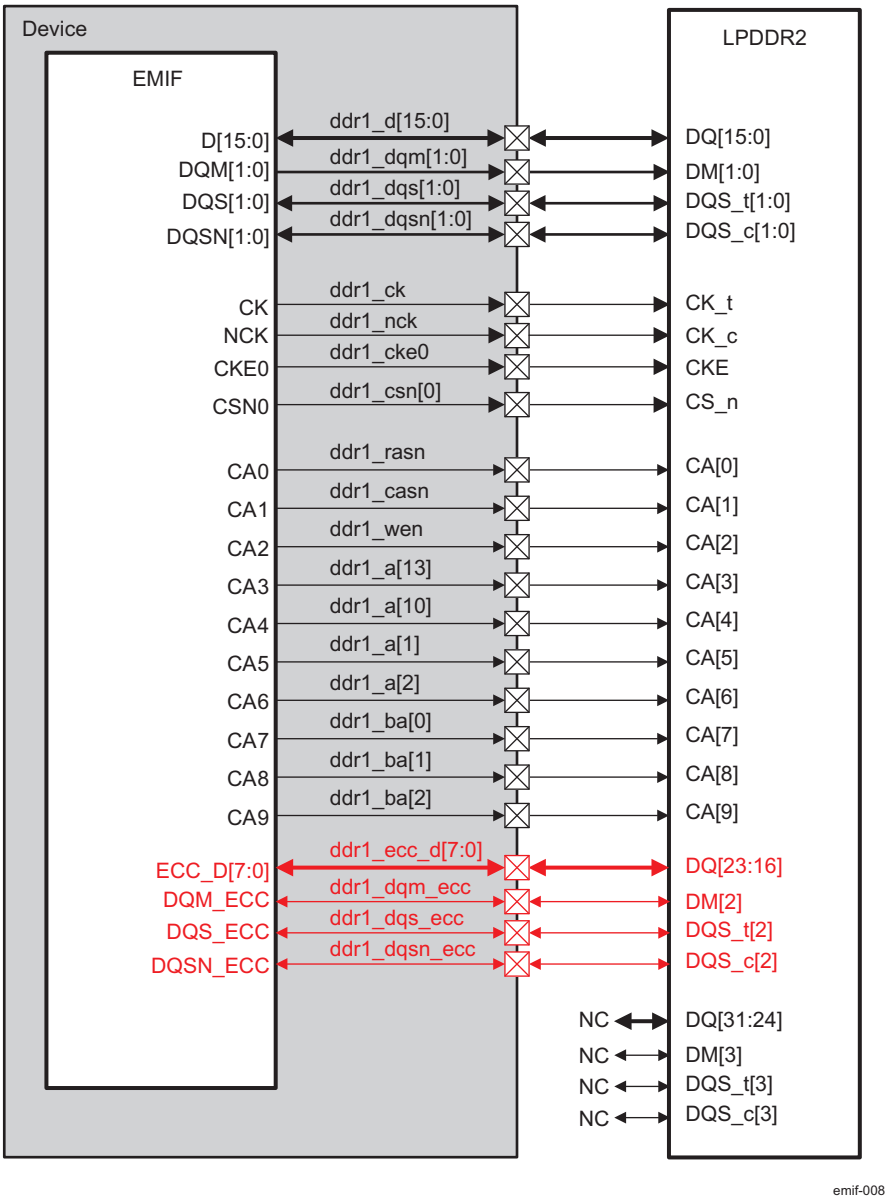


Table 10-1 describes the EMIF module associated I/O signals used for connection to DDR2/DDR3/DDR3L memory types.

Table 10-2 describes the EMIF module associated I/O signals used for connection to LPDDR2 memory types.

Table 10-1. EMIF Module I/O Signals Used For Connection to DDR2/DDR3/DDR3L Memories

EMIF Pin Name	Device I/O Signal Names	I/O[1]	Description
EMIF Data PHYs			
D[31:0]	ddr1_d[31:0]	I/O	Data bus
DQM[3:0]	ddr1_dqm[3:0]	O	Data mask
DQS[3:0]	ddr1_dqs[3:0]	I/O	Data strobe
DQSN[3:0]	ddr1_dqsn[3:0]	I/O	Data strobe invert

(1) I = Input; O = Output

Table 10-1. EMIF Module I/O Signals Used For Connection to DDR2/DDR3/DDR3L Memories (continued)

EMIF Pin Name	Device I/O Signal Names	I/O[1]	Description
ECC_D	ddr1_ecc_d[7:0]	I/O	Data bus used for ECC
DQM_ECC	ddr1_dqm_ecc	O	Data mask used for ECC
DQS_ECC	ddr1_dqs_ecc	I/O	Data strobe used for ECC
DQSN_ECC	ddr1_dqsn_ecc	I/O	Data strobe invert used for ECC
EMIF Command PHYs			
A[15:0]	ddr1_a[15:0]	O	Row/column address bus
BA[2:0]	ddr1_ba[2:0]	O	Bank select
CK	ddr1_ck	O	Differential clock
NCK	ddr1_nck	O	Differential clock
CSN0	ddr1_csn[0]	O	Active low rank select signal (chip select 0)
CKE0	ddr1_cke0	O	Clock enable
CASN	ddr1_casn	O	Command
RASN	ddr1_rasn	O	Command
WEN	ddr1_wen	O	Command
RST	ddr1_rst	O	Active low asynchronous reset (DDR3/3L only)
ODT0	ddr1_odt[0]	O	On-die termination enable signal

Table 10-2. EMIF Module I/O Signals Used For Connection to LPDDR2 Memories

EMIF Pin Name	Device I/O Signal Names	I/O[1]	Description
EMIF Data PHYs			
D[31:0]	ddr1_d[31:0]	I/O	Data bus
DQM[3:0]	ddr1_dqm[3:0]	O	Data mask
DQS[3:0]	ddr1_dqs[3:0]	I/O	Data strobe
DQSN[3:0]	ddr1_dqsn[3:0]	I/O	Data strobe invert
ECC_D	ddr1_ecc_d[7:0]	I/O	Data bus used for ECC [2]
DQM_ECC	ddr1_dqm_ecc	O	Data mask used for ECC [2]
DQS_ECC	ddr1_dqs_ecc	I/O	Data strobe used for ECC [2]
DQSN_ECC	ddr1_dqsn_ecc	I/O	Data strobe invert used for ECC [2]
EMIF Command PHYs			
CK	ddr1_ck	O	Differential clock
NCK	ddr1_nck	O	Differential clock
CSN0	ddr1_csn[0]	O	Active low rank select signal (chip select 0)
CKE0	ddr1_cke0	O	Clock enable
CA0	ddr1_rasn	O	Command/Address bus - bit 0
CA1	ddr1_casn	O	Command/Address bus - bit 1
CA2	ddr1_wen	O	Command/Address bus - bit 2
CA3	ddr1_a[13]	O	Command/Address bus - bit 3
CA4	ddr1_a[10]	O	Command/Address bus - bit 4
CA5	ddr1_a[1]	O	Command/Address bus - bit 5
CA6	ddr1_a[2]	O	Command/Address bus - bit 6
CA7	ddr1_ba[0]	O	Command/Address bus - bit 7
CA8	ddr1_ba[1]	O	Command/Address bus - bit 8

(1) I = Input; O = Output

(2) These signals are available only in non-PoP package. For more information about signal availability in PoP and non-PoP packages, see the corresponding device data manual.

Table 10-2. EMIF Module I/O Signals Used For Connection to LPDDR2 Memories (continued)

EMIF Pin Name	Device I/O Signal Names	I/O[1]	Description
CA9	ddr1_ba[2]	O	Command/Address bus - bit 9

The CKE0 memory pin is dynamically driven by the EMIF module according to the memory interface activity.

The ddr1_cke0 pad can also be forced to tri-state by a dedicated Control Module register. For more information, see Section 10.2.4.18 *Forcing CKE to tri-state*.

The DDR memory connected to the DDR ECC bus does NOT need to be the same part number as the DDR memories connected to the DDR data bus. However, some constraints do apply. When selecting a memory for the DDR ECC bus, the following restrictions must be adhered to.

Compared to the DDR memories on the data bus, the DDR ECC memory must:

- Match the same DDR type (for example, DDR3) and speed grade
- Have an equal number of internal banks
- Have an equal number of columns
- Have a greater or equal number of rows

In addition,

- Unused pins should be properly tied off as described in the routing guidelines of the device data manual.
- EMIF register settings should be configured to satisfy the larger minimum timing requirements and the smaller maximum timing requirements between the two different DDR memories to ensure that all DDR memories connected to the EMIF channel are running within their specified range.

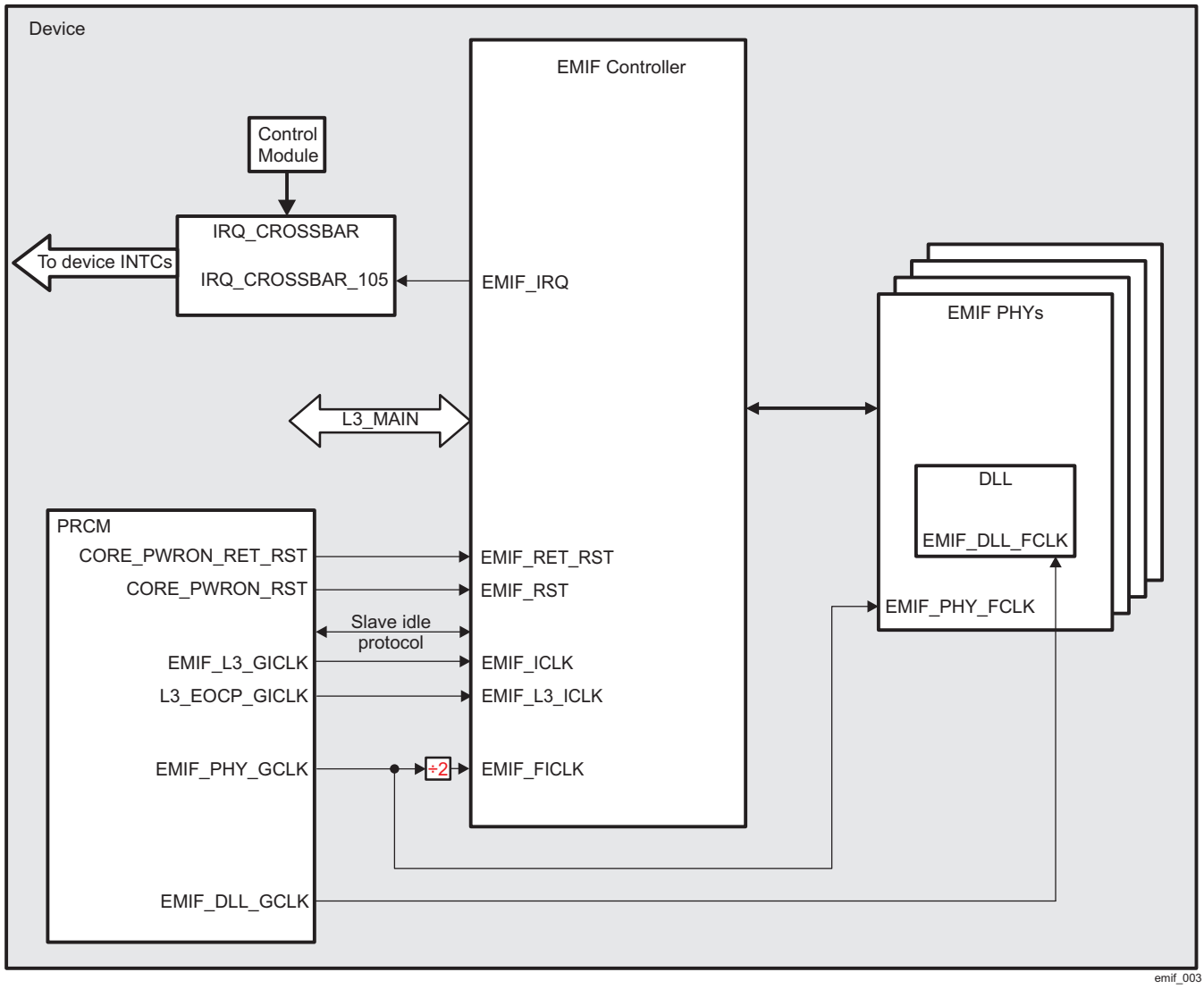
NOTE: For a full list of supported DDR device types, frequencies, and topologies, refer to the routing guidelines of the device data manual.

10.2.3 EMIF Module Integration

This section describes the integration of the EMIF module in the device and includes information about clocks, resets, and hardware requests.

Figure 10-7 shows the integration of the EMIF module in the device.

Figure 10-7. EMIF Module Integration



NOTE: For more information about the slave idle protocol, see Section 3.1.1.1.2, *Module-Level Clock Management*, in Chapter 3, *Power, Reset, and Clock Management*.

Table 10-3 through Table 10-5 summarize the integration of the EMIF module in the device.

Table 10-3. EMIF Integration Attributes

Module Instance	Attributes		
	Power Domain	Wake-Up Capability	Interconnect
EMIF	PD_COREAON	No	L3_MAIN

Table 10-4. EMIF Clocks and Resets

Clocks				
Module Instance	**Destination Signal Name**	**Source Signal Name**	**Source**	**Description**
EMIF	EMIF_ICLK	EMIF_L3_GICLK	PRCM	Interface clock for EMIF used for driving the L3 interface logic and for SDRAM Read Data FIFO.
	EMIF_L3_ICLK	L3_EOCP_GICLK	PRCM	Additional interface clock for EMIF which frequency is equal to EMIF_L3_GICLK interface clock. Used for command/write data pre-FIFO to Command/Write Data FIFO paths.
	EMIF_PHY_FCLK	EMIF_PHY_GCLK	PRCM	Common functional clock for the EMIF associated PHYs. This clock is equal to the DDR-SDRAM clock rate.
	EMIF_FICLK	EMIF_PHY_GCLK/2	PRCM	Functional and interface clock for the EMIF. This clock runs at half the DDR-SDRAM clock rate.
	EMIF_DLL_FCLK	EMIF_DLL_GCLK	PRCM	Common functional clock for all DLLs associated with the EMIF PHYs.
Resets				
Module Instance	**Destination Signal Name**	**Source Signal Name**	**Source**	**Description**
EMIF	EMIF_RET_RST	CORE_PWRON_RET_RST	PRCM	Power-on reset
	EMIF_RST	CORE_PWRON_RST	PRCM	Power-on reset

Table 10-5. EMIF Hardware Requests

Interrupt Requests				
Module Instance	**IRQ Source Name**	**IRQ_CROSSBAR Input**	**Default IRQ Source Mapping**	**Description**
EMIF	EMIF_IRQ	IRQ_CROSSBAR_105	-	EMIF interrupt request. This IRQ source signal is not mapped by default to any device INTC.

NOTE: The "**Default IRQ Source Mapping**" column in Table 10-5 *EMIF Hardware Requests* shows the default mapping of the IRQ sources listed in column "**IRQ Source Name**" to a certain interrupt line of one of the device interrupt controllers. These IRQ sources can also be mapped to other interrupt lines of each device interrupt controller through the IRQ_CROSSBAR module. For more information about the IRQ_CROSSBAR module, see Section 13.4.6.4, *IRQ_CROSSBAR Module Functional Description*, in Chapter 13, *Control Module*. For more information about the device interrupt controllers, see Chapter 12, *Interrupt Controllers*.

NOTE: For the description of the interrupt source, see Section 10.2.4.5, *Interrupt Requests*.

10.2.4 EMIF Functional Description

10.2.4.1 Block Diagram

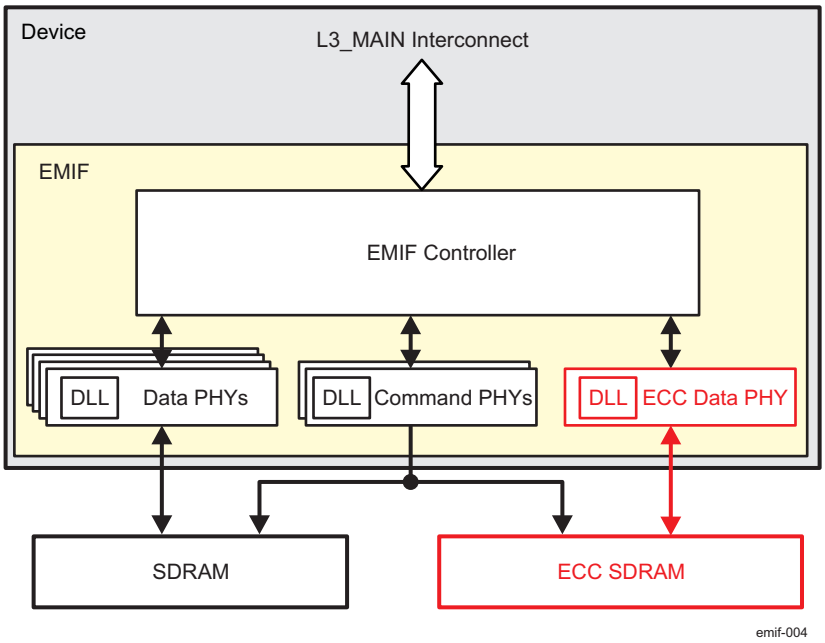
The EMIF module provides an interface to LPDDR2, DDR2 and DDR3/DDR3L SDRAM memories.

Figure 10-8 shows the interconnection between the EMIF module and the other modules.

Digital locked loops (DLLs) are used to delay the input DQS signals during reads so that these strobe signals can be used to latch incoming data on the DQ pins, as required by the DDR standard.

Physical layers (PHYs) convert single-data rate (SDR) signals to DDR signals.

Figure 10-8. EMIF Block Diagram



10.2.4.1.1 Local Interface

The EMIF controller has only one local (on-chip) interface. It is called system interface and connects the EMIF to the L3_MAIN interconnect. The system interface is used to request all external memory device accesses, to access the EMIF registers, and to transfer all data to and from the EMIF controller.

10.2.4.1.2 FIFO Description

The EMIF module contains the following FIFOs:

- Command FIFO
- Write data FIFO
- Return command FIFO
- Two read data FIFOs

Figure 10-9 shows the overall architecture of the EMIF FIFOs.

Figure 10-9. FIFO Block Diagram

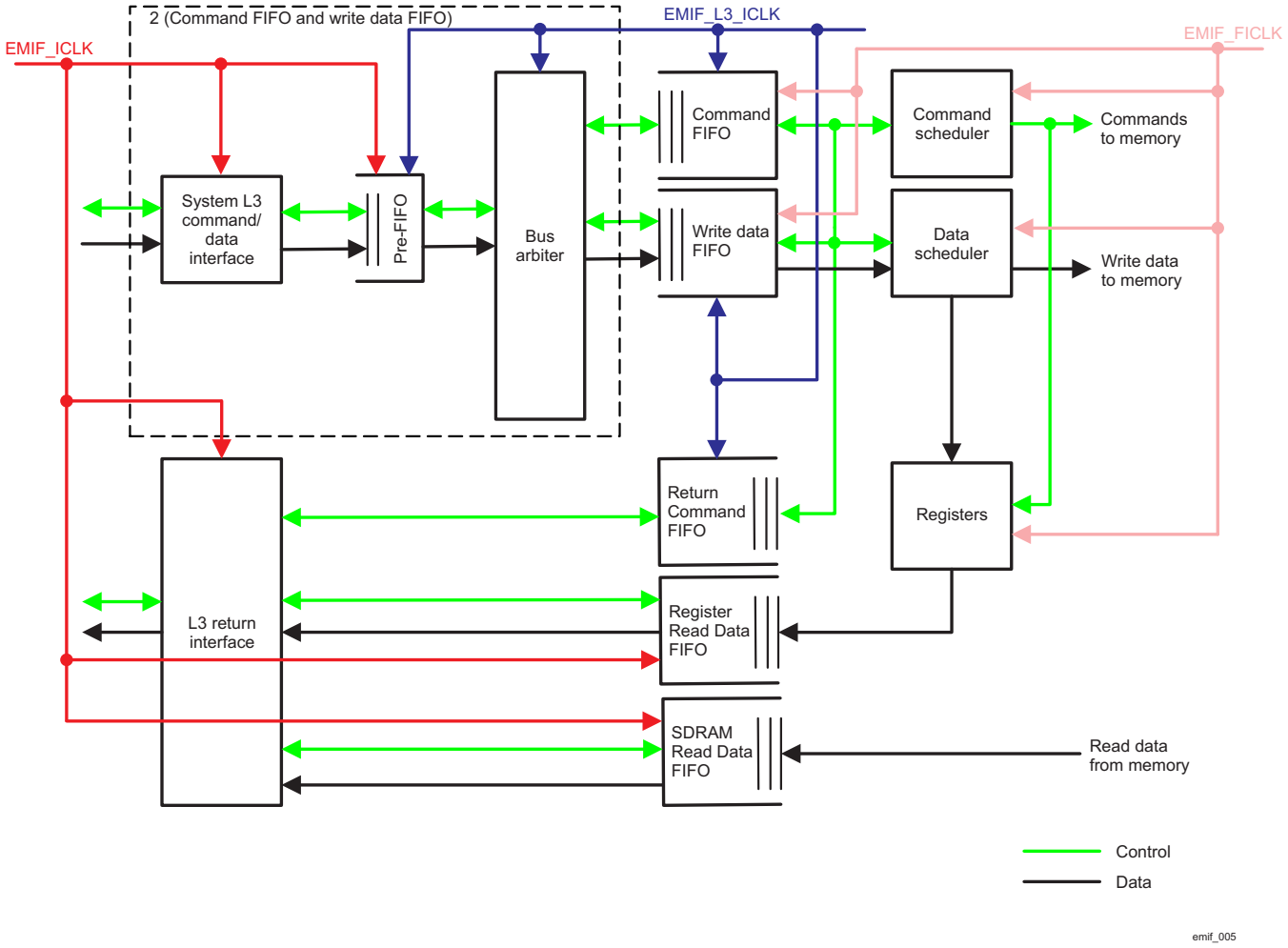


Table 10-6 lists the allocation of the entries.

Table 10-6. FIFO Allocation

Parameter	Local System Interface Entries
Pre Command FIFO	6
Command FIFO	Programmable (1)
Pre Write FIFO	6
Write Data FIFO (256-bit)	Up to (19 × 256 bits) + 6
Return Command FIFO	22
SDRAM Read Data FIFO	22
Register Read Data FIFO	2

(1) The total number of entries in the command FIFO is 10.

The command FIFO stores all the commands coming in on the local command interface. The allocation of entries in the command FIFO is programmable using the EMIF_OCP_CONFIG[27:24] SYS_THRESH_MAX bit field.

The write data FIFO stores the write data for all write transactions coming in on the local write data interface.

The return command FIFO stores all the return transactions that have to be issued to the local return interface. These include the write status return and the read data return commands.

Two read data FIFOs store the read data to be sent to the local return interface. One read data FIFO stores read data from the memory mapped registers. The other read data FIFO stores read data from external memory.

A write command is executed only when four local words of the corresponding data are received in the write data FIFO, or when it is the last data, and if there is space in the return command FIFO. The EMIF schedules a read-only if the results can fit into the return command FIFO and the corresponding read data FIFOs.

10.2.4.1.3 Arbitration of Commands in the Command FIFO

The EMIF looks at all the commands stored in the command FIFO to schedule commands to the external memory. All commands with the same MTagID on a particular local interface complete in order. The EMIF does not ensure ordering between commands with different MTagIDs. For information about MTagID, see Section 9.1.1, *Terminology* in Chapter 9, *Interconnect*.

The EMIF must also open and close SDRAM banks and maintain the refresh counts for an SDRAM. The priority of SDRAM commands with respect to refresh levels are:

1. SDRAM refresh request when refresh-must level is reached (highest priority)
2. ZQ calibration
3. Leveling
4. local request for a read or write
5. local request for a write
6. SDRAM activate commands
7. SDRAM deactivate commands
8. SDRAM power-down request
9. SDRAM refresh request when refresh-may or release level is reached
10. SDRAM self-refresh request (lowest priority)

To avoid continuous blocking effect which can be caused by a continuous stream of high-priority commands which thus block the lower priority commands, the EMIF momentarily raises the priority of the oldest command over all other commands when the time for the oldest command configured through the EMIF_COS_CONFIG[7:0] PR_OLD_COUNT bit field expires.

It should be taken into account that while performing the scheduling algorithm described, the EMIF may also encounter a condition in which continuous stream of SDRAM commands to a row in an open bank can block commands to another row in the same bank.

In addition to this scheduling, the highest priority condition is a reset command. If this condition occurs, the EMIF abandons what it is currently doing and begins its start-up sequence. In this case, commands and data stored in the FIFOs are lost. The EMIF also starts its start-up sequence whenever the EMIF_SDRAM_CONFIG register is written and the EMIF_SDRAM_REFRESH_CONTROL[31] INITREF_DIS bit is set to 0. In this case, commands and data stored in the FIFOs are not lost. The EMIF ensures that in-flight read or write transactions to the SDRAM are complete before starting the initialization sequence.

All the accesses to an SDRAM are pipelined to maximize use of the external bus. All of these are done while fulfilling the access timing requirements of an SDRAM.

10.2.4.2 Clock Management

The EMIF can gate EMIF_FICLK. There is an internal mechanism that can stop EMIF_FICLK automatically. EMIF_FICLK is stopped only after the SDRAM is put into self-refresh mode and the power-idle protocol on the local bus completes. The EMIF_FICLK frequency can be changed only after putting the external SDRAM in self-refresh mode.

The EMIF waits for the DLL lock before performing any memory access.

EMIF_FICLK frequency is equal to half of the EMIF_PHY_FCLK frequency.

10.2.4.3 Reset

The EMIF does not support a software reset.

The EMIF supports a global warm reset mode, during which the EMIF keeps the SDRAM content. Upon a request from the PRCM module indicating a need to enter global warm reset mode, the EMIF does the following:

1. During leveling operation, EMIF will immediately exit this mode and automatically perform a write to the MR1 register of DDR3 memory to disable the leveling at the memory side too.
2. EMIF completes the ongoing access, and then puts the SDRAM in self-refresh mode. If the EMIF_SDRAM_REFRESH_CONTROL[31] INITREF_DIS field is set to 1, the EMIF does not put SDRAM in self-refresh mode.
3. EMIF clears all its FIFO contents.
4. EMIF does not wait for all interrupts to be serviced.

To exit the global warm reset:

1. If the EMIF was in Self Refresh state, it will exit Self Refresh state.
2. If leveling was enabled at the time of a global warm reset, a PHY reset must occur to bring the PHY back into a known state, as it may have been left in a leveling state upon warm reset assertion. To guarantee that the SDRAM memory clocks are off when issuing PHY reset, software can use the EMIF_POWER_MANAGEMENT_CONTROL register to enter self refresh before asserting the PHY reset.

10.2.4.4 System Power Management

10.2.4.4.1 Power-Down Mode

The EMIF supports SDRAM power-down mode for low power. The EMIF automatically puts the SDRAM into power-down mode after it is idle for EMIF_POWER_MANAGEMENT_CONTROL[15:12] PD_TIM number of DDR clock cycles and the EMIF_POWER_MANAGEMENT_CONTROL[10:8] LP_MODE bit field is set to 0x4. In power-down mode, the EMIF does not stop the clocks to the SDRAM. The EMIF maintains CKE pin low to maintain the power-down mode.

If refresh-must level is not reached before power-down entering, EMIF will not precharge all SDRAM banks before it issues the power-down command. As a result of this EMIF puts the SDRAM in active power-down mode. If refresh-must level is reached before power-down entering, EMIF will precharge all SDRAM banks and before it issues the power-down command, EMIF issues refreshes until refresh-release level is reached. As a result of this EMIF puts the SDRAM in precharge power-down mode.

When the SDRAM is in power-down mode, the EMIF services register accesses normally.

If the SDRAM is in power-down mode and one of the following occurs, the EMIF brings SDRAM out of power-down mode:

- EMIF_POWER_MANAGEMENT_CONTROL[10:8] LP_MODE bit field is changed from 0x4 to some other value.
- An SDRAM access is requested.
- The refresh-must level is reached.

If refresh-must level brings the SDRAM out of power-down mode, EMIF puts it in power-down again when the refreshes are complete and keeps this state until the next SDRAM request.

To exit power-down, the EMIF:

1. Drives CKE high after t_cke + 1 cycles have elapsed since the power-down command was issued. The value of t_cke is taken from EMIF_SDRAM_TIMING_2[2:0] T_CKE bit field.
2. Waits for EMIF_SDRAM_TIMING_2[30:28] T_XP + 1 cycles
3. Enters its idle state and can issue any commands

10.2.4.4.2 LPDDR2 Deep Power-Down Mode

NOTE: LPDDR2 is not supported on the DRA78x family of devices.

For ultimate power savings, the EMIF supports deep power-down mode for LPDDR2.

The SDRAM can be forced into deep power-down mode through software by setting the EMIF_POWER_MANAGEMENT_CONTROL[11] DPD_EN bit to 0x1. In this case the EMIF continues normal operation until all SDRAM access requests are serviced. At this point the EMIF issues a deep power-down command. The EMIF then maintains CKE pin low to maintain deep power-down mode. In deep power-down mode, the EMIF automatically stops the clocks to the SDRAM.

Setting the EMIF_POWER_MANAGEMENT_CONTROL[11] DPD_EN bit to 1 overrides the setting of the EMIF_POWER_MANAGEMENT_CONTROL[10:8] LP_MODE bit field. Therefore, if the SDRAM is in any other power-saving mode, and the EMIF_POWER_MANAGEMENT_CONTROL[11] DPD_EN bit is set to 1, the EMIF exits those modes and goes into deep power-down mode.

When the SDRAM is in deep power-down mode, the EMIF services register accesses normally.

If the EMIF_POWER_MANAGEMENT_CONTROL[11] DPD_EN bit is set to 0x0, the EMIF brings the SDRAM out of deep power-down mode.

10.2.4.4.3 Self-Refresh Mode

The EMIF supports SDRAM self-refresh mode for low power. The EMIF automatically puts the SDRAM into self-refresh mode after the EMIF is idle for EMIF_POWER_MANAGEMENT_CONTROL[7:4] SR_TIMING number of DDR clock cycles and the EMIF_POWER_MANAGEMENT_CONTROL[10:8] LP_MODE bit field is set to 0x2. The EMIF will complete all pending refreshes before it puts the SDRAM into self-refresh. Therefore, after the expiration of SR_TIMING, the EMIF will start issuing refreshes to complete the refresh backlog, and then issue a self-refresh command to the SDRAM.

In self-refresh mode, the EMIF automatically stops the SDRAM clock. The EMIF drives CKE pin low to maintain self-refresh mode.

When the SDRAM is in self-refresh mode, the EMIF services register accesses normally.

If the SDRAM is in self-refresh mode and one of the following occurs, the EMIF brings SDRAM out of self-refresh mode:

- The EMIF_POWER_MANAGEMENT_CONTROL[10:8] LP_MODE bit field is changed from 0x2 to some other value.
- SDRAM access is requested.
- EMIF_SDRAM_TIMING_2[2:0] T_CKE + 1 cycles have elapsed since the last self-refresh command.

To exit self-refresh, for LPDDR2, the EMIF:

1. Waits for EMIF_SDRAM_TIMING_2[2:0] T_CKE + 1 cycles have elapsed since the self-refresh command was issued and enables clocks
2. Drives CKE pin high
3. Waits for EMIF_SDRAM_TIMING_2[24:16] T_XSNR + 1 cycles
4. Starts an auto-refresh cycle in the next cycle. It also services all refreshes down to the refresh-release level.
5. Enters its idle state and can issue any commands

To exit self-refresh, for DDR2, the EMIF:

1. Enables the SDRAM clock
2. Drives CKE pin high
3. Waits EMIF_SDRAM_TIMING_2[24:16] T_XSNR + 1 cycles
4. Starts a refresh cycle in the next cycle.
5. Enters its idle state and can issue any other command except write or read command. A write or a read command is issued only after EMIF_SDRAM_TIMING_2[15:6] T_XSRD + 1 cycles clock cycles

have elapsed since pin CKE is driven high.

To exit self-refresh, for DDR3, the EMIF does the following:

1. Enables the SDRAM clock
2. Drives CKE pin high
3. Waits EMIF_SDRAM_TIMING_2[24:16] T_XSNR + 1 cycles
4. Starts a refresh cycle in the next cycle. EMIF also services all refreshes down to the refresh-release level.
5. Enters its idle state and can issue any other command except write or read command. A write or a read command is issued only after EMIF_SDRAM_TIMING_2[15:6] T_XSRD + 1 cycles clock cycles have elapsed since pin CKE is driven high.

To use partial array self-refresh, the EMIF_SDRAM_REFRESH_CONTROL[26:24] PASR bits must be appropriately programmed. The EMIF performs bank interleaving when EMIF_SDRAM_CONFIG[28:27] IBANK_POS = 0x0. Because the SDRAM is partially refreshed during partial array self-refresh, for software ease, it is recommended that the IBANK_POS bit field to be set to 0x1, 0x2, or 0x3 depending on the scheme used. If IBANK_POS is set to 0x0, software must move critical data into the banks that are going to be refreshed during partial array self-refresh.

10.2.4.5 Interrupt Requests

The EMIF controller generates one interrupt request (EMIF_IRQ) which is connected to the IRQ_CROSSBAR module. This interrupt line can be asserted by one of the interrupt events listed in Table 10-7.

The EMIF controller generates an interrupt on its interrupt line only if the interrupts are enabled by setting to 0x1 the corresponding bits in the EMIF_SYSTEM_OCP_INTERRUPT_ENABLE_SET register. These interrupts can be disabled by setting to 0x1 the corresponding bits in the EMIF_SYSTEM_OCP_INTERRUPT_ENABLE_CLEAR register. After the interrupt has been serviced the corresponding status flag must be cleared by software. This is done by setting to 0x1 the corresponding bit in the EMIF_SYSTEM_OCP_INTERRUPT_STATUS register which also clears the corresponding bit in the EMIF_SYSTEM_OCP_INTERRUPT_RAW_STATUS register. The status flags in the EMIF_SYSTEM_OCP_INTERRUPT_RAW_STATUS register are set even if the corresponding interrupt is disabled unlike those in the EMIF_SYSTEM_OCP_INTERRUPT_STATUS register, which are set only if the corresponding interrupt is enabled. An interrupt is also generated by the EMIF controller, if certain bit in the EMIF_SYSTEM_OCP_INTERRUPT_RAW_STATUS register is set to 0x1 and the corresponding interrupt is enabled through the EMIF_SYSTEM_OCP_INTERRUPT_ENABLE_SET register. This feature is useful when user software debugging is performed. In addition, even if interrupts are not enabled, certain status bit in EMIF_SYSTEM_OCP_INTERRUPT_RAW_STATUS register can be cleared by setting to 0x1 the corresponding bit in the EMIF_SYSTEM_OCP_INTERRUPT_STATUS register.

The EMIF sets both the EMIF_SYSTEM_OCP_INTERRUPT_RAW_STATUS[0] ERR_SYS and EMIF_SYSTEM_OCP_INTERRUPT_STATUS[0] ERR_SYS bits to 0x1, if access request for an unsupported command type or an unsupported addressing mode is received.

The EMIF sets both the EMIF_SYSTEM_OCP_INTERRUPT_RAW_STATUS[3] WR_ECC_ERR_SYS and EMIF_SYSTEM_OCP_INTERRUPT_STATUS[3] WR_ECC_ERR_SYS bits to 0x1, if a write access with byte count that is not multiple of the ECC quanta or with a non ECC quanta aligned address is performed within the address range protected by the ECC.

The EMIF sets both the EMIF_SYSTEM_OCP_INTERRUPT_RAW_STATUS[4] TWOBIT_ECC_ERR_SYS and EMIF_SYSTEM_OCP_INTERRUPT_STATUS[4] TWOBIT_ECC_ERR_SYS bits to 0x1, if 2-bit ECC error for a read access performed within the address range protected by the ECC occurs.

The EMIF sets both the EMIF_SYSTEM_OCP_INTERRUPT_RAW_STATUS[5] ONEBIT_ECC_ERR_SYS and EMIF_SYSTEM_OCP_INTERRUPT_STATUS[5] ONEBIT_ECC_ERR_SYS bits to 0x1, if the threshold for 1-bit ECC error is reached. For more information about the EMIF ECC feature, see Section 10.2.4.15, *Error Correction And Detection Feature*.

Table 10-7 lists the event flags and their corresponding event mask bits of the sources which can cause module interrupts.

Table 10-7. Events

Event Flag	Event Mask	Description
EMIF_SYSTEM_OCP_INTERRUPT_RAW_STATUS[5] ONEBIT_ECC_ERR_SYS/ EMIF_SYSTEM_OCP_INTERRUPT_STATUS[5] ONEBIT_ECC_ERR_SYS	EMIF_SYSTEM_OCP_INTERRUPT_ENABLE_SET[5] ONEBIT_ECC_ERR_SYS/ EMIF_SYSTEM_OCP_INTERRUPT_ENABLE_CLEAR[5] ONEBIT_ECC_ERR_SYS	Interrupt if one bit ECC error threshold is reached
EMIF_SYSTEM_OCP_INTERRUPT_RAW_STATUS[4] TWOBIT_ECC_ERR_SYS/ EMIF_SYSTEM_OCP_INTERRUPT_STATUS[4] TWOBIT_ECC_ERR_SYS	EMIF_SYSTEM_OCP_INTERRUPT_ENABLE_SET[4] TWOBIT_ECC_ERR_SYS/ EMIF_SYSTEM_OCP_INTERRUPT_ENABLE_CLEAR[4] TWOBIT_ECC_ERR_SYS	Interrupt for two bit error detection
EMIF_SYSTEM_OCP_INTERRUPT_RAW_STATUS[3] WR_ECC_ERR_SYS/ EMIF_SYSTEM_OCP_INTERRUPT_STATUS[3] WR_ECC_ERR_SYS	EMIF_SYSTEM_OCP_INTERRUPT_ENABLE_SET[3] WR_ECC_ERR_SYS/ EMIF_SYSTEM_OCP_INTERRUPT_ENABLE_CLEAR[3] WR_ECC_ERR_SYS	Interrupt for memory access made to a non-quanta aligned location or done with byte count not multiple of the ECC quanta
EMIF_SYSTEM_OCP_INTERRUPT_RAW_STATUS[1] TA_SYS / EMIF_SYSTEM_OCP_INTERRUPT_STATUS[1] TA_SYS	EMIF_SYSTEM_OCP_INTERRUPT_ENABLE_SET[1] EN_TA_SYS / EMIF_SYSTEM_OCP_INTERRUPT_ENABLE_CLEAR[1] EN_TA_SYS	Interrupt for LPDDR2 temperature alert
EMIF_SYSTEM_OCP_INTERRUPT_RAW_STATUS[0] ERR_SYS/ EMIF_SYSTEM_OCP_INTERRUPT_STATUS[0] ERR_SYS	EMIF_SYSTEM_OCP_INTERRUPT_ENABLE_SET[0] EN_ERR_SYS/ EMIF_SYSTEM_OCP_INTERRUPT_ENABLE_CLEAR[0] EN_ERR_SYS	Interrupt for command or address error

10.2.4.6 SDRAM Refresh Scheduling

The EMIF uses two counters to schedule the Refresh (REF) commands: a 13-bit decrementing refresh interval counter and a 4-bit refresh backlog counter. The interval counter is used to define the rate at which connected SDRAM devices are refreshed. It is loaded with the value of the EMIF_SDRAM_REFRESH_CONTROL[15:0] REFRESH_RATE bit field at reset (only the 13 LSBs are taken). The interval counter decrements by 1 each cycle until it reaches 0x0, at which point it reloads from the EMIF_SDRAM_REFRESH_CONTROL[15:0] REFRESH_RATE bit field and restarts decrementing. The counter also reloads and restarts decrementing whenever the EMIF_SDRAM_REFRESH_CONTROL[15:0] REFRESH_RATE bit field is updated.

The refresh backlog counter records the number of the outstanding REF commands which the EMIF controller currently has. The backlog counter increments by 1 each time the interval counter reloads (unless it has reached its maximum value of 8). The backlog counter decrements by 1 each time the EMIF issues a REF command (unless it is already 0). For the range of values that the backlog counter can take, there are three levels of urgency with which the EMIF must perform refresh cycle in which it issues REF commands:

1. Refresh-may level is reached when the backlog count is greater than 0x0, which indicates that there is a refresh backlog and if the EMIF is not busy and there are no open SDRAM banks, the EMIF must perform refresh cycle.
2. Refresh-release level is reached when the backlog count is greater than 0x4, which indicates that the refresh backlog is getting bigger and if the EMIF is not busy it must perform refresh cycle even if there is an open SDRAM bank.
3. Refresh-must level is reached when the backlog count is greater than 0x7, which indicates that the refresh backlog is becoming excessive and the EMIF must perform refresh cycle before any new memory access request being serviced. The EMIF starts servicing new memory accesses after the refresh-release level is cleared.

The two counters do not operate when SDRAM is in self-refresh mode. They start tracking the missed refreshes (the outstanding REF commands) only after initialization is complete in case of DDR2/DDR3. For LPDDR2, the refresh counters will start tracking missed refreshes after RESET command is issued. For LPDDR2, the EMIF will also ensure that no more than 8 REF (Refresh) commands are issued in any rolling tREFBW (=4*8*tRFC) window.

The time between two REF commands is set through the EMIF_SDRAM_TIMING_3[12:4] T_RFC bit field.

10.2.4.7 SDRAM Initialization

NOTE: To avoid error responses from the EMIF controller the SDRAM initialization must be performed per 16-Byte blocks if EMIF_SDRAM_CONFIG[15:14] NARROW_MODE = 0x0 or per 8-Byte blocks if EMIF_SDRAM_CONFIG[15:14] NARROW_MODE = 0x1.

10.2.4.7.1 DDR2 SDRAM Initialization

On coming out of reset, the EMIF controller begins the DDR2 initialization sequence after a write to any one of the following three registers, providing that the corresponding listed conditions are met:

1. EMIF_SDRAM_CONFIG
   a. Condition 1: EMIF_SDRAM_REFRESH_CONTROL[31] INITREF_DIS = 0 (that is, cleared before the write to EMIF_SDRAM_CONFIG)
2. EMIF_SDRAM_TIMING_1
   a. Condition 1: EMIF_SDRAM_REFRESH_CONTROL[31] INITREF_DIS = 0 (that is, cleared before the write to EMIF_SDRAM_TIMING_1) AND
   b. Condition 2: The write access modifies the register bit field T_WR (bits 20:17)
3. EMIF_SDRAM_REFRESH_CONTROL
   a. Condition 1: EMIF_SDRAM_REFRESH_CONTROL[31] INITREF_DIS = 0 AND
   b. Condition 2: The write access modifies the register bit field SRT (bit 29), ASR (bit 28), or PASR (bits 26:24).

For the first DDR2 initialization sequence, the EMIF controller performs the following actions:

1. Drives the CKE pin low.
2. After 16 SDRAM refresh rate intervals, issues a NOP command with CKE pin held high. The SDRAM refresh rate is as defined in the EMIF_SDRAM_REFRESH_CONTROL[15:0] REFRESH_RATE bit field.
3. After 1 SDRAM refresh rate interval, issues Precharge all Banks command.
4. Issues EMRS command to the DDR2 EMR(2) register (bits BA[2:0] = 0x2) with bits A[15:0] = 0x0.
5. Issues EMRS command to the DDR2 EMR(3) register (bits BA[2:0] = 0x3) with bits A[15:0] = 0x0.
6. Issues EMRS command to the DDR2 EMR(1) register (bits BA[2:0] = 0x1) with bits A[15:0] set as in Table 10-8.

Table 10-8. Load Value For The EMR(1) Register During DDR2 SDRAM Initialization

Bits	Value	Description
A[15:13]	0x0	Reserved
A[12]	0x0	Output buffer enabled
A[11]	0x0	RDQS disable
A[10]	EMIF_SDRAM_CONFIG[23] DDR2_DDQS	Differential DQS enable value
A[9:7]	0x0	Exit OCD calibration mode
A[6], A[2]	EMIF_SDRAM_CONFIG[26:24] DDR_TERM	DDR2 termination resistor value. For DDR2 the EMIF_SDRAM_CONFIG[26] bit is not used.
A[5:3]	0x0	Additive latency = 0
A[1]	EMIF_SDRAM_CONFIG[19:18] SDRAM_DRIVE	SDRAM drive strength. For DDR2 the EMIF_SDRAM_CONFIG[19] bit is not used.
A[0]	EMIF_SDRAM_CONFIG[20] DDR_DISABLE_DLL = 0x0	Enable DLL

7. Issues MRS command to the DDR2 MR register (BA[2:0] = 0x0) with A[15:0] set as in Table 10-9.

Table 10-9. Load Value For The MR Register During DDR2 SDRAM Initialization

Bits	Value	Description
A[15:13]	0x0	Reserved
A[12]	0x0	Active power down exit time - fast exit
A[11:9]	EMIF_SDRAM_TIMING_1[20:17] T_WR	Write recovery for autoprecharge
A[8]	0x1	DLL reset
A[7]	0x0	Normal mode
A[6:4]	EMIF_SDRAM_CONFIG[13:10] CL	CAS latency value. For DDR2 the EMIF_SDRAM_CONFIG[13] bit is not used.
A[3]	0x0	Sequential burst type
A[2:0]	0x3	Burst length of 8

8. After 267 clock cycles, issues Precharge all Banks command.
9. After two Refresh commands, issues MRS command to the DDR2 MR register (BA[2:0] = 0x0) with A[15:0] set as in Table 10-10.

Table 10-10. Another Load Value For The MR Register During DDR2 SDRAM Initialization

Bits	Value	Description
A[15:9]	Equal to Step 7	
A[8]	0x0	No DLL reset
A[7:0]	Equal to Step 7	

10. Issues EMRS command to the DDR2 EMR(1) register (bits BA[2:0] = 0x1) with bits A[15:0] equal to Step 6.
11. The EMIF enters its IDLE state.

The EMIF updates the DDR Mode registers if the DDR2 initialization sequence is triggered again. However, the EMIF controller starts from Step 3.

The EMIF does not perform any transactions until the DDR2 initialization sequence is complete.

When the EMIF comes out of reset, the delay time in Step 2 resulting from the 16 refresh rate intervals + 8 cycles is approximately 16 × REFRESH_RATE / input frequency.

NOTE: The values of the bit fields in the EMIF_SDRAM_CONFIG register are loaded by the control module at reset. These values must not be modified during run time, because they reflect the used hardware SDRAM memory configurations.

10.2.4.7.2 DDR3/DDR3L SDRAM Initialization

On coming out of reset, the EMIF controller begins the DDR3 initialization sequence after a write to any one of the following three registers, providing that the corresponding listed conditions are met.

1. EMIF_SDRAM_CONFIG
   a. Condition 1: EMIF_SDRAM_REFRESH_CONTROL[31] INITREF_DIS = 0 (that is, cleared before the write to EMIF_SDRAM_CONFIG)
2. EMIF_SDRAM_TIMING_1
   a. Condition 1: EMIF_SDRAM_REFRESH_CONTROL[31] INITREF_DIS = 0 (that is, cleared before the write to EMIF_SDRAM_TIMING_1) AND
   b. Condition 2: The write access modifies the register bit field T_WR (bits 20:17)
3. EMIF_SDRAM_REFRESH_CONTROL
   a. Condition 1: EMIF_SDRAM_REFRESH_CONTROL[31] INITREF_DIS = 0 AND
   b. Condition 2: The write access modifies the register bit field SRT (bit 29), ASR (bit 28), or PASR (bits 26:24).

For the first DDR3 initialization sequence, the EMIF controller performs the following actions:

1. After 7 SDRAM refresh rate intervals, de-asserts the RST pin.
2. After 16 SDRAM refresh rate intervals, issues a NOP command with CKE pin held high. The SDRAM refresh rate is as defined in the EMIF_SDRAM_REFRESH_CONTROL[15:0] REFRESH_RATE bit field.
3. After 1 SDRAM refresh rate interval, issues MRS command to the DDR3/DDR3L MR2 register (bits BA[2:0] = 0x2) with bits A[15:0] set as in Table 10-11

Table 10-11. Load Value For The MR2 Register During DDR3/DDR3L SDRAM Initialization

Bits	Value	Description
A[15:11]	0x0	Reserved
A[10:9]	EMIF_SDRAM_CONFIG[22:21] DYN_ODT	Dynamic ODT value
A[8]	0x0	Reserved
A[7]	EMIF_SDRAM_REFRESH_CONTROL[29] SRT	Self-refresh temperature range
A[6]	EMIF_SDRAM_REFRESH_CONTROL[28] ASR	Auto self-refresh enable
A[5]	0x0	Reserved
A[4:3]	EMIF_SDRAM_CONFIG[17:16] CWL	CAS write latency
A[2:0]	EMIF_SDRAM_REFRESH_CONTROL[26:24] PASR	Partial array self-refresh

4. After T_RFC + 1 DDR clock cycles, issues MRS command to the DDR3/DDR3L MR3 register (bits BA[2:0] = 0x3) with A[15:0] = 0x0
5. After T_RFC + 1 DDR clock cycles, issues MRS command to the DDR3/DDR3L MR1 register (BA[2:0] = 0x1) with A[15:0] set as in Table 10-12

Table 10-12. Load Value For The MR1 Register During DDR3/DDR3L SDRAM Initialization

Bits	Value	Description
A[15:13]	0x0	Reserved
A[12]	0x0	Output buffer enabled
A[11]	0x0	TDQS disable
A[10]	0x0	Reserved
A[9], A[6], A[2]	EMIF_SDRAM_CONFIG[26:24] DDR_TERM	DDR3/DDR3L termination resistor value
A[8]	0x0	Reserved
A[7]	0x0	Write leveling disabled
A[5], A[1]	EMIF_SDRAM_CONFIG[19:18] SDRAM_DRIVE	SDRAM drive strength
A[4:3]	0x0	Additive latency = 0
A[0]	EMIF_SDRAM_CONFIG[20] DDR_DISABLE_DLL = 0x0	Enable SDRAM DLL

6. After T_RFC + 1 DDR clock cycles, issues MRS command to the DDR3/DDR3L MR0 register (BA[2:0] = 0x0) with A[15:0] set as in Table 10-13

Table 10-13. Load Value For The MR0 Register During DDR3/DDR3L SDRAM Initialization

Bits	Value	Description
A[15:13]	0x0	Reserved
A[12]	0x0	Slow exit. The DDR3 SDRAM DLL is “OFF” after entering precharge power-down.

Table 10-13. Load Value For The MR0 Register During DDR3/DDR3L SDRAM Initialization (continued)

Bits	Value	Description
A[11:9]	EMIF_SDRAM_TIMING_1[20:17] T_WR	Write recovery for autoprecharge
A[8]	0x1	DLL reset
A[7]	0x0	Normal mode
A[6:4], A[2]	EMIF_SDRAM_CONFIG[13:10] CL	Value for CAS latency
A[3]	0x0	Nibble sequential read burst type
A[1:0]	0x0	Burst length of 8

7. After T_RFC + 1 DDR clock cycles, issues a ZQCL command to start long ZQ calibration
8. Waits for t_{DLLK} and t_{ZQinit} to complete
9. Issues REF command
10. The EMIF enters its IDLE state.

The EMIF updates the DDR Mode registers if the DDR3 initialization sequence is triggered again. However, the EMIF controller first issues a precharge command and then starts from Step 3.

The EMIF does not perform any transactions until the DDR3/DDR3L initialization sequence is complete.

When the EMIF comes out of reset, the delay time in Step 2 resulting from the 16 refresh rate intervals + 8 cycles is approximately 16 × REFRESH_RATE / input frequency.

10.2.4.7.3 LPDDR2 SDRAM Initialization

NOTE: LPDDR2 is not supported on the DRA78x family of devices.

When coming out of reset, if the EMIF_SDRAM_CONFIG[31:29] SDRAM_TYPE bit field is equal to 4, the EMIF performs an LPDDR2 initialization sequence as follows:

1. Drives the CKE pin high and starts to issue NOP commands continuously
2. After 16 SDRAM refresh rate intervals, issues a Precharge-All command. The SDRAM refresh rate is defined in the EMIF_SDRAM_REFRESH_CONTROL[15:0] REFRESH_RATE bit field.
3. Issues a RESET command
4. The EMIF enters its IDLE state.
5. Software then must initialize LPDDR2 using the EMIF_LPDDR2_MODE_REG_CONFIG and EMIF_LPDDR2_MODE_REG_DATA registers. Software must enable refreshes by writing 1 in the EMIF_LPDDR2_MODE_REG_CONFIG[30] REFRESH_EN bit during the last MRW command.

The EMIF also performs the initialization sequence whenever the EMIF_SDRAM_CONFIG register is written and the LPDDR2 initialization was not performed previously because the EMIF_SDRAM_REFRESH_CONTROL[31] INITREF_DIS bit was set to 0. Once the EMIF performs initialization, rewriting to the EMIF_SDRAM_CONFIG register does not cause reinitialization.

The EMIF does not perform any transactions until the LPDDR2 initialization sequence completes.

When the EMIF comes out of reset, the delay time in step 2 resulting from the 16 refresh rate intervals + 8 cycles is approximately 16 × EMIF_SDRAM_REFRESH_CONTROL[15:0] REFRESH_RATE / input frequency.

10.2.4.8 DDR3/DDR3L Read-Write Leveling

The EMIF supports one type of write/read leveling for DDR3/DDR3L memories - full leveling.

The full leveling consists of three parts:

1. Write leveling
2. Read data eye training

3. Read DQS gate training

Write leveling

The goal of write leveling is to locate the delay between the rising edge of the write DQS signal and the rising edge of the SDRAM memory clock (CK). When this delay is identified, the system is able to accurately align the write DQS signal with the DDR3 memory clock. During Write leveling, the ODT function must be on and proper ODT values must be selected at the external memory side by setting Rtt_Nom (A9, A6, A2) bits in MR1 register of the external DDR3 memory. For more information about write leveling, see the *DDR3 SDRAM Standard*, section *Write leveling*.

Read data eye training

Through the read data eye training the delay between the rising edge of the read DQS signal and the rising and falling edges of the associated DQ data eye is determined. By identifying these delays, the midpoint between them can be calculated and thus the rising edge of the read DQS signal can be accurately centered within the DQ data eye.

Read DQS gate training

Read DQS Gate training is used for timing the internal read window during a read operation as opposed to the write leveling and read data eye training which are used for skew compensation of external signals. The goal of read DQS gate training is to locate the shortest delay that can be applied to each DQS gate such that it functions properly, then find the longest delay that can be applied to each DQS gate and keep its proper function again, and then align the midpoint of the DQS gate delay between these two.

10.2.4.8.1 Full Leveling

The EMIF does not perform full leveling after initialization upon reset. Full leveling must be triggered by software after the EMIF's registers are properly configured.

Full leveling is triggered by setting the EMIF_READ_WRITE_LEVELING_CONTROL[31] RDWRLVLFULL_START bit to 0x1. The leveling execution order is as follows:

1. Write leveling
2. Read DQS gate training
3. Read data eye training

After leveling procedure has finished the EMIF_READ_WRITE_LEVELING_CONTROL[31] RDWRLVLFULL_START bit clears itself automatically.

NOTE: SDRAM Refreshes must be disabled before triggering full leveling.

NOTE: The EMIF_EXT_PHY_CONTROL_2 through EMIF_EXT_PHY_CONTROL_21 registers have to be configured only in case of software leveling.

10.2.4.8.2 Software Leveling

In case of software leveling, the following registers must be configured:

- EMIF_EXT_PHY_CONTROL_2 to EMIF_EXT_PHY_CONTROL_6 containing the PHY_REG_FIFO_WE_SLAVE_RATIO value
- EMIF_EXT_PHY_CONTROL_26 to EMIF_EXT_PHY_CONTROL_30 containing the REG_PHY_GATELVL_INIT_RATIO value
- EMIF_EXT_PHY_CONTROL_25 containing the DQ offset value

The bit fields of the EMIF_EXT_PHY_CONTROL_2 to EMIF_EXT_PHY_CONTROL_6 registers must be loaded with same values. The bit fields of the EMIF_EXT_PHY_CONTROL_26 to EMIF_EXT_PHY_CONTROL_30 registers must also be loaded with same values.

The REG_PHY_DQ_OFFSET fields in register EMIF_EXT_PHY_CONTROL_25 must be loaded with 0x40. That value corresponds to quarter cycle shift between the DQS signals and the data to be written to the SDRAM.

To calculate the PHY_REG_FIFO_WE_SLAVE_RATIO value one of the following two formulas can be used:

- 0x80 + 2*[(Board Delay in ps)*0x100]/(Clock Period in ps). This formula is used when the EMIF_DDR_PHY_CONTROL_1[18] PHY_INVERT_CLKOUT bit is set to 0x1.
- 2*[(Board Delay in ps)*0x100]/(Clock Period in ps). This formula is used when the EMIF_DDR_PHY_CONTROL_1[18] PHY_INVERT_CLKOUT bit is set to 0x0.

The Board Delay in the previously described formulas is measured directly from the board. It depends on the trace length. When the calculated value for PHY_REG_FIFO_WE_SLAVE_RATIO is greater than 0x20, then a value of 0x27 must be used. That means, 0x27 must be loaded in registers EMIF_EXT_PHY_CONTROL_2 to EMIF_EXT_PHY_CONTROL_6.

To calculate the REG_PHY_GATELVL_INIT_RATIO value the following formula is used:

REG_PHY_GATELVL_INIT_RATIO = PHY_REG_FIFO_WE_SLAVE_RATIO – 0x20

When the calculated value for REG_PHY_GATELVL_INIT_RATIO is less than 0x0, then a value of 0x0 must be used. When the calculated value is greater than or equal to 0x0, then that value is used for registers EMIF_EXT_PHY_CONTROL_26 to EMIF_EXT_PHY_CONTROL_30.

In addition, when read DQS gate training is not performed the PHY_REG_FIFO_WE_SLAVE_RATIO value is used. When read DQS gate training is performed REG_PHY_GATELVL_INIT_RATIO is used and PHY_REG_FIFO_WE_SLAVE_RATIO is don't care.

NOTE: Software leveling is not recommended to be used. Hardware leveling must be used instead.

10.2.4.9 EMIF Access Cycles

By default, the EMIF keeps its SDRAM $\overline{CS}$ signal high (CS is active-low). To direct a command to only one of the SDRAMs, EMIF asserts its $\overline{CS}$ signal to the SDRAM for the duration of the command.

The EMIF always performs burst accesses to the SDRAM. Multiple SDRAM bursts may need to service a single local burst request. Table 10-14 through Table 10-19 show a few examples how EMIF performs SDRAM accesses for a linear incrementing transaction type. T0, T1, etc. are clock cycles. R0 is read starting at column 0, R8 is read starting at column 8, and R16 is read starting at column 16. D0-1 is the data from column 0 and 1, D2-3 is the data from column 2 and 3, and so on.

Table 10-14. 64-Byte Linear Read Starting at Address 0x0 (All DDR)

T0	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11
R0				R8							
				D0-1	D2-3	D4-5	D6-7	D8-9	D10-11	D12-13	D14-15

Table 10-15. 64-Byte Linear Read Starting at Address 0x8 (LPDDR2-S2)

T0	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12	T13	T14
R2			R8				R16							
				D2-3	D4-5	D6-7	D8-9	D10-11	D12-13	D14-15	D16-17	Unused	Unused	Unused

Table 10-16. 64-Byte Linear Read Starting at Address 0x8 (DDR2)

T0	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12	T13	T14	T15
R2		R4		R8				R16							
				D2-3	Unused	D4-5	D6-7	D8-9	D10-11	D12-13	D14-15	D16-17	Unused	Unused	Unused

Table 10-17. 64-Byte Linear Read Starting at Address 0x8 (LPDDR2-S4)

T0	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12	T13	T14	T15
R2				R8				R16							
				D2-3	D4-5	D6-7	Unused	D8-9	D10-11	D12-13	D14-15	D16-17	Unused	Unused	Unused

Table 10-18. 64-Byte Linear Read Starting at Address 0x10 (All DDR)

T0	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12	T13
R4		R8				R16							
				D4-5	D6-7	D8-9	D10-11	D12-13	D14-15	D16-17	D18-19	Unused	Unused

Table 10-19. 64-Byte Linear Read Starting at Address 0x18 (All DDR)

T0	T1	T2	T3	T4	T5	T6	T7	T8	T9	T10	T11	T12	T13
R6		R8				R16							
				D6-7	Unused	D8-9	D10-11	D12-13	D14-15	D16-17	D18-19	D20-21	Unused

The EMIF uses the unused data phases in the preceding figures by issuing successive read commands if there are reads to open banks pending in the command FIFO.

The write data conversion from SDR to DDR is done outside the EMIF.

10.2.4.10 Turnaround Time

Table 10-20 lists the turnaround time that EMIF introduces on the data bus for various back-to-back accesses. The EMIF takes advantage of the CAS latencies and packs the commands as close as possible on the control bus to introduce the following turnaround time on the data bus.

Table 10-20. Turnaround Time

Current Access	Next Access	Turnaround Time (Number of DDR Clock Cycles)
SDRAM write	SDRAM read	EMIF_SDRAM_TIMING_1[2:0] T_WTR + 1 + CL
SDRAM read	SDRAM write	EMIF_SDRAM_TIMING_1[31:29] T_RTW + 1

10.2.4.11 PHY DLL Calibration

When running in normal locked mode, the PHY DLL gets a reference clock (EMIF_DLL_FCLK) from the PRCM, which is used by the DLL master to lock to the right frequency and provide the control code for a full period phase shift to the slave. The slave uses this code as a control for its internal delay line to produce the required delay for the signal considered.

When working in locked mode, the delay lines only get an updated control value from the master DLLs when an explicit dll_calib command is issued by the EMIF controller. Failure to send such commands on a timely basis will result in inaccurate delay-line information if there is a significant voltage and temperature drift in the system. It is recommended to issue at least one command every 100 µs. EMIF automatically sends ctrl_update commands for:

- Refresh Exit
- Self Refresh Exit
- phy_ready asserted during initialization

The PHY also internally generates a control value update upon completion of a leveling operation. Control is also added when leveling is not used and there are gradual voltage changes during frequency change. The EMIF_DLL_CALIB_CTRL register can be programmed to generate a phy_dll_calib for a periodic interval based on EMIF_FICLK cycles, so allow continued memory access as voltage is changing. A safe window of no activity will be guaranteed for this periodic generation of phy_dll_calib. In addition, a one shot generator for phy_pll_calib has also been added that will generate a single phy_dll_calib by setting the EMIF_MISC_REG[0] DLL_CALIB_OS bit to 1.

10.2.4.12 SDRAM Address Mapping

From the system point of view, the external SDRAM is seen as one block of SDRAM. If two external 64-MiB devices are used, a 128-MiB memory block is observed. If two external 32-MiB devices are used, a 64-MiB block is observed.

Table 10-21 shows the SDRAM address space.

Table 10-21. SDRAM Addressing Space

Module Name	Base Address
EMIF-CSN0-SDRAM	0x8000 0000

When addressing SDRAM, if the EMIF_SDRAM_CONFIG[28:27] IBANK_POS bit field is set to 0, the EMIF uses the following two bit fields to determine the mapping from the source address to the SDRAM row, column and bank:

- EMIF_SDRAM_CONFIG[6:4] IBANK
- EMIF_SDRAM_CONFIG[2:0] PAGESIZE

If the EMIF_SDRAM_CONFIG[28:27] IBANK_POS bit field is set to 1, 2, or 3, the EMIF uses the following three bit fields to determine the mapping from the source address to the SDRAM row, column and bank:

- EMIF_SDRAM_CONFIG[6:4] IBANK
- EMIF_SDRAM_CONFIG[2:0] PAGESIZE
- EMIF_SDRAM_CONFIG[9:7] ROWSIZE

In all cases the EMIF considers its SDRAM address space to be a single logical block, regardless of the number of physical devices.

10.2.4.12.1 Address Mapping for IBANK_POS = 0 and EBANK_POS = 0

For EMIF_SDRAM_CONFIG[28:27] IBANK_POS = 0 and EMIF_SDRAM_CONFIG_2[27] EBANK_POS = 0, Table 10-22 lists which source address bits (MAddr) are mapped to the SDRAM row, column and bank bits for all combinations of IBANK and PAGESIZE.

Table 10-22. Local Address to SDRAM Address Mapping for IBANK_POS = 0 and EBANK_POS = 0

MAddr[31:N] N = 1 if 16-bit data bus width; N = 2 if 32-bit data bus width					
row address		bank address		column address	
ROWSIZE value	row width (bits)	IBANK value	bank[2:0] width (bits)	PAGESIZE value	col width (bits)
In this case the ROWSIZE bit field is not used	16	0	0	0	8
		1	1	1	9
		2	2	2	10
		3	3	3	11

NOTE: The ROWSIZE bit field is unused in case of IBANK_POS = 0 and EBANK_POS = 0.

For EMIF_SDRAM_CONFIG[28:27] IBANK_POS = 0, the effect of the address-mapping scheme is that as the source address increments across the SDRAM pages, EMIF moves to page with the same number as in the previous bank. This movement across the banks continues until the same page is accessed in all banks and then EMIF moves to the next page in the first bank. The EMIF uses this movement across internal banks while remaining on the same page to maximize the number of the open SDRAM banks within the overall SDRAM space.

Thus, the EMIF can keep a maximum of 8 banks open at a time, and can interleave among all of them.

10.2.4.12.2 Address Mapping for IBANK_POS = 1 and EBANK_POS = 0

Table 10-23 list the local address to SDRAM address mapping when IBANK_POS = 1 and EBANK_POS = 0.

Table 10-23. Local Address to SDRAM Address Mapping for IBANK_POS = 1 and EBANK_POS = 0

MAddr[31:N] N = 1 if 16-bit data bus width; N = 2 if 32-bit data bus width							
bank address[2]		row address		bank address[1:0]		column address	
IBANK value	bank[2] width (bits)	ROWSIZE value	row width (bits)	IBANK value	bank[1:0] width (bits)	PAGESIZE value	col width (bits)
0	0	0	9	0	0	0	8
1	0	1	10	1	1	1	9
2	0	2	11	2	2	2	10
3	1	3	12	3	2	3	11
		4	13				
		5	14				
		6	15				
		7	16				

For EMIF_SDRAM_CONFIG[28:27] IBANK_POS = 1, the EMIF interleaves banks the same as for EMIF_SDRAM_CONFIG[28:27] IBANK_POS = 0 but the interleaving of banks is limited to four banks. Thus, the EMIF can keep a maximum of 8 banks open at a time but can interleave among only 4 of them.

10.2.4.12.3 Address Mapping for IBANK_POS = 2 and EBANK_POS = 0

Table 10-24 list the local address to SDRAM address mapping when IBANK_POS = 2 and EBANK_POS = 0.

Table 10-24. Local Address to SDRAM Address Mapping for IBANK_POS = 2 and EBANK_POS = 0

MAddr[31:N] N = 1 if 16-bit data bus width; N = 2 if 32-bit data bus width							
bank address[2:1]		row address		bank address[0]		column address	
IBANK value	bank[2:1] width (bits)	ROWSIZE value	row width (bits)	IBANK value	bank[0] width (bits)	PAGESIZE value	col width (bits)
0	0	0	9	0	0	0	8
1	0	1	10	1	1	1	9
2	1	2	11	2	1	2	10
3	2	3	12	3	1	3	11
		4	13				
		5	14				
		6	15				
		7	16				

For EMIF_SDRAM_CONFIG[28:27] IBANK_POS = 2, the EMIF interleaves banks the same as for EMIF_SDRAM_CONFIG[28:27] IBANK_POS = 0 but the interleaving of banks is limited to two banks. Thus, the EMIF can keep a maximum of 8 banks open at a time but can interleave among only 2 of them.

10.2.4.12.4 Address Mapping for IBANK_POS = 3 and EBANK_POS = 0

Table 10-25 list the local address to SDRAM address mapping when IBANK_POS = 3 and EBANK_POS = 0.

Table 10-25. Local Address to SDRAM Address Mapping for IBANK_POS = 3 and EBANK_POS = 0

MAddr[31:N] N = 1 if 16-bit data bus width; N = 2 if 32-bit data bus width					
bank address		row address		column address	
IBANK value	bank width (bits)	ROWSIZE value	row width (bits)	PAGESIZE value	col width (bits)
0	0	0	9	0	8
1	1	1	10	1	9
2	2	2	11	2	10
3	3	3	12	3	11
		4	13		
		5	14		
		6	15		
		7	16		

If EMIF_SDRAM_CONFIG[28:27] IBANK_POS = 3, the bank interleaving is not possible for EMIF.

10.2.4.12.5 Address Mapping for IBANK_POS = 0 and EBANK_POS = 1

Table 10-26 list the local address to SDRAM address mapping when IBANK_POS = 0 and EBANK_POS = 1.

Table 10-26. Local Address to SDRAM Address Mapping for IBANK_POS = 0 and EBANK_POS = 1

MAddr[31:N] N = 1 if 16-bit data bus width; N = 2 if 32-bit data bus width					
row address		bank address		column address	
ROWSIZE value	row width (bits)	IBANK value	bank width (bits)	PAGESIZE value	col width (bits)
0	9	0	0	0	8
1	10	1	1	1	9
2	11	2	2	2	10
3	12	3	3	3	11
4	13				
5	14				
6	15				
7	16				

For EMIF_SDRAM_CONFIG[28:27] IBANK_POS = 0, and EMIF_SDRAM_CONFIG_2[27] EBANK_POS = 1, the EMIF can keep a maximum of 8 banks open at a time and can interleave among all of them.

10.2.4.12.6 Address Mapping for IBANK_POS = 1 and EBANK_POS = 1

Table 10-27 list the local address to SDRAM address mapping when IBANK_POS = 1 and EBANK_POS = 1.

Table 10-27. Local Address to SDRAM Address Mapping for IBANK_POS = 1 and EBANK_POS = 1

MAddr[31:N] N = 1 if 16-bit data bus width; N = 2 if 32-bit data bus width							
bank address[2]		row address		bank address[1:0]		column address	
IBANK value	bank[2] width (bits)	ROWSIZE value	row width (bits)	IBANK value	bank[1:0] width (bits)	PAGESIZE value	col width (bits)
0	0	0	9	0	0	0	8
1	0	1	10	1	1	1	9

Table 10-27. Local Address to SDRAM Address Mapping for IBANK_POS = 1 and EBANK_POS = 1 (continued)

MAddr[31:N] N = 1 if 16-bit data bus width; N = 2 if 32-bit data bus width							
bank address[2]		row address		bank address[1:0]		column address	
IBANK value	bank[2] width (bits)	ROWSIZE value	row width (bits)	IBANK value	bank[1:0] width (bits)	PAGESIZE value	col width (bits)
2	0	2	11	2	2	2	10
3	1	3	12	3	2	3	11
		4	13				
		5	14				
		6	15				
		7	16				

For EMIF_SDRAM_CONFIG[28:27] IBANK_POS = 1, and EMIF_SDRAM_CONFIG_2[27] EBANK_POS = 1, the EMIF can keep a maximum of 8 banks open at a time but can interleave among only 4 of them.

10.2.4.12.7 Address Mapping for IBANK_POS = 2 and EBANK_POS = 1

Table 10-28 list the local address to SDRAM address mapping when IBANK_POS = 2 and EBANK_POS = 1.

Table 10-28. Local Address to SDRAM Address Mapping for IBANK_POS = 2 and EBANK_POS = 1

MAddr[31:N] N = 1 if 16-bit data bus width; N = 2 if 32-bit data bus width							
bank address[2:1]		row address		bank address[0]		column address	
IBANK value	bank[2:1] width (bits)	ROWSIZE value	row width (bits)	IBANK value	bank[0] width (bits)	PAGESIZE value	col width (bits)
0	0	0	9	0	0	0	8
1	0	1	10	1	1	1	9
2	1	2	11	2	1	2	10
3	2	3	12	3	1	3	11
		4	13				
		5	14				
		6	15				
		7	16				

For EMIF_SDRAM_CONFIG[28:27] IBANK_POS = 2 and EMIF_SDRAM_CONFIG_2[27] EBANK_POS = 1, the EMIF can keep a maximum of 8 banks open at a time but can interleave among only 2 of them.

10.2.4.12.8 Address Mapping for IBANK_POS = 3 and EBANK_POS = 1

Table 10-29 list the local address to SDRAM address mapping when IBANK_POS = 3 and EBANK_POS = 1.

Table 10-29. Local Address to SDRAM Address Mapping for IBANK_POS = 3 and EBANK_POS = 1

MAddr[31:N] N = 1 if 16-bit data bus width; N = 2 if 32-bit data bus width					
bank address		row address		column address	
IBANK value	bank width (bits)	ROWSIZE value	row width (bits)	PAGESIZE value	col width (bits)
0	0	0	9	0	8
1	1	1	10	1	9
2	2	2	11	2	10
3	3	3	12	3	11
		4	13		

Table 10-29. Local Address to SDRAM Address Mapping for IBANK_POS = 3 and EBANK_POS = 1 (continued)

MAddr[31:N] N = 1 if 16-bit data bus width; N = 2 if 32-bit data bus width					
bank address		row address		column address	
IBANK value	bank width (bits)	ROWSIZE value	row width (bits)	PAGESIZE value	col width (bits)
		5	14		
		6	15		
		7	16		

For EMIF_SDRAM_CONFIG[28:27] IBANK_POS = 3 and EMIF_SDRAM_CONFIG_2[27] EBANK_POS = 1, the bank interleaving is not possible for EMIF.

10.2.4.13 Output Impedance Calibration

The EMIF controller supports automatic output impedance (ZQ) calibration for LPDDR2 and DDR3/DDR3L memories. This feature is not supported by DDR2 memories. The ZQ calibration can be enabled by setting to 0x1 the EMIF_SDRAM_OUTPUT_IMPEDANCE_CALIBRATION_CONFIG[30] ZQ_CS0EN bit. The EMIF supports three types of ZQ calibration commands:

- ZQINIT: ZQ calibration command during initialization
- ZQCL: ZQ calibration long command
- ZQCS: ZQ calibration short command

For DDR3/DDR3L, the EMIF automatically issues ZQINIT command during initialization. For LPDDR2, software must issue a ZQINIT command (MRW to mode register 10). The EMIF issues ZQCS command each time the EMIF_SDRAM_OUTPUT_IMPEDANCE_CALIBRATION_CONFIG[15:0] ZQ_REFINTERVAL bit field expires. In other words, the ZQ_REFINTERVAL defines the interval between two ZQCS commands. When ZQCS command is issued, the EMIF waits and blocks any other command for EMIF_SDRAM_TIMING_3[20:15] ZQ_ZQCS + 1 number of DDR clock cycles.

If the EMIF_SDRAM_OUTPUT_IMPEDANCE_CALIBRATION_CONFIG[28] ZQ_SFEXITEN bit field is set to 0x1, the EMIF issues ZQCL command every time it exits self-refresh, active power-down and precharge power-down modes. When ZQCL command is issued, the EMIF waits and blocks any other command for (EMIF_SDRAM_OUTPUT_IMPEDANCE_CALIBRATION_CONFIG[17:16] ZQ_ZQCL_MULT + 1) × (EMIF_SDRAM_TIMING_3[20:15] ZQ_ZQCS + 1) number of DDR clock cycles.

The ZQINIT is a non periodic command issued only once during DDR initialization as opposed to the ZQCL and ZQCS calibration commands which are issued by the EMIF periodically at regular intervals.

10.2.4.14 LPDDR2 Temperature Monitoring

NOTE: LPDDR2 is not supported on the DRA78x family of devices.

The EMIF supports automatic temperature monitoring to facilitate the software update of the refresh rate according to the temperature changes of the LPDDR2. Temperature monitoring can be enabled per chip select by setting the EMIF_TEMPERATURE_ALERT_CONFIG[31] TA_CS1EN and EMIF_TEMPERATURE_ALERT_CONFIG[30] TA_CS0EN bits.

The EMIF polls the temperature of the LPDDR2 (issues an MRR command to mode register 4) every time the EMIF_TEMPERATURE_ALERT_CONFIG[21:0] TA_REFINTERVAL bit field expires. In other words, TA_REFINTERVAL defines the interval between temperature alert polls. If the EMIF sees a 1 on bit 7 of the read data value from the MRR, which indicates the temperature has changed, the EMIF_SYSTEM_OCP_INTERRUPT_STATUS[1] TA_SYS bit is set and the module sends an interrupt. When the interrupt is received, software must update the EMIF_SDRAM_REFRESH_CONTROL[15:0] REFRESH_RATE bit field to the required value as per the temperature change.

If the EMIF_TEMPERATURE_ALERT_CONFIG[28] TA_SFEXITEN bit is set to 1, the EMIF polls for temperature change every time it exits self-refresh, active power-down, and precharge power-down modes.

Because the EMIF is performing a LPDDR2 mode register read, it needs information about how the LPDDR2 is connected. The EMIF_TEMPERATURE_ALERT_CONFIG[27:26] TA_DEVWDT and EMIF_TEMPERATURE_ALERT_CONFIG[25:24] TA_DEVCNT bit fields provide the necessary information to the EMIF for MRR data compare. For example, if TA_DEVWDT is set to 0, which indicates 8-bit devices are used, and if TA_DEVCNT is set to 2, which indicates four devices are used to form a 32-bit bus, the mask used for checking is 0b1111, that is, the EMIF expects data on each byte lane on a 32-bit bus.

10.2.4.15 Error Correction And Detection Feature

For data integrity, the EMIF supports ECC on the data written or read from the SDRAM and is enabled by programming the EMIF_ECC_CTRL_REG register. 7-bit ECC is calculated over 32-bit data when in 32-bit DDR mode. 6-bit ECC is calculated over 16-bit data when in 16-bit DDR mode. The ECC is calculated for all accesses that are within the address ranges protected by ECC. The address ranges are specified in the EMIF_ECC_ADDRESS_RANGE_1 and EMIF_ECC_ADDRESS_RANGE_2 registers. Both registers have identical bits and functionality. This provides flexibility allowing two non-overlapping memory regions to be ECC protected.

NOTE: For LPDDR2 ECC is supported only in 16-bit narrow mode.

The system must ensure that any burst access with starting address in the ECC protected region must not cross over to the un-protected region and vice-versa. The ECC is stored inside the SDRAM during writes. If a write access with byte count that is not a multiple of ECC quanta or with a non quanta aligned address is performed within the address range protected by ECC, the EMIF will send out a write ECC error interrupt. The EMIF will also report an error on the system response interface. In this case, the EMIF will perform the write to the SDRAM. However, the ECC value written to the SDRAM will be corrupted. The EMIF will also log the MConnID, MCmd and MBurstSeq for the first error transaction in the EMIF_OCP_ERROR_LOG register.

The ECC quanta is either 32-bit or 16-bit based on the EMIF_SDRAM_CONFIG[15:14] NARROW_MODE bit field. For 32-bit mode (128 bits per EMIF clock cycle), an ECC quanta is 32 bits. For 16-bit narrow mode (64 bits per EMIF clock cycle), the ECC quanta is 16 bits.

Once ECC is enabled the entire protected region must be initialized with data. These writes must be quanta-sized and quanta-aligned.

The ECC is read and verified during reads. For 1-bit ECC error in the data, the EMIF will correct the data and send it on the system return interface. The EMIF will log the starting address of the SDRAM burst in an internal 4 deep address FIFO. The internal FIFO will store the first four 1-bit ECC errors. The EMIF_1B_ECC_ERR_ADDR_LOG register will display the address on top of the internal FIFO. The software must write a 0x1 to the EMIF_1B_ECC_ERR_ADDR_LOG register to pop the FIFO and display the next address stored. For subsequent reads in the ECC regions, the FIFO will be loaded with the address for the next 1-bit ECC error if it is not full. It must be noted that no address comparison will be performed, that is, if a single address has ECC errors back-to-back, that address will be logged twice.

The number of 1-bit ECC errors can be counted using the EMIF_1B_ECC_ERR_CNT register. The EMIF also supports programming a threshold and a window in the EMIF_1B_ECC_ERR_THRSH register. The window is programmed in number of refresh periods. When the programmed window value is 0x0, that is, window is disabled, and the internal error count meets the programmed threshold, the EMIF will generate a 1-bit ECC error interrupt. When the programmed window value is non-zero, that is, window is enabled, the EMIF will generate a 1-bit ECC error interrupt only if the internal error count meets the programmed threshold in that window. The internal error count is reset every time the window expires. The software can use this to gauge the degree of 1-bit ECC errors occurring in the system.

For diagnosis, the EMIF supports a 1-Bit ECC data error distribution register (EMIF_1B_ECC_ERR_DIST_1) that represent whether an error has occurred in a given data channel location. This is advantageous to detect whether the errors are random or systemic. The distribution registers will be overlay of all 1-bit ECC errors until the software clears the register. Therefore, multiple bits could be set as a result of multiple 1-bit ECC errors occurring over multiple read accesses.

For 2-bit ECC errors in the data, the EMIF will generate a 2-bit ECC error interrupt. For any bit errors in the address, the EMIF will generate an address error interrupt. It must be noted that the EMIF will neither correct the data for these uncorrectable errors. Along with generating the interrupts, the EMIF will also report an error on the system return interface. In this case the EMIF will send the resultant data from the ECC correction logic. The read data received from the memory may have further been corrupted by the ECC correction logic since it will have attempted to correct the read data but failed due to uncorrectable error.

For all uncorrectable ECC errors listed above, the EMIF will log the starting address of the SDRAM burst in the EMIF_2B_ECC_ERR_ADDR_LOG register. This register will show the address of the first uncorrectable error. After the software clears the register, it will be loaded with the address for the next uncorrectable error.

In the event that the EMIF detects a single bit ECC error, although the error is corrected on the returned data, the data in the SDRAM is not corrected. It is the responsibility of the system software to correct the ECC error at that location. To the extent possible, the system software should correct multiple bit errors with the caveat that the returned data was not corrected but corrupted by the ECC correction logic.

10.2.4.16 Class of Service

The commands in the Command FIFO can be mapped to two classes of service namely 1 and 2. The mapping of commands to a particular class of service can be done based on the priority or the master ID. The mapping based on priority can be done by setting the appropriate values in the EMIF_PRIORITY_TO_CLASS_OF_SERVICE_MAPPING register. The mapping based on master ID can be done by setting the appropriate values of master ID and the masks in the EMIF_CONNECTION_ID_TO_CLASS_OF_SERVICE_1_MAPPING and EMIF_CONNECTION_ID_TO_CLASS_OF_SERVICE_2_MAPPING registers. There are 3 master ID and mask values that can be set for each class of service. In conjunction with the masks, each class of service can have a maximum of 144 master IDs mapped to it. For example, a master ID value of 0xFF along with a mask value of 0x3 will map all master IDs from 0xF8 to 0xFF to that particular class of service.

Each class of service has an associated latency counter (EMIF_COS_CONFIG[23:16] COS_COUNT_1 and EMIF_COS_CONFIG[15:8] COS_COUNT_2). When the latency counter for a command expires, that is, reaches the value programmed for the class of service that the command belongs to, that command is executed next. If there is more than one command that has expired latency counter, the command with the highest priority is executed first. One exception to this rule is, if the EMIF_COS_CONFIG[7:0] PR_OLD_COUNT value expires for the oldest command in the queue. That command is executed first irrespective of priority or class of service. This is done to prevent the continuous blocking effect.

The EMIF_COS_CONFIG[7:0] PR_OLD_COUNT value is used to identify when the oldest command in the command FIFO has timed out. At this point during the arbitration process, this oldest command is issued regardless of the priority of the other commands in the FIFO. This feature is disabled when writing 0x0 to the EMIF_COS_CONFIG[7:0] PR_OLD_COUNT bit field. After issuing the oldest command, the other remaining commands in the FIFO are reordered by age. The next oldest command in the FIFO is given highest priority again and issued after the EMIF_COS_CONFIG[7:0] PR_OLD_COUNT value expires. If a new value in the PR_OLD_COUNT bit field is written during counting, that is, before PR_OLD_COUNT expires, the counter keeps working but if this value is smaller the oldest command is issued sooner and if this value is larger the oldest command is issued later.

The master ID mapping allows the same master ID to be put in both class of service 1 and 2. Also, a transaction might belong to one class of service if viewed by master ID and might belong to another class of service if viewed by priority. In these cases, the command will belong to both class of service. The EMIF will try executing the command as soon as possible, when the smaller of the two counters (EMIF_COS_CONFIG[23:16] COS_COUNT_1 and EMIF_COS_CONFIG[15:8] COS_COUNT_2) expires.

10.2.4.17 Performance Counters

The EMIF_PERFORMANCE_COUNTER_1 and EMIF_PERFORMANCE_COUNTER_2 registers are used to monitor or calculate the EMIF Controller bandwidth and efficiency. These counters are able to count events such as accesses made to EMIF, Activate (ACT) commands sent to SDRAM, read and write accesses made to EMIF, and other events. Each counter counts independently of the other. In addition to the ability of events counting, the counters can also filter the events from a particular master or address space. The events counting and filter enabling are configured using the EMIF_PERFORMANCE_COUNTER_CONFIG register. The filter value used is configured through the EMIF_PERFORMANCE_COUNTER_MASTER_REGION_SELECT register. Each counter can be configured independently.

Table 10-30 lists all the events that can be counted and whether a filter can be applied to a particular event. A filter is applied to an event if the following bits are set to 0x1 for that event:

- For Performance Counter 1: EMIF_PERFORMANCE_COUNTER_CONFIG[15] CNTR1_MCONNID_EN and EMIF_PERFORMANCE_COUNTER_CONFIG[14] CNTR1_REGION_EN;
- For Performance Counter 2: EMIF_PERFORMANCE_COUNTER_CONFIG[31] CNTR2_MCONNID_EN and EMIF_PERFORMANCE_COUNTER_CONFIG[30] CNTR2_REGION_EN.

Table 10-30. Performance Counter Filter Configuration

CNTRn_CFG[1]	CNTRn_REGION_EN	CNTRn_MCONNID_EN	Description
0x0	0x0	0x0 or 0x1	Count the accesses made to EMIF
0x1	0x0	0x0 or 0x1	Count the Activate (ACT) commands sent to SDRAM
0x2	0x0 or 0x1	0x0 or 0x1	Count the read accesses made to EMIF
0x3	0x0 or 0x1	0x0 or 0x1	Count the write accesses made to EMIF
0x4	0x0	0x0	Count number of EMIF_FICLK clock cycles during which the local Command FIFO is full
0x5	0x0	0x0	Count number of EMIF_FICLK clock cycles during which the local Write Data FIFO is full
0x6	0x0	0x0	Count number of EMIF_FICLK clock cycles during which the local Read Data FIFO is full
0x7	0x0	0x0	Count number of EMIF_FICLK clock cycles during which the local Return Command FIFO is full
0x8	0x0 or 0x1	0x0 or 0x1	Count number of priority elevations
0x9	0x0	0x0	Count number of EMIF_FICLK clock cycles that a command was pending
0xA	0x0	0x0	Count number of EMIF_FICLK cycles used by the EMIF controller for reads and writes.
0xB - 0xF	0x0	0x0	Reserved for future use.

(1) n = 1 or 2

NOTE: When the MReqDebug qualifier is set to 0x1 for a particular local command, the performance counters are not incremented for that particular command if the CNTRn_CFG values are equal to 0x0, 0x1, 0x2, 0x3, or 0xA.

NOTE: The EMIF performance counters cannot distinguish between single access and burst access. In both cases they are incremented by 1. If the actual SDRAM bandwidth of an initiator has to be measured the EMIF performance counters may not be sufficient.

10.2.4.17.1 Performance Counters General Examples

- **General Example for Counting All Write Accesses made to EMIF**
  If the EMIF_PERFORMANCE_COUNTER_1 register is used to count all write accesses made to EMIF from master with connection ID equal to 0x86 (this is the system MMU) the following steps should be performed:

  - To enable the writes counting, the EMIF_PERFORMANCE_COUNTER_CONFIG[3:0] CNTR1_CFG bit field must be set to 0x3.
  - The EMIF_PERFORMANCE_COUNTER_MASTER_REGION_SELECT[15:8] MCONNID1 bit field must be set to 0x86.
  - To enable filtering, the EMIF_PERFORMANCE_COUNTER_CONFIG[15] CNTR1_MCONNID_EN bit must be set to 0x1.

    With this configuration EMIF_PERFORMANCE_COUNTER_1 counts every write made to the EMIF from master 0x86 to any address space. This does not include accesses from other masters and commands other than writes.

- **General Example for Counting Total Accesses made to EMIF**
  If the EMIF_PERFORMANCE_COUNTER_2 register is used to count total accesses made to EMIF regardless of the address space or master the following steps should be performed:
  - To enable counting of all accesses to the SDRAM, the EMIF_PERFORMANCE_COUNTER_CONFIG[19:16] CNTR2_CFG bit field must be set to 0x0.
  - To disable filtering, both the EMIF_PERFORMANCE_COUNTER_CONFIG[31] CNTR2_MCONNID_EN and EMIF_PERFORMANCE_COUNTER_CONFIG[30] CNTR2_REGION_EN bits must be set to 0x0.

    With this configuration EMIF_PERFORMANCE_COUNTER_2 counts every access made to the EMIF. This includes all accesses from all masters and to any address space.
- **General Example for Counting All Read Accesses made to EMIF**
  If the EMIF_PERFORMANCE_COUNTER_1 register is used to count all read accesses made to EMIF from master with connection ID equal to 0x86 (this is the system MMU) to address space 0x0 the following steps should be performed:
  - To enable the reads counting, the EMIF_PERFORMANCE_COUNTER_CONFIG[3:0] CNTR1_CFG bit field must be set to 0x2.
  - The EMIF_PERFORMANCE_COUNTER_MASTER_REGION_SELECT[15:8] MCONNID1bit field must be set to 0x86
  - The EMIF_PERFORMANCE_COUNTER_MASTER_REGION_SELECT[1:0] REGION_SEL1 bit field must be set to 0x0.
  - To enable filtering, both the EMIF_PERFORMANCE_COUNTER_CONFIG[15] CNTR1_MCONNID_EN and the EMIF_PERFORMANCE_COUNTER_CONFIG[14] CNTR1_REGION_EN bits must be set to 0x1.

    With this configuration, EMIF_PERFORMANCE_COUNTER_1 counts every read made to the EMIF from master 0x86 to address space 0x0. This does not include accesses from other masters or to other address spaces and does not include commands other than reads.

10.2.4.18 Forcing CKE to tri-state

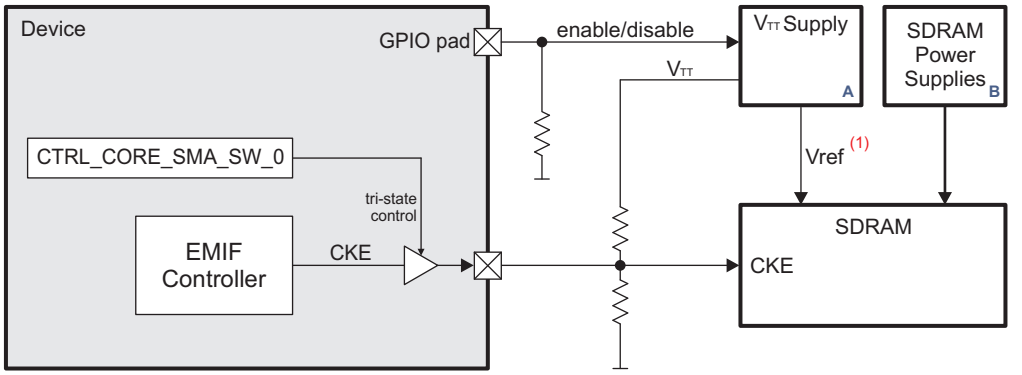
The CKE pad can be forced to tri-state when the CTRL_CORE_SMA_SW_0[0] CKE_ASSERTION bit is set to 0x1. This functionality facilitates fast resume by allowing a strong external pull-down resistor to hold the CKE memory pad low thus keeping the SDRAM in self-refresh while the device is completely powered off. Figure 10-10 and the following example sequence provide more details how this functionality can be used:

- When device is running configure EMIF for self refresh. As a result it drives CKE low.
- Disable V_{TT} supply (can be controlled through a device GPIO pad).
- Ramp down all power rails to the device. When power is already off, EMIF doesn't drive CKE low. The external pull-down does this instead. Note that only V_{TT} must be off. All SDRAM supplies (including V_{REF}) must be on.
- After these three steps the device is completely powered off and the SDRAM is in self-refresh. The waking-up from this state is system specific.
- To resume from self-refresh the CKE pad must be forced to tri-state by writing 0x1 to one of the CKE_ASSERTION bits previously mentioned. Then EMIF must be configured for self-refresh so that its state becomes consistent with the actual SDRAM state.
- Write 0x0 to the corresponding CKE_ASSERTION bit to release CKE driver from tri-state. This allows

EMIF to take control over CKE which is now driven low as EMIF has already been configured for self-refresh.

- Enable V_{TT} supply.
- Access the SDRAM to bring it out of self refresh and resume application.

Figure 10-10. Example for Using the CKE Tri-state Functionality



NOTE: In case the device is powered off but the SDRAM is in self-refresh (as previously described) it must be taken into account that the SDRAM RESET# signal has to be controlled externally to preserve the SDRAM contents. This is needed as ddr1_rst signal is not controlled by the CTRL_CORE_SMA_SW_0 register and therefore cannot be used.

10.2.5 EMIF Programming Guide

10.2.5.1 EMIF Low-Level Programming Models

This section describers the low-level hardware programming sequences for the configuration and use of the EMIF module.

10.2.5.1.1 Global Initialization

Table 10-31. Global Initialization of Surrounding Modules

Surrounding Modules	Comments
PRCM	The module interface and functional clocks must be enabled. See Chapter 3, *Power, Reset, and Clock Management*.
IRQ_CROSSBAR	IRQ_CROSSBAR configuration must be done to allow module IRQs to be mapped to certain device INTC line. For more information see Section 13.4.6.4, *IRQ_CROSSBAR Module Functional Description*, in Chapter 13, *Control Module*
Device INTCs	Device INTCs must be configured to enable the interrupt request generation. For more information see Chapter 12 *Interrupt Controllers*

10.2.5.1.1.1 EMIF Configuration Sequence

Table 10-32 shows all steps needed to configure and use the EMIF.

Table 10-32. EMIF Configuration Sequence

Step	Register/ Bit Field	Value
Configure DPLL_DDR to the required frequency:		
· **IF** DPLL_DDR is locked:	CM_IDLEST_DPLL_DDR[0] ST_DPLL_CLK	0x1
Unlock DPLL_DDR	CM_CLKMODE_DPLL_DDR[2:0] DPLL_EN	0x6
· **ENDIF**		
· Configure the DPLL multiplier and divider factors	CM_CLKSEL_DPLL_DDR[18:8] DPLL_MULT and CM_CLKSEL_DPLL_DDR[6:0] DPLL_DIV	0x-
· Configure the M2 post-divider factor	CM_DIV_M2_DPLL_DDR[4:0] DIVHS	0x-
· Configure the H11 post-divider factor	CM_DIV_H11_DPLL_DDR[5:0] DIVHS	0x-
· Lock the DPLL_DDR	CM_CLKMODE_DPLL_DDR[2:0] DPLL_EN	0x7
Disable DLL Override	CM_DLL_CTRL[0] DLL_OVERRIDE	0x0
Configure the output impedance, slew rate and weak pull resistors of the DDR IO cells. For more information, see *Software Controls for the DDR3 I/O Cells*.	CTRL_CORE_CONTROL_DDRCACH1_0	0x-
	CTRL_CORE_CONTROL_DDRCH1_0	
	CTRL_CORE_CONTROL_DDRCH1_1	
	CTRL_CORE_CONTROL_DDRCH1_2	
If needed, configure the Vref-Generation Cells. For more information, see *Reference Voltage for the Device DDR3 Receivers*.	CTRL_CORE_CONTROL_DDRIO_0	0x-
Set the number of DQ samples required for read leveling	CTRL_WKUP_EMIF1_SDRAM_CONFIG_EXT[15:14] EMIF1_REG_PHY_NUM_OF_SAMPLES	0x3
Choose SDRAM read response on only one DQ bit during read leveling	CTRL_WKUP_EMIF1_SDRAM_CONFIG_EXT[12] EMIF1_REG_PHY_ALL_DQ_MPR_RD_RESP	0x0
Configure ODT for the device DDR IOs	CTRL_WKUP_EMIF1_SDRAM_CONFIG_EXT [6:5] EMIF1_PHY_RD_LOCAL_ODT	0x1 for 60 Ohms
		0x2 for 80 Ohms
		0x3 for 120 Ohms

Table 10-32. EMIF Configuration Sequence (continued)

Step	Register/ Bit Field	Value
IF ECC is used:		
Enable ECC	CTRL_WKUP_EMIF1_SDRAM_CONFIG_EXT[16] EMIF1_EN_ECC	0x1
ENDIF		
Based on the memory type other bits from CTRL_WKUP_EMIF1_SDRAM_CONFIG_EXT can also be configured.		
Read the EMIF_IODFT_TLGC register	EMIF_IODFT_TLGC	RD_VAL_1
Read the EMIF_TEMPERATURE_ALERT_CONFIG register	EMIF_TEMPERATURE_ALERT_CONFIG	RD_VAL_2
Read the EMIF_DDR_PHY_CONTROL_2 register	EMIF_DDR_PHY_CONTROL_2	RD_VAL_3
IF External warm reset or global warm software reset has occurred:	PRM_RSTST[5]EXTERNAL_WARM_RST or PRM_RSTST[1] GLOBAL_WARM_SW_RST	0x1 0x1
Reset the DDR PHY	EMIF_IODFT_TLGC[10] RESET_PHY	0x1
ENDIF		
Program the necessary ratio values to the EMIF_EXT_PHY_CONTROL_1 register	EMIF_EXT_PHY_CONTROL_1[29:20] PHY_REG_CTRL_SLAVE_RATIO2 (LPDDR2 only)	0x80 if PHY_INVERT_CLKOUT = 0x0 or 0x100 if PHY_INVERT_CLKOUT = 0x1
	EMIF_EXT_PHY_CONTROL_1[19:10] PHY_REG_CTRL_SLAVE_RATIO1 (DDR2/DDR3 only)	0x80 if PHY_INVERT_CLKOUT = 0x0 or 0x100 if PHY_INVERT_CLKOUT = 0x1
	EMIF_EXT_PHY_CONTROL_1[9:0] PHY_REG_CTRL_SLAVE_RATIO0 (DDR2/DDR3 only)	0x80 if PHY_INVERT_CLKOUT = 0x0 or 0x100 if PHY_INVERT_CLKOUT = 0x1
Program the shadow register of EMIF_EXT_PHY_CONTROL_1	EMIF_EXT_PHY_CONTROL_1_SHADOW	EMIF_EXT_PHY_CONTROL_1
If software leveling will be used, program registers EMIF_EXT_PHY_CONTROL_2 through EMIF_EXT_PHY_CONTROL_21 and their corresponding shadow registers **NOTE: Software leveling is not recommended to be used. Hardware leveling must be used instead.**	EMIF_EXT_PHY_CONTROL_2/EMIF_EXT_PHY_CONTROL_2_SHADOW through EMIF_EXT_PHY_CONTROL_21/EMIF_EXT_PHY_CONTROL_21_SHADOW	0x-

Table 10-32. EMIF Configuration Sequence (continued)

Step	Register/ Bit Field	Value
Program the necessary delay values to the EMIF_EXT_PHY_CONTROL_22 register	EMIF_EXT_PHY_CONTROL_22 [24:16] PHY_REG_FIFO_WE_IN_DELAY (1) (2)	Recommended value is 0x0
	EMIF_EXT_PHY_CONTROL_22 [8:0] PHY_REG_CTRL_SLAVE_DELAY (1) (2)	Recommended value is 0x0
Program the shadow register of EMIF_EXT_PHY_CONTROL_22	EMIF_EXT_PHY_CONTROL_22_SHADOW (1) (2)	EMIF_EXT_PHY_CONTROL_22
Program the necessary delay values to the EMIF_EXT_PHY_CONTROL_23 register	EMIF_EXT_PHY_CONTROL_23[24:16] PHY_REG_WR_DQS_SLAVE_DELAY (1) (2)	Recommended value is 0x0
	EMIF_EXT_PHY_CONTROL_23[8:0] PHY_REG_RD_DQS_SLAVE_DELAY (1) (2)	Recommended value is 0x0
Program the shadow register of EMIF_EXT_PHY_CONTROL_23	EMIF_EXT_PHY_CONTROL_23_SHADOW (1) (2)	EMIF_EXT_PHY_CONTROL_23
Program the EMIF_EXT_PHY_CONTROL_24 register	EMIF_EXT_PHY_CONTROL_24[30:24] REG_PHY_DQ_OFFSET_HI (3)	0x-
	EMIF_EXT_PHY_CONTROL_24[16] REG_PHY_GATELVL_INIT_MODE	Recommended value is 0x1
	EMIF_EXT_PHY_CONTROL_24[12] REG_PHY_USE_RANK0_DELAYS(2)	0x1
	EMIF_EXT_PHY_CONTROL_24[8:0] REG_PHY_WR_DATA_SLAVE_DELAY (1)(2)	Recommended value is 0x0
Program the shadow register of EMIF_EXT_PHY_CONTROL_24	EMIF_EXT_PHY_CONTROL_24_SHADOW	EMIF_EXT_PHY_CONTROL_24
Program the necessary offset ratio values to the EMIF_EXT_PHY_CONTROL_25 register	EMIF_EXT_PHY_CONTROL_25[27:21] REG_PHY_DQ_OFFSET3 (3)	0x-
	EMIF_EXT_PHY_CONTROL_25[20:14] REG_PHY_DQ_OFFSET2 (3)	0x-
	EMIF_EXT_PHY_CONTROL_25[13:7] REG_PHY_DQ_OFFSET1 (3)	0x-
	EMIF_EXT_PHY_CONTROL_25[6:0] REG_PHY_DQ_OFFSET0 (3)	0x-
Program the shadow register of EMIF_EXT_PHY_CONTROL_25	EMIF_EXT_PHY_CONTROL_25_SHADOW	EMIF_EXT_PHY_CONTROL_25
If hardware leveling (read-write leveling) will be used, program with zeros registers EMIF_EXT_PHY_CONTROL_26 through EMIF_EXT_PHY_CONTROL_35 and their corresponding shadow registers	EMIF_EXT_PHY_CONTROL_26/ EMIF_EXT_PHY_CONTROL_26_SHADOW through EMIF_EXT_PHY_CONTROL_35/ EMIF_EXT_PHY_CONTROL_35_SHADOW	0x0
Program the EMIF_EXT_PHY_CONTROL_36 register	EMIF_EXT_PHY_CONTROL_36	0x-
Program the shadow register of EMIF_EXT_PHY_CONTROL_36	EMIF_EXT_PHY_CONTROL_36_SHADOW	EMIF_EXT_PHY_CONTROL_36
Define the SDRAM refresh rate in the shadow register of EMIF_SDRAM_REFRESH_CONTROL	EMIF_SDRAM_REFRESH_CONTROL_SHADOW [15:0] REFRESH_RATE_SHDW	0x-

(1) These values are only used when CM_DLL_CTRL[0] DLL_OVERRIDE is set to 0x1. This is used only for debug purposes.
(2) The values of these fields do not depend on the board topology.
(3) These fields control the delay between the corresponding DQ byte lane and DQS/DQSN pair. Their values should correspond to delay of 1/4 clock cycle to center the DQ relative to the rising/falling edges of DQS/DQSN. Regardless of topology, it is recommended that all the DQ bits are skew matched to the corresponding DQS/DQSN pair.

Table 10-32. EMIF Configuration Sequence (continued)

Step	Register/ Bit Field	Value
Define the SDRAM refresh rate	EMIF_SDRAM_REFRESH_CONTROL[15:0] REFRESH_RATE	REFRESH_RATE_SHDW
Disable SDRAM initialization and refreshes	EMIF_SDRAM_REFRESH_CONTROL[31] INITREF_DIS	0x1
Configure the timing parameters in EMIF_SDRAM_TIMING_1 (4)		0x-
Program the shadow register of EMIF_SDRAM_TIMING_1	EMIF_SDRAM_TIMING_1_SHADOW (4)	EMIF_SDRAM_TIMING_1
Configure the timing parameters in EMIF_SDRAM_TIMING_2 (4)		0x-
Program the shadow register of EMIF_SDRAM_TIMING_2	EMIF_SDRAM_TIMING_2_SHADOW (4)	EMIF_SDRAM_TIMING_2
Configure the timing parameters in EMIF_SDRAM_TIMING_3 (4)		0x-
Program the shadow register of EMIF_SDRAM_TIMING_3	EMIF_SDRAM_TIMING_3_SHADOW (4)	EMIF_SDRAM_TIMING_3
Program the EMIF_LPDDR2_NVM_TIMING (5) and EMIF_LPDDR2_NVM_TIMING_SHADOW (5) registers. **NOTE: These registers are not supported. They are kept only for code compatibility.**		0x0
Disable automatic power management	EMIF_POWER_MANAGEMENT_CONTROL[10:8] LP_MODE	0x0
Define the number of DDR clock cycles after which the EMIF puts the external SDRAM in Power Down mode when EMIF is idle	EMIF_POWER_MANAGEMENT_CONTROL[15:12] PD_TIM	0x-
Define the number of DDR clock cycles after which the EMIF puts the external SDRAM in Self Refresh mode when EMIF is idle	EMIF_POWER_MANAGEMENT_CONTROL[7:4] SR_TIM	0x-
Program the shadow register of EMIF_POWER_MANAGEMENT_CONTROL	EMIF_POWER_MANAGEMENT_CONTROL_SHADOW	EMIF_POWER_MANAGEMENT_CONTROL
Configure the System and MPU maximum number of commands in the command FIFO	EMIF_OCP_CONFIG[27:24] SYS_THRESH_MAX	0x-
	EMIF_OCP_CONFIG[23:20] MPU_THRESH_MAX	0x-
Program the EMIF_IODFT_TLGC register	EMIF_IODFT_TLGC	RD_VAL_1
Determine the required wait time after a phy_dll_calib is generated before another command can be sent	EMIF_DLL_CALIB_CTRL[19:16] ACK_WAIT	0x-
Determine the interval between phy_dll_calib generation	EMIF_DLL_CALIB_CTRL[8:0] DLL_CALIB_INTERVAL	0x-
Program the shadow register of EMIF_DLL_CALIB_CTRL	EMIF_DLL_CALIB_CTRL_SHADOW	EMIF_DLL_CALIB_CTRL
Define the interval (number of refresh periods) between ZQCS commands	EMIF_SDRAM_OUTPUT_IMPEDANCE_CALIBRATION_CONFIG[15:0] ZQ_REFINTERVAL	0x-
Define the number of ZQCS intervals that build the ZQCL duration	EMIF_SDRAM_OUTPUT_IMPEDANCE_CALIBRATION_CONFIG[17:16] ZQ_ZQCL_MULT	0x-
Enable issuing of ZQCL on Self-Refresh, Active Power-Down, and Precharge Power-Down exit	EMIF_SDRAM_OUTPUT_IMPEDANCE_CALIBRATION_CONFIG[28] ZQ_SFEXITEN	0x1
Enable ZQ calibration for CS0	EMIF_SDRAM_OUTPUT_IMPEDANCE_CALIBRATION_CONFIG[30] ZQ_CS0EN	0x1
Program the EMIF_TEMPERATURE_ALERT_CONFIG register	EMIF_TEMPERATURE_ALERT_CONFIG	RD_VAL_2

(4) Values loaded in these registers depend on OPP.

(5) Writes to these registers do not have any impact. It's up to the user to skip these steps.

Table 10-32. EMIF Configuration Sequence (continued)

Step	Register/ Bit Field	Value
Program the EMIF_READ_WRITE_LEVELING_RAMP_WINDOW register. **NOTE: Incremental leveling is not supported on this device.**		0x-
IF Memory type == DDR3 **AND** SDRAM content doesn't need to be maintained:		
Enable read-write leveling	EMIF_READ_WRITE_LEVELING_RAMP_CONTROL[31] RDWRLVL_EN	0x1
ELSE:		
Disable read-write leveling	EMIF_READ_WRITE_LEVELING_RAMP_CONTROL[31] RDWRLVL_EN	0x0
ENDIF		
Program bits [30:0] of EMIF_READ_WRITE_LEVELING_RAMP_CONTROL **NOTE: Incremental leveling is not supported on this device.**		0x-
Program the EMIF_READ_WRITE_LEVELING_CONTROL register		0x0
Define the read latency for the read data from SDRAM in number of DDR clock cycles	EMIF_DDR_PHY_CONTROL_1[4:0] READ_LATENCY	0x- (typical value >= (CL + 4))
Configure whether the MDLL lock is asserted based on single sample or average of 16 samples	EMIF_DDR_PHY_CONTROL_1[9] PHY_FAST_DLL_LOCK	0x-
Define the maximum number of delay line taps variation while maintaining the master DLL lock. The recommended value is 0x10	EMIF_DDR_PHY_CONTROL_1[17:10] PHY_DLL_LOCK_DIFF	0x10
Configure whether the clock to the SDRAM is inverted or not	EMIF_DDR_PHY_CONTROL_1[18] PHY_INVERT_CLKOUT	0x-
When leveling is used set PHY_DIS_CALIB_RST to 0x0	EMIF_DDR_PHY_CONTROL_1[19] PHY_DIS_CALIB_RST	0x0
Program the slave delay line delays to support 2× mode	EMIF_DDR_PHY_CONTROL_1[21] PHY_HALF_DELAYS	0x1
IF Memory type == DDR3:		
Unmask read data eye training, DQS gate training and write leveling training during full leveling	EMIF_DDR_PHY_CONTROL_1[27] RDLVL_MASK	0x0
	EMIF_DDR_PHY_CONTROL_1[26] RDLVLGATE_MASK	0x0
	EMIF_DDR_PHY_CONTROL_1[25] WRLVL_MASK	0x0
ELSE:		
Mask read data eye training, DQS gate training and write leveling training during full leveling	EMIF_DDR_PHY_CONTROL_1[27] RDLVL_MASK	0x1
	EMIF_DDR_PHY_CONTROL_1[26] RDLVLGATE_MASK	0x1
	EMIF_DDR_PHY_CONTROL_1[25] WRLVL_MASK	0x1
ENDIF		
Program the shadow register of EMIF_DDR_PHY_CONTROL_1	EMIF_DDR_PHY_CONTROL_1_SHADOW	EMIF_DDR_PHY_CONTROL_1
Program the EMIF_DDR_PHY_CONTROL_2 register	EMIF_DDR_PHY_CONTROL_2 (5)	RD_VAL_3
If needed, configure and enable the priority-to-class-of-service mapping for the commands in the command FIFO	EMIF_PRIORITY_TO_CLASS_OF_SERVICE_MAPPING	0x-
If needed, configure and enable the master-ID-to-class-of-service mapping for the commands in the command FIFO	EMIF_CONNECTION_ID_TO_CLASS_OF_SERVICE_1_MAPPING	0x-
	EMIF_CONNECTION_ID_TO_CLASS_OF_SERVICE_2_MAPPING	0x-
Chose whether Mflag or Class of Service is used	EMIF_READ_WRITE_EXECUTION_THRESHOLD[31] MFLAG_OVERRIDE	0x-
Configure the write threshold after which EMIF switches to read commands	EMIF_READ_WRITE_EXECUTION_THRESHOLD[12:8] WR_THRSH	0x-

Table 10-32. EMIF Configuration Sequence (continued)

Step	Register/ Bit Field	Value
Configure the read threshold after which EMIF switches to write commands	EMIF_READ_WRITE_EXECUTION_THRESHOLD[4:0] RD_THRSH	0x-
Configure the priority rise counters	EMIF_COS_CONFIG	0x-
IF Memory type == DDR3:		
Configure the SDRAM refresh rate with a value of 31.25µs to get 500µs delay between RESET de-assertion to CKE assertion after power-up	EMIF_SDRAM_REFRESH_CONTROL_SHADOW [15:0] REFRESH_RATE_SHDW	0x-
Program the EMIF_SDRAM_REFRESH_CONTROL register with value same as its shadow register	EMIF_SDRAM_REFRESH_CONTROL[15:0] REFRESH_RATE	REFRESH_RATE_SHDW
ELSEIF Memory type == DDR2:		
Configure the SDRAM refresh rate with value used initially to get 200µs delay between POWER UP and PRECHARGE ALL command	EMIF_SDRAM_REFRESH_CONTROL_SHADOW [15:0] REFRESH_RATE_SHDW	0x-
Program the EMIF_SDRAM_REFRESH_CONTROL register with value same as its shadow register	EMIF_SDRAM_REFRESH_CONTROL[15:0] REFRESH_RATE	REFRESH_RATE_SHDW
ELSEIF Memory type == LPDDR2:		
Configure the SDRAM refresh rate with value used initially to get 200µs delay between POWER UP and RESET command to LPDDR2	EMIF_SDRAM_REFRESH_CONTROL_SHADOW [15:0] REFRESH_RATE_SHDW	0x-
Program the EMIF_SDRAM_REFRESH_CONTROL register with value same as its shadow register	EMIF_SDRAM_REFRESH_CONTROL[15:0] REFRESH_RATE	REFRESH_RATE_SHDW
ENDIF		
Assign the external bank address bits from lower or higher L3 address as shown in Section 10.2.4.12, *SDRAM Address Mapping*	EMIF_SDRAM_CONFIG_2[27] EBANK_POS	0x-
Select the SDRAM type	EMIF_SDRAM_CONFIG[31:29] SDRAM_TYPE	0x-
Assign internal bank address bits from L3 address as shown in Section 10.2.4.12, *SDRAM Address Mapping*	EMIF_SDRAM_CONFIG[28:27] IBANK_POS	0x-
Choose SDRAM data bus width	EMIF_SDRAM_CONFIG[15:14] NARROW_MODE	0x-
Define CAS latency when accessing connected SDRAM devices.	EMIF_SDRAM_CONFIG[13:10] CL	0x-
Define the number of row address bits of connected SDRAM device.	EMIF_SDRAM_CONFIG[9:7] ROWSIZE	0x-
Define the number of banks inside connected SDRAM device.	EMIF_SDRAM_CONFIG[6:4] IBANK	0x-
Define the internal page size of connected SDRAM device.	EMIF_SDRAM_CONFIG[2:0] PAGESIZE	0x-
Wait 1ms		
Configure the SDRAM refresh rate with value according to the actual memory refresh period requirements	EMIF_SDRAM_REFRESH_CONTROL_SHADOW [15:0] REFRESH_RATE_SHDW	0x-
Program the EMIF_SDRAM_REFRESH_CONTROL register with value same as its shadow register	EMIF_SDRAM_REFRESH_CONTROL[15:0] REFRESH_RATE	REFRESH_RATE_SHDW

Table 10-32. EMIF Configuration Sequence (continued)

Step	Register/ Bit Field	Value
IF Memory type == LPDDR2:		
Initialize the LPDDR2 Mode Registers (MRs) by writing to the EMIF_LPDDR2_MODE_REG_CONFIG and EMIF_LPDDR2_MODE_REG_DATA registers:		
· Configure calibration command after initialization (write to LPDDR2 MR10)	EMIF_LPDDR2_MODE_REG_CONFIG EMIF_LPDDR2_MODE_REG_DATA	0xA 0xFF
· Configure burst parameters and nWR (write to LPDDR2 MR1)	EMIF_LPDDR2_MODE_REG_CONFIG EMIF_LPDDR2_MODE_REG_DATA	0x1 0x-
· Configure read and write latencies (write to LPDDR2 MR2)	EMIF_LPDDR2_MODE_REG_CONFIG EMIF_LPDDR2_MODE_REG_DATA	0x2 0x-
· If needed, other mode registers can also be configured.		
LPDDR2 MR0 will be read in next step.	EMIF_LPDDR2_MODE_REG_CONFIG	0x0
Wait until LPDDR2 auto initialization process completes (read LPDDR2 MR0).	EMIF_LPDDR2_MODE_REG_DATA	
If needed, other read only mode registers of the LPDDR2 memory can also be read by first writing the mode register address to EMIF_LPDDR2_MODE_REG_CONFIG and then reading EMIF_LPDDR2_MODE_REG_DATA.		
ENDIF		
IF Memory type == DDR3 **AND** Hardware leveling is used **AND** SDRAM content doesn't need to be maintained:		
IF ECC is used:		
Perform dummy ECC setup just to allow hardware leveling of ECC memories	EMIF_ECC_ADDRESS_RANGE_1	0x0
	EMIF_ECC_ADDRESS_RANGE_2	0x0
	EMIF_ECC_CTRL_REG	0xC000 0000
ENDIF		
Clear the phy_reg_fifo_we_in_misaligned_sticky status flag	EMIF_EXT_PHY_CONTROL_36[8] REG_PHY_FIFO_WE_IN_MISALIGNED_CLR	0x1
Temporarily disable SDRAM refreshes	EMIF_SDRAM_REFRESH_CONTROL[31] INITREF_DIS	0x1
Trigger read-write leveling	EMIF_READ_WRITE_LEVELING_CONTROL[31] RDWRLVLFULL_START	0x1
Wait 300µs		
Wait till read-write leveling completes	EMIF_READ_WRITE_LEVELING_CONTROL[31] RDWRLVLFULL_START	0x0
Enable the temporarily disabled SDRAM refreshes	EMIF_SDRAM_REFRESH_CONTROL[31] INITREF_DIS	0x0
Check EMIF_STATUS for leveling timeouts	EMIF_STATUS[6] RDLVLGATETO	0x1
	EMIF_STATUS[5] RDLVLTO	0x1
	EMIF_STATUS[4] WRLVLTO	0x1
IF special (6) use-cases:		

(6) In some special use-cases with features like suspend to RAM where EMIF may be reset but SDRAM content needs to be maintained correctly across such an event. In such cases, when EMIF is configured at first boot with hardware leveling enabled, the hardware leveling output from EMIF_PHY_STATUS_x registers is copied to the EMIF_EXT_PHY_CONTROL_x registers. When EMIF comes back from a reset state and needs to be reconfigured without losing SDRAM content, the EMIF_READ_WRITE_LEVELING_RAMP_CONTROL[31] RDWRLVL_EN bit must be 0 and hardware leveling must not be triggered. In this case, the EMIF_EXT_PHY_CONTROL_2 through EMIF_EXT_PHY_CONTROL_21 registers take effect. Here it is expected that SDRAM is moved to self-refresh mode before EMIF enters reset state and SDRAM content is maintained correctly by appropriate circuitry external to the SoC.

Table 10-32. EMIF Configuration Sequence (continued)

Step	Register/ Bit Field	Value
Copy EMIF_PHY_STATUS_12 through EMIF_PHY_STATUS_16 to EMIF_EXT_PHY_CONTROL_2 through EMIF_EXT_PHY_CONTROL_6		
Copy EMIF_PHY_STATUS_7 through EMIF_PHY_STATUS_11 to EMIF_EXT_PHY_CONTROL_7 through EMIF_EXT_PHY_CONTROL_11		
Copy EMIF_PHY_STATUS_17 through EMIF_PHY_STATUS_26 to EMIF_EXT_PHY_CONTROL_12 through EMIF_EXT_PHY_CONTROL_21		
Mask read data eye training, DQS gate training and write leveling training to disable the use of registers EMIF_PHY_STATUS_7 through EMIF_PHY_STATUS_26 registers	EMIF_DDR_PHY_CONTROL_1[27:25]	0x7
Program the shadow register of EMIF_DDR_PHY_CONTROL_1	EMIF_DDR_PHY_CONTROL_1_SHADOW	EMIF_DDR_PHY_CONTROL_1
Disable read-write leveling	EMIF_READ_WRITE_LEVELING_RAMP_CONTROL[31] RDWRLVLFULL_START	0x0
ENDIF		
Clear the ECC control register	EMIF_ECC_CTRL_REG	0x0
ENDIF		

10.2.5.1.2 Operational Modes Configuration

10.2.5.1.2.1 EMIF Output Impedance Calibration Mode

Table 10-33. EMIF Output Impedance Calibration Mode

Step	Register/ Bit Field / Programming Model	Value
Enable ZQ callibration for CS0.	EMIF_SDRAM_OUTPUT_IMPEDANCE_CALIBRATION_CONFIG[30] ZQ_CS0EN	0x1
Define the interval (number of refresh periods) between ZQCS commands.	EMIF_SDRAM_OUTPUT_IMPEDANCE_CALIBRATION_CONFIG[15:0] ZQ_REFINTERVAL	0x-
Define the number of ZQCL durations that build the ZQINIT duration.	EMIF_SDRAM_OUTPUT_IMPEDANCE_CALIBRATION_CONFIG[19:18] ZQ_ZQINIT_MULT	0x-
Define the number of ZQCS intervals that build the ZQCL duration.	EMIF_SDRAM_OUTPUT_IMPEDANCE_CALIBRATION_CONFIG[17:16] ZQ_ZQCL_MULT	0x-
Enable the issuing of ZQ-Long Command on Self-Refresh, Active Power-Down, and Precharge Power-Down exit.	EMIF_SDRAM_OUTPUT_IMPEDANCE_CALIBRATION_CONFIG[28] ZQ_SFEXITEN	0x1

10.2.5.1.2.2 EMIF SDRAM Self-Refresh

Table 10-34. EMIF SDRAM Self-Refresh Entering

Step	Register/ Bit Field / Programming Model	Value
Define the number of DDR clock cycles after which the EMIF puts the external SDRAM in Self Refresh mode, when EMIF is idle.	EMIF_POWER_MANAGEMENT_CONTROL[7:4] SR_TIM	0x-
Enable the Self-Refresh mode	EMIF_POWER_MANAGEMENT_CONTROL[10:8] LP_MODE	0x2
Write the shadow register of EMIF_POWER_MANAGEMENT_CONTROL	EMIF_POWER_MANAGEMENT_CONTROL_SHADOW	0x-

Table 10-35. EMIF SDRAM Self-Refresh Exiting

Step	Register/ Bit Field / Programming Model	Value
Change LP_MODE bitfield from 0x2 to any value.	EMIF_POWER_MANAGEMENT_CONTROL[10:8] LP_MODE	0x-
Write the shadow register of EMIF_POWER_MANAGEMENT_CONTROL	EMIF_POWER_MANAGEMENT_CONTROL_SHADOW	0x-

10.2.5.1.2.3 EMIF SDRAM Power-Down Mode

Table 10-36. EMIF SDRAM Power-Down Mode Entering

Step	Register/ Bit Field / Programming Model	Value
Define the number of DDR clock cycles after which the EMIF puts the external SDRAM in Power Down mode, when EMIF is idle.	EMIF_POWER_MANAGEMENT_CONTROL[15:12] PD_TIM	0x-
Enable (enter) the Power-down mode	EMIF_POWER_MANAGEMENT_CONTROL[10:8] LP_MODE	0x4
Write the shadow register of EMIF_POWER_MANAGEMENT_CONTROL	EMIF_POWER_MANAGEMENT_CONTROL_SHADOW	0x-

Table 10-37. EMIF SDRAM Power-Down Mode Exiting

Step	Register/ Bit Field / Programming Model	Value
Change LP_MODE bitfield from 0x4 to any value.	EMIF_POWER_MANAGEMENT_CONTROL[10:8] LP_MODE	0x-
Write the shadow register of EMIF_POWER_MANAGEMENT_CONTROL	EMIF_POWER_MANAGEMENT_CONTROL_SHADOW	0x-

10.2.5.1.2.4 EMIF LPDDR2-SDRAM Deep Power-Down Mode

Table 10-38. EMIF LPDDR2-SDRAM Deep Power-Down Mode Entering

Step	Register/ Bit Field / Programming Model	Value
Define the number of DDR clock cycles after which the EMIF puts the external SDRAM in Power saving mode, when EMIF is idle.	EMIF_POWER_MANAGEMENT_CONTROL[3:0] CS_TIM	0x-
Enable (enter) the Deep Power-Down mode	EMIF_POWER_MANAGEMENT_CONTROL[11] DPD_EN	0x1
Write the shadow register of EMIF_POWER_MANAGEMENT_CONTROL	EMIF_POWER_MANAGEMENT_CONTROL_SHADOW	0x-

Table 10-39. EMIF LPDDR2-SDRAM Deep Power-Down Mode Exiting

Step	Register/ Bit Field / Programming Model	Value
Clear DPD_EN bit.	EMIF_POWER_MANAGEMENT_CONTROL[11] DPD_EN	0
Write the shadow register of EMIF_POWER_MANAGEMENT_CONTROL	EMIF_POWER_MANAGEMENT_CONTROL_SHADOW	0x-

10.2.5.1.2.5 LPDDR2 Temperature Monitoring Mode

Table 10-40. LPDDR2 Temperature Monitoring Mode

Step	Register/ Bit Field / Programming Model	Value
Determine which chip-select (0 or 1) is used for temperature alert polling	EMIF_TEMPERATURE_ALERT_CONFIG[31] TA_CS1EN EMIF_TEMPERATURE_ALERT_CONFIG[30] TA_CS0EN	0x-
Define the interval (number of refresh periods) between temperature alert polls.	EMIF_TEMPERATURE_ALERT_CONFIG[21:0] TA_REFINTERVAL	0x-
Define the width of the physical memory device	EMIF_TEMPERATURE_ALERT_CONFIG[27:26] TA_DEVWDT	0x-
Define which external byte lanes contain a device for temperature monitoring (which devices to be polled).	EMIF_TEMPERATURE_ALERT_CONFIG[25:24] TA_DEVCNT	0x-
(Optional) Enable the issuing of a temperature alert poll on Self-Refresh exit (poll for temperature change every time EMIF exits Self-Refresh, Active Power-Down, and Precharge Power-Down modes).	EMIF_TEMPERATURE_ALERT_CONFIG[28] TA_SFEXITEN	0x1
Enable System L3 interrupt for SDRAM temperature alert.	EMIF_SYSTEM_OCP_INTERRUPT_ENABLE_SET[1] EN_TA_SYS	0x1

10.2.5.1.2.6 EMIF ECC Configuration

Table 10-41. EMIF ECC Configuration

Step	Register/ Bit Field / Programming Model	Value
Before configuring the EMIF ECC registers, the ECC must be enabled from the Control Module	CTRL_WKUP_EMIF1_SDRAM_CONFIG_EXT [16] EMIF1_EN_ECC	0x1
Configure the address ranges which have to be ECC protected	EMIF_ECC_ADDRESS_RANGE_1 and EMIF_ECC_ADDRESS_RANGE_2 (1)	0x-
Enable the first ECC protected address range	EMIF_ECC_CTRL_REG[0] REG_ECC_ADDR_RGN_1_EN	0x1
Enable the second ECC protected address range	EMIF_ECC_CTRL_REG[1] REG_ECC_ADDR_RGN_2_EN (1)	0x1
Configure whether the EMIF accesses within the address ranges defined by EMIF_ECC_ADDRESS_RANGE_1 and EMIF_ECC_ADDRESS_RANGE_2 are ECC protected (=0x1) or the EMIF accesses outside these address ranges are ECC protected (=0x0).	EMIF_ECC_CTRL_REG[30] REG_ECC_ADDR_RGN_PROT	0x-
(Optional) Configure the thresholds to generate 1-bit error interrupt	EMIF_1B_ECC_ERR_THRSH	0x-
Enable ECC	EMIF_ECC_CTRL_REG[31] REG_ECC_EN	0x1
Initialize the ECC protected memory regions with quanta-sized and quanta-aligned data		
Clear the status flags and other status history	EMIF_1B_ECC_ERR_CNT	Read the register value and write it back to clear
	EMIF_1B_ECC_ERR_DIST_1	0xFFFF FFFF
	EMIF_2B_ECC_ERR_ADDR_LOG	0x1
	EMIF_SYSTEM_OCP_INTERRUPT_STATUS [5:3]	0x7

(1) It is not mandatory to enable both address ranges. This is provided for flexibility allowing two non-overlapping ECC protected regions to be created.

NOTE: The xxx_RATIO8 and xxx_RATIO9 bit fields are associated with the ECC data PHY and must be loaded with values same as in the xxx_RATIO0 through xxx_RATIO7 bit fields from registers EMIF_EXT_PHY_CONTROL_xx. This must be taken into account to avoid unexpected behavior and possible errors, if ECC is used.

10.2.6 EMIF Register Manual

NOTE: LPDDR2 is not supported on the DRA78x family of devices.

Table 10-42 lists the EMIF instance.

10.2.6.1 EMIF Instance Summary

Table 10-42. EMIF Instance Summary

Module Name	Base Address	Size
EMIF	0x4C00 0000	16 MiB

10.2.6.2 EMIF Registers

Many register values are programmed with full-speed DDR clock cycles, whereas EMIF always runs at half the speed of this clock. Because EMIF is only capable of decrementing these values at half speed (taking into account the minus 1 programming model) the following model applies for data issued on a lower bus:

- REG_VALUE = 0 → 2 DDR clocks
- REG_VALUE = 1 → 2 DDR clocks
- REG_VALUE = 2 → 4 DDR clocks
- REG_VALUE = 3 → 4 DDR clocks
- REG_VALUE = 4 → 6 DDR clocks
- REG_VALUE = 5 → 6 DDR clocks
- REG_VALUE = 6 → 8 DDR clocks
- REG_VALUE = 7 → 8 DDR clocks

Activate and deactivate commands use the upper bus. Register values associated with these operations follow the general rule:

- REG_VALUE = 0 → 1 DDR clocks
- REG_VALUE = 1 → 2 DDR clocks
- REG_VALUE = 2 → 2 DDR clocks
- REG_VALUE = 3 → 4 DDR clocks
- REG_VALUE = 4 → 4 DDR clocks
- REG_VALUE = 5 → 6 DDR clocks
- REG_VALUE = 6 → 6 DDR clocks

NOTE: Shadow registers are loaded on any frequency change or SidleReq/SidleAck transition at the point where the EMIF has put SDRAM into self-refresh mode.

Table 10-43 summarizes the EMIF register mapping.

10.2.6.2.1 EMIF Register Summary

Table 10-43. EMIF Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	EMIF Physical Address
EMIF_REVISION	R	32	0x0000 0000	0x4C00 0000
EMIF_STATUS	R	32	0x0000 0004	0x4C00 0004
EMIF_SDRAM_CONFIG	RW	32	0x0000 0008	0x4C00 0008

Table 10-43. EMIF Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	EMIF Physical Address
EMIF_SDRAM_CONFIG_2	RW	32	0x0000 000C	0x4C00 000C
EMIF_SDRAM_REFRESH_CONTROL	RW	32	0x0000 0010	0x4C00 0010
EMIF_SDRAM_REFRESH_CONTROL_SHADOW	RW	32	0x0000 0014	0x4C00 0014
EMIF_SDRAM_TIMING_1	RW	32	0x0000 0018	0x4C00 0018
EMIF_SDRAM_TIMING_1_SHADOW	RW	32	0x0000 001C	0x4C00 001C
EMIF_SDRAM_TIMING_2	RW	32	0x0000 0020	0x4C00 0020
EMIF_SDRAM_TIMING_2_SHADOW	RW	32	0x0000 0024	0x4C00 0024
EMIF_SDRAM_TIMING_3	RW	32	0x0000 0028	0x4C00 0028
EMIF_SDRAM_TIMING_3_SHADOW	RW	32	0x0000 002C	0x4C00 002C
EMIF_LPDDR2_NVM_TIMING	RW	32	0x0000 0030	0x4C00 0030
EMIF_LPDDR2_NVM_TIMING_SHADOW	RW	32	0x0000 0034	0x4C00 0034
EMIF_POWER_MANAGEMENT_CONTROL	RW	32	0x0000 0038	0x4C00 0038
EMIF_POWER_MANAGEMENT_CONTROL_SHADOW	RW	32	0x0000 003C	0x4C00 003C
EMIF_LPDDR2_MODE_REG_DATA	RW	32	0x0000 0040	0x4C00 0040
EMIF_LPDDR2_MODE_REG_CONFIG	RW	32	0x0000 0050	0x4C00 0050
EMIF_OCP_CONFIG	RW	32	0x0000 0054	0x4C00 0054
EMIF_OCP_CONFIG_VALUE_1	R	32	0x0000 0058	0x4C00 0058
EMIF_OCP_CONFIG_VALUE_2	R	32	0x0000 005C	0x4C00 005C
EMIF_IODFT_TLGC	RW	32	0x0000 0060	0x4C00 0060
EMIF_PERFORMANCE_COUNTER_1	R	32	0x0000 0080	0x4C00 0080
EMIF_PERFORMANCE_COUNTER_2	R	32	0x0000 0084	0x4C00 0084
EMIF_PERFORMANCE_COUNTER_CONFIG	RW	32	0x0000 0088	0x4C00 0088
EMIF_PERFORMANCE_COUNTER_MASTER_REGION_SELECT	RW	32	0x0000 008C	0x4C00 008C
EMIF_PERFORMANCE_COUNTER_TIME	R	32	0x0000 0090	0x4C00 0090
EMIF_MISC_REG	RW	32	0x0000 0094	0x4C00 0094
EMIF_DLL_CALIB_CTRL	RW	32	0x0000 0098	0x4C00 0098
EMIF_DLL_CALIB_CTRL_SHADOW	RW	32	0x0000 009C	0x4C00 009C
EMIF_END_OF_INTERRUPT	RW	32	0x0000 00A0	0x4C00 00A0
EMIF_SYSTEM_OCP_INTERRUPT_RAW_STATUS	RW	32	0x0000 00A4	0x4C00 00A4
RESERVED	R	32	0x0000 00A8	0x4C00 00A8
EMIF_SYSTEM_OCP_INTERRUPT_STATUS	RW	32	0x0000 00AC	0x4C00 00AC
RESERVED	R	32	0x0000 00B0	0x4C00 00B0
EMIF_SYSTEM_OCP_INTERRUPT_ENABLE_SET	RW	32	0x0000 00B4	0x4C00 00B4
RESERVED	R	32	0x0000 00B8	0x4C00 00B8
EMIF_SYSTEM_OCP_INTERRUPT_ENABLE_CLEAR	RW	32	0x0000 00BC	0x4C00 00BC
RESERVED	R	32	0x0000 00C0	0x4C00 00C0
EMIF_SDRAM_OUTPUT_IMPEDANCE_CALIBRATION_CONFIG	RW	32	0x0000 00C8	0x4C00 00C8
EMIF_TEMPERATURE_ALERT_CONFIG	RW	32	0x0000 00CC	0x4C00 00CC
EMIF_OCP_ERROR_LOG	R	32	0x0000 00D0	0x4C00 00D0
EMIF_READ_WRITE_LEVELING_RAMP_WINDOW	RW	32	0x0000 00D4	0x4C00 00D4

Table 10-43. EMIF Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	EMIF Physical Address
EMIF_READ_WRITE_LEVELING_RAMP_CONTROL	RW	32	0x0000 00D8	0x4C00 00D8
EMIF_READ_WRITE_LEVELING_CONTROL	RW	32	0x0000 00DC	0x4C00 00DC
EMIF_DDR_PHY_CONTROL_1	RW	32	0x0000 00E4	0x4C00 00E4
EMIF_DDR_PHY_CONTROL_1_SHADOW	RW	32	0x0000 00E8	0x4C00 00E8
EMIF_DDR_PHY_CONTROL_2	RW	32	0x0000 00EC	0x4C00 00EC
EMIF_PRIORITY_TO_CLASS_OF_SERVICE_MAPPING	RW	32	0x0000 0100	0x4C00 0100
EMIF_CONNECTION_ID_TO_CLASS_OF_SERVICE_1_MAPPING	RW	32	0x0000 0104	0x4C00 0104
EMIF_CONNECTION_ID_TO_CLASS_OF_SERVICE_2_MAPPING	RW	32	0x0000 0108	0x4C00 0108
EMIF_ECC_CTRL_REG	RW	32	0x0000 0110	0x4C00 0110
EMIF_ECC_ADDRESS_RANGE_1	RW	32	0x0000 0114	0x4C00 0114
EMIF_ECC_ADDRESS_RANGE_2	RW	32	0x0000 0118	0x4C00 0118
EMIF_READ_WRITE_EXECUTION_THRESHOLD	RW	32	0x0000 0120	0x4C00 0120
EMIF_COS_CONFIG	RW	32	0x0000 0124	0x4C00 0124
EMIF_1B_ECC_ERR_CNT	RW	32	0x0000 0130	0x4C00 0130
EMIF_1B_ECC_ERR_THRSH	RW	32	0x0000 0134	0x4C00 0134
EMIF_1B_ECC_ERR_DIST_1	RW	32	0x0000 0138	0x4C00 0138
EMIF_1B_ECC_ERR_ADDR_LOG	RW	32	0x0000 013C	0x4C00 013C
EMIF_2B_ECC_ERR_ADDR_LOG	RW	32	0x0000 0140	0x4C00 0140
EMIF_PHY_STATUS_1	R	32	0x0000 0144	0x4C00 0144
EMIF_PHY_STATUS_2	R	32	0x0000 0148	0x4C00 0148
EMIF_PHY_STATUS_3	R	32	0x0000 014C	0x4C00 014C
EMIF_PHY_STATUS_4	R	32	0x0000 0150	0x4C00 0150
EMIF_PHY_STATUS_5	R	32	0x0000 0154	0x4C00 0154
EMIF_PHY_STATUS_6	R	32	0x0000 0158	0x4C00 0158
EMIF_PHY_STATUS_7	R	32	0x0000 015C	0x4C00 015C
EMIF_PHY_STATUS_8	R	32	0x0000 0160	0x4C00 0160
EMIF_PHY_STATUS_9	R	32	0x0000 0164	0x4C00 0164
EMIF_PHY_STATUS_10	R	32	0x0000 0168	0x4C00 0168
EMIF_PHY_STATUS_11	R	32	0x0000 016C	0x4C00 016C
EMIF_PHY_STATUS_12	R	32	0x0000 0170	0x4C00 0170
EMIF_PHY_STATUS_13	R	32	0x0000 0174	0x4C00 0174
EMIF_PHY_STATUS_14	R	32	0x0000 0178	0x4C00 0178
EMIF_PHY_STATUS_15	R	32	0x0000 017C	0x4C00 017C
EMIF_PHY_STATUS_16	R	32	0x0000 0180	0x4C00 0180
EMIF_PHY_STATUS_17	R	32	0x0000 0184	0x4C00 0184
EMIF_PHY_STATUS_18	R	32	0x0000 0188	0x4C00 0188
EMIF_PHY_STATUS_19	R	32	0x0000 018C	0x4C00 018C
EMIF_PHY_STATUS_20	R	32	0x0000 0190	0x4C00 0190
EMIF_PHY_STATUS_21	R	32	0x0000 0194	0x4C00 0194
EMIF_PHY_STATUS_22	R	32	0x0000 0198	0x4C00 0198
EMIF_PHY_STATUS_23	R	32	0x0000 019C	0x4C00 019C
EMIF_PHY_STATUS_24	R	32	0x0000 01A0	0x4C00 01A0
EMIF_PHY_STATUS_25	R	32	0x0000 01A4	0x4C00 01A4

Table 10-43. EMIF Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	EMIF Physical Address
EMIF_PHY_STATUS_26	R	32	0x0000 01A8	0x4C00 01A8
EMIF_PHY_STATUS_27	R	32	0x0000 01AC	0x4C00 01AC
EMIF_PHY_STATUS_28	R	32	0x0000 01B0	0x4C00 01B0
EMIF_EXT_PHY_CONTROL_1	RW	32	0x0000 0200	0x4C00 0200
EMIF_EXT_PHY_CONTROL_1_SHADOW	RW	32	0x0000 0204	0x4C00 0204
EMIF_EXT_PHY_CONTROL_2	RW	32	0x0000 0208	0x4C00 0208
EMIF_EXT_PHY_CONTROL_2_SHADOW	RW	32	0x0000 020C	0x4C00 020C
EMIF_EXT_PHY_CONTROL_3	RW	32	0x0000 0210	0x4C00 0210
EMIF_EXT_PHY_CONTROL_3_SHADOW	RW	32	0x0000 0214	0x4C00 0214
EMIF_EXT_PHY_CONTROL_4	RW	32	0x0000 0218	0x4C00 0218
EMIF_EXT_PHY_CONTROL_4_SHADOW	RW	32	0x0000 021C	0x4C00 021C
EMIF_EXT_PHY_CONTROL_5	RW	32	0x0000 0220	0x4C00 0220
EMIF_EXT_PHY_CONTROL_5_SHADOW	RW	32	0x0000 0224	0x4C00 0224
EMIF_EXT_PHY_CONTROL_6	RW	32	0x0000 0228	0x4C00 0228
EMIF_EXT_PHY_CONTROL_6_SHADOW	RW	32	0x0000 022C	0x4C00 022C
EMIF_EXT_PHY_CONTROL_7	RW	32	0x0000 0230	0x4C00 0230
EMIF_EXT_PHY_CONTROL_7_SHADOW	RW	32	0x0000 0234	0x4C00 0234
EMIF_EXT_PHY_CONTROL_8	RW	32	0x0000 0238	0x4C00 0238
EMIF_EXT_PHY_CONTROL_8_SHADOW	RW	32	0x0000 023C	0x4C00 023C
EMIF_EXT_PHY_CONTROL_9	RW	32	0x0000 0240	0x4C00 0240
EMIF_EXT_PHY_CONTROL_9_SHADOW	RW	32	0x0000 0244	0x4C00 0244
EMIF_EXT_PHY_CONTROL_10	RW	32	0x0000 0248	0x4C00 0248
EMIF_EXT_PHY_CONTROL_10_SHADOW	RW	32	0x0000 024C	0x4C00 024C
EMIF_EXT_PHY_CONTROL_11	RW	32	0x0000 0250	0x4C00 0250
EMIF_EXT_PHY_CONTROL_11_SHADOW	RW	32	0x0000 0254	0x4C00 0254
EMIF_EXT_PHY_CONTROL_12	RW	32	0x0000 0258	0x4C00 0258
EMIF_EXT_PHY_CONTROL_12_SHADOW	RW	32	0x0000 025C	0x4C00 025C
EMIF_EXT_PHY_CONTROL_13	RW	32	0x0000 0260	0x4C00 0260
EMIF_EXT_PHY_CONTROL_13_SHADOW	RW	32	0x0000 0264	0x4C00 0264
EMIF_EXT_PHY_CONTROL_14	RW	32	0x0000 0268	0x4C00 0268
EMIF_EXT_PHY_CONTROL_14_SHADOW	RW	32	0x0000 026C	0x4C00 026C
EMIF_EXT_PHY_CONTROL_15	RW	32	0x0000 0270	0x4C00 0270
EMIF_EXT_PHY_CONTROL_15_SHADOW	RW	32	0x0000 0274	0x4C00 0274
EMIF_EXT_PHY_CONTROL_16	RW	32	0x0000 0278	0x4C00 0278
EMIF_EXT_PHY_CONTROL_16_SHADOW	RW	32	0x0000 027C	0x4C00 027C
EMIF_EXT_PHY_CONTROL_17	RW	32	0x0000 0280	0x4C00 0280
EMIF_EXT_PHY_CONTROL_17_SHADOW	RW	32	0x0000 0284	0x4C00 0284
EMIF_EXT_PHY_CONTROL_18	RW	32	0x0000 0288	0x4C00 0288
EMIF_EXT_PHY_CONTROL_18_SHADOW	RW	32	0x0000 028C	0x4C00 028C
EMIF_EXT_PHY_CONTROL_19	RW	32	0x0000 0290	0x4C00 0290
EMIF_EXT_PHY_CONTROL_19_SHADOW	RW	32	0x0000 0294	0x4C00 0294
EMIF_EXT_PHY_CONTROL_20	RW	32	0x0000 0298	0x4C00 0298
EMIF_EXT_PHY_CONTROL_20_SHADOW	RW	32	0x0000 029C	0x4C00 029C
EMIF_EXT_PHY_CONTROL_21	RW	32	0x0000 02A0	0x4C00 02A0
EMIF_EXT_PHY_CONTROL_21_SHADOW	RW	32	0x0000 02A4	0x4C00 02A4
EMIF_EXT_PHY_CONTROL_22	RW	32	0x0000 02A8	0x4C00 02A8
EMIF_EXT_PHY_CONTROL_22_SHADOW	RW	32	0x0000 02AC	0x4C00 02AC

Table 10-43. EMIF Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	EMIF Physical Address
EMIF_EXT_PHY_CONTROL_23	RW	32	0x0000 02B0	0x4C00 02B0
EMIF_EXT_PHY_CONTROL_23_SHADOW	RW	32	0x0000 02B4	0x4C00 02B4
EMIF_EXT_PHY_CONTROL_24	RW	32	0x0000 02B8	0x4C00 02B8
EMIF_EXT_PHY_CONTROL_24_SHADOW	RW	32	0x0000 02BC	0x4C00 02BC
EMIF_EXT_PHY_CONTROL_25	RW	32	0x0000 02C0	0x4C00 02C0
EMIF_EXT_PHY_CONTROL_25_SHADOW	RW	32	0x0000 02C4	0x4C00 02C4
EMIF_EXT_PHY_CONTROL_26	RW	32	0x0000 02C8	0x4C00 02C8
EMIF_EXT_PHY_CONTROL_26_SHADOW	RW	32	0x0000 02CC	0x4C00 02CC
EMIF_EXT_PHY_CONTROL_27	RW	32	0x0000 02D0	0x4C00 02D0
EMIF_EXT_PHY_CONTROL_27_SHADOW	RW	32	0x0000 02D4	0x4C00 02D4
EMIF_EXT_PHY_CONTROL_28	RW	32	0x0000 02D8	0x4C00 02D8
EMIF_EXT_PHY_CONTROL_28_SHADOW	RW	32	0x0000 02DC	0x4C00 02DC
EMIF_EXT_PHY_CONTROL_29	RW	32	0x0000 02E0	0x4C00 02E0
EMIF_EXT_PHY_CONTROL_29_SHADOW	RW	32	0x0000 02E4	0x4C00 02E4
EMIF_EXT_PHY_CONTROL_30	RW	32	0x0000 02E8	0x4C00 02E8
EMIF_EXT_PHY_CONTROL_30_SHADOW	RW	32	0x0000 02EC	0x4C00 02EC
EMIF_EXT_PHY_CONTROL_31	RW	32	0x0000 02F0	0x4C00 02F0
EMIF_EXT_PHY_CONTROL_31_SHADOW	RW	32	0x0000 02F4	0x4C00 02F4
EMIF_EXT_PHY_CONTROL_32	RW	32	0x0000 02F8	0x4C00 02F8
EMIF_EXT_PHY_CONTROL_32_SHADOW	RW	32	0x0000 02FC	0x4C00 02FC
EMIF_EXT_PHY_CONTROL_33	RW	32	0x0000 0300	0x4C00 0300
EMIF_EXT_PHY_CONTROL_33_SHADOW	RW	32	0x0000 0304	0x4C00 0304
EMIF_EXT_PHY_CONTROL_34	RW	32	0x0000 0308	0x4C00 0308
EMIF_EXT_PHY_CONTROL_34_SHADOW	RW	32	0x0000 030C	0x4C00 030C
EMIF_EXT_PHY_CONTROL_35	RW	32	0x0000 0310	0x4C00 0310
EMIF_EXT_PHY_CONTROL_35_SHADOW	RW	32	0x0000 0314	0x4C00 0314
EMIF_EXT_PHY_CONTROL_36	RW	32	0x0000 0318	0x4C00 0318
EMIF_EXT_PHY_CONTROL_36_SHADOW	RW	32	0x0000 031C	0x4C00 031C

10.2.6.2.2 EMIF Register Description

Table 10-44. EMIF_REVISION

Address Offset	0x0000 0000		
Physical Address	0x4C00 0000	**Instance**	EMIF
Description	Revision number register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REVISION																															

Bits	Field Name	Description	Type	Reset
31:0	REVISION	Module revision	R	0x- (1)

(1) TI internal data

Table 10-45. Register Call Summary for Register EMIF_REVISION

EMIF Controller

- EMIF Register Summary: [0]

Table 10-46. EMIF_STATUS

Address Offset	0x0000 0004		
Physical Address	0x4C00 0004	**Instance**	EMIF
Description	SDRAM Status Register (STATUS)		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BE	DUAL_CLK_MODE	FAST_INIT	RESERVED																					RDLVLGATETO	RDLVLTO	WRLVLTO	RESERVED	PHY_DLL_READY	RESERVED		

Bits	Field Name	Description	Type	Reset
31	BE	Big endian mode select for 8 and 16-bit devices, set to 1 for big endian or 0 for little endian operation. In current implementation, only 32-bit devices are supported - this bit is don't care.	R	0
30	DUAL_CLK_MODE	Dual Clock mode. Defines whether the EMIF_L3_ICLK and EMIF_FICLK clock are asynchronous. EMIF_L3_ICLK and EMIF_FICLK clock are asynchronous, if set to 1.	R	0
29	FAST_INIT	Fast Init. Defines whether the EMIF fast initialization mode has been enabled. Fast initialization is enabled if set to 1.	R	0
28:7	RESERVED	Reserved	R	0x00 0000
6	RDLVLGATETO	Read DQS Gate Training Timeout. Value of 1 indicates read DQS gate training has timed out because read DQS gate training done was not received from the PHY.	R	0
5	RDLVLTO	Read Data Eye Training Timeout. Value of 1 indicates read data eye training has timed out because read data eye training done was not received from the PHY.	R	0
4	WRLVLTO	Write Leveling Timeout. Value of 1 indicates write leveling has timed out because write leveling done was not received from the PHY.	R	0
3	RESERVED		R	0
2	PHY_DLL_READY	DDR PHY Ready. The DDR PHY is ready for normal operation, if set to 1.	R	0
1:0	RESERVED	Reserved	R	0x0

Table 10-47. Register Call Summary for Register EMIF_STATUS

EMIF Controller

- Global Initialization: [0][1][2][3]
- EMIF Register Summary: [4]

Table 10-48. EMIF_SDRAM_CONFIG

Address Offset	0x0000 0008		
Physical Address	0x4C00 0008	**Instance**	EMIF
Description	SDRAM Config Register. A write to this register will cause the EMIF to start the SDRAM initialization sequence. **CAUTION:** This register is loaded with values by control module at device reset.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDRAM_TYPE			IBANK_POS		DDR_TERM			DDR2_DDQS		DYN_ODT		DDR_DISABLE_DLL		SDRAM_DRIVE		CWL		NARROW_MODE		CL		ROWSIZE		IBANK			RESERVED	PAGESIZE			

Bits	Field Name	Description	Type	Reset
31:29	SDRAM_TYPE	SDRAM Type selection. This field is loaded from e-fuse. Set to 2 for DDR2 Set to 3 for DDR3 Set to 4 for LPDDR2 All other values are reserved. **Please check the device data manual for supported DDR memory types.**	RW	0x0
28:27	IBANK_POS	Internal bank position. See Section 10.2.4.12, *SDRAM Address Mapping*.	RW	0x0
26:24	DDR_TERM	DDR3 termination resistor value. Set to 0 to disable termination. For DDR3, set to 1 for RZQ/4, set to 2 for RZQ/2, set to 3 for RZQ/6, set to 4 for RZQ/12, and set to 5 for RZQ/8. All other values are reserved.	RW	0x0
23	DDR2_DDQS	Differential DQS enable. Set to 0 for single ended DQS (**Not supported**). Set to 1 for differential DQS.	RW	0
22:21	DYN_ODT	DDR3 Dynamic ODT. **Not supported**. Set to 0 to turn off dynamic ODT.	RW	0x0
20	DDR_DISABLE_DLL	Disable DLL select. Set to 1 to disable DLL inside SDRAM.	RW	0
19:18	SDRAM_DRIVE	SDRAM drive strength. For DDR2, set to 0 for normal, and set to 1 for weak drive strength. For DDR3, set to 0 for RZQ/6 and set to 1 for RZQ/7. All other values are reserved.	RW	0x0
17:16	CWL	DDR3 CAS Write latency. Value of 0, 1, 2, and 3 (CAS write latency of 5, 6, 7, and 8) are supported. Use the lowest value supported for best performance. All other values are reserved.	RW	0x0
15:14	NARROW_MODE	SDRAM data bus width. Set to 0 for 32-bit data bus width. Set to 1 for 16-bit data bus width. All other values are reserved.	RW	0x0

Bits	Field Name	Description	Type	Reset
13:10	CL	CAS Latency (referred to as read latency (RL) in some SDRAM specs). The value of this field defines the CAS latency to be used when accessing connected SDRAM devices. Values of 2, 3, 4 and 5 (CAS latency of 2, 3, 4 and 5) are supported for DDR2. Values of 2, 4, 6, 8, 10, 12 and 14 (CAS latency of 5, 6, 7, 8, 9, 10 and 11) are supported for DDR3. Values of 3, 4, 5, 6, 7, and 8 (CAS latency of 3, 4, 5, 6, 7, and 8) are supported for LPDDR2-SDRAM. The write latency (WL) is tied to the RL for the LPDDR2. See the LPDDR2 SDRAM standard. All other values are reserved.	RW	0x0
9:7	ROWSIZE	Row Size. Defines the number of row address bits of connected SDRAM devices. Set to 0 for 9 row bits, Set to 1 for 10 row bits, Set to 2 for 11 row bits, Set to 3 for 12 row bits, Set to 4 for 13 row bits, Set to 5 for 14 row bits, Set to 6 for 15 row bits, Set to 7 for 16 row bits. This field is only used when EMIF_SDRAM_CONFIG[28:27] IBANK_POS field is set to 1, 2, or 3 or EBANK_POS field in EMIF_SDRAM_CONFIG_2 register is set to 1.	RW	0x0
6:4	IBANK	Internal Bank setup. Defines number of banks inside connected SDRAM devices. Set to 0 for 1 bank, Set to 1 for 2 banks, Set to 2 for 4 banks, Set to 3 for 8 banks. All other values are reserved.	RW	0x0
3	RESERVED		R	0
2:0	PAGESIZE	Page Size. Defines the internal page size of connected SDRAM devices. Set to 0 for 256-word page (8 column bits), Set to 1 for 512-word page (9 column bits), Set to 2 for 1024-word page (10 column bits), Set to 3 for 2048-word page (11 column bits). All other values are reserved.	RW	0x0

Table 10-49. Register Call Summary for Register EMIF_SDRAM_CONFIG

EMIF Controller

- Arbitration of Commands in the Command FIFO: [0]
- Self-Refresh Mode: [1]
- SDRAM Initialization: [2][3]
- DDR2 SDRAM Initialization: [4][5][6][7][8][9][10][11][12][13][14]
- DDR3/DDR3L SDRAM Initialization: [15][16][17][18][19][20][21][22]
- LPDDR2 SDRAM Initialization: [23][24][25]
- SDRAM Address Mapping: [26][27][29][30][31][32][34]
- Address Mapping for IBANK_POS = 0 and EBANK_POS = 0: [35][36]
- Address Mapping for IBANK_POS = 1 and EBANK_POS = 0: [37][38]
- Address Mapping for IBANK_POS = 2 and EBANK_POS = 0: [39][40]
- Address Mapping for IBANK_POS = 3 and EBANK_POS = 0: [41]
- Address Mapping for IBANK_POS = 0 and EBANK_POS = 1: [42]
- Address Mapping for IBANK_POS = 1 and EBANK_POS = 1: [43]
- Address Mapping for IBANK_POS = 2 and EBANK_POS = 1: [44]
- Address Mapping for IBANK_POS = 3 and EBANK_POS = 1: [45]
- Error Correction And Detection Feature: [46]
- Global Initialization: [47][48][49][50][51][52][53]
- EMIF Register Summary: [54]
- EMIF Register Description: [55]

Table 10-50. EMIF_SDRAM_CONFIG_2

Address Offset	0x0000 000C		
Physical Address	0x4C00 000C	**Instance**	EMIF
Description	SDRAM Config Register 2 **CAUTION:** This register is loaded with values by control module at device reset.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				EBANK_POS				RESERVED																							

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27	EBANK_POS	External bank position. Set to 0 to assign external bank address bits from lower OCP address. Set to 1 to assign external bank address bits from higher OCP address bits. See Section 10.2.4.12, *SDRAM Address Mapping*.	RW	0
26:0	RESERVED		R	0x00 0000

Table 10-51. Register Call Summary for Register EMIF_SDRAM_CONFIG_2

EMIF Controller

- Address Mapping for IBANK_POS = 0 and EBANK_POS = 0: [0]
- Address Mapping for IBANK_POS = 0 and EBANK_POS = 1: [1]
- Address Mapping for IBANK_POS = 1 and EBANK_POS = 1: [2]
- Address Mapping for IBANK_POS = 2 and EBANK_POS = 1: [3]
- Address Mapping for IBANK_POS = 3 and EBANK_POS = 1: [4]
- Global Initialization: [5]
- EMIF Register Summary: [6]
- EMIF Register Description: [7]

Table 10-52. EMIF_SDRAM_REFRESH_CONTROL

Address Offset	0x0000 0010		
Physical Address	0x4C00 0010	**Instance**	EMIF
Description	SDRAM Refresh Control Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INITREF_DIS	RESERVED	SRT	ASR	RESERVED	PASR			RESERVED								REFRESH_RATE															

Bits	Field Name	Description	Type	Reset
31	INITREF_DIS	Initialization and Refresh disable. When set to 1, EMIF will disable SDRAM initialization and refreshes, but will carry out SDRAM write/read transactions.	RW	1
30	RESERVED		R	0
29	SRT	DDR3 Self Refresh temperature range. Set to 0 for normal operating temperature range and set to 1 for extended operating temperature range when the ASR field is set to 0. This bit must be set to 0 if the ASR field is set to 1. A write to this field will cause the EMIF to start the SDRAM initialization sequence.	RW	0
28	ASR	DDR3 Auto Self Refresh enable. Set to 1 for auto Self Refresh enable. Set to 0 for manual Self Refresh reference indicated by the SRT field. A write to this field will cause the EMIF to start the SDRAM initialization sequence.	RW	0
27	RESERVED		R	0
26:24	PASR	Partial Array Self Refresh. These bits get loaded into the Extended Mode Register of DDR3 during initialization. For DDR3, set to 0 for full array, set to 1 or 5 for 1/2 array, set to 2 or 6 for 1/4 array, set to 3 or 7 for 1/8 array, and set to 4 for 3/4 array to be refreshed. All other values are reserved. A write to this field will cause the EMIF to start the SDRAM initialization sequence.	RW	0x0
23:16	RESERVED		R	0x00
15:0	REFRESH_RATE	Refresh Rate. Value in this field is used to define the rate at which connected SDRAM devices will be refreshed. SDRAM refresh rate = REFRESH_RATE / EMIF_PHY_FCLK. A 533-MHz DDR clock rate system that requires a 7.8 µs refresh rate would need 7.8 × 533 = 4157 or 0x103D value to be written. To avoid lock-up situations, the programmer must not program REFRESH_RATE < (6 × EMIF_SDRAM_TIMING_3[12:4] T_RFC). **NOTE:** The SDRAM refresh rate can be changed on-the-fly by writing to this field. When changing the SDRAM refresh rate all timing parameters that use the refresh rate value have to be recalculated. For example, tRASmax specified in the EMIF_SDRAM_TIMING_3[3:0] T_RAS_MAX field must be recalculated.	RW	0x0000

Table 10-53. Register Call Summary for Register EMIF_SDRAM_REFRESH_CONTROL

EMIF Controller

- Arbitration of Commands in the Command FIFO: [0]
- Reset: [1]
- Self-Refresh Mode: [2]
- SDRAM Refresh Scheduling: [3][4][5]
- DDR2 SDRAM Initialization: [6][7][8][9][10]
- DDR3/DDR3L SDRAM Initialization: [11][12][13][14][15][16][17][18]
- LPDDR2 SDRAM Initialization: [19][20][21]
- LPDDR2 Temperature Monitoring: [22]
- Global Initialization: [23][24][25][26][27][28][29][30][31][32][33][34][35]
- EMIF Register Summary: [36]
- EMIF Register Description: [37][38][39][40][41][42][43]

Table 10-54. EMIF_SDRAM_REFRESH_CONTROL_SHADOW

Address Offset	0x0000 0014		
Physical Address	0x4C00 0014	**Instance**	EMIF
Description	SDRAM Refresh Control Shadow Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																REFRESH_RATE_SHDW															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Reserved	R	0x0000
15:0	REFRESH_RATE_SHDW	Shadow field for REFRESH_RATE. This field is loaded into EMIF_SDRAM_REFRESH_CONTROL[15:0] REFRESH_RATE field when SIdleAck is asserted.	RW	0x0000

Table 10-55. Register Call Summary for Register EMIF_SDRAM_REFRESH_CONTROL_SHADOW

EMIF Controller

- Global Initialization: [0][1][2][3][4]
- EMIF Register Summary: [5]

Table 10-56. EMIF_SDRAM_TIMING_1

Address Offset	0x0000 0018		
Physical Address	0x4C00 0018	**Instance**	EMIF
Description	SDRAM Timing 1 Register. If this register is byte written, care must be taken that all the fields are written before performing any accesses to the SDRAM.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
T_RTW			T_RP				T_RCD				T_WR				T_RAS					T_RC						T_RRD			T_WTR		

Bits	Field Name	Description	Type	Reset
31:29	T_RTW	Minimum number of DDR clock cycles between Read to Write data phases, minus one.	RW	0x0
28:25	T_RP	Minimum number of DDR clock cycles from Precharge to Activate or Refresh, minus one.	RW	0x0
24:21	T_RCD	Minimum number of DDR clock cycles from Activate to Read or Write, minus one.	RW	0x0

Bits	Field Name	Description	Type	Reset
20:17	T_WR	Minimum number of DDR clock cycles from last Write transfer to Precharge, minus one.	RW	0x0
16:12	T_RAS	Minimum number of DDR clock cycles from Activate to Precharge, minus one. T_RAS value needs to be bigger than or equal to T_RDC value.	RW	0x00
11:6	T_RC	Minimum number of DDR clock cycles from Activate to Activate, minus one.	RW	0x00
5:3	T_RRD	Minimum number of DDR clock cycles from Activate to Activate for a different bank, minus one. For an 8-bank, this field must be equal to ((tFAW / (4 × tCK)) - 1).	RW	0x0
2:0	T_WTR	Minimum number of DDR clock cycles from last Write to Read, minus one.	RW	0x0

Table 10-57. Register Call Summary for Register EMIF_SDRAM_TIMING_1

EMIF Controller

- DDR2 SDRAM Initialization: [0][1][2]
- DDR3/DDR3L SDRAM Initialization: [3][4][5]
- Turnaround Time: [6][7]
- Global Initialization: [8][9][10]
- EMIF Register Summary: [11]
- EMIF Register Description: [12][13][14][15][16][17][18][19]

Table 10-58. EMIF_SDRAM_TIMING_1_SHADOW

Address Offset	0x0000 001C		
Physical Address	0x4C00 001C	**Instance**	EMIF
Description	SDRAM Timing 1 Shadow Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
T_RTW_SHDW			T_RP_SHDW					T_RCD_SHDW			T_WR_SHDW					T_RAS_SHDW				T_RC_SHDW				T_RRD_SHDW			T_WTR_SHDW			

Bits	Field Name	Description	Type	Reset
31:29	T_RTW_SHDW	Shadow field for T_RTW. This field is loaded into EMIF_SDRAM_TIMING_1[31:29] T_RTW field when SIdleAck is asserted.	RW	0x0
28:25	T_RP_SHDW	Shadow field for T_RP. This field is loaded into EMIF_SDRAM_TIMING_1[28:25] T_RP field when SIdleAck is asserted.	RW	0x0
24:21	T_RCD_SHDW	Shadow field for T_RCD. This field is loaded into EMIF_SDRAM_TIMING_1[24:21] T_RCD field when SIdleAck is asserted.	RW	0x0
20:17	T_WR_SHDW	Shadow field for T_WR. This field is loaded into EMIF_SDRAM_TIMING_1[20:17] T_WR field when SIdleAck is asserted.	RW	0x0
16:12	T_RAS_SHDW	Shadow field for T_RAS. This field is loaded into EMIF_SDRAM_TIMING_1[16:12] T_RAS field when SIdleAck is asserted.	RW	0x00
11:6	T_RC_SHDW	Shadow field for T_RC. This field is loaded into EMIF_SDRAM_TIMING_1[11:6] T_RC field when SIdleAck is asserted.	RW	0x00

Bits	Field Name	Description	Type	Reset
5:3	T_RRD_SHDW	Shadow field for T_RRD. This field is loaded into EMIF_SDRAM_TIMING_1[5:3] T_RRD field when SIdleAck is asserted.	RW	0x0
2:0	T_WTR_SHDW	Shadow field for T_WTR. This field is loaded into EMIF_SDRAM_TIMING_1[2:0] T_WTR field when SIdleAck is asserted.	RW	0x0

Table 10-59. Register Call Summary for Register EMIF_SDRAM_TIMING_1_SHADOW

EMIF Controller
- Global Initialization: [0]
- EMIF Register Summary: [1]

Table 10-60. EMIF_SDRAM_TIMING_2

Address Offset	0x0000 0020		
Physical Address	0x4C00 0020	**Instance**	EMIF
Description	SDRAM Timing 2 Register. If this register is byte written, care must be taken that all the fields are written before performing any accesses to the SDRAM.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	T_XP			RESERVED				T_XSNR								T_XSRD								T_RTP					T_CKE		

Bits	Field Name	Description	Type	Reset
31	RESERVED	Reserved	R	0
30:28	T_XP	Minimum number of DDR clock cycles from power-down exit to any command other than a read command, minus one.	RW	0x0
27:25	RESERVED	Reserved	RW	0x0
24:16	T_XSNR	Minimum number of DDR clock cycles from Self-Refresh exit to any command other than a Read command, minus one.	RW	0x000
15:6	T_XSRD	Minimum number of DDR clock cycles from Self-Refresh exit to a Read command, minus one.	RW	0x000
5:3	T_RTP	Minimum number of DDR clock cycles for the last read command to a Precharge command, minus one.	RW	0x0
2:0	T_CKE	Minimum number of DDR clock cycles between CKE pin changes, minus one.	RW	0x0

Table 10-61. Register Call Summary for Register EMIF_SDRAM_TIMING_2

EMIF Controller
- Power-Down Mode: [0][1]
- Self-Refresh Mode: [2][3][4][5][6][7][8]
- Global Initialization: [9][10][11]
- EMIF Register Summary: [12]
- EMIF Register Description: [13][14][15][16][17]

Table 10-62. EMIF_SDRAM_TIMING_2_SHADOW

Address Offset	0x0000 0024		
Physical Address	0x4C00 0024	**Instance**	EMIF
Description	SDRAM Timing 2 Shadow Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED			T_XP_SHDW				RESERVED								T_XSNR_SHDW								T_XSRD_SHDW						T_RTP_SHDW				T_CKE_SHDW		

Bits	Field Name	Description	Type	Reset
31	RESERVED	Reserved	R	0
30:28	T_XP_SHDW	Shadow field for T_XP. This field is loaded into EMIF_SDRAM_TIMING_2[30:28] T_XP field when SIdleAck is asserted.	RW	0x0
27:25	RESERVED	Reserved	RW	0x0
24:16	T_XSNR_SHDW	Shadow field for T_XSNR. This field is loaded into EMIF_SDRAM_TIMING_2[24:16] T_XSNR field when SIdleAck is asserted.	RW	0x000
15:6	T_XSRD_SHDW	Shadow field for T_XSRD. This field is loaded into EMIF_SDRAM_TIMING_2[15:6] T_XSRD field when SIdleAck is asserted.	RW	0x000
5:3	T_RTP_SHDW	Shadow field for T_RTP. This field is loaded into EMIF_SDRAM_TIMING_2[5:3] T_RTP field when SIdleAck is asserted.	RW	0x0
2:0	T_CKE_SHDW	Shadow field for T_CKE. This field is loaded into EMIF_SDRAM_TIMING_2[2:0] T_CKE field when SIdleAck is asserted.	RW	0x0

Table 10-63. Register Call Summary for Register EMIF_SDRAM_TIMING_2_SHADOW

EMIF Controller

- Global Initialization: [0]
- EMIF Register Summary: [1]

Table 10-64. EMIF_SDRAM_TIMING_3

Address Offset	0x0000 0028		
Physical Address	0x4C00 0028	**Instance**	EMIF
Description	SDRAM Timing 3 Register. If this register is byte written, care must be taken that all the fields are written before performing any accesses to the SDRAM.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
T_PDLL_UL				RESERVED				T_CKESR			ZQ_ZQCS					T_TDQSCKMAX			T_RFC										T_RAS_MAX						

Bits	Field Name	Description	Type	Reset
31:28	T_PDLL_UL	Minimum number of DDR clock cycles for PHY DLL to unlock. A value of N will be equal to N x 128 clocks.	RW	0x0
27:24	RESERVED		R	0x0
23:21	T_CKESR	Minimum number of DDR clock cycles for which SDRAM must remain in Self Refresh, minus one.	RW	0x0
20:15	ZQ_ZQCS	Number of DDR clock cycles for a ZQCS command, minus one.	RW	0x00
14:13	T_TDQSCKMAX	Number of DDR clock that satisfies tDQSCKmax for LPDDR2, minus one.	RW	0x0
12:4	T_RFC	Minimum number of DDR clock cycles from Refresh or Load Mode to Refresh or Activate, minus one.	RW	0x000
3:0	T_RAS_MAX	Maximum number of REFRESH_RATE intervals from Activate to Precharge command. This field must be equal to ((tRASmax / tREFI)-1) rounded down to the next lower integer. Value for T_RAS_MAX can be calculated as follows: If tRASmax = 120 us and tREFI = 15.7 us, then T_RAS_MAX = ((120/15.7)-1) = 6.64. Round down to the next lower integer. Therefore, the programmed value must be 6.	RW	0x0

Table 10-65. Register Call Summary for Register EMIF_SDRAM_TIMING_3

EMIF Controller

- SDRAM Refresh Scheduling: [0]
- Turnaround Time:
- Output Impedance Calibration: [3][4]
- Global Initialization: [5][6][7]
- EMIF Register Summary: [8]
- EMIF Register Description: [9][10][11][12][13][14][15][16][17][18]

Table 10-66. EMIF_SDRAM_TIMING_3_SHADOW

Address Offset	0x0000 002C		
Physical Address	0x4C00 002C	**Instance**	EMIF
Description	SDRAM Timing 3 Shadow Register		
Type	RW		

31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
T_PDLL_UL_SHDW	T_CSTA_SHDW	T_CKESR_SHDW	ZQ_ZQCS_SHDW	T_TDQSCKMAX_SHDW	T_RFC_SHDW		T_RAS_MAX_SHDW

Bits	Field Name	Description	Type	Reset
31:28	T_PDLL_UL_SHDW	Shadow field for T_PDLL_UL. This field is loaded into T_PDLL_UL field in EMIF_SDRAM_TIMING_3 register when SIdleAck is asserted.	RW	0x0
27:24	T_CSTA_SHDW	Shadow field for T_CSTA. This field is loaded into T_CSTA field in EMIF_SDRAM_TIMING_3 register when SIdleAck is asserted.	RW	0x0
23:21	T_CKESR_SHDW	Shadow field for T_CKESR. This field is loaded into T_CKESR field in EMIF_SDRAM_TIMING_3 register when SIdleAck is asserted.	RW	0x0

Bits	Field Name	Description	Type	Reset
20:15	ZQ_ZQCS_SHDW	Shadow field for ZQ_ZQCS. This field is loaded into ZQ_ZQCS field in EMIF_SDRAM_TIMING_3 register when SIdleAck is asserted.	RW	0x00
14:13	T_TDQSCKMAX_SHDW	Shadow field for T_TDQSCKMAX. This field is loaded into EMIF_SDRAM_TIMING_3[14:13] T_TDQSCKMAX field when SIdleAck is asserted.	RW	0x0
12:4	T_RFC_SHDW	Shadow field for T_RFC. This field is loaded into EMIF_SDRAM_TIMING_3[12:4] T_RFC when SIdleAck is asserted.	RW	0x000
3:0	T_RAS_MAX_SHDW	Shadow field for T_RAS_MAX. This field is loaded into EMIF_SDRAM_TIMING_3[3:0] T_RAS_MAX field when SIdleAck is asserted.	RW	0x0

Table 10-67. Register Call Summary for Register EMIF_SDRAM_TIMING_3_SHADOW

EMIF Controller
- Global Initialization: [0]
- EMIF Register Summary: [1]

Table 10-68. EMIF_LPDDR2_NVM_TIMING

Address Offset	0x0000 0030		
Physical Address	0x4C00 0030	**Instance**	EMIF
Description	**NOTE: This register is not supported. It is kept only for code compatibility.**		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															

Bits	Field Name	Description	Type	Reset
31:0	RESERVED	Reserved	RW	0x0

Table 10-69. Register Call Summary for Register EMIF_LPDDR2_NVM_TIMING

EMIF Controller
- Global Initialization: [0]
- EMIF Register Summary: [1]

Table 10-70. EMIF_LPDDR2_NVM_TIMING_SHADOW

Address Offset	0x0000 0034		
Physical Address	0x4C00 0034	**Instance**	EMIF
Description	**NOTE: This register is not supported. It is kept only for code compatibility.**		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															

Bits	Field Name	Description	Type	Reset
31:0	RESERVED	Reserved	RW	0x0

Table 10-71. Register Call Summary for Register EMIF_LPDDR2_NVM_TIMING_SHADOW

EMIF Controller

- Global Initialization: [0]
- EMIF Register Summary: [1]

Table 10-72. EMIF_POWER_MANAGEMENT_CONTROL

Address Offset	0x0000 0038		
Physical Address	0x4C00 0038	**Instance**	EMIF
Description	Power Management Control Register. Updating the *_TIM fields must be followed by at least one access to SDRAM for the new value to take an effect.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																PD_TIM				DPD_EN	LP_MODE			SR_TIM				RESERVED			

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:12	PD_TIM	Power Management timer for Power-Down. The EMIF will put the external SDRAM in Power-Down mode after the EMIF is idle for these number of DDR clock cycles and if LP_MODE field is set to 4. Set to 0 to immediately enter Power-Down mode. Set to 1 for 16 clocks, set to 2 for 32 clocks, set to 3 for 64 clocks, set to 4 for 128 clocks, set to 5 for 256 clocks, set to 6 for 512 clocks, set to 7 for 1024 clocks, set to 8 for 2048 clocks, set to 9 for 4096 clocks, set to 10 for 8192 clocks, set to 11 for 16384 clocks, set to 12 for 32768 clocks, set to 13 for 65536 clocks, set to 14 for 131072 clocks, and set to 15 for 262144 clocks.	RW	0x0
11	DPD_EN	Deep Power Down enable. Set to 0 for normal operation. Set to 1 to enter deep power-down mode. This mode will override the LP_MODE field setting.	RW	0
10:8	LP_MODE	Automatic Power Management enable. 0x0: Disable automatic power management 0x1: Reserved 0x2: Self Refresh mode 0x3: Disable automatic power management 0x4: Power-Down mode All other values disable automatic power management.	RW	0x0
7:4	SR_TIM	Power Management timer for Self Refresh. The EMIF will put the external SDRAM in Self Refresh mode after the EMIF is idle for these number of DDR clock cycles and if LP_MODE field is set to 2. Set to 0 to immediately enter Self Refresh mode. Set to 1 for 16 clocks, set to 2 for 32 clocks, set to 3 for 64 clocks, set to 4 for 128 clocks, set to 5 for 256 clocks, set to 6 for 512 clocks, set to 7 for 1024 clocks, set to 8 for 2048 clocks, set to 9 for 4096 clocks, set to 10 for 8192 clocks, set to 11 for 16384 clocks, set to 12 for 32768 clocks, set to 13 for 65536 clocks, set to 14 for 131072 clocks, and set to 15 for 262144 clocks.	RW	0x0
3:0	RESERVED		RW	0x0

Table 10-73. Register Call Summary for Register EMIF_POWER_MANAGEMENT_CONTROL

EMIF Controller

- Reset: [0]
- Power-Down Mode: [1][2][3]
- LPDDR2 Deep Power-Down Mode: [4][5][6][7][8]
- Self-Refresh Mode: [9][10][11]
- Global Initialization: [12][13][14][15][16]
- Operational Modes Configuration: [17][18][19][20][21][22][23][24][25][26][27][28][29][30][31]
- EMIF Register Summary: [32]
- EMIF Register Description: [33][34]

Table 10-74. EMIF_POWER_MANAGEMENT_CONTROL_SHADOW

Address Offset	0x0000 003C		
Physical Address	0x4C00 003C	**Instance**	EMIF
Description	Power Management Control Shadow Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																PD_TIM_SHDW				RESERVED				SR_TIM_SHDW				RESERVED			

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:12	PD_TIM_SHDW	Shadow field for PD_TIM. This field is loaded into PD_TIM field in EMIF_POWER_MANAGEMENT_CONTROL register when SIdleAck is asserted.	RW	0x0
11:8	RESERVED		R	0x0
7:4	SR_TIM_SHDW	Shadow field for SR_TIM. This field is loaded into SR_TIM field in EMIF_POWER_MANAGEMENT_CONTROL register when SIdleAck is asserted.	RW	0x0
3:0	RESERVED		RW	0x0

Table 10-75. Register Call Summary for Register EMIF_POWER_MANAGEMENT_CONTROL_SHADOW

EMIF Controller

- Global Initialization: [0]
- Operational Modes Configuration: [1][2][3][4][5][6]
- EMIF Register Summary: [7]

Table 10-76. EMIF_LPDDR2_MODE_REG_DATA

Address Offset	0x0000 0040		
Physical Address	0x4C00 0040	**Instance**	EMIF
Description	LPDDR2 Mode Reg Data Register A write to this register will cause a Mode Register write command to be sent to the LPDDR2 device with write data as specified in the VALUE_0 field. The address and chip select are taken from the EMIF_LPDDR2_MODE_REG_CONFIG register. A read to this register will cause a Mode Register read command to be sent to the LPDDR2 device. The address and chip select are taken from the EMIF_LPDDR2_MODE_REG_CONFIG register. The read data will appear in VALUE_0 field.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								VALUE_0							

Bits	Field Name	Description	Type	Reset
31:7	RESERVED		R	0x000 0000
6:0	VALUE_0	Mode register value.	RW	0x00

Table 10-77. Register Call Summary for Register EMIF_LPDDR2_MODE_REG_DATA

EMIF Controller

- LPDDR2 SDRAM Initialization: [0]
- Global Initialization: [1][2][3][4][5][6]
- EMIF Register Summary: [7]
- EMIF Register Description: [8]

Table 10-78. EMIF_LPDDR2_MODE_REG_CONFIG

Address Offset	0x0000 0050		
Physical Address	0x4C00 0050	**Instance**	EMIF
Description	LPDDR2 Mode Reg Config Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CS	REFRESH_EN	RESERVED																						ADDRESS							

Bits	Field Name	Description	Type	Reset
31	CS	Chip select to issue mode register command. Set to 0 for CSN0.	RW	0
30	REFRESH_EN	Refresh Enable after MRW write. If a EMIF_LPDDR2_MODE_REG_DATA register write occurs with this bit set to 1, the refresh operations will commence.	RW	0
29:8	RESERVED	Reserved	R	0x00 0000
7:0	ADDRESS	Mode register address.	RW	0x00

Table 10-79. Register Call Summary for Register EMIF_LPDDR2_MODE_REG_CONFIG

EMIF Controller

- LPDDR2 SDRAM Initialization: [0][1]
- Global Initialization: [2][3][4][5][6][7]
- EMIF Register Summary: [8]
- EMIF Register Description: [9][10]

Table 10-80. EMIF_OCP_CONFIG

Address Offset	0x0000 0054		
Physical Address	0x4C00 0054	**Instance**	EMIF
Description	OCP Config Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				SYS_THRESH_MAX				RESERVED																							

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:24	SYS_THRESH_MAX	System OCP Threshold Maximum. The number of commands the system interface can consume in the command FIFO. The value is used to determine when to stop future request, writing a zero will reserve no space for the associated interface. In the event the value is set to zero and a request is seen for that interface, the command FIFO will assume a value of 1.	RW	0x7
23:0	RESERVED		R	0x770000

Table 10-81. Register Call Summary for Register EMIF_OCP_CONFIG

EMIF Controller

- FIFO Description: [0]
- Global Initialization: [1][2]
- EMIF Register Summary: [3]

Table 10-82. EMIF_OCP_CONFIG_VALUE_1

Address Offset	0x0000 0058		
Physical Address	0x4C00 0058	**Instance**	EMIF
Description	OCP Config Value 1 Register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SYS_BUS_WIDTH								RESERVED								WR_FIFO_DEPTH								CMD_FIFO_DEPTH							

Bits	Field Name	Description	Type	Reset
31:30	SYS_BUS_WIDTH	System OCP data bus width 0 = 32-bit wide, 1 = 64-bit wide, 2 = 128-bit wide, 3 = Reserved	R	0x2
29:16	RESERVED		R	0x1000
15:8	WR_FIFO_DEPTH	Write Data FIFO depth	R	0x19
7:0	CMD_FIFO_DEPTH	Command FIFO depth	R	0x0A

Table 10-83. Register Call Summary for Register EMIF_OCP_CONFIG_VALUE_1

EMIF Controller
- EMIF Register Summary: [0]

Table 10-84. EMIF_OCP_CONFIG_VALUE_2

Address Offset	0x0000 005C		
Physical Address	0x4C00 005C	**Instance**	EMIF
Description	OCP Config Value 2 Register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								RREG_FIFO_DEPTH								RSD_FIFO_DEPTH								RCMD_FIFO_DEPTH							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED	Reserved	R	0x00
23:16	RREG_FIFO_DEPTH	Register Read Data FIFO depth	R	0x04
15:8	RSD_FIFO_DEPTH	SDRAM Read Data FIFO depth	R	0x27
7:0	RCMD_FIFO_DEPTH	Read Command FIFO depth	R	0x27

Table 10-85. Register Call Summary for Register EMIF_OCP_CONFIG_VALUE_2

EMIF Controller
- EMIF Register Summary: [0]

Table 10-86. EMIF_IODFT_TLGC

Address Offset	0x0000 0060		
Physical Address	0x4C00 0060	**Instance**	EMIF
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RESERVED	RESERVED	RESERVED	RESERVED	RESERVED	RESET_PHY	RESERVED	RESERVED	RESERVED		RESERVED		RESERVED			RESERVED

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Reserved. This field must not be modified.	RW	0x0
15	RESERVED	Reserved	R	0x0
14	RESERVED	Reserved. This bit must not be modified.	RW	0x0

Bits	Field Name	Description	Type	Reset
13	RESERVED	Reserved. This bit must not be modified.	RW	0x1
12	RESERVED	Reserved. This bit must not be modified.	RW	0x0
11	RESERVED	Reserved	R	0x0
10	RESET_PHY	Reset the DDR PHY. Writing 1 to this bit resets the DDR PHY. This bit will self clear to 0.	RW	0x0
9	RESERVED	Reserved	R	0x0
8	RESERVED	Reserved. This bit must not be modified.	RW	0x0
7:6	RESERVED	Reserved	R	0x0
5:4	RESERVED	Reserved. This field must not be modified.	RW	0x1
3:1	RESERVED	Reserved. This field must not be modified.	RW	0x0
0	RESERVED	Reserved. This bit must not be modified.	RW	0x1

Table 10-87. Register Call Summary for Register EMIF_IODFT_TLGC

EMIF Controller
- Global Initialization: [0][1][2][3][4]
- EMIF Register Summary: [5]

Table 10-88. EMIF_PERFORMANCE_COUNTER_1

Address Offset	0x0000 0080		
Physical Address	0x4C00 0080	**Instance**	EMIF
Description	Performance Counter 1 Register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
COUNTER1																															

Bits	Field Name	Description	Type	Reset
31:0	COUNTER1	32-bit counter that can be configured as specified in the EMIF_PERFORMANCE_COUNTER_CONFIG register and EMIF_PERFORMANCE_COUNTER_MASTER_REGION_SELECT register.	R	0x0000 0000

Table 10-89. Register Call Summary for Register EMIF_PERFORMANCE_COUNTER_1

EMIF Controller
- Performance Counters: [0]
- Performance Counters General Examples: [1][2][3][4]
- EMIF Register Summary: [5]
- EMIF Register Description: [6][7][8][9]

Table 10-90. EMIF_PERFORMANCE_COUNTER_2

Address Offset	0x0000 0084		
Physical Address	0x4C00 0084	**Instance**	EMIF
Description	Performance Counter 2 Register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
COUNTER2																															

Bits	Field Name	Description	Type	Reset
31:0	COUNTER2	32-bit counter that can be configured as specified in the EMIF_PERFORMANCE_COUNTER_CONFIG register and EMIF_PERFORMANCE_COUNTER_MASTER_REGION_SELECT register.	R	0x0000 0000

Table 10-91. Register Call Summary for Register EMIF_PERFORMANCE_COUNTER_2

EMIF Controller

- Performance Counters: [0]
- Performance Counters General Examples: [1][2]
- EMIF Register Summary: [3]
- EMIF Register Description: [4][5][6][7]

Table 10-92. EMIF_PERFORMANCE_COUNTER_CONFIG

Address Offset	0x0000 0088		
Physical Address	0x4C00 0088	**Instance**	EMIF
Description	Performance Counter Config Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNTR2_MCONNID_EN	CNTR2_REGION_EN	RESERVED										CNTR2_CFG				CNTR1_MCONNID_EN	CNTR1_REGION_EN	RESERVED										CNTR1_CFG			

Bits	Field Name	Description	Type	Reset
31	CNTR2_MCONNID_EN	MConnID filter enable for EMIF_PERFORMANCE_COUNTER_2 register.	RW	0
30	CNTR2_REGION_EN	Chip Select filter enable for EMIF_PERFORMANCE_COUNTER_2 register.	RW	0
29:20	RESERVED	Reserved for future use	R	0x000
19:16	CNTR2_CFG	Filter configuration for EMIF_PERFORMANCE_COUNTER_2. Refer to Table 10-30 for details.	RW	0x1
15	CNTR1_MCONNID_EN	MConnID filter enable for EMIF_PERFORMANCE_COUNTER_1 register.	RW	0
14	CNTR1_REGION_EN	Chip Select filter enable for EMIF_PERFORMANCE_COUNTER_1 register.	RW	0
13:4	RESERVED	Reserved for future use	R	0x000

Bits	Field Name	Description	Type	Reset
3:0	CNTR1_CFG	Filter configuration for EMIF_PERFORMANCE_COUNTER_1. Refer to Table 10-30 for details.	RW	0x0

Table 10-93. Register Call Summary for Register EMIF_PERFORMANCE_COUNTER_CONFIG

EMIF Controller

- Performance Counters: [0][1][2][3][4]
- Performance Counters General Examples: [5][6][7][8][9][10][11][12]
- EMIF Register Summary: [13]
- EMIF Register Description: [14][15]

Table 10-94. EMIF_PERFORMANCE_COUNTER_MASTER_REGION_SELECT

Address Offset	0x0000 008C		
Physical Address	0x4C00 008C	**Instance**	EMIF
Description	Performance Counter Master Region Select Register The values programmed into the MCONNIDx fields are those in the *ConnID Values* table in Chapter 9, *Interconnect*.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MCONNID2								RESERVED								MCONNID1								RESERVED							

Bits	Field Name	Description	Type	Reset
31:24	MCONNID2	MConnID for EMIF_PERFORMANCE_COUNTER_2 register.	RW	0x0
23:16	RESERVED	Reserved	R	0x0
15:8	MCONNID1	MConnID for EMIF_PERFORMANCE_COUNTER_1 register.	RW	0x0
7:0	RESERVED	Reserved	R	0x0

Table 10-95. Register Call Summary for Register EMIF_PERFORMANCE_COUNTER_MASTER_REGION_SELECT

EMIF Controller

- Performance Counters: [0]
- Performance Counters General Examples: [1][2][3]
- EMIF Register Summary: [4]
- EMIF Register Description: [5][6]

Table 10-96. EMIF_PERFORMANCE_COUNTER_TIME

Address Offset	0x0000 0090		
Physical Address	0x4C00 0090	**Instance**	EMIF
Description	Performance Counter Time Register. This is a free running counter.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TOTAL_TIME																															

Bits	Field Name	Description	Type	Reset
31:0	TOTAL_TIME	32-bit counter that continuously counts number for EMIF_FICLK clock cycles elapsed after EMIF is brought out of reset.	R	0x0000 0000

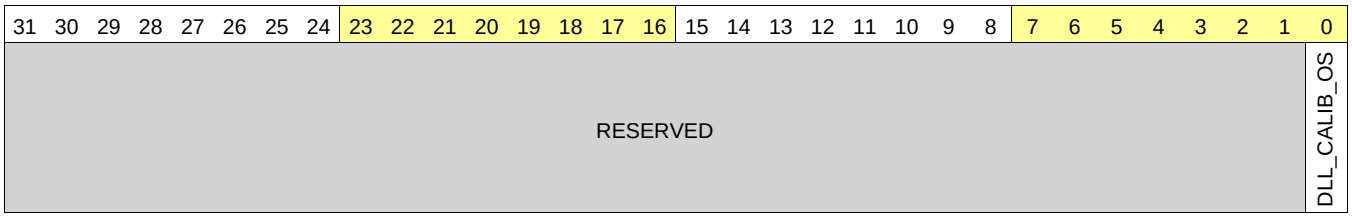
Table 10-97. Register Call Summary for Register EMIF_PERFORMANCE_COUNTER_TIME

EMIF Controller

- EMIF Register Summary: [0]

Table 10-98. EMIF_MISC_REG

Address Offset	0x0000 0094		
Physical Address	0x4C00 0094	**Instance**	EMIF
Description			
Type	RW		



Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved	R	0x0
0	DLL_CALIB_OS	Phy_dll_calib one shot : Setting bit to 1 generates a phy_pll_calib pulse. Bit is self cleared when pll_calib gets generated and ack_wait has been satisfied. Software can poll to confirm completion. Uses the EMIF_DLL_CALIB_CTRL[19:16] ACK_WAIT bit field for time to wait after firing off the phy_dll_calib.	RW	0x0

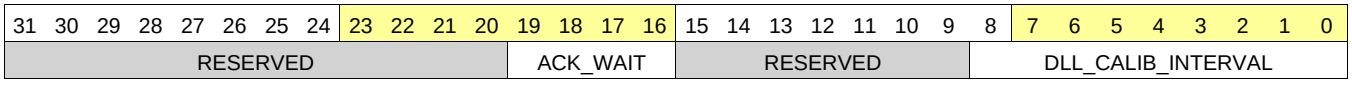
Table 10-99. Register Call Summary for Register EMIF_MISC_REG

EMIF Controller

- PHY DLL Calibration: [0]
- EMIF Register Summary: [1]
- EMIF Register Description: [2]

Table 10-100. EMIF_DLL_CALIB_CTRL

Address Offset	0x0000 0098		
Physical Address	0x4C00 0098	**Instance**	EMIF
Description	Control register to force idle window time to generate a phy_dll_calib that can be used for updating PHY DLLs during voltage ramps. NOTE: Should always be loaded via the shadow register.		
Type	RW		



Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x000
19:16	ACK_WAIT	The ack_wait determines the required wait time after a phy_dll_calib is generated before another command can be sent. Value program is in terms of EMIF_FICLK cycle count. **CAUTION:** 5 must be the minimum value ever programmed.	RW	0x9
15:9	RESERVED		R	0x00

Bits	Field Name	Description	Type	Reset
8:0	DLL_CALIB_INTERVAL	This field determines the interval between phy_dll_calib generation. This value is multiplied by a precounter of 16 EMIF_FICLK cycles. Program this field one less the value you are targeting; program 1 to achieve interval of 2 (minimum interval supported). Programming zero turns off function. Note the final intervals between dll_calib generation is also a function of ACK_WAIT. Final periodic interval is calculated by: ((DLL_CALIB_INTERVAL + 1) × 16) + ACK_WAIT	RW	0x000

Table 10-101. Register Call Summary for Register EMIF_DLL_CALIB_CTRL

EMIF Controller
- PHY DLL Calibration: [0]
- Global Initialization: [1][2][3][4]
- EMIF Register Summary: [5]
- EMIF Register Description: [6][7][8]

Table 10-102. EMIF_DLL_CALIB_CTRL_SHADOW

Address Offset	0x0000 009C		
Physical Address	0x4C00 009C	**Instance**	EMIF
Description	Read Idle Control Shadow Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												ACK_WAIT_SHDW				RESERVED								DLL_CALIB_INTERVAL_SHDW							

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x000
19:16	ACK_WAIT_SHDW	Shadow field for ACK_WAIT. This field is loaded into ACK_WAIT field in EMIF_DLL_CALIB_CTRL register when SIdleAck is asserted	RW	0x9
15:9	RESERVED		R	0x00
8:0	DLL_CALIB_INTERVAL_SHDW	Shadow field for DLL_CALIB_INTERVAL. This field is loaded into DLL_CALIB_INTERVAL field in the EMIF_DLL_CALIB_CTRL register when SIdleAck is asserted	RW	0x000

Table 10-103. Register Call Summary for Register EMIF_DLL_CALIB_CTRL_SHADOW

EMIF Controller
- Global Initialization: [0]
- EMIF Register Summary: [1]

Table 10-104. EMIF_END_OF_INTERRUPT

Address Offset	0x0000 00A0		
Physical Address	0x4C00 00A0	**Instance**	EMIF
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															EOI

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	EOI	Software End Of Interrupt (EOI) control. Write 0x0 for system OCP interrupt. This field always reads 0 (no EOI memory).	RW	0x0

Table 10-105. Register Call Summary for Register EMIF_END_OF_INTERRUPT

EMIF Controller
- EMIF Register Summary: [0]

Table 10-106. EMIF_SYSTEM_OCP_INTERRUPT_RAW_STATUS

Address Offset	0x0000 00A4		
Physical Address	0x4C00 00A4	**Instance**	EMIF
Description	System OCP Interrupt Raw Status Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																										ONEBIT_ECC_ERR_SYS	TWOBIT_ECC_ERR_SYS	WR_ECC_ERR_SYS	RESERVED	TA_SYS	ERR_SYS

Bits	Field Name	Description	Type	Reset
31:6	RESERVED		R	0x000 0000
5	ONEBIT_ECC_ERR_SYS	Raw status of system ECC one bit error correction interrupt.	RW	0x0
4	TWOBIT_ECC_ERR_SYS	Raw status of system ECC two bit error detection interrupt.	RW	0x0
3	WR_ECC_ERR_SYS	Raw status of system ECC Error interrupt when a memory access is made to a non-quanta aligned location.	RW	0x0
2	RESERVED		R	0x0
1	TA_SYS	Raw status of system OCP interrupt for SDRAM temperature alert. Write 1 to set the (raw) status, mostly for debug. Writing a 0 has no effect.	RW	0
0	ERR_SYS	Raw status of system OCP interrupt for command or address error. Write 1 to set the (raw) status, mostly for debug. Writing a 0 has no effect.	RW	0

Table 10-107. Register Call Summary for Register EMIF_SYSTEM_OCP_INTERRUPT_RAW_STATUS

EMIF Controller
- Interrupt Requests: [0][1][2][3][4][5][6][7][8][9][10][11][12]
- EMIF Register Summary: [13]

Table 10-108. EMIF_SYSTEM_OCP_INTERRUPT_STATUS

Address Offset	0x0000 00AC		
Physical Address	0x4C00 00AC	**Instance**	EMIF
Description	System OCP Interrupt Status Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																										ONEBIT_ECC_ERR_SYS	TWOBIT_ECC_ERR_SYS	WR_ECC_ERR_SYS	RESERVED	TA_SYS	ERR_SYS

Bits	Field Name	Description	Type	Reset
31:6	RESERVED		R	0x000 0000
5	ONEBIT_ECC_ERR_SYS	Enabled status of system ECC one bit error correction interrupt. Write 1 to clear the status after interrupt has been serviced (raw status gets cleared, that is even if not enabled). Writing a 0 has no effect	RW	0x0
4	TWOBIT_ECC_ERR_SYS	Enabled status of system ECC two bit error detection interrupt. Write 1 to clear the status after interrupt has been serviced (raw status gets cleared, that is even if not enabled). Writing a 0 has no effect.	RW	0x0
3	WR_ECC_ERR_SYS	Enabled status of system ECC Error interrupt when a memory access is made to a non-quanta aligned location. Write 1 to clear the status after interrupt has been serviced (raw status gets cleared, that is even if not enabled). Writing a 0 has no effect.	RW	0x0
2	RESERVED		R	0x0
1	TA_SYS	Enabled status of system OCP interrupt for SDRAM temperature alert. Write 1 to clear the status after interrupt has been serviced (raw status gets cleared, that is, even if not enabled). Writing a 0 has no effect.	RW	0
0	ERR_SYS	Enabled status of system OCP interrupt interrupt for command or address error. Write 1 to clear the status after interrupt has been serviced (raw status gets cleared, that is, even if not enabled). Writing a 0 has no effect.	RW	0

Table 10-109. Register Call Summary for Register EMIF_SYSTEM_OCP_INTERRUPT_STATUS

EMIF Controller

- Interrupt Requests: [0][1][2][3][4][5][6][7][8][9][10][11]
- LPDDR2 Temperature Monitoring: [12]
- Operational Modes Configuration: [13]
- EMIF Register Summary: [14]
- EMIF Register Description: [15]

Table 10-110. EMIF_SYSTEM_OCP_INTERRUPT_ENABLE_SET

Address Offset	0x0000 00B4		
Physical Address	0x4C00 00B4	**Instance**	EMIF
Description	System OCP Interrupt Enable Set Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																										ONEBIT_ECC_ERR_SYS	TWOBIT_ECC_ERR_SYS	WR_ECC_ERR_SYS	RESERVED	EN_TA_SYS	EN_ERR_SYS

Bits	Field Name	Description	Type	Reset
31:6	RESERVED		R	0x000 0000
5	ONEBIT_ECC_ERR_SYS	Enabled status of sysem ECC one bit error correction interrupt. Writing a 1 will enable the interrupt, and set this bit as well as the corresponding Interrupt Enable Clear Register. Writing a 0 has no effect.	RW W1toSet	0x0
4	TWOBIT_ECC_ERR_SYS	Enabled status of system ECC two bit error detection interrupt. Writing a 1 will enable the interrupt, and set this bit as well as the corresponding Interrupt Enable Clear Register. Writing a 0 has no effect.	RW W1toSet	0x0
3	WR_ECC_ERR_SYS	Enabled status of system ECC Error interrupt when a memory access is made to a non-quanta aligned location. Writing a 1 will enable the interrupt, and set this bit as well as the corresponding Interrupt Enable Clear Register. Writing a 0 has no effect.	RW W1toSet	0x0
2	RESERVED		R	0x0
1	EN_TA_SYS	Enable set for system OCP interrupt for SDRAM temperature alert. . Writing a 1 will enable the interrupt, and set this bit as well as the corresponding Interrupt Enable Clear Register. Writing a 0 has no effect.	RW W1toSet	0
0	EN_ERR_SYS	Enable set for system OCP interrupt for command or address error. Writing a 1 will enable the interrupt, and set this bit as well as the corresponding Interrupt Enable Clear Register. Writing a 0 has no effect.	RW W1toSet	0

Table 10-111. Register Call Summary for Register EMIF_SYSTEM_OCP_INTERRUPT_ENABLE_SET

EMIF Controller

- Interrupt Requests: [0][1][2][3][4][5][6]
- Operational Modes Configuration: [7]
- EMIF Register Summary: [8]

Table 10-112. EMIF_SYSTEM_OCP_INTERRUPT_ENABLE_CLEAR

Address Offset	0x0000 00BC		
Physical Address	0x4C00 00BC	**Instance**	EMIF
Description	System OCP Interrupt Enable Clear Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																										ONEBIT_ECC_ERR_SYS	TWOBIT_ECC_ERR_SYS	WR_ECC_ERR_SYS	RESERVED	EN_TA_SYS	EN_ERR_SYS

Bits	Field Name	Description	Type	Reset
31:6	RESERVED		R	0x000 0000
5	ONEBIT_ECC_ERR_SYS	Enabled status of system ECC one bit error correction interrupt. Writing a 1 will disable the interrupt, and clear this bit as well as the corresponding Interrupt Enable Set Register. Writing a 0 has no effect.	RW W1toClr	0x0
4	TWOBIT_ECC_ERR_SYS	Enabled status of system ECC two bit error detection interrupt. Writing a 1 will disable the interrupt, and clear this bit as well as the corresponding Interrupt Enable Set Register. Writing a 0 has no effect.	RW W1toClr	0x0
3	WR_ECC_ERR_SYS	Enabled status of system ECC Error interrupt when a memory access is made to a non-quanta aligned location. Writing a 1 will disable the interrupt, and clear this bit as well as the corresponding Interrupt Enable Set Register. Writing a 0 has no effect.	RW W1toClr	0x0
2	RESERVED		R	0x0
1	EN_TA_SYS	Enable clear for system OCP interrupt for SDRAM temperature alert. Writing a 1 will disable the interrupt, and clear this bit as well as the corresponding Interrupt Enable Set Register. Writing a 0 has no effect.	RW W1toClr	0
0	EN_ERR_SYS	Enable clear for system OCP interrupt for command or address error. Writing a 1 will disable the interrupt, and clear this bit as well as the corresponding Interrupt Enable Set Register. Writing a 0 has no effect.	RW W1toClr	0

Table 10-113. Register Call Summary for Register EMIF_SYSTEM_OCP_INTERRUPT_ENABLE_CLEAR

EMIF Controller

- Interrupt Requests: [0][1][2][3][4][5]
- EMIF Register Summary: [6]

Table 10-114. EMIF_SDRAM_OUTPUT_IMPEDANCE_CALIBRATION_CONFIG

Address Offset	0x0000 00C8		
Physical Address	0x4C00 00C8	**Instance**	EMIF
Description	SDRAM Output Impedance Calibration Config Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	ZQ_CS0EN	RESERVED	ZQ_SFEXITEN	RESERVED								ZQ_ZQINIT_MULT		ZQ_ZQCL_MULT		ZQ_REFINTERVAL																	

Bits	Field Name	Description	Type	Reset
31	RESERVED		R	0x0
30	ZQ_CS0EN	Writing a 1 enables ZQ calibration for CSN0.	RW	0x0
29	RESERVED		R	0x0
28	ZQ_SFEXITEN	Writing a 1 enables the issuing of ZQCL on Self-Refresh, Active Power-Down, and Precharge Power-Down exit.	RW	0x0
27:20	RESERVED		R	0x00
19:18	ZQ_ZQINIT_MULT	Indicates the number of ZQCL durations that make up a ZQINIT duration, minus one.	RW	0x0
17:16	ZQ_ZQCL_MULT	Indicates the number of ZQCS intervals that make up a ZQCL duration, minus one. ZQCS interval is defined by ZQ_ZQCS in EMIF_SDRAM_TIMING_3.	RW	0x0
15:0	ZQ_REFINTERVAL	Number of refresh periods between ZQCS commands. This field supports between one refresh period to 256 ms between ZQCS calibration commands. Refresh period is defined by REFRESH_RATE in EMIF_SDRAM_REFRESH_CONTROL register.	RW	0x0000

Table 10-115. Register Call Summary for Register EMIF_SDRAM_OUTPUT_IMPEDANCE_CALIBRATION_CONFIG

EMIF Controller

- Output Impedance Calibration: [0][2][3][4]
- Global Initialization: [6][7][8][9]
- Operational Modes Configuration: [10][11][12][13][14]
- EMIF Register Summary: [15]

Table 10-116. EMIF_TEMPERATURE_ALERT_CONFIG

Address Offset	0x0000 00CC		
Physical Address	0x4C00 00CC	**Instance**	EMIF
Description	Temperature Alert Config Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	TA_CS0EN	RESERVED	TA_SFEXITEN	TA_DEVWDT		TA_DEVCNT		RESERVED								TA_REFINTERVAL															

Bits	Field Name	Description	Type	Reset
31	RESERVED		R	0x0
30	TA_CS0EN	Writing a 1 enables temperature alert polling for CSN0.	RW	0x0
29	RESERVED	Reserved	R	0x0
28	TA_SFEXITEN	Temperature Alert Poll on Self-Refresh, Active Power-Down, and Precharge Power-Down exit enable. Writing a 1 enables the issuing of a temperature alert poll on Self-Refresh exit.	RW	0x0
27:26	TA_DEVWDT	This field indicates how wide a physical device is. It is used in conjunction with the TA_DEVCNT field to determine which byte lanes contain the temperature alert info. A value of 0: 8-bit wide, 1: 16- bit wide, 2: 32-bit wide. All others are reserved. If this field is set to 1 and the TA_DEVCNT field is set to 1 the byte mask for checking will be 4'b0101.	RW	0x0
25:24	TA_DEVCNT	This field indicates which external byte lanes contain a device for temperature monitoring. A value of 0 = one device, 1 = two devices, 2 = four devices. All other reserved.	RW	0x0
23:22	RESERVED	Reserved	R	0x0
21:0	TA_REFINTERVAL	Number of refresh periods between temperature alert polls. This field supports between one refresh period to 10 seconds between temperature alert polls. Refresh period is defined by REFRESH_RATE in EMIF_SDRAM_REFRESH_CONTROL register.	RW	0x0

Table 10-117. Register Call Summary for Register EMIF_TEMPERATURE_ALERT_CONFIG

EMIF Controller

- LPDDR2 Temperature Monitoring: [0][1][2][3][4][5]
- Global Initialization: [6][7][8][9]
- Operational Modes Configuration: [10][11][12][13][14][15]
- EMIF Register Summary: [16]

Table 10-118. EMIF_OCP_ERROR_LOG

Address Offset	0x0000 00D0		
Physical Address	0x4C00 00D0	**Instance**	EMIF
Description	OCP Error Log Register. This register is overwritten by any first error transaction once after the interrupt is serviced and cleared by writing 0x1 to the EMIF_SYSTEM_OCP_INTERRUPT_STATUS[0] ERR_SYS bit.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																		MBURSTSEQ			MCMD			MCONNID							

Bits	Field Name	Description	Type	Reset
31:14	RESERVED	Reserved for future use.	R	0x0000
13:11	MBURSTSEQ	Addressing mode of the first errored transaction. (see Section 9.2.1, *L3_MAIN Interconnect* for more information)	R	0x0
10:8	MCMD	Command type of the first errored transaction. (see Section 9.2.1, *L3_MAIN Interconnect* for more information)	R	0x0

Bits	Field Name	Description	Type	Reset
7:0	MCONNID	Connection ID of the first errored transaction.	R	0x00

Table 10-119. Register Call Summary for Register EMIF_OCP_ERROR_LOG

EMIF Controller

- Error Correction And Detection Feature: [0]
- EMIF Register Summary: [1]

Table 10-120. EMIF_READ_WRITE_LEVELING_RAMP_WINDOW

Address Offset	0x0000 00D4		
Physical Address	0x4C00 00D4	**Instance**	EMIF
Description	Read/write leveling ramp window register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								RDWRLVLINC_RMP_WIN							

Bits	Field Name	Description	Type	Reset
31:13	RESERVED		R	0x0 0000
12:0	RDWRLVLINC_RMP_WIN	Incremental leveling ramp window in number of refresh periods. The value programmed is minus one the required value. Refresh period is defined by REFRESH_RATE in EMIF_SDRAM_REFRESH_CONTROL register. **NOTE: Incremental leveling is not supported on this device.**	RW	0x0000

Table 10-121. Register Call Summary for Register EMIF_READ_WRITE_LEVELING_RAMP_WINDOW

EMIF Controller

- Global Initialization: [3]
- EMIF Register Summary: [4]

Table 10-122. EMIF_READ_WRITE_LEVELING_RAMP_CONTROL

Address Offset	0x0000 00D8		
Physical Address	0x4C00 00D8	**Instance**	EMIF
Description	Read/write leveling ramp control register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RDWRLVL_EN	RDWRLVLINC_RMP_PRE							RDLVLINC_RMP_INT								RDLVLGATEINC_RMP_INT								WRLVLINC_RMP_INT							

Bits	Field Name	Description	Type	Reset
31	RDWRLVL_EN	Read-Write Leveling enable. Set 1 to enable leveling. Set 0 to disable leveling.	RW	0
30:24	RDWRLVLINC_RMP_PRE	Incremental leveling pre-scalar in number of refresh periods during ramp window. The value programmed is minus one the required value. Refresh period is defined by REFRESH_RATE in EMIF_SDRAM_REFRESH_CONTROL register. **NOTE: Incremental leveling is not supported on this device.**	RW	0x00
23:16	RDLVLINC_RMP_INT	Incremental read data eye training interval during ramp window. Number of RDWRLVLINC_RMP_PRE intervals between incremental read data eye training during ramp window. A value of 0 will disable incremental read data eye training.**NOTE: Incremental leveling is not supported on this device.**	RW	0x00
15:8	RDLVLGATEINC_RMP_INT	Incremental read DQS gate training interval during ramp window. Number of RDWRLVLINC_RMP_PRE intervals between incremental read DQS gate training during ramp window. A value of 0 will disable incremental read DQS gate training.**NOTE: Incremental leveling is not supported on this device.**	RW	0x00
7:0	WRLVLINC_RMP_INT	Incremental write leveling interval during ramp window. Number of RDWRLVLINC_RMP_PRE intervals between incremental write leveling during ramp window. A value of 0 will disable incremental write leveling.**NOTE: Incremental leveling is not supported on this device.**	RW	0x00

Table 10-123. Register Call Summary for Register EMIF_READ_WRITE_LEVELING_RAMP_CONTROL

EMIF Controller

- Global Initialization: [3][4][5][6][7]
- EMIF Register Summary: [8]

Table 10-124. EMIF_READ_WRITE_LEVELING_CONTROL

Address Offset	0x0000 00DC		
Physical Address	0x4C00 00DC	**Instance**	EMIF
Description	Read/write leveling control register		
Type	RW		

31	30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
RDWRLVLFULL_START	RDWRLVLINC_PRE	RDLVLINC_INT	RDLVLGATEINC_INT	WRLVLINC_INT

Bits	Field Name	Description	Type	Reset
31	RDWRLVLFULL_START	Full leveling trigger. Writing a 1 to this field triggers full read and write leveling. This bit will self clear to 0.	RW	0
30:24	RDWRLVLINC_PRE	Incremental leveling pre-scalar in number of refresh periods. The value programmed is minus one the required value. Refresh period is defined by REFRESH_RATE in EMIF_SDRAM_REFRESH_CONTROL register. **NOTE: Incremental leveling is not supported on this device.**	RW	0x00

Bits	Field Name	Description	Type	Reset
23:16	RDLVLINC_INT	Incremental read data eye training interval. Number of RDWRLVLINC_PRE intervals between incremental read data eye training. A value of 0 will disable incremental read data eye training. **NOTE: Incremental leveling is not supported on this device.**	RW	0x00
15:8	RDLVLGATEINC_INT	Incremental read DQS gate training interval. Number of RDWRLVLINC_PRE intervals between incremental read DQS gate training. A value of 0 will disable incremental read DQS gate training. **NOTE: Incremental leveling is not supported on this device.**	RW	0x00
7:0	WRLVLINC_INT	Incremental write leveling interval. Number of RDWRLVLINC_PRE intervals between incremental write leveling. A value of 0 will disable incremental write leveling. **NOTE: Incremental leveling is not supported on this device.**	RW	0x00

Table 10-125. Register Call Summary for Register EMIF_READ_WRITE_LEVELING_CONTROL

EMIF Controller

- Full Leveling: [0][1]
- Global Initialization: [4][5][6]
- EMIF Register Summary: [7]

Table 10-126. EMIF_DDR_PHY_CONTROL_1

Address Offset	0x0000 00E4		
Physical Address	0x4C00 00E4	**Instance**	EMIF
Description	PHY control register 1		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				RDLVL_MASK	RDLVLGATE_MASK	WRLVL_MASK	RESERVED			PHY_HALF_DELAYS	PHY_CLK_STALL_LEVEL	PHY_DIS_CALIB_RST	PHY_INVERT_CLKOUT	PHY_DLL_LOCK_DIFF								PHY_FAST_DLL_LOCK	RESERVED				READ_LATENCY						

Bits	Field Name	Description	Type	Reset
31:28	RESERVED	Reserved	RW	0x0
27	RDLVL_MASK	Writing a 1 to this field will mask read data eye training during full leveling command, plus drives reg_phy_use_rd_data_eye_level control low to allow user to use programmed ratio values.	RW	0
26	RDLVLGATE_MASK	Writing a 1 to this field will mask dqs gate training during full leveling command, plus drives reg_phy_use_rd_dqs_level control low to allow user to use programmed ratio values.	RW	0
25	WRLVL_MASK	Writing a 1 to this field will mask write leveling training during full leveling command, plus drives reg_phy_use_wr_level control low to allow user to use programmed ratio values.	RW	0
24:22	RESERVED	Reserved	RW	0x0
21	PHY_HALF_DELAYS	Adjust slave delay line delays to support 2× mode 1: 2× mode (MDLL clock is half the rate of PHY) 0: 1× mode (MDLL clock rate is same as PHY)	RW	0

Bits	Field Name	Description	Type	Reset
20	PHY_CLK_STALL_LEVEL	Enable variable idle value for delay lines. Enable during normal operations to avoid differential aging in the delay lines.	RW	0
19	PHY_DIS_CALIB_RST	Disable the dll_calib (internally generated) signal from resetting the Read Capture FIFO pointers and portions of data PHYs. Debug only. Note: dll_calib is generated by 1. EMIF_MISC_REG[0] DLL_CALIB_OS set to 1,or 2. by the PHY when it detects that the clock frequency variation has exceeded the bounds set by PHY_DLL_LOCK_DIFF or 3. periodically throughout the leveling process.	RW	0
18	PHY_INVERT_CLKOUT	Inverts the polarity of DRAM clock. 0: core clock is passed on to DRAM 1: inverted core clock is passed on to DRAM	RW	0
17:10	PHY_DLL_LOCK_DIFF	The maximum number of delay line taps variation while maintaining the master DLL lock. When the PHY is in locked state and the variation on the clock exceeds the variation indicated by this field, the lock signal is de-asserted and a dll_calib signal is generated. To prevent the dll_calib signal from being asserted in the middle of traffic when the clock jitter exceeds the variation, this register needs to be set to a value which will ensure that the lock will not be lost. Recommended value is 16.	RW	0x02
9	PHY_FAST_DLL_LOCK	Controls master DLL to lock fast or average logic must be part of locking process. Set to 1 before OPP transition commences, and set back to 0 after OPP transition completes. 1: MDLL lock is asserted based on single sample 0: MDLL lock is asserted based on average of 16 samples.	RW	0
8:5	RESERVED	Reserved	RW	0x00
4:0	READ_LATENCY	This field defines the read latency for the read data from SDRAM in number of DDR clock cycles. This field is used by the EMIF as well as the PHY. READ_LATENCY = RL + reg_phy_rdc_we_to_re -1. EMIF uses above equation to calculate reg_phy_rdc_we_to_re and forward it to the PHY. For DDR3, the true RL is used, not the decoded value. See JEDEC spec. The write latency (WL) is tied to the RL for the LPDDR2. See the LPDDR2 SDRAM standard.	RW	0x01E

Table 10-127. Register Call Summary for Register EMIF_DDR_PHY_CONTROL_1

EMIF Controller

- Software Leveling: [0][1]
- Global Initialization: [2][3][4][5][6][7][8][9][10][11][12][13][14][15][16][17][18]
- EMIF Register Summary: [19]

Table 10-128. EMIF_DDR_PHY_CONTROL_1_SHADOW

Address Offset	0x0000 00E8		
Physical Address	0x4C00 00E8	**Instance**	EMIF
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				RDLVL_MASK_SHDW	RDLVLGATE_MASK_SHDW	WRLVL_MASK_SHDW	RESERVED			PHY_HALF_DELAYS_SHDW	PHY_CLK_STALL_LEVEL_SHDW	PHY_DIS_CALIB_RST_SHDW	PHY_INVERT_CLKOUT_SHDW	PHY_DLL_LOCK_DIFF_SHDW								PHY_FAST_DLL_SHDW	RESERVED						READ_LATENCY_SHDW				

Bits	Field Name	Description	Type	Reset
31:28	RESERVED	Reserved	RW	0x0
27	RDLVL_MASK_SHDW	Shadow field for RDLVL_MASK	RW	0
26	RDLVLGATE_MASK_SHDW	Shadow field for RDLVLGATE_MASK	RW	0
25	WRLVL_MASK_SHDW	Shadow field for WRLVL_MASK	RW	0
24:22	RESERVED	Reserved	RW	0x0
21	PHY_HALF_DELAYS_SHDW	Shadow field for PHY_HALF_DELAYS	RW	0
20	PHY_CLK_STALL_LEVEL_SHDW	Shadow field for PHY_CLK_STALL_LEVEL	RW	0
19	PHY_DIS_CALIB_RST_SHDW	Shadow field for PHY_DIS_CALIB_RST	RW	0
18	PHY_INVERT_CLKOUT_SHDW	Shadow field for PHY_INVERT_CLKOUT	RW	0
17:10	PHY_DLL_LOCK_DIFF_SHDW	Shadow field for PHY_DLL_LOCK_DIFF	RW	0x00
9	PHY_FAST_DLL_SHDW	Shadow field for PHY_FAST_DLL	RW	0
8:5	RESERVED	Reserved	RW	0x00
4:0	READ_LATENCY_SHDW	Shadow field for READ_LATENCY	RW	0x000

Table 10-129. Register Call Summary for Register EMIF_DDR_PHY_CONTROL_1_SHADOW

EMIF Controller
- Global Initialization: [0][1]
- EMIF Register Summary: [2]

Table 10-130. EMIF_DDR_PHY_CONTROL_2

Address Offset	0x0000 00EC		
Physical Address	0x4C00 00EC	**Instance**	EMIF
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															

Bits	Field Name	Description	Type	Reset
31:0	RESERVED	Reserved	RW	0x0

Table 10-131. Register Call Summary for Register EMIF_DDR_PHY_CONTROL_2

EMIF Controller
- Global Initialization: [0][1][2][3]
- EMIF Register Summary: [4]

Table 10-132. EMIF_PRIORITY_TO_CLASS_OF_SERVICE_MAPPING

Address offset	0x0000 0100		
Physical Address	0x4C00 0100	**Instance**	EMIF
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PRI_COS_MAP_EN	RESERVED															PRI_7_COS		PRI_6_COS		PRI_5_COS		PRI_4_COS		PRI_3_COS		PRI_2_COS		PRI_1_COS		PRI_0_COS	

Bits	Field Name	Description	Type	Reset
31	PRI_COS_MAP_EN	Set 1 to enable priority to class of service mapping. Set 0 to disable mapping.	RW	0x0
30:16	RESERVED		R	0x0
15:14	PRI_7_COS	Class of service for commands with priority of 7. Value can be 1, 2, or 3. Setting a value of 0 will have similar effects as a value of 3.	RW	0x0
13:12	PRI_6_COS	Class of service for commands with priority of 6. Value can be 1, 2, or 3. Setting a value of 0 will have similar effects as a value of 3.	RW	0x0
11:10	PRI_5_COS	Class of service for commands with priority of 5. Value can be 1, 2, or 3. Setting a value of 0 will have similar effects as a value of 3.	RW	0x0
9:8	PRI_4_COS	Class of service for commands with priority of 4. Value can be 1, 2, or 3. Setting a value of 0 will have similar effects as a value of 3.	RW	0x0
7:6	PRI_3_COS	Class of service for commands with priority of 3. Value can be 1, 2, or 3. Setting a value of 0 will have similar effects as a value of 3.	RW	0x0
5:4	PRI_2_COS	Class of service for commands with priority of 2. Value can be 1, 2, or 3. Setting a value of 0 will have similar effects as a value of 3.	RW	0x0
3:2	PRI_1_COS	Class of service for commands with priority of 1. Value can be 1, 2, or 3. Setting a value of 0 will have similar effects as a value of 3.	RW	0x0
1:0	PRI_0_COS	Class of service for commands with priority of 0. Value can be 1, 2, or 3. Setting a value of 0 will have similar effects as a value of 3.	RW	0x0

Table 10-133. Register Call Summary for Register EMIF_PRIORITY_TO_CLASS_OF_SERVICE_MAPPING

EMIF Controller

- Class of Service: [0]
- Global Initialization: [1]
- EMIF Register Summary: [2]

Table 10-134. EMIF_CONNECTION_ID_TO_CLASS_OF_SERVICE_1_MAPPING

Address offset	0x0000 0104		
Physical Address	0x4C00 0104	**Instance**	EMIF
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CONNID_COS_1_MAP_EN	CONNID_1_COS_1							MSK_1_COS_1				CONNID_2_COS_1								MSK_2_COS_1		CONNID_3_COS_1								MSK_3_COS_1	

Bits	Field Name	Description	Type	Reset
31	CONNID_COS_1_MAP_EN	Set 1 to enable Connection ID to class of service 1 mapping. Set 0 to disable mapping.	RW	0x0
30:23	CONNID_1_COS_1	Connection ID value 1 for class of service 1.	RW	0x0
22:20	MSK_1_COS_1	Mask for Connection ID value 1 for class of service 1. Value of 0 will disable masking. Value of 1 will mask Connection ID bit 0. Value of 2 will mask Connection ID bits 1:0. Value of 3 will mask Connection ID bits 2:0. Value of 4 will mask Connection ID bits 3:0. Value of 5 will mask Connection ID bits 4:0. Value of 6 will mask Connection ID bits 5:0. Value of 7 will mask Connection ID bits 6:0.	RW	0x0
19:12	CONNID_2_COS_1	Connection ID value 2 for class of service 1.	RW	0x0
11:10	MSK_2_COS_1	Mask for Connection ID value 2 for class of service 1. Value of 0 will disable masking. Value of 1 will mask Connection ID bit 0. Value of 2 will mask Connection ID bits 1:0. Value of 3 will mask Connection ID bits 2:0.	RW	0x0
9:2	CONNID_3_COS_1	Connection ID value 3 for class of service 1.	RW	0x0
1:0	MSK_3_COS_1	Mask for Connection ID value 3 for class of service 1. Value of 0 will disable masking. Value of 1 will mask Connection ID bit 0. Value of 2 will mask Connection ID bits 1:0. Value of 3 will mask Connection ID bits 2:0.	RW	0x0

Table 10-135. Register Call Summary for Register EMIF_CONNECTION_ID_TO_CLASS_OF_SERVICE_1_MAPPING

EMIF Controller

- Class of Service: [0]
- Global Initialization: [1]
- EMIF Register Summary: [2]

Table 10-136. EMIF_CONNECTION_ID_TO_CLASS_OF_SERVICE_2_MAPPING

Address offset	0x0000 0108		
Physical Address	0x4C00 0108	**Instance**	EMIF
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CONNID_COS_2_MAP_EN	CONNID_1_COS_2							MSK_1_COS_2				CONNID_2_COS_2				MSK_2_COS_2		CONNID_3_COS_2						MSK_3_COS_2							

Bits	Field Name	Description	Type	Reset
31	CONNID_COS_2_MAP_EN	Set 1 to enable Connection ID to class of service 2 mapping. Set 0 to disable mapping.	RW	0x0
30:23	CONNID_1_COS_2	Connection ID value 1 for class of service 2.	RW	0x0
22:20	MSK_1_COS_2	Mask for Connection ID value 1 for class of service 2. Value of 0 will disable masking. Value of 1 will mask Connection ID bit 0. Value of 2 will mask Connection ID bits 1:0. Value of 3 will mask Connection ID bits 2:0. Value of 4 will mask Connection ID bits 3:0. Value of 5 will mask Connection ID bits 4:0. Value of 6 will mask Connection ID bits 5:0. Value of 7 will mask Connection ID bits 6:0.	RW	0x0
19:12	CONNID_2_COS_2	Connection ID value 2 for class of service 2.	RW	0x0
11:10	MSK_2_COS_2	Mask for Connection ID value 2 for class of service 2. Value of 0 will disable masking. Value of 1 will mask Connection ID bit 0. Value of 2 will mask Connection ID bits 1:0. Value of 3 will mask Connection ID bits 2:0.	RW	0x0
9:2	CONNID_3_COS_2	Connection ID value 3 for class of service 2.	RW	0x0
1:0	MSK_3_COS_2	Mask for Connection ID value 3 for class of service 2. Value of 0 will disable masking. Value of 1 will mask Connection ID bit 0. Value of 2 will mask Connection ID bits 1:0. Value of 3 will mask Connection ID bits 2:0.	RW	0x0

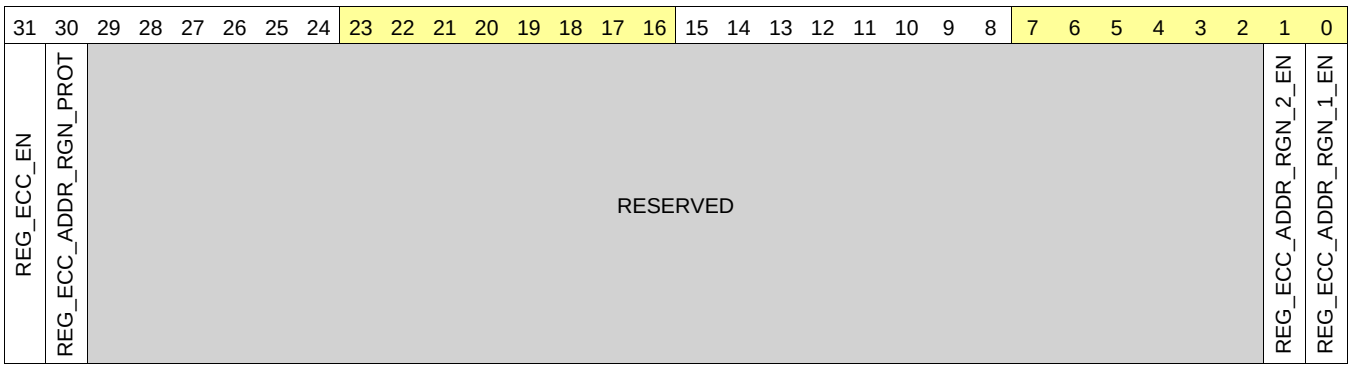
Table 10-137. Register Call Summary for Register EMIF_CONNECTION_ID_TO_CLASS_OF_SERVICE_2_MAPPING

EMIF Controller

- Class of Service: [0]
- Global Initialization: [1]
- EMIF Register Summary: [2]

Table 10-138. EMIF_ECC_CTRL_REG

Address offset	0x4C00 0110		
Physical Address	0x4C00 0110	**Instance**	EMIF
Description			
Type	RW		



Bits	Field Name	Description	Type	Reset
31	REG_ECC_EN	Set 1 to enable ECC. Set 0 to disable ECC.	RW	0x0
30	REG_ECC_ADDR_RGN_PROT	Setting this field to 1 and reg_ecc_en to a 1 will enable ECC calculation for accesses within the address ranges and disable ECC calculation for accesses outside the address ranges. Setting this field to 0 and reg_ecc_en to a 1 will disable ECC calculation for accesses within the address ranges and enable ECC calculation for accesses outside the address ranges. The address ranges can be specified using the ECC Address Range 1 and 2 registers.	RW	0x0
29:2	RESERVED		R	0x0
1	REG_ECC_ADDR_RGN_2_EN	Set 1 to enable ECC address range 2. Set 0 to disable ECC address range 2.	RW	0x0
0	REG_ECC_ADDR_RGN_1_EN	Set 1 to enable ECC address range 1. Set 0 to disable ECC address range 1.	RW	0x0

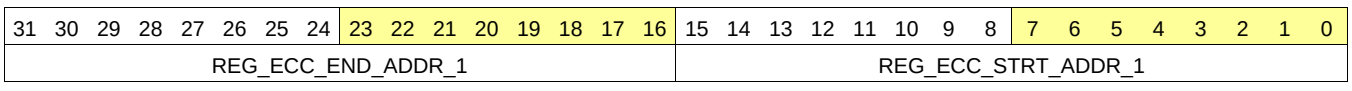
Table 10-139. Register Call Summary for Register EMIF_ECC_CTRL_REG

EMIF Controller

- Error Correction And Detection Feature: [0]
- Global Initialization: [1][2]
- Operational Modes Configuration: [3][4][5][6]
- EMIF Register Summary: [7]

Table 10-140. EMIF_ECC_ADDRESS_RANGE_1

Address offset	0x4C00 0114		
Physical Address	0x4C00 0114	**Instance**	EMIF
Description			
Type	RW		



Bits	Field Name	Description	Type	Reset
31:16	REG_ECC_END_ADDR_1	End address[32:17] for ECC address range 1. If this bit field is set to 0x1000, this indicates that the SDRAM physical end address on which the ECC applies is 0x1000 FFFF. If this bit field is set to 0x0FFF the physical end address on which the ECC applies is 0x0FFF FFFF. This bit field controls only the 16 MSbs of the physical end address of the ECC protected range. The other 16 LSbs are always 0xFFFF.	RW	0x0
15:0	REG_ECC_STRT_ADDR_1	Start address[32:17] for ECC address range 1. If this bit field is set to 0x0000, this indicates that the SDRAM physical start address on which the ECC applies is 0x0000 0000. This bit field controls only the 16 MSbs of the physical start address of the ECC protected range. The other 16 LSbs are always 0x0000.	RW	0x0

Table 10-141. Register Call Summary for Register EMIF_ECC_ADDRESS_RANGE_1

EMIF Controller

- Error Correction And Detection Feature: [0]
- Global Initialization: [1]
- Operational Modes Configuration: [2][3]
- EMIF Register Summary: [4]

Table 10-142. EMIF_ECC_ADDRESS_RANGE_2

Address offset	0x4C00 0118		
Physical Address	0x4C00 0118	**Instance**	EMIF
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REG_ECC_END_ADDR_2																REG_ECC_STRT_ADDR_2															

Bits	Field Name	Description	Type	Reset
31:16	REG_ECC_END_ADDR_2	End address[32:17] for ECC address range 2. If this bit field is set to 0x1000, this indicates that the SDRAM physical end address on which the ECC applies is 0x1000 FFFF. If this bit field is set to 0x0FFF the physical end address on which the ECC applies is 0x0FFF FFFF. This bit field controls only the 16 MSbs of the physical end address of the ECC protected range. The other 16 LSbs are always 0xFFFF.	RW	0x0
15:0	REG_ECC_STRT_ADDR_2	Start address[32:17] for ECC address range 2. If this bit field is set to 0x0000, this indicates that the SDRAM physical start address on which the ECC applies is 0x0000 0000. This bit field controls only the 16 MSbs of the physical start address of the ECC protected range. The other 16 LSbs are always 0x0000.	RW	0x0

Table 10-143. Register Call Summary for Register EMIF_ECC_ADDRESS_RANGE_2

EMIF Controller

- Error Correction And Detection Feature: [0]
- Global Initialization: [1]
- Operational Modes Configuration: [2][3]
- EMIF Register Summary: [4]

Table 10-144. EMIF_READ_WRITE_EXECUTION_THRESHOLD

Address Offset	0x0000 0120		
Physical Address	0x4C00 0120	**Instance**	EMIF
Description			

Table 10-144. EMIF_READ_WRITE_EXECUTION_THRESHOLD (continued)

Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MFLAG_OVERRIDE	RESERVED																		WR_THRSH					RESERVED					RD_THRSH		

Bits	Field Name	Description	Type	Reset
31	MFLAG_OVERRIDE	Mflag override. 0x0: Use MFLAG 0x1: Use Priority Class of Service	RW	0
30:13	RESERVED	Reserved	R	0x0000
12:8	WR_THRSH	Write Threshold. Number of SDRAM write bursts after which the EMIF arbitration will switch to executing read commands. The value programmed is always minus one the required number	RW	0x03
7:5	RESERVED	Reserved	R	0x0
4:0	RD_THRSH	Read threshold. Number of SDRAM read bursts after which the EMIF arbitration will switch to executing write commands. The value that is programmed is always minus one the required number	R	0x05

Table 10-145. Register Call Summary for Register EMIF_READ_WRITE_EXECUTION_THRESHOLD

EMIF Controller
- Global Initialization: [0][1][2]
- EMIF Register Summary: [3]

Table 10-146. EMIF_COS_CONFIG

Address Offset	0x0000 0124		
Physical Address	0x4C00 0124	**Instance**	EMIF
Description	Priority Raise Counter Register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								COS_COUNT_1								COS_COUNT_2								PR_OLD_COUNT							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x00
23:16	COS_COUNT_1	Priority Raise Counter for class of service 1. Number of EMIF_FICLK cycles after which the EMIF momentarily raises the priority of the class of service 1 commands in the Command FIFO. A value of N will be equal to N x 16 clocks.	RW	0xFF
15:8	COS_COUNT_2	Priority Raise Counter for class of service 2. Number of EMIF_FICLK cycles after which the EMIF momentarily raises the priority of the class of service 2 commands in the Command FIFO. A value of N will be equal to N x 16 clocks.	RW	0xFF

Bits	Field Name	Description	Type	Reset
7:0	PR_OLD_COUNT	Priority Raise Old Counter. Number of EMIF_FICLK cycles after which the EMIF momentarily raises the priority of the oldest command in the Command FIFO. A value of N will be equal to N x 16 clocks.	RW	0xFF

Table 10-147. Register Call Summary for Register EMIF_COS_CONFIG

EMIF Controller

- Arbitration of Commands in the Command FIFO: [0]
- Class of Service: [1][2][3][4][5][6][7][8]
- Global Initialization: [9]
- EMIF Register Summary: [10]

Table 10-148. EMIF_1B_ECC_ERR_CNT

Address offset	0x130		
Physical Address	0x4C00 0130	**Instance**	EMIF
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REG_1B_ECC_ERR_CNT																															

Bits	Field Name	Description	Type	Reset
31:0	REG_1B_ECC_ERR_CNT	32 bit counter that displays number of 1-bit ECC errors. Writing a value will decrement the count by that value. For example, if the count is 0x1234_ABF3, writing 0x1234_ABF3 to this register will clear it.	RW	0x0

Table 10-149. Register Call Summary for Register EMIF_1B_ECC_ERR_CNT

EMIF Controller

- Error Correction And Detection Feature: [0]
- Operational Modes Configuration: [1]
- EMIF Register Summary: [2]

Table 10-150. EMIF_1B_ECC_ERR_THRSH

Address offset	0x134		
Physical Address	0x4C00 0134	**Instance**	EMIF
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REG_1B_ECC_ERR_THRSH								RESERVED								REG_1B_ECC_ERR_WIN															

Bits	Field Name	Description	Type	Reset
31:24	REG_1B_ECC_ERR_THRSH	1-bit ECC error threshold. The EMIF will generate an interrupt when the 1-bit ECC error count is greater than or equal to this threshold. A value of 0 will disable the generation of the interrupt.	RW	0x0
23:16	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
15:0	REG_1B_ECC_ERR_WIN	1-bit ECC error window in number of refresh periods. The EMIF will generate an interrupt when the 1-bit ECC error count is equal to or greater than the threshold within this window. A value of 0 will disable the window. Refresh period is defined by EMIF_SDRAM_REFRESH_CONTROL[15:0] REFRESH_RATE. The software can set this bitfield to 0x0 to reset the internal counter.	RW	0x0

Table 10-151. Register Call Summary for Register EMIF_1B_ECC_ERR_THRSH

EMIF Controller

- Error Correction And Detection Feature: [0]
- Operational Modes Configuration: [1]
- EMIF Register Summary: [2]

Table 10-152. EMIF_1B_ECC_ERR_DIST_1

Address offset	0x138		
Physical Address	0x4C00 0138	**Instance**	EMIF
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REG_1B_ECC_ERR_DIST_1																															

Bits	Field Name	Description	Type	Reset
31:0	REG_1B_ECC_ERR_DIST_1	1-bit ECC error distribution over data bus bit 31:0. A value of 1 on a bit indicates 1-bit error on the corresponding bit on the data bus. Writing a 1 to any bit will clear that bit. Writing a 0 has no effect.	RW	0x0

Table 10-153. Register Call Summary for Register EMIF_1B_ECC_ERR_DIST_1

EMIF Controller

- Error Correction And Detection Feature: [0]
- Operational Modes Configuration: [1]
- EMIF Register Summary: [2]

Table 10-154. EMIF_1B_ECC_ERR_ADDR_LOG

Address offset	0x13C		
Physical Address	0x4C00 013C	**Instance**	EMIF
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REG_1B_ECC_ERR_ADDR																															

Bits	Field Name	Description	Type	Reset
31:0	REG_1B_ECC_ERR_ADDR	1-bit ECC error address. Most significant bits of the starting address(es) related to the SDRAM reads that had a 1-bit ECC error. This field displays up to four addresses logged in the 4 deep address logging FIFO. Writing a 0x1 will pop one element of the FIFO. Writing a 0x2 will pop all elements of the FIFO. Writing any other alue will have no effect.	RW	0x0

Table 10-155. Register Call Summary for Register EMIF_1B_ECC_ERR_ADDR_LOG

EMIF Controller

- Error Correction And Detection Feature: [0][1]
- EMIF Register Summary: [2]

Table 10-156. EMIF_2B_ECC_ERR_ADDR_LOG

Address offset	0x140		
Physical Address	0x4C00 0140	**Instance**	EMIF
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REG_2B_ECC_ERR_ADDR																															

Bits	Field Name	Description	Type	Reset
31:0	REG_2B_ECC_ERR_ADDR	2-bit ECC error address. Most significant bits of the starting address of the first SDRAM burst that had the 2-bit ECC error. Writing a 1 will clear this field. Writing any other value has no effect.	RW	0x0

Table 10-157. Register Call Summary for Register EMIF_2B_ECC_ERR_ADDR_LOG

EMIF Controller

- Error Correction And Detection Feature: [0]
- Operational Modes Configuration: [1]
- EMIF Register Summary: [2]

Table 10-158. EMIF_PHY_STATUS_1

Address Offset	0x0000 0144		
Physical Address	0x4C00 0144	**Instance**	EMIF
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				PHY_REG_PHY_CTRL_DLL_SLAVE_VALUE																RESERVED				PHY_REG_STATUS_DLL_LOCK				RESERVED		PHY_REG_PHY_CTRL_DLL_LOCK	

Bits	Field Name	Description	Type	Reset
31:30	RESERVED		R	0x0
29:12	PHY_REG_PHY_CTRL_DLL_SLAVE_VALUE	DLL Slave Value	R	0x0
11:9	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
8:4	PHY_REG_STATUS_DLL_LOCK	Lock Status for Data DLLs	R	0x0
3:2	RESERVED		R	0x0
1:0	PHY_REG_PHY_CTRL_DLL_LOCK	Lock Status for Command DLLs	R	0x0

Table 10-159. Register Call Summary for Register EMIF_PHY_STATUS_1

EMIF Controller
- EMIF Register Summary: [0]

Table 10-160. EMIF_PHY_STATUS_2

Address Offset	0x0000 0148		
Physical Address	0x4C00 0148	**Instance**	EMIF
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PHY_REG_STATUS_DLL_SLAVE_VALUE_LO																															

Bits	Field Name	Description	Type	Reset
31:0	PHY_REG_STATUS_DLL_SLAVE_VALUE_LO	Bits 31:0 of Phy_reg_status_dll_slave_value	R	0x0

Table 10-161. Register Call Summary for Register EMIF_PHY_STATUS_2

EMIF Controller
- EMIF Register Summary: [0]

Table 10-162. EMIF_PHY_STATUS_3

Address Offset	0x0000 014C		
Physical Address	0x4C00 014C	**Instance**	EMIF
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	PHY_REG_RDFIFO_RDPTR															RESERVED	PHY_REG_STATUS_DLL_SLAVE_VALUE_HI														

Bits	Field Name	Description	Type	Reset
31	RESERVED		R	0x0
30:16	PHY_REG_RDFIFO_RDPTR	Read FIFO Read Pointer	R	0x0
15:13	RESERVED		R	0x0
12:0	PHY_REG_STATUS_DLL_SLAVE_VALUE_HI	Bits 44:32 of Phy_reg_status_dll_slave_value	R	0x0

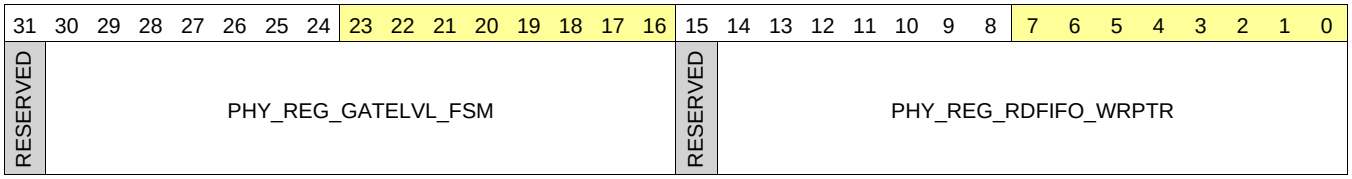
Table 10-163. Register Call Summary for Register EMIF_PHY_STATUS_3

EMIF Controller

- EMIF Register Summary: [0]

Table 10-164. EMIF_PHY_STATUS_4

Address Offset	0x0000 0150		
Physical Address	0x4C00 0150	**Instance**	EMIF
Description			
Type	R		



Bits	Field Name	Description	Type	Reset
31	RESERVED		R	0x0
30:16	PHY_REG_GATELVL_FSM	Gate Levelling FSM	R	0x0
15	RESERVED		R	0x0
14:0	PHY_REG_RDFIFO_WRPTR	Read FIFO Write Pointer	R	0x0

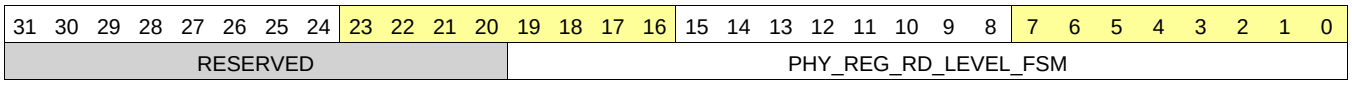
Table 10-165. Register Call Summary for Register EMIF_PHY_STATUS_4

EMIF Controller

- EMIF Register Summary: [0]

Table 10-166. EMIF_PHY_STATUS_5

Address Offset	0x0000 0154		
Physical Address	0x4C00 0154	**Instance**	EMIF
Description			
Type	R		



Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19:0	PHY_REG_RD_LEVEL_FSM	Read Levelling FSM	R	0x0

Table 10-167. Register Call Summary for Register EMIF_PHY_STATUS_5

EMIF Controller

- EMIF Register Summary: [0]

Table 10-168. EMIF_PHY_STATUS_6

Address Offset	0x0000 0158		
Physical Address	0x4C00 0158	**Instance**	EMIF
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																PHY_REG_WR_LEVEL_FSM															

Bits	Field Name	Description	Type	Reset
31:15	RESERVED		R	0x0
14:0	PHY_REG_WR_LEVEL_FSM	Writel Levelling FSM	R	0x0

Table 10-169. Register Call Summary for Register EMIF_PHY_STATUS_6

EMIF Controller

- EMIF Register Summary: [0]

Table 10-170. EMIF_PHY_STATUS_7

Address Offset	0x0000 015C		
Physical Address	0x4C00 015C	**Instance**	EMIF
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								PHY_REG_RDLVL_DQS_RATIO1								RESERVED								PHY_REG_RDLVL_DQS_RATIO0							

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:16	PHY_REG_RDLVL_DQS_RATIO1	Read levelling DQS ratio1	R	0x0
15:10	RESERVED		R	0x0
9:0	PHY_REG_RDLVL_DQS_RATIO0	Read levelling DQS ratio0	R	0x0

Table 10-171. Register Call Summary for Register EMIF_PHY_STATUS_7

EMIF Controller

- Global Initialization: [0][1]
- EMIF Register Summary: [2]

Table 10-172. EMIF_PHY_STATUS_8

Address Offset	0x0000 0160		
Physical Address	0x4C00 0160	**Instance**	EMIF
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								PHY_REG_RDLVL_DQS_RATIO3								RESERVED								PHY_REG_RDLVL_DQS_RATIO2							

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:16	PHY_REG_RDLVL_DQS_RATIO3	Read levelling DQS ratio3	R	0x0
15:10	RESERVED		R	0x0
9:0	PHY_REG_RDLVL_DQS_RATIO2	Read levelling DQS ratio2	R	0x0

Table 10-173. Register Call Summary for Register EMIF_PHY_STATUS_8

EMIF Controller

- EMIF Register Summary: [0]

Table 10-174. EMIF_PHY_STATUS_9

Address Offset	0x0000 0164		
Physical Address	0x4C00 0164	**Instance**	EMIF
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								PHY_REG_RDLVL_DQS_RATIO5								RESERVED								PHY_REG_RDLVL_DQS_RATIO4							

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:16	PHY_REG_RDLVL_DQS_RATIO5	Read Levelling DQS ratio5	R	0x0
15:10	RESERVED		R	0x0
9:0	PHY_REG_RDLVL_DQS_RATIO4	Read Levelling DQS ratio4	R	0x0

Table 10-175. Register Call Summary for Register EMIF_PHY_STATUS_9

EMIF Controller

- EMIF Register Summary: [0]

Table 10-176. EMIF_PHY_STATUS_10

Address Offset	0x0000 0168		
Physical Address	0x4C00 0168	**Instance**	EMIF
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								PHY_REG_RDLVL_DQS_RATIO7								RESERVED								PHY_REG_RDLVL_DQS_RATIO6							

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:16	PHY_REG_RDLVL_DQS_RATIO7	Read levelling DQS ratio7	R	0x0
15:10	RESERVED		R	0x0
9:0	PHY_REG_RDLVL_DQS_RATIO6	Read levelling DQS ratio6	R	0x0

Table 10-177. Register Call Summary for Register EMIF_PHY_STATUS_10

EMIF Controller

- EMIF Register Summary: [0]

Table 10-178. EMIF_PHY_STATUS_11

Address Offset	0x0000 016C		
Physical Address	0x4C00 016C	**Instance**	EMIF
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								PHY_REG_RDLVL_DQS_RATIO9								RESERVED								PHY_REG_RDLVL_DQS_RATIO8							

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:16	PHY_REG_RDLVL_DQS_RATIO9	Read levelling DQS ratio9	R	0x0
15:10	RESERVED		R	0x0
9:0	PHY_REG_RDLVL_DQS_RATIO8	Read levelling DQS ratio8	R	0x0

Table 10-179. Register Call Summary for Register EMIF_PHY_STATUS_11

EMIF Controller

- Global Initialization: [0]
- EMIF Register Summary: [1]

Table 10-180. EMIF_PHY_STATUS_12

Address Offset	0x0000 0170		
Physical Address	0x4C00 0170	**Instance**	EMIF
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								PHY_REG_RDLVL_FIFOWEIN_RATIO1								RESERVED								PHY_REG_RDLVL_FIFOWEIN_RATIO0							

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:16	PHY_REG_RDLVL_FIFOWEIN_RATIO1	Read levelling FIFO Write Enable Ratio1	R	0x0
15:11	RESERVED		R	0x0
10:0	PHY_REG_RDLVL_FIFOWEIN_RATIO0	Read levelling FIFO Write Enable Ratio0	R	0x0

Table 10-181. Register Call Summary for Register EMIF_PHY_STATUS_12

EMIF Controller

- Global Initialization: [0]
- EMIF Register Summary: [1]

Table 10-182. EMIF_PHY_STATUS_13

Address Offset	0x0000 0174		
Physical Address	0x4C00 0174	**Instance**	EMIF
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED					PHY_REG_RDLVL_FIFOWEIN_RATIO3											RESERVED					PHY_REG_RDLVL_FIFOWEIN_RATIO2												

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:16	PHY_REG_RDLVL_FIFOWEIN_RATIO3	Read levelling FIFO Write Enable Ratio3	R	0x0
15:11	RESERVED		R	0x0
10:0	PHY_REG_RDLVL_FIFOWEIN_RATIO2	Read levelling FIFO Write Enable Ratio2	R	0x0

Table 10-183. Register Call Summary for Register EMIF_PHY_STATUS_13

EMIF Controller

- EMIF Register Summary: [0]

Table 10-184. EMIF_PHY_STATUS_14

Address Offset	0x0000 0178		
Physical Address	0x4C00 0178	**Instance**	EMIF
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED					PHY_REG_RDLVL_FIFOWEIN_RATIO5											RESERVED					PHY_REG_RDLVL_FIFOWEIN_RATIO4												

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:16	PHY_REG_RDLVL_FIFOWEIN_RATIO5	Read levelling FIFO Write Enable Ratio5	R	0x0
15:11	RESERVED		R	0x0
10:0	PHY_REG_RDLVL_FIFOWEIN_RATIO4	Read levelling FIFO Write Enable Ratio4	R	0x0

Table 10-185. Register Call Summary for Register EMIF_PHY_STATUS_14

EMIF Controller

- EMIF Register Summary: [0]

Table 10-186. EMIF_PHY_STATUS_15

Address Offset	0x0000 017C		
Physical Address	0x4C00 017C	**Instance**	EMIF
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								PHY_REG_RDLVL_FIFOWEIN_RATIO7								RESERVED								PHY_REG_RDLVL_FIFOWEIN_RATIO6							

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:16	PHY_REG_RDLVL_FIFOWEIN_RATIO7	Read levelling FIFO Wrie Enable Ratio7	R	0x0
15:11	RESERVED		R	0x0
10:0	PHY_REG_RDLVL_FIFOWEIN_RATIO6	Read levelling FIFO Wrie Enable Ratio6	R	0x0

Table 10-187. Register Call Summary for Register EMIF_PHY_STATUS_15

EMIF Controller

- EMIF Register Summary: [0]

Table 10-188. EMIF_PHY_STATUS_16

Address Offset	0x0000 0180		
Physical Address	0x4C00 0180	**Instance**	EMIF
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								PHY_REG_RDLVL_FIFOWEIN_RATIO9								RESERVED								PHY_REG_RDLVL_FIFOWEIN_RATIO8							

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:16	PHY_REG_RDLVL_FIFOWEIN_RATIO9	Read levelling FIFO Write Enable Ratio9	R	0x0
15:11	RESERVED		R	0x0
10:0	PHY_REG_RDLVL_FIFOWEIN_RATIO8	Read levelling FIFO Write Enable Ratio8	R	0x0

Table 10-189. Register Call Summary for Register EMIF_PHY_STATUS_16

EMIF Controller

- Global Initialization: [0]
- EMIF Register Summary: [1]

Table 10-190. EMIF_PHY_STATUS_17

Address Offset	0x0000 0184		
Physical Address	0x4C00 0184	**Instance**	EMIF
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								PHY_REG_WRLVL_DQ_RATIO1								RESERVED								PHY_REG_WRLVL_DQ_RATIO0							

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:16	PHY_REG_WRLVL_DQ_RATIO1	Write levelling DQ ratio1	R	0x0
15:10	RESERVED		R	0x0
9:0	PHY_REG_WRLVL_DQ_RATIO0	Write levelling DQ ratio0	R	0x0

Table 10-191. Register Call Summary for Register EMIF_PHY_STATUS_17

EMIF Controller
- Global Initialization: [0]
- EMIF Register Summary: [1]

Table 10-192. EMIF_PHY_STATUS_18

Address Offset	0x0000 0188		
Physical Address	0x4C00 0188	**Instance**	EMIF
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								PHY_REG_WRLVL_DQ_RATIO3								RESERVED								PHY_REG_WRLVL_DQ_RATIO2							

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:16	PHY_REG_WRLVL_DQ_RATIO3	Write levelling DQ ratio3	R	0x0
15:10	RESERVED		R	0x0
9:0	PHY_REG_WRLVL_DQ_RATIO2	Write levelling DQ ratio2	R	0x0

Table 10-193. Register Call Summary for Register EMIF_PHY_STATUS_18

EMIF Controller
- EMIF Register Summary: [0]

Table 10-194. EMIF_PHY_STATUS_19

Address Offset	0x0000 018C		
Physical Address	0x4C00 018C	**Instance**	EMIF
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								PHY_REG_WRLVL_DQ_RATIO5								RESERVED								PHY_REG_WRLVL_DQ_RATIO4							

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:16	PHY_REG_WRLVL_DQ_RATIO5	Write levelling DQ ratio5	R	0x0
15:10	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
9:0	PHY_REG_WRLVL_DQ_RATIO4	Write levelling DQ ratio4	R	0x0

Table 10-195. Register Call Summary for Register EMIF_PHY_STATUS_19

EMIF Controller
- EMIF Register Summary: [0]

Table 10-196. EMIF_PHY_STATUS_20

Address Offset	0x0000 0190		
Physical Address	0x4C00 0190	**Instance**	EMIF
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								PHY_REG_WRLVL_DQ_RATIO7								RESERVED								PHY_REG_WRLVL_DQ_RATIO6							

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:16	PHY_REG_WRLVL_DQ_RATIO7	Write levelling DQ ratio7	R	0x0
15:10	RESERVED		R	0x0
9:0	PHY_REG_WRLVL_DQ_RATIO6	Write levelling DQ ratio6	R	0x0

Table 10-197. Register Call Summary for Register EMIF_PHY_STATUS_20

EMIF Controller
- EMIF Register Summary: [0]

Table 10-198. EMIF_PHY_STATUS_21

Address Offset	0x0000 0194		
Physical Address	0x4C00 0194	**Instance**	EMIF
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								PHY_REG_WRLVL_DQ_RATIO9								RESERVED								PHY_REG_WRLVL_DQ_RATIO8							

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:16	PHY_REG_WRLVL_DQ_RATIO9	Write levelling DQ ratio9	R	0x0
15:10	RESERVED		R	0x0
9:0	PHY_REG_WRLVL_DQ_RATIO8	Write levelling DQ ratio8	R	0x0

Table 10-199. Register Call Summary for Register EMIF_PHY_STATUS_21

EMIF Controller

- EMIF Register Summary: [0]

Table 10-200. EMIF_PHY_STATUS_22

Address Offset	0x0000 0198		
Physical Address	0x4C00 0198	**Instance**	EMIF
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								PHY_REG_WRLVL_DQS_RATIO1								RESERVED								PHY_REG_WRLVL_DQS_RATIO0							

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:16	PHY_REG_WRLVL_DQS_RATIO1	Write levelling DQS ratio 1	R	0x0
15:10	RESERVED		R	0x0
9:0	PHY_REG_WRLVL_DQS_RATIO0	Write levelling DQS ratio 0	R	0x0

Table 10-201. Register Call Summary for Register EMIF_PHY_STATUS_22

EMIF Controller

- EMIF Register Summary: [0]

Table 10-202. EMIF_PHY_STATUS_23

Address Offset	0x0000 019C		
Physical Address	0x4C00 019C	**Instance**	EMIF
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								PHY_REG_WRLVL_DQS_RATIO3								RESERVED								PHY_REG_WRLVL_DQS_RATIO2							

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:16	PHY_REG_WRLVL_DQS_RATIO3	Write levelling DQS ratio3	R	0x0
15:10	RESERVED		R	0x0
9:0	PHY_REG_WRLVL_DQS_RATIO2	Write levelling DQS ratio2	R	0x0

Table 10-203. Register Call Summary for Register EMIF_PHY_STATUS_23

EMIF Controller

- EMIF Register Summary: [0]

Table 10-204. EMIF_PHY_STATUS_24

Address Offset	0x0000 01A0		
Physical Address	0x4C00 01A0	**Instance**	EMIF
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								PHY_REG_WRLVL_DQS_RATIO5								RESERVED								PHY_REG_WRLVL_DQS_RATIO4							

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:16	PHY_REG_WRLVL_DQS_RATIO5	Write levelling DQS ratio5	R	0x0
15:10	RESERVED		R	0x0
9:0	PHY_REG_WRLVL_DQS_RATIO4	Write levelling DQS ratio4	R	0x0

Table 10-205. Register Call Summary for Register EMIF_PHY_STATUS_24

EMIF Controller

- EMIF Register Summary: [0]

Table 10-206. EMIF_PHY_STATUS_25

Address Offset	0x0000 01A4		
Physical Address	0x4C00 01A4	**Instance**	EMIF
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								PHY_REG_WRLVL_DQS_RATIO7								RESERVED								PHY_REG_WRLVL_DQS_RATIO6							

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:16	PHY_REG_WRLVL_DQS_RATIO7	Write levelling DQS ratio7	R	0x0
15:10	RESERVED		R	0x0
9:0	PHY_REG_WRLVL_DQS_RATIO6	Write levelling DQS ratio6	R	0x0

Table 10-207. Register Call Summary for Register EMIF_PHY_STATUS_25

EMIF Controller

- EMIF Register Summary: [0]

Table 10-208. EMIF_PHY_STATUS_26

Address Offset	0x0000 01A8		
Physical Address	0x4C00 01A8	**Instance**	EMIF
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								PHY_REG_WRLVL_DQS_RATIO9								RESERVED								PHY_REG_WRLVL_DQS_RATIO8							

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:16	PHY_REG_WRLVL_DQS_RATIO9	Write levelling DQS ratio9	R	0x0
15:10	RESERVED		R	0x0
9:0	PHY_REG_WRLVL_DQS_RATIO8	Write levelling DQS ratio8	R	0x0

Table 10-209. Register Call Summary for Register EMIF_PHY_STATUS_26

EMIF Controller
- Global Initialization: [0][1]
- EMIF Register Summary: [2]

Table 10-210. EMIF_PHY_STATUS_27

Address Offset	0x0000 01AC		
Physical Address	0x4C00 01AC	**Instance**	EMIF
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED		PHY_REG_PHY_CONTROL_MDLL_UNLOCK_STICKY		RESERVED				PHY_REG_STATUS_MDLL_UNLOCK_STICKY								PHY_REG_RDC_FIFO_RST_ERR_CNT															

Bits	Field Name	Description	Type	Reset
31:30	RESERVED		R	0x0
29:28	PHY_REG_PHY_CONTROL_MDLL_UNLOCK_STICKY	Phy control MDLL unlock sticky	R	0x0
27:25	RESERVED		R	0x0
24:20	PHY_REG_STATUS_MDLL_UNLOCK_STICKY	Phy data MDLL unlock sticky	R	0x0
19:0	PHY_REG_RDC_FIFO_RST_ERR_CNT	RDC FIFO reset error count	R	0x0

Table 10-211. Register Call Summary for Register EMIF_PHY_STATUS_27

EMIF Controller

- EMIF Register Summary: [0]

Table 10-212. EMIF_PHY_STATUS_28

Address Offset	0x0000 01B0		
Physical Address	0x4C00 01B0	**Instance**	EMIF
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED			PHY_REG_GATELVL_INC_FAIL					RESERVED			PHY_REG_WRLVL_INC_FAIL					RESERVED			PHY_REG_RDLVL_INC_FAIL					RESERVED			PHY_REG_FIFO_WE_IN_MIASALIGNED_STICKY				

Bits	Field Name	Description	Type	Reset
31:29	RESERVED		R	0x0
28:24	PHY_REG_GATELVL_INC_FAIL	Gate levelling failure. **NOTE: Incremental leveling is not supported on this device.**	R	0x0
23:21	RESERVED		R	0x0
20:16	PHY_REG_WRLVL_INC_FAIL	Write levelling failure. **NOTE: Incremental leveling is not supported on this device.**	R	0x0
15:13	RESERVED		R	0x0
12:8	PHY_REG_RDLVL_INC_FAIL	Read levelling failure. **NOTE: Incremental leveling is not supported on this device.**	R	0x0
7:5	RESERVED		R	0x0
4:0	PHY_REG_FIFO_WE_IN_MIASALIGNED_STICKY	FIFO write enable in misaligned sticky	R	0x0

Table 10-213. Register Call Summary for Register EMIF_PHY_STATUS_28

EMIF Controller

- EMIF Register Summary: [0]

Table 10-214. EMIF_EXT_PHY_CONTROL_1

Address Offset	0x0000 0200		
Physical Address	0x4C00 0200	**Instance**	EMIF
Description	Control DLL Slave Ratio Register		

Table 10-214. EMIF_EXT_PHY_CONTROL_1 (continued)

Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED		PHY_REG_CTRL_SLAVE_RATIO2														PHY_REG_CTRL_SLAVE_RATIO1								PHY_REG_CTRL_SLAVE_RATIO0							

Bits	Field Name	Description	Type	Reset
31:30	RESERVED		R	0x0
29:20	PHY_REG_CTRL_SLAVE_RATIO2	The user programmable ratio value for address/command launch timing in PHY control macro 2. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. **Required for LPDDR2 only. Please check the device data manual for supported DDR memory types.**	RW	0x40
19:10	PHY_REG_CTRL_SLAVE_RATIO1	The user programmable ratio value for address/command launch timing in PHY control macro 1. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. Set to: • 0x080 for PHY_INVERT_CLKOUT = 0 • 0x100 for PHY_INVERT_CLKOUT = 1 **Required for DDR2/DDR3 only. Please check the device data manual for supported DDR memory types.**	RW	0x80
9:0	PHY_REG_CTRL_SLAVE_RATIO0	The user programmable ratio value for address/command launch timing in PHY control macro 0. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. Set to: • 0x080 for PHY_INVERT_CLKOUT = 0 • 0x100 for PHY_INVERT_CLKOUT = 1 **Required for DDR2/DDR3 only. Please check the device data manual for supported DDR memory types.**	RW	0x80

Table 10-215. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_1

EMIF Controller

- Global Initialization: [0][1][2][3][4][5]
- EMIF Register Summary: [6]
- EMIF Register Description: [7]

Table 10-216. EMIF_EXT_PHY_CONTROL_1_SHADOW

Address Offset	0x0000 0204		
Physical Address	0x4C00 0204	**Instance**	EMIF
Description	Control DLL Slave Ratio Shadow Register. Its value is loaded in register EMIF_EXT_PHY_CONTROL_1 on any frequency change.		

Table 10-216. EMIF_EXT_PHY_CONTROL_1_SHADOW (continued)

Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								PHY_REG_CTRL_SLAVE_RATIO2								PHY_REG_CTRL_SLAVE_RATIO1								PHY_REG_CTRL_SLAVE_RATIO0							

Bits	Field Name	Description	Type	Reset
31:30	RESERVED		R	0x0
29:20	PHY_REG_CTRL_SLAVE_RATIO2	The user programmable ratio value for address/command launch timing in PHY control macro 2. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. **Required for LPDDR2 only. Please check the device data manual for supported DDR memory types.**	RW	0x40
19:10	PHY_REG_CTRL_SLAVE_RATIO1	The user programmable ratio value for address/command launch timing in PHY control macro 1. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. Set to: • 0x080 for PHY_INVERT_CLKOUT = 0 • 0x100 for PHY_INVERT_CLKOUT = 1 **Required for DDR2/DDR3 only. Please check the device data manual for supported DDR memory types.**	RW	0x80
9:0	PHY_REG_CTRL_SLAVE_RATIO0	The user programmable ratio value for address/command launch timing in PHY control macro 0. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. Set to: • 0x080 for PHY_INVERT_CLKOUT = 0 • 0x100 for PHY_INVERT_CLKOUT = 1 **Required for DDR2/DDR3 only. Please check the device data manual for supported DDR memory types.**	RW	0x80

Table 10-217. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_1_SHADOW

EMIF Controller
- Global Initialization: [0]
- EMIF Register Summary: [1]

Table 10-218. EMIF_EXT_PHY_CONTROL_2

Address Offset	0x0000 0208		
Physical Address	0x4C00 0208	**Instance**	EMIF
Description	Data macro 0, FIFO write enable (read DQS gate) DLL Slave Ratio Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								PHY_REG_FIFO_WE_SLAVE_RATIO1								RESERVED								PHY_REG_FIFO_WE_SLAVE_RATIO0							

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:16	PHY_REG_FIFO_WE_SLAVE_RATIO1	The user programmable ratio value for the FIFO write enable slave DLL in PHY data macro 0, chip select 1. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. **NOTE: Chip select 1 is not supported on this device.**	RW	0x0
15:11	RESERVED		R	0x0
10:0	PHY_REG_FIFO_WE_SLAVE_RATIO0	The user programmable ratio value for the FIFO write enable slave DLL in PHY data macro 0, chip select 0. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line.	RW	0x0

Table 10-219. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_2

EMIF Controller

- Full Leveling: [0]
- Software Leveling: [1][2][3]
- Global Initialization: [4][5][6][7]
- EMIF Register Summary: [8]
- EMIF Register Description: [9]

Table 10-220. EMIF_EXT_PHY_CONTROL_2_SHADOW

Address Offset	0x0000 020C		
Physical Address	0x4C00 020C	**Instance**	EMIF
Description	Data macro 0, FIFO write enable (read DQS gate) DLL Slave Ratio Shadow Register. Its value is loaded in register EMIF_EXT_PHY_CONTROL_2 on any frequency change.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								PHY_REG_FIFO_WE_SLAVE_RATIO1								RESERVED								PHY_REG_FIFO_WE_SLAVE_RATIO0							

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:16	PHY_REG_FIFO_WE_SLAVE_RATIO1	The user programmable ratio value for the FIFO write enable slave DLL in PHY data macro 0, chip select 1. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. **NOTE: Chip select 1 is not supported on this device.**	RW	0x0
15:11	RESERVED		R	0x0
10:0	PHY_REG_FIFO_WE_SLAVE_RATIO0	The user programmable ratio value for the FIFO write enable slave DLL in PHY data macro 0, chip select 0. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line.	RW	0x0

Table 10-221. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_2_SHADOW

EMIF Controller

- Global Initialization: [0]
- EMIF Register Summary: [1]

Table 10-222. EMIF_EXT_PHY_CONTROL_3

Address Offset	0x0000 0210		
Physical Address	0x4C00 0210	**Instance**	EMIF
Description	Data macro 1, FIFO write enable (read DQS gate) DLL Slave Ratio Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								PHY_REG_FIFO_WE_SLAVE_RATIO3								RESERVED								PHY_REG_FIFO_WE_SLAVE_RATIO2							

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:16	PHY_REG_FIFO_WE_SLAVE_RATIO3	The user programmable ratio value for the FIFO write enable slave DLL in PHY data macro 1, chip select 1. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. **NOTE: Chip select 1 is not supported on this device.**	RW	0x0
15:11	RESERVED		R	0x0
10:0	PHY_REG_FIFO_WE_SLAVE_RATIO2	The user programmable ratio value for the FIFO write enable slave DLL in PHY data macro 1, chip select 0. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line.	RW	0x0

Table 10-223. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_3

EMIF Controller

- EMIF Register Summary: [0]
- EMIF Register Description: [1]

Table 10-224. EMIF_EXT_PHY_CONTROL_3_SHADOW

Address Offset	0x0000 0214		
Physical Address	0x4C00 0214	**Instance**	EMIF
Description	Data macro 1, FIFO write enable (read DQS gate) DLL Slave Ratio Shadow Register. Its value is loaded in register EMIF_EXT_PHY_CONTROL_3 on any frequency change.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								PHY_REG_FIFO_WE_SLAVE_RATIO3								RESERVED								PHY_REG_FIFO_WE_SLAVE_RATIO2							

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:16	PHY_REG_FIFO_WE_SLAVE_RATIO3	The user programmable ratio value for the FIFO write enable slave DLL in PHY data macro 1, chip select 1. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. **NOTE: Chip select 1 is not supported on this device.**	RW	0x0
15:11	RESERVED		R	0x0
10:0	PHY_REG_FIFO_WE_SLAVE_RATIO2	The user programmable ratio value for the FIFO write enable slave DLL in PHY data macro 1, chip select 0. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line.	RW	0x0

Table 10-225. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_3_SHADOW

EMIF Controller

- EMIF Register Summary: [0]

Table 10-226. EMIF_EXT_PHY_CONTROL_4

Address Offset	0x0000 0218		
Physical Address	0x4C00 0218	**Instance**	EMIF
Description	Data macro 2, FIFO write enable (read DQS gate) DLL Slave Ratio Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								PHY_REG_FIFO_WE_SLAVE_RATIO5								RESERVED								PHY_REG_FIFO_WE_SLAVE_RATIO4							

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:16	PHY_REG_FIFO_WE_SLAVE_RATIO5	The user programmable ratio value for the FIFO write enable slave DLL in PHY data macro 2, chip select 1. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. **NOTE: Chip select 1 is not supported on this device.**	RW	0x0
15:11	RESERVED		R	0x0
10:0	PHY_REG_FIFO_WE_SLAVE_RATIO4	The user programmable ratio value for the FIFO write enable slave DLL in PHY data macro 2, chip select 0. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line.	RW	0x0

Table 10-227. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_4

EMIF Controller

- EMIF Register Summary: [0]
- EMIF Register Description: [1]

Table 10-228. EMIF_EXT_PHY_CONTROL_4_SHADOW

Address Offset	0x0000 021C		
Physical Address	0x4C00 021C	**Instance**	EMIF
Description	Data macro 2, FIFO write enable (read DQS gate) DLL Slave Ratio Shadow Register. Its value is loaded in register EMIF_EXT_PHY_CONTROL_4 on any frequency change.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								PHY_REG_FIFO_WE_SLAVE_RATIO5								RESERVED								PHY_REG_FIFO_WE_SLAVE_RATIO4							

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:16	PHY_REG_FIFO_WE_SLAVE_RATIO5	The user programmable ratio value for the FIFO write enable slave DLL in PHY data macro 2, chip select 1. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. **NOTE: Chip select 1 is not supported on this device.**	RW	0x0
15:11	RESERVED		R	0x0
10:0	PHY_REG_FIFO_WE_SLAVE_RATIO4	The user programmable ratio value for the FIFO write enable slave DLL in PHY data macro 2, chip select 0. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line.	RW	0x0

Table 10-229. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_4_SHADOW

EMIF Controller

- EMIF Register Summary: [0]

Table 10-230. EMIF_EXT_PHY_CONTROL_5

Address Offset	0x0000 0220		
Physical Address	0x4C00 0220	**Instance**	EMIF
Description	Data macro 3, FIFO write enable (read DQS gate) DLL Slave Ratio Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								PHY_REG_FIFO_WE_SLAVE_RATIO7								RESERVED								PHY_REG_FIFO_WE_SLAVE_RATIO6							

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:16	PHY_REG_FIFO_WE_SLAVE_RATIO7	The user programmable ratio value for the FIFO write enable slave DLL in PHY data macro 3, chip select 1. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. **NOTE: Chip select 1 is not supported on this device.**	RW	0x0
15:11	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
10:0	PHY_REG_FIFO_WE_SLAVE_RATIO6	The user programmable ratio value for the FIFO write enable slave DLL in PHY data macro 3, chip select 0. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line.	RW	0x0

Table 10-231. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_5

EMIF Controller

- EMIF Register Summary: [0]
- EMIF Register Description: [1]

Table 10-232. EMIF_EXT_PHY_CONTROL_5_SHADOW

Address Offset	0x0000 0224		
Physical Address	0x4C00 0224	**Instance**	EMIF
Description	Data macro 3, FIFO write enable (read DQS gate) DLL Slave Ratio Register. Its value is loaded in register EMIF_EXT_PHY_CONTROL_5 on any frequency change.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								PHY_REG_FIFO_WE_SLAVE_RATIO7								RESERVED								PHY_REG_FIFO_WE_SLAVE_RATIO6							

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:16	PHY_REG_FIFO_WE_SLAVE_RATIO7	The user programmable ratio value for the FIFO write enable slave DLL in PHY data macro 3, chip select 1. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. **NOTE: Chip select 1 is not supported on this device.**	RW	0x0
15:11	RESERVED		R	0x0
10:0	PHY_REG_FIFO_WE_SLAVE_RATIO6	The user programmable ratio value for the FIFO write enable slave DLL in PHY data macro 3, chip select 0. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line.	RW	0x0

Table 10-233. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_5_SHADOW

EMIF Controller

- EMIF Register Summary: [0]

Table 10-234. EMIF_EXT_PHY_CONTROL_6

Address Offset	0x0000 0228		
Physical Address	0x4C00 0228	**Instance**	EMIF
Description	ECC Data macro, FIFO write enable (read DQS gate) DLL Slave Ratio Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								PHY_REG_FIFO_WE_SLAVE_RATIO9								RESERVED								PHY_REG_FIFO_WE_SLAVE_RATIO8							

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:16	PHY_REG_FIFO_WE_SLAVE_RATIO9	The user programmable ratio value for the FIFO write enable slave DLL in PHY ECC data macro, chip select 1. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. **NOTE: Chip select 1 is not supported on this device.**	RW	0x0
15:11	RESERVED		R	0x0
10:0	PHY_REG_FIFO_WE_SLAVE_RATIO8	The user programmable ratio value for the FIFO write enable slave DLL in PHY ECC data macro, chip select 0. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line.	RW	0x0

Table 10-235. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_6

EMIF Controller

- Software Leveling: [0][1][2]
- Global Initialization: [3]
- EMIF Register Summary: [4]
- EMIF Register Description: [5]

Table 10-236. EMIF_EXT_PHY_CONTROL_6_SHADOW

Address Offset	0x0000 022C		
Physical Address	0x4C00 022C	**Instance**	EMIF
Description	ECC Data macro, FIFO write enable (read DQS gate) DLL Slave Ratio Shadow Register. Its value is loaded in register EMIF_EXT_PHY_CONTROL_6 on any frequency change.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								PHY_REG_FIFO_WE_SLAVE_RATIO9								RESERVED								PHY_REG_FIFO_WE_SLAVE_RATIO8							

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:16	PHY_REG_FIFO_WE_SLAVE_RATIO9	The user programmable ratio value for the FIFO write enable slave DLL in PHY ECC data macro, chip select 1. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. **NOTE: Chip select 1 is not supported on this device.**	RW	0x0
15:11	RESERVED		R	0x0
10:0	PHY_REG_FIFO_WE_SLAVE_RATIO8	The user programmable ratio value for the FIFO write enable slave DLL in PHY ECC data macro, chip select 0. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line.	RW	0x0

Table 10-237. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_6_SHADOW

EMIF Controller

- EMIF Register Summary: [0]

Table 10-238. EMIF_EXT_PHY_CONTROL_7

Address Offset	0x0000 0230		
Physical Address	0x4C00 0230	**Instance**	EMIF
Description	Data macro 0, read DQS DLL Slave Ratio Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								PHY_REG_RD_DQS_SLAVE_RATIO1								RESERVED								PHY_REG_RD_DQS_SLAVE_RATIO0							

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:16	PHY_REG_RD_DQS_SLAVE_RATIO1	The user programmable ratio value for the read DQS slave DLL in PHY data macro 0, chip select 1. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. **NOTE: Chip select 1 is not supported on this device.**	RW	0x40
15:10	RESERVED		R	0x0
9:0	PHY_REG_RD_DQS_SLAVE_RATIO0	The user programmable ratio value for the read DQS slave DLL in PHY data macro 0, chip select 0. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line.	RW	0x40

Table 10-239. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_7

EMIF Controller

- Global Initialization: [0]
- EMIF Register Summary: [1]
- EMIF Register Description: [2]

Table 10-240. EMIF_EXT_PHY_CONTROL_7_SHADOW

Address Offset	0x0000 0234		
Physical Address	0x4C00 0234	**Instance**	EMIF
Description	Data macro 0, read DQS DLL Slave Ratio Shadow Register. Its value is loaded in register EMIF_EXT_PHY_CONTROL_7 on any frequency change.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								PHY_REG_RD_DQS_SLAVE_RATIO1								RESERVED								PHY_REG_RD_DQS_SLAVE_RATIO0							

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:16	PHY_REG_RD_DQS_SLAVE_RATIO1	The user programmable ratio value for the read DQS slave DLL in PHY data macro 0, chip select 1. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. **NOTE: Chip select 1 is not supported on this device.**	RW	0x40
15:10	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
9:0	PHY_REG_RD_DQS_SLAVE_RATIO0	The user programmable ratio value for the read DQS slave DLL in PHY data macro 0, chip select 0. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line.	RW	0x40

Table 10-241. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_7_SHADOW

EMIF Controller

- EMIF Register Summary: [0]

Table 10-242. EMIF_EXT_PHY_CONTROL_8

Address Offset	0x0000 0238		
Physical Address	0x4C00 0238	**Instance**	EMIF
Description	Data macro 1, read DQS DLL Slave Ratio Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								PHY_REG_RD_DQS_SLAVE_RATIO3								RESERVED								PHY_REG_RD_DQS_SLAVE_RATIO2							

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:16	PHY_REG_RD_DQS_SLAVE_RATIO3	The user programmable ratio value for the read DQS slave DLL in PHY data macro 1, chip select 1. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. **NOTE: Chip select 1 is not supported on this device.**	RW	0x40
15:10	RESERVED		R	0x0
9:0	PHY_REG_RD_DQS_SLAVE_RATIO2	The user programmable ratio value for the read DQS slave DLL in PHY data macro 1, chip select 0. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line.	RW	0x40

Table 10-243. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_8

EMIF Controller

- EMIF Register Summary: [0]
- EMIF Register Description: [1]

Table 10-244. EMIF_EXT_PHY_CONTROL_8_SHADOW

Address Offset	0x0000 023C		
Physical Address	0x4C00 023C	**Instance**	EMIF
Description	Data macro 1, read DQS DLL Slave Ratio Shadow Register. Its value is loaded in register EMIF_EXT_PHY_CONTROL_8 on any frequency change.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								PHY_REG_RD_DQS_SLAVE_RATIO3								RESERVED								PHY_REG_RD_DQS_SLAVE_RATIO2							

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:16	PHY_REG_RD_DQS_SLAVE_RATIO3	The user programmable ratio value for the read DQS slave DLL in PHY data macro 1, chip select 1. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. **NOTE: Chip select 1 is not supported on this device.**	RW	0x40
15:10	RESERVED		R	0x0
9:0	PHY_REG_RD_DQS_SLAVE_RATIO2	The user programmable ratio value for the read DQS slave DLL in PHY data macro 1, chip select 0. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line.	RW	0x40

Table 10-245. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_8_SHADOW

EMIF Controller

- EMIF Register Summary: [0]

Table 10-246. EMIF_EXT_PHY_CONTROL_9

Address Offset	0x0000 0240		
Physical Address	0x4C00 0240	**Instance**	EMIF
Description	Data macro 2, read DQS DLL Slave Ratio Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								PHY_REG_RD_DQS_SLAVE_RATIO5								RESERVED								PHY_REG_RD_DQS_SLAVE_RATIO4							

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:16	PHY_REG_RD_DQS_SLAVE_RATIO5	The user programmable ratio value for the read DQS slave DLL in PHY data macro 2, chip select 1. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. **NOTE: Chip select 1 is not supported on this device.**	RW	0x40
15:10	RESERVED		R	0x0
9:0	PHY_REG_RD_DQS_SLAVE_RATIO4	The user programmable ratio value for the read DQS slave DLL in PHY data macro 2, chip select 0. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line.	RW	0x40

Table 10-247. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_9

EMIF Controller

- EMIF Register Summary: [0]
- EMIF Register Description: [1]

Table 10-248. EMIF_EXT_PHY_CONTROL_9_SHADOW

Address Offset	0x0000 0244		
Physical Address	0x4C00 0244	**Instance**	EMIF
Description	Data macro 2, read DQS DLL Slave Ratio Shadow Register. Its value is loaded in register EMIF_EXT_PHY_CONTROL_9 on any frequency change.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								PHY_REG_RD_DQS_SLAVE_RATIO5								RESERVED								PHY_REG_RD_DQS_SLAVE_RATIO4							

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:16	PHY_REG_RD_DQS_SLAVE_RATIO5	The user programmable ratio value for the read DQS slave DLL in PHY data macro 2, chip select 1. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. **NOTE: Chip select 1 is not supported on this device.**	RW	0x40
15:10	RESERVED		R	0x0
9:0	PHY_REG_RD_DQS_SLAVE_RATIO4	The user programmable ratio value for the read DQS slave DLL in PHY data macro 2, chip select 0. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line.	RW	0x40

Table 10-249. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_9_SHADOW

EMIF Controller

- EMIF Register Summary: [0]

Table 10-250. EMIF_EXT_PHY_CONTROL_10

Address Offset	0x0000 0248		
Physical Address	0x4C00 0248	**Instance**	EMIF
Description	Data macro 3, read DQS DLL Slave Ratio Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								PHY_REG_RD_DQS_SLAVE_RATIO7								RESERVED								PHY_REG_RD_DQS_SLAVE_RATIO6							

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:16	PHY_REG_RD_DQS_SLAVE_RATIO7	The user programmable ratio value for the read DQS slave DLL in PHY data macro 3, chip select 1. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. **NOTE: Chip select 1 is not supported on this device.**	RW	0x40
15:10	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
9:0	PHY_REG_RD_DQS_SLAVE_RATIO6	The user programmable ratio value for the read DQS slave DLL in PHY data macro 3, chip select 0. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line.	RW	0x40

Table 10-251. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_10

EMIF Controller

- EMIF Register Summary: [0]
- EMIF Register Description: [1]

Table 10-252. EMIF_EXT_PHY_CONTROL_10_SHADOW

Address Offset	0x0000 024C		
Physical Address	0x4C00 024C	**Instance**	EMIF
Description	Data macro 3, read DQS DLL Slave Ratio Shadow Register. Its value is loaded in register EMIF_EXT_PHY_CONTROL_10 on any frequency change.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								PHY_REG_RD_DQS_SLAVE_RATIO7								RESERVED								PHY_REG_RD_DQS_SLAVE_RATIO6							

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:16	PHY_REG_RD_DQS_SLAVE_RATIO7	The user programmable ratio value for the read DQS slave DLL in PHY data macro 3, chip select 1. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. **NOTE: Chip select 1 is not supported on this device.**	RW	0x40
15:10	RESERVED		R	0x0
9:0	PHY_REG_RD_DQS_SLAVE_RATIO6	The user programmable ratio value for the read DQS slave DLL in PHY data macro 3, chip select 0. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line.	RW	0x40

Table 10-253. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_10_SHADOW

EMIF Controller

- EMIF Register Summary: [0]

Table 10-254. EMIF_EXT_PHY_CONTROL_11

Address Offset	0x0000 0250		
Physical Address	0x4C00 0250	**Instance**	EMIF
Description	ECC Data macro, read DQS DLL Slave Ratio Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								PHY_REG_RD_DQS_SLAVE_RATIO9								RESERVED								PHY_REG_RD_DQS_SLAVE_RATIO8							

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:16	PHY_REG_RD_DQS_SLAVE_RATIO9	The user programmable ratio value for the read DQS slave DLL in PHY ECC data macro, chip select 1. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. **NOTE: Chip select 1 is not supported on this device.**	RW	0x40
15:10	RESERVED		R	0x0
9:0	PHY_REG_RD_DQS_SLAVE_RATIO8	The user programmable ratio value for the read DQS slave DLL in PHY ECC data macro, chip select 0. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line.	RW	0x40

Table 10-255. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_11

EMIF Controller

- Global Initialization: [0]
- EMIF Register Summary: [1]
- EMIF Register Description: [2]

Table 10-256. EMIF_EXT_PHY_CONTROL_11_SHADOW

Address Offset	0x0000 0254		
Physical Address	0x4C00 0254	**Instance**	EMIF
Description	ECC Data macro, read DQS DLL Slave Ratio Shadow Register. Its value is loaded in register EMIF_EXT_PHY_CONTROL_11 on any frequency change.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								PHY_REG_RD_DQS_SLAVE_RATIO9								RESERVED								PHY_REG_RD_DQS_SLAVE_RATIO8							

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:16	PHY_REG_RD_DQS_SLAVE_RATIO9	The user programmable ratio value for the read DQS slave DLL in PHY ECC data macro, chip select 1. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. **NOTE: Chip select 1 is not supported on this device.**	RW	0x40
15:10	RESERVED		R	0x0
9:0	PHY_REG_RD_DQS_SLAVE_RATIO8	The user programmable ratio value for the read DQS slave DLL in PHY ECC data macro, chip select 0. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line.	RW	0x40

Table 10-257. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_11_SHADOW

EMIF Controller

- EMIF Register Summary: [0]

Table 10-258. EMIF_EXT_PHY_CONTROL_12

Address Offset	0x0000 0258		
Physical Address	0x4C00 0258	**Instance**	EMIF
Description	Data macro 0, write DQ (data) DLL Slave Ratio Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								PHY_REG_WR_DATA_SLAVE_RATIO1								RESERVED								PHY_REG_WR_DATA_SLAVE_RATIO0							

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:16	PHY_REG_WR_DATA_SLAVE_RATIO1	The user programmable ratio value for the write DQ slave DLL in PHY data macro 0, chip select 1. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. **NOTE: Chip select 1 is not supported on this device.**	RW	0x40
15:10	RESERVED		R	0x0
9:0	PHY_REG_WR_DATA_SLAVE_RATIO0	The user programmable ratio value for the write DQ slave DLL in PHY data macro 0, chip select 0. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line.	RW	0x40

Table 10-259. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_12

EMIF Controller

- Global Initialization: [0]
- EMIF Register Summary: [1]
- EMIF Register Description: [2]

Table 10-260. EMIF_EXT_PHY_CONTROL_12_SHADOW

Address Offset	0x0000 025C		
Physical Address	0x4C00 025C	**Instance**	EMIF
Description	Data macro 0, write DQ (data) DLL Slave Ratio Shadow Register. Its value is loaded in register EMIF_EXT_PHY_CONTROL_12 on any frequency change.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								PHY_REG_WR_DATA_SLAVE_RATIO1								RESERVED								PHY_REG_WR_DATA_SLAVE_RATIO0							

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:16	PHY_REG_WR_DATA_SLAVE_RATIO1	The user programmable ratio value for the write DQ slave DLL in PHY data macro 0, chip select 1. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. **NOTE: Chip select 1 is not supported on this device.**	RW	0x40
15:10	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
9:0	PHY_REG_WR_DATA_SLAVE_RATIO0	The user programmable ratio value for the write DQ slave DLL in PHY data macro 0, chip select 0. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line.	RW	0x40

Table 10-261. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_12_SHADOW

EMIF Controller

- EMIF Register Summary: [0]

Table 10-262. EMIF_EXT_PHY_CONTROL_13

Address Offset	0x0000 0260		
Physical Address	0x4C00 0260	**Instance**	EMIF
Description	Data macro 1, write DQ (data) DLL Slave Ratio Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								PHY_REG_WR_DATA_SLAVE_RATIO3								RESERVED								PHY_REG_WR_DATA_SLAVE_RATIO2							

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:16	PHY_REG_WR_DATA_SLAVE_RATIO3	The user programmable ratio value for the write DQ slave DLL in PHY data macro 1, chip select 1. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. **NOTE: Chip select 1 is not supported on this device.**	RW	0x40
15:10	RESERVED		R	0x0
9:0	PHY_REG_WR_DATA_SLAVE_RATIO2	The user programmable ratio value for the write DQ slave DLL in PHY data macro 1, chip select 0. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line.	RW	0x40

Table 10-263. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_13

EMIF Controller

- EMIF Register Summary: [0]
- EMIF Register Description: [1]

Table 10-264. EMIF_EXT_PHY_CONTROL_13_SHADOW

Address Offset	0x0000 0264		
Physical Address	0x4C00 0264	**Instance**	EMIF
Description	Data macro 1, write DQ (data) DLL Slave Ratio Shadow Register. Its value is loaded in register EMIF_EXT_PHY_CONTROL_13 on any frequency change.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								PHY_REG_WR_DATA_SLAVE_RATIO3								RESERVED								PHY_REG_WR_DATA_SLAVE_RATIO2							

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:16	PHY_REG_WR_DATA_SLAVE_RATIO3	The user programmable ratio value for the write DQ slave DLL in PHY data macro 1, chip select 1. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. **NOTE: Chip select 1 is not supported on this device.**	RW	0x40
15:10	RESERVED		R	0x0
9:0	PHY_REG_WR_DATA_SLAVE_RATIO2	The user programmable ratio value for the write DQ slave DLL in PHY data macro 1, chip select 0. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line.	RW	0x40

Table 10-265. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_13_SHADOW

EMIF Controller

- EMIF Register Summary: [0]

Table 10-266. EMIF_EXT_PHY_CONTROL_14

Address Offset	0x0000 0268		
Physical Address	0x4C00 0268	**Instance**	EMIF
Description	Data macro 2, write DQ (data) DLL Slave Ratio Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								PHY_REG_WR_DATA_SLAVE_RATIO5								RESERVED								PHY_REG_WR_DATA_SLAVE_RATIO4							

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:16	PHY_REG_WR_DATA_SLAVE_RATIO5	The user programmable ratio value for the write DQ slave DLL in PHY data macro 2, chip select 1. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. **NOTE: Chip select 1 is not supported on this device.**	RW	0x40
15:10	RESERVED		R	0x0
9:0	PHY_REG_WR_DATA_SLAVE_RATIO4	The user programmable ratio value for the write DQ slave DLL in PHY data macro 2, chip select 0. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line.	RW	0x40

Table 10-267. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_14

EMIF Controller

- EMIF Register Summary: [0]
- EMIF Register Description: [1]

Table 10-268. EMIF_EXT_PHY_CONTROL_14_SHADOW

Address Offset	0x0000 026C		
Physical Address	0x4C00 026C	**Instance**	EMIF
Description	Data macro 2, write DQ (data) DLL Slave Ratio Shadow Register. Its value is loaded in register EMIF_EXT_PHY_CONTROL_14 on any frequency change.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								PHY_REG_WR_DATA_SLAVE_RATIO5								RESERVED								PHY_REG_WR_DATA_SLAVE_RATIO4							

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:16	PHY_REG_WR_DATA_SLAVE_RATIO5	The user programmable ratio value for the write DQ slave DLL in PHY data macro 2, chip select 1. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. **NOTE: Chip select 1 is not supported on this device.**	RW	0x40
15:10	RESERVED		R	0x0
9:0	PHY_REG_WR_DATA_SLAVE_RATIO4	The user programmable ratio value for the write DQ slave DLL in PHY data macro 2, chip select 0. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line.	RW	0x40

Table 10-269. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_14_SHADOW

EMIF Controller

- EMIF Register Summary: [0]

Table 10-270. EMIF_EXT_PHY_CONTROL_15

Address Offset	0x0000 0270		
Physical Address	0x4C00 0270	**Instance**	EMIF
Description	Data macro 3, write DQ (data) DLL Slave Ratio Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								PHY_REG_WR_DATA_SLAVE_RATIO7								RESERVED								PHY_REG_WR_DATA_SLAVE_RATIO6							

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:16	PHY_REG_WR_DATA_SLAVE_RATIO7	The user programmable ratio value for the write DQ slave DLL in PHY data macro 3, chip select 1. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. **NOTE: Chip select 1 is not supported on this device.**	RW	0x40
15:10	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
9:0	PHY_REG_WR_DATA_SLAVE_RATIO6	The user programmable ratio value for the write DQ slave DLL in PHY data macro 3, chip select 0. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line.	RW	0x40

Table 10-271. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_15

EMIF Controller

- EMIF Register Summary: [0]
- EMIF Register Description: [1]

Table 10-272. EMIF_EXT_PHY_CONTROL_15_SHADOW

Address Offset	0x0000 0274		
Physical Address	0x4C00 0274	**Instance**	EMIF
Description	Data macro 3, write DQ (data) DLL Slave Ratio Shadow Register. Its value is loaded in register EMIF_EXT_PHY_CONTROL_15 on any frequency change.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								PHY_REG_WR_DATA_SLAVE_RATIO7								RESERVED								PHY_REG_WR_DATA_SLAVE_RATIO6							

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:16	PHY_REG_WR_DATA_SLAVE_RATIO7	The user programmable ratio value for the write DQ slave DLL in PHY data macro 3, chip select 1. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. **NOTE: Chip select 1 is not supported on this device.**	RW	0x40
15:10	RESERVED		R	0x0
9:0	PHY_REG_WR_DATA_SLAVE_RATIO6	The user programmable ratio value for the write DQ slave DLL in PHY data macro 3, chip select 0. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line.	RW	0x40

Table 10-273. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_15_SHADOW

EMIF Controller

- EMIF Register Summary: [0]

Table 10-274. EMIF_EXT_PHY_CONTROL_16

Address Offset	0x0000 0278		
Physical Address	0x4C00 0278	**Instance**	EMIF
Description	ECC Data macro, write DQ (data) DLL Slave Ratio Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								PHY_REG_WR_DATA_SLAVE_RATIO9								RESERVED								PHY_REG_WR_DATA_SLAVE_RATIO8							

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:16	PHY_REG_WR_DATA_SLAVE_RATIO9	The user programmable ratio value for the write DQ slave DLL in PHY ECC data macro, chip select 1. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. **NOTE: Chip select 1 is not supported on this device.**	RW	0x40
15:10	RESERVED		R	0x0
9:0	PHY_REG_WR_DATA_SLAVE_RATIO8	The user programmable ratio value for the write DQ slave DLL in PHY ECC data macro, chip select 0. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line.	RW	0x40

Table 10-275. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_16

EMIF Controller

- EMIF Register Summary: [0]
- EMIF Register Description: [1]

Table 10-276. EMIF_EXT_PHY_CONTROL_16_SHADOW

Address Offset	0x0000 027C		
Physical Address	0x4C00 027C	**Instance**	EMIF
Description	ECC Data macro, write DQ (data) DLL Slave Ratio Shadow Register. Its value is loaded in register EMIF_EXT_PHY_CONTROL_16 on any frequency change.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								PHY_REG_WR_DATA_SLAVE_RATIO9								RESERVED								PHY_REG_WR_DATA_SLAVE_RATIO8							

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:16	PHY_REG_WR_DATA_SLAVE_RATIO9	The user programmable ratio value for the write DQ slave DLL in PHY ECC data macro, chip select 1. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. **NOTE: Chip select 1 is not supported on this device.**	RW	0x40
15:10	RESERVED		R	0x0
9:0	PHY_REG_WR_DATA_SLAVE_RATIO8	The user programmable ratio value for the write DQ slave DLL in PHY ECC data macro, chip select 0. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line.	RW	0x40

Table 10-277. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_16_SHADOW

EMIF Controller

- EMIF Register Summary: [0]

Table 10-278. EMIF_EXT_PHY_CONTROL_17

Address Offset	0x0000 0280		
Physical Address	0x4C00 0280	**Instance**	EMIF
Description	Data macro 0, write DQS DLL Slave Ratio Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								PHY_REG_WR_DQS_SLAVE_RATIO1								RESERVED								PHY_REG_WR_DQS_SLAVE_RATIO0							

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:16	PHY_REG_WR_DQS_SLAVE_RATIO1	The user programmable ratio value for the write DQS slave DLL in PHY data macro 0, chip select 1. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. **NOTE: Chip select 1 is not supported on this device.**	RW	0x0
15:10	RESERVED		R	0x0
9:0	PHY_REG_WR_DQS_SLAVE_RATIO0	The user programmable ratio value for the write DQS slave DLL in PHY data macro 0, chip select 0. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line.	RW	0x0

Table 10-279. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_17

EMIF Controller

- EMIF Register Summary: [0]
- EMIF Register Description: [1]

Table 10-280. EMIF_EXT_PHY_CONTROL_17_SHADOW

Address Offset	0x0000 0284		
Physical Address	0x4C00 0284	**Instance**	EMIF
Description	Data macro 0, write DQS DLL Slave Ratio Shadow Register. Its value is loaded in register EMIF_EXT_PHY_CONTROL_17 on any frequency change.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								PHY_REG_WR_DQS_SLAVE_RATIO1								RESERVED								PHY_REG_WR_DQS_SLAVE_RATIO0							

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:16	PHY_REG_WR_DQS_SLAVE_RATIO1	The user programmable ratio value for the write DQS slave DLL in PHY data macro 0, chip select 1. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. **NOTE: Chip select 1 is not supported on this device.**	RW	0x0
15:10	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
9:0	PHY_REG_WR_DQS_SLAVE_RATIO0	The user programmable ratio value for the write DQS slave DLL in PHY data macro 0, chip select 0. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line.	RW	0x0

Table 10-281. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_17_SHADOW

EMIF Controller

- EMIF Register Summary: [0]

Table 10-282. EMIF_EXT_PHY_CONTROL_18

Address Offset	0x0000 0288		
Physical Address	0x4C00 0288	**Instance**	EMIF
Description	Data macro 1, write DQS DLL Slave Ratio Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								PHY_REG_WR_DQS_SLAVE_RATIO3								RESERVED								PHY_REG_WR_DQS_SLAVE_RATIO2							

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:16	PHY_REG_WR_DQS_SLAVE_RATIO3	The user programmable ratio value for the write DQS slave DLL in PHY data macro 1, chip select 1. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. **NOTE: Chip select 1 is not supported on this device.**	RW	0x0
15:10	RESERVED		R	0x0
9:0	PHY_REG_WR_DQS_SLAVE_RATIO2	The user programmable ratio value for the write DQS slave DLL in PHY data macro 1, chip select 0. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line.	RW	0x0

Table 10-283. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_18

EMIF Controller

- EMIF Register Summary: [0]
- EMIF Register Description: [1]

Table 10-284. EMIF_EXT_PHY_CONTROL_18_SHADOW

Address Offset	0x0000 028C		
Physical Address	0x4C00 028C	**Instance**	EMIF
Description	Data macro 1, write DQS DLL Slave Ratio Shadow Register. Its value is loaded in register EMIF_EXT_PHY_CONTROL_18 on any frequency change.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								PHY_REG_WR_DQS_SLAVE_RATIO3								RESERVED								PHY_REG_WR_DQS_SLAVE_RATIO2							

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:16	PHY_REG_WR_DQS_SLAVE_RATIO3	The user programmable ratio value for the write DQS slave DLL in PHY data macro 1, chip select 1. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. **NOTE: Chip select 1 is not supported on this device.**	RW	0x0
15:10	RESERVED		R	0x0
9:0	PHY_REG_WR_DQS_SLAVE_RATIO2	The user programmable ratio value for the write DQS slave DLL in PHY data macro 1, chip select 0. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line.	RW	0x0

Table 10-285. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_18_SHADOW

EMIF Controller

- EMIF Register Summary: [0]

Table 10-286. EMIF_EXT_PHY_CONTROL_19

Address Offset	0x0000 0290		
Physical Address	0x4C00 0290	**Instance**	EMIF
Description	Data macro 2, write DQS DLL Slave Ratio Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								PHY_REG_WR_DQS_SLAVE_RATIO5								RESERVED								PHY_REG_WR_DQS_SLAVE_RATIO4							

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:16	PHY_REG_WR_DQS_SLAVE_RATIO5	The user programmable ratio value for the write DQS slave DLL in PHY data macro 2, chip select 1. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. **NOTE: Chip select 1 is not supported on this device.**	RW	0x0
15:10	RESERVED		R	0x0
9:0	PHY_REG_WR_DQS_SLAVE_RATIO4	The user programmable ratio value for the write DQS slave DLL in PHY data macro 2, chip select 0. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line.	RW	0x0

Table 10-287. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_19

EMIF Controller

- EMIF Register Summary: [0]
- EMIF Register Description: [1]

Table 10-288. EMIF_EXT_PHY_CONTROL_19_SHADOW

Address Offset	0x0000 0294		
Physical Address	0x4C00 0294	**Instance**	EMIF
Description	Data macro 2, write DQS DLL Slave Ratio Shadow Register. Its value is loaded in register EMIF_EXT_PHY_CONTROL_19 on any frequency change.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								PHY_REG_WR_DQS_SLAVE_RATIO5								RESERVED								PHY_REG_WR_DQS_SLAVE_RATIO4							

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:16	PHY_REG_WR_DQS_SLAVE_RATIO5	The user programmable ratio value for the write DQS slave DLL in PHY data macro 2, chip select 1. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. **NOTE: Chip select 1 is not supported on this device.**	RW	0x0
15:10	RESERVED		R	0x0
9:0	PHY_REG_WR_DQS_SLAVE_RATIO4	The user programmable ratio value for the write DQS slave DLL in PHY data macro 2, chip select 0. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line.	RW	0x0

Table 10-289. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_19_SHADOW

EMIF Controller

- EMIF Register Summary: [0]

Table 10-290. EMIF_EXT_PHY_CONTROL_20

Address Offset	0x0000 0298		
Physical Address	0x4C00 0298	**Instance**	EMIF
Description	Data macro 3, write DQS DLL Slave Ratio Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								PHY_REG_WR_DQS_SLAVE_RATIO7								RESERVED								PHY_REG_WR_DQS_SLAVE_RATIO6							

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:16	PHY_REG_WR_DQS_SLAVE_RATIO7	The user programmable ratio value for the write DQS slave DLL in PHY data macro 3, chip select 1. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. **NOTE: Chip select 1 is not supported on this device.**	RW	0x0
15:10	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
9:0	PHY_REG_WR_DQS_SLAVE_RATIO6	The user programmable ratio value for the write DQS slave DLL in PHY data macro 3, chip select 0. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line.	RW	0x0

Table 10-291. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_20

EMIF Controller

- EMIF Register Summary: [0]
- EMIF Register Description: [1]

Table 10-292. EMIF_EXT_PHY_CONTROL_20_SHADOW

Address Offset	0x0000 029C		
Physical Address	0x4C00 029C	**Instance**	EMIF
Description	Data macro 3, write DQS DLL Slave Ratio Shadow Register. Its value is loaded in register EMIF_EXT_PHY_CONTROL_20 on any frequency change.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								PHY_REG_WR_DQS_SLAVE_RATIO7								RESERVED								PHY_REG_WR_DQS_SLAVE_RATIO6							

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:16	PHY_REG_WR_DQS_SLAVE_RATIO7	The user programmable ratio value for the write DQS slave DLL in PHY data macro 3, chip select 1. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. **NOTE: Chip select 1 is not supported on this device.**	RW	0x0
15:10	RESERVED		R	0x0
9:0	PHY_REG_WR_DQS_SLAVE_RATIO6	The user programmable ratio value for the write DQS slave DLL in PHY data macro 3, chip select 0. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line.	RW	0x0

Table 10-293. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_20_SHADOW

EMIF Controller

- EMIF Register Summary: [0]

Table 10-294. EMIF_EXT_PHY_CONTROL_21

Address Offset	0x0000 02A0		
Physical Address	0x4C00 02A0	**Instance**	EMIF
Description	ECC Data macro, write DQS DLL Slave Ratio Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								PHY_REG_WR_DQS_SLAVE_RATIO9								RESERVED								PHY_REG_WR_DQS_SLAVE_RATIO8							

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:16	PHY_REG_WR_DQS_SLAVE_RATIO9	The user programmable ratio value for the write DQS slave DLL in PHY ECC data macro, chip select 1. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. **NOTE: Chip select 1 is not supported on this device.**	RW	0x0
15:10	RESERVED		R	0x0
9:0	PHY_REG_WR_DQS_SLAVE_RATIO8	The user programmable ratio value for the write DQS slave DLL in PHY ECC data macro, chip select 0. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line.	RW	0x0

Table 10-295. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_21

EMIF Controller

- Full Leveling: [0]
- Global Initialization: [1][2][3][4]
- EMIF Register Summary: [5]
- EMIF Register Description: [6]

Table 10-296. EMIF_EXT_PHY_CONTROL_21_SHADOW

Address Offset	0x0000 02A4		
Physical Address	0x4C00 02A4	**Instance**	EMIF
Description	ECC Data macro, write DQS DLL Slave Ratio Shadow Register. Its value is loaded in register EMIF_EXT_PHY_CONTROL_21 on any frequency change.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								PHY_REG_WR_DQS_SLAVE_RATIO9								RESERVED								PHY_REG_WR_DQS_SLAVE_RATIO8							

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:16	PHY_REG_WR_DQS_SLAVE_RATIO9	The user programmable ratio value for the write DQS slave DLL in PHY ECC data macro, chip select 1. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. **NOTE: Chip select 1 is not supported on this device.**	RW	0x0
15:10	RESERVED		R	0x0
9:0	PHY_REG_WR_DQS_SLAVE_RATIO8	The user programmable ratio value for the write DQS slave DLL in PHY ECC data macro, chip select 0. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line.	RW	0x0

Table 10-297. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_21_SHADOW

EMIF Controller

- Global Initialization: [0]
- EMIF Register Summary: [1]

Table 10-298. EMIF_EXT_PHY_CONTROL_22

Address Offset	0x0000 02A8		
Physical Address	0x4C00 02A8	**Instance**	EMIF
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								PHY_REG_FIFO_WE_IN_DELAY								RESERVED								PHY_REG_CTRL_SLAVE_DELAY							

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	PHY_REG_FIFO_WE_IN_DELAY	The user programmable FIFO write enable delay value used when DLL_OVERRIDE = 1.	RW	0x80
15:9	RESERVED		R	0x0
8:0	PHY_REG_CTRL_SLAVE_DELAY	The user programmable command delay value used when DLL_OVERRIDE = 1.	RW	0x80

Table 10-299. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_22

EMIF Controller

- Global Initialization: [0][1][2][3][4]
- EMIF Register Summary: [5]

Table 10-300. EMIF_EXT_PHY_CONTROL_22_SHADOW

Address Offset	0x0000 02AC		
Physical Address	0x4C00 02AC	**Instance**	EMIF
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								PHY_REG_FIFO_WE_IN_DELAY								RESERVED								PHY_REG_CTRL_SLAVE_DELAY							

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	PHY_REG_FIFO_WE_IN_DELAY	The user programmable FIFO write enable delay value used when DLL_OVERRIDE = 1.	RW	0x80
15:9	RESERVED		R	0x0
8:0	PHY_REG_CTRL_SLAVE_DELAY	The user programmable command delay value used when DLL_OVERRIDE = 1.	RW	0x80

Table 10-301. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_22_SHADOW

EMIF Controller

- Global Initialization: [0]
- EMIF Register Summary: [1]

Table 10-302. EMIF_EXT_PHY_CONTROL_23

Address Offset	0x0000 02B0		
Physical Address	0x4C00 02B0	**Instance**	EMIF
Description			

Table 10-302. EMIF_EXT_PHY_CONTROL_23 (continued)

Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								PHY_REG_WR_DQS_SLAVE_DELAY								RESERVED								PHY_REG_RD_DQS_SLAVE_DELAY							

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	PHY_REG_WR_DQS_SLAVE_DELAY	The user programmable write DQS delay value used when DLL_OVERRIDE = 1.	RW	0x80
15:9	RESERVED		R	0x0
8:0	PHY_REG_RD_DQS_SLAVE_DELAY	The user programmable read DQS delay value used when DLL_OVERRIDE = 1.	RW	0x80

Table 10-303. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_23

EMIF Controller

- Global Initialization: [0][1][2][3][4]
- EMIF Register Summary: [5]

Table 10-304. EMIF_EXT_PHY_CONTROL_23_SHADOW

Address Offset	0x0000 02B4		
Physical Address	0x4C00 02B4	Instance	EMIF
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								PHY_REG_WR_DQS_SLAVE_DELAY								RESERVED								PHY_REG_RD_DQS_SLAVE_DELAY							

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	PHY_REG_WR_DQS_SLAVE_DELAY	The user programmable write DQS delay value used when DLL_OVERRIDE = 1.	RW	0x80
15:9	RESERVED		R	0x0
8:0	PHY_REG_RD_DQS_SLAVE_DELAY	The user programmable read DQS delay value used when DLL_OVERRIDE = 1.	RW	0x80

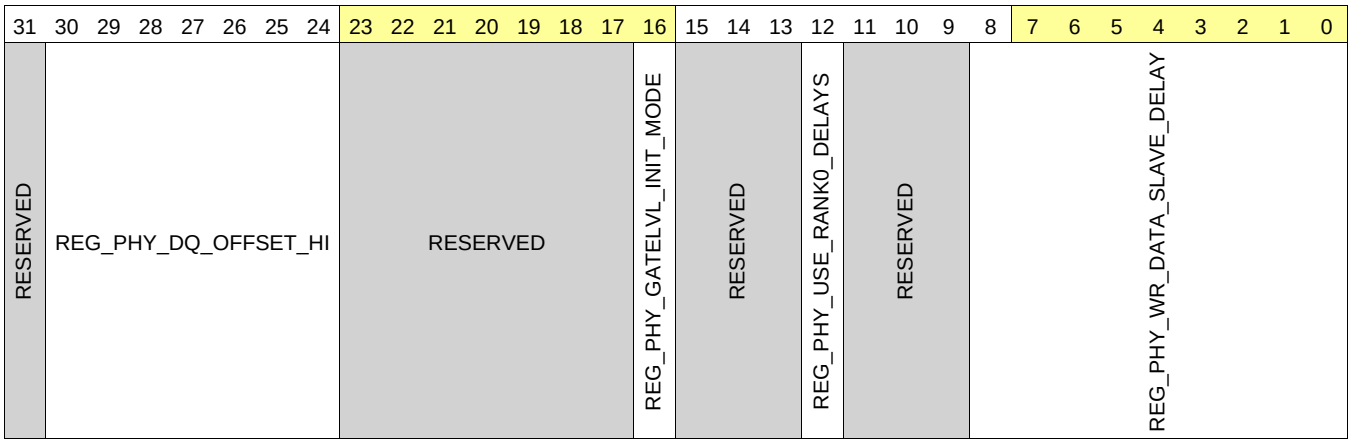
Table 10-305. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_23_SHADOW

EMIF Controller

- Global Initialization: [0]
- EMIF Register Summary: [1]

Table 10-306. EMIF_EXT_PHY_CONTROL_24

Address Offset	0x0000 02B8		
Physical Address	0x4C00 02B8	**Instance**	EMIF
Description			
Type	RW		



Bits	Field Name	Description	Type	Reset
31	RESERVED		R	0x0
30:24	REG_PHY_DQ_OFFSET_HI	The user programmable offset ratio value from write DQS to write DQ. This value is used for the write DQ slave DLL in PHY ECC data macro. The ratio represents the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 (in addition to the write DQS delay) to get the delay value for the slave delay line.	RW	0x0
23:17	RESERVED		R	0x0
16	REG_PHY_GATELVL_INIT_MODE	The user programmable init ratio selection mode. Recommended value is 0x1.	RW	0x1
15:13	RESERVED		R	0x0
12	REG_PHY_USE_RANK0_DELAYS	Delay selection. Chip select 0 delay line ratios are used for all chip selects when set to 1. Each chip select uses its own delays when set to 0.	RW	0x0
11:9	RESERVED		R	0x0
8:0	REG_PHY_WR_DATA_SLAVE_DELAY	The user programmable write DQ delay value used when DLL_OVERRIDE = 1.	RW	0x80

Table 10-307. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_24

EMIF Controller

- Global Initialization: [0][1][2][3][4][5][6]
- EMIF Register Summary: [7]

Table 10-308. EMIF_EXT_PHY_CONTROL_24_SHADOW

Address Offset	0x0000 02BC		
Physical Address	0x4C00 02BC	**Instance**	EMIF
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	REG_PHY_DQ_OFFSET_HI							RESERVED							REG_PHY_GATELVL_INIT_MODE	RESERVED			REG_PHY_USE_RANK0_DELAYS	RESERVED				REG_PHY_WR_DATA_SLAVE_DELAY							

Bits	Field Name	Description	Type	Reset
31	RESERVED		R	0x0
30:24	REG_PHY_DQ_OFFSET_HI	The user programmable offset ratio value from write DQS to write DQ. This value is used for the write DQ slave DLL in PHY ECC data macro. The ratio represents the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 (in addition to the write DQS delay) to get the delay value for the slave delay line.	RW	0x0
23:17	RESERVED		R	0x0
16	REG_PHY_GATELVL_INIT_MODE	The user programmable init ratio selection mode. Recommended value is 0x1.	RW	0x1
15:13	RESERVED		R	0x0
12	REG_PHY_USE_RANK0_DELAYS	Delay selection. Chip select 0 delay line ratios are used for all chip selects when set to 1. Each chip select uses its own delays when set to 0.	RW	0x0
11:9	RESERVED		R	0x0
8:0	REG_PHY_WR_DATA_SLAVE_DELAY	The user programmable write DQ delay value used when DLL_OVERRIDE = 1.	RW	0x80

Table 10-309. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_24_SHADOW

EMIF Controller

- Global Initialization: [0]
- EMIF Register Summary: [1]

Table 10-310. EMIF_EXT_PHY_CONTROL_25

Address Offset	0x0000 02C0		
Physical Address	0x4C00 02C0	**Instance**	EMIF
Description	DQ DLL Slave Ratio Offset Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								REG_PHY_DQ_OFFSET3								REG_PHY_DQ_OFFSET2								REG_PHY_DQ_OFFSET1							
REG_PHY_DQ_OFFSET0																															

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:21	REG_PHY_DQ_OFFSET3	The user programmable offset ratio value from write DQS to write DQ. This value is used for the write DQ slave DLL in PHY data macro 3. The ratio represents the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 (in addition to the write DQS delay) to get the delay value for the slave delay line.	RW	0x0
20:14	REG_PHY_DQ_OFFSET2	The user programmable offset ratio value from write DQS to write DQ. This value is used for the write DQ slave DLL in PHY data macro 2. The ratio represents the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 (in addition to the write DQS delay) to get the delay value for the slave delay line.	RW	0x0
13:7	REG_PHY_DQ_OFFSET1	The user programmable offset ratio value from write DQS to write DQ. This value is used for the write DQ slave DLL in PHY data macro 1. The ratio represents the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 (in addition to the write DQS delay) to get the delay value for the slave delay line.	RW	0x0
6:0	REG_PHY_DQ_OFFSET0	The user programmable offset ratio value from write DQS to write DQ. This value is used for the write DQ slave DLL in PHY data macro 0. The ratio represents the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 (in addition to the write DQS delay) to get the delay value for the slave delay line.	RW	0x0

Table 10-311. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_25

EMIF Controller

- Software Leveling: [0][1]
- Global Initialization: [2][3][4][5][6][7][8]
- EMIF Register Summary: [9]
- EMIF Register Description: [10]

Table 10-312. EMIF_EXT_PHY_CONTROL_25_SHADOW

Address Offset	0x0000 02C4		
Physical Address	0x4C00 02C4	**Instance**	EMIF
Description	DQ DLL Slave Ratio Offset Shadow Register. Its value is loaded in register EMIF_EXT_PHY_CONTROL_25 on any frequency change.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								REG_PHY_DQ_OFFSET3								REG_PHY_DQ_OFFSET2								REG_PHY_DQ_OFFSET1							
REG_PHY_DQ_OFFSET0																															

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:21	REG_PHY_DQ_OFFSET3	The user programmable offset ratio value from write DQS to write DQ. This value is used for the write DQ slave DLL in PHY data macro 3. The ratio represents the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 (in addition to the write DQS delay) to get the delay value for the slave delay line.	RW	0x0
20:14	REG_PHY_DQ_OFFSET2	The user programmable offset ratio value from write DQS to write DQ. This value is used for the write DQ slave DLL in PHY data macro 2. The ratio represents the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 (in addition to the write DQS delay) to get the delay value for the slave delay line.	RW	0x0
13:7	REG_PHY_DQ_OFFSET1	The user programmable offset ratio value from write DQS to write DQ. This value is used for the write DQ slave DLL in PHY data macro 1. The ratio represents the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 (in addition to the write DQS delay) to get the delay value for the slave delay line.	RW	0x0
6:0	REG_PHY_DQ_OFFSET0	The user programmable offset ratio value from write DQS to write DQ. This value is used for the write DQ slave DLL in PHY data macro 0. The ratio represents the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 (in addition to the write DQS delay) to get the delay value for the slave delay line.	RW	0x0

Table 10-313. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_25_SHADOW

EMIF Controller

- Global Initialization: [0]
- EMIF Register Summary: [1]

Table 10-314. EMIF_EXT_PHY_CONTROL_26

Address Offset	0x0000 02C8		
Physical Address	0x4C00 02C8	**Instance**	EMIF
Description	Data macro 0, FIFO write enable (read DQS gate) DLL Slave Init Ratio Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								REG_PHY_GATELVL_INIT_RATIO1								RESERVED								REG_PHY_GATELVL_INIT_RATIO0							

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:16	REG_PHY_GATELVL_INIT_RATIO1	The user programmable initialization ratio value used by the gate training finite state machine (hardware leveling) for the FIFO write enable slave DLL in PHY data macro 0, chip select 1. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. **NOTE: Chip select 1 is not supported on this device.**	RW	0x150
15:11	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
10:0	REG_PHY_GATELVL_INIT_RATIO0	The user programmable initialization ratio value used by the gate training finite state machine (hardware leveling) for the FIFO write enable slave DLL in PHY data macro 0, chip select 0. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line.	RW	0x150

Table 10-315. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_26

EMIF Controller

- Software Leveling: [0][1][2]
- Global Initialization: [3][4]
- EMIF Register Summary: [5]
- EMIF Register Description: [6]

Table 10-316. EMIF_EXT_PHY_CONTROL_26_SHADOW

Address Offset	0x0000 02CC		
Physical Address	0x4C00 02CC	**Instance**	EMIF
Description	Data macro 0, FIFO write enable (read DQS gate) DLL Slave Init Ratio Shadow Register. Its value is loaded in register EMIF_EXT_PHY_CONTROL_26 on any frequency change.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								REG_PHY_GATELVL_INIT_RATIO1								RESERVED								REG_PHY_GATELVL_INIT_RATIO0							

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:16	REG_PHY_GATELVL_INIT_RATIO1	The user programmable initialization ratio value used by the gate training finite state machine (hardware leveling) for the FIFO write enable slave DLL in PHY data macro 0, chip select 1. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. **NOTE: Chip select 1 is not supported on this device.**	RW	0x150
15:11	RESERVED		R	0x0
10:0	REG_PHY_GATELVL_INIT_RATIO0	The user programmable initialization ratio value used by the gate training finite state machine (hardware leveling) for the FIFO write enable slave DLL in PHY data macro 0, chip select 0. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line.	RW	0x150

Table 10-317. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_26_SHADOW

EMIF Controller

- Global Initialization: [0]
- EMIF Register Summary: [1]

Table 10-318. EMIF_EXT_PHY_CONTROL_27

Address Offset	0x0000 02D0		
Physical Address	0x4C00 02D0	**Instance**	EMIF
Description	Data macro 1, FIFO write enable (read DQS gate) DLL Slave Init Ratio Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								REG_PHY_GATELVL_INIT_RATIO3								RESERVED								REG_PHY_GATELVL_INIT_RATIO2							

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:16	REG_PHY_GATELVL_INIT_RATIO3	The user programmable initialization ratio value used by the gate training finite state machine (hardware leveling) for the FIFO write enable slave DLL in PHY data macro 1, chip select 1. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. **NOTE: Chip select 1 is not supported on this device.**	RW	0x150
15:11	RESERVED		R	0x0
10:0	REG_PHY_GATELVL_INIT_RATIO2	The user programmable initialization ratio value used by the gate training finite state machine (hardware leveling) for the FIFO write enable slave DLL in PHY data macro 1, chip select 0. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line.	RW	0x150

Table 10-319. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_27

EMIF Controller

- EMIF Register Summary: [0]
- EMIF Register Description: [1]

Table 10-320. EMIF_EXT_PHY_CONTROL_27_SHADOW

Address Offset	0x0000 02D4		
Physical Address	0x4C00 02D4	**Instance**	EMIF
Description	Data macro 1, FIFO write enable (read DQS gate) DLL Slave Init Ratio Shadow Register. Its value is loaded in register EMIF_EXT_PHY_CONTROL_27 on any frequency change.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								REG_PHY_GATELVL_INIT_RATIO3								RESERVED								REG_PHY_GATELVL_INIT_RATIO2							

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:16	REG_PHY_GATELVL_INIT_RATIO3	The user programmable initialization ratio value used by the gate training finite state machine (hardware leveling) for the FIFO write enable slave DLL in PHY data macro 1, chip select 1. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. **NOTE: Chip select 1 is not supported on this device.**	RW	0x150
15:11	RESERVED		R	0x0
10:0	REG_PHY_GATELVL_INIT_RATIO2	The user programmable initialization ratio value used by the gate training finite state machine (hardware leveling) for the FIFO write enable slave DLL in PHY data macro 1, chip select 0. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line.	RW	0x150

Table 10-321. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_27_SHADOW

EMIF Controller

- EMIF Register Summary: [0]

Table 10-322. EMIF_EXT_PHY_CONTROL_28

Address Offset	0x0000 02D8		
Physical Address	0x4C00 02D8	**Instance**	EMIF
Description	Data macro 2, FIFO write enable (read DQS gate) DLL Slave Init Ratio Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								REG_PHY_GATELVL_INIT_RATIO5								RESERVED								REG_PHY_GATELVL_INIT_RATIO4							

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:16	REG_PHY_GATELVL_INIT_RATIO5	The user programmable initialization ratio value used by the gate training finite state machine (hardware leveling) for the FIFO write enable slave DLL in PHY data macro 2, chip select 1. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. **NOTE: Chip select 1 is not supported on this device.**	RW	0x150
15:11	RESERVED		R	0x0
10:0	REG_PHY_GATELVL_INIT_RATIO4	The user programmable initialization ratio value used by the gate training finite state machine (hardware leveling) for the FIFO write enable slave DLL in PHY data macro 2, chip select 0. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line.	RW	0x150

Table 10-323. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_28

EMIF Controller

- EMIF Register Summary: [0]
- EMIF Register Description: [1]

Table 10-324. EMIF_EXT_PHY_CONTROL_28_SHADOW

Address Offset	0x0000 02DC		
Physical Address	0x4C00 02DC	**Instance**	EMIF
Description	Data macro 2, FIFO write enable (read DQS gate) DLL Slave Init Ratio Shadow Register. Its value is loaded in register EMIF_EXT_PHY_CONTROL_28 on any frequency change.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								REG_PHY_GATELVL_INIT_RATIO5								RESERVED								REG_PHY_GATELVL_INIT_RATIO4							

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:16	REG_PHY_GATELVL_INIT_RATIO5	The user programmable initialization ratio value used by the gate training finite state machine (hardware leveling) for the FIFO write enable slave DLL in PHY data macro 2, chip select 1. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. **NOTE: Chip select 1 is not supported on this device.**	RW	0x150
15:11	RESERVED		R	0x0
10:0	REG_PHY_GATELVL_INIT_RATIO4	The user programmable initialization ratio value used by the gate training finite state machine (hardware leveling) for the FIFO write enable slave DLL in PHY data macro 2, chip select 0. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line.	RW	0x150

Table 10-325. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_28_SHADOW

EMIF Controller

- EMIF Register Summary: [0]

Table 10-326. EMIF_EXT_PHY_CONTROL_29

Address Offset	0x0000 02E0		
Physical Address	0x4C00 02E0	**Instance**	EMIF
Description	Data macro 3, FIFO write enable (read DQS gate) DLL Slave Init Ratio Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								REG_PHY_GATELVL_INIT_RATIO7								RESERVED								REG_PHY_GATELVL_INIT_RATIO6							

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:16	REG_PHY_GATELVL_INIT_RATIO7	The user programmable initialization ratio value used by the gate training finite state machine (hardware leveling) for the FIFO write enable slave DLL in PHY data macro 3, chip select 1. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. **NOTE: Chip select 1 is not supported on this device.**	RW	0x150
15:11	RESERVED		R	0x0
10:0	REG_PHY_GATELVL_INIT_RATIO6	The user programmable initialization ratio value used by the gate training finite state machine (hardware leveling) for the FIFO write enable slave DLL in PHY data macro 3, chip select 0. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line.	RW	0x150

Table 10-327. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_29

EMIF Controller

- EMIF Register Summary: [0]
- EMIF Register Description: [1]

Table 10-328. EMIF_EXT_PHY_CONTROL_29_SHADOW

Address Offset	0x0000 02E4		
Physical Address	0x4C00 02E4	**Instance**	EMIF
Description	Data macro 3, FIFO write enable (read DQS gate) DLL Slave Init Ratio Shadow Register. Its value is loaded in register EMIF_EXT_PHY_CONTROL_29 on any frequency change.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								REG_PHY_GATELVL_INIT_RATIO7								RESERVED								REG_PHY_GATELVL_INIT_RATIO6							

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:16	REG_PHY_GATELVL_INIT_RATIO7	The user programmable initialization ratio value used by the gate training finite state machine (hardware leveling) for the FIFO write enable slave DLL in PHY data macro 3, chip select 1. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. **NOTE: Chip select 1 is not supported on this device.**	RW	0x150
15:11	RESERVED		R	0x0
10:0	REG_PHY_GATELVL_INIT_RATIO6	The user programmable initialization ratio value used by the gate training finite state machine (hardware leveling) for the FIFO write enable slave DLL in PHY data macro 3, chip select 0. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line.	RW	0x150

Table 10-329. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_29_SHADOW

EMIF Controller

- EMIF Register Summary: [0]

Table 10-330. EMIF_EXT_PHY_CONTROL_30

Address Offset	0x0000 02E8		
Physical Address	0x4C00 02E8	**Instance**	EMIF
Description	ECC Data macro, FIFO write enable (read DQS gate) DLL Slave Init Ratio Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								REG_PHY_GATELVL_INIT_RATIO9								RESERVED								REG_PHY_GATELVL_INIT_RATIO8							

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:16	REG_PHY_GATELVL_INIT_RATIO9	The user programmable initialization ratio value used by the gate training finite state machine (hardware leveling) for the FIFO write enable slave DLL in PHY ECC data macro, chip select 1. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. **NOTE: Chip select 1 is not supported on this device.**	RW	0x150
15:11	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
10:0	REG_PHY_GATELVL_INIT_RATIO8	The user programmable initialization ratio value used by the gate training finite state machine (hardware leveling) for the FIFO write enable slave DLL in PHY ECC data macro, chip select 0. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line.	RW	0x150

Table 10-331. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_30

EMIF Controller

- Software Leveling: [0][1][2]
- EMIF Register Summary: [3]
- EMIF Register Description: [4]

Table 10-332. EMIF_EXT_PHY_CONTROL_30_SHADOW

Address Offset	0x0000 02EC		
Physical Address	0x4C00 02EC	**Instance**	EMIF
Description	ECC Data macro, FIFO write enable (read DQS gate) DLL Slave Init Ratio Shadow Register. Its value is loaded in register EMIF_EXT_PHY_CONTROL_30 on any frequency change.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								REG_PHY_GATELVL_INIT_RATIO9								RESERVED								REG_PHY_GATELVL_INIT_RATIO8							

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:16	REG_PHY_GATELVL_INIT_RATIO9	The user programmable initialization ratio value used by the gate training finite state machine (hardware leveling) for the FIFO write enable slave DLL in PHY ECC data macro, chip select 1. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. **NOTE: Chip select 1 is not supported on this device.**	RW	0x150
15:11	RESERVED		R	0x0
10:0	REG_PHY_GATELVL_INIT_RATIO8	The user programmable initialization ratio value used by the gate training finite state machine (hardware leveling) for the FIFO write enable slave DLL in PHY ECC data macro, chip select 0. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line.	RW	0x150

Table 10-333. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_30_SHADOW

EMIF Controller

- EMIF Register Summary: [0]

Table 10-334. EMIF_EXT_PHY_CONTROL_31

Address Offset	0x0000 02F0		
Physical Address	0x4C00 02F0	**Instance**	EMIF
Description	Data macro 0, write DQS DLL Slave Init Ratio Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								REG_PHY_WRLVL_INIT_RATIO1								RESERVED								REG_PHY_WRLVL_INIT_RATIO0							

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:16	REG_PHY_WRLVL_INIT_RATIO 1	The user programmable initialization ratio value used by the write leveling finite state machine (hardware leveling) for the write DQS slave DLL in PHY data macro 0, chip select 1. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. **NOTE: Chip select 1 is not supported on this device.**	RW	0x0
15:10	RESERVED		R	0x0
9:0	REG_PHY_WRLVL_INIT_RATIO 0	The user programmable initialization ratio value used by the write leveling finite state machine (hardware leveling) for the write DQS slave DLL in PHY data macro 0, chip select 0. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line.	RW	0x0

Table 10-335. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_31

EMIF Controller

- EMIF Register Summary: [0]
- EMIF Register Description: [1]

Table 10-336. EMIF_EXT_PHY_CONTROL_31_SHADOW

Address Offset	0x0000 02F4		
Physical Address	0x4C00 02F4	**Instance**	EMIF
Description	Data macro 0, write DQS DLL Slave Init Ratio Shadow Register. Its value is loaded in register EMIF_EXT_PHY_CONTROL_31 on any frequency change.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								REG_PHY_WRLVL_INIT_RATIO1								RESERVED								REG_PHY_WRLVL_INIT_RATIO0							

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:16	REG_PHY_WRLVL_INIT_RATIO 1	The user programmable initialization ratio value used by the write leveling finite state machine (hardware leveling) for the write DQS slave DLL in PHY data macro 0, chip select 1. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. **NOTE: Chip select 1 is not supported on this device.**	RW	0x0
15:10	RESERVED		R	0x0
9:0	REG_PHY_WRLVL_INIT_RATIO 0	The user programmable initialization ratio value used by the write leveling finite state machine (hardware leveling) for the write DQS slave DLL in PHY data macro 0, chip select 0. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line.	RW	0x0

Table 10-337. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_31_SHADOW

EMIF Controller

- EMIF Register Summary: [0]

Table 10-338. EMIF_EXT_PHY_CONTROL_32

Address Offset	0x0000 02F8		
Physical Address	0x4C00 02F8	**Instance**	EMIF
Description	Data macro 1, write DQS DLL Slave Init Ratio Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								REG_PHY_WRLVL_INIT_RATIO3								RESERVED								REG_PHY_WRLVL_INIT_RATIO2							

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:16	REG_PHY_WRLVL_INIT_RATIO3	The user programmable initialization ratio value used by the write leveling finite state machine (hardware leveling) for the write DQS slave DLL in PHY data macro 1, chip select 1. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. **NOTE: Chip select 1 is not supported on this device.**	RW	0x0
15:10	RESERVED		R	0x0
9:0	REG_PHY_WRLVL_INIT_RATIO2	The user programmable initialization ratio value used by the write leveling finite state machine (hardware leveling) for the write DQS slave DLL in PHY data macro 1, chip select 0. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line.	RW	0x0

Table 10-339. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_32

EMIF Controller

- EMIF Register Summary: [0]
- EMIF Register Description: [1]

Table 10-340. EMIF_EXT_PHY_CONTROL_32_SHADOW

Address Offset	0x0000 02FC		
Physical Address	0x4C00 02FC	**Instance**	EMIF
Description	Data macro 1, write DQS DLL Slave Init Ratio Shadow Register. Its value is loaded in register EMIF_EXT_PHY_CONTROL_32 on any frequency change.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								REG_PHY_WRLVL_INIT_RATIO3								RESERVED								REG_PHY_WRLVL_INIT_RATIO2							

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:16	REG_PHY_WRLVL_INIT_RATIO3	The user programmable initialization ratio value used by the write leveling finite state machine (hardware leveling) for the write DQS slave DLL in PHY data macro 1, chip select 1. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. **NOTE: Chip select 1 is not supported on this device.**	RW	0x0
15:10	RESERVED		R	0x0
9:0	REG_PHY_WRLVL_INIT_RATIO2	The user programmable initialization ratio value used by the write leveling finite state machine (hardware leveling) for the write DQS slave DLL in PHY data macro 1, chip select 0. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line.	RW	0x0

Table 10-341. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_32_SHADOW

EMIF Controller

- EMIF Register Summary: [0]

Table 10-342. EMIF_EXT_PHY_CONTROL_33

Address Offset	0x0000 0300		
Physical Address	0x4C00 0300	**Instance**	EMIF
Description	Data macro 2, write DQS DLL Slave Init Ratio Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								REG_PHY_WRLVL_INIT_RATIO5								RESERVED								REG_PHY_WRLVL_INIT_RATIO4							

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:16	REG_PHY_WRLVL_INIT_RATIO5	The user programmable initialization ratio value used by the write leveling finite state machine (hardware leveling) for the write DQS slave DLL in PHY data macro 2, chip select 1. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. **NOTE: Chip select 1 is not supported on this device.**	RW	0x0
15:10	RESERVED		R	0x0
9:0	REG_PHY_WRLVL_INIT_RATIO4	The user programmable initialization ratio value used by the write leveling finite state machine (hardware leveling) for the write DQS slave DLL in PHY data macro 2, chip select 0. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line.	RW	0x0

Table 10-343. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_33

EMIF Controller

- EMIF Register Summary: [0]
- EMIF Register Description: [1]

Table 10-344. EMIF_EXT_PHY_CONTROL_33_SHADOW

Address Offset	0x0000 0304		
Physical Address	0x4C00 0304	**Instance**	EMIF
Description	Data macro 2, write DQS DLL Slave Init Ratio Shadow Register. Its value is loaded in register EMIF_EXT_PHY_CONTROL_33 on any frequency change.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								REG_PHY_WRLVL_INIT_RATIO5								RESERVED								REG_PHY_WRLVL_INIT_RATIO4							

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:16	REG_PHY_WRLVL_INIT_RATIO5	The user programmable initialization ratio value used by the write leveling finite state machine (hardware leveling) for the write DQS slave DLL in PHY data macro 2, chip select 1. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. **NOTE: Chip select 1 is not supported on this device.**	RW	0x0
15:10	RESERVED		R	0x0
9:0	REG_PHY_WRLVL_INIT_RATIO4	The user programmable initialization ratio value used by the write leveling finite state machine (hardware leveling) for the write DQS slave DLL in PHY data macro 2, chip select 0. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line.	RW	0x0

Table 10-345. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_33_SHADOW

EMIF Controller

- EMIF Register Summary: [0]

Table 10-346. EMIF_EXT_PHY_CONTROL_34

Address Offset	0x0000 0308		
Physical Address	0x4C00 0308	**Instance**	EMIF
Description	Data macro 3, write DQS DLL Slave Init Ratio Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								REG_PHY_WRLVL_INIT_RATIO7								RESERVED								REG_PHY_WRLVL_INIT_RATIO6							

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:16	REG_PHY_WRLVL_INIT_RATIO7	The user programmable initialization ratio value used by the write leveling finite state machine (hardware leveling) for the write DQS slave DLL in PHY data macro 3, chip select 1. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. **NOTE: Chip select 1 is not supported on this device.**	RW	0x0
15:10	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
9:0	REG_PHY_WRLVL_INIT_RATIO6	The user programmable initialization ratio value used by the write leveling finite state machine (hardware leveling) for the write DQS slave DLL in PHY data macro 3, chip select 0. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line.	RW	0x0

Table 10-347. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_34

EMIF Controller

- EMIF Register Summary: [0]
- EMIF Register Description: [1]

Table 10-348. EMIF_EXT_PHY_CONTROL_34_SHADOW

Address Offset	0x0000 030C		
Physical Address	0x4C00 030C	**Instance**	EMIF
Description	Data macro 3, write DQS DLL Slave Init Ratio Shadow Register. Its value is loaded in register EMIF_EXT_PHY_CONTROL_34 on any frequency change.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								REG_PHY_WRLVL_INIT_RATIO7								RESERVED								REG_PHY_WRLVL_INIT_RATIO6							

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:16	REG_PHY_WRLVL_INIT_RATIO7	The user programmable initialization ratio value used by the write leveling finite state machine (hardware leveling) for the write DQS slave DLL in PHY data macro 3, chip select 1. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. **NOTE: Chip select 1 is not supported on this device.**	RW	0x0
15:10	RESERVED		R	0x0
9:0	REG_PHY_WRLVL_INIT_RATIO6	The user programmable initialization ratio value used by the write leveling finite state machine (hardware leveling) for the write DQS slave DLL in PHY data macro 3, chip select 0. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line.	RW	0x0

Table 10-349. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_34_SHADOW

EMIF Controller

- EMIF Register Summary: [0]

Table 10-350. EMIF_EXT_PHY_CONTROL_35

Address Offset	0x0000 0310		
Physical Address	0x4C00 0310	**Instance**	EMIF
Description	ECC Data macro, write DQS DLL Slave Init Ratio Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								REG_PHY_WRLVL_INIT_RATIO9								RESERVED								REG_PHY_WRLVL_INIT_RATIO8							

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:16	REG_PHY_WRLVL_INIT_RATIO9	The user programmable initialization ratio value used by the write leveling finite state machine (hardware leveling) for the write DQS slave DLL in PHY ECC data macro, chip select 1. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. **NOTE: Chip select 1 is not supported on this device.**	RW	0x0
15:10	RESERVED		R	0x0
9:0	REG_PHY_WRLVL_INIT_RATIO8	The user programmable initialization ratio value used by the write leveling finite state machine (hardware leveling) for the write DQS slave DLL in PHY ECC data macro, chip select 0. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line.	RW	0x0

Table 10-351. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_35

EMIF Controller

- Global Initialization: [0][1]
- EMIF Register Summary: [2]
- EMIF Register Description: [3]

Table 10-352. EMIF_EXT_PHY_CONTROL_35_SHADOW

Address Offset	0x0000 0314		
Physical Address	0x4C00 0314	**Instance**	EMIF
Description	ECC Data macro, write DQS DLL Slave Init Ratio Shadow Register. Its value is loaded in register EMIF_EXT_PHY_CONTROL_35 on any frequency change.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								REG_PHY_WRLVL_INIT_RATIO9								RESERVED								REG_PHY_WRLVL_INIT_RATIO8							

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:16	REG_PHY_WRLVL_INIT_RATIO9	The user programmable initialization ratio value used by the write leveling finite state machine (hardware leveling) for the write DQS slave DLL in PHY ECC data macro, chip select 1. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line. **NOTE: Chip select 1 is not supported on this device.**	RW	0x0
15:10	RESERVED		R	0x0
9:0	REG_PHY_WRLVL_INIT_RATIO8	The user programmable initialization ratio value used by the write leveling finite state machine (hardware leveling) for the write DQS slave DLL in PHY ECC data macro, chip select 0. This is the fraction of a clock cycle in units of 256ths. In other words, the full-cycle tap value from the master DLL will be scaled by this number over 256 to get the delay value for the slave delay line.	RW	0x0

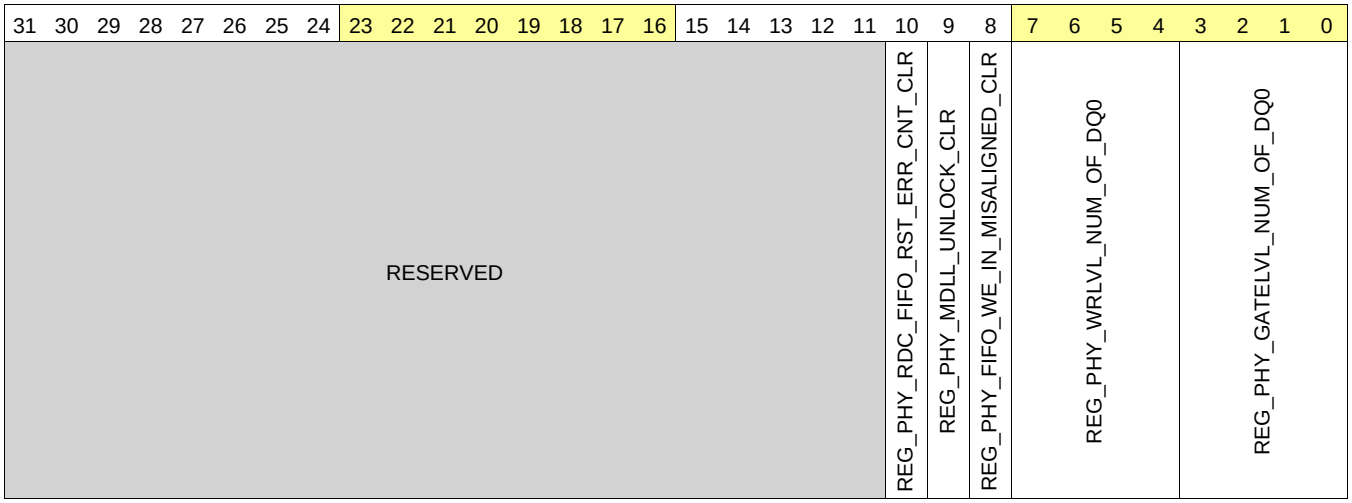
Table 10-353. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_35_SHADOW

EMIF Controller

- Global Initialization: [0]
- EMIF Register Summary: [1]

Table 10-354. EMIF_EXT_PHY_CONTROL_36

Address Offset	0x0000 0318		
Physical Address	0x4C00 0318	**Instance**	EMIF
Description			
Type	RW		



Bits	Field Name	Description	Type	Reset
31:11	RESERVED		R	0x0
10	REG_PHY_RDC_FIFO_RST_ERR_CNT_CLR	Clear/reset the phy_reg_rdc_fifo_rst_err_cnt, phy_reg_rdfifo_wrptr and phy_reg_rdfifo_rdptr status flags. A value of 0x1 clears the flag. A value of 0x0 has no effect.	RW	0x0
9	REG_PHY_MDLL_UNLOCK_CLR	Clears the phy_reg_status_mdll_unlock_sticky flag. A value of 0x1 clears the flag. A value of 0x0 has no effect.	RW	0x0
8	REG_PHY_FIFO_WE_IN_MISALIGNED_CLR	Clears the phy_reg_fifo_we_in_misaligned_sticky status flag. A value of 0x1 clears the flag. A value of 0x0 has no effect.	RW	0x0
7:4	REG_PHY_WRLVL_NUM_OF_DQ0	Determines the number of samples for *dq0_in* for each ratio increment by the write leveling finite state machine (hardware leveling). The minimum value supported is 3; however, the default setting of 7 is recommended. **NOTE:** In this case *dq0_in* represents all 8 DQ bits OR-ed together.	RW	0x7
3:0	REG_PHY_GATELVL_NUM_OF_DQ0	Determines the number of samples for *dq0_in* for each ratio increment by the gate training finite state machine (hardware leveling). The minimum value supported is 3; however, the default setting of 7 is recommended. **NOTE:** In this case *dq0_in* represents the corresponding DQS signal.	RW	0x7

Table 10-355. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_36

EMIF Controller

- Global Initialization: [0][1][2][3][4]
- EMIF Register Summary: [5]

Table 10-356. EMIF_EXT_PHY_CONTROL_36_SHADOW

Address Offset	0x0000 031C		
Physical Address	0x4C00 031C	**Instance**	EMIF
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																					REG_PHY_RDC_FIFO_RST_ERR_CNT_CLR	REG_PHY_MDLL_UNLOCK_CLR	REG_PHY_FIFO_WE_IN_MISALIGNED_CLR	REG_PHY_WRLVL_NUM_OF_DQ0				REG_PHY_GATELVL_NUM_OF_DQ0			

Bits	Field Name	Description	Type	Reset
31:11	RESERVED		R	0x0
10	REG_PHY_RDC_FIFO_RST_ERR_CNT_CLR	Clear/reset the phy_reg_rdc_fifo_rst_err_cnt, phy_reg_rdfifo_wrptr and phy_reg_rdfifo_rdptr status flags. A value of 0x1 clears the flag. A value of 0x0 has no effect.	RW	0x0
9	REG_PHY_MDLL_UNLOCK_CLR	Clears the phy_reg_status_mdll_unlock_sticky flag. A value of 0x1 clears the flag. A value of 0x0 has no effect.	RW	0x0
8	REG_PHY_FIFO_WE_IN_MISALIGNED_CLR	Clears the phy_reg_fifo_we_in_misaligned_sticky status flag. A value of 0x1 clears the flag. A value of 0x0 has no effect.	RW	0x0
7:4	REG_PHY_WRLVL_NUM_OF_DQ0	Determines the number of samples for *dq0_in* for each ratio increment by the write leveling finite state machine (hardware leveling). The minimum value supported is 3; however, the default setting of 7 is recommended. **NOTE:** In this case *dq0_in* represents all 8 DQ bits OR-ed together.	RW	0x7
3:0	REG_PHY_GATELVL_NUM_OF_DQ0	Determines the number of samples for *dq0_in* for each ratio increment by the gate training finite state machine (hardware leveling). The minimum value supported is 3; however, the default setting of 7 is recommended. **NOTE:** In this case *dq0_in* represents the corresponding DQS signal.	RW	0x7

Table 10-357. Register Call Summary for Register EMIF_EXT_PHY_CONTROL_36_SHADOW

EMIF Controller

- Global Initialization: [0]
- EMIF Register Summary: [1]

10.3 General-Purpose Memory Controller

This section describes the features and functions of the device GPMC controller.

10.3.1 GPMC Overview

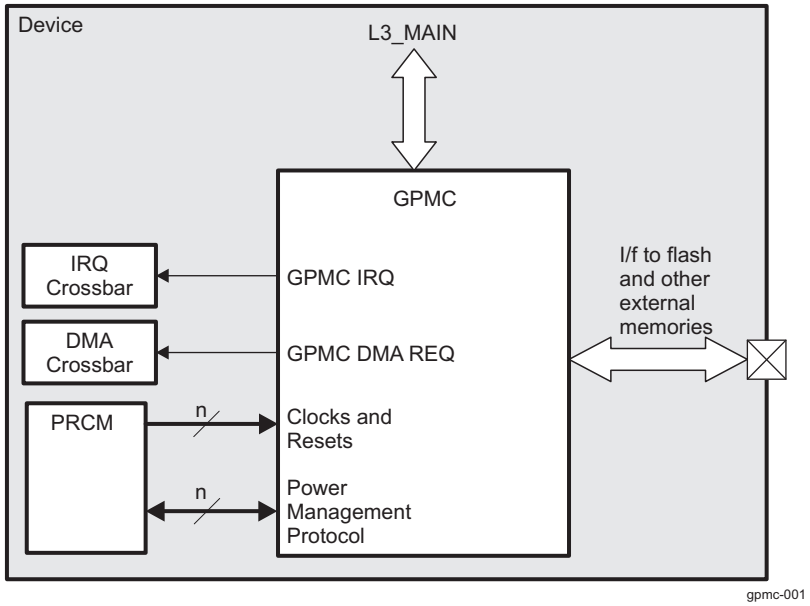
The general-purpose memory controller (GPMC) is a unified memory controller dedicated for interfacing with external memory devices like:

- Asynchronous SRAM-like memories and application-specific integrated circuit (ASIC) devices
- Asynchronous, synchronous, and page mode (available only in nonmultiplexed mode) burst NOR flash devices
- NAND flash
- Pseudo-SRAM devices

Figure 10-11 shows the GPMC module overview.

The GPMC features are introduced in Section 10.1.2, *GPMC Overview* of Section 10.1, *Memory Subsystem Overview*.

Figure 10-11. GPMC Overview



10.3.2 GPMC Environment

This section describes the GPMC application fields from an environment point of view (external connections). It describes GPMC connectivity options, and gives three possible interfaces.

10.3.2.1 GPMC Modes

This section shows three GPMC external connections options:

- Figure 10-12 shows a connection between the GPMC and a 16-bit synchronous address/data-multiplexed (or AAD-multiplexed, but this protocol uses fewer address pins) external memory device.
- Figure 10-13 shows a connection between the GPMC and a 16-bit synchronous nonmultiplexed external memory device.
- Figure 10-14 shows a connection between the GPMC and an 8-bit synchronous non-multiplexed external memory device.
- Figure 10-15 shows a connection between the GPMC and a 8-bit NAND device.

Figure 10-12. GPMC to 16-Bit Address/Data-Multiplexed Memory

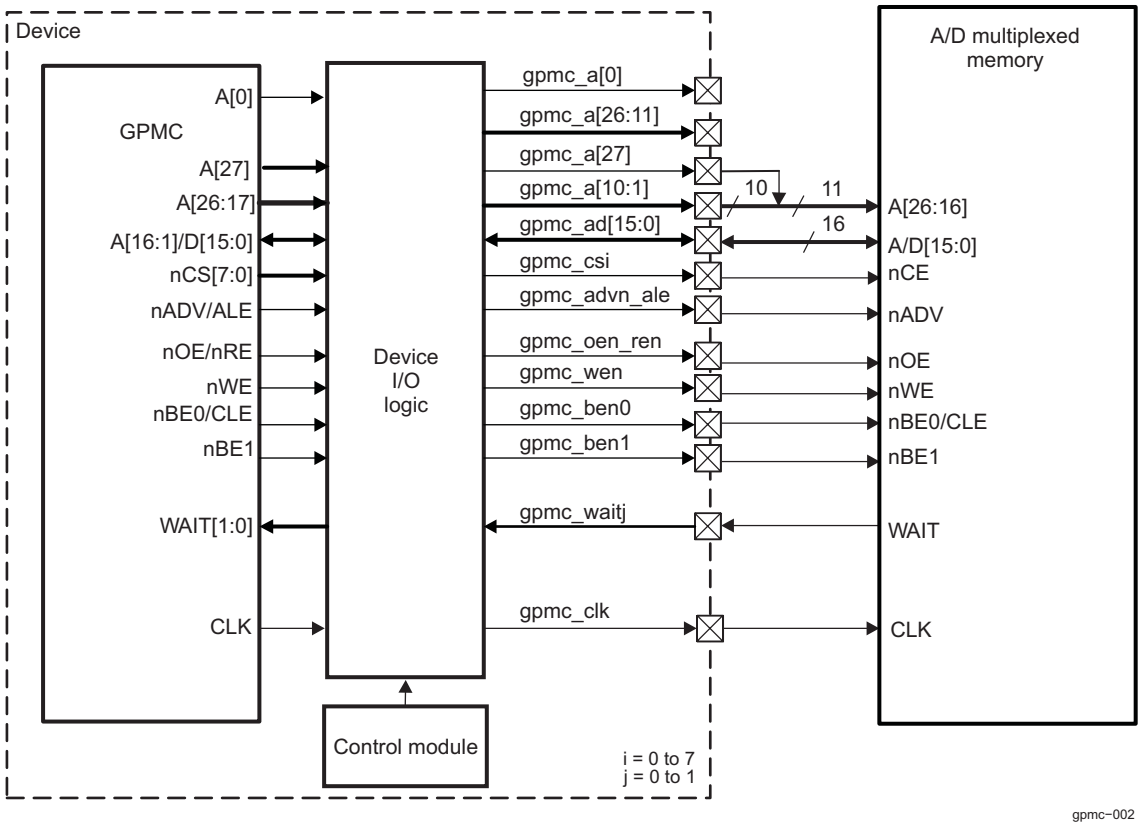
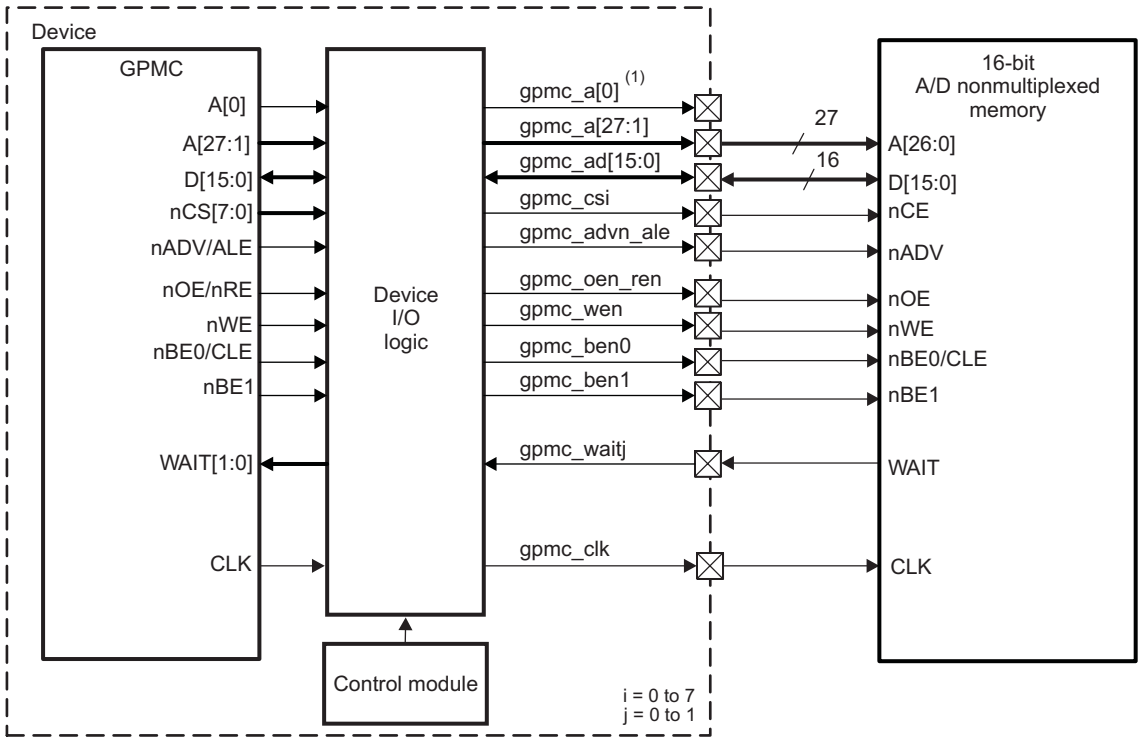


Figure 10-13. GPMC to 16-Bit Nonmultiplexed Memory



Note (1): the gpmc_a[0] pin is not used with 16-bit memory devices

Figure 10-14. GPMC to 8-Bit Nonmultiplexed Memory

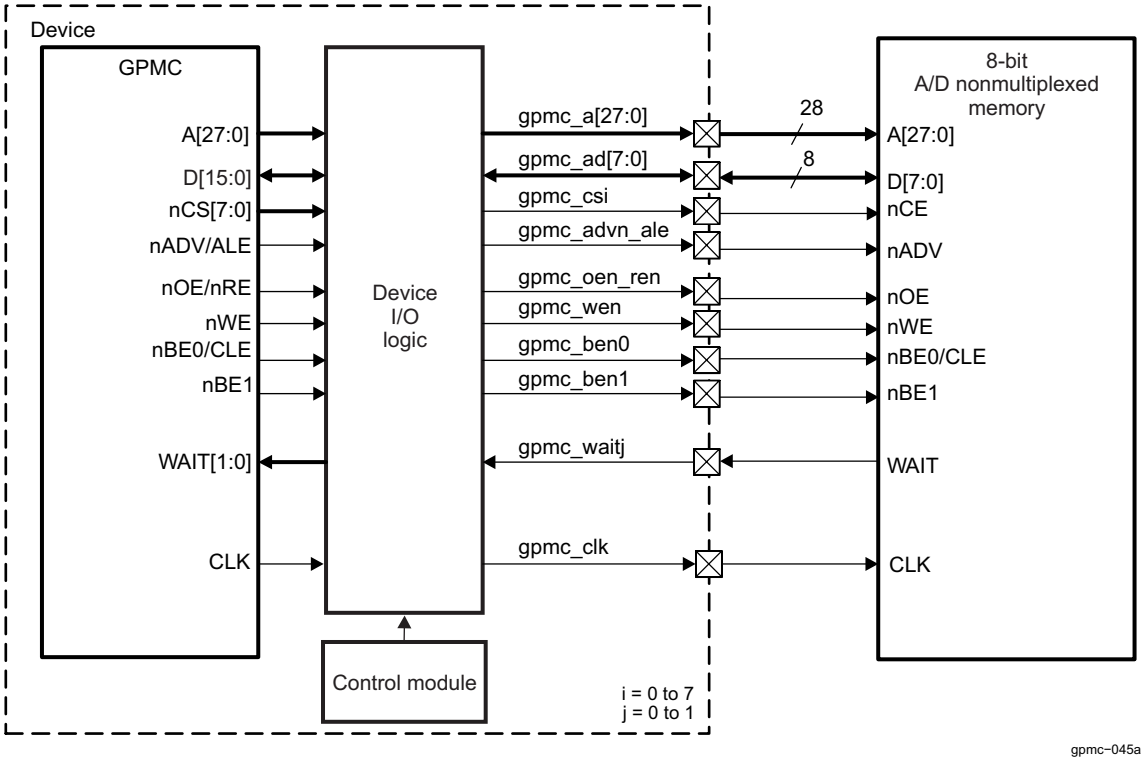
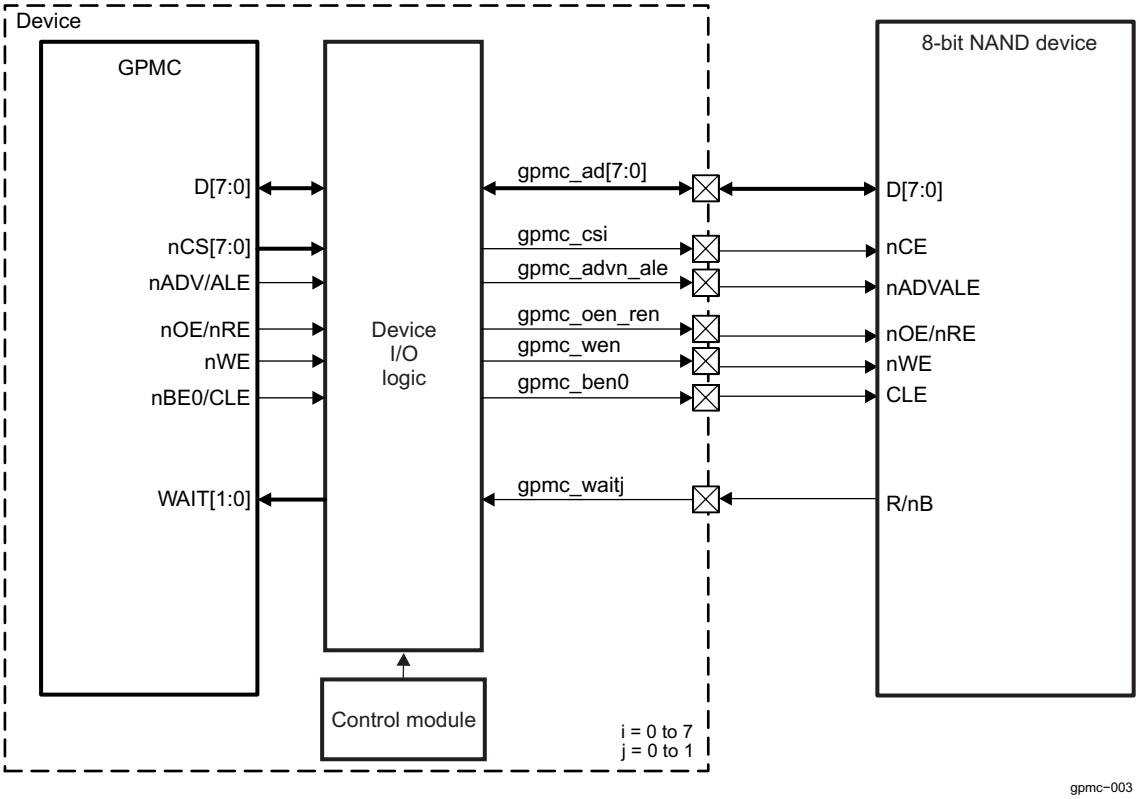


Figure 10-15. GPMC to 8-Bit NAND Device



10.3.2.2 GPMC Signals

Table 10-358 lists the GPMC subsystem input/output (I/O) pins.

Table 10-358. GPMC I/O Description

Pin Name	Device Signal	I/O[1]	Description
A[27:0]	gpmc_a[27:0]	O	28-bit output address bus
A[16:1]/D[15:0]	gpmc_ad[15:0]	I/O	Multiplexed address/data
nCS[7:0]	gpmc_cs[7:0]	O	Chip-selects (active low)
CLK	gpmc_clk	O	Clock generated for the external memory or device
nADV/ALE	gpmc_advn_ale	O	Address valid (active low). Also used as address latch enable (active high) for NAND protocol memories.
nOE/nRE	gpmc_oen_ren	O	Output enable (active low). Also used as read enable (active low) for NAND protocol memories.
nWE	gpmc_wen	O	Write enable (active low)
nBE0/CLE	gpmc_ben0	O	Lower-byte enable (active low). Also used as command latch enable for NAND protocol memories.
nBE1	gpmc_ben1	O	Upper-byte enable (active low).
WAIT[1:0]	gpmc_wait[1:0]	I	External wait signal for NOR and NAND protocol memories. The wait signals can be mapped on any of the chip-select.

(1) I = Input; O = Output

NOTE: For the gpmc_clk signal to work properly, the INPUTENABLE bit of the appropriate CTRL_CORE_PAD_x register should be set to 0x1 because of retiming purposes.

Table 10-359 shows the use of address and data GPMC controller pins based on the type of external device.

Table 10-359. GPMC Pin Multiplexing Options

GPMC Pin	Multiplexed Address Data 16-Bit Device	Nonmultiplexed Address Data 16-Bit Device (complete 28-bit address range)	Nonmultiplexed Address Data 8-Bit Device (complete 28-bit address range)	16-Bit NAND Device	8-Bit NAND Device
gpmc_a[27]	A27	A27	A27	Not used	Not used
gpmc_a[26]	Not used	A26	A26	Not used	Not used
gpmc_a[25]	Not used	A25	A25	Not used	Not used
gpmc_a[24]	Not used	A24	A24	Not used	Not used
gpmc_a[23]	Not used	A23	A23	Not used	Not used
gpmc_a[22]	Not used	A22	A22	Not used	Not used
gpmc_a[21]	Not used	A21	A21	Not used	Not used
gpmc_a[20]	Not used	A20	A20	Not used	Not used
gpmc_a[19]	Not used	A19	A19	Not used	Not used
gpmc_a[18]	Not used	A18	A18	Not used	Not used
gpmc_a[17]	Not used	A17	A17	Not used	Not used
gpmc_a[16]	Not used	A16	A16	Not used	Not used
gpmc_a[15]	Not used	A15	A15	Not used	Not used
gpmc_a[14]	Not used	A14	A14	Not used	Not used
gpmc_a[13]	Not used	A13	A13	Not used	Not used
gpmc_a[12]	Not used	A12	A12	Not used	Not used
gpmc_a[11]	Not used	A11	A11	Not used	Not used
gpmc_a[10]	A26	A10	A10	Not used	Not used

Table 10-359. GPMC Pin Multiplexing Options (continued)

GPMC Pin	Multiplexed Address Data 16-Bit Device	Nonmultiplexed Address Data 16-Bit Device (complete 28-bit address range)	Nonmultiplexed Address Data 8-Bit Device (complete 28-bit address range)	16-Bit NAND Device	8-Bit NAND Device
gpmc_a[9]	A25	A9	A9	Not used	Not used
gpmc_a[8]	A24	A8	A8	Not used	Not used
gpmc_a[7]	A23	A7	A7	Not used	Not used
gpmc_a[6]	A22	A6	A6	Not used	Not used
gpmc_a[5]	A21	A5	A5	Not used	Not used
gpmc_a[4]	A20	A4	A4	Not used	Not used
gpmc_a[3]	A19	A3	A3	Not used	Not used
gpmc_a[2]	A18	A2	A2	Not used	Not used
gpmc_a[1]	A17	A1	A1	Not used	Not used
gpmc_a[0] (1)	A0 - Not used	Not used	A0	Not used	Not used
gpmc_ad[15]	A16/D15	D15	Not used	D15	Not used
gpmc_ad[14]	A15/D14	D14	Not used	D14	Not used
gpmc_ad[13]	A14/D13	D13	Not used	D13	Not used
gpmc_ad[12]	A13/D12	D12	Not used	D12	Not used
gpmc_ad[11]	A12/D11	D11	Not used	D11	Not used
gpmc_ad[10]	A11/D10	D10	Not used	D10	Not used
gpmc_ad[9]	A10/D9	D9	Not used	D9	Not used
gpmc_ad[8]	A9/D8	D8	Not used	D8	Not used
gpmc_ad[7]	A8/D7	D7	D7	D7	D7
gpmc_ad[6]	A7/D6	D6	D6	D6	D6
gpmc_ad[5]	A6/D5	D5	D5	D5	D5
gpmc_ad[4]	A5/D4	D4	D4	D4	D4
gpmc_ad[3]	A4/D3	D3	D3	D3	D3
gpmc_ad[2]	A3/D2	D2	D2	D2	D2
gpmc_ad[1]	A2/D1	D1	D1	D1	D1
gpmc_ad[0]	A1/D0	D0	D0	D0	D0

(1) Used to effectively address 8-bit (only) non-multiplexed memories

With all device types, the GPMC does not drive unnecessary address lines. They stay at their reset value of 0x00.

Address mapping supports address/data-multiplexed 16-bit-wide devices:

- The NOR flash memory controller still supports nonmultiplexed address and data memory devices.
- Multiplexing mode can be selected through the GPMC_CONFIG1_i[9:8] MUXADDDATA bit field (where i = 0 to 7).
- Asynchronous page mode is not supported for multiplexed address and data devices.

10.3.3 GPMC Integration

This section describes module integration in the device, including information about clocks, resets, and hardware requests.

- No master standby protocol
- Supports slave idle protocol with device PRCM
- Supports Auto Idle
- No wake-up request
- One direct memory access request mapped to the device DMA crossbar (DMA_CROSSBAR)

- One interrupt request mapped via the device interrupt crossbar (IRQ_CROSSBAR) to all device integrated interrupt controllers
- One clock for functional and interface domains

Figure 10-16 shows GPMC integration.

Figure 10-16. GPMC Integration

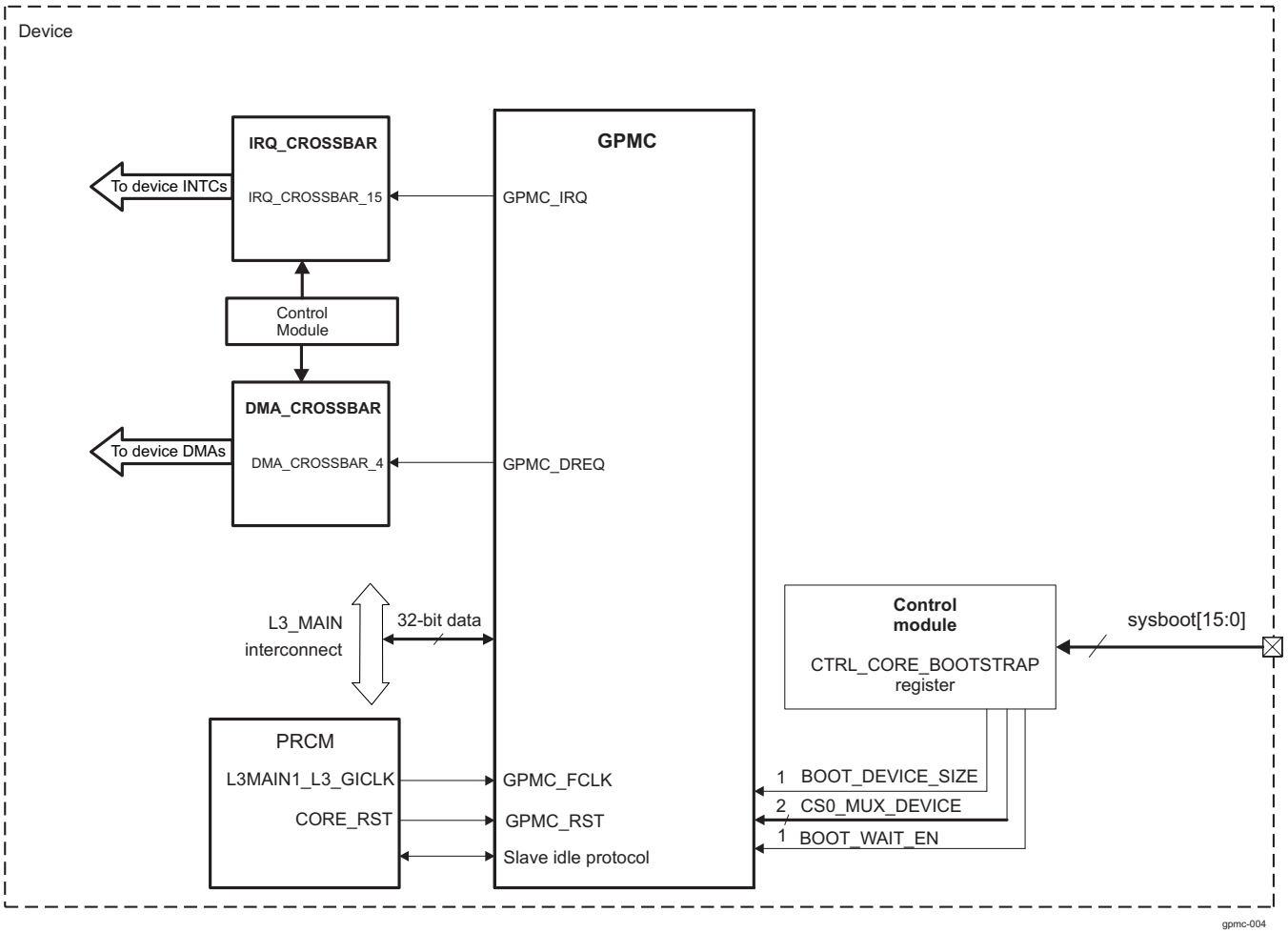


Table 10-360, Table 10-361 and Table 10-362 summarize the integration of the module in the device.

Table 10-360. GPMC Integration Attributes

Module Instance	Attributes		
	Power Domain	Wake-Up Capability	Interconnect
GPMC	PD_COREAON	No	L3_MAIN

NOTE:

- For the description of the operation performance point (OPP) configuration, see Section 3.1.1.1.2, *Module Level Clock Management*, in Chapter 3, *Power, Reset, and Clock Management*.
- For information about frequencies associated with each OPP, see the device data manual.

Table 10-361. GPMC Clocks and Resets

Clocks				
Module Instance	**Destination Signal Name**	**Source Signal Name**	**Source**	**Description**
GPMC	GPMC_FCLK	L3MAIN1_L3_GICLK	PRCM	Functional clock
Resets				
Module Instance	**Destination Signal Name**	**Source Signal Name**	**Source**	**Description**
GPMC	GPMC_RST	CORE_RET_RST	PRCM	GPMC reset

NOTE: For the clock description, see Section 10.3.4.2, *GPMC Clock Configuration*.

Table 10-362. GPMC Hardware Requests

Interrupt Requests				
Module Instance	**Source Signal Name**	**Destination IRQ_CROSSBAR Input**	**Default Mapping**	**Description**
GPMC	GPMC_IRQ	IRQ_CROSSBAR_15	DSP1_IRQ_46 DSP2_IRQ_46	GPMC interrupt
DMA Requests				
Module Instance	**Source Signal Name**	**Destination DMA_CROSSBAR Input**	**Default Mapping**	**Description**
GPMC	GPMC_DREQ	DMA_CROSSBAR_4	DMA_EDMA_DREQ_3	DMA request from GPMC prefetch engine.

NOTE: The "**Default Mapping**" column in Table 10-362 *GPMC Hardware Requests* shows the default mapping of module IRQ and DREQ source signals. These module IRQ and DREQ source signals can also be mapped to other lines of each device Interrupt or DMA controller through the IRQ_CROSSBAR and DMA_CROSSBAR modules, respectively.
For more information about the IRQ_CROSSBAR module, see Section 13.4.6.4, *IRQ_CROSSBAR Module Functional Description*, in Chapter 13, *Control Module*.
For more information about the DMA_CROSSBAR module, see Section 13.4.6.5, *DMA_CROSSBAR Module Functional Description*, in Chapter 13, *Control Module*.
For more information about the device interrupt controllers, see Chapter 12, *Interrupt Controllers*.
For more information about the device EDMA module, see Chapter 11, *Enhanced DMA*.

NOTE: For the description of the interrupt source, see Section 10.3.4.5, *GPMC Interrupt Requests*.

10.3.4 GPMC Functional Description

The GPMC basic programming model offers maximum flexibility to support various access protocols for each of the eight configurable chip-selects. Use optimal chip-select settings, based on the characteristics of the external device:

- Different protocols can be selected to support generic asynchronous or synchronous random-access devices (NOR flash, SRAM) or to support specific NAND devices.
- The address and data bus can be multiplexed on the same external bus.
- Read and write access can be independently defined as asynchronous or synchronous.
- System requests (byte, 16-bit word, burst) are performed through single or multiple accesses. External access profiles (single, multiple with optimized burst length, native- or emulated-wrap) are based on external device characteristics (supported protocol, bus width, data buffer size, native-wrap support).
- System burst read or write requests are synchronous-burst (multiple-read or multiple-write). When neither burst nor page mode is supported by external memory or ASIC devices, system burst read or write requests are translated to successive single synchronous or asynchronous accesses (single reads or single writes). 8-bit wide devices are supported only in single synchronous or single asynchronous read or write mode.
- To simulate a programmable internal-wait-state, an external wait pin can be monitored to dynamically control external access at the beginning (initial access time) of and during a burst access.

Each control signal is controlled independently for each chip-select. The internal functional clock of the GPMC (GPMC_FCLK) is used as a time reference to specify the following:

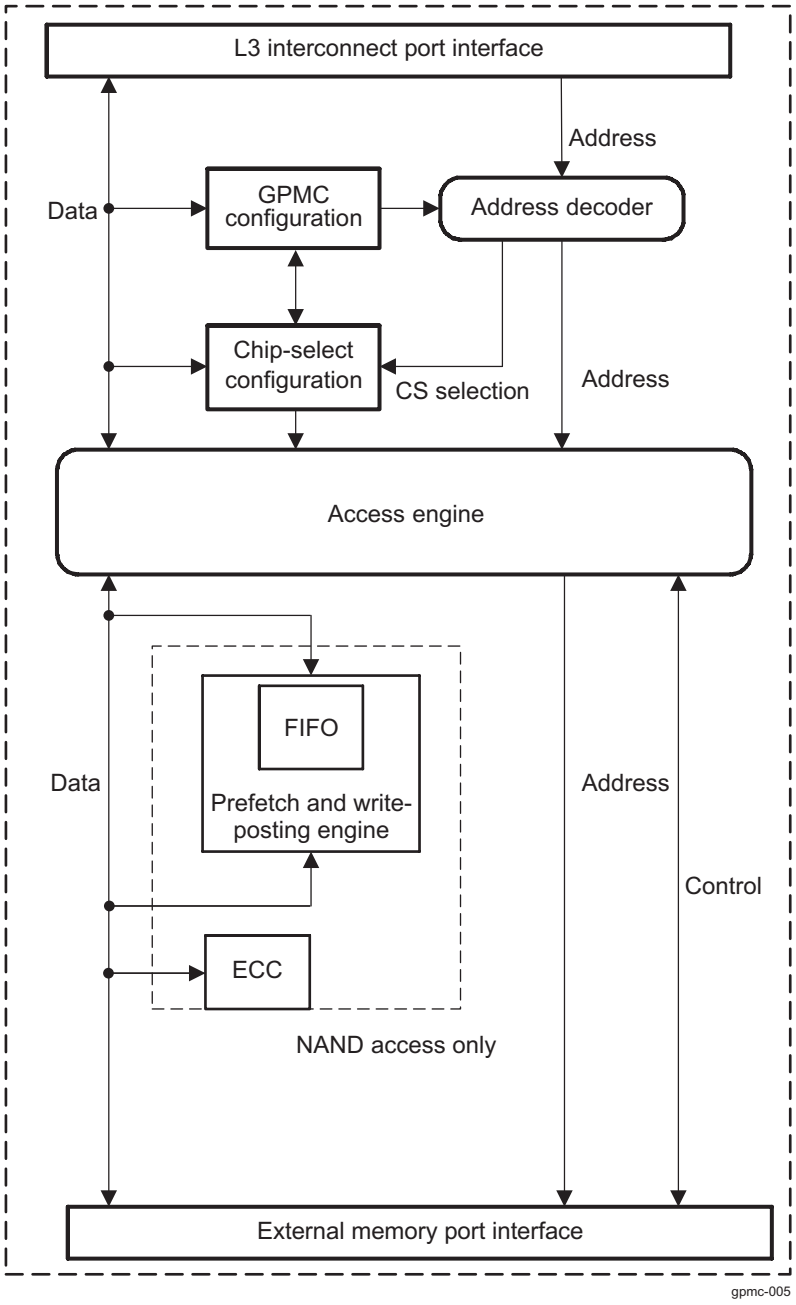
- Read- and write-access duration
- Most GPMC external interface control-signal assertion and deassertion times
- Data-capture time during read access
- External wait-pin monitoring time
- Duration of idle time between accesses, when required

10.3.4.1 GPMC Block Diagram

Figure 10-17 shows the GPMC functional block diagram. The GPMC consists of six blocks:

- L3 interconnect port interface
- Address decoder, GPMC configuration, and chip-select configuration register file
- Access engine
- Prefetch and write-posting engine
- Error correction code engine (ECC)
- External device/memory port interface

Figure 10-17. GPMC Block Diagram



The GPMC can access various external devices. The flexible programming model allows a wide range of attached device types and access schemes.

Based on the programmed configuration bit fields stored in the GPMC registers, the GPMC can generate the timing of all control signals depending on the attached device and access type.

Given the chip-select decoding and its associated configuration registers, the GPMC selects the appropriate control signal timing for the device type.

10.3.4.2 GPMC Clock Configuration

Table 10-363 describes the GPMC clocks.

Table 10-363. GPMC Clocks

Signal	I/O[1]	Description
GPMC_FCLK	I	Functional and interface clock
gpmc_clk	O	External clock provided to synchronous external memory devices

(1) I = Input; O = Output

The gpmc_clk is generated by the GPMC from the internal GPMC_FCLK clock. The source of the GPMC_FCLK is described in Table 10-361. The gpmc_clk is configured using the GPMC_CONFIG1_i[1:0] GPMCFCLKDIVIDER bit field (where i = 0 to 7), as shown in Table 10-364.

Table 10-364. gpmc_clk Configuration

Source Clock	GPMC_CONFIG1_i[1:0] GPMCFCLKDIVIDER	gpmc_clk Generated Clock Provided to External Memory Device
GPMC_FCLK	00	GPMC_FCLK
	01	GPMC_FCLK/2
	10	GPMC_FCLK/3
	11	GPMC_FCLK/4

10.3.4.3 GPMC Software Reset

The GPMC can be reset by software through the GPMC_SYSCONFIG[1] SOFTRESET bit. Setting the bit to 1 enables an active software reset that is functionally equivalent to a hardware reset. Hardware and software resets initialize all GPMC registers and the finite state-machine (FSM) immediately and unconditionally. The GPMC_SYSSTATUS[0] RESETDONE bit indicates that the software reset is complete when its value is 1. Software must ensure that the software reset completes before performing GPMC operations.

10.3.4.4 GPMC Power Management

GPMC power is supplied by the CORE power domain, and GPMC power management complies with system power-management guidelines.

Table 10-365 describes the power-management features available for the GPMC module.

NOTE:

- For information about source clock gating and sleep/wake-up transitions, see Section 3.1.1.1, *Clock Management*, in Chapter 3, *Power, Reset, and Clock Management*.
- For descriptions of the EnaWakeUp, IdleMode, ClockActivity, and StandbyMode features, see Section 3.6, *Clock Management Functional Description*, in Chapter 3, *Power, Reset, and Clock Management*.

Table 10-365. GPMC Local Power-Management Features

Feature	Registers	Description
Clock autogating	GPMC_SYSCONFIG[0] AUTOIDLE	This bit allows a local power optimization inside the module, by gating the GPMC_FCLK clock upon the internal activity.
Slave idle modes	GPMC_SYSCONFIG[4:3] SIDLEMODE	Force-idle, no-idle and smart-idle wake-up modes are available.
Clock activity	N/A	Feature not available
Master standby modes	N/A	Feature not available
Global wake-up enable	N/A	Feature not available
Wake-up sources enable	N/A	Feature not available

10.3.4.5 GPMC Interrupt Requests

The GPMC generates one interrupt request (see Figure 10-16).

Table 10-366 lists the event flags, and their mask, that can cause module interrupts.

Table 10-366. GPMC Interrupt Events

Event Flag	Event Mask	Sensitivity	Description
GPMC_IRQSTATUS[9] WAIT1EDGE DETECTIONSTATUS	GPMC_IRQENABLE[9] WAIT1EDGE DETECTIONENABLE	Edge	Wait1 edge detection interrupt: Triggered if a rising or falling edge is detected on the gpmc_wait1 signal. The rising or falling edge detection of Wait1 is selected through the GPMC_CONFIG[9] WAIT1PINPOLARITY bit.
GPMC_IRQSTATUS[8] WAIT0EDGE DETECTIONSTATUS	GPMC_IRQENABLE[8] WAIT0EDGE DETECTIONENABLE	Edge	Wait0 edge detection interrupt: Triggered if a rising or falling edge is detected on the gpmc_wait0 signal. The rising or falling edge detection of Wait0 is selected through the GPMC_CONFIG[8] WAIT0PINPOLARITY bit.
GPMC_IRQSTATUS[1] TERMINAL COUNTSTATUS	GPMC_IRQENABLE[1] TERMINAL COUNTENABLE	Level	Terminal count event: Triggered on prefetch process completion; that is, when the number of currently remaining data to be requested reaches 0
GPMC_IRQSTATUS[0] FIFOEVENTSTATUS	GPMC_IRQENABLE[0] FIFOEVENTENABLE	Level	FIFO event interrupt: Indicates available FIFO levels for write-posting mode and prefetch mode. GPMC_PREFETCH_CONFIG1[2] DMAMODE must be set to 0.

10.3.4.6 L3 Interconnect Interface

The GPMC L3 interconnect interface is a pipelined interface including a 16 × 32-bit word write buffer.

Any system host can issue external access requests through the GPMC.

The device system can issue the following requests through this interface:

- One 8-, 16-, or 32-bit interconnect access (read/write)
- Two incrementing 32-bit interconnect accesses (read/write)
- Two wrapped 32-bit interconnect accesses (read/write)
- Four incrementing 32-bit interconnect accesses (read/write)
- Four wrapped 32-bit interconnect accesses (read/write)
- Eight incrementing 32-bit interconnect accesses (read/write)
- Eight wrapped 32-bit interconnect accesses (read/write)

Only linear burst transactions are supported; interleaved burst transactions are not supported. Only power-of-two-length precise bursts 2 × 32, 4 × 32, 8 × 32, and 16 × 32, with the burst base address aligned on the total burst size, are supported (this limitation applies to incrementing bursts only).

This interface also provides one interrupt and one DMA request line for specific event control.

It is recommended to program the GPMC_CONFIG1_i[24:23] ATTACHEDDEVICEPAGELENGTH bit field according to the page length of the effective attached device and to enable the GPMC_CONFIG1_i[31] WRAPBURST bit if the attached device supports wrapping burst.

It is possible, however, to emulate wrapping burst on a nonwrapping memory by providing relevant addresses within the page or by splitting transactions. Bursts larger than the memory page length are chopped into multiple burst transactions. Because of the alignment requirements, a page boundary is never crossed.

10.3.4.7 GPMC Address and Data Bus

The current application supports GPMC connection to NAND devices and to address/data-multiplexed memories or devices. Connection to address/data-nonmultiplexed memories or devices is supported with an address range up to 256 MB (that is, 28 address bits in 8-bit mode or 27 address bits in 16-bit mode).

Depending on the GPMC configuration of each chip-select, address and data bus lines that are not required for a particular access protocol are not updated (changed from current value) and are not sampled when input (input data bus).

- For address/data-multiplexed and AAD-multiplexed NOR devices, the address is multiplexed on the data bus.
- 8-bit-wide NOR devices do not use GPMC I/O: gpmc_ad[15:8] for data (they are used for address if needed).
- 16-bit-wide NAND devices do not use GPMC I/O: gpmc_a[27:16].
- 8-bit-wide NAND devices do not use GPMC I/O: gpmc_a[27:16] and GPMC I/O: gpmc_ad[15:8].

10.3.4.7.1 GPMC I/O Configuration Setting

NOTE: In this section and the following sections, the i in GPMC_CONFIGx_i stands for the GPMC chip-select i, where i = 0 to 7.

To select a NAND device, program the following register fields:

- GPMC_CONFIG1_i[11:10] DEVICETYPE = 0b10
- GPMC_CONFIG1_i[9:8] MUXADDDATA = 0b00

To select an address/data-multiplexed device, program the following register fields:

- GPMC_CONFIG1_i[11:10] DEVICETYPE = 0b00
- GPMC_CONFIG1_i[9:8] MUXADDDATA = 0b10

To select an address/address/data-multiplexed device, program the following register fields:

- GPMC_CONFIG1_i[11:10] DEVICETYPE = 0b00
- GPMC_CONFIG1_i[9:8] MUXADDDATA = 0b01

To select an address/data-nonmultiplexed device, program the following register fields:

- GPMC_CONFIG1_i[11:10] DEVICETYPE = 0b00
- GPMC_CONFIG1_i[9:8] MUXADDDATA = 0b00

10.3.4.7.2 GPMC CS0 Default Configuration at Device Reset

To ensure a correct, fast external boot with a GPMC access on device reset, several pins are sampled:

- The "sysboot0" through "sysboot5" pins (device boundary) define the sequence of interfaces and devices to use for booting (i.e. SYSBOOT[5:0] vector). They are sampled by the control module at reset and used later by the device ROM code. For more information, see Section 25.2.4, *Sysboot Configuration*, in Chapter 25, *Initialization*.
- Additional pins are used to configure reset values in the GPMC_CONFIG1_i register (where i = 0) as explained in the following and in Table 10-367:
  - The *bootdevicesize* input pin (at the GPMC boundary) defines the size of the attached device on chip-select 0 (CS0) and is used to configure the GPMC_CONFIG1_i[13:12] DEVICESIZE bit field (where i = 0). The BOOT_DEVICE_SIZE signal is propagated from the device Control Module. Its value 0b0 (8-bit memories) or 0b1 (16-bit memories) can be externally determined upon booting by **user hardware** via the device external input signal - "sysboot13".
  - The *cs0muxdevice* input pin (at the GPMC boundary) selects whether or not the device attached to CS0 is an address/data-multiplexed device. The input pin is used to configure the GPMC_CONFIG1_i[9:8] MUXADDDATA bit field (where i = 0). The CS0_MUX_DEVICE[1:0] signal is propagated from the Control Module. Its value 0x0 **(non-muxed memory attached)** or 0x2 **(Addr-Data Mux memory attached)** can be externally determined upon booting by **user hardware** via combining the device external input signals - "sysboot12" and "sysboot11", i.e. SYSBOOT[12:11].
  - The *bootwaiten* input pin (at the GPMC boundary) enables the monitoring on CS0 of the wait pin at device reset release time for read accesses. The input pin is used to configure the GPMC_CONFIG1_i[22] WAITREADMONITORING bit (where i = 0). The BOOT_WAIT_EN signal is

propagated from the Control Module. Its value 0x0 **(wait pin is not monitored for read accesses)** or 0x1 **(wait pin is monitored for read accesses)** can be externally determined upon booting by **user hardware** via the device external input signal - "sysboot10".

NOTE: If WAIT pin monitoring function is enabled upon booting (i.e. BOOT_WAIT_EN="1"), the default (power-on-reset) monitored input for CS0 is always the device gpmc_wait0 input.

Table 10-367. Boot Control Interface Input Signals Description

Signal Name	Width	Description
BOOT_DEVICE_SIZE	1	Size of the device attached on CS0 0b00: 8-bit 0b01: 16-bit 0b10: Reserved (not used) 0b11: Reserved (not used)
CS0_MUX_DEVICE	2	Multiplexing mode of the device on CS0 0b00: Nonmultiplexed device on CS0 0b01: AAD-multiplexed device on CS0 (address-address-data) 0b10: Address/data-multiplexed device on CS0 0b11: Reserved
BOOT_WAIT_EN	1	Wait monitoring on CS0 at device reset release time for read accesses 0: Wait pin is not monitored 1: Wait pin is monitored

CAUTION

Using the internal boot code, the entire CS0 configuration can be modified before the first CS0 access. For more information, see Section 25.8, *Memory Booting*, and Section 25.9 , *Image Format*, in Chapter 25, *Initialization*. This modification of internal boot code is necessary for two external devices:

- NAND device attached to CS0
- Nonmultiplexed memory device attached to CS0

At reset time, the device can boot from the internal ROM.

The reset values of the timing control parameters are defined to cope with direct boot on address and data-multiplexed NOR flash devices, nonmultiplexed NOR flash devices, or any asynchronous device with large timing margins, assuming a low GPMC_FCLK frequency (for example, 19.2 MHz) at boot time.

For a multiplexed access, the address 16 low-order bits are presented onto gpmc_ad[15:0], while the high-order bits are presented onto gpmc_a[26:16]. If the external chip interface to the memories uses a 16-bit data bus, the high-order address bits are sampled on the address bus.

The reset values of timing parameters used at boot time are:

- CSONTIME = 1
- CSRDOFFTIME = 16
- ADVONTIME = 4
- ADVRDOFFTIME = 5
- OEONTIME = 6
- OEOFFTIME = 16
- RDACCESSTIME = 15
- RDCYCLETIME = 17

For an AAD-multiplexed access, all address bits are passed onto the data bus using two nADV rising edges. The first rising edge latches the address most-significant bit (MSB) down to bit 17, while the second rising edge latches address bits 16 down to 1. This configuration is only used for 16-bit memories.

The reset values of these timing parameters used at boot time are:

- ADVAADMUXONTIME = 1
- ADVAADMUXRDOFFTIME = 2
- OEAADMUXONTIME = 1
- OEAADMUXOFFTIME = 3

10.3.4.8 Address Decoder and Chip-Select Configuration

Addresses are decoded accordingly with the address request of the chip-select and the content of the chip-select base address register file, which includes a set of global GPMC configuration registers and eight sets of chip-select configuration registers.

The GPMC configuration register file is memory-mapped and can be read or written with byte, 16-bit word, or 32-bit word accesses. The register file must be configured as a noncacheable, nonbufferable region to prevent any desynchronization between host execution (write request) and the completion of register configuration (write completed with register updated). Section 10.3.7, *GPMC Register Manual*, provides the GPMC register locations. For the map of GPMC memory locations, see Chapter 2, *Memory Mapping*.

After the chip-select is configured, the access engine accesses the external device, drives the external interface control signals, and applies the interface protocol based on user-defined timing parameters and settings.

10.3.4.8.1 Chip-Select Base Address and Region Size

Any external memory or ASIC device attached to the GPMC external interface can be accessed by any device system host within the GPMC 512-MiB address space. For more information, see Chapter 2, *Memory Mapping*.

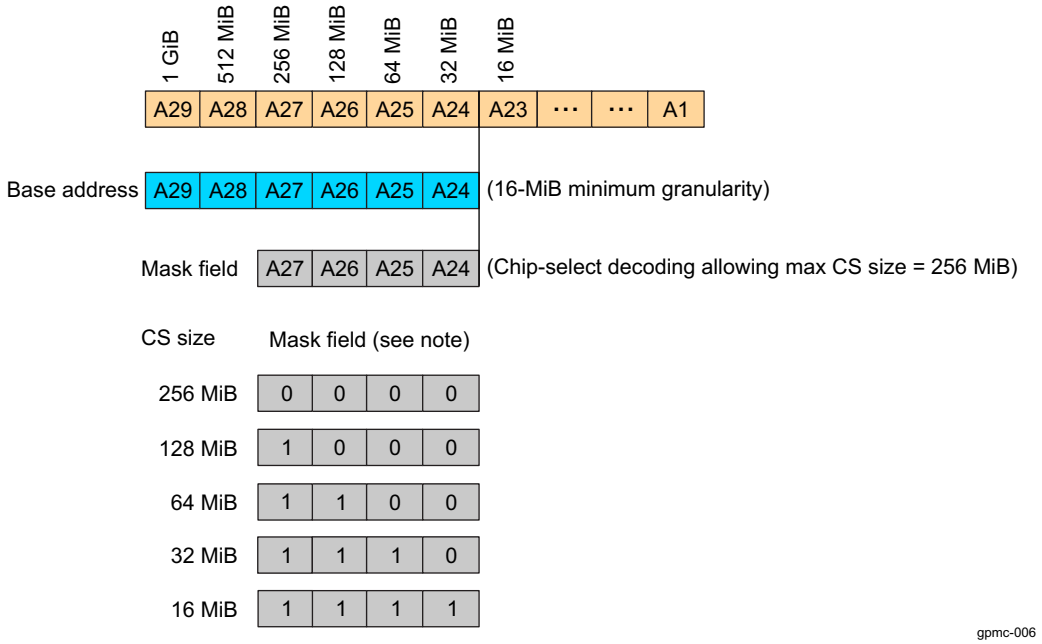
NOTE: Even though GPMC supports total address space of 1GB, only 512MB are physically available in this device.

The GPMC 512-MiB address space can be divided into a maximum of eight chip-select regions with programmable base address and programmable chip-select size. The chip-select size is programmable from 16 MiB to 256 MiB (must be a power-of-two) and is defined by the mask field. Attached memory smaller than the programmed chip-select region size is accessed through the entire chip-select region (aliasing).

Each chip-select has a 6-bit base address encoding and 4-bit decoding mask, which must be programmed according to the following rules:

- The programmed chip-select region base address must be aligned on the chip-select region size address boundary and is limited to a power-of-two address value. During access decoding, the value of the register base address is used to compare the address with the address bit line mapping, as shown in Figure 10-18 (with A0 as the device system byte-address line). The base address is programmed through the GPMC_CONFIG7_i[5:0] BASEADDRESS bit field.
- The register mask is used to exclude some address lines from the decoding. A register mask bit field set to 0 suppresses the associated address line from the address comparison (incoming address bit line is don't care). The value of the register mask must be limited to the subsequent value, based on the desired chip-select region size. Any other value has an undefined result. When multiple chip-select regions with overlapping addresses are enabled concurrently, access to these chip-select regions is cancelled and a GPMC access error is posted. The mask field is programmed through the GPMC_CONFIG7_i[11:8] MASKADDRESS bit field.

Figure 10-18. Chip-Select Address Mapping and Decoding Mask



Following is an example mapping that divides the 512MB memory reach into two 256MB regions:

- 1st 256MB region – 0x0000 0000 to 0x0FFF FFFF – program base address for that chip select equal to 0x0
- 2nd 256MB region – 0x1000 0000 to 0x1FFF FFFF – program base address for that chip select equal to 0x10

Chip-select configuration (base and mask address or any protocol and timing settings) must be performed while the associated chip-select is disabled through the GPMC_CONFIG7_i[6] CSVALID bit (where i stands for the GPMC chip-select value, 0 to 7). In addition, a chip-select configuration can be disabled only if there is no ongoing access to that chip-select. This requires monitoring the activity of the prefetch or write-posting engine if the engine is active on the chip-select. Also, the write buffer state must be monitored to wait for any posted write completion to the chip-select.

Any access attempted to a nonvalid GPMC address region (CSVALID disabled or address decoding outside a valid chip-select region) is not propagated to the external interface and a GPMC access error is posted. In case of overlapping chip-selects, an error is generated and no access occurs on either chip-select.

CS0 is the only chip-select region enabled after a power up or GPMC reset.

CAUTION

Although the GPMC interface can drive up to eight chip-selects, the frequency specified for this interface is for a specific load. If this load is exceeded, the maximum frequency cannot be reached. One solution is to implement a board with buffers to allow the slowest device to maintain the total load on the lines.

10.3.4.8.2 Access Protocol

10.3.4.8.2.1 Supported Devices

The access protocol of each chip-select can be independently specified through the GPMC_CONFIG1_i[11:10] DEVICETYPE parameter (where i = 0 to 7) for:

- Random-access synchronous or asynchronous memory, such as NOR flash and SRAM

- NAND flash asynchronous devices

NOTE: For more information about the NAND flash GPMC basic programming model and NAND support, see Section 10.3.4.12, *NAND Device Basic Programming Model*, and Section 10.3.4.12.1, *NAND Memory Device in Byte or Word16 Stream Mode*.

10.3.4.8.2.2 Access Size Adaptation and Device Width

Each chip-select can be independently configured through the GPMC_CONFIG1_i[13:12] DEVICESIZE bit field (where i = 0 to 7) to interface with a 16- or 8-bit-wide device. System requests with data width greater than the external device data bus width are split into successive accesses according to the external device data-bus width and little-endian data organization.

10.3.4.8.2.3 Address/Data-Multiplexing Interface

For random synchronous or asynchronous memory interfacing (DEVICETYPE = 0b00), an address- and data-multiplexing protocol can be selected through the GPMC_CONFIG1_i[9:8] MUXADDDATA bit field (where i = 0 to 7). The nADV signal must be used as the external device address latch control signal. For the associated chip-select configuration, nADV assertion and deassertion time and nOE assertion time must be set to the appropriate value to meet the address latch setup/hold time requirements of the external device. See Section 10.3.3, *GPMC Integration*.

NOTE: This address/data-multiplexing interface is not applicable to NAND device interfacing. NAND devices require a specific address, command, and data-multiplexing protocol. See Section 10.3.4.12, *NAND Device Basic Programming Model*.

10.3.4.8.3 External Signals

NOTE: The "DIR" (direction control) output signal is NOT pinned out on any of the device pads. It is an internal signal only representing a signal direction on the GPMC data bus.

10.3.4.8.3.1 Wait Pin Monitoring Control

GPMC access time can be dynamically controlled using an external gpmc_wait pin when the external device access time is not deterministic and cannot be defined and controlled using only the GPMC internal RDACCESSTIME, WRACCESSTIME, and PAGEBURSTACCESSTIME wait-state generator.

The GPMC features two input wait pins: gpmc_wait1, and gpmc_wait0. These pins allow control of external devices with different wait pin polarity. They also allow the overlap of wait pin assertion from different devices without affecting access to devices for which the wait pin is not asserted.

- The GPMC_CONFIG1_i[17:16] WAITPINSELECT bit field (where i = 0 to 7) selects which input gpmc_wait pin is used for the device attached to the corresponding chip-select.
- The polarity of the wait pin is defined through the WAITxPINPOLARITY bit of the GPMC_CONFIG register. A wait pin configured to be active low means that low level on the WAIT signal indicates that the data is not ready and that the data bus is invalid. When a wait pin is inactive, data is valid.

The GPMC access engine can be configured per chip-select to monitor or not the wait pin of the external memory device, based on the access type: read or write.

- The GPMC_CONFIG1_i[22] WAITREADMONITORING bit defines whether or not the wait pin must be monitored during read accesses.
- The GPMC_CONFIG1_i[21] WAITWRITEMONITORING bit defines whether or not the wait pin must be monitored during write accesses.

The GPMC access engine can be configured to monitor the wait pin of the external memory device asynchronously or synchronously with the GPMC_CLK clock, depending on the access type: synchronous or asynchronous (the GPMC_CONFIG1_i[29] READTYPE and GPMC_CONFIG1_i[27] WRITETYPE bits).

10.3.4.8.3.1.1 Wait Monitoring During Asynchronous Read Access

When wait pin monitoring is enabled for read accesses (WAITREADMONITORING), the effective access time is a logical AND combination of the RDACCESSTIME timing completion and the wait-deasserted state.

During asynchronous read accesses with wait pin monitoring enabled, the wait pin must be at a valid level (asserted or deasserted) for at least two GPMC clock cycles before RDACCESSTIME completes, to ensure correct dynamic access-time control through wait pin monitoring. The advance pipelining of the two GPMC clock cycles is the result of the internal synchronization requirements for the WAIT signal.

In this context, RDACCESSTIME is used as a wait invalid timing window and is set to such a value that the wait pin is at a valid state two GPMC clock cycles before RDACCESSTIME completes.

Similarly, during a multiple-access cycle (for example, asynchronous read page mode), the effective access time is a logical AND combination of PAGEBURSTACCESSTIME timing completion and the wait-deasserted state. Wait monitoring pipelining is also applicable to multiple accesses (access within a page).

- Wait monitored as active freezes the CYCLETIME counter. For an access within a page, when the CYCLETIME counter is by definition in a lock state, wait monitored as asserted extends the current access time in the page. Control signals are kept in their current state. The data bus is considered invalid, and no data are captured during this clock cycle.
- Wait monitored as inactive unfreezes the CYCLETIME counter. For an access within a page, when the CYCLETIME counter is by definition in a lock state, wait monitored as inactive completes the current access time and starts the next access phase in the page. The data bus is considered valid, and data are captured during this clock cycle. In case of a single access or if this was the last access in a multiple-access cycle, all signals are controlled according to their related control timing value and according to the CYCLETIME counter status.

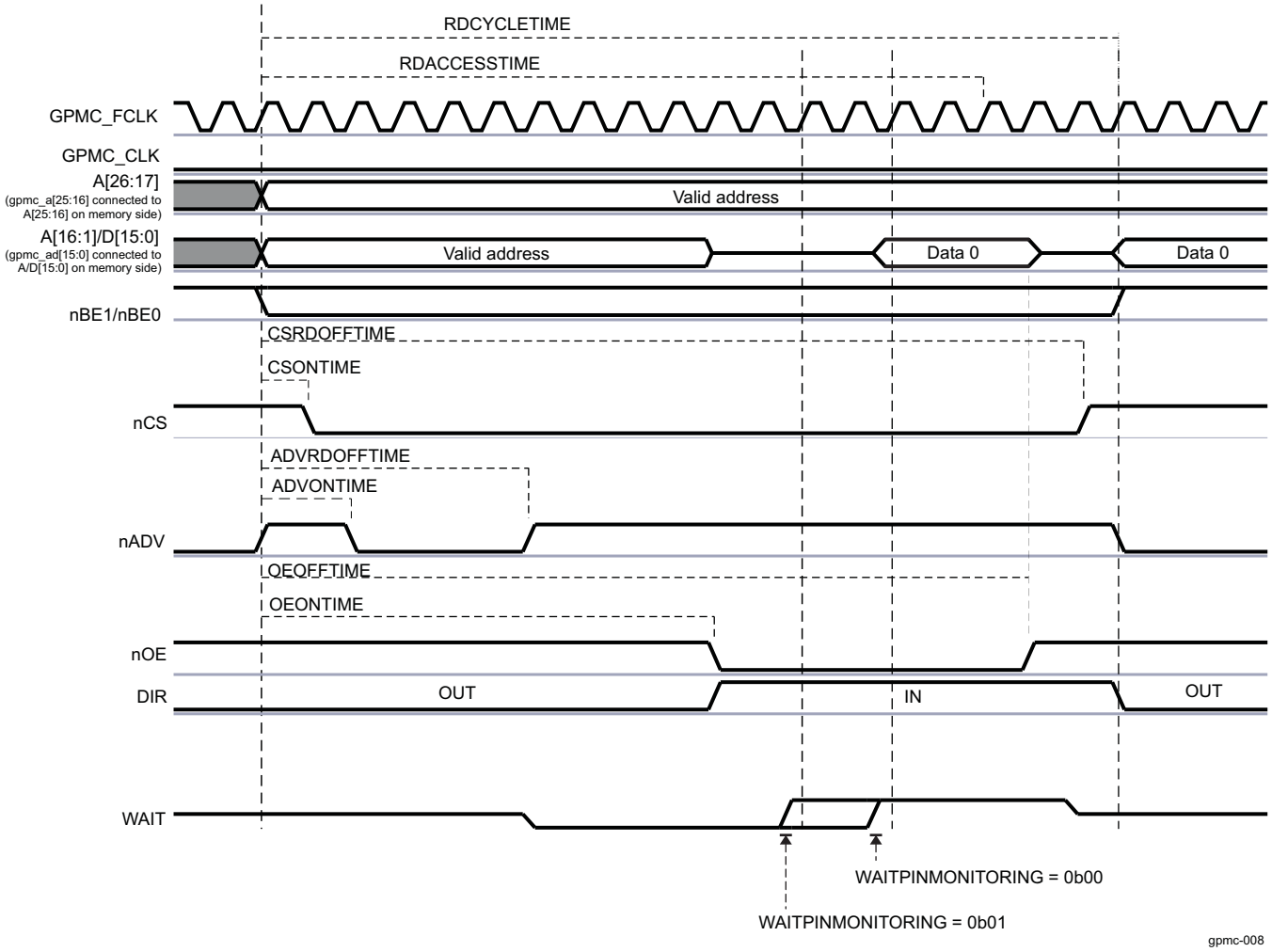
When a delay larger than two GPMC clocks must be observed between wait-pin deactivation time and data valid time (including the required GPMC and the device data setup time), an extra delay can be added between wait-pin deassertion time detection and effective data-capture time and the effective unlock of the CYCLETIME counter. This extra delay can be programmed in the GPMC_CONFIG1_i[19:18] WAITMONITORINGTIME bit field (where i = 0 to 7).

NOTE:

- The WAITMONITORINGTIME parameter does not delay the wait pin active or inactive detection, nor does it modify the two GPMC clocks pipelined detection delay.
- This extra delay is expressed as a number of GPMC_CLK clock cycles, even though the access is defined as asynchronous, and no GPMC_CLK clock is provided to the external device. Still, because GPMCFCLKDIVIDER is used as a divider for the GPMC clock, it must be programmed to define the correct WAITMONITORINGTIME delay.

Figure 10-19 shows wait behavior during an asynchronous single read access.

Figure 10-19. Wait Behavior During an Asynchronous Single Read Access (GPMCFCLKDivider = 1)



NOTE: The WAIT signal is active low. GPMC_CONFIG1_i[19:18] WAITMONITORINGTIME = 0b00, or 0b01.

10.3.4.8.3.1.2 Wait Monitoring During Asynchronous Write Access

When wait pin monitoring is enabled for write accesses (GPMC_CONFIG1_i[21] WAITWRITEMONITORING bit = 0x1), the wait invalid timing window is defined by the WRACCESSTIME field. WRACCESSTIME must be set so that the wait pin is at a valid state two GPMC clock cycles before WRACCESSTIME completes. The advance pipelining of the two GPMC clock cycles is the result of the internal synchronization requirements for the WAIT signal.

- Wait monitored as active freezes the CYCLETIME counter. This informs the GPMC that the data bus is not captured by the external device. The control signals are kept in their current state. The data bus still drives the data.
- Wait monitored as inactive unfreezes the CYCLETIME counter. This informs that the data bus is correctly captured by the external device. All signals, including the data bus, are controlled according to their related control timing value and to the CYCLETIME counter status.

When a delay larger than two GPMC clock cycles must be observed between wait-pin deassertion time and the effective data write into the external device (including the required GPMC data setup time and the device data setup time), an extra delay can be added between wait-pin deassertion time detection and effective data write time into the external device and the effective unfreezing of the CYCLETIME counter. This extra delay can be programmed in the GPMC_CONFIG1_i[19:18] WAITMONITORINGTIME bit field (where i = 0 to 7).

NOTE:

- The WAITMONITORINGTIME parameter does not delay the wait pin assertion or deassertion detection, nor does it modify the two GPMC clock cycles pipelined detection delay.
- This extra delay is expressed as a number of GPMC_CLK clock cycles, even though the access is defined as asynchronous, and even though no clock is provided to the external device. Still, because the GPMC_CONFIG1_i[1:0] GPMCFCLKDIVIDER bit field is used as a divider for the GPMC clock, it must be programmed to define the correct WAITMONITORINGTIME delay.

10.3.4.8.3.1.3 Wait Monitoring During Synchronous Read Access

During synchronous accesses with wait pin monitoring enabled, the wait pin is captured synchronously with GPMC_CLK, using the rising edge of this clock.

The WAIT signal can be programmed to apply to the same clock cycle in which it is captured. Alternatively, it can be sampled one or two GPMC_CLK cycles ahead of the clock cycle to which it applies. This pipelining is applicable to the entire burst access and to all data phases in the burst access. This wait pipelining depth is programmed in the GPMC_CONFIG1_i[19:18] WAITMONITORINGTIME bit field (where i = 0 to 7), and is expressed as a number of GPMC_CLK clock cycles.

In synchronous mode, when wait pin monitoring is enabled (the GPMC_CONFIG1_i[22] WAITREADMONITORING bit), the effective access time is a logical AND combination of the RDACCESSTIME timing completion and the wait-deasserted state detection.

Depending on the programmed value of WAITMONITORINGTIME, the wait pin must be at a valid level, either asserted or deasserted:

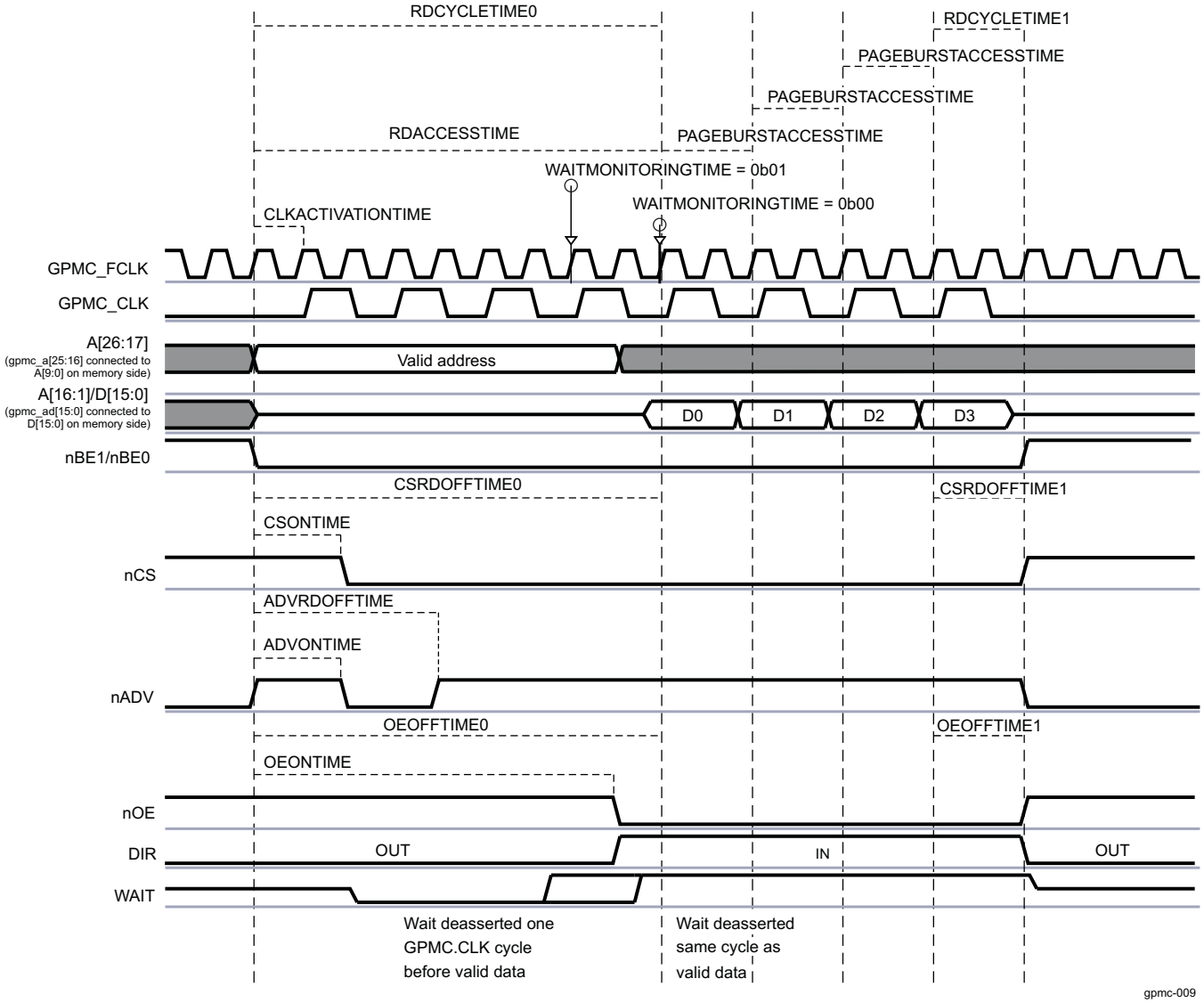
- In the same clock cycle the data is valid if WAITMONITORINGTIME = 0 (at RDACCESSTIME completion)
- In the WAITMONITORINGTIME x (GPMCFCLKDIVIDER + 1) GPMC_FCLK clock cycles before RDACCESSTIME completion if WAITMONITORINGTIME is not equal to 0

Similarly, during a multiple-access cycle (burst mode), the effective access time is a logical AND combination of PAGEBURSTACCESSTIME timing completion and the WAIT-INACTIVE state. The wait pipelining-depth programming applies to the whole burst access.

- Wait monitored as active freezes the CYCLETIME counter. For an access within a burst (when the CYCLETIME counter is by definition in a lock state), wait monitored as active extends the current access time in the burst. Control signals are kept in their current state. The data bus is considered invalid, and no data are captured during this clock cycle.
- Wait monitored as inactive unfreezes the CYCLETIME counter. For an access within a burst (when the CYCLETIME counter is by definition in lock state), wait monitored as inactive completes the current access time and starts the next access phase in the burst. The data bus is considered valid, and data are captured during this clock cycle. In a single access or if this was the last access in a multiple-access cycle, all signals are controlled according to their relative control timing value and the CYCLETIME counter status.

Figure 10-20 shows wait behavior during a synchronous read burst access.

Figure 10-20. Wait Behavior During a Synchronous Read Burst Access



NOTE: The WAIT signal is active low. WAITMONITORINGTIME = 00, 01.

10.3.4.8.3.1.4 Wait Monitoring During Synchronous Write Access

During synchronous accesses with wait pin monitoring enabled (the WAITWRITEMONITORING bit), the wait pin is captured synchronously with GPMC_CLK, using the rising edge of this clock.

If enabled, external wait pin monitoring can be used in combination with WRACCESSTIME to delay the GPMC_CLK capture edge of the effective memory device.

Wait-monitoring pipelining depth is similar to synchronous read access:

- At WRACCESSTIME completion if WAITMONITORINGTIME = 0
- In the WAITMONITORINGTIME x (GPMCFCLKDIVIDER + 1) GPMC_FCLK cycles before WRACCESSTIME completion if WAITMONITORINGTIME is not equal to 0

Wait-monitoring pipelining definition applies to whole burst accesses:

- Wait monitored as active freezes the CYCLETIME counter. For accesses within a burst, when the CYCLETIME counter is by definition in a lock state, wait monitored as active indicates that the data

bus is not being captured by the external device. Control signals are kept in their current state. The data bus is kept in its current state.

- Wait monitored as inactive unfreezes the CYCLETIME counter. For accesses within a burst, when the CYCLETIME counter is by definition in a lock state, wait monitored as inactive indicates the effective data capture of the bus by the external device and starts the next access of the burst. In case of a single access or if this was the last access in a multiple access cycle, all signals, including the data bus, are controlled according to their related control timing value and the CYCLETIME counter status.

NOTE: Wait monitoring is supported for all configurations except GPMC_CONFIG1_i[19:18] WAITMONITORINGTIME = 0x0 (where i = 0 to 7) for write bursts with a clock divider of 1 or 2 (the GPMC_CONFIG1_i[1:0] GPMCFCLKDIVIDER bit field is equal to 0x0 or 0x1, respectively).

10.3.4.8.3.1.5 Wait With NAND Device

For information about the use of the wait pin for communication with a NAND flash external device, see Section 10.3.4.12.2, *NAND Device-Ready Pin*.

10.3.4.8.3.1.6 Idle Cycle Control Between Successive Accesses

10.3.4.8.3.1.6.1 Bus Turnaround (BUSTURNAROUND)

To prevent data-bus contention, an access that follows a read access to a slow memory/device must be delayed (in other words, control the nCS/nOE deassertion to data bus in high-impedance delay).

The bus turnaround is a time-out counter starting after nCS or nOE deassertion time, whichever occurs first, and delays the next access start-cycle time. The counter is programmed through the GPMC_CONFIG6_i[3:0] BUSTURNAROUND bit field (where i = 0 to 7).

After a read access to a chip-select with a nonzero BUSTURNAROUND, the next access is delayed until the BUSTURNAROUND delay completes, if the next access is one of the following:

- A write access to any chip-select (the same or different chip-select from which the data was read)
- A read access to a different chip-select than the chip-select from which the data was read access
- A read or write access to a chip-select associated with an address/data-multiplexed device

NOTE: Bus keeping starts after bus turnaround completion so that DIR changes from IN to OUT after bus turnaround. The bus does not have enough time to go into high-impedance even though it can be driven with the same value before bus turnaround timing.

BUSTURNAROUND delay runs in parallel with GPMC_CONFIG6_i[3:0] CYCLE2CYCLEDELAY bi field delays. BUSTURNAROUND is a timing parameter for the ending chip-select access, while CYCLE2CYCLEDELAY is a timing parameter for the following chip-select access. The effective minimum delay between successive accesses is driven by these delay timing parameters and by the access type of the following access (see Figure 10-21 through Figure 10-23).

Another way to prevent bus contention is to define an earlier nCS or nOE deassertion time for slow devices or to extend the value of RDCYCLETIME. Doing this prevents bus contention, but it also affects all accesses of this specific chip-select.

Figure 10-21. Read-to-Read for an Address-Data Multiplexed Device, on Different Chip-Select, Without Bus Turnaround (nCS Attached to a Fast Device)

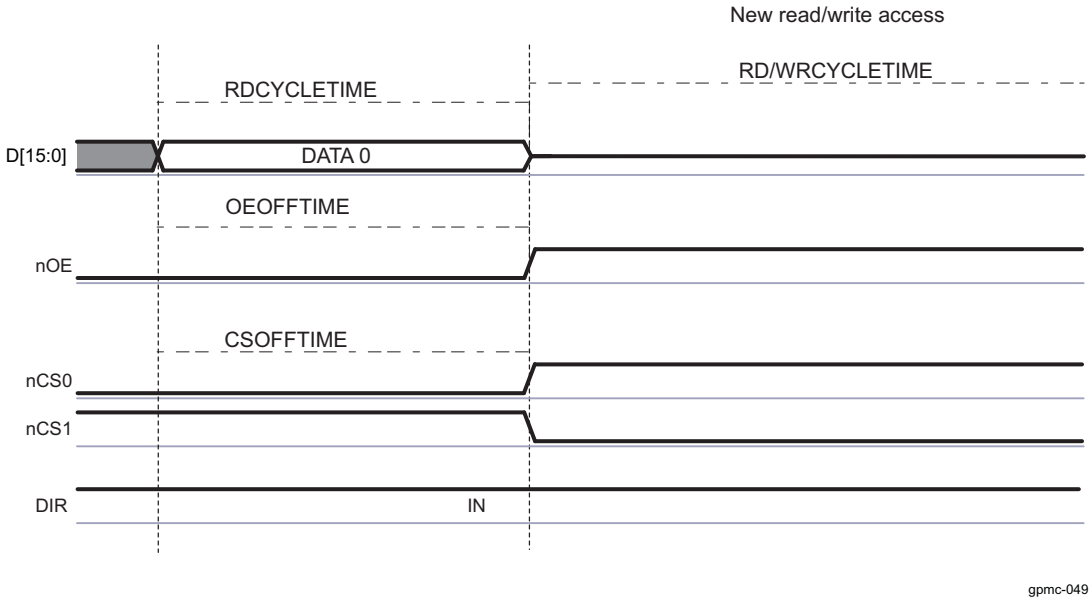


Figure 10-22. Read- to-Read/Write for an Address-Data Multiplexed Device, on Different Chip-Select, With Bus Turnaround

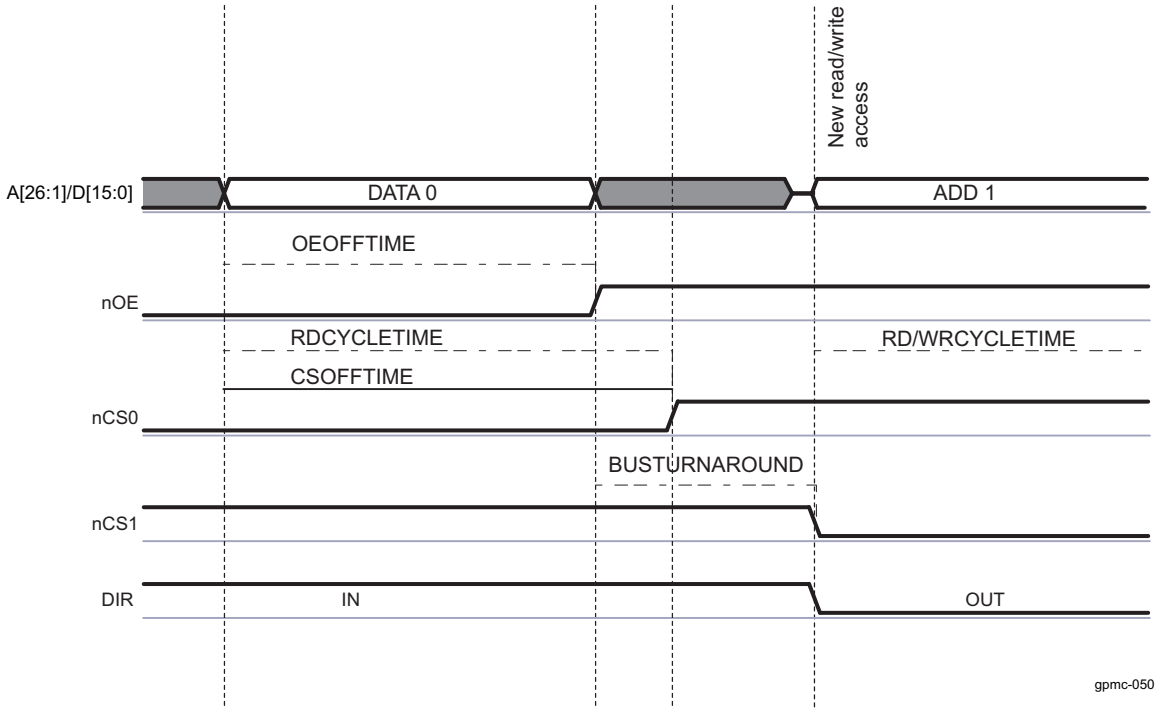
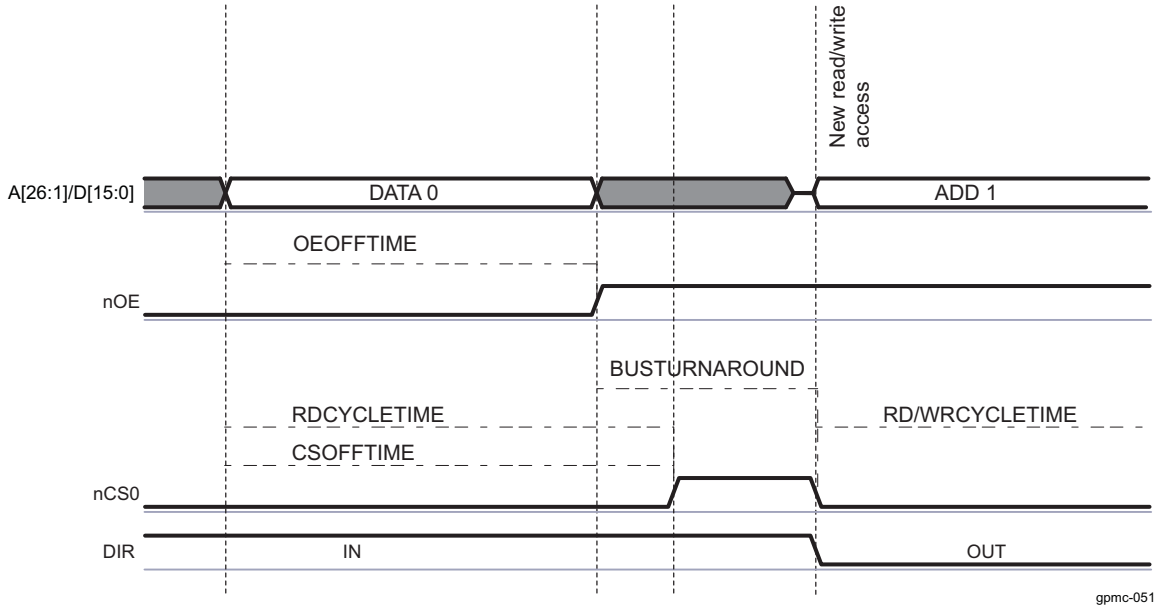


Figure 10-23. Read-to-Read/Write for a Address-Data or AAD-Multiplexed Device, on Same Chip-Select, With Bus Turnaround



10.3.4.8.3.1.6.2 Idle Cycles Between Accesses to Same Chip-Select (CYCLE2CYCLESAMECSEN, CYCLE2CYCLEDELAY)

Some devices require a minimum chip-select signal inactive time between accesses. The GPMC_CONFIG6_i[7] CYCLE2CYCLESAMECSEN bit (where i = 0 to 7) enables insertion of a minimum number of GPMC_FCLK cycles, defined by the GPMC_CONFIG6_i[11:8] CYCLE2CYCLEDELAY bit field, between successive accesses of any type (read or write) to the same chip-select.

If CYCLE2CYCLESAMECSEN is enabled, any subsequent access to the same chip-select is delayed until its CYCLE2CYCLEDELAY completes. The CYCLE2CYCLEDELAY counter starts when CSRDOFFTIME/CSWROFFTIME completes.

The same applies to successive accesses occurring during 32-bit word or burst accesses split into successive single accesses when the single-access mode is used (GPMC_CONFIG1_i[30] READMULTIPLE = 0 or GPMC_CONFIG1_i[28] WRITEMULTIPLE = 0).

All control signals (CS, ADV#/ALE, BE0#/CLE, WE#, and GPMC.CLK) are kept inactive (ADV#/ALE, BE0#/CLE, and GPMC.CLK at low level; and CS, OE#/RE, and WE at high level) during the idle GPMC.FCLK cycles. This prevents back-to-back accesses to the same chip-select without idle cycles between accesses.

10.3.4.8.3.1.6.3 Idle Cycles Between Accesses to Different Chip-Select (CYCLE2CYCLEDIFFCSEN, CYCLE2CYCLEDELAY)

Because of the pipelined behavior of the system, successive accesses to different chip-selects can occur back-to-back with no idle cycles between accesses. Depending on the control signals (nCS, nADV/ALE, nBE0/CLE, nOE/RE, nWE) assertion and deassertion timing parameters and on the device timing parameters, the assertion times of some control signals may overlap between the successive accesses to a different chip-select. Similarly, some control signals (WE, OE/RE) may not respect required transition times.

To work around overlapping and to observe the required control-signal transitions, a minimum of CYCLE2CYCLEDELAY inactive cycles is inserted between the access being initiated to this chip-select and the previous access ending for a different chip-select. This applies to any type of access (read or write).

If the GPMC_CONFIG6_i[6] CYCLE2CYCLEDIFFCSEN bit is enabled, the chip-select access is delayed until CYCLE2CYCLEDELAY cycles have expired since the end of a previous access to a different chip-select. CYCLE2CYCLEDELAY count starts at CSRDOFFTIME/CSWROFFTIME completion. All control signals are kept inactive during the idle GPMC_FCLK cycles.

NOTE: CYCLE2CYCLESAMECSEN and CYCLE2CYCLEDIFFCSEN must be set in the GPMC_CONFIG6_i registers to get idle cycles inserted between accesses on this chip-select and after accesses to a different chip-select, respectively.

The CYCLE2CYCLEDELAY delay runs in parallel with the BUSTURNAROUND delay. The BUSTURNAROUND is a timing parameter defined for the ending chip-select access, whereas CYCLE2CYCLEDELAY is a timing parameter defined for the starting chip-select access. The effective minimum delay between successive accesses is based on the larger delay timing parameter and on access type combination, because bus turnaround does not apply to all access types. For more information about bus turnaround, see Section 10.3.4.8.3.1.6.1, *Bus Turnaround (BUSTURNAROUND)*.

Table 10-368 describes the configuration required for idle cycle insertion.

Table 10-368. Idle Cycle Insertion Configuration

1st Access Type	BUSTURNAROUND Timing Parameter	Second Access Type	Chip-Select	Add/Data Multiplexed	CYCLE2CYCLE SAMECSEN Parameter	CYCLE2CYCLE DIFFCSEN Parameter	Idle Cycle Insertion Between the Two Accesses
R/W	= 0	R/W	Any	Any	0	x	No idle cycles are inserted if the two accesses are well pipelined.
R	> 0	R	Same	Nonmuxed	x	0	No idle cycles are inserted if the two accesses are well pipelined.
R	> 0	R	Different	Nonmuxed	0	0	BUSTURNAROUND cycles are inserted.
R	> 0	R/W	Any	Muxed	0	0	BUSTURNAROUND cycles are inserted.
R	> 0	W	Any	Any	0	0	BUSTURNAROUND cycles are inserted.
W	> 0	R/W	Any	Any	0	0	No idle cycles are inserted if the two accesses are well pipelined.
R/W	= 0	R/W	Same	Any	1	x	CYCLE2CYCLEDELAY cycles are inserted.
R/W	= 0	R/W	Different	Any	x	1	CYCLE2CYCLEDELAY cycles are inserted.
R/W	> 0	R/W	Same	Any	1	x	CYCLE2CYCLEDELAY cycles are inserted. If BTA idle cycles already apply on these two back-to-back accesses, the effective delay is max (BUSTURNAROUND, CYCLE2CYCLEDELAY).
R/W	> 0	R/W	Different	Any	x	1	CYCLE2CYCLEDELAY cycles are inserted. If BTA idle cycles already apply on these two back-to-back accesses, the effective delay is maximum (BUSTURNAROUND, CYCLE2CYCLEDELAY).

10.3.4.8.3.1.7 Slow Device Support (TIMEPARAGRANULARITY Parameter)

All access-timing parameters can be multiplied by 2 by setting the GPMC_CONFIG1_i[4] TIMEPARAGRANULARITY bit (where i stands for the GPMC chip-select value, 0 to 7). Increasing all access timing parameters allows support of slow devices.

10.3.4.8.3.2 Reset

No reset signal is sent to the external memory device by the GPMC. For more information about external-device reset, see Chapter 3, *Power, Reset, and Clock Management*.

The PRCM module provides an input pin, global_rst_n, to the GPMC:

- The global_rst_n pin is activated during device warm reset and cold reset.
- The global_rst_n pin initializes the internal state-machine and the internal configuration registers.

10.3.4.8.3.3 Byte Enable (nBE1/nBE0)

Byte enable signals (nBE1/nBE0) are:

- Valid (asserted or nonasserted according to the incoming system request) from access start to access completion for asynchronous and synchronous single accesses
- Asserted low from access start to access completion for asynchronous and synchronous multiple read accesses
- Valid (asserted or nonasserted, according to the incoming system request) synchronously to each written data for synchronous multiple write accesses

10.3.4.8.4 Error Handling

When an error occurs in the GPMC, the error information is stored in the GPMC_ERR_TYPE register and the address of the illegal access is stored in the GPMC_ERR_ADDRESS register. The GPMC keeps only the first error abort information until the GPMC_ERR_TYPE register is reset. Subsequent accesses that cause errors are not logged until the error is cleared by hardware with the GPMC_ERR_TYPE[0] ERRORVALID bit.

- ERRORNOTSUPPADD occurs when an incoming system request address decoding does not match any valid chip-select region, or if two chip-select regions are defined as overlapped, or if a register file access is tried outside the valid address range of 1KiB.
- ERRORNOTSUPPMCMD occurs when an unsupported command request is decoded at the L3 interconnect interface.
- ERRORTIMEOUT: A time-out mechanism prevents the system from hanging. The start value of the 9-bit time-out counter is defined in the GPMC_TIMEOUT_CONTROL register and enabled with the GPMC_TIMEOUT_CONTROL[0] TIMEOUTENABLE bit. When enabled, the counter starts at start-cycle time until it reaches 0 and data is not responded to from memory, and then a time-out error occurs. When data are sent from memory, this counter is reset to its start value. With multiple accesses (asynchronous page mode or synchronous burst mode), the counter is reset to its start value for each data access within the burst.

The GPMC does not generate interrupts on these errors. An interrupt generation is handled at interconnect level.

10.3.4.9 Timing Setting

The GPMC offers maximum flexibility to support various access protocols. Most of the timing parameters of the protocol access used by the GPMC to communicate with attached memories or devices are programmable on a chip-select basis. Assertion and deassertion times of control signals are defined to match the attached memory or device timing specifications and to get maximum performance during accesses. For more information about GPMC_CLK and GPMC_FCLK see Section 10.3.4.9.6, *GPMC_CLK*.

NOTE: In the following sections, the start access time refers to the time at which the access begins.

10.3.4.9.1 Read Cycle Time and Write Cycle Time (RDCYCLETIME / WRCYCLETIME)

The GPMC_CONFIG5_i[4:0] RDCYCLETIME and GPMC_CONFIG5_i[12:8] WRCYCLETIME bit fields (where i = 0 to 7) define the address bus and byte-enable valid times for read and write accesses. To ensure a correct duty cycle of GPMC_CLK between accesses, RDCYCLETIME and WRCYCLETIME are expressed in GPMC_FCLK cycles and must be multiples of the GPMC_CLK cycle. The RDCYCLETIME and WRCYCLETIME bit fields can be set with a granularity of 1 or 2 through the GPMC_CONFIG1_i[4] TIMEPARAGRANULARITY bit.

When RDCYCLETIME or WRCYCLETIME completes, if they are not already deasserted, all control signals (nCS, nADV/ALE, nOE/RE, nWE, and BE0/CLE) are deasserted to their reset values, regardless of their deassertion time parameters.

An exception to this forced deassertion occurs when a pipelined request to the same chip-select or to a different chip-select is pending. In such a case, it is not necessary to deassert a control signal with deassertion time parameters equal to the cycle-time parameter. This exception to forced deassertion prevents any unnecessary glitches. This requirement also applies to BE signals, thus avoiding an unnecessary BE glitch transition when pipelining requests.

NOTE: All control signals (CS, ADV#/ALE, BE0#/CLE, WE#, and GPMC.CLK) are kept inactive (ADV#/ALE, BE0#/CLE, and GPMC.CLK at low level; and CS, OE#/RE, and WE at high level) during the idle GPMC.FCLK cycles.

If no inactive cycles are required between successive accesses to the same chip-select or a different chip-select (GPMC_CONFIG6_i[7] CYCLE2CYCLESAMECSEN = 0 or GPMC_CONFIG6_i[6] CYCLE2CYCLEDIFFCSEN = 0, where i = 0 to 7), and if assertion-time parameters associated with the pipelined access are equal to 0, asserted control signals (nCS, nADV/ALE, nBE0/CLE, nWE, and nOE/RE) are kept asserted. This applies to any read/write to read/write access combination.

If inactive cycles are inserted between successive accesses (that is, CYCLE2CYCLESAMECSEN = 1 or CYCLE2CYCLEDIFFCSEN = 1), the control signals are forced to their respective default reset values for the number of GPMC_FCLK cycles defined in CYCLE2CYCLEDELAY.

10.3.4.9.2 nCS: Chip-Select Signal Control Assertion/Deassertion Time (CSONTIME / CSRDOFFTIME / CSWROFFTIME / CSEXTRADELAY)

The GPMC_CONFIG2_i[3:0] CSONTIME bit field (where i = 0 to 7) defines the nCS signal-assertion time relative to the start access time. It is common for read and write accesses.

The GPMC_CONFIG2_i[12:8] CSRDOFFTIME (read access) and GPMC_CONFIG2_i[20:16] CSWROFFTIME (write access) bit fields define the nCS signal deassertion time relative to start access time.

The CSONTIME, CSRDOFFTIME, and CSWROFFTIME parameters apply to synchronous and asynchronous modes. CSONTIME can be used to control an address and byte-enable setup time before chip-select assertion. CSRDOFFTIME and CSWROFFTIME can be used to control an address and byte-enable hold time after chip-select deassertion.

nCS signal transitions, as controlled through CSONTIME, CSRDOFFTIME, and CSWROFFTIME, can be delayed by a half-GPMC_FCLK period by enabling the GPMC_CONFIG2_i[7] CSEXTRADELAY bit. This half-GPMC_FCLK period provides more granularity on the nCS assertion and deassertion time to ensure proper setup and hold time relative to GPMC_CLK. CSEXTRADELAY is especially useful in configurations where GPMC_CLK and GPMC_FCLK have the same frequency, but it can also be used for all GPMC configurations. If enabled, CSEXTRADELAY applies to all parameters that control nCS transitions.

The CSEXTRADELAY bit must be used carefully to avoid control signal overlap between successive accesses to different chip-selects. This implies the need to program the RDCYCLETIME and WRCYCLETIME bit fields to be greater than the nCS signal-deassertion time, including the extra half-GPMC_FCLK-period delay.

10.3.4.9.3 nADV/ALE: Address Valid/Address Latch Enable Signal Control Assertion/Deassertion Time (ADVONTIME / ADVRDOFFTIME / ADVWROFFTIME /

ADVEXTRADELAY/ADVAADMUXONTIME/ADVAADMUXRDOFFTIME/ADVAADMUXWROFFTIME)

The GPMC_CONFIG3_i[3:0] ADVONTIME field (where i = 0 to 7) defines the nADV/ALE signal-assertion time relative to start access time. It is common to read and write accesses.

The GPMC_CONFIG3_i[12:8] ADVRDOFFTIME (read access) and GPMC_CONFIG3_i[20:16] ADVWROFFTIME (write access) bit fields define the nADV/ALE signal-deassertion time relative to start access time.

ADVONTIME can be used to control an address and byte-enable valid setup time control before nADV/ALE assertion. ADVRDOFFTIME and ADVWROFFTIME can be used to control an address and byte-enable valid hold time control after nADV/ALE deassertion. ADVRDOFFTIME and ADVWROFFTIME apply to synchronous and asynchronous modes.

The nADV/ALE signal transitions as controlled through ADVONTIME, ADVRDOFFTIME, and ADVWROFFTIME can be delayed by a half-GPMC_FCLK period by enabling the GPMC_CONFIG3_i[7] ADVEXTRADELAY bit. This half-GPMC_FCLK period provides more granularity on nADV/ALE assertion and deassertion time to ensure proper setup and hold time relative to GPMC_CLK. The ADVEXTRADELAY configuration parameter is especially useful in configurations where GPMC_CLK and GPMC_FCLK have the same frequency, but can be used for all GPMC configurations. If enabled, ADVEXTRADELAY applies to all parameters controlling nADV/ALE transitions.

ADVEXTRADELAY must be used carefully to avoid control-signal overlap between successive accesses to different chip-selects. This implies the need to program the RDCYCLETIME and WRCYCLETIME bit fields to be greater than nADV/ALE signal-deassertion time, including the extra half-GPMC_FCLK-period delay.

GPMC_CONFIG3_i[6:4] ADVAADMUXONTIME, GPMC_CONFIG3_i[26:24] ADVAADMUXRDOFFTIME, and GPMC_CONFIG3_i[30:28] ADVAADMUXWROFFTIME parameters have the same functions as ADVONTIME, ADVRDOFFTIME, and ADVWROFFTIME, but apply to the first address phase in the AAD-multiplexed protocol. The user must ensure that ADVAADMUXxxOFFTIME is programmed to a value less than or equal to ADVxxOFFTIME. Functionality in AAD-multiplexed mode is undefined if the settings do not comply with this requirement. ADVAADMUXxxOFFTIME can be programmed to the same value as ADVONTIME if no high nADV pulse is needed between the two AAD-multiplexed address phases, which is the typical case in synchronous mode. In this configuration, nADV is kept low until it reaches the correct ADVxxOFFTIME.

For more information about the use of ADVONTIME, ADVRDOFFTIME, ADVWROFFTIME, and ADVAADMUXRDOFFTIME and ADVAADMUXWROFFTIME for command latch enable (CLE) and address latch enable (ALE) use for a NAND flash interface, see Section 10.3.4.12, *NAND Access Description*.

10.3.4.9.4 nOE/nRE: Output Enable/Read Enable Signal Control Assertion/Deassertion Time (OEONTIME / OEOFFTIME / OEEXTRADELAY / OEAADMUXONTIME / OEAADMUXOFFTIME)

The GPMC_CONFIG4_i[3:0] OEONTIME bit field (where i = 0 to 7) defines the nOE/nRE signal assertion time relative to start access time. It applies only to read accesses.

The GPMC_CONFIG4_i[12:8] OEOFFTIME bit field defines the nOE/nRE signal deassertion time relative to start access time. It applies only to read accesses. nOE/nRE is not asserted during a write cycle.

The OEONTIME, OEOFFTIME, OEAADMUXONTIME, and OEAADMUXOFFTIME parameters apply to synchronous and asynchronous modes. OEONTIME can be used to control an address and byte enable valid setup time control before nOE/nRE assertion. OEOFFTIME can be used to control an address and byte-enable valid hold time control after nOE/nRE assertion.

The OEAADMUXONTIME and OEAADMUXOFFTIME parameters have the same functions as OEONTIME and OEOFFTIME, but apply to the first OE assertion in the AAD-multiplexed protocol for a read phase, or to the only OE assertion for a write phase. The user mut ensure that OEAADMUXOFFTIME is programmed to a value less than OEONTIME. Functionality in AAD-multiplexed mode is undefined if the settings do not comply with this requirement. OEAADMUXOFFTIME must never be equal to OEONTIME because the AAD-multiplexed protocol requires a second address phase with the nOE signal deasserted before nOE can be asserted again to define a read command.

The nOE/RE signal transitions as controlled through OEONTIME, OEOFFTIME, OEAADMUXONTIME, and OEAADMUXOFFTIME can be delayed by a half-GPMC_FCLK period by enabling the GPMC_CONFIG4_i[7] OEEXTRADELAY bit. This half-GPMC_FCLK period provides more granularity on the nOE/RE assertion and deassertion time to ensure proper setup and hold time relative to GPMC_CLK. If enabled, OEEXTRADELAY applies to all parameters controlling nOE/nRE transitions.

OEEXTRADELAY must be used carefully, to avoid control-signal overlap between successive accesses to different chip-selects. This implies the need to program RDCYCLETIME and WRCYCLETIME to be greater than the nOE/RE signal-deassertion time, including the extra half-GPMC_FCLK-period delay.

NOTE: When the GPMC generates a read access to an address-/data-multiplexed device, it drives the address bus until nOE assertion time.

10.3.4.9.5 nWE: Write Enable Signal Control Assertion/Deassertion Time (WEONTIME / WEOFFTIME / WEEXTRADELAY)

The GPMC_CONFIG4_i[19:16] WEONTIME bit field (where i = 0 to 7) defines the nWE signal-assertion time relative to start access time. The GPMC_CONFIG4_i[28:24] WEOFFTIME bit field defines the nWE signal-deassertion time relative to start access time. These bit fields apply only to write accesses. nWE is not asserted during a read cycle.

WEONTIME can be used to control an address and byte-enable valid setup time control before nWE assertion. WEOFFTIME can be used to control an address and byte-enable valid hold time control after nWE assertion.

nWE signal transitions as controlled through WEONTIME, and WEOFFTIME can be delayed by a half-GPMC_FCLK period by enabling the GPMC_CONFIG4_i[23] WEEXTRADELAY bit. This half-GPMC_FCLK period provides more granularity on nWE assertion and deassertion time to ensure proper setup and hold time relative to GPMC_CLK. If enabled, WEEXTRADELAY applies to all parameters controlling nWE transitions.

The WEEXTRADELAY bit must be used carefully to avoid control-signal overlap between successive accesses to different chip-selects. This implies the need to program the WRCYCLETIME bit field to be greater than the nWE signal-deassertion time, including the extra half-GPMC_FCLK-period delay.

10.3.4.9.6 GPMC_CLK

GPMC_CLK is the external clock provided to the attached synchronous memory or device.

- The GPMC_CLK clock frequency is the GPMC_FCLK functional clock frequency divided by 1, 2, 3, or 4, depending on the GPMC_CONFIG1_i[1:0] GPMCFCLKDIVIDER bit field (where i = 0 to 7), with a guaranteed 50-percent duty cycle. For information about the duty cycle error, see the device data manual.
- The GPMC_CLK clock is activated only when the access in progress is defined as synchronous (read or write access).
- The GPMC_CLK clock frequency is defined by the GPMC_CONFIG1_i[26:25] CLKACTIVATIONTIME bit field (where i = 0 to 7) defines the number of GPMC_FCLK cycles from start access time to GPMC_CLK activation.
- The GPMC_CLK clock is stopped when cycle time completes and is asserted low between accesses.
- The GPMC_CLK clock is kept low when access is defined as asynchronous.

CAUTION

When the cycle time completes, the GPMC_CLK may be high because of the GPMCFCLKDIVIDER bit field. To ensure correct stoppage of the GPMC_CLK clock within the required 50-percent duty cycle, the user must extend the RDCYCLETIME or WRCYCLETIME value.

NOTE: To ensure a correct external clock cycle, the following rules must be applied:

- (RDCYCLETIME CLKACTIVATIONTIME) must be a multiple of (GPMCFCLKDIVIDER + 1).
- The PAGEBURSTACCESSTIME value must be a multiple of (GPMCFCLKDIVIDER + 1).

10.3.4.9.7 GPMC_CLK and Control Signals Setup and Hold

Control-signal transition (assertion and deassertion) setup and hold values with respect to the GPMC_CLK edge can be controlled in the following ways:

- For the GPMC_CLK signal, the GPMC_CONFIG1_i[26:25] CLKACTIVATIONTIME bit field (where i = 0 to 7) allows setup and hold control of control-signal assertion time.
- The use of a divided GPMC_CLK allows setup and hold control of the control-signal assertion and deassertion times.
- When GPMC_CLK runs at the GPMC_FCLK frequency so that GPMC_CLK edge and control-signal transitions refer to the same GPMC_FCLK edge, the control-signal transitions can be delayed by a half-GPMC_FCLK period to provide minimum setup and hold times. This half-GPMC_FCLK delay is enabled with the CSEXTRADELAY, ADVEXTRADELAY, OEEXTRADELAY, or WEEXTRADELAY parameter. This delay must be used carefully to prevent control-signal overlap between successive accesses to different chip-selects. This implies that the RDCYCLETIME and WRCYCLETIME are greater than the last control-signal deassertion time, including the extra half-GPMC_FCLK cycle.

10.3.4.9.8 Access Time (RDACCESSTIME / WRACCESSTIME)

The read/write access time durations can be programmed independently through the GPMC_CONFIG5_i[20:16] RDACCESSTIME and GPMC_CONFIG6_i[28:24] WRACCESSTIME bit fields (where i = 0 to 7). This allows nOE and GPMC data-capture timing parameters to be independent of nWE and memory device data capture timing parameters. The RDACCESSTIME and WRACCESSTIME bit fields can be set with a granularity of 1 or 2 through the GPMC_CONFIG1_i[4] TIMEPARAGRANULARITY bit.

10.3.4.9.8.1 Access Time on Read Access

In asynchronous read mode, for single and paged accesses, the GPMC_CONFIG5_i[20:16] RDACCESSTIME bit field (where i = 0 to 7) defines the number of GPMC_FCLK cycles from start access time to the GPMC_FCLK rising edge used for the first data capture. RDACCESSTIME must be programmed to the rounded greater value (in GPMC_FCLK cycles) of the read access time of the attached memory device.

In synchronous read mode, for single or burst accesses, RDACCESSTIME defines the number of GPMC_FCLK cycles from the start access time to the GPMC_FCLK rising edge corresponding to the GPMC_CLK rising edge used for the first data capture.

GPMC_CLK, which is sent to the memory device for synchronization with the GPMC controller, is internally retimed to correctly latch the returned data. The GPMC_CONFIG5_i[4:0] RDCYCLETIME bit field must be greater than RDACCESSTIME to let the GPMC latch the last return data using the internally retimed GPMC_CLK.

The external WAIT signal can be used in conjunction with RDACCESSTIME to control the effective GPMC data-capture GPMC_FCLK edge on read access in asynchronous and synchronous modes. For more information about wait monitoring, see Section 10.3.4.8.3.1, *Wait Pin Monitoring Control*.

10.3.4.9.8.2 Access Time on Write Access

In asynchronous write mode, the GPMC_CONFIG6_i[28:24] WRACCESSTIME timing parameter is not used to define the effective write access time. Instead, it is used as a wait invalid timing window and must be set to a correct value so that the gpmc_wait pin is at a valid state two GPMC_CLK cycles before WRACCESSTIME completes. For more information about wait monitoring, see Section 10.3.4.8.3.1, *Wait Pin Monitoring Control*.

In synchronous write mode , for single or burst accesses, WRACCESSTIME defines the number of GPMC_FCLK cycles from the start access time to the GPMC_CLK rising edge used by the memory device for the first data capture.

The external WAIT signal can be used in conjunction with WRACCESSTIME to control the effective memory device data-capture GPMC_CLK edge for a synchronous write access. For more information about wait monitoring, see Section 10.3.4.8.3.1, *Wait Pin Monitoring Control*.

10.3.4.9.9 Page Burst Access Time (PAGEBURSTACCESSTIME)

The GPMC_CONFIG5_i[27:24] PAGEBURSTACCESSTIME bit field (where i = 0 to 7) can be set with a granularity of 1 or 2 through theGPMC_CONFIG1_i[4] TIMEPARAGRANULARITY bit.

10.3.4.9.9.1 Page Burst Access Time on Read Access

In asynchronous page read mode, the delay between successive word captures in a page is controlled through the PAGEBURSTACCESSTIME bit field. The PAGEBURSTACCESSTIME parameter must be programmed to the rounded greater value (in GPMC_FCLK cycles) of the read access time of the attached device.

In synchronous burst read mode, the delay between successive word captures in a burst is controlled through the PAGEBURSTACCESSTIME bit field.

The external WAIT signal can be used in conjunction with PAGEBURSTACCESSTIME to control the effective GPMC data-capture GPMC_FCLK edge on read access. For more information about wait monitoring, see Section 10.3.4.8.3.1, *Wait Pin Monitoring Control*.

10.3.4.9.9.2 Page Burst Access Time on Write Access

Asynchronous page write mode is not supported. PAGEBURSTACCESSTIME is irrelevant in this case.

In synchronous burst write mode, PAGEBURSTACCESSTIME controls the delay between successive memory device word captures in a burst.

The external WAIT signal can be used in conjunction with PAGEBURSTACCESSTIME to control the effective memory device data capture GPMC_CLK edge in synchronous write mode. For more information about wait monitoring, see Section 10.3.4.8.3.1, *Wait Pin Monitoring Control*.

10.3.4.9.10 Bus Keeping Support

At the end cycle time of a read access, if no other access is pending, the GPMC drives the bus with the last data read after RDCYCLETIME completes to prevent bus floating and reduce power consumption.

After a write access, if no other access is pending, the GPMC keeps driving the data bus after WRCYCLETIME completes with the same data to prevent bus floating and power consumption.

10.3.4.10 NOR Access Description

For each chip-select configuration, the read access can be specified as asynchronous or synchronous access through the GPMC_CONFIG1_i[29] READTYPE bit (where i = 0 to 7). For each chip-select configuration, the write access can be specified as synchronous or asynchronous access through the GPMC_CONFIG1_i[27] WRITETYPE bit where (i = 0 to 7).

Asynchronous and synchronous read and write access time and related control signals are controlled through timing parameters that refer to GPMC_FCLK. The primary difference of synchronous mode is the availability of a configurable clock interface (GPMC_CLK) to control the external device. Synchronous mode also affects data-capture and wait-pin monitoring schemes in read access.

For more information about asynchronous and synchronous access, see the descriptions of GPMC_CLK, RdAccessTime, WrAccessTime, and wait pin monitoring.

For more information about timing-parameter settings, see the sample timing diagrams in this chapter.

NOTE: The address bus and nBE[1:0] are fixed for the duration of a synchronous burst read access, but they are updated for each beat of an asynchronous page-read access.

10.3.4.10.1 Asynchronous Access Description

This section describes:

- Asynchronous single-read operation on an address/data multiplexed device
- Asynchronous single write operation on an address/data-multiplexed device
- Asynchronous single read operation on an AAD-multiplexed device
- Asynchronous single write operation on an AAD-multiplexed device
- Asynchronous multiple (page) read operation on a non-multiplexed device

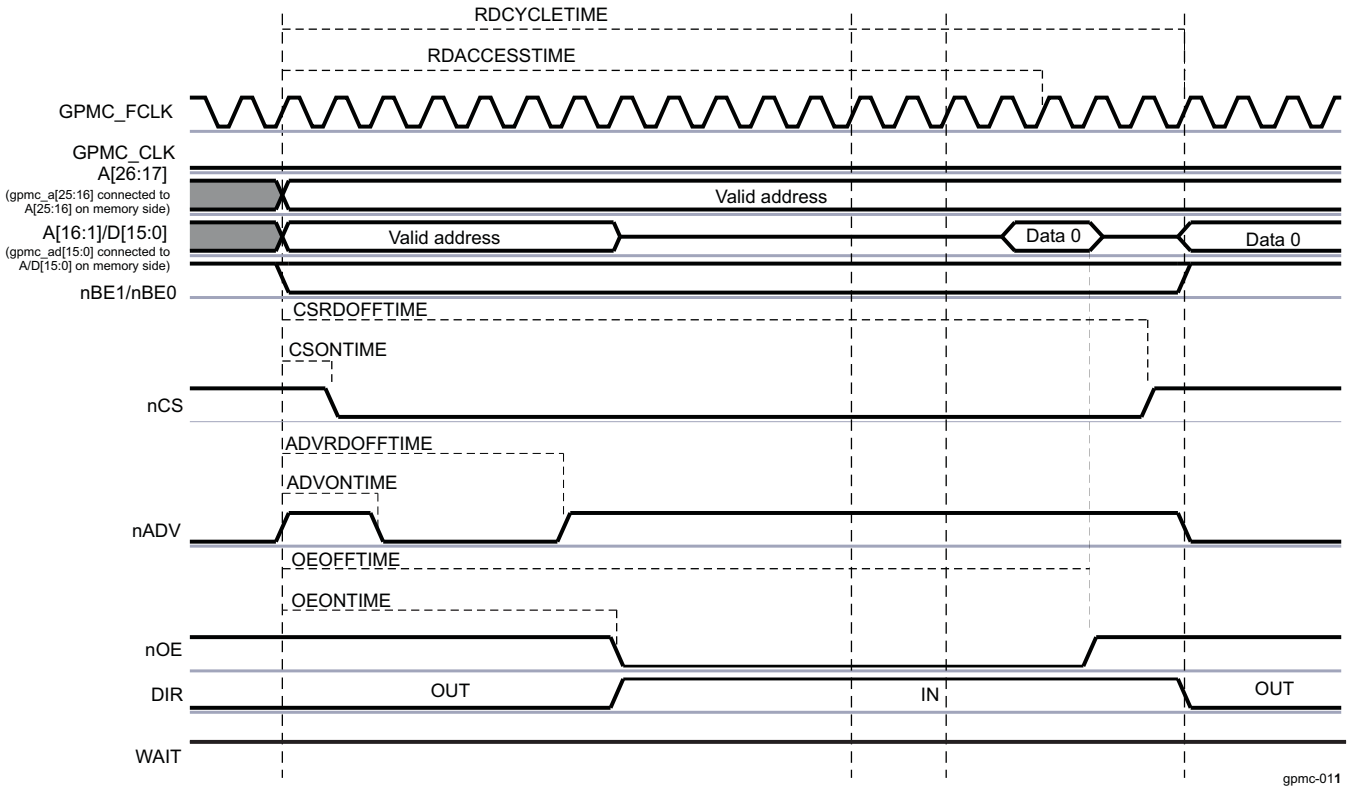
In asynchronous operations GPMC_CLK is not provided outside the GPMC and is kept low.

10.3.4.10.1.1 Access on Address/Data Multiplexed Devices

10.3.4.10.1.1.1 Asynchronous Single-Read Operation on an Address/Data Multiplexed Device

Figure 10-24 shows an asynchronous single read operation on an address/data-multiplexed device.

Figure 10-24. Asynchronous Single Read on an Address/Data-Multiplexed Device



For formulas to calculate timing parameters, see Section 10.3.5.6.1, *GPMC Timing Parameters Formulas*.

Table 10-400 lists the timing bit fields to set up to configure the GPMC in asynchronous single-read mode.

When the GPMC generates a read access to an address/data-multiplexed device, it drives the address bus until nOE assertion time. For more information, see Section 10.3.4.8.2.3, *Address/Data-Multiplexing Interface*.

Address bits (A[16:1] from a GPMC perspective, A[15:0] from an external device perspective) are placed on the address/data bus, and the remaining address bits gpmc_a[27:16] are placed on the address bus. The address phase ends at nOE assertion, when the DIR signal goes from OUT to IN.

- Chip-select signal nCS:
  - nCS assertion time is controlled by the GPMC_CONFIG2_i[3:0] CSONTIME bit field. It controls the address setup time to nCS assertion.
  - nCS deassertion time is controlled by the GPMC_CONFIG2_i[12:8] CSRDOFFTIME bit field. It controls the address hold time from nCS deassertion.
- Address valid signal nADV:
  - nADV assertion time is controlled by the GPMC_CONFIG3_i[3:0] ADVONTIME bit field.
  - nADV deassertion time is controlled by the GPMC_CONFIG3_i[12:8] ADVRDOFFTIME bit field.
- Output enable signal nOE:
  - nOE assertion indicates a read cycle.
  - nOE assertion time is controlled by the GPMC_CONFIG4_i[3:0] OEONTIME bit field.
  - nOE deassertion time is controlled by the GPMC_CONFIG4_i[12:8] OEOFFTIME bit field.

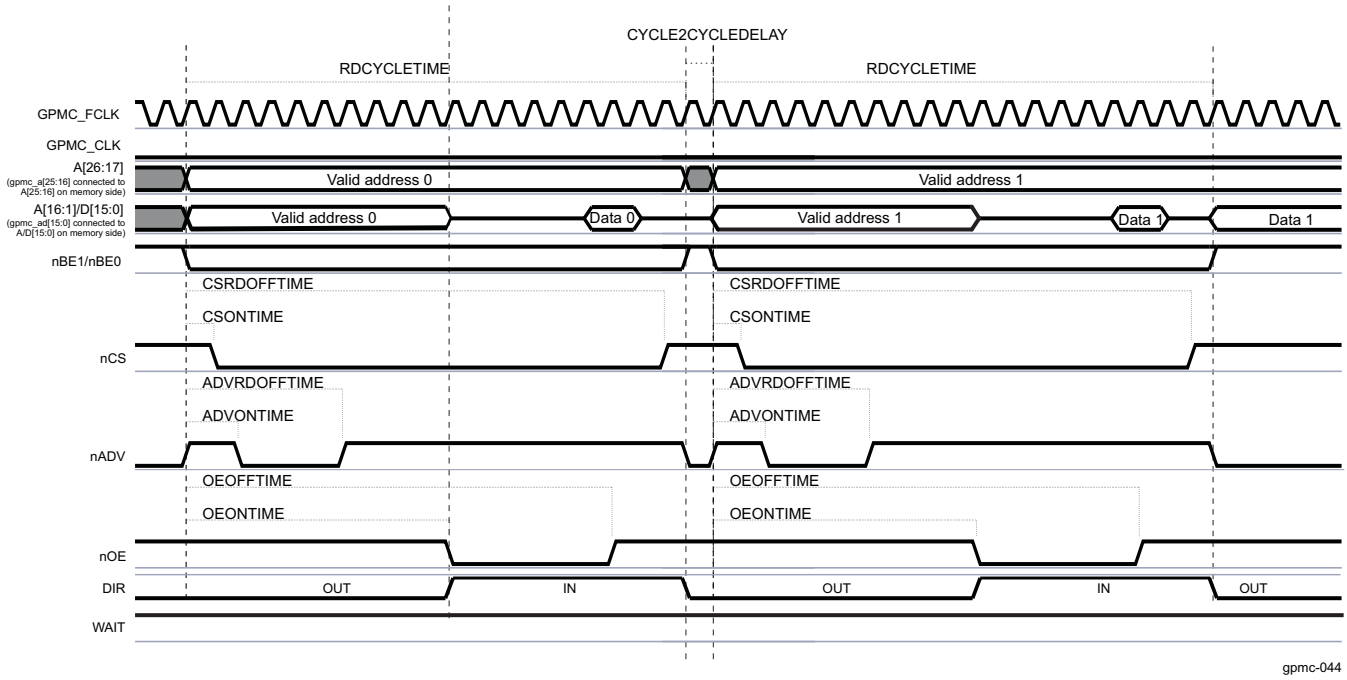
- Read data is latched when RDACCESSTIME completes. Access time is defined in the GPMC_CONFIG5_i[20:16] RDACCESSTIME bit field.
- Direction signal DIR: DIR goes from OUT to IN at the same time that nOE is asserted.
- The end of the access is defined by the GPMC_CONFIG5_i[4:0] RDCYCLETIME parameter.

In the GPMC, when a 16-bit wide device is attached to the controller, a 32-bit word write access is split into two 16-bit word write accesses. For more information about GPMC access size and type adaptation, see Section 10.3.4.10.5, *System Burst Versus External Device Burst Support*.

Between two successive accesses, if an nCS pulse is needed:

- The GPMC_CONFIG6_i[11:8] CYCLE2CYCLEDELAY bit field can be programmed with the GPMC_CONFIG6_i[7] CYCLE2CYCLESAMECSEN bit enabled.
- The CSWROFFTIME and CSONTIME parameters also allow a chip-select pulse, but this affects all other types of access.

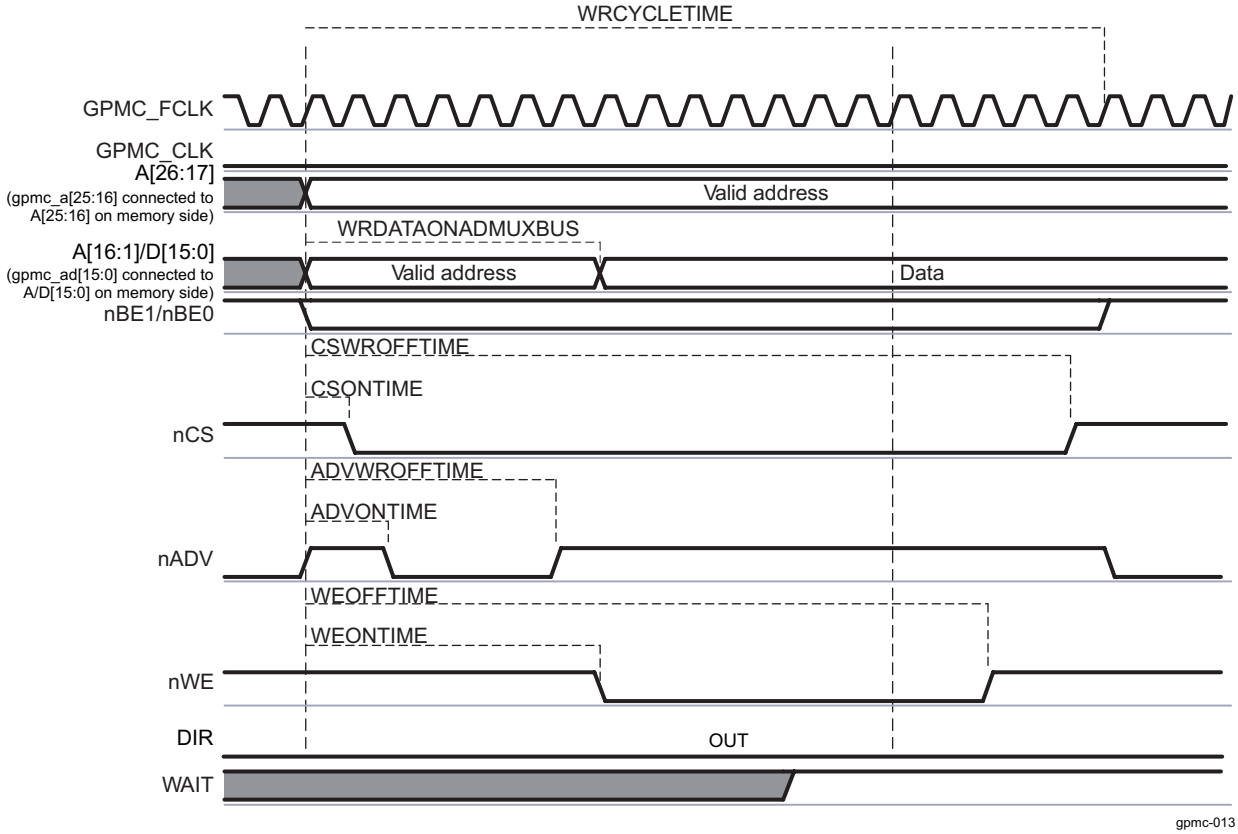
Figure 10-25. Two Asynchronous Single-Read Accesses on an Address/Data-Multiplexed Device (32-Bit Read Split Into 2 x 16-Bit Read)



10.3.4.10.1.1.2 Asynchronous Single-Write Operation on an Address/Data-Multiplexed Device

Figure 10-26 shows an asynchronous single-write operation on an address/data-multiplexed device.

Figure 10-26. Asynchronous Single-Write on an Address/Data-Multiplexed Device



For formulas to calculate timing parameters, see Section 10.3.5.6.1, *GPMC Timing Parameters Formulas*.

Table 10-400 lists the timing bit fields to set up to configure the GPMC in asynchronous single-write mode.

When the GPMC generates a write access to an address/data-multiplexed device, it drives the address bus until nWE assertion time. For more information, see Section 10.3.4.8.2.3, *Address/Data-Multiplexing Interface*.

The nCS and nADV signals are controlled in the same way as for a asynchronous single-read operation on an address/data-multiplexed device.

- Write enable signal nWE:
  - nWE assertion indicates a write cycle.
  - nWE assertion time is controlled by the GPMC_CONFIG4_i[19:16] WEONTIME bit field.
  - nWE deassertion time is controlled by the GPMC_CONFIG4_i[28:24] WEOFFTIME bit field.
- Direction signal DIR: DIR signal is OUT during the entire access.
- The end of the access is defined by the GPMC_CONFIG5_i[12:8] WRCYCLETIME parameter.

Address bits A[16:1] (GPMC point of view) are placed on the address/data bus at the start of cycle time, and the remaining address bits A[26:17] are placed on the address bus.

Data is driven on the address/data bus at a GPMC_CONFIG6_i[19:16] WRDATAONADMUXBUS time.

NOTE: Write multiple access in asynchronous mode is not supported. If WRITEMULTIPLE is enabled with WRITETYPE as asynchronous, the GPMC processes single asynchronous accesses.

After a write operation, if no other access (read or write) is pending, the data bus keeps its previous value. See Section 10.3.4.9.10, *Bus Keeping Support*.

10.3.4.10.1.1.3 Asynchronous Multiple (Page) Write Operation on an Address/Data-Multiplexed Device

Write multiple (page) access in asynchronous mode is not supported for address/data-multiplexed devices.

If the GPMC_CONFIG1_i[28] WRITEMULTIPLE bit is enabled (0x1) with the GPMC_CONFIG1_i[27] WRITETYPE bit as asynchronous (0x0), the GPMC processes single asynchronous accesses.

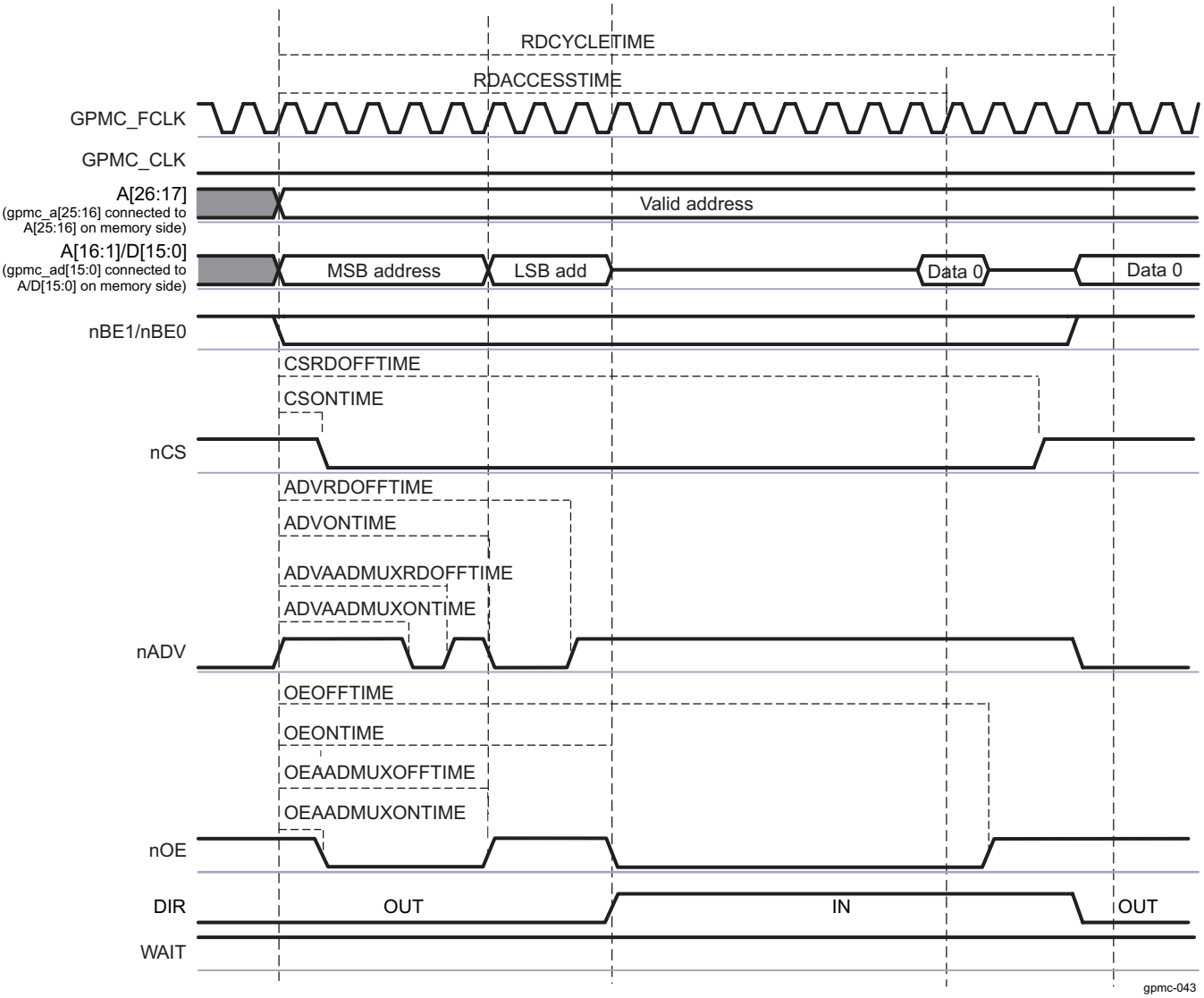
For accesses on nonmultiplexed devices, see Section 10.3.4.10.3, *Asynchronous and Synchronous Accesses in Nonmultiplexed Mode*.

10.3.4.10.1.2 Access on Address/Address/Data-Multiplexed Devices

10.3.4.10.1.2.1 Asynchronous Single Read Operation on an AAD-Multiplexed Device

Figure 10-27 shows an asynchronous single-read operation on an AAD-multiplexed device.

Figure 10-27. Asynchronous Single Read on an AAD-Multiplexed Device



For formulas to calculate timing parameters, see Section 10.3.5.6.1, *GPMC Timing Parameters Formulas*.

Table 10-400 lists the timing bit fields to set up to configure the GPMC in asynchronous single write mode.

When the GPMC generates a read access to an AAD-multiplexed device, all address bits are driven onto the address/data bus in two separate phases. The first phase is used for the MSB address and is qualified with nOE driven low. The first address phase ends at the first nOE deassertion time. The second phase for LSB address is qualified with nOE driven high. The second address phase ends at the second nOE assertion time, when the DIR signal goes from OUT to IN.

The nCS and DIR signals are controlled in the same way as for an asynchronous single-read operation on an address/data-multiplexed device.

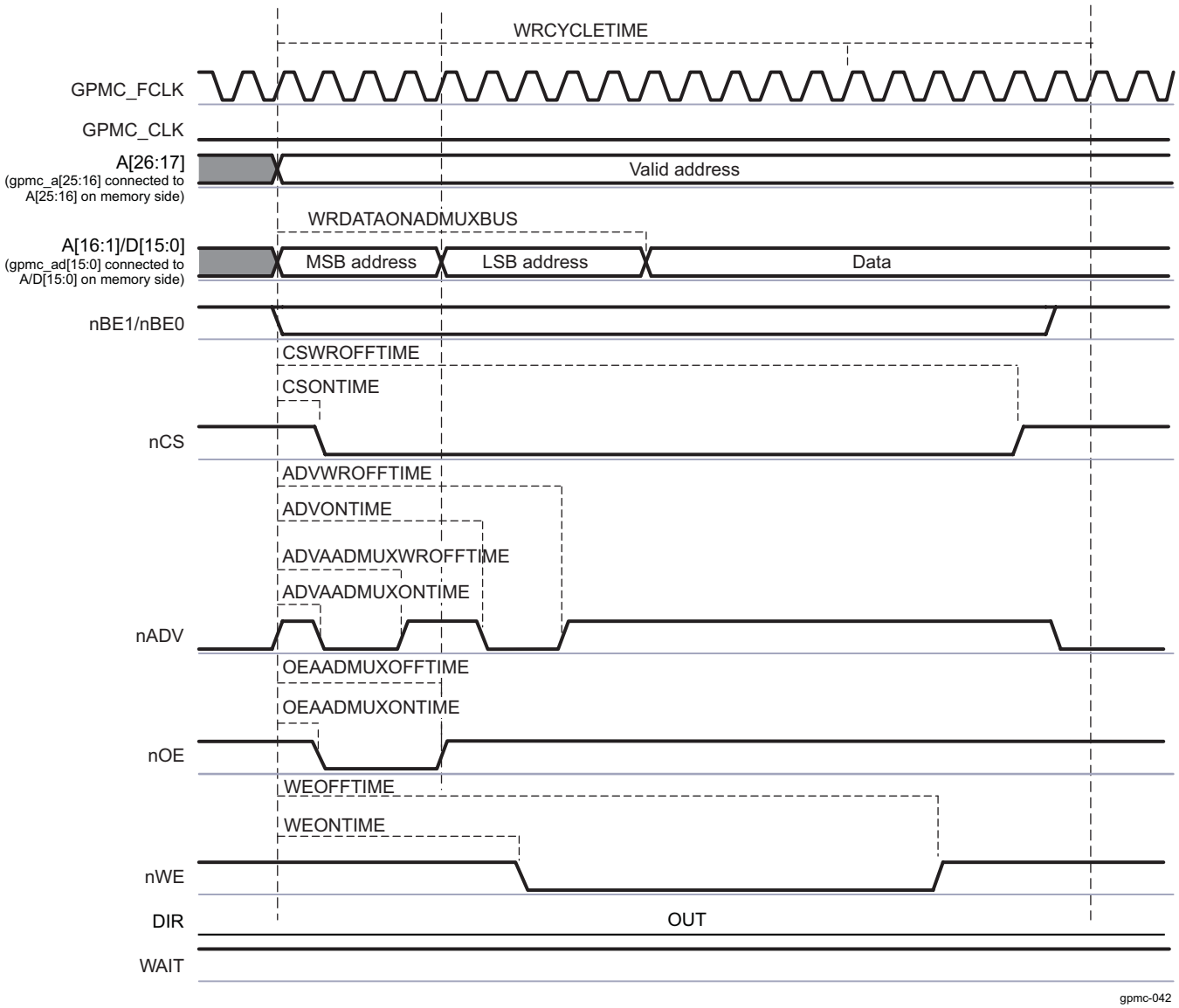
- Address valid signal nADV. nADV is asserted and deasserted twice during a read transaction:
  - nADV first assertion time is controlled by the GPMC_CONFIG3_i[6:4] ADVAADMUXONTIME bit field.

  - nADV first deassertion time is controlled by the GPMC_CONFIG3_i[26:24] ADVAADMUXRDOFFTIME bit field.
  - nADV second assertion time is controlled by the GPMC_CONFIG3_i[3:0] ADVONTIME bit field.
  - nADV second deassertion time is controlled by the GPMC_CONFIG3_i[12:8] ADVRDOFFTIME bit field.
- Output Enable signal nOE. nOE is asserted and deasserted twice during a read transaction (nOE second assertion indicates a read cycle):
  - nOE first assertion time is controlled by the GPMC_CONFIG4_i[6:4] OEAADMUXONTIME bit field.
  - nOE first deassertion time is controlled by the GPMC_CONFIG3_i[15:13] OEAADMUXOFFTIME bit field.
  - nOE second assertion time is controlled by the GPMC_CONFIG4_i[3:0] OEONTIME bit field.
  - nOE second deassertion time is controlled by the GPMC_CONFIG4_i[12:8] OEOFFTIME bit field.

10.3.4.10.1.2.2 Asynchronous Single-Write Operation on an AAD-Multiplexed Device

Figure 10-28 shows an asynchronous single-write operation on an AAD-multiplexed device.

Figure 10-28. Asynchronous Single Write on an AAD-Multiplexed Device



For formulas to calculate timing parameters, see Section 10.3.5.6.1, *GPMC Timing Parameters Formulas*.

Table 10-400 lists the timing bit fields to set up to configure the GPMC in asynchronous single-write mode.

When the GPMC generates a write access to an AAD-multiplexed device, all address bits are driven onto the address/data bus in two separate phases. The first phase is used for the MSB address and is qualified with nOE driven low. The second phase for LSB address is qualified with nOE driven high. The address phase ends at nWE assertion time.

The nCS, nWE, and DIR signals are controlled in the same way as for an asynchronous single-write operation on an address/data-multiplexed device. See Table 10-391.

- Address valid signal nADV is asserted and deasserted twice during a write transaction:
  - nADV first assertion time is controlled by the GPMC_CONFIG3_i[6:4] ADVAADMUXONTIME bit field.
  - nADV first deassertion time is controlled by the GPMC_CONFIG3_i[30:28] ADVAADMUXWROFFTIME bit field.
  - nADV second assertion time is controlled by the GPMC_CONFIG3_i[3:0] ADVONTIME bit field.
  - nADV second deassertion time is controlled by the GPMC_CONFIG3_i[20:16] ADVWROFFTIME bit field.
- Output enable signal nOE is asserted during the address phase of a write transaction:
  - nOE assertion time is controlled by the GPMC_CONFIG4_i[6:4] OEAADMUXONTIME bit field.
  - nOE deassertion time is controlled by the GPMC_CONFIG3_i[15:13] OEAADMUXOFFTIME bit field.

The address bits for the first address phase are driven onto the data bus until nOE deassertion. Data is driven onto the address/data bus at the clock edge defined by the GPMC_CONFIG6_i[19:16] WRDATAONADMUXBUS parameter.

10.3.4.10.1.2.3 Asynchronous Multiple (Page) Read Operation on an AAD-Multiplexed Device

Write multiple (page) access in asynchronous mode is not supported for AAD-multiplexed devices.

If the GPMC_CONFIG1_i[28] WRITEMULTIPLE bit is enabled (0x1) with the GPMC_CONFIG1_i[27] WRITETYPE bit as asynchronous (0x0), the GPMC processes single asynchronous accesses.

For accesses on nonmultiplexed devices, see Section 10.3.4.10.3, *Asynchronous and Synchronous Accessed in Nonmultiplexed Mode*.

10.3.4.10.2 Synchronous Access Description

This section describes read and write synchronous accesses on address/data-multiplexed devices. All information in this section can be applied to any type of memory (nonmultiplexed, address and data-multiplexed, or AAD-multiplexed) with the difference limited to the address phase. For accesses on nonmultiplexed devices, see Section 10.3.4.10.3, *Asynchronous and Synchronous Accessed in Nonmultiplexed Mode*.

In synchronous operations:

- The GPMC_CLK clock is provided outside the GPMC when accessing the memory device.
- The GPMC_CLK clock is derived from the GPMC_FCLK clock using the GPMC_CONFIG1_i[1:0] GPMCFCLKDIVIDER bit field. In the following section i stands for the chip-select number, i = 0 to 7.
- The GPMC_CONFIG1_i[26:25] CLKACTIVATIONTIME bit field specifies that the GPMC_CLK is provided outside the GPMC for 0 to 2 GPMC_FCLK cycles after start access time until RDCYCLETIME or WRCYCLETIME completes.

10.3.4.10.2.1 Synchronous Single Read

Figure 10-29 and Figure 10-30 show a synchronous single-read operation with GPMCFCLKDIVIDER equal to 0 and 1, respectively.

Figure 10-29. Synchronous Single Read (GPMCFCLKDIVIDER = 0)

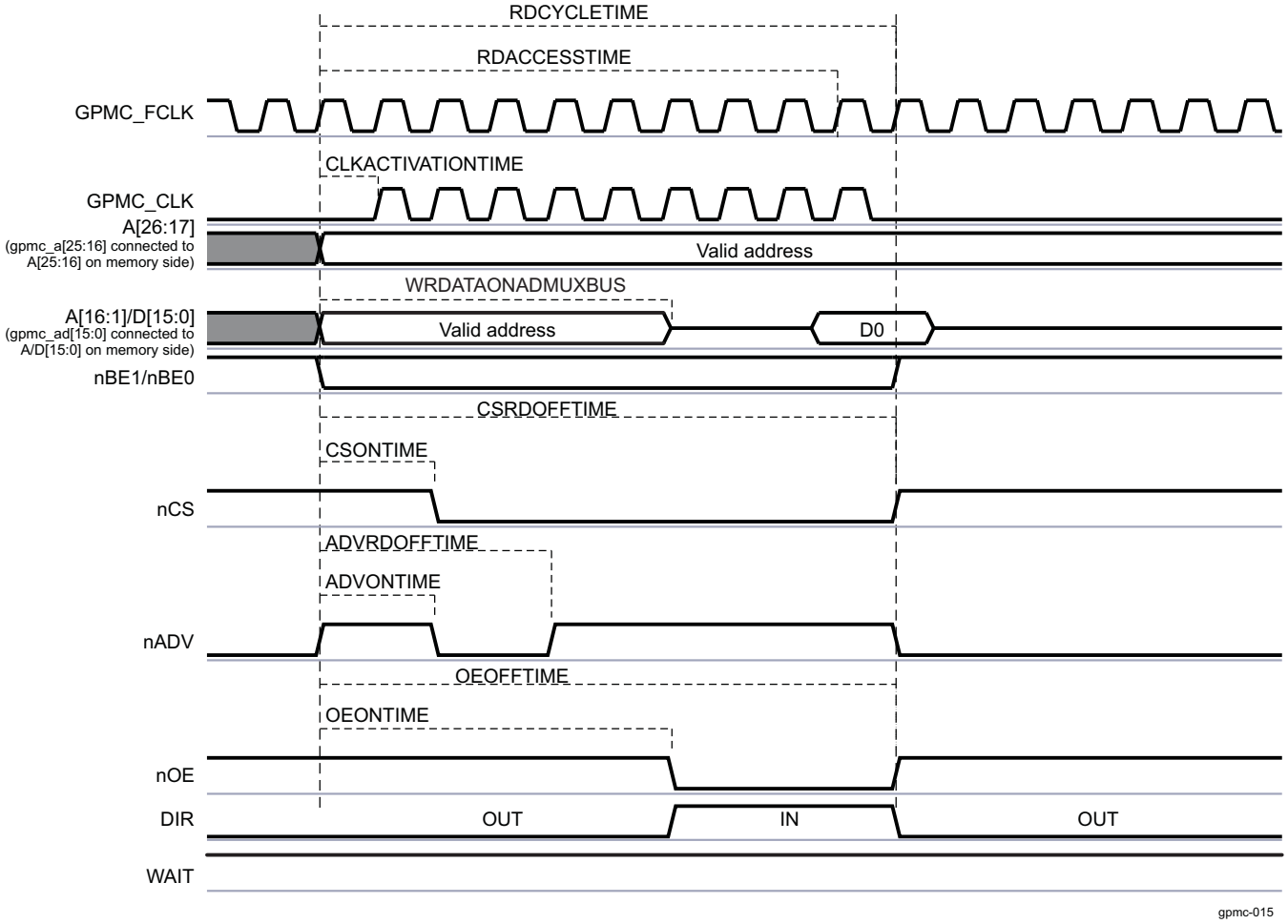
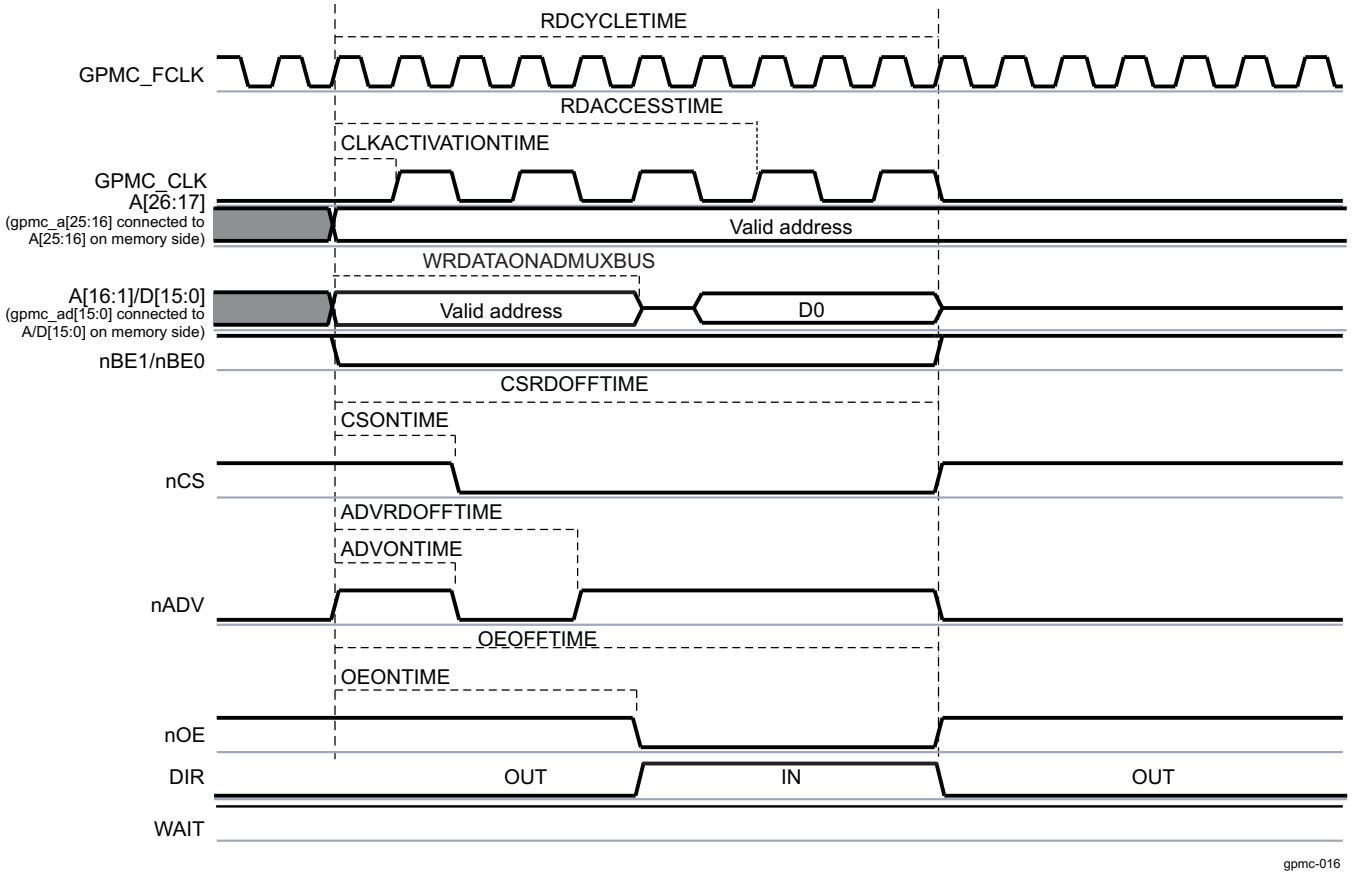


Figure 10-30. Synchronous Single Read (GPMCFCLKDIVIDER = 1)



For formulas to calculate timing parameters, see Section 10.3.5.6.1, *GPMC Timing Parameters Formulas*.

Table 10-400 lists the timing bit fields to set up to configure the GPMC in asynchronous single-read mode.

When the GPMC generates a read access to an address/data-multiplexed device, it drives the address bus until nOE assertion time. For more information, see Section 10.3.4.8.2.3, *Address/Data-Multiplexing Interface*.

- Chip-select signal nCS:
  - nCS assertion time is controlled by the GPMC_CONFIG2_i[3:0] CSONTIME bit field and ensures address setup time to nCS assertion.
  - nCS deassertion time is controlled by the GPMC_CONFIG2_i[12:8] CSRDOFFTIME bit field and ensures address hold time to nCS deassertion.
- Address valid signal nADV:
  - nADV assertion time is controlled by the GPMC_CONFIG3_i[3:0] ADVONTIME bit field.
  - nADV deassertion time is controlled by the GPMC_CONFIG3_i[12:8] ADVRDOFFTIME bit field.
- Output enable signal nOE:
  - nOE assertion indicates a read cycle.
  - nOE assertion time is controlled by the GPMC_CONFIG4_i[3:0] OEONTIME bit field.
  - nOE deassertion time is controlled by the GPMC_CONFIG4_i[12:8] OEOFFTIME bit field.
- Initial latency for the first read data is controlled by GPMC_CONFIG5_i[20:16] RDACCESSTIME bit field or by monitoring the WAIT signal.
- Total access time (the GPMC_CONFIG5_i[4:0] RDCYCLETIME bit field) corresponds to RDACCESSTIME plus the address hold time from nCS deassertion, plus time from RDACCESSTIME to CSRDOFFTIME.

- Direction signal DIR: DIR goes from OUT to IN at the same time as nOE assertion.

When the GPMC generates a write access to an AAD-multiplexed device, all address bits are driven onto the address/data bus in two separate phases. The first phase is used for the MSB address and is qualified with nOE driven low. The second phase for LSB address is qualified with nOE driven high. The address phase ends at nWE assertion time.

The nCS and DIR signals are controlled in the same way as for a synchronous single-read operation on an address/data-multiplexed device.

- Address valid signal nADV is asserted and deasserted twice during a read transaction:
  - nADV first assertion time is controlled by the GPMC_CONFIG3_i[6:4] ADVAADMUXONTIME bit field.
  - nADV first deassertion time is controlled by the GPMC_CONFIG3_i[26:24] ADVAADMUXRDOFFTIME bit field.
  - nADV second assertion time is controlled by the GPMC_CONFIG3_i[3:0] ADVONTIME bit field.
  - nADV second deassertion time is controlled by the GPMC_CONFIG3_i[12:8] ADVRDOFFTIME bit field.
- Output Enable signal nOE is asserted and deasserted twice during a read transaction (nOE second assertion indicates a read cycle):
  - nOE first assertion time is controlled by the GPMC_CONFIG4_i[6:4] OEAADMUXONTIME bit field.
  - nOE first deassertion time is controlled by the GPMC_CONFIG3_i[15:13] OEAADMUXOFFTIME bit field.
  - nOE second assertion time is controlled by the GPMC_CONFIG4_i[3:0] OEONTIME bit field.
  - nOE second deassertion time is controlled by the GPMC_CONFIG4_i[12:8] OEOFFTIME bit field.

After a read operation, if no other access (read or write) is pending, the data bus is driven with the previous read value. See Section 10.3.4.9.10, *Bus Keeping Support*.

10.3.4.10.2.2 Synchronous Multiple (Burst) Read (4-, 8-, 16-Word16 Burst With Wraparound Capability)

Figure 10-31 and Figure 10-32 show a synchronous multiple-read operation with GPMCFCLKDivider equal to 0 and 1, respectively.

Figure 10-31. Synchronous Multiple (Burst) Read (GPMCFCLKDIVIDER = 0)

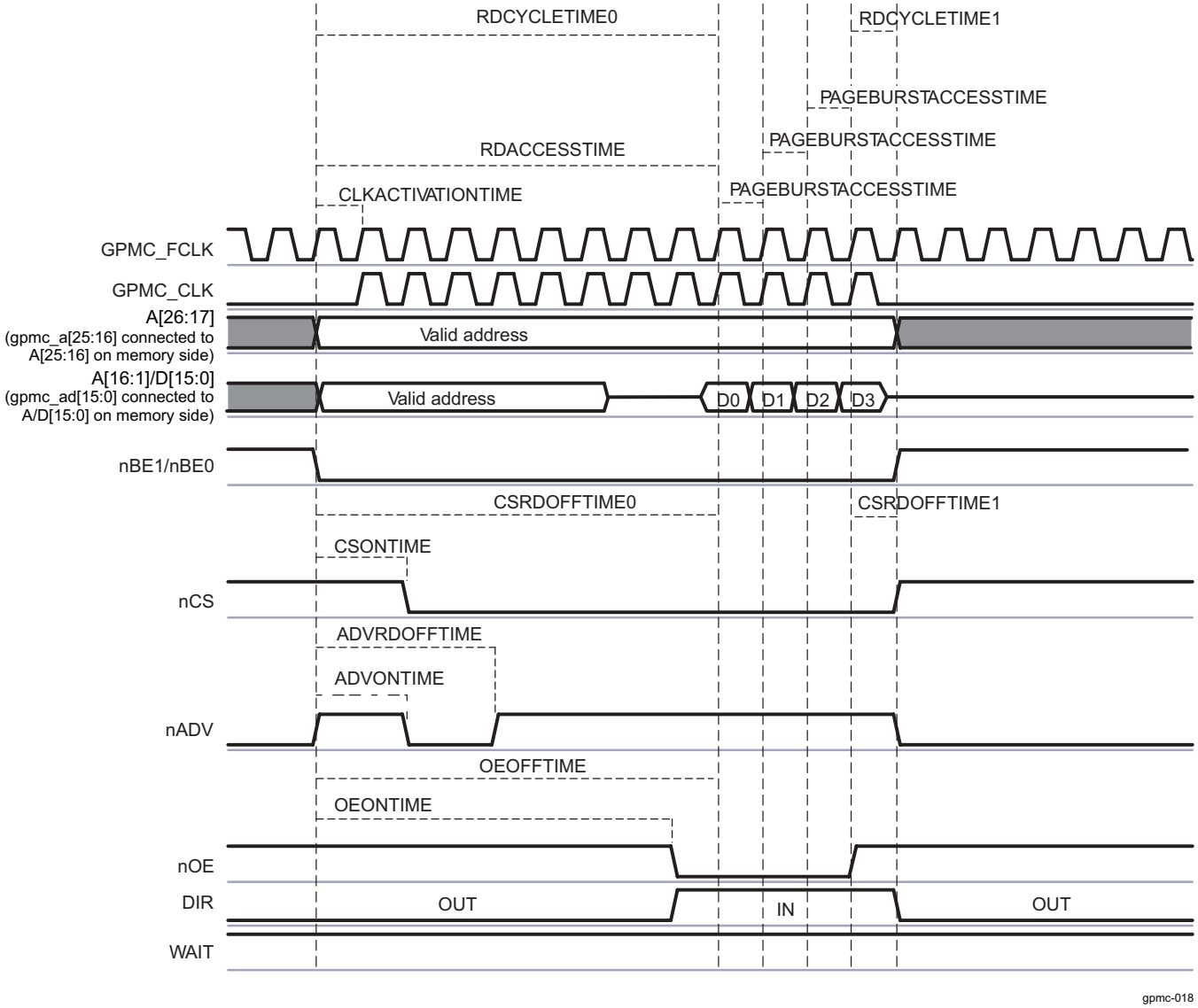
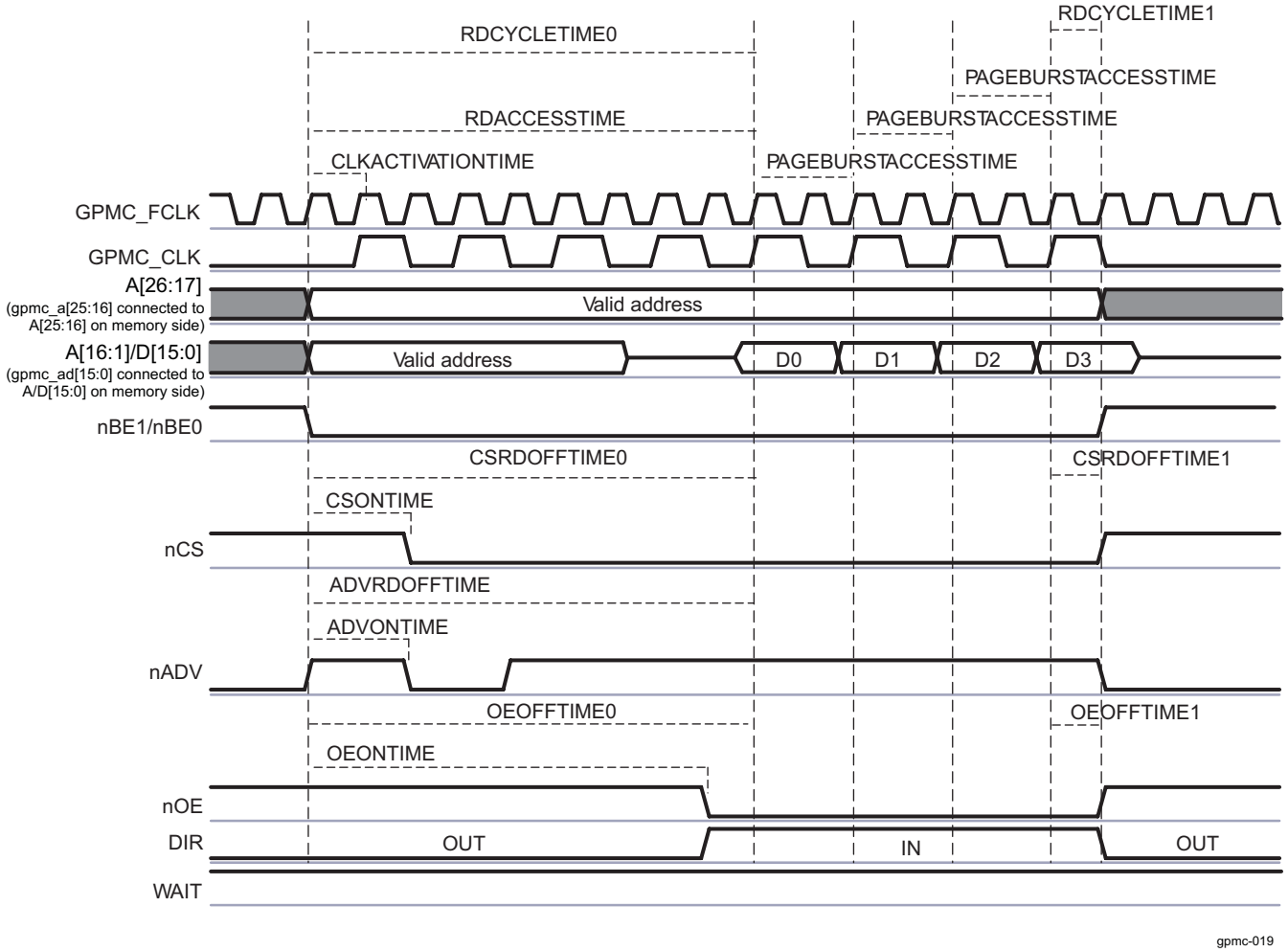


Figure 10-32. Synchronous Multiple (Burst) Read (GPMCFCLKDIVIDER = 1)



When the GPMC_CONFIG5_i[20:16] RDACCESSTIME bit field completes, control-signal timings are frozen during the multiple data transactions, corresponding to the GPMC_CONFIG5_i[27:24] PAGEBURSTACCESSTIME bit field multiplied by the number of remaining data transactions.

The nCS, nADV, nOE, and DIR signals are controlled in the same way as for a synchronous single-read operation. See Table 10-386.

Initial latency for the first read data is controlled by RDACCESSTIME or by monitoring the WAIT signal. Successive read data are provided by the memory device every one or two GPMC_CLK cycles. The PAGEBURSTACCESSTIME parameter must be set accordingly with the GPMC_CONFIG1_i[1:0] GPMCFCLKDIVIDER bit field and the memory-device internal configuration. Depending on the device page length, the GPMC checks the device page crossing during a new burst request and purposely inserts initial latency (of RDACCESSTIME) when required.

Total access time GPMC_CONFIG5_i[4:0] RDCYCLETIME corresponds to RDACCESSTIME plus the address hold time from nCS deassertion. In Figure 10-32, the programmed value of RDCYCLETIME equals RDCYCLETIME0 + RDCYCLETIME1.

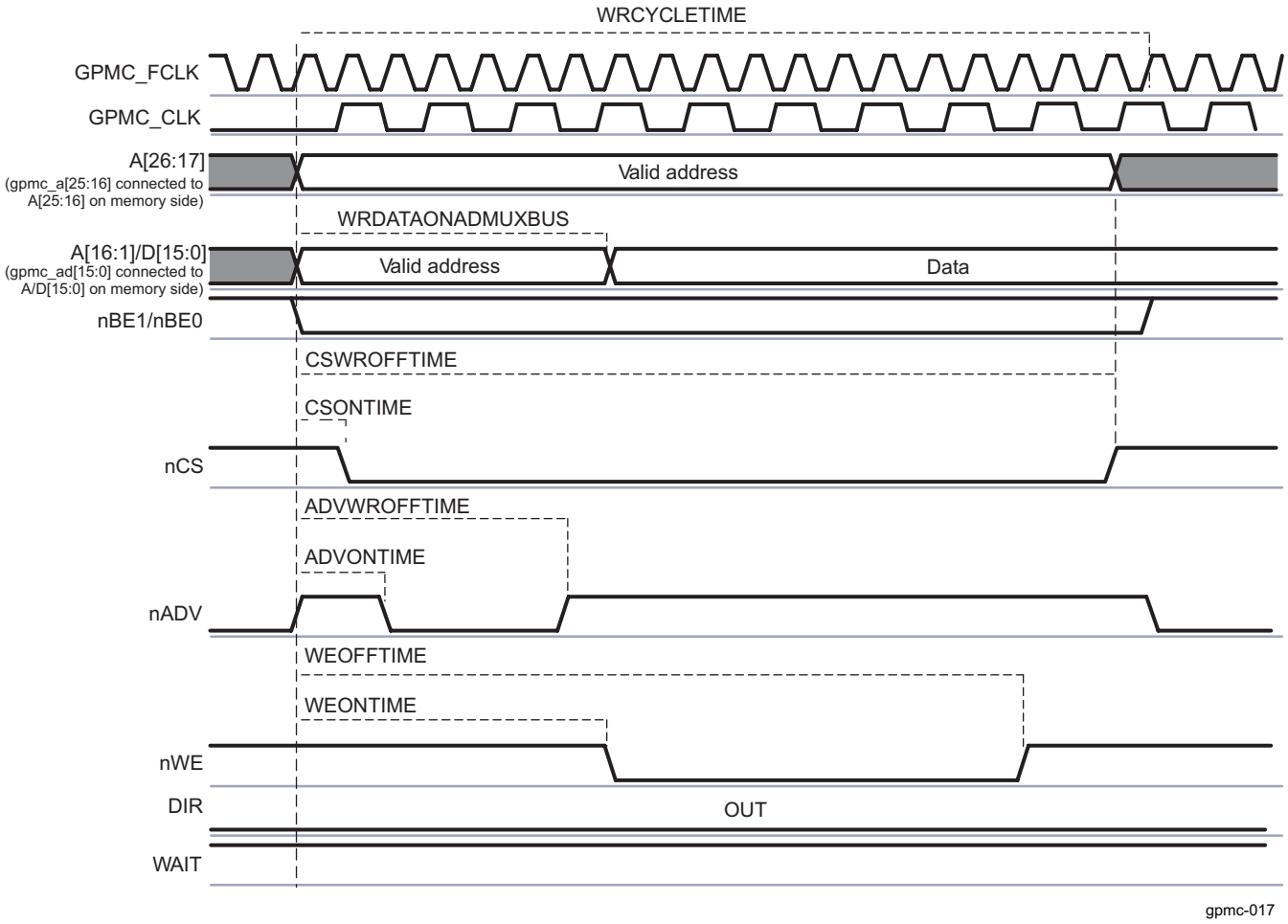
After a read operation, if no other access (read or write) is pending, the data bus is driven with the previous read value. See Section 10.3.4.9.10, *Bus Keeping Support*.

Burst wraparound is enabled through the GPMC_CONFIG1_i[31] WRAPBURST bit and allows a 4-, 8-, or 16-Word16 linear burst access to wrap within its burst-length boundary through the GPMC_CONFIG1_i[24:23] ATTACHEDDEVICEPAGELENGTH bit field.

10.3.4.10.2.3 Synchronous Single Write

Burst write mode is used for synchronous single or burst accesses.

Figure 10-33. Synchronous Single Write on an Address/Data-Multiplexed Device



When the GPMC generates a write access to an address/data-multiplexed device, it drives the data bus (with address bits A[16:1]) until the GPMC_CONFIG6_i[19:16] WRDATAONADMUXBUS bit field time. The first data of the burst is driven on the address/data bus at WRDATAONADMUXBUS time.

10.3.4.10.2.4 Synchronous Multiple (Burst) Write

Synchronous burst write mode provides synchronous single or consecutive accesses.

Figure 10-34 shows a synchronous burst write access when the chip-select is configured in address/data-multiplexed mode.

Figure 10-34. Synchronous Multiple Write (Burst Write) in Address/Data-Multiplexed Mode

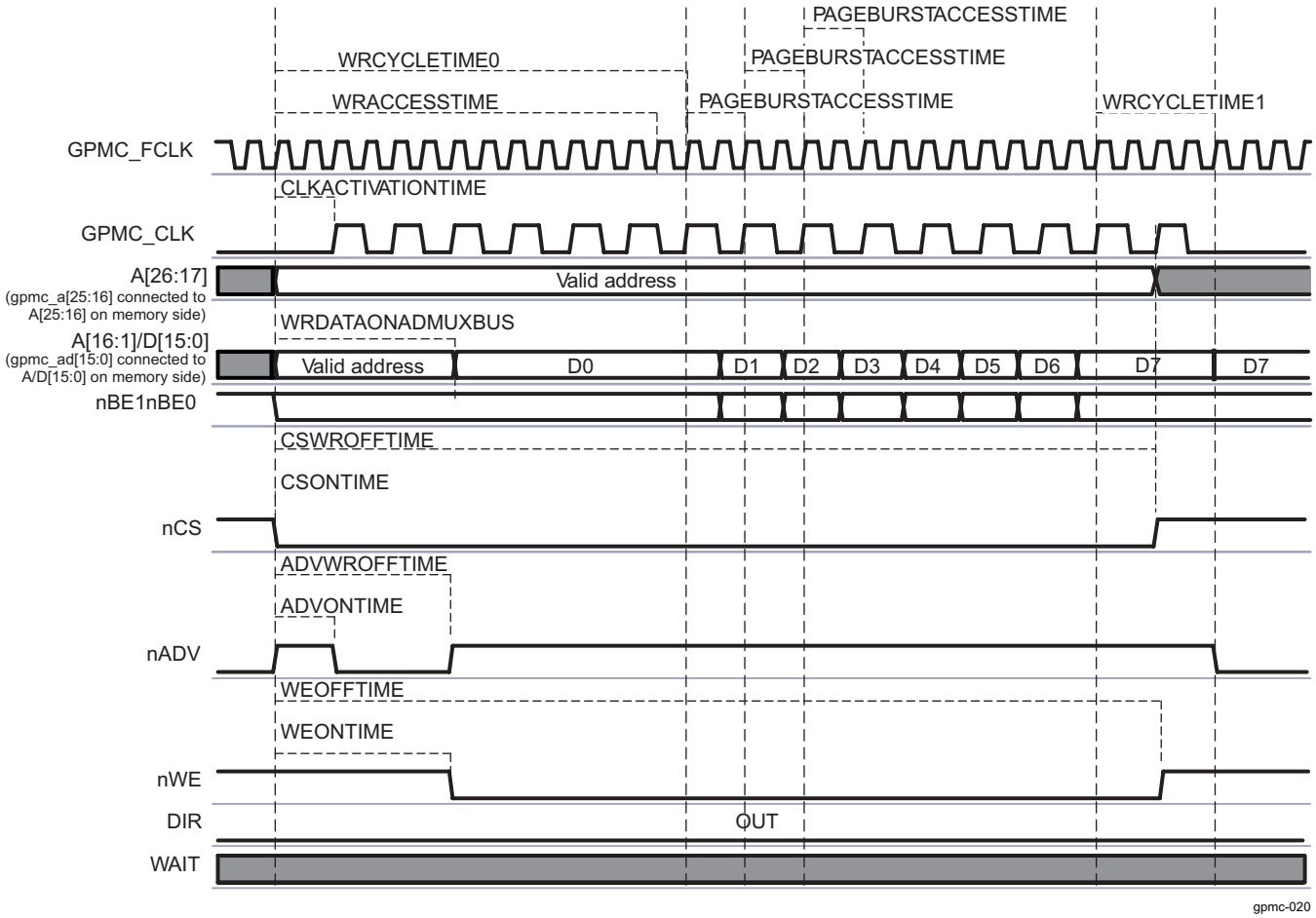
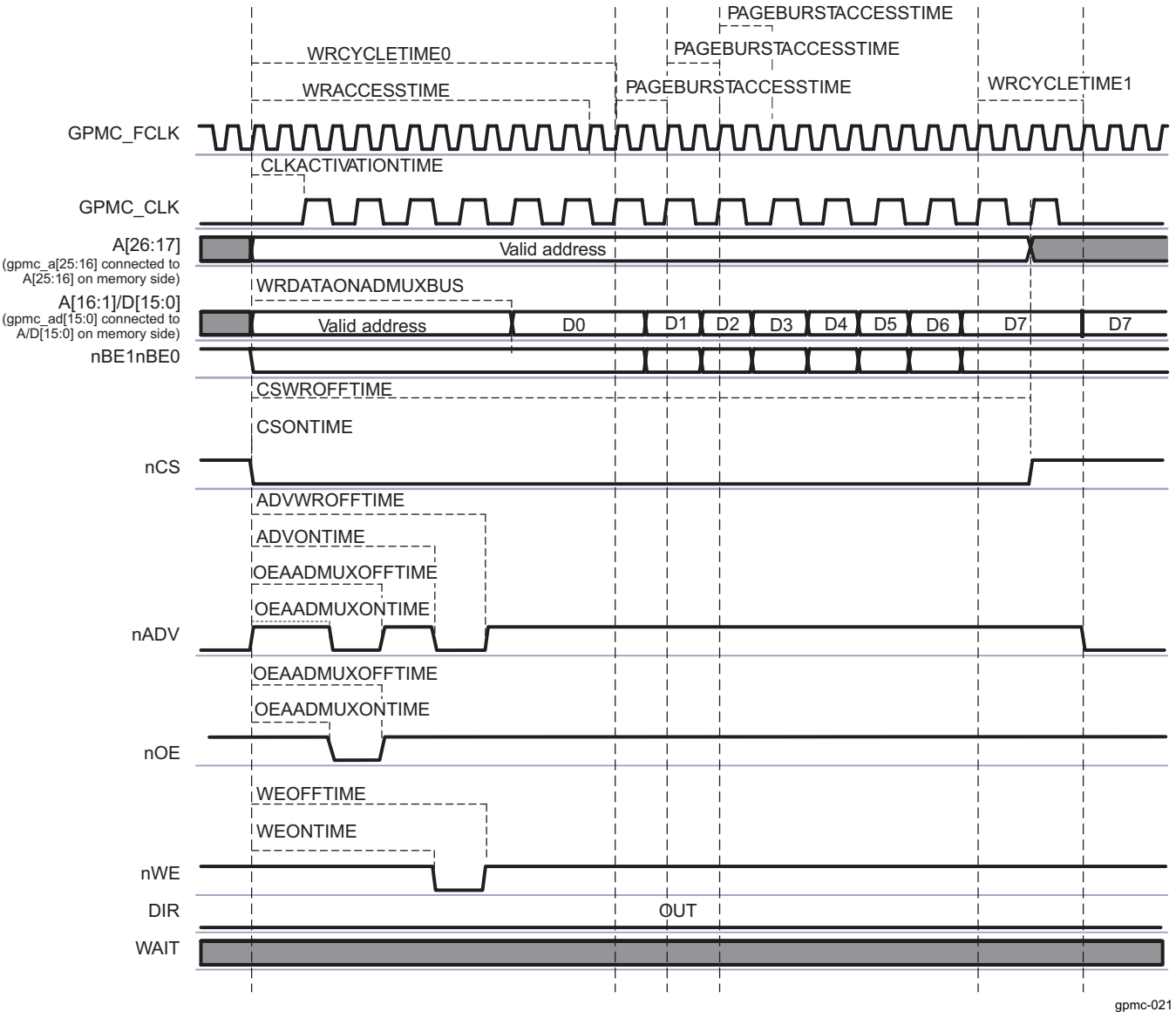


Figure 10-35 shows the same synchronous burst write access when the chip-select is configured in address/address/data-multiplexed (AAD-multiplexed) mode.

Figure 10-35. Synchronous Multiple Write (Burst Write) in Address/Address/Data-Multiplexed Mode



The first data of the burst is driven on the A/D bus at the GPMC_CONFIG6_i[19:16] WRDATAONADMUXBUS bit field.

When WRACCESSTIME completes, control-signal timings are frozen during the multiple data transactions, corresponding to the GPMC_CONFIG5_i[27:24] PAGEBURSTACCESSTIME bit field multiplied by the number of remaining data transactions.

When the GPMC generates a read access to an address/data-multiplexed device, it drives the address bus until nOE assertion time. For more information, see Section 10.3.4.8.2.3, *Address/Data-Multiplexing Interface*.

- Chip-select signal nCS:
  - nCS assertion time is controlled by the GPMC_CONFIG2_i[3:0] CSONTIME bit field (where i = 0 to 7) and ensures address setup time to nCS assertion.
  - nCS deassertion time controlled by the GPMC_CONFIG2_i[20:16] CSWROFFTIME bit field and ensures address hold time to nCS deassertion.
- Address valid signal nADV:
  - nADV assertion time is controlled by the GPMC_CONFIG3_i[3:0] ADVONTIME bit field.

  - nADV deassertion time is controlled by the GPMC_CONFIG3_i[20:16] ADVWROFFTIME bit field.
- Write enable signal nWE:
  - nWE assertion indicates a read cycle.
  - nWE assertion time is controlled by the GPMC_CONFIG4_i[19:16] WEONTIME bit field.
  - nWE deassertion time is controlled by the GPMC_CONFIG4_i[28:24] WEOFFTIME bit field.

NOTE: The nWE falling edge must not be used to control the time when the burst first data is driven in the address/data bus, because some new devices require the nWE signal to be low during the address phase.

- Direction signal DIR is OUT during the entire access.

When the GPMC generates a write access to an AAD-multiplexed device, all address bits are driven onto the address/data bus in two separate phases. The first phase is used for the MSB address and is qualified with nOE driven low. The second phase for LSB address is qualified with nOE driven high. The address phase ends at nWE assertion time.

The nCS, and DIR signals are controlled as previously described..

- Address valid signal nADV is asserted and deasserted twice during a read transaction:
  - nADV first assertion time is controlled by the GPMC_CONFIG3_i[6:4] ADVAADMUXONTIME bit field.
  - nADV first deassertion time is controlled by the GPMC_CONFIG3_i[26:24] ADVAADMUXRDOFFTIME bit field.
  - nADV second assertion time is controlled by the GPMC_CONFIG3_i[3:0] ADVONTIME bit field.
  - nADV second deassertion time is controlled by the GPMC_CONFIG3_i[12:8] ADVRDOFFTIME bit field.
- Output Enable signal nOE is asserted and deasserted twice during a read transaction (nOE second assertion indicates a read cycle):
  - nOE first assertion time is controlled by the GPMC_CONFIG4_i[6:4] OEAADMUXONTIME bit field.
  - nOE first deassertion time is controlled by the GPMC_CONFIG4_i[15:13] OEAADMUXOFFTIME bit field.
  - nOE second assertion time is controlled by the GPMC_CONFIG4_i[3:0] OEONTIME bit field.
  - nOE second deassertion time is controlled by the GPMC_CONFIG4_i[12:8] OEOFFTIME bit field.

First write data is driven by the GPMC at GPMC_CONFIG6_i[19:16] WRDATAONADMUXBUS, when in address/data-multiplexed configuration. The next write data of the burst is driven on the bus at WRACCESSTIME + 1 during GPMC_CONFIG5_i[27:24] PAGEBURSTACCESSTIME GPMC_FCLK cycles. The last data of the synchronous burst write is driven until GPMC_CONFIG5_i[12:8] WRCYCLETIME completes.

- WRACCESSTIME is defined in the GPMC_CONFIG6_i[28:24] bit field.
- The PAGEBURSTACCESSTIME parameter must be set accordingly with GPMCFCLKDIVIDER and the memory-device internal configuration.

Total access time GPMC_CONFIG5_i[12:8] WRCYCLETIME corresponds to WRACCESSTIME plus the address hold time from nCS deassertion. In Figure 10-34, the programmed value of WRCYCLETIME equals WRCYCLETIME0 + WRCYCLETIME1. WRCYCLETIME0 and WRCYCLETIME1 delays are not actual parameters and are only a graphical representation of the full WRCYCLETIME value.

After a write operation, if no other access (read or write) is pending, the data bus keeps the previous value. See Section 10.3.4.9.10, *Bus Keeping Support*.

10.3.4.10.3 Asynchronous and Synchronous Accesses in Nonmultiplexed Mode

Page mode is available only in nonmultiplexed mode.

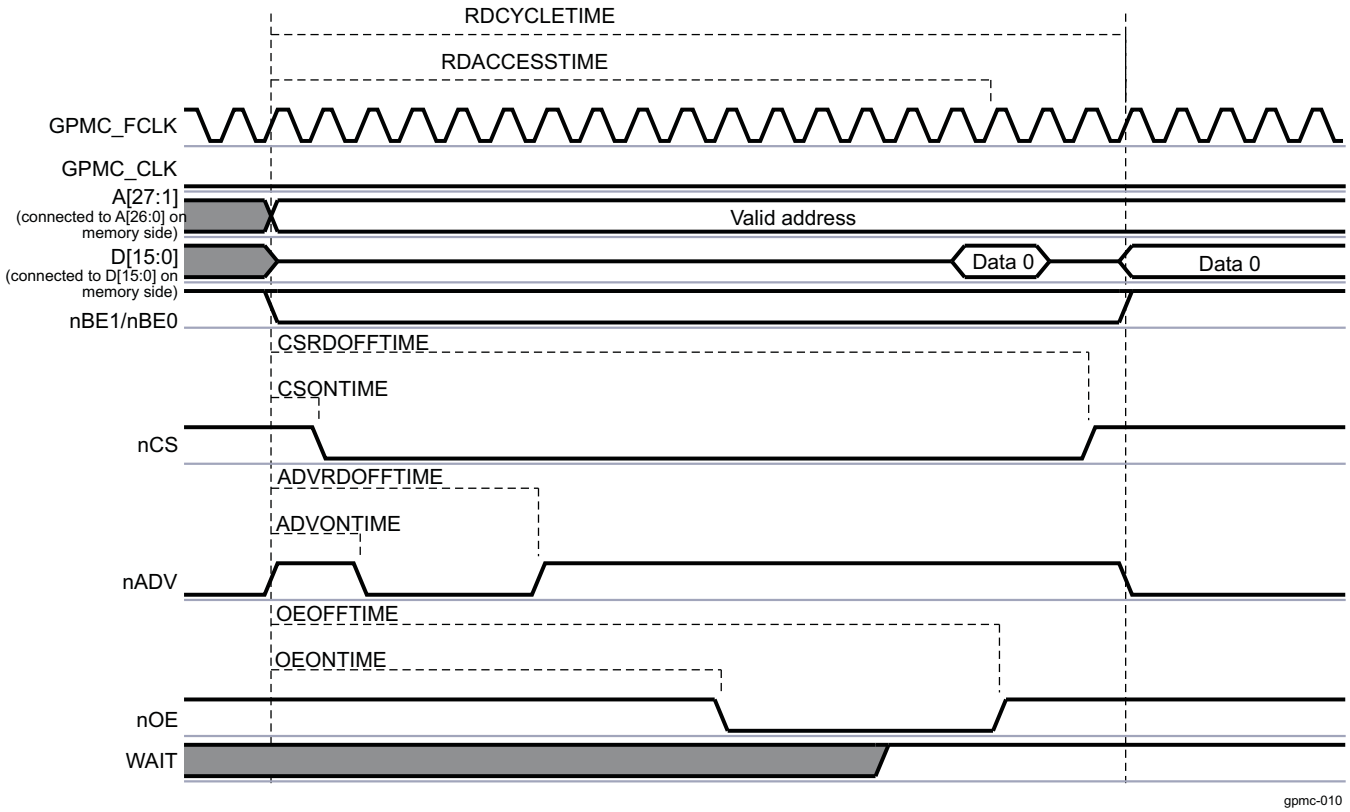
- Asynchronous single-read operation on a nonmultiplexed device
- Asynchronous single-write operation on a nonmultiplexed device
- Asynchronous multiple- (page mode) read operation on a nonmultiplexed device

- Synchronous operations on a nonmultiplexed device

10.3.4.10.3.1 Asynchronous Single-Read Operation on Nonmultiplexed Device

Figure 10-36 shows an asynchronous single-read operation on a nonmultiplexed device.

Figure 10-36. Asynchronous Single Read on an Address/Data-Nonmultiplexed Device



The 27-bit address (For a 16-bit data memory device, hence GPMC A[0] is not necessary to be output) is driven onto the address bus A[27:1] and the 16-bit data is driven onto the data bus D[15:0].

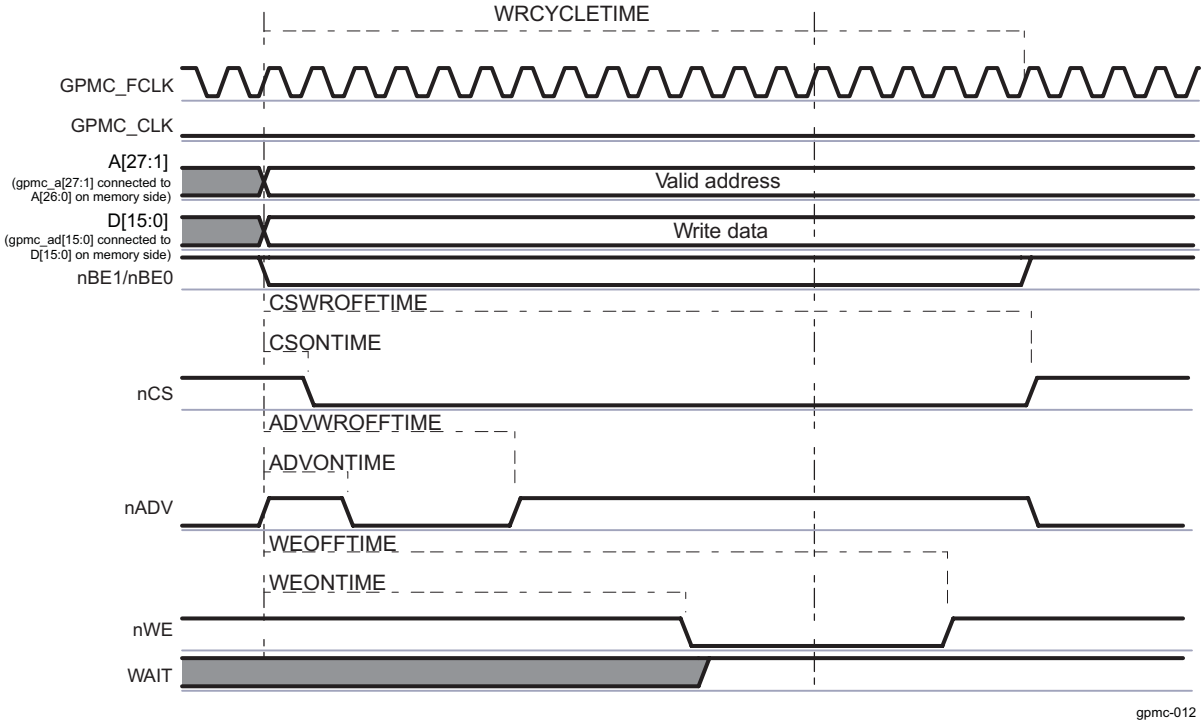
Read data is latched at GPMC_CONFIG1_5[20:16] RDACCESSTIME completion time. The end of the access is defined by the GPMC_CONFIG1_5[4:0] RDCYCLETIME parameter.

The nCS, nADV, nOE, and DIR signals are controlled in the same way as address/data-multiplexed accesses (see Table 10-391).

10.3.4.10.3.2 Asynchronous Single-Write Operation on Nonmultiplexed Device

Figure 10-37 shows an asynchronous single-write operation on a nonmultiplexed device.

Figure 10-37. Asynchronous Single Write on an Address/Data-Nonmultiplexed Device

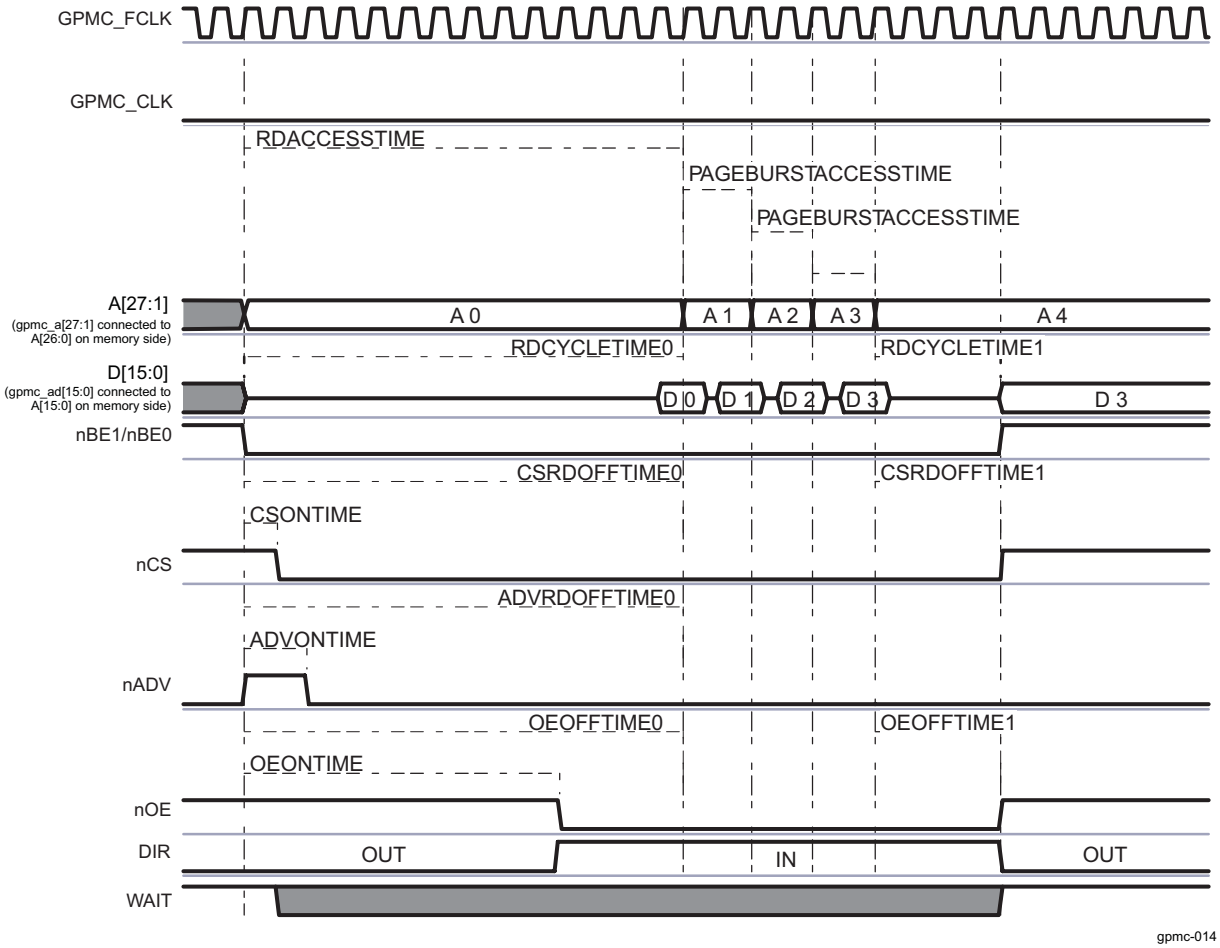


The nCS, nADV, nWE, and DIR signals are controlled in the same way as address/data-multiplexed accesses (see Table 10-391).

10.3.4.10.3.3 Asynchronous Multiple (Page Mode) Read Operation on Nonmultiplexed Device

Figure 10-38 shows an asynchronous multiple-read operation on a nonmultiplexed device in which two word32 host read accesses to the GPMC are split into one multiple- (page mode of 4 word16) read access to the attached device.

Figure 10-38. Asynchronous Multiple (Page Mode) Read



NOTE: The WAIT signal is active low.

The nCS, nADV, nOE, and DIR signals are controlled in the same way as address/data-multiplexed accesses (see Table 10-391).

When RDACCESSTIME completes, control signal timings are frozen during the multiple data transactions, corresponding to PAGEBURSTACCESSTIME multiplied by the number of remaining data transactions.

Read data is latched at GPMC_CONFIG5_i[20:16] RDACCESSTIME completion time (where i = 0 to 7). The end of the access is defined by the GPMC_CONFIG5_i[4:0] RDCYCLETIME parameter.

During consecutive accesses, the GPMC increments the address after each data read completes.

Delay between successive read data in the page is controlled by the GPMC_CONFIG5_i[27:24] PAGEBURSTACCESSTIME parameter. Depending on the device page length, the GPMC can control device page crossing during a burst request and insert initial RDACCESSTIME latency. Page crossing is possible only with a new burst access, meaning a new initial access phase is initiated.

Total access time RDCYCLETIME corresponds to RDACCESSTIME, plus the address hold time, starting from the nCS deassertion.

- The read cycle time is defined in the GPMC_CONFIG5_i[4:0] RDCYCLETIME bit field.
- In Figure 10-38, the programmed value of RDCYCLETIME equals RDCYCLETIME0 (before paged accesses) + RDCYCLETIME1 (after paged accesses).

10.3.4.10.3.4 Synchronous Operations on a Nonmultiplexed Device

All information for this section is equivalent to similar operations for address/data-multiplexed or AAD-multiplexed accesses. The only difference resides in the address phase. See Section 10.3.5.3, *GPMC Configuration in NOR Mode*.

10.3.4.10.4 Page and Burst Support

Each chip-select can be configured to process system single or burst requests into successive single accesses or asynchronous page/synchronous burst accesses, with appropriate access size adaptation.

Depending on the external device page or burst capability, read and write accesses can be independently configured through the GPMC. The GPMC_CONFIG1_i[30] READMULTIPLE and GPMC_CONFIG1_i[28] WRITMULTIPLE bits (where i = 0 to 7) are associated with the READTYPE and WRITETYPE parameters.

NOTE:

- Asynchronous write page mode is not supported.
- 8-bit-wide device support is limited to nonburstable devices (READMULTIPLE and WRITEMULTIPLE are ignored).
- Not applicable to NAND device interfacing

10.3.4.10.5 System Burst vs External Device Burst Support

The device system can issue the following requests to the GPMC:

- Byte, 16-bit word, 32-bit word requests (byte-enable-controlled). This is always a single request from the interconnect point of view.
- Incrementing fixed-length bursts of two, four, and eight words
- Wrapped (critical word access first) fixed-length burst of two, four, or eight words

To process a system request with the optimal protocol, the READMULTIPLE (and READTYPE) and WRITEMULTIPLE (and WRITETYPE) parameters must be set according to the burstable capability (synchronous or asynchronous) of the attached device.

The GPMC access engine issues only fixed-length bursts. The maximum length that can be issued is defined per chip-select by the GPMC_CONFIG1_i[24:23] ATTACHEDDEVICEPAGELENGTH bit field (where i = 0 to 7). When the value of ATTACHEDDEVICEPAGELENGTH is less than the length of the system burst request (including the appropriate access size adaptation according to the device width), the GPMC splits the system burst request into multiple bursts. Within the specified 4-, 8-, or 16-word value, the value of the ATTACHEDDEVICEPAGELENGTH bit field must correspond to the maximum length burst supported by the memory device configured in fixed-length burst mode (as opposed to continuous burst mode).

To get optimal performance from memory devices that natively support 16 Word16-length-wrapping burst capability (critical word access first), the ATTACHEDDEVICEPAGELENGTH parameter must be set to 16 words and the GPMC_CONFIG1_i[31] WRAPBURST bit (where i = 0 to 7) must be set to 1. Similarly DEVICEPAGELENGTH is set to 4 and 8 for memories supporting 4 and 8 Word16-length-wrapping burst, respectively.

When the memory device does not offer (or is not configured to offer) native 16 Word16-length-wrapping burst, the WRAPBURST parameter must be cleared, and the GPMC access engine emulates the wrapping burst by issuing the appropriate burst sequences according to the value of ATTACHEDDEVICEPAGELENGTH.

When the memory device does not support native-wrapping burst, there is usually no difference in behavior between a fixed-burst length mode and a continuous-burst mode configuration (except for a potential power increase from a memory-speculative data prefetch in a continuous burst read). However, even though continuous burst mode is compatible with GPMC behavior, because the GPMC access engine issues only fixed-length burst and does not benefit from continuous burst mode, it is best to configure the memory device in fixed-length burst mode.

The memory device maximum-length burst (configured in fixed-length burst wrap or nonwrap mode) usually corresponds to the memory device data buffer size. Memory devices with a minimum of 16 half-word buffers are the most appropriate (especially with wrap support), but memory devices with smaller buffer size (4 or 8) are also supported, assuming that the GPMC_CONFIG1_i[24:23] ATTACHEDDEVICEPAGELENGTH bit field is set accordingly to 4 or 8 words.

The device system issues only requests with addresses or starting addresses for nonwrapping burst requests; that is, the request size boundary is aligned. In case of an eight-word-wrapping burst, the wrapping address always occurs on the eight-word boundary. As a consequence, all words requested must be available from the memory data buffer when the buffer size is equal to or greater than the value of ATTACHEDDEVICEPAGELENGTH. This usually means that data can be read from or written to the buffer at a constant rate (number of cycles between data) without wait-states between data accesses. If the memory does not behave this way (nonzero wait-state burstable memory), wait pin monitoring must be enabled to dynamically control data access completion within the burst.

NOTE: When the system burst request length is less than the value of ATTACHEDDEVICEPAGELENGTH, the GPMC proceeds with the required accesses.

10.3.4.11 pSRAM Access Specificities

pSRAM devices are SRAM-pin-compatible low-power memories that contain a self-refreshed DRAM memory array. The GPMC_CONFIG1_i[11:10] DEVICETYPE bit field (where i = 0 to 7) must be set to 0b00.

The pSRAM device uses the NOR protocol. It supports the following operations:

- Asynchronous single read
- Asynchronous page read
- Asynchronous single write
- Synchronous single read and write
- Synchronous burst read
- Synchronous burst write (not supported by NOR flash memory)

pSRAM devices must be powered up and initialized in a predefined manner according to the specifications of the attached device.

pSRAM devices can be programmed to use either mode: fixed or variable latency. pSRAM devices can automatically schedule autorefresh operations, which force the GPMC to use its WAIT signal capability when read or write operations occur during an internal self-refresh operation, or they can automatically include the autorefresh operation in the access time. These devices do not require additional WAIT signal capability or a minimum nCS high pulse width between consecutive accesses to ensure that the correct internal refresh operation is scheduled.

10.3.4.12 NAND Access Description

NAND (8-bit and 16-bit) memory devices using a standard NAND asynchronous address/data-multiplexing scheme can be supported on any chip-select with the appropriate asynchronous configuration settings.

As for any other type of memory compatible with the GPMC interface, accesses to a chip-select allocated to a NAND device can be interleaved with accesses to chip-selects allocated to other external devices. This interleaved capability limits the system to *chip enable don't care* NAND devices, because the chip-select allocated to the NAND device must be deasserted if accesses to other chip-selects are requested.

10.3.4.12.1 NAND Memory Device in Byte or 16-bit Word Stream Mode

NAND devices require correct command and address programming before data array read or write accesses. The GPMC does not include specific hardware to translate a random address system request into a NAND-specific multiphase access. In that sense, GPMC NAND support, as opposed to random memory-map device support, is data stream-oriented (byte or 16-bit word).

The GPMC NAND programming model relies on a software driver for address and command formatting with the correct data address pointer value according to the block and page structure. Because of NAND structure and protocol interface diversity, the GPMC does not support automatic command and address phase programming, and software drivers must access the NAND device ID to ensure that correct command and address formatting are used for the identified device.

NAND device data read and write accesses are achieved through an asynchronous read or write access. The associated chip-select signal timing control must be programmed according to the NAND device timing specification.

Any chip-select region can be qualified as a NAND region to constrain the nADV/ALE signal as ALE (ALE active high, default state value at low) during address program access, and the nBE0/CLE signal as CLE (CLE active high, default state value at low) during command program access. GPMC address lines are not used (the previous value is not changed) during NAND access.

10.3.4.12.1.1 Chip-Select Configuration for NAND Interfacing in Byte or Word Stream Mode

The GPMC_CONFIG7_i register (where i = 0 to 7) associated with a NAND device region interfaced in byte or word stream mode can be initialized with a minimum size of 16MiB, because any address location in the chip-select memory region can be used to access a NAND data array. The NAND flash protocol specifies an address sequence where address bits are passed through the data bus in a series of write accesses with the ALE pin asserted. After this address phase, all operations are streamed and the system requests address is irrelevant.

CAUTION

To allow correct command, address, and data-access controls, the GPMC_CONFIG1_i register associated with a NAND device region must be initialized in asynchronous read and write modes with the parameters listed in Table 10-369. Failure to comply with these settings corrupts the NAND interface protocol.

Table 10-369. Chip-Select Configuration for NAND Interfacing

Bit Field	Register	Value	Comments
WRAPBURST	GPMC_CONFIG1_i[31] (1)	0	No wrap
READMULTIPLE	GPMC_CONFIG1_i[30]	0	Single access
READTYPE	GPMC_CONFIG1_i[29]	0	Asynchronous mode
WRITEMULTIPLE	GPMC_CONFIG1_i[28]	0	Single access
WRITETYPE	GPMC_CONFIG1_i[27]	0	Asynchronous mode
CLKACTIVATIONTIME	GPMC_CONFIG1_i[26:25]	0b00	
ATTACHEDDEVICEPAGELENGTH	GPMC_CONFIG1_i[24:23]	Don't care	Single-access mode
WAITREADMONITORING	GPMC_CONFIG1_i[22]	0	Wait not monitored by GPMC access engine
WAITWRITEMONITORING	GPMC_CONFIG1_i[21]	0	Wait not monitored by GPMC access engine
WAITMONITORINGTIME	GPMC_CONFIG1_i[19:18]	Don't care	Wait not monitored by GPMC access engine
WAITPINSELECT	GPMC_CONFIG1_i[17:16]		Select which wait is monitored by edge detectors
DEVICESIZE	GPMC_CONFIG1_i[13:12]	0b00 or 0b01	8- or 16-bit interface
DEVICETYPE	GPMC_CONFIG1_i[11:10]	0b10	NAND device in stream mode
MUXADDDATA	GPMC_CONFIG1_i[9:8]	0b00	Nonmultiplexed mode

(1) i = 0 to 7

Table 10-369. Chip-Select Configuration for NAND Interfacing (continued)

Bit Field	Register	Value	Comments
TIMEPARAGRANULARITY	GPMC_CONFIG1_i[4]	0	Timing achieved with best GPMC clock granularity
GPMCFCLKDIVIDER	GPMC_CONFIG1_i[1:0]	Don't care	Asynchronous mode

The GPMC_CONFIG1_i to GPMC_CONFIG4_i registers (where i = 0 to 7) associated with a NAND device region must be initialized with the correct control-signal timing value according to the NAND device timing parameters.

10.3.4.12.1.2 NAND Device Command and Address Phase Control

NAND devices require multiple address programming phases. The software driver must issue the correct number of command and address program accesses, according to the device command set and the device address-mapping scheme.

NAND device-command and address-phase programming is achieved through write requests to the GPMC_NAND_COMMAND_i and GPMC_NAND_ADDRESS_i register locations (where i = 0 to 7) with the correct command and address values. These locations are mapped in the associated chip-select register region. The associated chip-select signal timing control must be programmed according to the NAND device timing specification.

Command and address values are not latched during the access and cannot be read back at the register location.

- Only write accesses must be issued to these locations, but the GPMC does not discard any read access. Accessing a NAND device with nOE and CLE or ALE asserted (read access) can produce undefined results.
- Write accesses to the GPMC_NAND_COMMAND_i and GPMC_NAND_ADDRESS_i register locations must be posted for faster operations (where i = 0 to 7). The GPMC_CONFIG[0] NANDFORCEPOSTEDWRITE bit enables write accesses to these locations as posted, even if they are defined as nonposted.

A write buffer is used to store write transaction information before the external device is accessed:

- Up to eight consecutive posted write accesses can be accepted and stored in the write buffer.
- For nonposted write, the pipeline is one deep.
- An GPMC_STATUS[0] EMPTYWRITEBUFFERSTATUS bit stores the empty status of the write buffer.

The GPMC_NAND_COMMAND_i and GPMC_NAND_ADDRESS_i registers (where i = 0 to 7) are 32-bit word locations, which means any 32-or 16-bit word access is split into 4- or 2-byte accesses if an 8-bit-wide NAND device is attached. For multiple-command phase or multiple-address phase, the software driver can use 32- or 16-bit word access to these registers, but it must consider the splitting and little-endian ordering scheme. When only one byte command or address phase is required, only byte write access to the GPMC_NAND_COMMAND_i and GPMC_NAND_ADDRESS_i registers can be used, and any of the four byte locations of the registers is valid.

The same applies to a GPMC_NAND_COMMAND_i and a GPMC_NAND_ADDRESS_i (where i = 0 to 7) 32-bit word write access to a 16-bit-wide NAND device (split into two 16-bit word accesses). In the case of a 16-bit word write access, the MSByte of the 16-bit word value must be set according to the NAND device requirement (usually 0). Either 16-bit word location or any one of the four byte locations of the registers is valid.

10.3.4.12.1.3 Command Latch Cycle

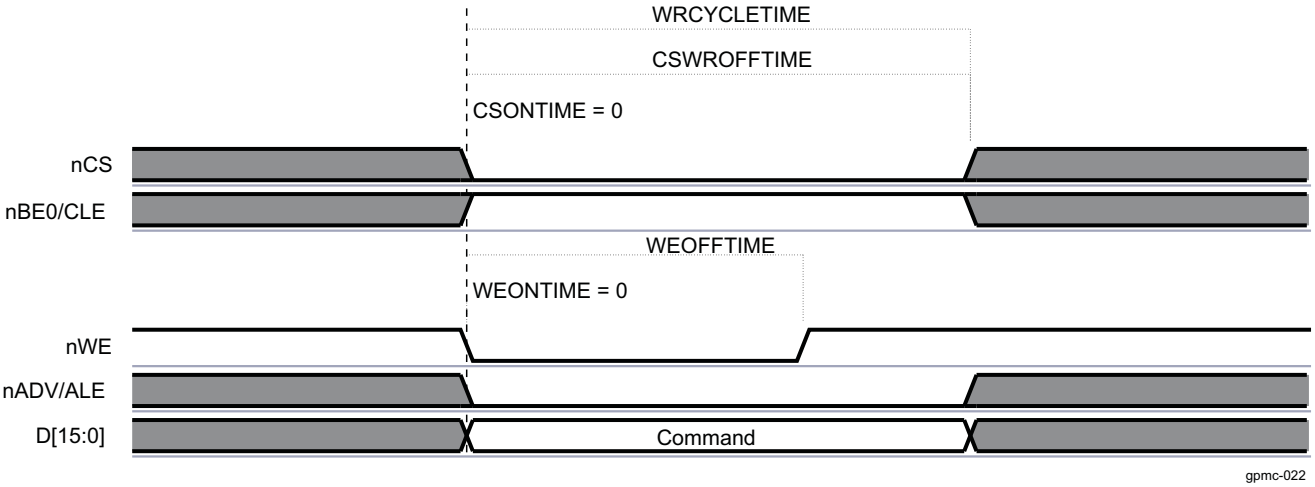
Writing data at the GPMC_NAND_COMMAND_i location (where i = 0 to 7) places the data as the NAND command value on the bus, using a regular asynchronous write access.

- nCE is controlled by the CSONTIME and CSWROFFTIME timing parameters.
- CLE is controlled by the ADVONTIME and ADVWROFFTIME timing parameters.
- nWE is controlled by the WEONTIME and WEOFFTIME timing parameters.

- ALE and nRE (nOE) are maintained inactive.

Figure 10-39 shows the NAND command latch cycle.

Figure 10-39. NAND Command Latch Cycle



NOTE: CLE is shared with the nBE0 output signal and has an inverted polarity from BE0. The NAND qualifier deals with this. During the asynchronous NAND data access cycle, nBE0 (also nBE1) must not toggle, because it is shared with CLE.

NAND flash memories do not use byte-enable signals.

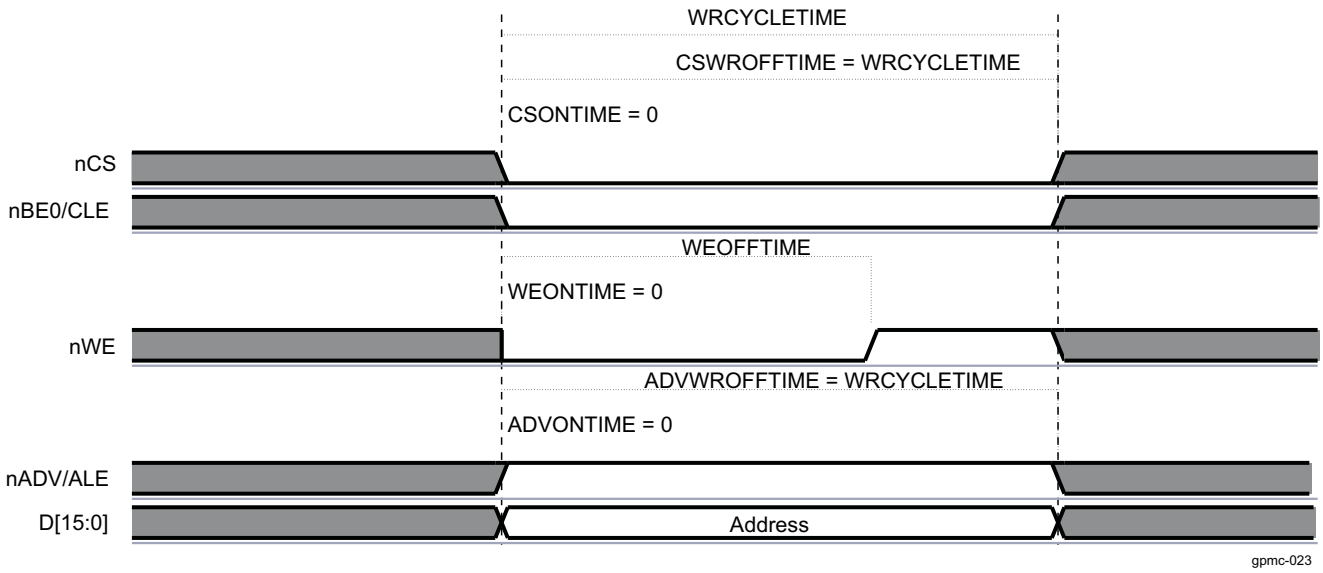
10.3.4.12.1.4 Address Latch Cycle

Writing data at the GPMC_NAND_ADDRESS_i location (where i = 0 to 7) places the data as the NAND partial address value on the bus, using a regular asynchronous write access.

- nCS is controlled by the CSONTIME and CSWROFFTIME timing parameters.
- ALE is controlled by the ADVONTIME and ADVWROFFTIME timing parameters.
- nWE is controlled by the WEONTIME and WEOFFTIME timing parameters.
- CLE and nRE (nOE) are maintained inactive.

Figure 10-40 shows the NAND address latch cycle.

Figure 10-40. NAND Address Latch Cycle



NOTE: ALE is shared with the nADV output signal and has an inverted polarity from ADV. The NAND qualifier deals with this. During the asynchronous NAND data access cycle, ALE is kept stable.

10.3.4.12.1.5 NAND Device Data Read and Write Phase Control in Stream Mode

NAND device data read and write accesses are achieved through a read or write request to the chip-select-associated memory region at any address location in the region or through a read or write request to the GPMC_NAND_DATA_i location (where i = 0 to 7) mapped in the chip-select-associated control register region. GPMC_NAND_DATA_i is not a true register, but an address location to enable nRE or nWE signal control. The associated chip-select signal timing control must be programmed according to the NAND device timing specification.

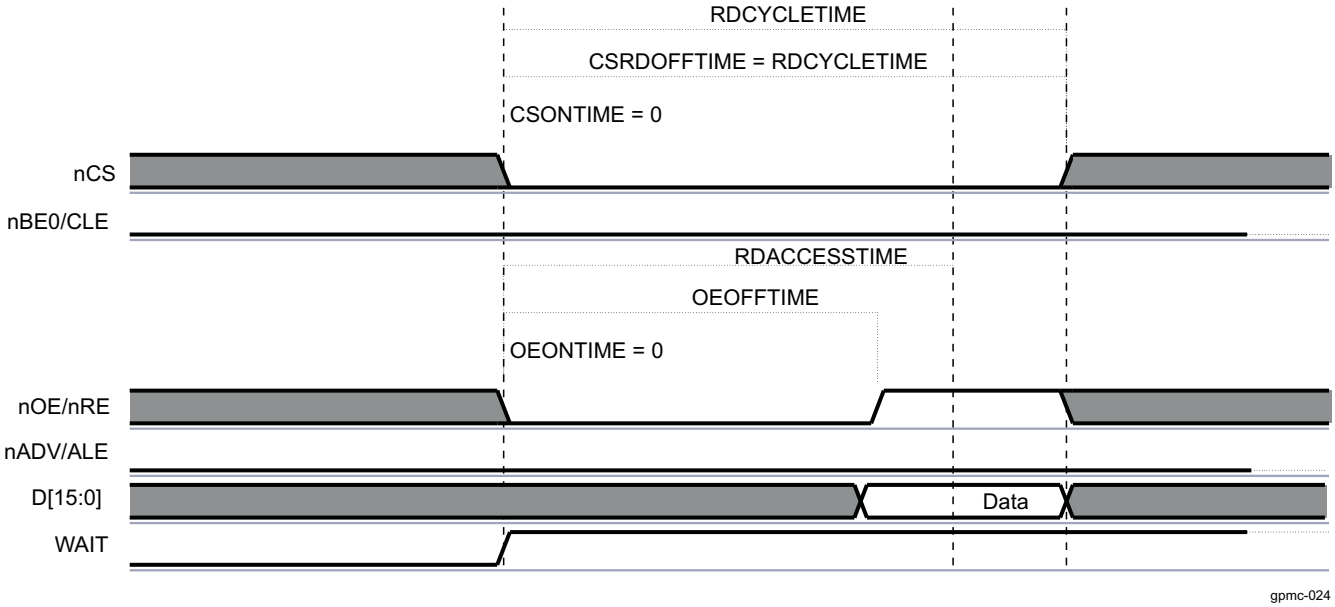
Reading data from the GPMC_NAND_DATA_i location or from any location in the associated chip-select memory region activates an asynchronous read access.

- nCS is controlled by the CSONTIME and CSRDOFFTIME timing parameters.
- nRE is controlled by the OEONTIME and OEOFFTIME timing parameters.
- To take advantage of nRE high-to-data invalid minimum timing value, RDACCESSTIME can be set so that data are effectively captured after nRE deassertion. This allows optimization of NAND read access cycle time completion. For optimal timing parameter settings, see the NAND device and the device timing parameters.

ALE, CLE, and nWE are maintained inactive.

Figure 10-41 shows the NAND data read cycle.

Figure 10-41. NAND Data Read Cycle

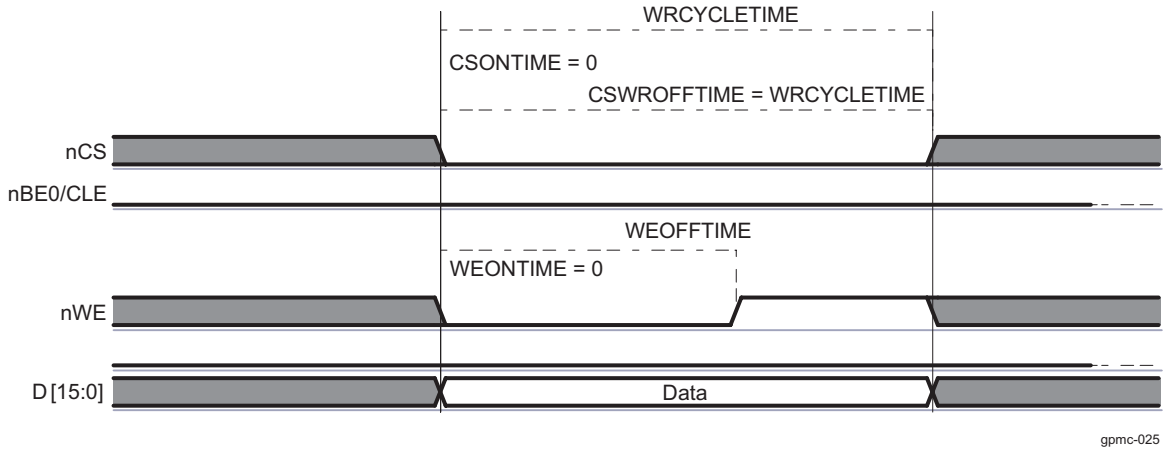


Writing data to the GPMC_NAND_DATA_i location or to any location in the associated chip-select memory region activates an asynchronous write access.

- nCS is controlled by the CSONTIME and CSWROFFTIME timing parameters.
- nWE is controlled by the WEONTIME and WEOFFTIME timing parameters.
- ALE, CLE, and nRE (nOE) are maintained inactive.

Figure 10-42 shows the NAND data write cycle.

Figure 10-42. NAND Data Write Cycle



10.3.4.12.1.6 NAND Device General Chip-Select Timing Control Requirement

For most NAND devices, read data access time is dominated by nCS-to-data-valid timing and has faster nRE-to-data-valid timing. Successive accesses with nCS deassertions between accesses are affected by this timing constraint. Because accesses to a NAND device can be interleaved with other chip-select accesses, there is no certainty that nCS always stays low between two accesses to the same chip-select. Moreover, an nCS deassertion time between the same chip-select NAND accesses is likely to be required as follows: the nCS deassertion requires programming CYCLETIME and RDACCESSTIME according to the nCS-to-data-valid critical timing.

To get full performance from NAND read and write accesses, the prefetch engine can dynamically reduce the following on back-to-back NAND accesses (to the same memory) and suppress the minimum nCS high pulse width between accesses:

- RDCYCLETIME
- WRCYCLETIME
- RDACCESSTIME
- WRACCESSTIME
- CSRDOFFTIME
- CSWROFFTIME
- ADVRDOFFTIME
- ADVWROFFTIME
- OEOFFTIME
- WEOFFTIME

For more information about optimal prefetch engine access, see Section 10.3.4.12.4, *Prefetch and Write-Posting Engine*.

Some NAND devices require minimum write-to-read idle time, especially for device-status read accesses following status-read command programming (write access). If such write-to-read transactions are used, a minimum nCS high pulse width must be set. For this, CYCLE2CYCLESAMECSEN and CYCLE2CYCLEDELAY must be set according to the appropriate timing requirement to prevent any timing violation.

NAND devices usually have an important nRE high-to-data bus in three-state mode. This requires a bus turnaround setting (BUSTURNAROUND = 1) so that the next access to a different chip-select is delayed until the BUSTURNAROUND delay completes. Back-to-back NAND read accesses to the same NAND flash are not affected by the programmed bus turnaround delay.

10.3.4.12.1.7 Read and Write Access Size Adaptation

10.3.4.12.1.7.1 8-Bit-Wide NAND Device

Host 16- and 32-bit word read and write access requests to a chip-select associated with an 8-bit-wide NAND device are split into successive read and write byte accesses to the NAND memory device. Byte access is ordered according to little-endian organization. A NAND 8-bit-wide device must be interfaced on the D0D7 interface bus lane. GPMC data accesses are justified on this bus lane when the cs is associated with an 8-bit-wide NAND device.

10.3.4.12.1.7.2 16-Bit-Wide NAND Device

Host 32-bit word read and write access requests to a chip-select associated with a 16-bit-wide NAND device are split into successive read and write 16-bit word accesses to the NAND memory device. 16-bit word access is ordered according to little-endian organization.

Host byte read and write access requests to a 16-bit-wide NAND device are completed as 16-bit accesses on the device itself, because there is no byte-addressing capability on 16-bit-wide NAND devices. This means that the NAND device address pointer is incremented on a 16-bit word basis and not on a byte basis. For a read access, only the requested byte is given back to the host, but the remaining byte is not stored or saved by the GPMC, and the next byte or 16-bit word read access gets the next 16-bit word NAND location. For a write access, the invalid byte part of the 16-bit word is driven to FF, and the next byte or 16-bit word write access programs the next 16-bit word NAND location.

Generally, byte access to a 16-bit-wide NAND device must be avoided, especially when ECC calculation is enabled. 8- or 16-bit ECC-based computations are corrupted by a byte read to a 16-bit-wide NAND device, because the nonrequested byte is considered invalid on a read access (not captured on the external data bus; FF is fed to the ECC engine) and is set to FF on a write access.

Host requests (read/write) issued in the chip-select memory region are translated in successive single or split accesses (read/write) to the attached device. Therefore, incrementing 32-bit burst requests are translated in multiple 32-bit sequential accesses following the access adaptation of the 32-bit to 8- or 16-bit device.

10.3.4.12.2 NAND Device-Ready Pin

The NAND memory device provides a ready pin to indicate data availability after a block/page opening and to indicate that data programming is complete. The ready pin can be connected to one of the wait GPMC input pins; data read accesses must not be tried when the ready pin is sampled inactive (device is not ready) even if the associated chip-select WAITREADMONITORING bit field is set. The duration of the NAND device busy state after the block/page opening is so long (up to 50 micro second) that accesses occurring when the ready pin is sampled inactive can stall GPMC access and eventually cause a system time-out.

NOTE: If a read access to a NAND flash is done using wait monitoring mode, the device is blocked during a page opening, and so is the GPMC. If the correct settings are used, other chip-selects can be used while the memory processes the page opening command.

To avoid a time-out caused by a block/page opening delay in NAND flash, disable the wait pin monitoring for read and write accesses (that is, set the GPMC_CONFIG1_i[21] WAITWRITEMONITORING and GPMC_CONFIG1_i[22] WAITREADMONITORING bits to 0, where i = 0 to 7), and use one of the following methods instead:

- Use software to poll the WAITxSTATUS bit (where x = 0 to 2) of the GPMC_STATUS .
- Configure an interrupt that is generated on the WAIT signal change (through the GPMC_IRQENABLE register bits[11:8]).

Even if the READWAITMONITORING bit is not set, the external memory nR/B pin status is captured in the programmed wait bit in the GPMC_STATUS register.

The READWAITMONITORING bit method must be used for other memories than NAND flash, if they require the use of a WAIT signal.

10.3.4.12.2.1 Ready Pin Monitored by Software Polling

The ready signal state can be monitored through the GPMC_STATUS WAITxSTATUS bit (where x = 0 to 2). Software must monitor the ready pin only when the signal is declared valid. Refer to the NAND device timing parameters to set the correct software temporization to monitor ready only after the invalid window is complete from the last read command written to the NAND device.

10.3.4.12.2.2 Ready Pin Monitored by Hardware Interrupt

Each gpmc_wait input pin can generate an interrupt when a wait-to-no-wait transition is detected. Depending on whether the GPMC_CONFIG WAITxPINPOLARITY bits (where x = 0 to 2) is active low or active high, the wait-to-no-wait transition is a low-to-high external WAIT signal transition or a high-to-low external WAIT signal transition, respectively.

The wait transition pin detector must be cleared before any transition detection. This is done by writing 1 to the WAITxEDGEDETECTIONSTATUS bit (where x = 0 to 2) of the GPMC_IRQSTATUS register according to the gpmc_wait pin used for the NAND device-ready signal monitoring. To detect a wait-to-no-wait transition, the transition detector requires a wait active time detection of a minimum of two GPMC_FCLK cycles. Software must incorporate precautions to clear the wait transition pin detector before wait (busy) time completes.

A wait-to-no-wait transition detection can issue a GPMC interrupt if the WAITxEDGEDETECTIONENABLE bit in the GPMC_IRQENABLE register is set and if the WAITxEDGEDETECTIONSTATUS bit field in the GPMC_IRQSTATUS register is set.

The WAITMONITORINGTIME bit field does not affect wait-to-no-wait transition time detection.

It is also possible to poll the WAITxEDGEDETECTIONSTATUS bit field in the GPMC_IRQSTATUS register according to the gpmc_wait pin used for NAND device ready signal monitoring.

10.3.4.12.3 ECC Calculator

The GPMC includes an error code correction (ECC) calculator circuitry that enables ECC calculation on the fly during data read or data program (that is, write) operations. The page size supported by the ECC calculator in one calculation/context is 512 bytes.

The user can choose from two different algorithms with different error correction capabilities through the GPMC_ECC_CONFIG[16] ECCALGORITHM bit:

1. Hamming code for 1-bit error code correction on 8- or 16-bit NAND flash organized with page size greater than 512 bytes
2. Bose-Chaudhurl-Hocquenghem (BCH) code for 4- to 16-bit error correction

The GPMC does not handle the error code correction directly. During writes, the GPMC computes parity bits. During reads, the GPMC provides enough information for the processor to correct errors without reading the data buffer all over again.

The Hamming code ECC is based on a 2-dimensional (2D) (row and column) bit parity accumulation. This parity accumulation is accomplished on the programmed number of bytes or 16-bit words read from the memory device, or is written to the memory device in stream mode.

Because the ECC engine includes only one accumulation context, it can be allocated to only one chip-select at a time through the GPMC_ECC_CONFIG[3:1] ECCCS bit field. Even if two chip-selects use different ECC algorithms, one the Hamming code and the other a BCH code, they must define separate ECC contexts because some of the ECC registers are common to all types of algorithms.

10.3.4.12.3.1 Hamming Code

All references to ECC in this subsection refer to the 1-bit error correction Hamming code.

The ECC is based on a 2D (row and column) bit parity accumulation known as the Hamming code. The parity accumulation is done for a programmed number of bytes or 16-bit word read from the memory device or written to the memory device in stream mode.

There is no automatic error detection or correction, and the software NAND driver must read the multiple ECC calculation results, compare them to the expected code value, and take the appropriate corrective actions according to the error handling strategy (ECC storage in spare byte, error correction on read, block invalidation).

The ECC engine includes a single accumulation context. It can be allocated to a single designated chip-select at a time, and parallel computations on different chip-selects are not possible. Because it is allocated to a single chip-select, the ECC computation is not affected by interleaved GPMC accesses to other chip-selects and devices. The ECC accumulation is sequentially processed in the order of data read from or written to the memory on the designated chip-select. The ECC engine does not differentiate read accesses from write accesses and does not differentiate data from command or status information. Software must ensure that only relevant data are passed to the NAND flash memory while the ECC computation engine is active.

The starting NAND page location must be programmed first, followed by an ECC accumulation context reset with an ECC enabling, if required. The NAND device accesses discussed in the following sections must be limited to data read or write until the specified number of ECC calculations is complete.

10.3.4.12.3.1.1 ECC Result Register and ECC Computation Accumulation Size

The GPMC includes up to nine ECC result registers (GPMC_ECCj_RESULT, where j = 1 to 9) to store ECC computation results when the specified number of bytes or 16-bit words has been computed.

The ECC result registers are used sequentially: one ECC result is stored in one ECC result register on the list, the next ECC result is stored in the next ECC result register on the list, and so forth, until the last ECC computation. The value of the GPMC_ECCj_RESULT register is valid only when the programmed number of bytes or 16-bit words has been accumulated, which means that the same number of bytes or 16-bit words has been read from or written to the NAND device in sequence.

The GPMC_ECC_CONTROL[3:0] ECCPOINTER bit field must be set to the correct value to select the ECC result register to be used first in the list for the incoming ECC computation process. The ECCPointer can be read to determine which ECC register is used in the next ECC result storage for the ongoing ECC computation. The value of the GPMC_ECCj_RESULT register (where j = 1 to 9) can be considered valid when ECCPOINTER equals j + 1. When the GPMC_ECCj_RESULT register (where j = 9) is updated, ECCPOINTER is frozen at 10, and ECC computing is stopped (ECCENABLE = 0).

The ECC accumulator must be reset before any ECC computation accumulation process. The GPMC_ECC_CONTROL[8] ECCCLEAR bit must be set to 1 (nonpersistent bit) to clear the accumulator and all ECC result registers.

For each ECC result (each GPMC_ECCj_RESULT register, where j = 1 to 9), the number of bytes or 16-bit words used for ECC computing accumulation can be selected from between two programmable values.

The ECCjRESULTSIZE bits (where j = 1 to 9) in the GPMC_ECC_SIZE_CONFIG register select which programmable size value (ECCSIZE0 or ECCSIZE1) must be used for this ECC result (stored in the GPMC_ECCj_RESULT register).

The ECCSIZE0 and ECCSIZE1 bit fields allow selection of the number of bytes or 16-bit words used for ECC computation accumulation. Any even values from 2 to 512 are allowed.

Flexibility in the number of ECCs computed and the number of bytes or 16-bit words used in the successive ECC computations enables different NAND page error-correction strategies. Usually based on 256 or 512 bytes and on 128 or 256 16-bit word, the number of ECC results required is a function of the NAND device page size. Specific ECC accumulation size can be used when computing the ECC on the NAND spare byte.

For example, with a 2-KiB data page, 8-bit-wide NAND device, eight ECCs accumulated on 256 bytes can be computed and added to one extra ECC computed on the 24 spare bytes area where the eight ECC results used for comparison and correction with the computed data page ECC are stored. The GPMC then provides nine GPMC_ECCj_RESULT registers (j = 1 to 9) to store the results. In this case, ECCSIZE0 is set to 256, and ECCSIZE1 is set to 24; the ECC[1:8] RESULTSIZE bits are set to 0, and the ECC9RESULTSIZE bit is set to 1.

10.3.4.12.3.1.2 ECC Enabling

The GPMC_ECC_CONFIG[3:1] ECCCS bit field selects the allocated chip-select. The GPMC_ECC_CONFIG[0] ECCENABLE bit enables ECC computation on the next detected read or write access to the selected chip-select.

The following fields must not be changed or cleared while an ECC computation is in progress:

- CCPOINTER
- ECCCLEAR
- ECCSIZE
- ECCjRESULTSIZE (where j = 1 to 9)
- ECC16B
- ECCCS

The ECC accumulator and ECC result register must not be changed or cleared while an ECC computation is in progress.

Table 10-370 describes the ECC enable settings.

Table 10-370. ECC Enable Settings

Bit Field	Register	Value	Comments
ECCCS	GPMC_ECC_CONFIG	0–3	Selects the chip-select where ECC is computed
ECC16B	GPMC_ECC_CONFIG	0/1	Selects column number for ECC calculation
ECCCLEAR	GPMC_ECC_CONTROL	0–7	Clears all ECC result registers
ECCPOINTER	GPMC_ECC_CONTROL	0–7	A write to this bit field selects the ECC result register where the first ECC computation is stored. Set to 1 by default.

Table 10-370. ECC Enable Settings (continued)

Bit Field	Register	Value	Comments
ECCSIZE1	GPMC_ECC_SIZE_CONFIG	0x00–0xFF	Defines ECCSIZE1
ECCSIZE0	GPMC_ECC_SIZE_CONFIG	0x00–0xFF	Defines ECCSIZE0
ECCjRESULTSIZE (j from 1 to 9)	GPMC_ECC_SIZE_CONFIG	0/1	Selects the size of ECCn result register
ECCENABLE	GPMC_ECC_CONFIG	1	Enables the ECC computation

10.3.4.12.3.1.3 ECC Computation

The ECC algorithm is a multiple parity bit accumulation computed on the odd and even bit streams extracted from the byte or Word16 streams. The parity accumulation is split into row and column accumulations, as shown in Figure 10-43 and Figure 10-44. The intermediate row and column parities are used to compute the upper level row and column parities. Only the final computation of each parity bit is used for ECC comparison and correction.

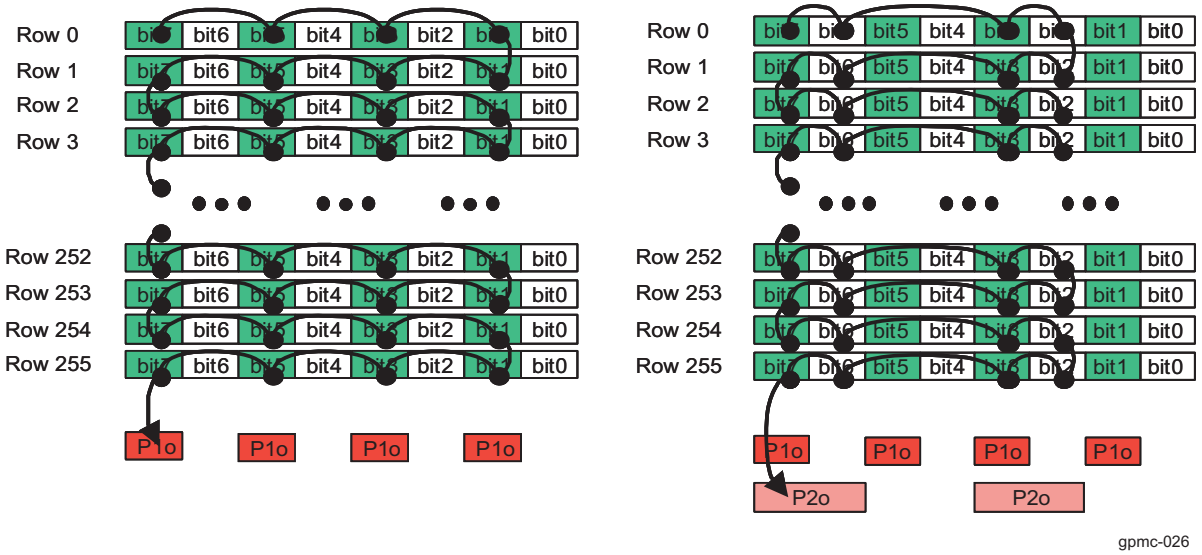
P1o = bit7 XOR bit5 XOR bit3 XOR bit1 on each byte of the data stream
P1e = bit6 XOR bit4 XOR bit2 XOR bit0 on each byte of the data stream

P2o = bit7 XOR bit6 XOR bit3 XOR bit2 on each byte of the data stream
P2e = bit5 XOR bit4 XOR bit1 XOR bit0 on each byte of the data stream

P4o = bit7 XOR bit6 XOR bit5 XOR bit4 on each byte of the data stream
P4e = bit3 XOR bit2 XOR bit1 XOR bit0 on each byte of the data stream

Each column parity bit is XORed with the previous accumulated value.

Figure 10-43. Hamming Code Accumulation Algorithm (1/2)

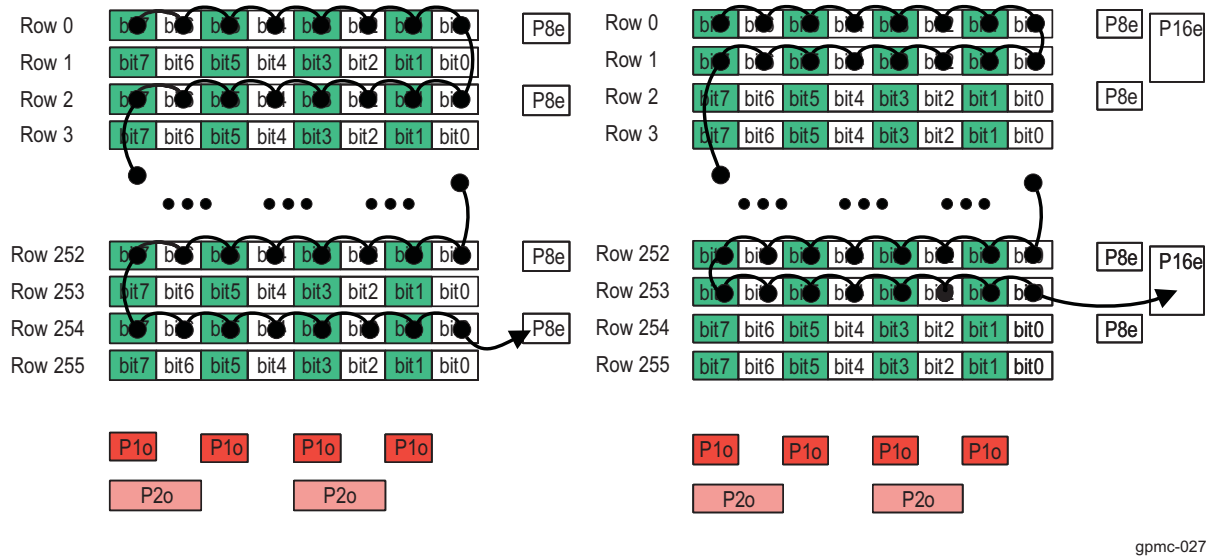


For line parities, the bits of each new data are XORed together, and line parity bits are computed as described below:

P8e = row0 XOR row2 XOR row4 XOR ... XOR row254
P8o = row1 XOR row3 XOR row5 XOR ... XOR row255

P16e = row0 XOR row1 XOR row4 XOR row5 XOR ... XOR row252 XOR row 253
P16o = row2 XOR row3 XOR row6 XOR row7 XOR ... XOR row254 XOR row 255

Figure 10-44. Hamming Code Accumulation Algorithm (2/2)



Unused parity bits in the result registers are set to 0.

Figure 10-45 shows ECC computation for a 256-byte data stream (read or write). The result includes six column parity bits (P1o-P2o-P4o for odd parities, and P1e-P2e-P4e for even parities) and sixteen row parity bits (P8o-P16o-P32o--P1024o for odd parities, and P8e-P16e-P32e--P1024e for even parities).

Figure 10-45. ECC Computation for a 256-Byte Data Stream (Read or Write)

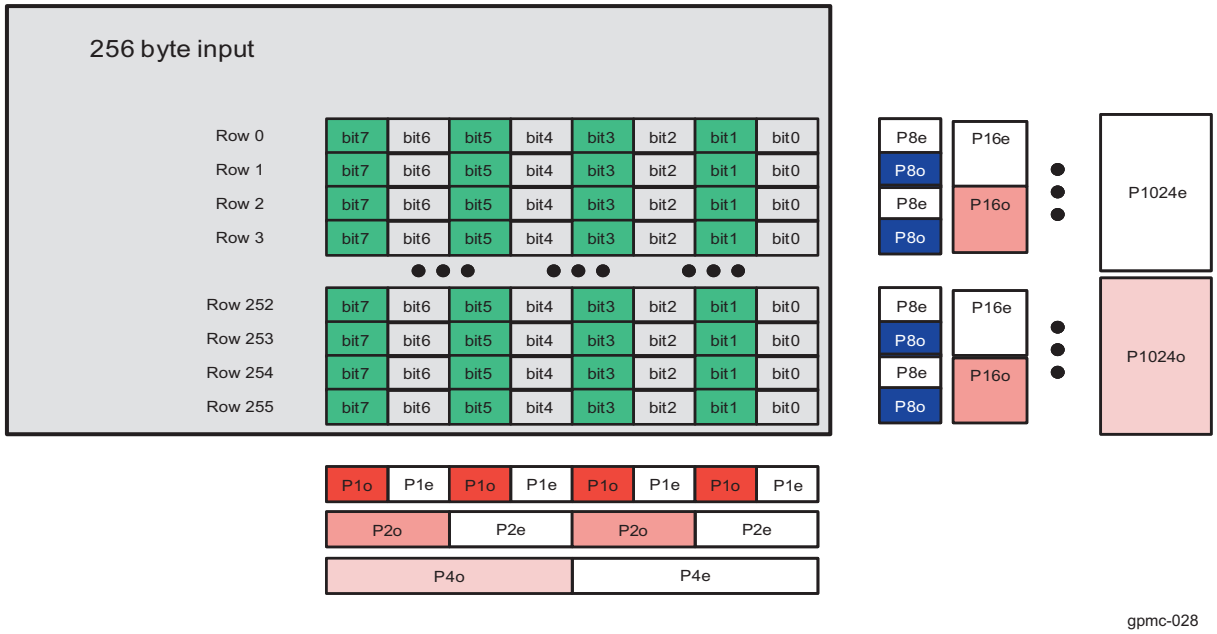
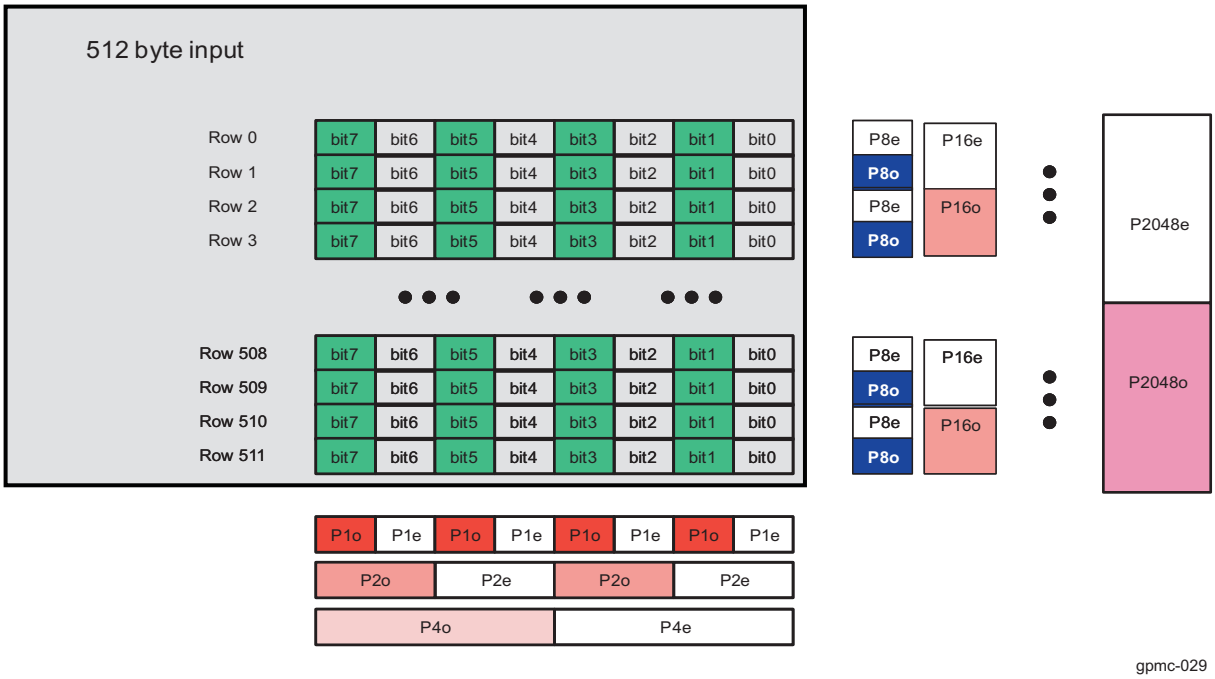


Figure 10-46 shows ECC computation for a 512-byte data stream (read or write). The result includes six column parity bits (P1o-P2o-P4o for odd parities, and P1e-P2e-P4e for even parities) and eighteen row parity bits (P8o-P16o-P32o--P1024o- - P2048o for odd parities, and P8e-P16e-P32e--P1024e- P2048e for even parities).

Figure 10-46. ECC Computation for a 512-Byte Data Stream (Read or Write)



For a 2-KiB page, four 512 bytes ECC calculations plus 1 for the spare area are required. Results are stored in the GPMC_ECCj_RESULT registers (where j = 1 to 9).

10.3.4.12.3.1.4 ECC Comparison and Correction

To detect an error, the computed ECC result must be XORed with the parity value stored in the spare area of the accessed page.

- If the result of this logical XOR is all 0s, no error is detected and the read data is correct.
- If every second bit in the parity result is a 1, 1 bit is corrupted and is located at bit address (P2048o, P1024o, P512o, P256o, P128o, P64o, P32o, P16o, P8o, P4o, P2o, P1o). Software must correct the corresponding bit.
- If only 1 bit in the parity result is 1, it is an ECC error and the read data is correct.

10.3.4.12.3.1.5 ECC Calculation Based on 8-Bit Word

The 8-bit-based ECC computation is used for 8-bit-wide NAND device interfacing.

The 8-bit-based ECC computation can be used for 16-bit-wide NAND device interfacing to get backward compatibility on the error-handling strategy used with 8-bit-wide NAND devices. In this case, the 16-bit-wide data read from or written to the NAND device is fragmented into 2 bytes. According to little-endian access, the LSB of the 16-bit-wide data is ordered first in the byte stream used for 8-bit-based ECC computation.

10.3.4.12.3.1.6 ECC Calculation Based on 16-Bit Word

ECC computation based on an 16-bit word is used for 16-bit-wide NAND device interfacing. This ECC computation is not supported when interfacing an 8-bit-wide NAND device, and the GPMC_ECC_CONFIG[7] ECC16B bit must be set to 0 when interfacing an 8-bit-wide NAND device.

The parity computation based on 16-bit words affects the row and column parity mapping. The main difference is that the odd and even parity bits P8o and P8e are computed on rows for an 8-bit-based ECC and on columns for a 16-bit based ECC. Figure 10-47 and Figure 10-48 show a 128 Word16 ECC computation scheme and a 256 16-bit word ECC computation scheme.

Figure 10-47. 128 Word16 ECC Computation

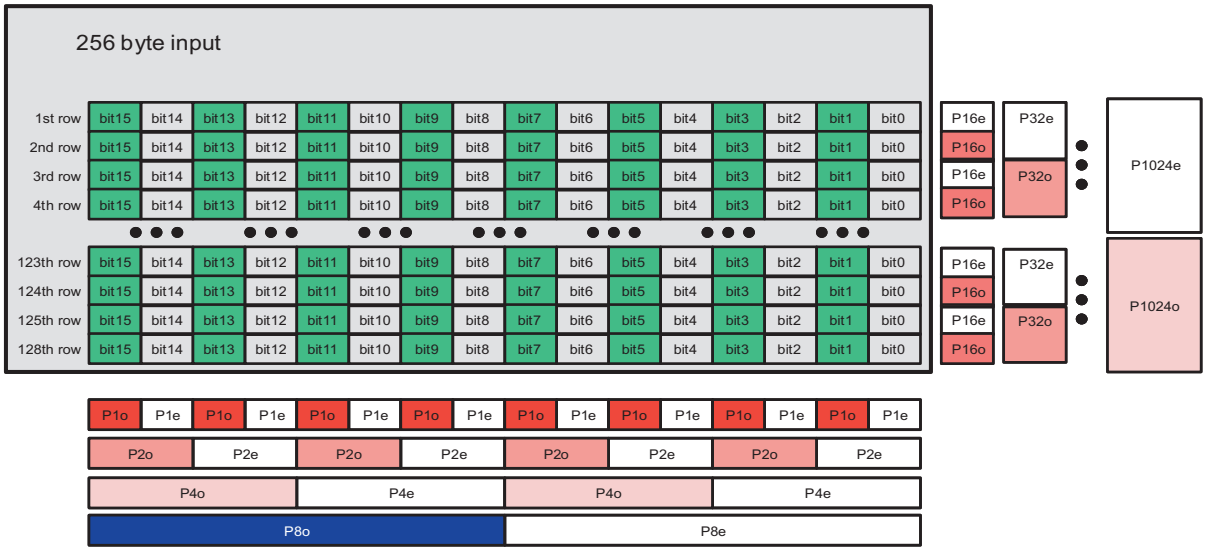
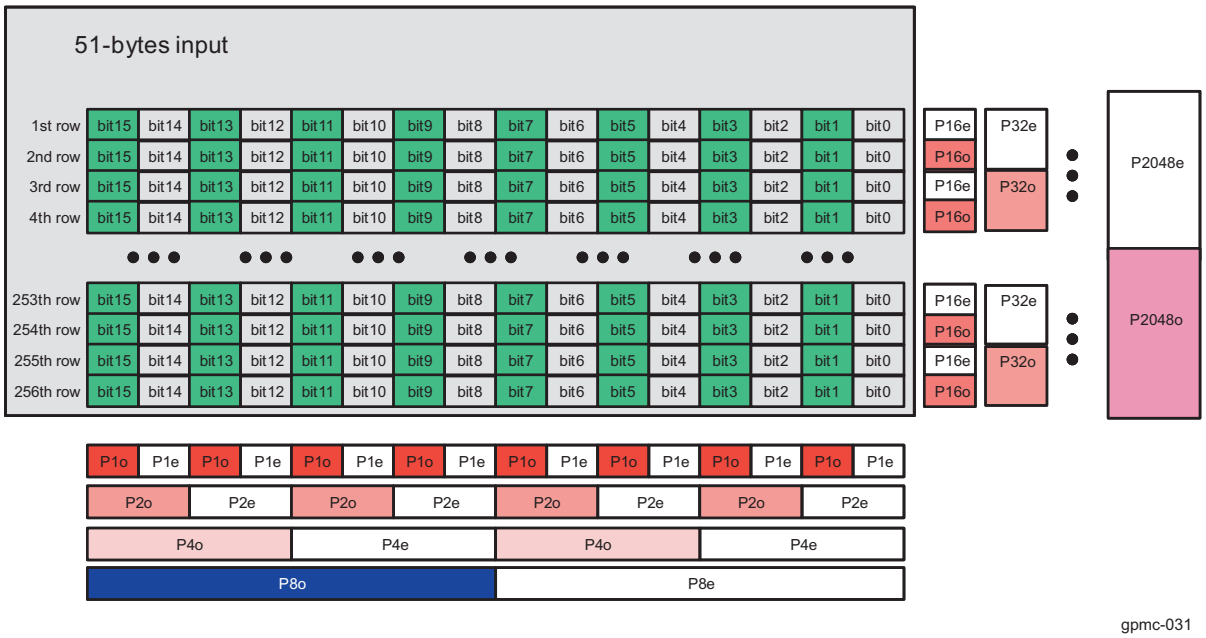


Figure 10-48. 256 Word16 ECC Computation



10.3.4.12.3.2 BCH Code

All references to ECC in this subsection refer to the 4- to 16-bit error correction BCH code.

10.3.4.12.3.2.1 Requirements

1. Read and write accesses to a NAND flash take place by whole pages, in a predetermined sequence: first the data byte page itself, and then some spare bytes, including the BCH ECC (and other information). The NAND device can cache a full page, including spares, for read and write accesses.
   - Typical page write sequence:
     – Sequential write to NAND cache of main data plus spare data, for a page. ECC is calculated on the fly. Calculated ECC can be inserted on the fly in the spares or replaced by dummy

accesses.
  - When the calculated ECC is replaced by dummy accesses, it must be written to the cache in a second, separate phase. The ECC module is disabled during that time.
  - NAND writes its cache line (page) to the array.
- Typical page read sequence:
  - Sequential read of a page. ECC is calculated on the fly.
  - The status of the ECC module buffers determines the presence of errors.

2. Accesses to several memories can be interleaved by the GPMC, but only one of those memories at a time can be a NAND using the BCH engine; in other words, only one BCH calculation (for example, for a single page) can be ongoing at any time. The sequential nature of NAND accesses ensures that the data is always written or read out in the same order. BCH-relevant accesses are selected by the chip-selects of the GPMC.
3. Each page can hold up to 4KiB of data, spare bytes not included. This means up to 8 × 512-byte BCH messages. Because all the data is written or read out first, followed by the BCH ECC, the BCH engine must be able to hold eight 104-bit remainders or syndromes (or smaller, 52-bit ones) at the same time.

   The BCH module can store all remainders internally. After the page start, an internal counter is used to detect the 512-byte sector boundaries. On those boundaries, the current remainder is stored and the divider reset for the next calculation. At the end of the page, the BCH module contains all remainders.
4. NAND access cycles hold 8 or 16 bits of data each (1 or 2 bytes); Each NAND cycle takes at least four cycles of the GPMC internal clock. This means the NAND flash timing parameters must define a RDCYCLETIME and a WRCYCLETIME of at least four clock cycles after optimization when using the BCH calculator.
5. The spare area is assumed to be large enough to hold the BCH ECC; that is, to have a message of at least 13 bytes available per 512-byte sector of data. The zone of unused spare area by the ECC may or may not be protected by the same ECC scheme, by extending the BCH message beyond 512 bytes (the maximum codeword is 1023 bytes long, ECC included, which leaves much space to cover spare bytes).

10.3.4.12.3.2.2 Memory Mapping of BCH Codeword

BCH encoding considers a block of data to protect as a polynomial message M(x). In a standard case, 512 bytes of data (that is, 2^{12} bits = 4096 bits) are seen as a polynomial of degree $2^{12} – 1 = 4095$, with parameters ranging from M0 to M4095. For 512 bytes of data, 52 bits are required for 4-bit error correction, 104 bits are required for 8-bit error correction, and 207 bits are required for 16-bit error correction. The ECC is a remainder polynomial R(x) of degree 103 (or 51, depending on the selected mode). The complete codeword C(x) is the concatenation of M(x) and R(x), as described in Table 10-371.

Table 10-371. Flattened BCH Codeword Mapping (512 Bytes + 104 Bits)

	Message M(x)			ECC R(x)		
Bit Number	M4095	...	M0	R103	...	R0

If the message is extended by the addition of spare bytes to be protected by the same ECC, the principle is still valid. For example, a 3-byte extension of the message gives a polynomial message M(x) of degree ((512 + 3) × 8) – 1 = 4119, for a total of 3 + 13 = 16 spare bytes of spare, all protected as part of the same codeword.

The message and the ECC bits are manipulated and mapped in the GPMC byte-oriented system. The ECC bits are stored in the following registers (where i = 0 to 7):

- GPMC_BCH_RESULT0_i
- GPMC_BCH_RESULT1_i
- GPMC_BCH_RESULT2_i
- GPMC_BCH_RESULT3_i

10.3.4.12.3.2.2.1 Memory Mapping of Data Message

The data message mapping must follow the following rules:

- Bit endianness within a byte is little-endian; that is, the bytes LSB is also the lowest-degree polynomial parameter: a byte b7-b0 (with b0 the LSB) represents a segment of polynomial b7 * $x^{(7+i)}$ + b6 * $x^{(6+i)}$ + ... + b0 * x^{i}
- The message is mapped in the NAND starting with the highest-order parameters, that is, in the lowest addresses of a NAND page.
- Byte endianness within the 16-bit words in the NAND is big-endian (that is, the same message mapped in 8- and 16-bit memories has the same content at the same byte address).

NOTE: The BCH module has no visibility over actual addresses. The most important point is the sequence of data words the BCH sees. However, the NAND page is always scanned incrementally in read and write accesses, which produces the mapping patterns described in the following.

Table 10-372 and Table 10-373 describe the mapping of the same 512-byte vector (typically, a BCH message) in the NAND memory space. The byte address is only an offset modulo 512 (0x200), because the same page may contain several contiguous 512-byte sectors (BCH blocks). The LSB and MSB are, respectively, the bits M0 and M(2^{12}–1) of the codeword mapping discussed previously. In both cases the data vectors are aligned; that is, their boundaries coincide with the RAM data word boundaries.

Table 10-372. Aligned Message Byte Mapping in 8-Bit NAND

Byte Offset	8-Bit Word
0x000	(MSB) Byte 511 (0x1FF)
0x001	Byte 510 (0x1FE)
...	...
0x1FF	Byte 0 (0x0) (LSB)

Table 10-373. Aligned Message Byte Mapping in 16-Bit NAND

Byte Offset	16-Bit Word MSB	16-Bit Word LSB
0x000	Byte 510 (0x1FE)	(MSB) Byte 511 (0x1FF)
0x002	Byte 508 (0x1FC)	Byte 509 (0x1FD)
...	...	...
0x1FE	Byte 0 (0x0)	(LSB) Byte 1 (0x1)

Table 10-374 through Table 10-379 list the mapping in memory of arbitrarily-sized messages, starting on access (byte or 16-bit word) boundaries for more clarity. Note that message may actually start and stop on arbitrary nibbles. A nibble is a 4-bit entity. The unused nibbles are not discarded, and they can still be used by the BCH module, but as part of the next message section (for example, on the ECC of another sector).

Table 10-374. Aligned Nibble Mapping of Message in 8-Bit NAND

Byte Offset	8-Bit Word	
	4-Bit Most Significant Nibble	4-Bit Least Significant Nibble
1	(MSB) Nibble S-1	Nibble S-2
2	Nibble S-3	Nibble S-4
...	...	...
S/2 – 2	Nibble 3	Nibble 2
S/2 – 1	Nibble 1	Nibble 0 (LSB)

Table 10-375. Misaligned Nibble Mapping of Message in 8-Bit NAND

Byte Offset	8-Bit Word	
	4-Bit Most Significant Nibble	4-Bit Least Significant Nibble
1	(MSB) Nibble S-1	Nibble S-2
2	Nibble S-3	Nibble S-4
...	...	...
(S + 1) / 2 – 2	Nibble 2	Nibble 1
(S + 1) / 2 – 1	Nibble 0 (LSB)	

Table 10-376. Aligned Nibble Mapping of Message in 16-Bit NAND

Byte Offset	16-Bit Word			
	4-Bit Most Significant Nibble		4-Bit Least Significant Nibble	
0	Nibble S-3	Nibble S-4	(MSB) Nibble S-1	Nibble S-2
2	Nibble S-7	Nibble S-8	Nibble S-5	Nibble S-6
...	...		...	
S/2 – 4	Nibble 5	Nibble 4	Nibble 7	Nibble 6
S/2 – 2	Nibble 1	Nibble 0 (LSB)	Nibble 3	Nibble 2

Table 10-377. Misaligned Nibble Mapping of Message in 16-Bit NAND (1 Unused Nibble)

Byte Offset	16-Bit Word			
	4-Bit Most Significant Nibble		4-Bit Least Significant Nibble	
0	Nibble S-3	Nibble S-4	(MSB) Nibble S-1	Nibble S-2
2	Nibble S-7	Nibble S-8	Nibble S-5	Nibble S-6
...	...		...	
(S + 1) / 2 – 4	Nibble 4	Nibble 3	Nibble 6	Nibble 5
(S + 1) / 2 – 2	Nibble 0 (LSB)		Nibble 2	Nibble 1

Table 10-378. Misaligned Nibble Mapping of Message in 16-Bit NAND (2 Unused Nibbles)

Byte Offset	16-Bit Word			
	4-Bit Most Significant Nibble		4-Bit Least Significant Nibble	
0	Nibble S-3	Nibble S-4	(MSB) Nibble S-1	Nibble S-2
2	Nibble S-7	Nibble S-8	Nibble S-5	Nibble S-6
...	...		...	
(S + 2) / 2 – 4	Nibble 3	Nibble 2	Nibble 5	Nibble 4
(S + 2) / 2 – 2			Nibble 1	Nibble 0 (LSB)

Table 10-379. Misaligned Nibble Mapping of Message in 16-Bit NAND (3 Unused Nibbles)

Byte Offset	16-Bit Word			
	4-Bit Most Significant Nibble		4-Bit Least Significant Nibble	
0	Nibble S-3	Nibble S-4	(MSB) Nibble S-1	Nibble S-2
2	Nibble S-7	Nibble S-8	Nibble S-5	Nibble S-6
...	...		...	
(S + 3) / 2 – 4	Nibble 2	Nibble 1	Nibble 4	Nibble 3
(S + 3) / 2 – 2			Nibble 0 (LSB)	

Many other cases exist than those given in the previous tables; for example, where the message does not start on a word boundary.

10.3.4.12.3.2.2.2 Memory-Mapping of the ECC

The ECC (or remainder) is presented by the BCH module as a single 104-bit (or 52-bit), little-endian vector. Software must fetch those 13 bytes (or 6 bytes) from the module interface and then store them to the spare area (page write) in the NAND or to an intermediate buffer for comparison with the stored ECC (page read). There are no constraints on the ECC mapping inside the spare area: it is software-controlled.

It is advised, however, to maintain a coherence in the respective formats of the message or the ECC remainder once they have been read out of the NAND. The error correction algorithm works from the complete codeword (concatenated message and remainder) once an error is detected. The creation of this codeword must be made as straightforward as possible.

There are cases in which the same NAND access contains both data and the ECC protecting that data. This is the case when the data/ECC boundary (which can be on any nibble) does not coincide with an access boundary. The ECC is calculated on the fly following the write. In that case, the write must also contain part of the ECC because it is impossible to insert the ECC on the fly. Instead:

- During the initial page write (BCH encoding), the ECC is replaced by dummy bits. The BCH encoder is by definition turned OFF during the ECC section, so the BCH result is unmodified.
- During a second phase, the ECC is written to the correct location, next to the actual data.
- The completed line buffer is then written to the NAND array.

10.3.4.12.3.2.2.3 Wrapping Modes

For a given wrapping mode, the module automatically goes through a specific number of sections as data is being fed into the module. For each section, the BCH core can be enabled (in which case the data is fed to the BCH divider) or not (in which case the BCH simply counts to the end of the section). When enabled, the data is added to the ongoing calculation for a given sector number (for example, number 0).

Wrapping modes are described as follows. To better understand and see the real-life read and write sequences implemented with each mode, see Section 10.3.4.12.3.2.3, *Supported NAND Page Mappings and ECC Schemes*.

For each mode:

- A sequence describes the mode in pseudo language, with, for each section, the size and the buffer used for ECC processing (if ON). The programmable lengths are size, size0, and size1.
- A checksum condition is given. If the checksum condition is not respected for a given mode, the module behavior is unpredictable. S is the number of sectors in the page; size0 and size1 are the section sizes programmed for the mode, in nibbles.

Wrapping modes 8 through 11 insert a 1-nibble padding where the BCH processing is OFF. This is intended for t = 4 ECC, where ECC is 6 bytes long and the ECC area is expected to include (at least) 1 unused nibble to remain byte-aligned.

10.3.4.12.3.2.2.4 Manual Mode (0x0)

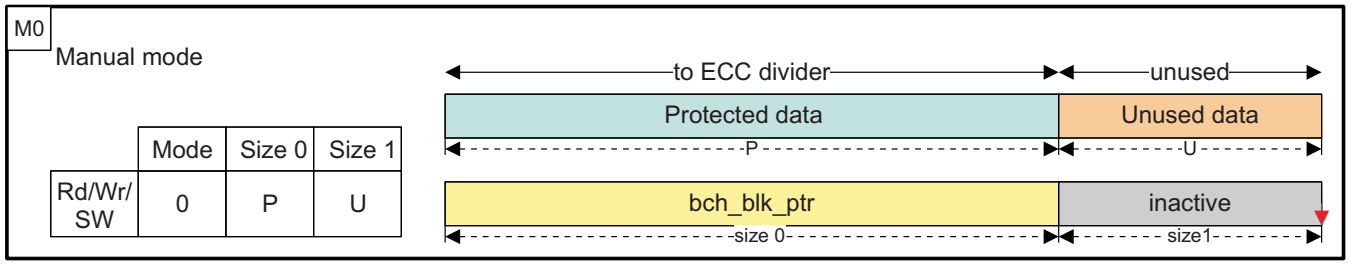
This mode is intended for short sequences, added manually to a given buffer through the software data port input. A complete page may be built out of several such sequences.

To process an arbitrary sequence of 4-bit nibbles, accesses to the software data port, containing the appropriate data, must be made. If the sequence end does not coincide with an access boundary (for example, to process 5 nibbles = 20 bits in 16-bit access mode) and those nibbles must be skipped, a number of unused nibbles must be programmed in GPMC_ECC_SIZE_CONFIG[29:22] ECCSIZE1 In the same example, 5 nibbles to process + 3 to discard = 8 nibbles = 2 × 16-bit accesses. Software must set:

- GPMC_ECC_SIZE_CONFIG[19:12] ECCSIZE0 = 0x5
- GPMC_ECC_SIZE_CONFIG[29:22] ECCSIZE1 = 0x3

NOTE: In the following figures, size and size0 are the same parameter.

Figure 10-49. Manual Mode Sequence and Mapping



gpmc-032

Section processing sequence:

- One time with buffer
  - size0 nibbles of data, processing ON
  - size1 nibbles of unused data, processing OFF

Checksum: size0 + size1 nibbles must fit in a whole number of accesses.

In the following sections, S is the number of sectors in the page.

10.3.4.12.3.2.2.5 Mode 0x1

Page processing sequence:

- Repeat with buffer 0 to S-1
  - 512-byte data, processing ON
- Repeat with buffer 0 to S-1
  - size0 nibbles spare, processing ON
  - size1 nibbles spare, processing OFF

Checksum: Spare area size (nibbles) = S – (size0 + size1)

10.3.4.12.3.2.2.6 Mode 0xA (10)

Page processing sequence:

- Repeat with buffer 0 to S-1
  - 512-byte data, processing ON
- Repeat with buffer 0 to S-1
  - size0 nibbles spare, processing ON
  - 1 nibble pad spare, processing OFF
  - size1 nibbles spare, processing OFF

Checksum: Spare area size (nibbles) = S – (size0 + 1 + size1)

10.3.4.12.3.2.2.7 Mode 0x2

Page processing sequence:

- Repeat with buffer 0 to S-1
  - 512-byte data, processing ON
- Repeat with buffer 0 to S-1
  - size0 nibbles spare, processing OFF
  - size1 nibbles spare, processing ON

Checksum: Spare area size (nibbles) = S – (size0 + size1)

10.3.4.12.3.2.2.8 Mode 0x3

Page processing sequence:

- Repeat with buffer 0 to S-1
  - 512-byte data, processing ON
- One time with buffer 0
  - size0 nibbles spare, processing ON
- Repeat with buffer 0 to S-1
  - size1 nibbles spare, processing ON

Checksum: Spare area size (nibbles) = size0 + (S – size1)

10.3.4.12.3.2.2.9 Mode 0x7

Page processing sequence:

- Repeat with buffer 0 to S-1
  - 512-byte data, processing ON
- One time with buffer 0
  - size0 nibbles spare, processing ON
- Repeat S times (no buffer used)
  - size1 nibbles spare, processing OFF

Checksum: Spare area size (nibbles) = size0 + (S – size1)

10.3.4.12.3.2.2.10 Mode 0x8

Page processing sequence:

- Repeat with buffer 0 to S-1
  - 512-byte data, processing ON
- One time with buffer 0
  - size0 nibbles spare, processing ON
- Repeat with buffer 0 to S-1
  - 1 nibble padding spare, processing OFF
  - size1 nibbles spare, processing ON

Checksum: Spare area size (nibbles) = size0 + (S – (1 + size1))

10.3.4.12.3.2.2.11 Mode 0x4

Page processing sequence:

- Repeat with buffer 0 to S-1
  - 512-byte data, processing ON
- One time (no buffer used)

  - size0 nibbles spare, processing OFF
- Repeat with buffer 0 to S-1
  - size1 nibbles spare, processing ON

Checksum: Spare area size (nibbles) = size0 + (S – size1)

10.3.4.12.3.2.2.12 Mode 0x9

Page processing sequence:

- Repeat with buffer 0 to S-1
  - 512-byte data, processing ON
- One time (no buffer used)
  - size0 nibbles spare, processing OFF
- Repeat with buffer 0 to S-1
  - 1 nibble padding spare, processing OFF
  - size1 nibbles spare, processing ON

Checksum: Spare area size (nibbles) = size0 + (S – (1 + size1))

10.3.4.12.3.2.2.13 Mode 0x5

Page processing sequence:

- Repeat with buffer 0 to S-1
  - 512-byte data, processing ON
- Repeat with buffer 0 to S-1
  - size0 nibbles spare, processing ON
- Repeat with buffer 0 to S-1
  - size1 nibbles spare, processing ON

Checksum: Spare area size (nibbles) = S – (size0 + size1)

10.3.4.12.3.2.2.14 Mode 0xB (11)

Page processing sequence:

- Repeat with buffer 0 to S-1
  - 512-byte data, processing ON
- Repeat with buffer 0 to S-1
  - size0 nibbles spare, processing ON
- Repeat with buffer 0 to S-1
  - 1 nibble padding spare, processing OFF
  - size1 nibbles spare, processing ON

Checksum: Spare area size (nibbles) = S – (size0 + 1 + size1)

10.3.4.12.3.2.2.15 Mode 0x6

Page processing sequence:

- Repeat with buffer 0 to S-1
  - 512-byte data, processing ON
- Repeat with buffer 0 to S-1
  - size0 nibbles spare, processing ON
- Repeat S times (no buffer used)
  - size1 nibbles spare, processing OFF

Checksum: Spare area size (nibbles) = S – (size0 + size1)

10.3.4.12.3.2.3 Supported NAND Page Mappings and ECC Schemes

The following rules apply to the entire mapping description:

- Main data area (sectors) size is hardcoded to 512 bytes.
- Spare area size is programmable.
- All page sections (of main area data bytes, protected spare bytes, unprotected spare bytes, and ECC) are defined as explained in Section 10.3.4.12.3.2.2.1, *Memory Mapping of Data Message*.

Each of the following sections gives a NAND page mapping example (per-sector spare mappings, pooled spare mapping, per-sector spare mapping, with ECC separated at the end of the page).

In the mapping diagrams, sections that belong to the same BCH codeword have the same color (blue or green); unprotected sections are not covered (orange) by the BCH scheme.

Below each mapping diagram, a write (encoding) and read (decoding: syndrome generation) sequence is given, with the number of the active buffers at each point in time (yellow). In the inactive zones (grey), no computing is taking place but the data counter is still active.

In Figure 10-50 through Figure 10-52, the tables on the left summarize the mode, size0, and size1 parameters to program for, respectively, write and read processing of a page, with the given mapping, where:

- P is the size of spare byte section Protected by the ECC (in nibbles)
- U is the size of spare byte section Unprotected by the ECC (in nibbles)
- E is the size of the ECC (in nibbles)
- S is the number of Sectors per page (two in the current diagrams)

Each time the processing of a BCH block is complete (ECC calculation for write/encoding, syndrome generation for read/decoding, indicated by red arrows), the update pointer is pulsed. The processing for block 0 can be the first or the last to complete, depending on the NAND page mapping and operation (read or write). All examples show a page size of 1 KiB + spares; that is, S = 2 sectors of 512 bytes. The same principles can be extended to larger pages by adding more sectors.

The actual BCH codeword size is used during the error location work to restrict the search range: by definition, errors can happen only in the codeword that was actually written to the NAND, not in the mathematical codeword of $n = 2^{13} – 1 = 8191$ bits; that codeword (higher-order bits) is all-zero and implicit during computations.

The actual BCH codeword size depends on the mode, the programmed sizes, and the sector number (all sizes in nibbles):

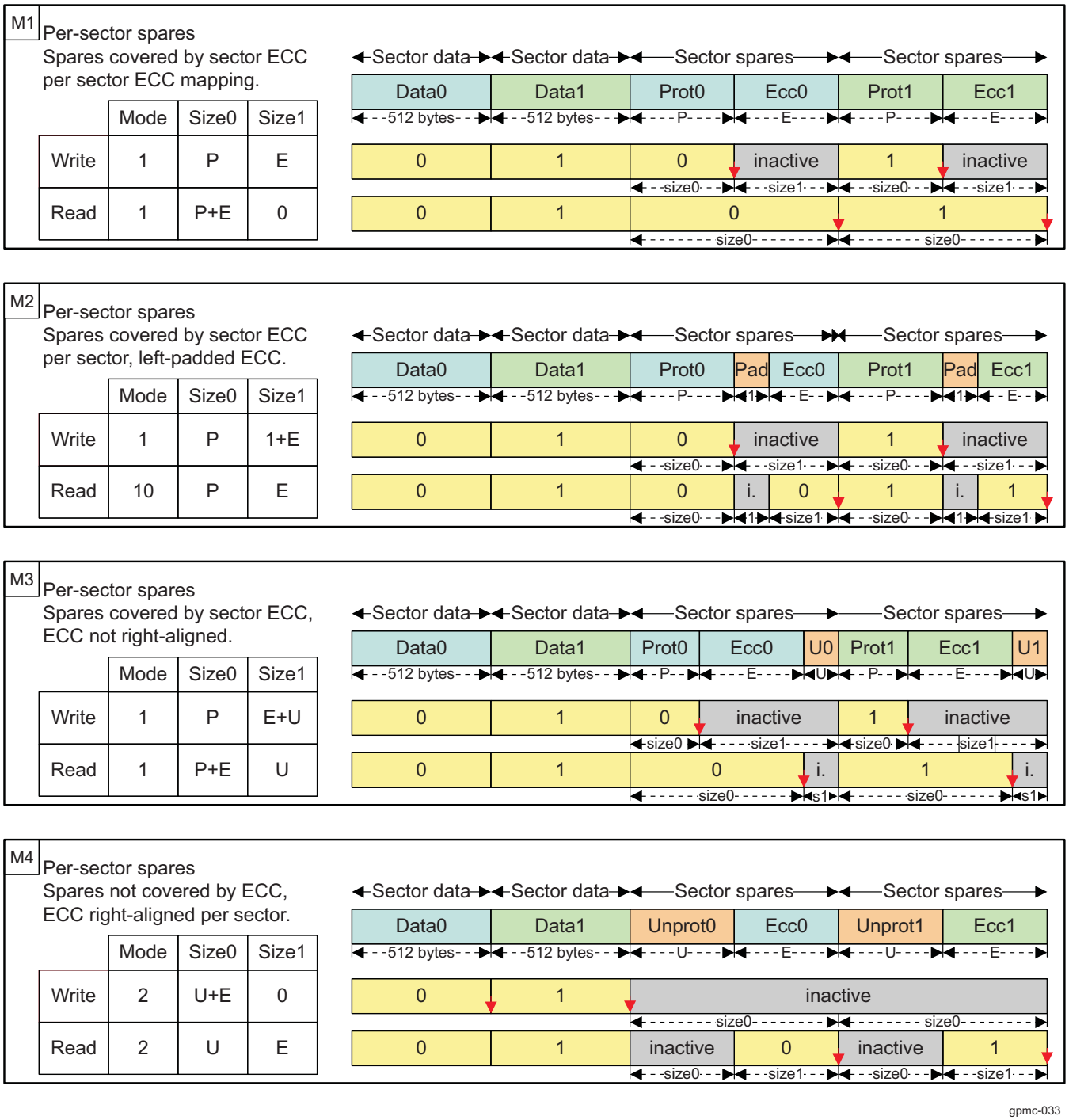
- Spares mapped and protected per sector (below: see M1-M2-M3-M9-M10):
  - All sectors: (512) + P + E
- Spares pooled and protected by sector 0 (below: see M5-M6):
  - Sector 0 codeword: (512) + P + E
  - Other sectors: (512) + E
- Unprotected spares (below: see M4-M7-M8-M11-M12):
  - All codewords (512) + E

10.3.4.12.3.2.3.1 Per-Sector Spare Mappings

In these schemes, each 512-byte sector of the main area has its own dedicated section of the spare area. The spare area of each sector is composed of:

- ECC, which must be located after the data it protects
- Other data, which may or may not be protected by the ECC for its sector.

Figure 10-50. NAND Page Mapping and ECC: Per-Sector Schemes



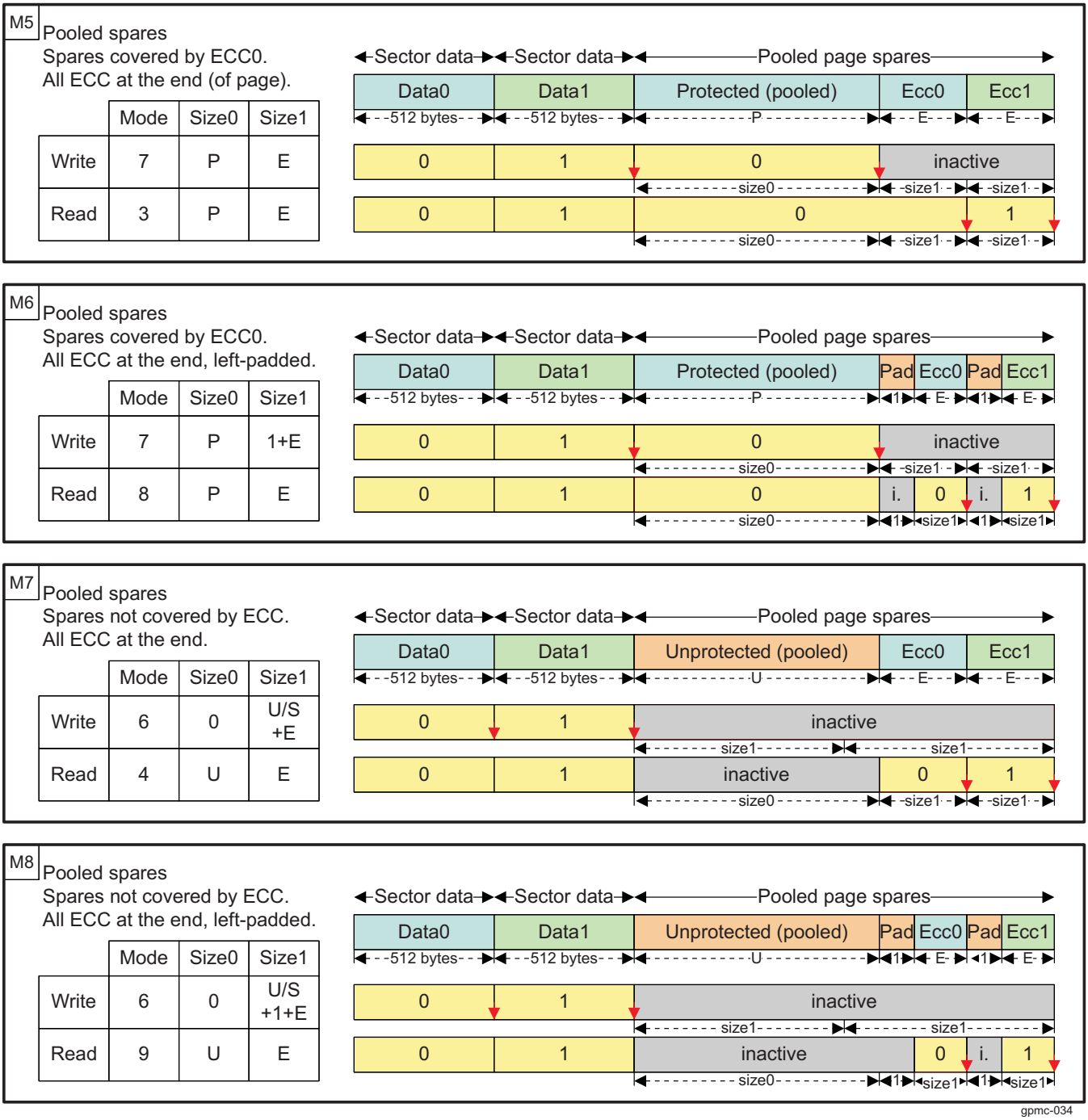
gpmc-033

10.3.4.12.3.2.3.2 Pooled Spare Mapping

In the following schemes, the spare area is pooled for the page.

- The ECC of each sector is aligned at the end of the spare area.
- The non-ECC spare data may or may not be covered by the ECC of sector 0.

Figure 10-51. NAND Page Mapping and ECC: Pooled Spare Schemes

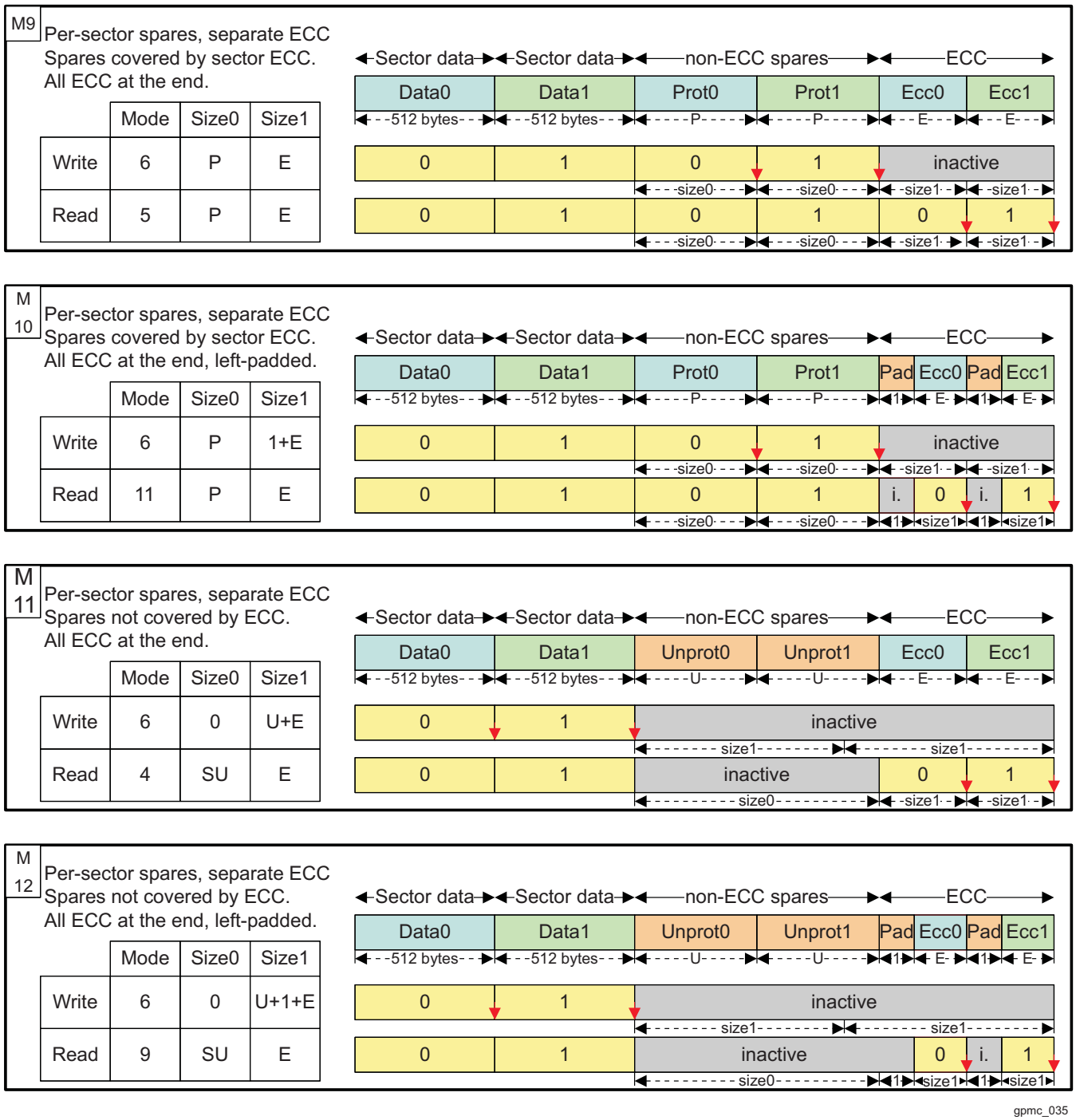


10.3.4.12.3.2.3.3 Per-Sector Spare Mapping, with ECC Separated at the End of the Page

In these schemes, each 512-byte sector of the main area is associated with two sections of the spare area.

- ECC section, all aligned at the end of the page
- Other data section, aligned before the ECCs, each of which may or may not be protected by the ECC for its sector.

Figure 10-52. NAND Page Mapping and ECC: Per-Sector Schemes, With Separate ECC



10.3.4.12.4 Prefetch and Write-Posting Engine

NAND device data access cycles are usually much slower than the IPU system frequency; such NAND read or write accesses issued by the processor affect the overall system performance, especially considering long read or write sequences required for NAND page loading or programming. To minimize this effect on system performance, the GPMC includes a prefetch and write-posting engine, which can be used to read from or write to any chip-select location in a buffered manner.

The prefetch and write-posting engine is a simplified embedded-access requester that presents requests to the access engine on a user-defined chip-select target. The access engine interleaves these requests with any request coming from the L3 interface; as a default, the prefetch and write-posting engine has the lowest priority.

The prefetch and write-posting engine is dedicated to data-stream access (as opposed to random data access); thus, it is primarily dedicated to NAND support. The engine does not include an address generator; the request is limited to chip-select target identification. It includes a 64-byte FIFO associated with a DMA request synchronization line, for optimal DMA-based use.

The prefetch and write-posting engine uses an embedded 64-byte (32 16-bit word) FIFO to prefetch data from the NAND device in read mode (prefetch mode) or to store host data to be programmed into the NAND device in write mode (write-posting mode). The FIFO draining and filling (read and write) can be controlled by a device host processor through interrupt synchronization (an interrupt is triggered whenever a programmable threshold is reached) or by a device DMA module through DMA request synchronization, with a programmable request byte size in prefetch or posting mode.

The prefetch and write-posting engine includes a single memory pool. Therefore, only one mode, read or write, can be used at any given time. In other words, the prefetch and write-posting engine is a single-context engine that can be allocated to only one chip-select at a time for a read prefetch or a write-posting process.

The engine does not support atomic command and address phase programming and is limited to linear memory read or write access. As a consequence, it is limited to NAND data-stream access. The engine relies on the NAND software driver to control block and page opening with the correct data address pointer initialization, before the engine can read from or write to the NAND memory device.

Once started, engine data read and write sequencing is based solely on FIFO location availability and until the total programmed number of bytes is read or written.

Any host-concurrent accesses to a different chip-select are correctly interleaved with ongoing engine accesses. The engine has the lowest priority access so that host accesses to a different chip-select do not suffer a large latency.

A round-robin arbitration scheme can be enabled to ensure minimum bandwidth to the prefetch and write-posting engine in the case of back-to-back direct memory requests to a different chip-select. If the GPMC_PREFETCH_CONFIG1[23] PFPWENROUNDROBIN bit is enabled, the arbitration grants the prefetch and write posting engine access to the GPMC bus for a number of requests programmed in the GPMC_PREFETCH_CONFIG1[19:16] PFPWWEIGHTEDPRIO bit field.

The prefetch/write-posting engine read or write request is routed to the access engine with the chip-select destination ID. After the required arbitration phase, the access engine processes the request as a single access with the data access size equal to the device size specified in the corresponding chip-select configuration.

NOTE: The destination chip-select configuration must be set to the NAND protocol-compatible configuration for which address lines are not used (the address bus is not changed from its current value). Selecting a different chip-select configuration can produce undefined behavior.

10.3.4.12.4.1 General Facts About the Engine Configuration

The engine can be configured only if the GPMC_PREFETCH_CONTROL[0] STARTENGINE bit is deasserted.

The engine must be correctly configured in prefetch or write-posting mode and must be linked to a NAND chip-select before it can be started. The chip-select is linked using the GPMC_PREFETCH_CONFIG1[26:24] ENGINECSSELECTOR bit field.

In prefetch and write-posting modes, the engine uses byte or 16-bit word access requests, respectively, for an 8- or 16-bit-wide NAND device attached to the linked chip-select. The FIFOTHRESHOLD and TRANSFERCOUNT bit fields must be programmed accordingly as a number of bytes.

When the GPMC_PREFETCH_CONFIG1[7] ENABLEENGINE bit is set, the FIFO entry on the L3 interconnect port side is accessible at any address in the associated chip-select memory region. When the ENABLEENGINE bit is set, any host access to this chip-select is rerouted to the FIFO input. Directly accessing the NAND device linked to this chip-select from the host is still possible through the following registers (where i = 0 to 7):

- GPMC_NAND_COMMAND_i
- GPMC_NAND_ADDRESS_i
- GPMC_NAND_DATA_i

The FIFO entry on the L3 interconnect port can be accessed with byte, 16-bit word, or 32-bit word access size, according to little-endian format, even though the FIFO input is 32 bits wide.

The FIFO control is made easier through the use of interrupts or DMA requests associated with the FIFOTHRESHOLD bit field. The GPMC_PREFETCH_STATUS[30:24] FIFOPOINTER bit field monitors the number of available bytes to be read in prefetch mode or the number of free empty slots that can be written in write-posting mode. The GPMC_PREFETCH_STATUS[13:0] COUNTVALUE bit field monitors the number of remaining bytes to be read or written by the engine according to the value of the TRANSFERCOUNT bit field. The FIFOPOINTER and COUNTVALUE bit fields are always expressed as a number of bytes even if a 16-bit-wide NAND device is attached to the linked chip-select.

In prefetch mode, when the FIFOPOINTER equals 0 (that is, the FIFO is empty), a host read access receives the byte last read from the FIFO as its response. In case of 32-bit word or 16-bit word read accesses, the last byte read from the FIFO is copied the required number of times to fit the requested word size. In write-posting mode, when the FIFOPOINTER equals 0 (that is, the FIFO is full), a host write overwrites the last FIFO byte location. There is no underflow or overflow error reporting in the GPMC.

10.3.4.12.4.2 Prefetch Mode

The prefetch mode is selected when the GPMC_PREFETCH_CONFIG1[0] ACCESSMODE bit is cleared.

The NAND software driver must issue the block and page opening (READ) command with the correct data address pointer initialization before the engine can be started to read from the NAND memory device. The engine is started by asserting the GPMC_PREFETCH_CONTROL[0] STARTENGINE bit. The STARTENGINE bit automatically clears when the prefetch process completes.

If required, the ECC calculator engine must be initialized (that is, reset, configured, and enabled) before the prefetch engine is started so that the ECC is computed correctly on all data read by the prefetch engine.

When the GPMC_PREFETCH_CONFIG1[3] SYNCHROMODE bit is cleared, the prefetch engine starts requesting data as soon as the STARTENGINE bit is set. If using this configuration, the host must monitor the NAND device-ready pin so that it sets the STARTENGINE bit only when the NAND device is in a ready state (that is, data is valid for prefetching).

When the SYNCHROMODE bit is set, the prefetch engine starts requesting data when an active-to-inactive WAIT signal transition is detected. The transition detector must be cleared before any transition detection (see Section 10.3.4.12.2.2, *Ready Pin Monitored by Hardware Interrupt*). The GPMC_PREFETCH_CONFIG1[5:4] WAITPINSELECTOR bit field selects which gpmc_wait pin edge detector triggers the prefetch engine in this synchronized mode.

If the STARTENGINE bit is set after the NAND address phase (page opening command), the engine is effectively started only after the actual NAND address phase completion. To prevent GPMC stall during this NAND address phase, set the STARTENGINE bit before NAND address phase completion when in synchronized mode. The prefetch engine starts when an active-to-inactive WAIT signal transition is detected. The STARTENGINE bit is automatically cleared on prefetch process completion.

The prefetch engine issues a read request to fill the FIFO with the amount of data specified by the GPMC_PREFETCH_CONFIG2[13:0] TRANSFERCOUNT bit field.

Table 10-380 describes the prefetch mode configuration.

Table 10-380. Prefetch Mode Configuration

Bit Field	Register	Value	Comments
STARTENGINE	GPMC_PREFETCH_CONTROL[0]	0	Prefetch engine can be configured only if STARTENGINE is set to 0.
ENGINECSSELECTOR	GPMC_PREFETCH_CONFIG1[26:24]	0 to 3	Selects the chip-select associated with a NAND device where the prefetch engine is active.
ACCESSMODE	GPMC_PREFETCH_CONFIG1[0]	0	Selects prefetch mode
FIFOTHRESHOLD	GPMC_PREFETCH_CONFIG1[14:8]		Selects the maximum number of bytes read or written by the host on DMA or interrupt request
TRANSFERCOUNT	GPMC_PREFETCH_CONFIG2[13:0]		Selects the number of bytes to be read or written by the engine to the selected chip-select
SYNCHROMODE	GPMC_PREFETCH_CONFIG1[3]	0/1	Selects when the engine starts the access to the chip-select
WAITPINSELECT	GPMC_PREFETCH_CONFIG1[17:16]	0 to 1	Selects wait pin edge detector (if GPMC_PREFETCH_CONFIG1[3] SYNCHROMODE = 0x1)
ENABLEOPTIMIZEDACCESS	GPMC_PREFETCH_CONFIG1[27]	0/1	See Section 10.3.4.12.4.6, *Optimizing NAND Access Using the Prefetch and Write-Posting Engine*.
CYCLEOPTIMIZATION	GPMC_PREFETCH_CONFIG1[30:28]		Number of clock cycle removed to timing parameters
ENABLEENGINE	GPMC_PREFETCH_CONFIG1[7]	1	Engine enabled
STARTENGINE	GPMC_PREFETCH_CONTROL[0]	1	Starts the prefetch engine

10.3.4.12.4.3 FIFO Control in Prefetch Mode

The FIFO can be drained directly by a device host processor or a DMA module channel.

In draining mode, the FIFO status can be monitored through the GPMC_PREFETCH_STATUS[30:24] FIFOPOINTER bit field or through the GPMC_PREFETCH_STATUS[16] FIFOTHRESHOLDSTATUS bit. The FIFOPOINTER indicates the current number of available data to be read; FIFOTHRESHOLDSTATUS set to 1 indicates that at least FIFOTHRESHOLD bytes are available from the FIFO.

An interrupt can be triggered by the GPMC if the GPMC_IRQENABLE[0] FIFOEVENTENABLE bit is set. The FIFO interrupt event is logged, and the GPMC_IRQSTATUS[0] FIFOEVENTSTATUS bit is set. To clear the interrupt, all the available bytes must be read, or at least enough bytes to get below the programmed FIFO threshold, and the FIFOEVENTSTATUS bit must be cleared to enable further interrupt events. The FIFOEVENTSTATUS bit must always be reset before asserting the FIFOEVENTENABLE bit to clear any out-of-date logged interrupt event. This interrupt generation must be enabled after enabling the STARTENGINE bit.

Prefetch completion can be monitored through the GPMC_PREFETCH_STATUS[13:0] COUNTVALUE bit field. COUNTVALUE indicates the number of currently remaining data to be requested according to the TRANSFERCOUNT value. An interrupt can be triggered by the GPMC when the prefetch process is complete (that is, COUNTVALUE equals 0) if the GPMC_IRQENABLE[1] TERMINALCOUNTEVENTENABLE bit is set. At prefetch completion, the TERMINALCOUNT interrupt event is also logged, and the GPMC_IRQSTATUS[1] TERMINALCOUNTSTATUS bit is set. To clear the interrupt, the TERMINALCOUNTSTATUS bit must be cleared. The TERMINALCOUNTSTATUS bit must always be cleared prior to asserting the TERMINALCOUNTEVENTENABLE bit to clear any out-of-date logged interrupt event.

NOTE: The COUNTVALUE value is valid only when the prefetch engine is active (started), and an interrupt is only triggered when COUNTVALUE reaches 0, that is, when the prefetch engine automatically goes from an active to an inactive state.

The number of bytes to be prefetched (programmed in TRANSFERCOUNT) must be a multiple of the programmed FIFOTHRESHOLD to trigger the correct number of interrupts allowing a deterministic and transparent FIFO control. If this guideline is respected, the number of ISR accesses is always required and the FIFO is always empty after the last interrupt is triggered. In other cases, the TERMINALCOUNT interrupt must be used to read the remaining bytes in the FIFO (the number of remaining bytes being lower than the FIFOTHRESHOLD value).

In DMA draining mode, the GPMC_PREFETCH_CONFIG1[2] DMAMODE bit must be set so that the GPMC issues a DMA hardware request when at least FIFOTHRESHOLD bytes are ready to be read from the FIFO. The DMA channel that owns this DMA request must be programmed so that the number of bytes programmed in FIFOTHRESHOLD is read from the FIFO during the DMA request process. The DMA request is kept active until this number of bytes has effectively been read from the FIFO, and no other DMA request can be issued until the ongoing active request is complete.

In prefetch mode, the TERMINALCOUNT event is also a source of DMA requests if the number of bytes to be prefetched is not a multiple of FIFOTHRESHOLD, the remaining bytes in the FIFO can be read by the DMA channel using the last DMA request. This assumes that the number of remaining bytes to be read is known and controlled through the DMA channel programming model.

Any potentially active DMA request is cleared when the prefetch engine goes from inactive-to-active prefetch (the STARTENGINE bit is set to 1). The associated DMA channel must always be enabled after setting the STARTENGINE bit so that the out-of-date active DMA request does not trigger spurious DMA transfers.

10.3.4.12.4.4 Write-Posting Mode

The write-posting mode is selected when the GPMC_PREFETCH_CONFIG1[0] ACCESSMODE bit is set.

The NAND software driver must issue the correct address pointer initialization command (page program) before the engine can start writing data into the NAND memory device. The engine starts when the GPMC_PREFETCH_CONTROL[0] STARTENGINE bit is set to 1. The STARTENGINE bit clears automatically when posting completes. When all data have been written to the NAND memory device, the NAND software driver must issue the second cycle program command and monitor the status for programming process completion (adding ECC handling, if required).

If used, the ECC calculator engine must be started (configured, reset, and enabled) before the posting engine is started so that the ECC parities are calculated properly on all data written by the prefetch engine to the associated chip-select.

In write-posting mode, the GPMC_PREFETCH_CONFIG1[3] SYNCHROMODE bit must be cleared so that posting starts as soon as the STARTENGINE bit is set and the FIFO is not empty.

If the STARTENGINE bit is set after the NAND address phase (page program command), the STARTENGINE setting is effective only after the actual NAND command completion. To prevent GPMC stall during this NAND command phase, set the STARTENGINE bit field before the NAND address completion and ensure that the associated DMA channel is enabled after the NAND address phase.

The posting engine issues a write request when valid data are available from the FIFO and until the programmed GPMC_PREFETCH_CONFIG2[13:0] TRANSFERCOUNT accesses are complete.

The STARTENGINE bit clears automatically when posting completes. When all data have been written to the NAND memory device, the NAND software driver must issue the second cycle program command and monitor the status for programming process completion. The closing program command phase must be issued only when the full NAND page has been written into the NAND flash write buffer, including the spare area data and the ECC parities, if used.

Table 10-381 describes the write-posting configuration.

Table 10-381. Write-Posting Mode Configuration

Bit Field	Register	Value	Comments
STARTENGINE	GPMC_PREFETCH_CONTROL[0]	0	Write-posting engine can be configured only if STARTENGINE is set to 0.

Table 10-381. Write-Posting Mode Configuration (continued)

Bit Field	Register	Value	Comments
ENGINECSSELECTOR	GPMC_PREFETCH_CONFIG1[26:24]	0 to 3	Selects the chip-select associated with a NAND device where the prefetch engine is active
ACCESSMODE	GPMC_PREFETCH_CONFIG1[0]	1	Selects write-posting mode
FIFOTHRESHOLD	GPMC_PREFETCH_CONFIG1[14:8]		Selects the maximum number of bytes read or written by the host on DMA or interrupt request
TRANSFERCOUNT	GPMC_PREFETCH_CONFIG2[13:0]		Selects the number of bytes to be read or written by the engine from/to the selected chip-select
SYNCHROMODE	GPMC_PREFETCH_CONFIG1[3]	0	Engine starts the access to chip-select as soon as STARTENGINE is set.
ENABLEOPTIMIZEDACCESS	GPMC_PREFETCH_CONFIG1[27]	0/1	See Section 10.3.4.12.4.6, *Optimizing NAND Access Using the Prefetch and Write-Posting Engine*.
CYCLEOPTIMIZATION	GPMC_PREFETCH_CONFIG1[30:28]		
ENABLEENGINE	GPMC_PREFETCH_CONFIG1[7]	1	Engine enabled
STARTENGINE	GPMC_PREFETCH_CONTROL[0]	1	Starts the prefetch engine

10.3.4.12.4.5 FIFO Control in Write-Posting Mode

The FIFO can be filled directly by a device host processor or a DMA module channel.

In filling mode, the FIFO status can be monitored through the FIFOPOINTER or through the GPMC_PREFETCH_STATUS[16] FIFOTHRESHOLDSTATUS bit. FIFOPOINTER indicates the current number of available free byte places in the FIFO, and the FIFOTHRESHOLDSTATUS bit, when set, indicates that at least FIFOTHRESHOLD free byte places are available in the FIFO.

An interrupt can be issued by the GPMC if the GPMC_IRQENABLE[0] FIFOEVENTENABLE bit is set. When the interrupt is fired, the GPMC_IRQSTATUS[0] FIFOEVENTSTATUS bit is set. To clear the interrupt, enough bytes must be written to fill the FIFO or to get below the programmed threshold, and the FIFOEVENTSTATUS bit must be cleared to get further interrupt events. The FIFOEVENTSTATUS bit must always be cleared before asserting the FIFOEVENTENABLE bit to clear any out-of-date logged interrupt event. This interrupt must be enabled after enabling the STARTENGINE bit.

The posting completion can be monitored through the GPMC_PREFETCH_STATUS[13:0] COUNTVALUE bit field. COUNTVALUE indicates the current number of remaining data to be written based on the value of the TRANSFERCOUNT bit field. An interrupt is issued by the GPMC when the write-posting process completes (that is, COUNTVALUE equal to 0) if the GPMC_IRQENABLE[1] TERMINALCOUNTEVENTENABLE bit is set. When the interrupt is fired, the GPMC_IRQSTATUS[1] TERMINALCOUNTSTATUS bit is set. To clear the interrupt, the TERMINALCOUNTSTATUS bit must be cleared. The TERMINALCOUNTSTATUS bit must always be cleared before asserting the TERMINALCOUNTEVENTENABLE bit to clear any out-of-date logged interrupt event.

NOTE: The value of the COUNTVALUE bit field is valid only if the write-posting engine is active and started, and an interrupt is issued only when COUNTVALUE reaches 0; that is, when the posting engine automatically goes from active to inactive.

In DMA filling mode, the DMAMode bit field in the GPMC_PREFETCH_CONFIG1[2] DMAMODE bit must be set so that the GPMC issues a DMA hardware request when at least FIFOTHRESHOLD bytes-free places are available in the FIFO. The DMA channel that owns this DMA request must be programmed so that a number of bytes equal to the value programmed in the FIFOTHRESHOLD bit field are written into the FIFO during the DMA access. The DMA request remains active until the associated number of bytes has effectively been written into the FIFO, and no other DMA request can be issued until the ongoing active request completes.

Any potentially active DMA request is cleared when the prefetch engine goes from inactive-to-active prefetch (STARTENGINE set to 1). The associated DMA channel must always be enabled after setting the STARTENGINE bit so that an out-of-date active DMA request does not trigger spurious DMA transfers.

In write-posting mode, the DMA or IPU fills the FIFO with no consideration for the associated byte enables. Any byte stored in the FIFO is written into the memory device.

10.3.4.12.4.6 Optimizing NAND Access Using the Prefetch and Write-Posting Engine

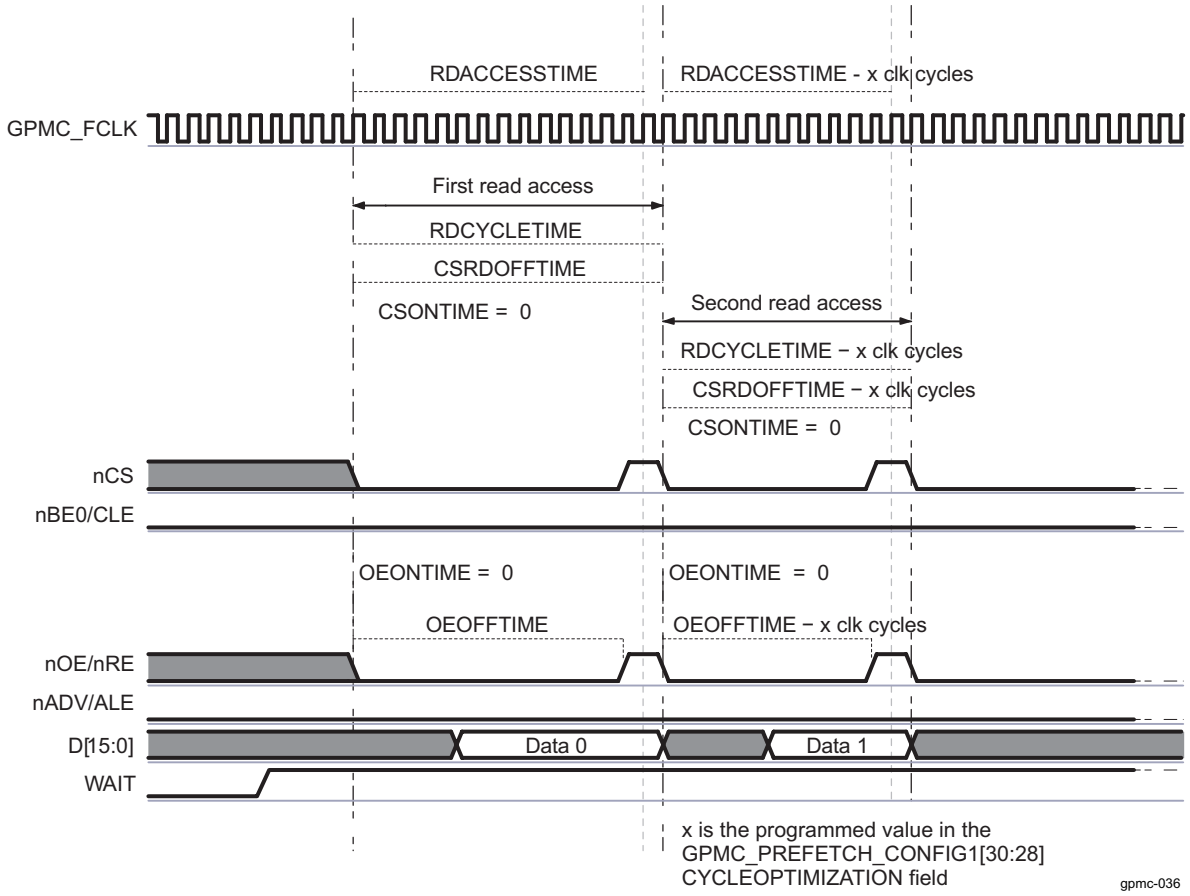
Access time to a NAND memory device can be optimized for back-to-back accesses if the associated nCS signal is not deasserted between accesses. The GPMC access engine can track prefetch engine accesses to optimize the access timing parameter programmed for the allocated chip-select, if no accesses to other chip-selects (that is, interleaved accesses) occur. Similarly, the access engine also eliminates CYCLE2CYCLEDELAY even if CYCLE2CYCLESAMECSEN is set. This capability is limited to the prefetch and write-posting engine accesses, and accesses to a NAND memory device (through the defined chip-select memory region or through the GPMC_NAND_DATA_i location, where i = 0 to 7) are never optimized.

The GPMC_PREFETCH_CONFIG1[27] ENABLEOPTIMIZEDACCESS bit must be set to enable optimized accesses. To optimize access time, the GPMC_PREFETCH_CONFIG1[30:28] CYCLEOPTIMIZATION bit field defines the number of GPMC_FCLK cycles to be suppressed from the following timing parameters:

- RDCYCLETIME
- WRCYCLETIME
- RDACCESSTIME
- WRACCESSTIME
- CSOFFTIME
- ADVOFFTIME
- OEOFFTIME
- WEOFFTIME

Figure 10-53 shows that in the case of back-to-back accesses to the NAND flash through the prefetch engine, CYCLE2CYCLESAMECSEN is forced to 0 when using optimized accesses. The first access uses the regular timing settings for this chip-select. All accesses after this one use settings reduced by x clock cycles, x being defined by the GPMC_PREFETCH_CONFIG1[30:28] CYCLEOPTIMIZATION bit field.

Figure 10-53. NAND Read Cycle Optimization Timing Description



10.3.4.12.4.7 Interleaved Accesses Between Prefetch and Write-Posting Engine and Other Chip-Selects

Any on-going read or write access from the prefetch and write-posting engine is completed before an access to any other chip-select can be initiated. As a default, the arbiter uses a fixed-priority algorithm, and the prefetch and write-posting engine has the lowest priority. The maximum latency added to access starting time in this case equals the RDCYCLETIME or WRCYCLETIME (optimized or not) plus the requested BUSTURNAROUND delay for bus turnaround completion programmed for the chip-select to which the NAND device is connected.

Alternatively, a round-robin arbitration can be used to prioritize accesses to the external bus. This arbitration scheme is enabled by setting the GPMC_PREFETCH_CONFIG1[23] PFPWENROUNDROBIN bit. When a request to another chip-select is received while the prefetch and write-posting engine is active, priority is given to the new request. The request processed thereafter is the prefetch and write-posting engine request, even if another interconnect request is passed in the mean time. The engine keeps control of the bus for an additional number of requests programmed in the GPMC_PREFETCH_CONFIG1[19:16] PFPWWEIGHTEDPRIO bit field. Control is then passed to the direct interconnect request.

As an example, the round-robin arbitration scheme is selected with PFPWWEIGHTEDPRIO set to 0x2. Considering that the prefetch and write-posting engine and the interconnect interface are always requesting access to the external interface, the GPMC grants priority to the direct interconnect access for one request. The GPMC then grants priority to the engine for three requests, and finally back to the direct interconnect access, until the arbiter is reset when one of the two initiators stops initiating requests.

10.3.5 GPMC Basic Programming Model

10.3.5.1 GPMC High-Level Programming Model Overview

The goal of the basic high-level programming model is to introduce a top-down approach to users that need to configure the GPMC module.

Figure 10-54 and Table 10-382 through Table 10-384 show a programming model top-level diagram for the GPMC, and a description of each step. Each block of the diagram is described in one of the following sections through a set of registers to configure.

Figure 10-54. Programming Model Top-Level Diagram

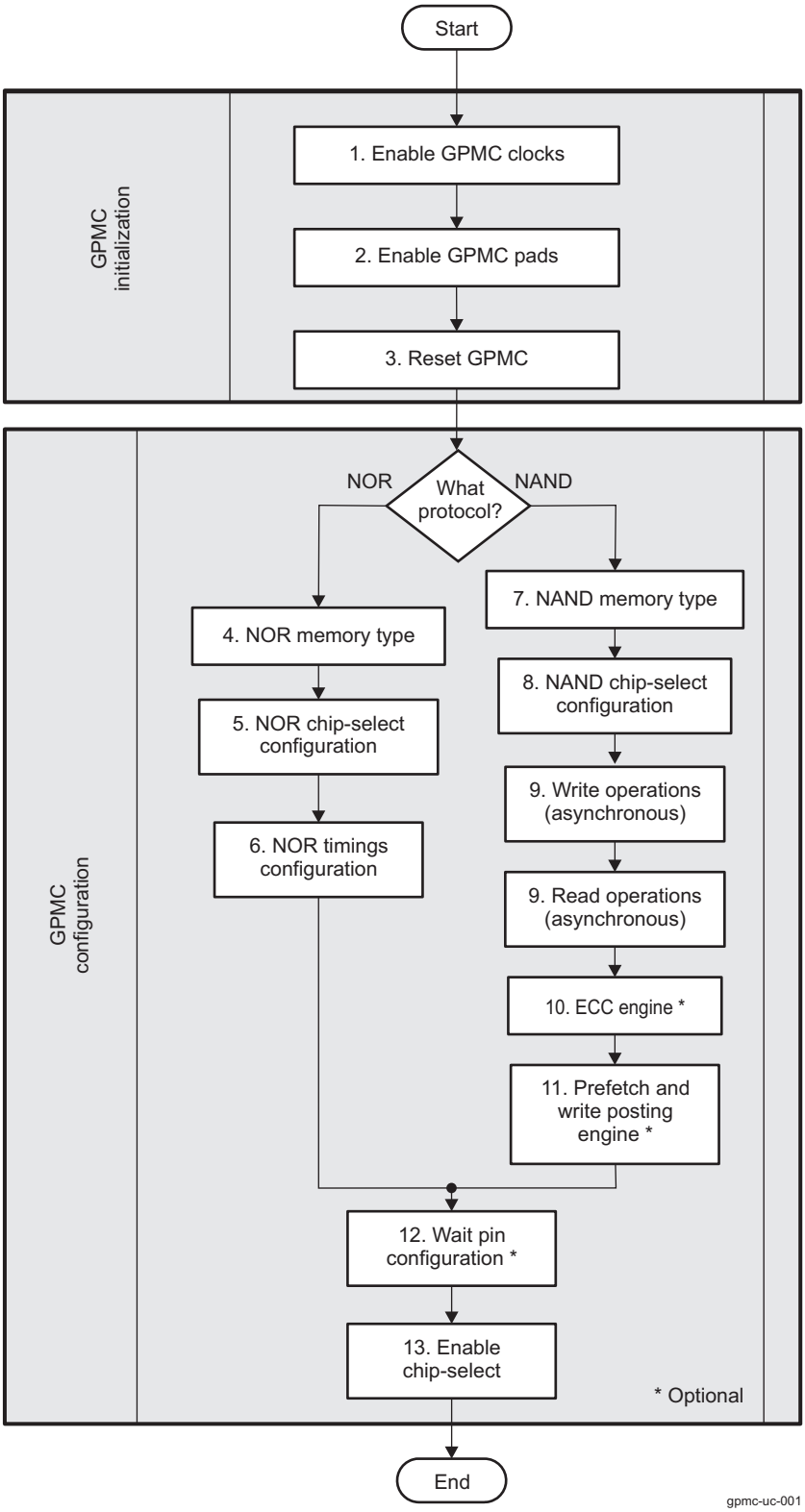


Table 10-382. GPMC Initialization

Step	Description
Enable GPMC clocks.	Module interface and functional clocks must be enabled. See Chapter 3, *Power, Reset, and Clock Management*.
Enable GPMC pads.	Module-specific pad multiplexing and configuration must be set in the control module. See Section 13.4.6.1, *Pad Configuration Registers* in Chapter 13, *Control Module*.
Reset GPMC.	See Table 10-385.

Table 10-383. GPMC Configuration in NOR Mode

Step	Description
NOR Memory Type	See Table 10-386.
NOR Chip-Select Configuration	See Table 10-387.
NOR Timings Configuration	See Table 10-388.
Wait Pin Configuration	See Table 10-396.
Enable Chip-Select	See Table 10-397.

Table 10-384. GPMC Configuration in NAND Mode

Step	Description
NAND Memory Type	See Table 10-391.
NAND Chip-Select Configuration	See Table 10-392.
Write Operations (Asynchronous)	See Table 10-393.
Read Operations (Asynchronous)	See Table 10-393.
ECC Engine	See Table 10-394.
Prefetch and Write-Posting Engine	See Table 10-395.
Wait Pin Configuration	See Table 10-396.
Enable Chip-Select	See Table 10-397.

10.3.5.2 GPMC Initialization

Table 10-413 describes the settings required to prepare the GPMC; that is enabling its clock and pads, and proceeding to a GPMC reset.

Table 10-385. Reset GPMC

Subprocess Name	Register/Bit Field	Value
Start a software reset.	GPMC_SYSCONFIG[1] SOFTRESET	0x1
Wait until	GPMC_SYSSTATUS[0] RESETDONE =	0x1

10.3.5.3 GPMC Configuration in NOR Mode

This section gives a generic configuration for parameters related to the NOR memory connected to the GPMC.

Table 10-386. NOR Memory Type

Subprocess Name	Register / Bit Field	Value
Set the NOR protocol.	GPMC_CONFIG1_i[11:10] DEVICETYPE	0x0
Set a device size.	GPMC_CONFIG1_i[13:12] DEVICESIZE	x
Select an address and data multiplexing protocol.	GPMC_CONFIG1_i[9] MUXADDDATA	x
Set the attached device page length.	GPMC_CONFIG1_i[24:23] ATTACHEDDEVICEPAGELENGTH	x

Table 10-386. NOR Memory Type (continued)

Subprocess Name	Register / Bit Field	Value
Set the wrapping burst capabilities.	GPMC_CONFIG1_i[31] WRAPBURST	x
Select a timing signals latencies factor.	GPMC_CONFIG1_i[4] TIMEPARAGRANULARITY	x
Select an output clock frequency (1).	GPMC_CONFIG1_i[1:0] GPMCFCLKDIVIDER	x
Choose an output clock activation time (1).	GPMC_CONFIG1_i[26:25] CLKACTIVATIONTIME	x
Set a single or multiple access for read operations (1).	GPMC_CONFIG1_i[30] READMULTIPLE	x
Set a synchronous or asynchronous mode for read operations.	GPMC_CONFIG1_i[29] READTYPE	x
Set a single or multiple access for write operations.	GPMC_CONFIG1_i[28] WRITEMULTIPLE	x
Set a synchronous or asynchronous mode for write operations.	GPMC_CONFIG1_i[27] WRITETYPE	x

(1) Applies only to synchronous configurations (or nonmultiplexed asynchronous for multiple access one)

Table 10-387. NOR Chip-Select Configuration

Subprocess Name	Register/Bit Field	Value
Select the chip-select base address.	GPMC_CONFIG7_i[5:0] BASEADDRESS	x
Select the chip-select mask address.	GPMC_CONFIG7_i[11:8] MASKADDRESS	x

Table 10-388. NOR Timings Configuration

Subprocess Name	Register/Bit Field	Value
Configure adequate timing parameters in various memory modes.	See Section 10.3.5.6, *GPMC Timing Parameters*	

Table 10-389. Wait Pin Configuration

Subprocess Name	Register/Bit Field	Value
Enable or disable wait pin monitoring for read operations.	GPMC_CONFIG1_i[22] WAITREADMONITORING	x
Enable or disable wait pin monitoring for write operations.	GPMC_CONFIG1_i[21] WAITWRITEMONITORING	x
Select a wait pin monitoring time.	GPMC_CONFIG1_i[19:18] WAITMONITORINGTIME	x
Choose the input wait pin for the chip-select.	GPMC_CONFIG1_i[17:16] WAITPINSELECT	x

Table 10-390. Enable Chip-Select

Subprocess Name	Register/Bit Field	Value
When all parameters are configured, enable the chip-select.	GPMC_CONFIG7_i[6] CSVALID	x

10.3.5.4 GPMC Configuration in NAND Mode

This section gives a generic configuration for parameters related to the NAND memory connected to the GPMC.

Table 10-391. NAND Memory Type

Subprocess Name	Register/Bit Field	Value
Set the NAND protocol.	GPMC_CONFIG1_i[11:10] DEVICETYPE	0x2
Set a device size.	GPMC_CONFIG1_i[13:12] DEVICESIZE	x
Set the address and data multiplexing protocol to non-multiplexed attached device.	GPMC_CONFIG1_i[9] MUXADDDATA	0x0
Select a timing signals latencies factor.	GPMC_CONFIG1_i[4] TIMEPARAGRANULARITY	x

Table 10-391. NAND Memory Type (continued)

Subprocess Name	Register/Bit Field	Value
Set a synchronous or asynchronous mode and a single or multiple access for read and write operations.	See Section 10.3.5.5, *Set Memory Access*.	x

Table 10-392. NAND Chip-Select Configuration

Subprocess Name	Register/Bit Field	Value
Select the chip-select base address.	GPMC_CONFIG7_i[5:0] BASEADDRESS	x
Select the chip-select minimum granularity (16MiB).	GPMC_CONFIG7_i[11:8] MASKADDRESS	x

Table 10-393. Asynchronous Read and Write Operations

Subprocess Name	Register/Bit Field	Value
Configure adequate timing parameters in asynchronous modes	See Section 10.3.5.6, *GPMC Timing Parameters*.	

Table 10-394. ECC Engine

Subprocess Name	Register/Bit Field	Value
Select the ECC result register where the first ECC computation is stored (applies only to Hamming).	GPMC_ECC_CONTROL[3:0] ECCPOINTER	x (1)
Clear all ECC result registers.	GPMC_ECC_CONTROL[8] ECCCLEAR	Write 1 to clear.
Define ECCSIZE0 and ECCSIZE1.	GPMC_ECC_SIZE_CONFIG[19:12] ECCSIZE0 and [29:22] ECCSIZE1	x (2)
Select the size of each of the 9 result registers (size specified by ECCSIZE0 or ECCSIZE1).	GPMC_ECC_SIZE_CONFIG[j-1] ECCjRESULTSIZE where j = 1 to 9	x
Select the chip-select where ECC is computed.	GPMC_ECC_CONFIG[3:1] ECCCS	x
Select the Hamming code or BCH code ECC algorithm in use.	GPMC_ECC_CONFIG[16] ECCALGORITHM	x
Select word size for ECC calculation.	GPMC_ECC_CONFIG[7] ECC16B	x
If the BCH code is used, Set an error correction capability and Select a number of sectors to process.	GPMC_ECC_CONFIG[13:12] ECCBCHTSEL and GPMC_ECC_CONFIG[6:4] ECCTOPSECTOR	x
Enable the ECC computation.	GPMC_ECC_CONFIG[0] ECCENABLE	0x1

(1) This parameter depends on the numbers of sectors in a page.
(2) Depends on the size of each sector in the NAND page

Table 10-395. Prefetch and Write-Posting Engine

Subprocess Name	Register/Bit Field	Value
Disable the engine before configuration.	GPMC_PREFETCH_CONTROL[0] STARTENGINE	0x0
Select the chip-select associated with a NAND device where the prefetch engine is active.	GPMC_PREFETCH_CONFIG1[26:24] ENGINECSSELECTOR	x
Select access direction through prefetch engine, read or write.	GPMC_PREFETCH_CONFIG1[0] ACCESSMODE	x
Select the threshold used to issue a DMA request.	GPMC_PREFETCH_CONFIG1[14:8] FIFOTHRESHOLD	x
Select DMA synchronized mode or software manual mode.	GPMC_PREFETCH_CONFIG1[2] DMAMODE	x
Select if the engine immediately starts accessing the memory upon STARTENGINE assertion or if hardware synchronization based on a WAIT signal is used.	GPMC_PREFETCH_CONFIG1[3] SYNCHROMODE	x
Select which wait pin edge detector should start the engine in synchronized mode.	GPMC_PREFETCH_CONFIG1[5:4] WAITPINSELECTOR	x

Table 10-395. Prefetch and Write-Posting Engine (continued)

Subprocess Name	Register/Bit Field	Value
Enter a number of clock cycles removed to timing parameters (for all back-to-back accesses to the NAND flash except the first one).	GPMC_PREFETCH_CONFIG1[30:28] CYCLEOPTIMIZATION	x
Enable the prefetch postwrite engine.	GPMC_PREFETCH_CONFIG1[7] ENABLEENGINE	0x1
Select the number of bytes to be read or written by the engine to the selected chip-select.	GPMC_PREFETCH_CONFIG2[13:0] TRANSFERCOUNT	x
Start the prefetch engine.	GPMC_PREFETCH_CONTROL[0] STARTENGINE	0x1

Table 10-396. Wait Pin Configuration

Subprocess Name	Register/Bit Field	Value
Selects when the engine starts the access to chip-select.	GPMC_PREFETCH_CONFIG1[3] SYNCHROMODE	x
Select which wait pin edge detector should start the engine in synchronized mode.	GPMC_PREFETCH_CONFIG1[5:4] WAITPINSELECTOR	x

Table 10-397. Enable Chip-Select

Subprocess Name	Register/Bit Field	Value
When all parameters are configured, enable the chip-select.	GPMC_CONFIG7_i[6] CSVALID	x

10.3.5.5 Set Memory Access

This section describes the bit field to configure to set the GPMC in various memory modes. Table 10-398 and Table 10-399 provide check lists for mode parameters and access type parameters, respectively.

Table 10-398. Mode Parameters Check List

Register	Bit	Name	Asynchronous				Synchronous			
			Single Read Access	Single Write Access	Multiple Read (Page) Access	Multiple Write (Page) Access	Single Read Access	Single Write Access	Multiple Read (Burst) Access	Multiple Write (Burst) Access
GPMC_CONFIG1_i	30	READMULTIPLE	0x0	–	0x1 (1)	N/S	0x0	–	0x1	–
GPMC_CONFIG1_i	29	READTYPE	0x0	–	0x0 (1)	N/S	0x1	–	0x1	–
GPMC_CONFIG1_i	28	WRITEMULTIPLE	–	0x0	- (1)	N/S	–	0x0	–	0x1
GPMC_CONFIG1_i	27	WRITETYPE	–	0x0	- (1)	N/S	–	0x1	–	0x1

(1) Multiple read is not supported in address/data-multiplexed and AAD-multiplexed modes. Multiple read is supported in nonmultiplexed mode.

Table 10-399. Access Type Parameters Check List

Register	Bit	Name	Access Type		
			Nonmultiplexed	Address/ Data-Multiplexed	AAD-Multiplexed
GPMC_CONFIG1_i	9:8	MUXADDDATA	0x0	0x2	0x1

10.3.5.6 GPMC Timing Parameters

Figure 10-55 shows a programming model diagram for the NOR interfacing timing parameters.

Figure 10-55. NOR Interfacing Timing Parameters Diagram

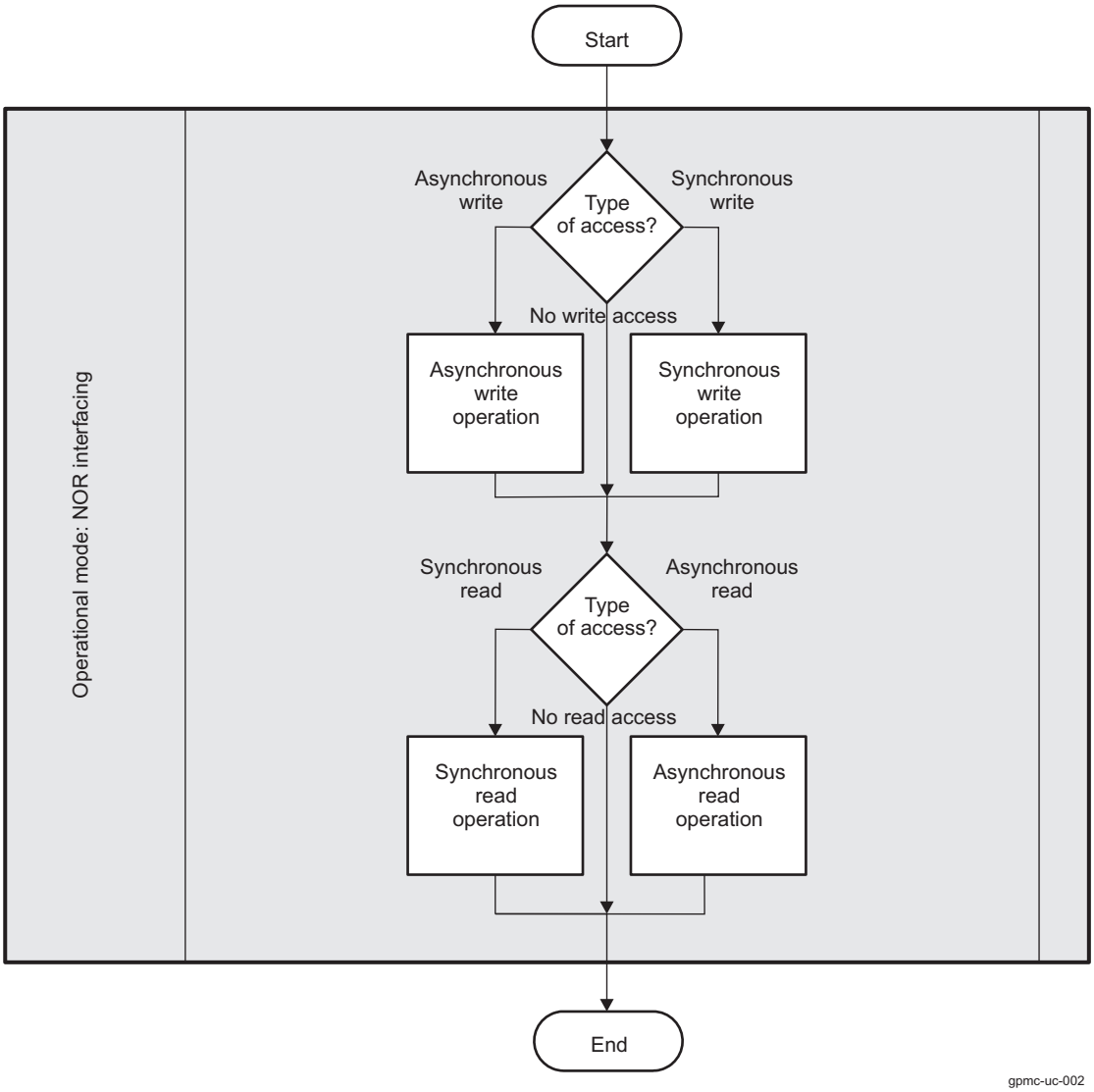


Table 10-400 lists the bit fields to configure adequate timing parameters in various memory modes.

Table 10-400. Timing Parameters

			Asynchronous			Synchronous				Access Type		
Register	Bit	Name	Single Read Access	Single Write Access	Multiple Read (Page) access	Single Read Access	Single Write Access	Multiple Read (Burst) Access	Multiple Write (Burst) Access	Non-multiplexed	Address / Data-Multiplexed	AAD Multiplexed
GPMC_CONFIG1_i	9	MUXADDDATA	y	y	y	y	y	y	y	y	y	y
GPMC_CONFIG1_i	29	READTYPE	y		y	y		y		y	y	y
GPMC_CONFIG1_i	30	READMULTIPLE	y		y	y		y		y	y	y
GPMC_CONFIG1_i	27	WRITETYPE		y			y		y	y	y	y
GPMC_CONFIG1_i	28	WRITEMULTIPLE		y			y		y	y	y	y
GPMC_CONFIG1_i	31	WRAPBURST						y	y	y	y	y
GPMC_CONFIG1_i	26:25	CLKACTIVATIONTIME				y	y	y	y	y	y	y
GPMC_CONFIG1_i	19:18	WAITMONITORINGTIME	y	y	y	y	y	y	y	y	y	y
GPMC_CONFIG1_i	4	TIMEPARAGRANULARITY	y	y	y	y	y	y	y	y	y	y
GPMC_CONFIG2_i	20:16	CSWROFFTIME		y			y		y	y	y	y
GPMC_CONFIG2_i	12:8	CSRDOFFTIME	y		y	y		y		y	y	y
GPMC_CONFIG2_i	7	CSEXTRADELAY	y	y	y	y	y	y	y	y	y	y
GPMC_CONFIG2_i	3:0	CSONTIME	y	y	y	y	y	y	y	y	y	y
GPMC_CONFIG3_i	30:28	ADVAADMUXWROFFTIME		y			y		y			y
GPMC_CONFIG3_i	30:29	ADVAADMUXRDOFFTIME	y		y	y		y				y
GPMC_CONFIG3_i	6:4	ADVAADMUXONTIME	y	y	y	y	y	y	y			y
GPMC_CONFIG3_i	20:16	ADVWROFFTIME		y			y		y	y	y	y
GPMC_CONFIG3_i	12:8	ADVRDOFFTIME	y		y	y		y		y	y	y
GPMC_CONFIG3_i	7	ADVEXTRADELAY	y	y	y	y	y	y	y	y	y	y
GPMC_CONFIG3_i	3:0	ADVONTIME	y	y	y	y	y	y	y	y	y	y
GPMC_CONFIG4_i	15:13	OEAADMUXOFFTIME	y	y	y	y	y	y	y			y
GPMC_CONFIG4_i	6:4	OEAADMUXONTIME	y	y	y	y	y	y	y			y
GPMC_CONFIG4_i	28:24	WEOFFTIME		y			y		y	y	y	y
GPMC_CONFIG4_i	23	WEEXTRADELAY		y			y		y	y	y	y
GPMC_CONFIG4_i	19:16	WEONTIME		y			y		y	y	y	y
GPMC_CONFIG4_i	12:8	OEOFFTIME	y		y	y		y		y	y	y
GPMC_CONFIG4_i	7	OEEXTRADELAY	y		y	y		y		y	y	y
GPMC_CONFIG4_i	3:0	OEONTIME	y		y	y		y		y	y	y
GPMC_CONFIG5_i	27:24	PAGEBURSTACCESSTIME			y			y	y	y	y	y
GPMC_CONFIG5_i	20:16	RDACCESSTIME	y		y	y		y		y	y	y

Table 10-400. Timing Parameters (continued)

			Asynchronous			Synchronous				Access Type		
GPMC_CONFIG5_i	12:8	WRCYCLETIME		y			y		y	y	y	y
GPMC_CONFIG5_i	4:0	RDCYCLETIME	y		y	y		y		y	y	y
GPMC_CONFIG6_i	28:24	WRACCESSTIME		y			y		y	y	y	y
GPMC_CONFIG6_i	19:16	WRDATAONADMUXBUS		y			y		y		y	y
GPMC_CONFIG6_i	11:8	CYCLE2CYCLEDELAY	y	y	y	y	y	y	y	y	y	y
GPMC_CONFIG6_i	7	CYCLE2CYCLESAMECSEN	y	y	y	y	y	y	y	y	y	y
GPMC_CONFIG6_i	6	CYCLE2CYCLEDIFFCSEN	y	y	y	y	y	y	y	y	y	y
GPMC_CONFIG6_i	3:0	BUSTURNAROUND	y	y	y	y	y	y	y	y	y	y
GPMC_CONFIG7_i	6	CSVALID	y	y	y	y	y	y	y	y	y	y

10.3.5.6.1 GPMC Timing Parameters Formulas

This section is intended to help the user calculate the GPMC timing bit field values. Formulas are not listed exhaustively.

The section describes:

- NAND flash interface timing parameters formulas
- Synchronous NOR flash timing parameters formulas
- Asynchronous NOR flash timing parameters formulas

For complete information, such as OPP and board effects on timings, see the device data manual.

10.3.5.6.1.1 NAND Flash Interface Timing Parameters Formulas

This section lists formulas to calculate NAND timing parameters. This is the case when GPMC_CONFIG1_i[11:10] DEVICETYPE = 0x2. Table 10-401 describes the NAND timing parameters.

Table 10-401. NAND Formulas Description

Configuration Parameter	Unit	Description
A	ns	Pulse duration – gpmc_wen valid time
B	ns	Delay time – gpmc_cs valid to gpmc_wen valid
C	ns	Delay time – gpmc_ben0/gpmc_advn_ale high to gpmc_wen valid
D	ns	Delay time – gpmc_ad[15:0] valid to gpmc_wen valid
E	ns	Delay time – gpmc_wen invalid to gpmc_ad[15:0] invalid
F	ns	Delay time – gpmc_wen invalid to gpmc_ben0/gpmc_advn_ale invalid
G	ns	Delay time – gpmc_wen invalid to gpmc_cs invalid
H	ns	Cycle time – Write cycle time
I	ns	Delay time – gpmc_cs valid to gpmc_oen_ren valid
J	ns	Setup time – gpmc_ad[15:0] valid to gpmc_oen_ren invalid
K	ns	Pulse duration – gpmc_oen_ren valid time
L	ns	Cycle time – Read cycle time
M	ns	Delay time – gpmc_oen_ren invalid to gpmc_cs invalid

The configuration parameters are calculated through the following formulas. For more information, see the device data manual.

A = (WEOffTime – WEOnTime) * (TimeParaGranularity + 1) * GPMC_FCLK period

B = ((WEOnTime – CSOnTime) * (TimeParaGranularity + 1) + 0.5 * (WEExtraDelay – CSExtraDelay)) * GPMC_FCLK period

C = ((WEOnTime – ADVOnTime) * (TimeParaGranularity + 1) + 0.5 * (WEExtraDelay - ADVExtraDelay)) * GPMC_FCLK period

D = (WEOnTime * (TimeParaGranularity + 1) + 0.5 * WEExtraDelay) * GPMC_FCLK period

E = (WrCycleTime – WEOffTime * (TimeParaGranularity + 1) – 0.5 * WEExtraDelay) * GPMC_FCLK period

F = (ADVWrOffTime – WEOffTime * (TimeParaGranularity + 1) + 0.5 * (ADVExtraDelay – WEExtraDelay) * GPMC_FCLK period

G = (CSWrOffTime – WEOffTime * (TimeParaGranularity + 1) + 0.5 * (CSExtraDelay – WEExtraDelay) * GPMC_FCLK period

H = WrCycleTime * (1 + TimeParaGranularity) * GPMC_FCLK period

I = ((OEOnTime – CSOnTime) * (TimeParaGranularity + 1) + 0.5 * (OEExtraDelay – CSExtraDelay)) * GPMC_FCLK period

J = ((RdAccessTime – OEOffTime) * (TimeParaGranularity + 1) – 0.5 * OEExtraDelay)) * GPMC_FCLK period

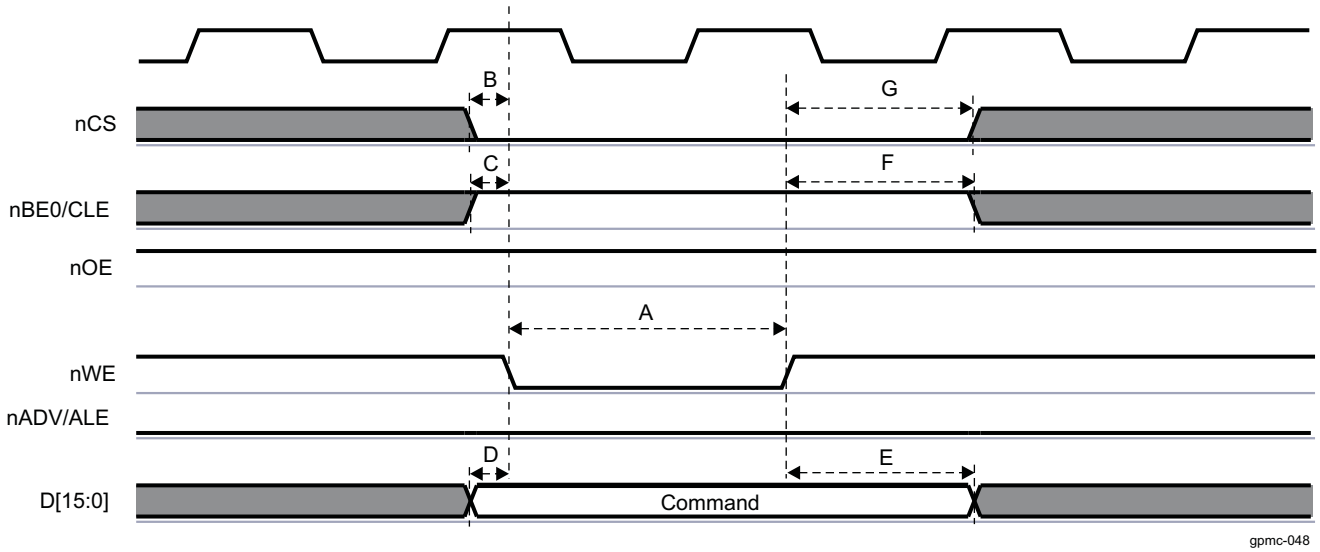
K = (OEOffTime – OEOnTime) * (1 + TimeParaGranularity) * GPMC_FCLK period

L = RdCycleTime * (1 + TimeParaGranularity) * GPMC_FCLK period

M = (CSRdOffTime – OEOffTime * (TimeParaGranularity + 1) + 0.5 * (CSExtraDelay - OEExtraDelay) * GPMC_FCLK period

Figure 10-56 shows a simplified example of command latch cycle timing where formulas are associated with signal waves.

Figure 10-56. NAND Command Latch Cycle Timing Simplified Example



10.3.5.6.1.2 Synchronous NOR Flash Timing Parameters Formulas

This section lists all formulas to calculate synchronous NOR timing parameters. This is the case when GPMC_CONFIG1_i[11:10] DEVICETYPE = 0x0 and when READTYPE or WRITETYPE are set to synchronous mode. Table 10-402 describes the synchronous NOR formulas.

Table 10-402. Synchronous NOR Formulas Description

Configuration Parameter	Unit	Description
A	ns	Pulse duration – gpmc_cs low
B	ns	Delay time – address bus valid to gpmc_clk first edge
		Delay time – gpmc_ben0/gpmc_ben1 valid to gpmc_clk first edge
C	ns	Pulse duration – gpmc_ben0/gpmc_ben1 low
D	ns	Delay time – gpmc_clk rising edge to gpmc_ben0/gpmc_ben1 invalid
		Delay time – gpmc_clk rising edge to gpmc_advn_ale invalid
E	ns	Delay time – gpmc_clk rising edge to gpmc_cs invalid
		Delay time – gpmc_clk rising edge to gpmc_oen_ren invalid
F	ns	Delay time – gpmc_clk rising edge to gpmc_cs transition
G	ns	Delay time – gpmc_clk rising edge to gpmc_advn_ale transition
H	ns	Delay time – gpmc_clk rising edge to gpmc_oen_ren transition
I	ns	Delay time – gpmc_clk rising edge to gpmc_wen transition
J	ns	Delay time – gpmc_clk rising edge to gpmc_ad data bus transition
		Delay time – gpmc_clk rising edge to gpmc_ben0/gpmc_ben1 transition
K	ns	Pulse duration – gpmc_advn_ale low
L	ns	Delay time – gpmc_wait invalid to first data latching gpmc_clk edge

The configuration parameters are calculated through the following formulas. For more information, see the device data manual.

1. For single read accesses:

   A = (CSRDOFFTIME – CSONTIME) * (TIMEPARAGRANULARITY + 1) * GPMC_FCLK period
   C = RDCYCLETIME * (TIMEPARAGRANULARITY + 1) * GPMC_FCLK period
   D = (RDCYCLETIME – RDACCESSTIME) * GPMC_FCLK period
   E = (CSRDOFFTIME – RDACCESSTIME) * GPMC_FCLK period

2. For burst read accesses (where n is the page burst access number):

   A = (CSRDOFFTIME – CSONTIME + (n – 1) * PAGEBURSTACCESSTIME) * (TIMEPARAGRANULARITY + 1) * GPMC_FCLK period
   C = (RDCYCLETIME + (n – 1) * PAGEBURSTACCESSTIME) * (TIMEPARAGRANULARITY + 1) * GPMC_FCLK period
   D = (RDCYCLETIME – (RDACCESSTIME + (n – 1) * PAGEBURSTACCESSTIME)) * GPMC_FCLK period
   E = (CSRDOFFTIME – (RDACCESSTIME + (n – 1) * PAGEBURSTACCESSTIME)) * GPMC_FCLK period

3. For burst write accesses (where n is the page burst access number):

   A = (CSWROFFTIME – CSONTIME + (n – 1) * PAGEBURSTACCESSTIME) * (TIMEPARAGRANULARITY + 1) * GPMC_FCLK period
   C = (WRCYCLETIME + (n – 1) * PAGEBURSTACCESSTIME) * (TIMEPARAGRANULARITY + 1) * GPMC_FCLK period
   D = (WRCYCLETIME – (RDACCESSTIME + (n – 1) * PAGEBURSTACCESSTIME)) * GPMC_FCLK period
   E = (CSWROFFTIME – (RDACCESSTIME + (n – 1) * PAGEBURSTACCESSTIME)) * GPMC_FCLK period

4. For all accesses:

   For nCS falling edge (chip-select activated):

   - Case where GPMC_CONFIG1_i[1:0] GPMCFCLKDIVIDER = 0x0:
     F = 0.5 * CSEXTRADELAY * GPMC_FCLK period
   - Case where GPMCFCLKDIVIDER = 0x1:
     F = 0.5 * CSEXTRADELAY * GPMC_FCLK period, when (CLKACTIVATIONTIME and CSONTIME are odd) or (CLKACTIVATIONTIME and CSONTIME are even)
     F = (1 + 0.5 * CSEXTRADELAY) * GPMC_FCLK period otherwise.
   - Case where GPMCFCLKDIVIDER = 0x2:
     F = 0.5 * CSEXTRADELAY * GPMC_FCLK period, when (CSONTIME – CLKACTIVATIONTIME) is a multiple of 3
     F = (1 + 0.5 * CSEXTRADELAY) * GPMC_FCLK period, when (CSONTIME – CLKACTIVATIONTIME – 1) is a multiple of 3
     F = (2 + 0.5 * CSEXTRADELAY) * GPMC_FCLK period, when (CSONTIME – CLKACTIVATIONTIME – 2) is a multiple of 3
   - Case where GPMCFCLKDIVIDER = 0x3:
     F = 0.5 * CSEXTRADELAY * GPMC_FCLK period, when (CSONTIME – CLKACTIVATIONTIME) is a multiple of 4
     F = (1 + 0.5 * CSEXTRADELAY) * GPMC_FCLK period, when (CSONTIME – CLKACTIVATIONTIME – 1) is a multiple of 4
     F = (2 + 0.5 * CSEXTRADELAY) * GPMC_FCLK period, when (CSONTIME – CLKACTIVATIONTIME – 2) is a multiple of 4
     F = (3 + 0.5 * CSEXTRADELAY) * GPMC_FCLK period, when (CSONTIME – CLKACTIVATIONTIME – 3) is a multiple of 4

   For nCS rising edge (chip-select deactivated) in reading mode:

   - Case where GPMC_CONFIG1_i[1:0] GPMCFCLKDIVIDER = 0x0:
     F = 0.5 * CSEXTRADELAY * GPMC_FCLK period
   - Case where GPMCFCLKDIVIDER = 0x1:
     F = 0.5 * CSEXTRADELAY * GPMC_FCLK period, when (CLKACTIVATIONTIME and CSRDOFFTIME are odd) or (CLKACTIVATIONTIME and CSRDOFFTIME are even)
     F = (1 + 0.5 * CSEXTRADELAY) * GPMC_FCLK period otherwise.

- Case where GPMCFCLKDIVIDER = 0x2:
  F = 0.5 * CSEXTRADELAY * GPMC_FCLK period, when (CSRDOFFTIME – CLKACTIVATIONTIME) is a multiple of 3
  F = (1 + 0.5 * CSEXTRADELAY) * GPMC_FCLK period, when (CSRDOFFTIME – CLKACTIVATIONTIME – 1) is a multiple of 3
  F = (2 + 0.5 * CSEXTRADELAY) * GPMC_FCLK period, when (CSRDOFFTIME – CLKACTIVATIONTIME – 2) is a multiple of 3
- Case where GPMCFCLKDIVIDER = 0x3:
  F = 0.5 * CSEXTRADELAY * GPMC_FCLK period, when (CSRDOFFTIME – CLKACTIVATIONTIME) is a multiple of 4
  F = (1 + 0.5 * CSEXTRADELAY) * GPMC_FCLK period, when (CSRDOFFTIME – CLKACTIVATIONTIME – 1) is a multiple of 4
  F = (2 + 0.5 * CSEXTRADELAY) * GPMC_FCLK period, when (CSRDOFFTIME – CLKACTIVATIONTIME – 2) is a multiple of 4
  F = (3 + 0.5 * CSEXTRADELAY) * GPMC_FCLK period, when (CSRDOFFTIME – CLKACTIVATIONTIME – 3) is a multiple of 4

For nCS rising edge (chip-select deactivated) in writing mode:

- Case where GPMC_CONFIG1_i[1:0] GPMCFCLKDIVIDER = 0x0:
  F = 0.5 * CSEXTRADELAY * GPMC_FCLK period
- Case where GPMCFCLKDIVIDER = 0x1:
  F = 0.5 * CSEXTRADELAY * GPMC_FCLK period, when (CLKACTIVATIONTIME and CSWROFFTIME are odd) or (CLKACTIVATIONTIME and CSWROFFTIME are even)
  F = (1 + 0.5 * CSEXTRADELAY) * GPMC_FCLK period otherwise.
- Case where GPMCFCLKDIVIDER = 0x2:
  F = 0.5 * CSEXTRADELAY * GPMC_FCLK period, when (CSWROFFTIME – CLKACTIVATIONTIME) is a multiple of 3
  F = (1 + 0.5 * CSEXTRADELAY) * GPMC_FCLK period, when (CSWROFFTIME – CLKACTIVATIONTIME – 1) is a multiple of 3
  F = (2 + 0.5 * CSEXTRADELAY) * GPMC_FCLK period, when (CSWROFFTIME – CLKACTIVATIONTIME – 2) is a multiple of 3
- Case where GPMCFCLKDIVIDER = 0x3:
  F = 0.5 * CSEXTRADELAY * GPMC_FCLK period, when (CSWROFFTIME – CLKACTIVATIONTIME) is a multiple of 4
  F = (1 + 0.5 * CSEXTRADELAY) * GPMC_FCLK period, when (CSWROFFTIME – CLKACTIVATIONTIME – 1) is a multiple of 4
  F = (2 + 0.5 * CSEXTRADELAY) * GPMC_FCLK period, when (CSWROFFTIME – CLKACTIVATIONTIME – 2) is a multiple of 4
  F = (3 + 0.5 * CSEXTRADELAY) * GPMC_FCLK period, when (CSWROFFTIME – CLKACTIVATIONTIME – 3) is a multiple of 4

For nADV falling edge (nADV activated):

- Case where GPMC_CONFIG1_i[1:0] GPMCFCLKDIVIDER = 0x0:
  G = 0.5 * ADVEXTRADELAY * GPMC_FCLK period
- Case where GPMCFCLKDIVIDER = 0x1:
  G = 0.5 * ADVEXTRADELAY * GPMC_FCLK period, when (CLKACTIVATIONTIME and ADVONTIME are odd) or (CLKACTIVATIONTIME and ADVONTIME are even)
  G = (1 + 0.5 * ADVEXTRADELAY) * GPMC_FCLK period otherwise.
- Case where GPMCFCLKDIVIDER = 0x2:
  G = 0.5 * ADVEXTRADELAY * GPMC_FCLK period, when (ADVONTIME – CLKACTIVATIONTIME) is a multiple of 3
  G = (1 + 0.5 * ADVEXTRADELAY) * GPMC_FCLK period, when (ADVONTIME – CLKACTIVATIONTIME – 1) is a multiple of 3
  G = (2 + 0.5 * ADVEXTRADELAY) * GPMC_FCLK period, when (ADVONTIME – CLKACTIVATIONTIME – 2) is a multiple of 3
- Case where GPMCFCLKDIVIDER = 0x3:
  G = 0.5 * ADVEXTRADELAY * GPMC_FCLK period, when (ADVONTIME – CLKACTIVATIONTIME) is a multiple of 4
  G = (1 + 0.5 * ADVEXTRADELAY) * GPMC_FCLK period, when (ADVONTIME –

CLKACTIVATIONTIME – 1) is a multiple of 4
G = (2 + 0.5 * ADVEXTRADELAY) * GPMC_FCLK period, when (ADVONTIME – CLKACTIVATIONTIME – 2) is a multiple of 4
G = (3 + 0.5 * ADVEXTRADELAY) * GPMC_FCLK period, when (ADVONTIME – CLKACTIVATIONTIME – 3) is a multiple of 4

For nADV rising edge (nADV deactivated) in reading mode:

- Case where GPMC_CONFIG1_i[1:0] GPMCFCLKDIVIDER = 0x0:
  G = 0.5 * ADVEXTRADELAY * GPMC_FCLK period
- Case where GPMCFCLKDIVIDER = 0x1:
  G = 0.5 * ADVEXTRADELAY * GPMC_FCLK period, when (CLKACTIVATIONTIME and ADVRDOFFTIME are odd) or (CLKACTIVATIONTIME and ADVRDOFFTIME are even)
  G = (1 + 0.5 * ADVEXTRADELAY) * GPMC_FCLK period otherwise.
- Case where GPMCFCLKDIVIDER = 0x2:
  G = 0.5 * ADVEXTRADELAY * GPMC_FCLK period, when (ADVRDOFFTIME - CLKACTIVATIONTIME) is a multiple of 3
  G = (1 + 0.5 * ADVEXTRADELAY) * GPMC_FCLK period, when (ADVRDOFFTIME - CLKACTIVATIONTIME - 1) is a multiple of 3
  G = (2 + 0.5 * ADVEXTRADELAY) * GPMC_FCLK period, when (ADVRDOFFTIME - CLKACTIVATIONTIME - 2) is a multiple of 3
- Case where GPMCFCLKDIVIDER = 0x3:
  G = 0.5 * ADVEXTRADELAY * GPMC_FCLK period, when (ADVRDOFFTIME – CLKACTIVATIONTIME) is a multiple of 4
  G = (1 + 0.5 * ADVEXTRADELAY) * GPMC_FCLK period, when (ADVRDOFFTIME – CLKACTIVATIONTIME – 1) is a multiple of 4
  G = (2 + 0.5 * ADVEXTRADELAY) * GPMC_FCLK period, when (ADVRDOFFTIME – CLKACTIVATIONTIME – 2) is a multiple of 4
  G = (3 + 0.5 * ADVEXTRADELAY) * GPMC_FCLK period, when (ADVRDOFFTIME – CLKACTIVATIONTIME – 3) is a multiple of 4

For nADV rising edge (nADV deactivated) in writing mode:

- Case where GPMC_CONFIG1_i[1:0] GPMCFCLKDIVIDER = 0x0:
  G = 0.5 * ADVEXTRADELAY * GPMC_FCLK period
- Case where GPMCFCLKDIVIDER = 0x1:
  G = 0.5 * ADVEXTRADELAY * GPMC_FCLK period, when (CLKACTIVATIONTIME and ADVWROFFTIME are odd) or (CLKACTIVATIONTIME and ADVWROFFTIME are even)
  G = (1 + 0.5 * ADVEXTRADELAY) * GPMC_FCLK period otherwise.
- Case where GPMCFCLKDIVIDER = 0x2:
  G = 0.5 * ADVEXTRADELAY * GPMC_FCLK period, when (ADVWROFFTIME – CLKACTIVATIONTIME) is a multiple of 3
  G = (1 + 0.5 * ADVEXTRADELAY) * GPMC_FCLK period, when (ADVWROFFTIME – CLKACTIVATIONTIME – 1) is a multiple of 3
  G = (2 + 0.5 * ADVEXTRADELAY) * GPMC_FCLK period, when (ADVWROFFTIME – CLKACTIVATIONTIME – 2) is a multiple of 3
- Case where GPMCFCLKDIVIDER = 0x3:
  G = 0.5 * ADVEXTRADELAY * GPMC_FCLK period, when (ADVWROFFTIME – CLKACTIVATIONTIME) is a multiple of 4
  G = (1 + 0.5 * ADVEXTRADELAY) * GPMC_FCLK period, when (ADVWROFFTIME – CLKACTIVATIONTIME – 1) is a multiple of 4
  G = (2 + 0.5 * ADVEXTRADELAY) * GPMC_FCLK period, when (ADVWROFFTIME – CLKACTIVATIONTIME – 2) is a multiple of 4
  G = (3 + 0.5 * ADVEXTRADELAY) * GPMC_FCLK period, when (ADVWROFFTIME – CLKACTIVATIONTIME – 3) is a multiple of 4

For nOE falling edge (nOE activated):

- Case where GPMC_CONFIG1_i[1:0] GPMCFCLKDIVIDER = 0x0:
  H = 0.5 * OEEXTRADELAY * GPMC_FCLK period
- Case where GPMCFCLKDIVIDER = 0x1:
  H = 0.5 * OEEXTRADELAY * GPMC_FCLK period, when (CLKACTIVATIONTIME and OEONTIME are odd) or (CLKACTIVATIONTIME and OEONTIME are even)

H = (1 + 0.5 * OEEXTRADELAY) * GPMC_FCLK period otherwise.

- Case where GPMCFCLKDIVIDER = 0x2:
  H = 0.5 * OEEXTRADELAY * GPMC_FCLK period, when (OEONTIME – CLKACTIVATIONTIME) is a multiple of 3
  H = (1 + 0.5 * OEEXTRADELAY) * GPMC_FCLK period, when (OEONTIME – CLKACTIVATIONTIME – 1) is a multiple of 3
  H = (2 + 0.5 * OEEXTRADELAY) * GPMC_FCLK period, when (OEONTIME – CLKACTIVATIONTIME – 2) is a multiple of 3
- Case where GPMCFCLKDIVIDER = 0x3:
  H = 0.5 * OEEXTRADELAY * GPMC_FCLK period, when (OEONTIME – CLKACTIVATIONTIME) is a multiple of 4
  H = (1 + 0.5 * OEEXTRADELAY) * GPMC_FCLK period, when (OEONTIME – CLKACTIVATIONTIME – 1) is a multiple of 4
  H = (2 + 0.5 * OEEXTRADELAY) * GPMC_FCLK period, when (OEONTIME – CLKACTIVATIONTIME – 2) is a multiple of 4
  H = (3 + 0.5 * OEEXTRADELAY) * GPMC_FCLK period, when (OEONTIME – CLKACTIVATIONTIME – 3) is a multiple of 4

For nOE rising edge (nOE deactivated):

- Case where GPMC_CONFIG1_i[1:0] GPMCFCLKDIVIDER = 0x0:
  H = 0.5 * OEEXTRADELAY * GPMC_FCLK period
- Case where GPMCFCLKDIVIDER = 0x1:
  H = 0.5 * OEEXTRADELAY * GPMC_FCLK period, when (CLKACTIVATIONTIME and OEOFFTIME are odd) or (CLKACTIVATIONTIME and OEOFFTIME are even)
  H = (1 + 0.5 * OEEXTRADELAY) * GPMC_FCLK period otherwise.
- Case where GPMCFCLKDIVIDER = 0x2:
  H = 0.5 * OEEXTRADELAY * GPMC_FCLK period, when (OEOFFTIME – CLKACTIVATIONTIME) is a multiple of 3
  H = (1 + 0.5 * OEEXTRADELAY) * GPMC_FCLK period, when (OEOFFTIME – CLKACTIVATIONTIME – 1) is a multiple of 3
  H = (2 + 0.5 * OEEXTRADELAY) * GPMC_FCLK period, when (OEOFFTIME – CLKACTIVATIONTIME – 2) is a multiple of 3
- Case where GPMCFCLKDIVIDER = 0x3:
  H = 0.5 * OEEXTRADELAY * GPMC_FCLK period, when (OEOFFTIME – CLKACTIVATIONTIME) is a multiple of 4
  H = (1 + 0.5 * OEEXTRADELAY) * GPMC_FCLK period, when (OEOFFTIME – CLKACTIVATIONTIME – 1) is a multiple of 4
  H = (2 + 0.5 * OEEXTRADELAY) * GPMC_FCLK period, when (OEOFFTIME – CLKACTIVATIONTIME – 2) is a multiple of 4
  H = (3 + 0.5 * OEEXTRADELAY) * GPMC_FCLK period, when (OEOFFTIME – CLKACTIVATIONTIME – 3) is a multiple of 4

For nWE falling edge (nWE activated):

- Case where GPMC_CONFIG1_i[1:0] GPMCFCLKDIVIDER = 0x0:
  I = 0.5 * WEEXTRADELAY * GPMC_FCLK period
- Case where GPMCFCLKDIVIDER = 0x1:
  I = 0.5 * WEEXTRADELAY * GPMC_FCLK period, when (CLKACTIVATIONTIME and WEONTIME are odd) or (CLKACTIVATIONTIME and WEONTIME are even)
  I = (1 + 0.5 * WEEXTRADELAY) * GPMC_FCLK period otherwise.
- Case where GPMCFCLKDIVIDER = 0x2:
  I = 0.5 * WEEXTRADELAY * GPMC_FCLK period, when (WEONTIME – CLKACTIVATIONTIME) is a multiple of 3
  I = (1 + 0.5 * WEEXTRADELAY) * GPMC_FCLK period, when (WEONTIME – CLKACTIVATIONTIME – 1) is a multiple of 3
  I = (2 + 0.5 * WEEXTRADELAY) * GPMC_FCLK period, when (WEONTIME – CLKACTIVATIONTIME – 2) is a multiple of 3
- Case where GPMCFCLKDIVIDER = 0x3:
  I = 0.5 * WEEXTRADELAY * GPMC_FCLK period, when (WEONTIME – CLKACTIVATIONTIME) is a multiple of 4

I = (1 + 0.5 * WEEXTRADELAY) * GPMC_FCLK period, when (WEONTIME – CLKACTIVATIONTIME – 1) is a multiple of 4
I = (2 + 0.5 * WEEXTRADELAY) * GPMC_FCLK period, when (WEONTIME – CLKACTIVATIONTIME – 2) is a multiple of 4
I = (3 + 0.5 * WEEXTRADELAY) * GPMC_FCLK period, when (WEONTIME – CLKACTIVATIONTIME – 3) is a multiple of 4

For nWE rising edge (nWE deactivated):

- Case where GPMC_CONFIG1_i[1:0] GPMCFCLKDIVIDER = 0x0:
  I = 0.5 * WEEXTRADELAY * GPMC_FCLK period
- Case where GPMCFCLKDIVIDER = 0x1:
  I = 0.5 * WEEXTRADELAY * GPMC_FCLK period, when (CLKACTIVATIONTIME and WEOFFTIME are odd) or (CLKACTIVATIONTIME and WEOFFTIME are even)
  I = (1 + 0.5 * WEEXTRADELAY) * GPMC_FCLK period otherwise.
- Case where GPMCFCLKDIVIDER = 0x2:
  I = 0.5 * WEEXTRADELAY * GPMC_FCLK period, when (WEOFFTIME – CLKACTIVATIONTIME) is a multiple of 3
  I = (1 + 0.5 * WEEXTRADELAY) * GPMC_FCLK period, when (WEOFFTIME – CLKACTIVATIONTIME – 1) is a multiple of 3
  I = (2 + 0.5 * WEEXTRADELAY) * GPMC_FCLK period, when (WEOFFTIME – CLKACTIVATIONTIME – 2) is a multiple of 3
- Case where GPMCFCLKDIVIDER = 0x3:
  I = 0.5 * WEEXTRADELAY * GPMC_FCLK period, when (WEOFFTIME – CLKACTIVATIONTIME) is a multiple of 4
  I = (1 + 0.5 * WEEXTRADELAY) * GPMC_FCLK period, when (WEOFFTIME – CLKACTIVATIONTIME – 1) is a multiple of 4
  I = (2 + 0.5 * WEEXTRADELAY) * GPMC_FCLK period, when (WEOFFTIME – CLKACTIVATIONTIME – 2) is a multiple of 4
  I = (3 + 0.5 * WEEXTRADELAY) * GPMC_FCLK period, when (WEOFFTIME – CLKACTIVATIONTIME – 3) is a multiple of 4

For gpmc_nadv low pulse duration:

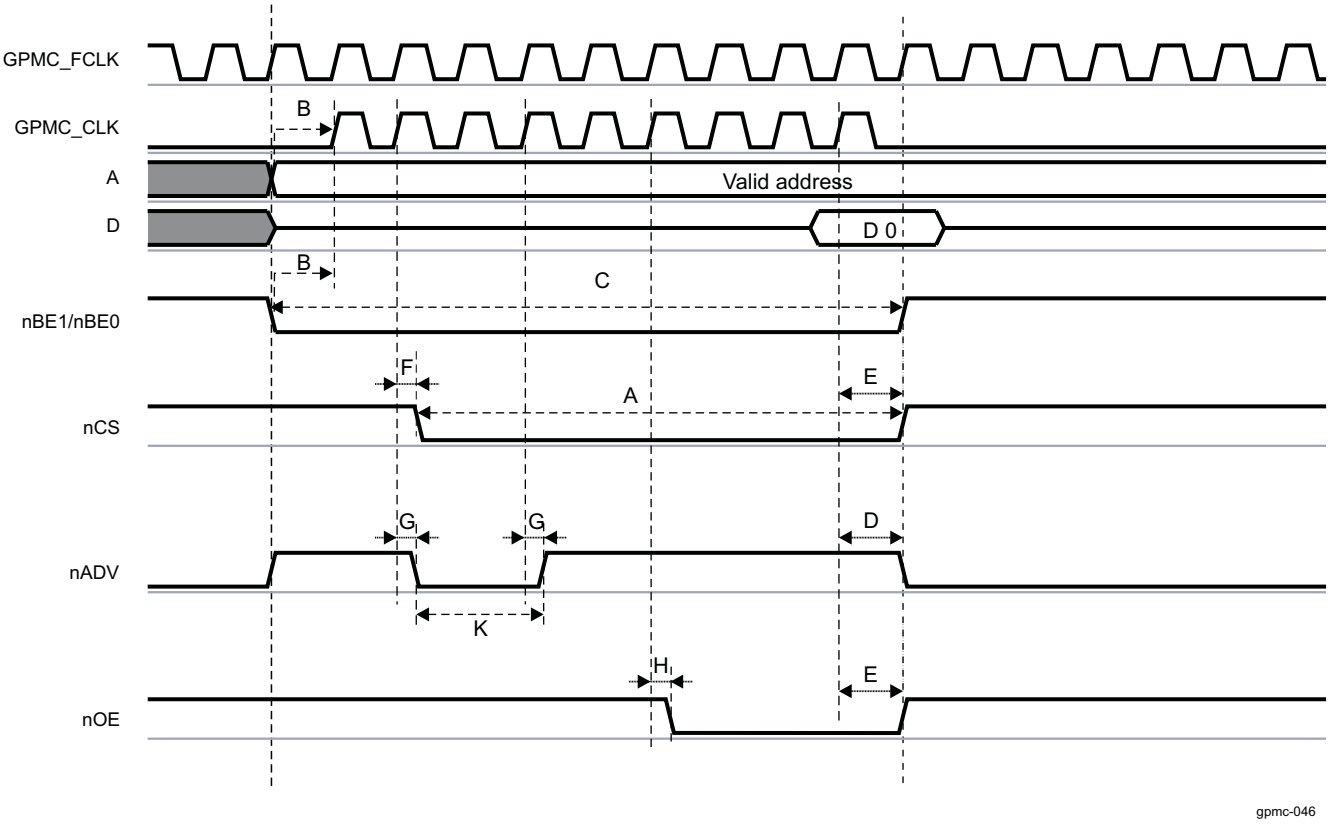
- Read operation:
  K = (ADVRDOFFTIME – ADVONTIME) * (TIMEPARAGRANULARITY + 1) * GPMC_FCLK period
- Write operation:
  K = (ADVWROFFTIME – ADVONTIME) * (TIMEPARAGRANULARITY + 1) * GPMC_FCLK period

For gpmc_wait invalid to first data latching gpmc_clk edge:

- L = WAITMONITORINGTIME * (GPMCFCLKDIVIDER + 1) * GPMC_FCLK period + GPMC_CLK period

Figure 10-57 shows a simplified example of a synchronous NOR single read where formulas are associated with signal waves.

Figure 10-57. Synchronous NOR Single Read Simplified Example



10.3.5.6.1.3 Asynchronous NOR Flash Timing Parameters Formulas

This section lists all the formulas to calculate asynchronous NOR timing parameters. This is the case when GPMC_CONFIG1_i[11:10] DEVICETYPE = 0x0 and when READTYPE or WRITETYPE are set to asynchronous mode. Table 10-403 describes the asynchronous NOR formulas.

Table 10-403. Asynchronous NOR Formulas Description

Configuration Parameter	Unit	Description
A	ns	Pulse duration – gpmc_cs low
B	ns	Delay time – gpmc_cs valid to gpmc_advn_ale invalid
C	ns	Delay time – gpmc_cs valid to gpmc_oen_ren invalid (single read)
D	ns	Pulse duration – address bus valid - 2nd, 3rd and 4th accesses
E	ns	Delay time – gpmc_cs valid to gpmc_wen valid
F	ns	Delay time – gpmc_cs valid to gpmc_wen invalid
G	ns	Address invalid duration between two successive R/W accesses
H	ns	Setup time – read data valid before gpmc_oen_ren high
I	ns	Delay time – gpmc_cs valid to gpmc_oen_ren invalid (burst read)
J	ns	Delay time – address bus valid to gpmc_cs valid
		Delay time – data bus valid to gpmc_cs valid
		Delay time – gpmc_ben0/gpmc_ben1 valid to gpmc_cs valid
K	ns	Delay time – gpmc_cs valid to gpmc_advn_ale valid
L	ns	Delay time – gpmc_cs valid to gpmc_oen_ren valid
M	ns	Delay time – gpmc_cs valid to first data latching edge
N	ns	Pulse duration – gpmc_ben0/gpmc_ben1 valid time

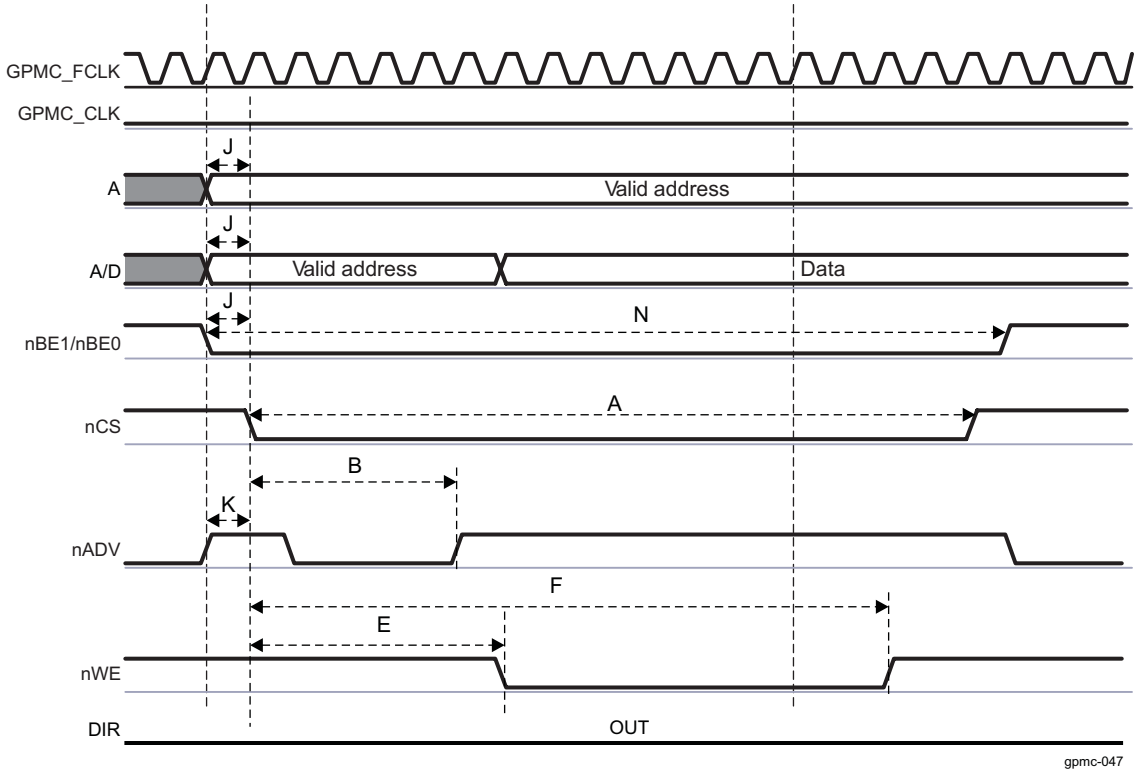
Table 10-403. Asynchronous NOR Formulas Description (continued)

Configuration Parameter	Unit	Description
O	ns	Delay time – gpmc_cs valid to gpmc_advn_ale valid

The configuration parameters are calculated through the following formulas. These formulas are not exhaustive. For more information, see the device data manual.

- gpmc_cs low pulse:
  For single read: A = (CSRDOFFTIME – CSONTIME) * (TIMEPARAGRANULARITY + 1) * GPMC_FCLK period
  For burst read: A = (CSRDOFFTIME – CSONTIME + (N – 1) * PAGEBURSTACCESSTIME) * (TIMEPARAGRANULARITY + 1) * GPMC_FCLK period, where N = page burst access number
  For single write: A = (CSWROFFTIME – CSONTIME) * (TIMEPARAGRANULARITY + 1) * GPMC_FCLK period
  For burst write: A = (CSWROFFTIME – CSONTIME + (N – 1) * PAGEBURSTACCESSTIME) * (TIMEPARAGRANULARITY + 1) * GPMC_FCLK period, where N = page burst access number
- gpmc_cs valid to gpmc_advn_ale invalid delay:
  For reading: B = ((ADVRDOFFTIME – CSONTIME) * (TIMEPARAGRANULARITY + 1) + 0.5 * (ADVEXTRADELAY – CSEXTRADELAY)) * GPMC_FCLK period
  For writing: B = ((ADVWROFFTIME – CSONTIME) * (TIMEPARAGRANULARITY + 1) + 0.5 * (ADVEXTRADELAY – CSEXTRADELAY)) * GPMC_FCLK period
- C = ((OEOFFTIME – CSONTIME) * (TIMEPARAGRANULARITY + 1) + 0.5 * (OEEXTRADELAY – CSEXTRADELAY)) * GPMC_FCLK period
- D = PAGEBURSTACCESSTIME * (TIMEPARAGRANULARITY + 1) * GPMC_FCLK period
- E = ((WEONTIME – CSONTIME) * (TIMEPARAGRANULARITY + 1) + 0.5 * (WEEXTRADELAY – CSEXTRADELAY)) * GPMC_FCLK period
- F = ((WEOFFTIME – CSONTIME) * (TIMEPARAGRANULARITY + 1) + 0.5 * (WEEXTRADELAY – CSEXTRADELAY)) * GPMC_FCLK period
- G = CYCLE2CYCLEDELAY * GPMC_FCLK period
- H = ((OEOFFTIME – RDACCESSTIME) * (TIMEPARAGRANULARITY + 1) + 0.5 * OEEXTRADELAY) * GPMC_FCLK period
- I = ((OEOFFTIME + (N – 1) * PAGEBURSTACCESSTIME – CSONTIME) * (TIMEPARAGRANULARITY + 1) + 0.5 * (OEEXTRADELAY – CSEXTRADELAY)) * GPMC_FCLK period, where N = page burst access number
- J = (CSONTIME * (TIMEPARAGRANULARITY + 1) + 0.5 * CSEXTRADELAY) * GPMC_FCLK period
- K = ((ADVONTIME – CSONTIME) * (TIMEPARAGRANULARITY + 1) + 0.5 * (ADVEXTRADELAY – CSEXTRADELAY)) * GPMC_FCLK period
- L = ((OEONTIME – CSONTIME) * (TIMEPARAGRANULARITY + 1) + 0.5 * (OEEXTRADELAY – CSEXTRADELAY)) * GPMC_FCLK period
- M = ((RDACCESSTIME - CSONTIME) * (TIMEPARAGRANULARITY + 1) – 0.5 * CSEXTRADELAY) * GPMC_FCLK period
- gpmc_ben0/gpmc_ben1 pulse:
  For single read: N = RDCYCLETIME * (TIMEPARAGRANULARITY + 1) * GPMC_FCLK period
  For burst read: N = (RDCYCLETIME + (N – 1) * PAGEBURSTACCESSTIME) * (TIMEPARAGRANULARITY + 1) * GPMC_FCLK period, where N = page burst access number
  For burst write: N = (WRCYCLETIME + (N – 1) * PAGEBURSTACCESSTIME) * (TIMEPARAGRANULARITY + 1) * GPMC_FCLK period, where N = page burst access number
- O = ((WRCYCLETIME + (N – 1) * PAGEBURSTACCESSTIME – CSONTIME) * (TIMEPARAGRANULARITY + 1) + 0.5 * (ADVEXTRADELAY – CSEXTRADELAY)) * GPMC_FCLK period
  Figure 10-58 shows a simplified example of an asynchronous NOR single write where formulas are associated with signal waves.

Figure 10-58. Asynchronous NOR Single Write Simplified Example



NOTE: Write multiple access is not supported in asynchronous mode. If WRITEMULTIPLE is enabled with WRITETYPE as asynchronous, the GPMC processes single asynchronous accesses.

10.3.6 GPMC Use Cases and Tips

10.3.6.1 How to Set GPMC Timing Parameters for Typical Accesses

10.3.6.1.1 External Memory Attached to the GPMC Module

As discussed in the introduction to this chapter, the GPMC module supports the following external memory types:

- Asynchronous or synchronous, 8- or 16-bit-wide memory or device
- 16-bit address/data-multiplexed or not multiplexed NOR flash device
- 8- or 16-bit NAND flash device

The following examples describe how to calculate GPMC timing parameters by showing a typical parameter setup for the access to be performed.

The example is based on a 512-Mb multiplexed NOR flash memory with the following characteristics:

- Type: NOR flash (address/data-multiplexed mode)
- Size: 512M bits
- Data Bus: 16 bits wide
- Speed: 104-MHz clock frequency
- Read access time: 80 ns

10.3.6.1.2 Typical GPMC Setup

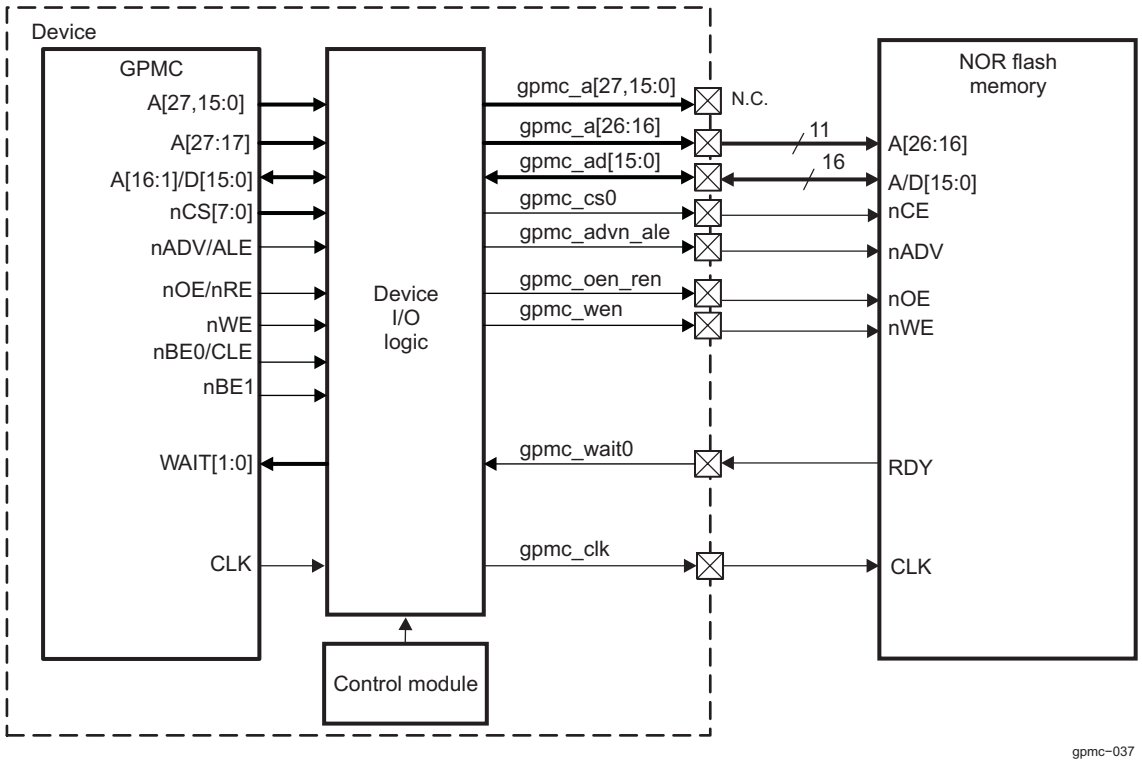
Table 10-404 lists some of the I/Os of the GPMC module.

Table 10-404. GPMC Signals

Signal Name	I/0	Description
GPMC_FCLK	Internal	Functional and interface clock. Acts as the time reference.
gpmc_clk	O	External clock provided to the external device for synchronous operations
gpmc_a[26:16]	O	Address
gpmc_ad[15:0]	I/O	Data-multiplexed with addresses A[16:1] on memory side
gpmc_csx	O	Chip-select (where x = 0, or 1)
gpmc_advn_ale	O	Address valid enable
gpmc_oen_ren	O	Output enable (read access only)
gpmc_wen	O	Write enable (write access only)
gpmc_wait[1:0]	I	Ready signal from memory device. Indicates when valid burst data is ready to be read

Figure 10-59 shows the typical connection between the GPMC module and an attached NOR Flash memory.

Figure 10-59. GPMC Connection to an External NOR Flash Memory



The following sections demonstrate how to calculate GPMC parameters for three access types:

- Synchronous burst read
- Asynchronous read
- Asynchronous single write

10.3.6.1.2.1 GPMC Configuration for Synchronous Burst Read Access

NOTE: The examples in Section 10.3.6.1.2.1 through Section 10.3.6.1.2.3 are based on a clock rate of 104 MHz. Refer to the device Data Manual for the maximum frequency appropriate for this device and to the memory datasheet for the maximum frequency for the particular memory device.

The clock runs at 104 MHz (f = 104 MHz; T = 9, 615 ns).

Table 10-405 shows the timing parameters (on the memory side) that determine the parameters on the GPMC side.

Table 10-406 shows how to calculate timings for the GPMC using the memory parameters.

Figure 10-60 shows the synchronous burst read access.

Table 10-405. Useful Timing Parameters on the Memory Side

AC Read Characteristics on the Memory Side	Description	Duration (ns)
tCES	nCS setup time to clock	0
tACS	Address setup time to clock	3
tIACC	Synchronous access time	80

Table 10-405. Useful Timing Parameters on the Memory Side (continued)

AC Read Characteristics on the Memory Side	Description	Duration (ns)
tBACC	Burst access time valid clock to output delay	5,2
tCEZ	Chip-select to High-Z	7
tOEZ	Output enable to High-Z	7
tAVC	nADV setup time	6
tAVD	nAVD pulse	6
tACH	Address hold time from clock	3

The following terms, which describe the timing interface between the controller and its attached device, are used to calculate the timing parameters on the GPMC side:

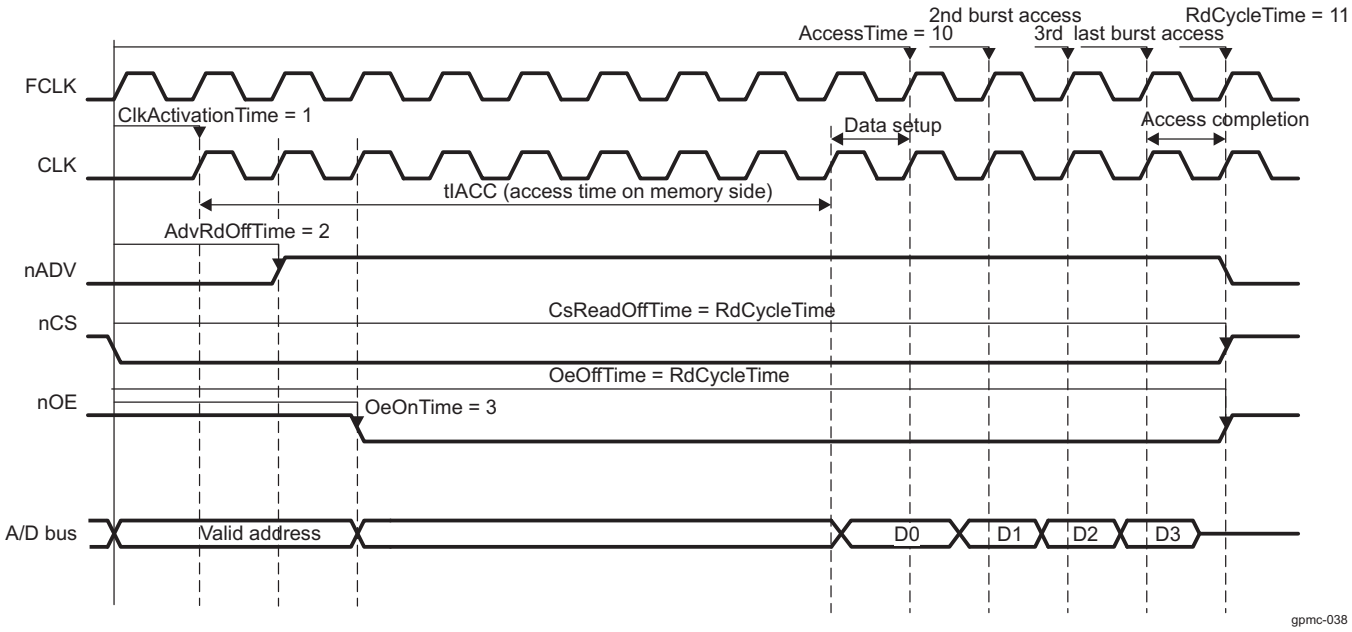
- Read access time (GPMC side): Time required to activate the clock + read access time requested on the memory side + data setup time required for optimal capture of a burst of data
- Data setup time (GPMC side): Ensures a good capture of a burst of data (as opposed to taking a burst of data out). One word of data is processed in one clock cycle (T = 9,615 ns). The read access time between two bursts of data is tBACC = 5.2 ns. Therefore, data setup time is a clock period – tBACC = 4,415 ns of data setup.
- Access completion (GPMC side): (Different from page burst access time) Time required between the last burst access and access completion: nCS/nOE hold time (nCS and nOE must be released at the end of an access. These signals are held to allow the access to complete).
- Read cycle time (GPMC side): Read access time + access completion
- Write cycle time for burst access: Not supported for NOR flash memory

Table 10-406. Calculating GPMC Timing Parameters

Parameter Name on GPMC Side	Formula	Duration (ns)	Number of Clock Cycles (F = 104 MHz)	GPMC Register Configurations
GPMC FCLK Divider	–	–	–	GPMCFCLKDIVIDER = 0x0
ClkActivation Time	min (tCES, tACS)	3	1	CLKACTIVATIONTIME = 0x1
RdAccessTime	roundmax (ClkActivationTime + tIACC + DataSetupTime)	94.03: (9,615 + 80 + 4,415)	10: roundmax (94.03 / 9,615)	RDACCESSTIME = 0x0A
PageBurst RdAccessTime	roundmax (tBACC)	roundmax (5.2)	1	PAGEBURSTACCESSTIME = 0x1
RdCycleTime	RdAccess time + max (tCEZ, tOEZ)	101.03: (94.03 + 7)	11	RDCYCLETIME = 0x0B
CsOnTime	tCES	0	0	CSONTIME = 0x0
CsReadOffTime	RdCycleTime	-	11	CSRDOFFTIME = 0x0B
AdvOnTime	tAVC (1)	0	0	ADVONTIME = 0x0
AdvRdOffTime	tAVD + tAVC (2)	12	2	ADVRDOFFTIME = 0x02
OeOnTime (3)	(ClkActivationTime + tACH) < OeOnTime (ClkActivationTime + tIACC)	–	3, for instance	OEONTIME = 0x3
OeOffTime	RdCycleTime	–	11	OEOFFTIME = 0x0B

(1) The external clock provided to the NOR flash is not yet available.
(2) AdvRdOffTime – AdvOnTime = tAVD; thus, AdvRdOffTime = tAVD + AdvOnTime = tAVD + tAVC.
(3) OeOnTime must ensure that addresses are available. It must not exceed the availability of the first burst of data.

Figure 10-60. Synchronous Burst Read Access (Timing Parameters in Clock Cycles)



10.3.6.1.2.2 GPMC Configuration for Asynchronous Read Access

The clock runs at 104 MHz (f = 104 MHz; T = 9, 615 ns).

Table 10-407 shows the timing parameters (on the memory side) that determine the parameters on the GPMC side.

Table 10-408 shows how to calculate timings for the GPMC using the memory parameters.

Figure 10-61 shows the asynchronous read access.

Table 10-407. AC Characteristics for Asynchronous Read Access

AC Read Characteristics on the Memory Side	Description	Duration (ns)
tCE	Read Access time from nCS low	80
tAAVDS	Address setup time to rising edge of nADV	3
tAVDP	nADV low time	6
tCAS	nCS setup time to nADV	0
tOE	Output enable to output valid	6
tOEZ	Output enable to High-Z	7

Use the following formula to calculate the RdCycleTime parameter for this typical access:

RdCycleTime = RdAccessTime + AccessCompletion = RdAccessTime + 1 clock cycle + tOEZ:

1. On the memory side, the external memory makes the data available to the output bus. This is the memory-side read access time defined in Table 10-407: the number of clock cycles between the address capture (nADV rising edge) and the data valid on the output bus.

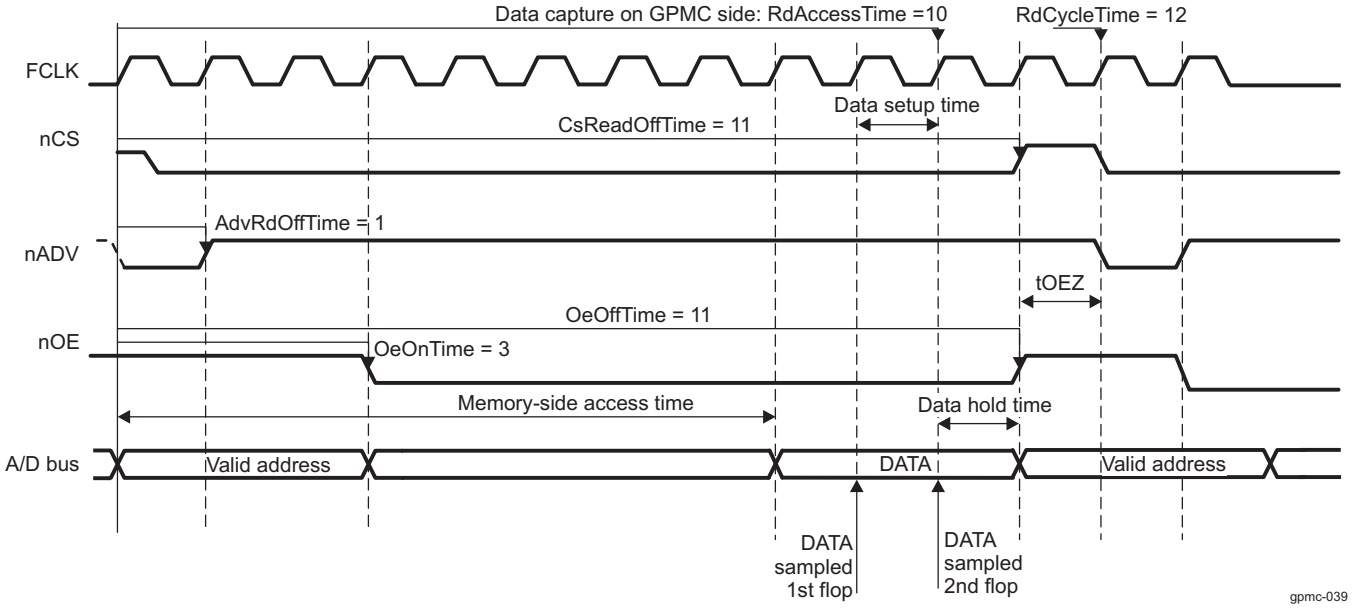
   The GPMC requires some hold time to allow the data to be captured correctly and the access to be finished.
2. To read the data correctly, the GPMC must be configured to meet the data setup time requirement of the memory; the GPMC module captures the data on the next rising edge. This is access time on the GPMC side.
3. There must also be a data hold time for correctly reading the data (checking that there is no nOE/nCS deassertion while reading the data). This data hold time is one clock cycle (that is, RdAccessTime + 1).

4. To complete the access, nOE/nCS signals are driven to High-Z. RdAccessTime + 1 + tOEZ is the read cycle time.
5. Addresses can now be relatched and a new read cycle begun.

Table 10-408. GPMC Timing Parameters for Asynchronous Read Access

Parameter Name on GPMC Side	Formula	Duration (ns)	Number of Clock Cycles (F = 104 MHz)	GPMC Register Configurations
ClkActivationTime		n/a (asynchronous mode)		
RdAccessTime	round max (tCE)	80	10	RDACCESSTIME = 0x0A
PageBurstAccess Time	N/A (single access)			
RdCycleTime	RdAccessTime + 1cycle + tOEZ	96, 615	12	RDCYCLETIME = 0x0C
CsOnTime	tCAS	0	0	CSONTIME = 0x0
CsReadOffTime	RdAccessTime + 1 cycle	89, 615	11	CSRDOFFTIME = 0x0B
AdvOnTime	tAAVDS	3	1	ADVONTIME = 0X1
AdvRdOffTime	tAAVDS + tAVDP	9	1	ADVRDOFFTIME = 0x01
OeOnTime	OeOnTime >= AdvRdOffTime (multiplexed mode)	-	3, for instance	OEONTIME = 0x3
OeOffTime	RdAccessTime + 1cycle	89, 615	11	OEOFFTIME = 0x0B

Figure 10-61. Asynchronous Single Read Access (Timing Parameters in Clock Cycles)



10.3.6.1.2.3 GPMC Configuration for Asynchronous Single Write Access

The clock runs at 104 MHz: (f = 104 MHz; T = 9, 615 ns).

Table 10-410 shows how to calculate timings for the GPMC using the memory parameters.

Table 10-409 shows the timing parameters (on the memory side) that determine the parameters on the GPMC side.

Figure 10-62 shows the synchronous burst write access.

Table 10-409. AC Characteristics for Asynchronous Single Write (Memory Side)

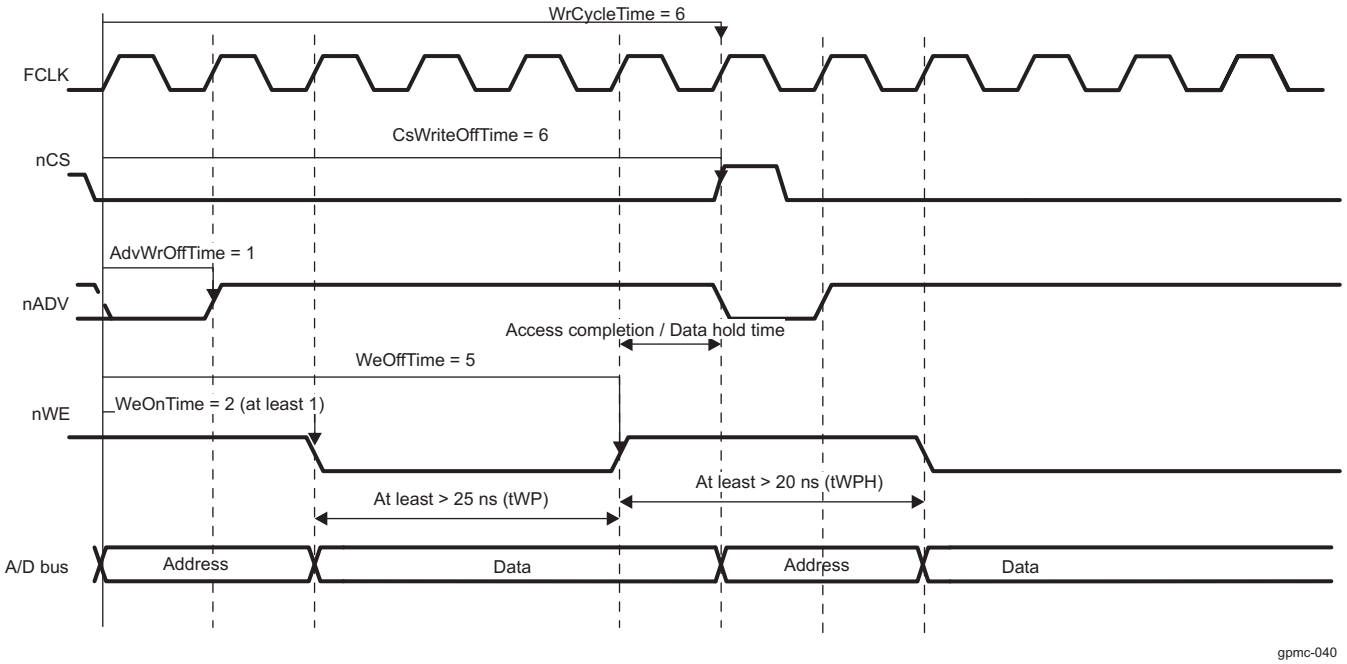
AC Characteristics on the Memory Side	Description	Duration (ns)
tWC	Write cycle time	60
tAVDP	nADV low time	6
tWP	Write pulse width	25
tWPH	Write pulse width high	20
tCS	nCS setup time to nWE	3
tCAS	nCS setup time to nADV	0
tAVSC	nADV setup time	3

For asynchronous single write access, write cycle time is WrCycleTime = WeOffTime + AccessCompletion = WeOffTime + 1. For the AccesCompletion, the GPMC requires one cycle of data hold time (nCS deassertion). For more information, see the device data manual.

Table 10-410. GPMC Timing Parameters for Asynchronous Single Write

Parameter Name on GPMC Side	Formula	Duration (ns)	Number of Clock Cycles (F = 104 MHz)	GPMC Registers Configuration
ClkActivationTime	N/A (asynchronous mode)			
WdAccessTime	Applicable only to WAITMONITORING (the value is the same as for read access)			
PageBurstAccessTime	N/A (single access)			
WrCycleTime	WeOffTime + AccessCompletion	57, 615	6	WRCYCLETIME = 0x06
CsOnTime	tCAS	0	0	CSONTIME = 0x0
CsWrOffTime	WeOffTime + 1	57, 615	6	CSWROFFTIME = 0x06
AdvOnTime	tAVSC	3	1	ADVONTIME = 0x1
AdvWrOffTime	tAVSC + tAVDP	9	1	ADVWROFFTIME = 0x01
WeOnTime	tCS	3	1	WEONTIME = 0x1
WeOffTime	tCS + tWP + tWPH	48	5	WEOFFTIME = 0x05

Figure 10-62. Asynchronous Single Write Access (Timing Parameters in Clock Cycles)



10.3.6.2 How to Choose a Suitable Memory to Use With the GPMC

This section is intended to help the user select a suitable memory device to interface with the GPMC controller.

10.3.6.2.1 Supported Memories or Devices

NAND flash and NOR flash architectures are the two flash technologies. The GPMC supports various types of external memory or devices, basically any one that supports NAND or NOR protocols:

- 8- and 16-bit-wide asynchronous or synchronous memory or device (only 8-bit: nonburst device)
- 16-bit address and data-multiplexed NOR flash devices (pSRAM, OneNAND™, etc.)
- 8- and 16-bit NAND flash devices

10.3.6.2.1.1 Memory Pin Multiplexing

This section describes the interfacing differences of the GPMC supported memories.

Table 10-411. Supported Memories Interfaces

Function	16-Bit Address/ Data-Multiplexed pSRAM or NOR Flash [1]	OneNAND	16-Bit NAND	8-Bit NAND
gpmc_a[27]	A27			
gpmc_a[26]				
gpmc_a[25]				
gpmc_a[24]				
gpmc_a[23]				
gpmc_a[22]				
gpmc_a[21]				
gpmc_a[20]				
gpmc_a[19]				
gpmc_a[18]				
gpmc_a[17]				
gpmc_a[16]				
gpmc_a[15]				
gpmc_a[14]				
gpmc_a[13]				
gpmc_a[12]				
gpmc_a[11]				
gpmc_a[10]	A26			
gpmc_a[9]	A25			
gpmc_a[8]	A24			
gpmc_a[7]	A23			
gpmc_a[6]	A22			
gpmc_a[5]	A21			
gpmc_a[4]	A20			
gpmc_a[3]	A19			
gpmc_a[2]	A18			
gpmc_a[1]	A17			
gpmc_a[0]	A16			
gpmc_ad[15]	D15 or A16		IO15	
gpmc_ad[14]	D14 or A15		IO14	

(1) Addresses seen from the device side. When interfacing to the external device, A1 is connected to the memory A0, A2 to the memory A1, and so on.

Table 10-411. Supported Memories Interfaces (continued)

Function	16-Bit Address/ Data-Multiplexed pSRAM or NOR Flash [1]	OneNAND	16-Bit NAND	8-Bit NAND
gpmc_ad[13]	D13 or A14		IO13	
gpmc_ad[12]	D12 or A13		IO12	
gpmc_ad[11]	D11 or A12		IO11	
gpmc_ad[10]	D10 or A11		IO10	
gpmc_ad[9]	D9 or A10		IO9	
gpmc_ad[8]	D8 or A9		IO8	
gpmc_ad[7]	D7 or A8		IO7	
gpmc_ad[6]	D6 or A7		IO6	
gpmc_ad[5]	D5 or A6		IO5	
gpmc_ad[4]	D4 or A5		IO4	
gpmc_ad[3]	D3 or A4		IO3	
gpmc_ad[2]	D2 or A3		IO2	
gpmc_ad[1]	D1 or A2		IO1	
gpmc_ad[0]	D0 or A1		IO0	
gpmc_clk	CLK			
gpmc_cs0	nCS0 (chip-select)		nCE0 (chip-enable)	
gpmc_cs1	nCS1		nCE1	
gpmc_cs2	nCS2		nCE2	
gpmc_cs3	nCS3		nCE3	
gpmc_cs4	nCS4		nCE4	
gpmc_cs5	nCS5		nCE5	
gpmc_cs6	nCS6		nCE6	
gpmc_cs7	nCS7		nCE7	
gpmc_advn_ale	nADV (address valid)		ALE (address latch enable)	
gpmc_oen_ren	nOE (output enable)		nRE (read enable)	
gpmc_wen	nWE (Write enable)		nWE (write enable)	
gpmc_ben0	nBE0 (byte enable)		CLE (command latch enable)	
gpmc_ben1	nBE1			
gpmc_wait0	WAIT0		R/nB0 (ready/busy)	
gpmc_wait1	WAIT1		R/nB1	

10.3.6.2.1.2 NAND Interface Protocol

NAND flash architecture, introduced in 1989, is a flash technology. NAND is a page-oriented memory device; that is, read and write accesses are done by pages. NAND achieves great density by sharing common areas of the storage transistor, which creates strings of serially connected transistors (in NOR devices, each transistor stands alone). Because of its high density NAND is best suited to devices that require high capacity data storage, such as pictures, music, and data files. NAND nonvolatility makes of it a good storage solution for many applications where mobility, low power, and speed are key factors. Low pin count and simple interface are other advantages of NAND.

Table 10-412 summarizes the NAND interface signals level applied to external device or memories.

Table 10-412. NAND Interface Bus Operations Summary

Bus Operation	CLE	ALE	nCE	nWE [1]	nRE [1]
Read (cmd input)	H	L	L	RE	H
Read (add input)	L	H	L	RE	H

[1] RE stands for rising edge; FE stands for falling edge

Table 10-412. NAND Interface Bus Operations Summary (continued)

Bus Operation	CLE	ALE	nCE	nWE [1]	nRE [1]
Write (cmd input)	H	L	L	RE	H
Write (add input)	L	H	L	RE	H
Data input	L	L	L	RE	H
Data output	L	L	L	H	FE
Busy (during read)	x	x	H [2]	H [2]	H [2]
Busy (during program)	x	x	x	x	x
Busy (during erase)	x	x	x	x	x
Standby	x	x	H	x	x

[2] Can be nCE high, or WE and nRE high.

10.3.6.2.1.3 NOR Interface Protocol

NOR flash architecture, introduced in 1988, is a flash technology. Unlike NAND, which is a sequential access device, NOR is directly addressable; that is, it is designed to be a random access device. NOR is best suited to devices used to store and run code or firmware, usually in small capacities. While NOR has fast read capabilities, it also has slow write and erase functions when compared to the NAND architecture.

Table 10-413 summarizes the level of the NOR interface signals applied to external devices or memories.

Table 10-413. NOR Interface Bus Operations Summary

Bus Operation	CLK	nADV	nCS	nOE	nWE	WAIT	DQ[15:0]
Read (asynchronous)	x	L	L	L	H	Asserted	Output
Read (synchronous)	Running	L	L	L	H	Driven	Output
Read (burst suspend)	Halted	x	L	H	H	Active	Output
Write	x	L	L	H	L	Asserted	Input
Output disable	x	x	L	H	H	Asserted	High-Z
Standby	x	x	H	x	x	High-Z	High-Z

10.3.6.2.1.4 Other Technologies

Other supported device types interact with the GPMC through the NOR interface protocol.

OneNAND is a high-density, low-power memory device. OneNAND is based on single- or multilevel-cell NAND core with SRAM and logic. It interfaces as a synchronous NOR flash and has synchronous write capability. It reads faster than conventional NAND and writes faster than conventional NOR flash. Hence, it is appropriate for mass storage and code storage.

pSRAM (pseudo-static random access memory) is a low-power memory device. pSRAM is based on the DRAM cell with internal refresh and address control features, and interfaces as a synchronous NOR flash. It also has synchronous write capability.

10.3.6.2.1.5 Supported Protocols

The GPMC supports the following interface protocols when communicating with external memory or external devices:

- Asynchronous read/write access
- Asynchronous read page access (4, 8, 16 Word16)
- Synchronous read/write access
- Synchronous read burst access without wrap capability (4, 8, 16 Word16)
- Synchronous read burst access with wrap capability (4, 8, 16 Word16)

10.3.6.2.2 GPMC Features and Settings

The features and settings of the GPMC are:

- Supported device type: Up to four NAND or NOR protocol external memories or devices
- Operating voltage: 1.8 V, 3.3 V
- Maximum operating frequency provided externally: See the device data manual for precise information.
- Maximum GPMC addressing capability: 512MiB divided into eight chip-selects
- Maximum supported memory size: 256MiB (must be a power-of-two)
- Minimum supported memory size: 16MiB (must be a power-of-two). Aliasing occurs when addressing smaller memories.
- Data path to external memory or device: 8, 16 bits wide
- Burst and page access: burst of 4, 8, 16 Word16
- Supports bus keeping
- Supports bus turnaround

10.3.7 GPMC Register Manual

This section provides information about the GPMC instance in this product. Table 10-415 provides a summary of the GPMC registers. The remaining parts of this section describe the registers within the module instance.

10.3.7.1 GPMC Register Summary

Table 10-414. GPMC Instance Summary

Module Name	Base Address	Size
GPMC	0x5000 0000	16 MiB

Table 10-415. GPMC Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	Physical Address GPMC
GPMC_REVISION	R	32	0x0000 0000	0x5000 0000
GPMC_SYSCONFIG	RW	32	0x0000 0010	0x5000 0010
GPMC_SYSSTATUS	R	32	0x0000 0014	0x5000 0014
GPMC_IRQSTATUS	RW	32	0x0000 0018	0x5000 0018
GPMC_IRQENABLE	RW	32	0x0000 001C	0x5000 001C
GPMC_TIMEOUT_CONTROL	RW	32	0x0000 0040	0x5000 0040
GPMC_ERR_ADDRESS	RW	32	0x0000 0044	0x5000 0044
GPMC_ERR_TYPE	RW	32	0x0000 0048	0x5000 0048
GPMC_CONFIG	RW	32	0x0000 0050	0x5000 0050
GPMC_STATUS	RW	32	0x0000 0054	0x5000 0054
GPMC_CONFIG1_i (1)	RW	32	0x0000 0060 + (0x0000 0030 * i)	0x5000 0060 + (0x0000 0030 * i)
GPMC_CONFIG2_i (1)	RW	32	0x0000 0064 + (0x0000 0030 * i)	0x5000 0064 + (0x0000 0030 * i)
GPMC_CONFIG3_i (1)	RW	32	0x0000 0068 + (0x0000 0030 * i)	0x5000 0068 + (0x0000 0030 * i)
GPMC_CONFIG4_i (1)	RW	32	0x0000 006C + (0x0000 0030 * i)	0x5000 006C + (0x0000 0030 * i)
GPMC_CONFIG5_i (1)	RW	32	0x0000 0070 + (0x0000 0030 * i)	0x5000 0070 + (0x0000 0030 * i)
GPMC_CONFIG6_i (1)	RW	32	0x0000 0074 + (0x0000 0030 * i)	0x5000 0074 + (0x0000 0030 * i)
GPMC_CONFIG7_i (1)	RW	32	0x0000 0078 + (0x0000 0030 * i)	0x5000 0078 + (0x0000 0030 * i)
GPMC_NAND_COMMAND_i (1)	W	32	0x0000 007C + (0x0000 0030 * i)	0x5000 007C + (0x0000 0030 * i)
GPMC_NAND_ADDRESS_i (1)	W	32	0x0000 0080 + (0x0000 0030 * i)	0x5000 0080 + (0x0000 0030 * i)
GPMC_NAND_DATA_i (1)	RW	32	0x0000 0084 + (0x0000 0030 * i)	0x5000 0084 + (0x0000 0030 * i)
GPMC_PREFETCH_CONFIG1	RW	32	0x0000 01E0	0x5000 01E0
GPMC_PREFETCH_CONFIG2	RW	32	0x0000 01E4	0x5000 01E4
GPMC_PREFETCH_CONTROL	RW	32	0x0000 01EC	0x5000 01EC
GPMC_PREFETCH_STATUS	RW	32	0x0000 01F0	0x5000 01F0
GPMC_ECC_CONFIG	RW	32	0x0000 01F4	0x5000 01F4
GPMC_ECC_CONTROL	RW	32	0x0000 01F8	0x5000 01F8
GPMC_ECC_SIZE_CONFIG	RW	32	0x0000 01FC	0x5000 01FC
GPMC_ECCj_RESULT (2)	RW	32	0x0000 0200 + (0x0000 0004 * j)	0x5000 0200 + (0x0000 0004 * j)
GPMC_BCH_RESULT0_i (1)	RW	32	0x0000 0240 + (0x0000 0010 * i)	0x5000 0240 + (0x0000 0010 * i)
GPMC_BCH_RESULT1_i (1)	RW	32	0x0000 0244 + (0x0000 0010 * i)	0x5000 0244 + (0x0000 0010 * i)
GPMC_BCH_RESULT2_i (1)	RW	32	0x0000 0248 + (0x0000 0010 * i)	0x5000 0248 + (0x0000 0010 * i)
GPMC_BCH_RESULT3_i (1)	RW	32	0x0000 024C + (0x0000 0010 * i)	0x5000 024C + (0x0000 0010 * i)
GPMC_BCH_RESULT4_i (1)	RW	32	0x0000 0300 + (0x0000 0010 * i)	0x5000 0300 + (0x0000 0010 * i)
GPMC_BCH_RESULT5_i (1)	RW	32	0x0000 0304 + (0x0000 0010 * i)	0x5000 0304 + (0x0000 0010 * i)
GPMC_BCH_RESULT6_i (1)	RW	32	0x0000 0308 + (0x0000 0010 * i)	0x5000 0308 + (0x0000 0010 * i)
GPMC_BCH_SWDATA	RW	32	0x0000 02D0	0x5000 02D0

(1) i = 0 to 7 for GPMC
(2) j = 0 to 8 for GPMC

10.3.7.2 GPMC Register Descriptions

NOTE: All GPMC registers are aligned to 32-bit address boundaries. All register file accesses, except to GPMC_NAND_DATA_i register, are little-endian. If the GPMC_NAND_DATA_i register location is accessed, the endianness is access-dependent.

In this section i corresponds to the chip-select number, where i = 0 to 7.

Table 10-416. GPMC_REVISION

Address Offset	0x0000 0000		
Physical Address	0x5000 0000	**Instance**	GPMC
Description	This register contains the IP revision code.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REVISION																															

Bits	Field Name	Description	Type	Reset
31:0	REVISION	IP revision	R	TI internal data

Table 10-417. Register Call Summary for Register GPMC_REVISION

GPMC Register Manual
- GPMC Register Summary: [0]

Table 10-418. GPMC_SYSCONFIG

Address Offset	0x0000 0010		
Physical Address	0x5000 0010	**Instance**	GPMC
Description	This register controls the various parameters of the interconnect.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																											IDLEMODE		RESERVED	SOFTRESET	AUTOIDLE

Bits	Field Name	Description	Type	Reset
31:5	RESERVED	Write 0s for future compatibility. Read returns 0.	RW	0x0000000
4:3	IDLEMODE	0x0: Force-idle. An idle request is acknowledged unconditionally. 0x1: No-idle. An idle request is never acknowledged. 0x2: Smart-idle. Acknowledgment to an idle request is given based on the internal activity of the module. 0x3: Do not use.	RW	0x0
2	RESERVED	Write 0 for future compatibility Read returns 0.	RW	0x0
1	SOFTRESET	Software reset. Set this bit to 1 triggers a module reset. This bit is automatically reset by hardware. During reads, it always returns 0. 0x0: Normal mode 0x1: The module is reset.	RW	0x0

Bits	Field Name	Description	Type	Reset
0	AUTOIDLE	Internal interface clock-gating strategy 0x0: Interface clock is free-running. 0x1: Automatic Interface clock gating strategy is applied, based on the interconnect activity.	RW	0x0

Table 10-419. Register Call Summary for Register GPMC_SYSCONFIG

GPMC Functional Description • GPMC Software Reset: [0] • GPMC Power Management: [1][2]
GPMC Basic Programming Model • GPMC Initialization: [3]
GPMC Register Manual • GPMC Register Summary: [4]

Table 10-420. GPMC_SYSSTATUS

Address Offset	0x0000 0014		
Physical Address	0x5000 0014	**Instance**	GPMC
Description	This register provides status information about the module, excluding the interrupt status information.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								RESERVED							RESETDONE

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Read returns 0.	R	0x000000
7:1	RESERVED	Read returns 0 (reserved for interconnect-socket status information).	R	0x00
0	RESETDONE	Internal reset monitoring 0x0: Internal module reset is ongoing. 0x1: Reset is complete.	R	0x-

Table 10-421. Register Call Summary for Register GPMC_SYSSTATUS

GPMC Functional Description • GPMC Software Reset: [0]
GPMC Basic Programming Model • GPMC Initialization: [1]
GPMC Register Manual • GPMC Register Summary: [2]

Table 10-422. GPMC_IRQSTATUS

Address Offset	0x0000 0018		
Physical Address	0x5000 0018	**Instance**	GPMC
Description	This interrupt status register regroups all the status of the module internal events that can generate an interrupt.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						WAIT1EDGEDETECTIONSTATUS	WAIT0EDGEDETECTIONSTATUS	RESERVED						TERMINALCOUNTSTATUS	FIFOEVENTSTATUS

Bits	Field Name	Description	Type	Reset
31:10	RESERVED	Write 0s for future compatibility. Read returns 0s.	RW	0x000000
9	WAIT1EDGEDETECTIONSTATUS	Status of the Wait1 Edge Detection interrupt Read 0x0: A transition on WAIT1 input pin has not been detected. Write 0x0: WAIT1EDGEDETECTIONSTATUS bit is unchanged. Read 0x1: A transition on WAIT1 input pin has been detected. Write 0x1: WAIT1EDGEDETECTIONSTATUS bit is reset.	RW	0x0
8	WAIT0EDGEDETECTIONSTATUS	Status of the Wait0 Edge Detection interrupt Read 0x0: A transition on WAIT0 input pin has not been detected. Write 0x0: WAIT0EDGEDETECTIONSTATUS bit is unchanged. Read 0x1: A transition on WAIT0 input pin has been detected. Write 0x1: WAIT0EDGEDETECTIONSTATUS bit is reset.	RW	0x0
7:2	RESERVED	Write 0s for future compatibility. Read returns 0s.	RW	0x00
1	TERMINALCOUNTSTATUS	Status of the TerminalCountEvent interrupt Read 0x0: Indicates that CountValue is greater than 0 Write 0x0: TERMINALCOUNTSTATUS bit is unchanged. Read 0x1: Indicates that CountValue is equal to 0 Write 0x1: TERMINALCOUNTSTATUS bit is reset.	RW	0x0
0	FIFOEVENTSTATUS	Status of the FIFOEvent interrupt Read 0x0: Indicates that less than GPMC_PREFETCH_STATUS[16] FIFOTHRESHOLDSTATUS bytes are available in prefetch mode and less than FIFOTHRESHOLD bytes free places are available in write-posting mode Write 0x0: FIFOEVENTSTATUS bit is unchanged. Read 0x1: Indicates that at least GPMC_PREFETCH_STATUS[16] FIFOTHRESHOLDSTATUS bytes are available in prefetch mode and at least FIFOTHRESHOLD bytes free places are available in write-posting mode Write 0x1: FIFOEVENTSTATUS bit is reset.	RW	0x0

Table 10-423. Register Call Summary for Register GPMC_IRQSTATUS

GPMC Functional Description

- GPMC Interrupt Requests: [0][1][2][3]
- Ready Pin Monitored by Hardware Interrupt: [4][5][6]
- FIFO Control in Prefetch Mode: [7][8]
- FIFO Control in Write-Posting Mode: [9][10]

GPMC Register Manual

- GPMC Register Summary: [11]

Table 10-424. GPMC_IRQENABLE

Address Offset	0x0000 001C		
Physical Address	0x5000 001C	**Instance**	GPMC
Description	The interrupt enable register allows to mask/unmask the module internal sources of interrupt, on a event-by-event basis.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						WAIT1EDGEDETECTIONENABLE	WAIT0EDGEDETECTIONENABLE	RESERVED						TERMINALCOUNTEVENTENABLE	FIFOEVENTENABLE

Bits	Field Name	Description	Type	Reset
31:10	RESERVED	Write 0s for future compatibility. Read returns 0s.	RW	0x000000
9	WAIT1EDGEDETECTIONENABLE	Enables the Wait1 Edge Detection interrupt 0x0: Wait1EdgeDetection interrupt is masked. 0x1: Wait1EdgeDetection event generates an interrupt if occurs.	RW	0x0
8	WAIT0EDGEDETECTIONENABLE	Enables the Wait0 Edge Detection interrupt 0x0: Wait0EdgeDetection interrupt is masked. 0x1: Wait0EdgeDetection event generates an interrupt if occurs.	RW	0x0
7:2	RESERVED	Write 0s for future compatibility. Read returns 0s.	RW	0x00
1	TERMINALCOUNTEVENTENABLE	Enables TerminalCountEvent interrupt issuing in prefetch or write-posting mode 0x0: TerminalCountEvent interrupt is masked. 0x1: TerminalCountEvent interrupt is not masked.	RW	0x0
0	FIFOEVENTENABLE	Enables the FIFOEvent interrupt 0x0: FIFOEvent interrupt is masked. 0x1: FIFOEvent interrupt is not masked.	RW	0x0

Table 10-425. Register Call Summary for Register GPMC_IRQENABLE

GPMC Functional Description • GPMC Interrupt Requests: [0][1][2][3] • NAND Device-Ready Pin: [4] • Ready Pin Monitored by Hardware Interrupt: [5] • FIFO Control in Prefetch Mode: [6][7] • FIFO Control in Write-Posting Mode: [8][9]
GPMC Register Manual • GPMC Register Summary: [10]

Table 10-426. GPMC_TIMEOUT_CONTROL

Address Offset	0x0000 0040		
Physical Address	0x5000 0040	**Instance**	GPMC
Description	The GPMC_TIMEOUT_CONTROL register allows the user to set the start value of the timeout counter.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																TIMEOUTSTARTVALUE								RESERVED							TIMEOUTENABLE

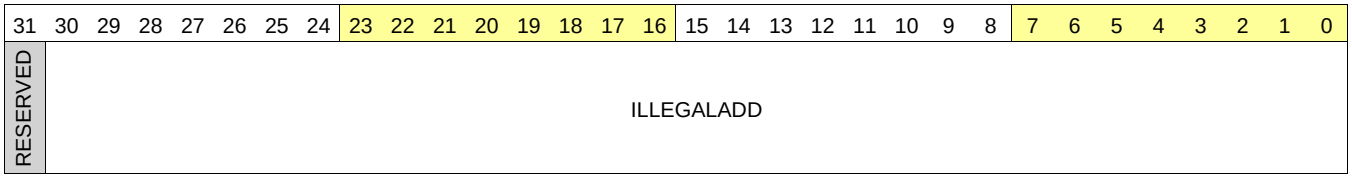
Bits	Field Name	Description	Type	Reset
31:13	RESERVED	Write 0s for future compatibility. Read returns 0s.	RW	0x00000
12:4	TIMEOUTSTARTVALUE	Start value of the time-out counter 0x000: Zero GPMC_FCLK cycle 0x001: One GPMC_FCLK cycle ... 0x1FF: 511 GPMC_FCLK cycles	RW	0x1FF
3:1	RESERVED	Write 0s for future compatibility. Read returns 0s.	RW	0x0
0	TIMEOUTENABLE	Enable bit of the TimeOut feature 0x0: TimeOut feature is disabled. 0x1: TimeOut feature is enabled.	RW	0x0

Table 10-427. Register Call Summary for Register GPMC_TIMEOUT_CONTROL

GPMC Functional Description • Error Handling: [0][1]
GPMC Register Manual • GPMC Register Summary: [2] • GPMC Register Descriptions: [3]

Table 10-428. GPMC_ERR_ADDRESS

Address Offset	0x0000 0044		
Physical Address	0x5000 0044	**Instance**	GPMC
Description	The GPMC_ERR_ADDRESS register stores the address of the illegal access when an error occurs.		
Type	RW		



Bits	Field Name	Description	Type	Reset
31	RESERVED	Write 0s for future compatibility. Read returns 0.	RW	0x0
30:0	ILLEGALADD	Address of illegal access A30: 0 for memory region, 1 for GPMC register region A29-A0: 1 GiB maximum	R	0x00000000

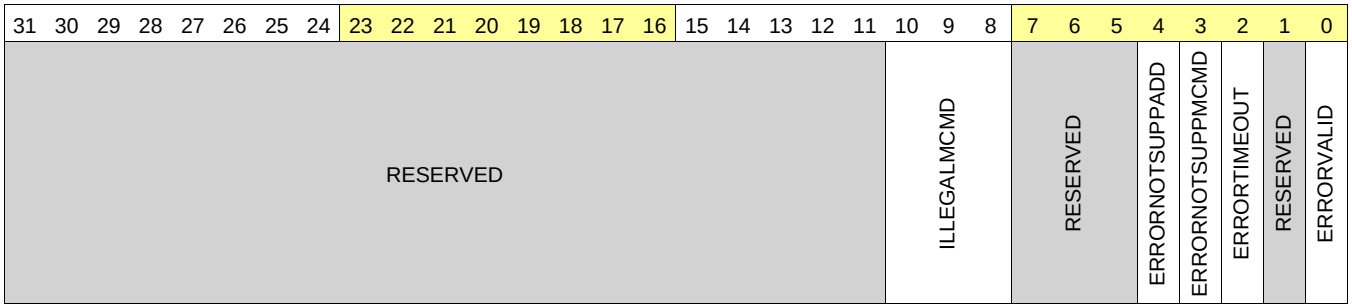
Table 10-429. Register Call Summary for Register GPMC_ERR_ADDRESS

GPMC Functional Description
- Error Handling: [0]

GPMC Register Manual
- GPMC Register Summary: [1]
- GPMC Register Descriptions: [2]

Table 10-430. GPMC_ERR_TYPE

Address Offset	0x0000 0048		
Physical Address	0x5000 0048	**Instance**	GPMC
Description	The GPMC_ERR_TYPE register stores the type of error when an error occurs.		
Type	RW		



Bits	Field Name	Description	Type	Reset
31:11	RESERVED	Write 0s for future compatibility. Read returns 0s.	RW	0x000000
10:8	ILLEGALMCMD	System command of the transaction that caused the error	R	0x0
7:5	RESERVED	Write 0s for future compatibility. Read returns 0s.	RW	0x0
4	ERRORNOTSUPPADD	Not supported address error 0x0: No error occurs. 0x1: The error is due to a nonsupported address.	R	0x0

Bits	Field Name	Description	Type	Reset
3	ERRORNOTSUPPMCMD	Not supported command error 0x0: No error occurs. 0x1: The error is due to a nonsupported command	R	0x0
2	ERRORTIMEOUT	Time-out error 0x0: No error occurs. 0x1: The error is due to a timeout.	R	0x0
1	RESERVED	Write 0s for future compatibility. Read returns 0.	RW	0x0
0	ERRORVALID	Error validity status - Must be explicitly cleared with a write 1 transaction 0x0: All error fields no longer valid 0x1: Error detected and logged in the other error fields	RW	0x0

Table 10-431. Register Call Summary for Register GPMC_ERR_TYPE

GPMC Functional Description
- Error Handling: [0][1][2]

GPMC Register Manual
- GPMC Register Summary: [3]
- GPMC Register Descriptions: [4]

Table 10-432. GPMC_CONFIG

Address Offset	0x0000 0050		
Physical Address	0x5000 0050	**Instance**	GPMC
Description	The configuration register allows global configuration of the GPMC.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						WAIT1PINPOLARITY	WAIT0PINPOLARITY	RESERVED						RESERVED	NANDFORCEPOSTEDWRITE

Bits	Field Name	Description	Type	Reset
31:10	RESERVED	Write 0s for future compatibility. Read returns 0s.	RW	0x000000
9	WAIT1PINPOLARITY	Selects the polarity of input pin WAIT1 0x0: WAIT1 active low 0x1: WAIT1 active high	RW	0x1
8	WAIT0PINPOLARITY	Selects the polarity of input pin WAIT0 0x0: WAIT0 active low 0x1: WAIT0 active high	RW	0x0
7:2	RESERVED	Write 0s for future compatibility. Read returns 0s.	RW	0x00
1	RESERVED	Write 0 for future compatibility. Read returns 0.	RW	0x0

Bits	Field Name	Description	Type	Reset
0	NANDFORCEPOSTEDWRITE	Enables the Force Posted Write feature to NAND Cmd/Add/Data location 0x0: Disables Force Posted Write 0x1: Enables Force Posted Write	RW	0x0

Table 10-433. Register Call Summary for Register GPMC_CONFIG

GPMC Functional Description
- GPMC Interrupt Requests: [0][1]
- Wait Pin Monitoring Control: [2]
- Asynchronous Single-Read Operation on Nonmultiplexed Device: [3][4]
- NAND Device Command and Address Phase Control: [5]
- Ready Pin Monitored by Hardware Interrupt: [6]

GPMC Register Manual
- GPMC Register Summary: [7]

Table 10-434. GPMC_STATUS

Address Offset	0x0000 0054		
Physical Address	0x5000 0054	**Instance**	GPMC
Description	The status register provides global status bits of the GPMC.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						WAIT1STATUS	WAIT0STATUS	RESERVED							EMPTYWRITEBUFFERSTATUS

Bits	Field Name	Description	Type	Reset
31:10	RESERVED	Write 0s for future compatibility. Read returns 0s.	RW	0x000000
9	WAIT1STATUS	Is a copy of input pin WAIT1. (Reset value is WAIT1 input pin sampled at device reset.) 0x0: WAIT1 asserted (inactive state) 0x1: WAIT1 deasserted	R	0x-
8	WAIT0STATUS	Is a copy of input pin WAIT0. (Reset value is WAIT0 input pin sampled at device reset.) 0x0: WAIT0 asserted (inactive state) 0x1: WAIT0 deasserted	R	0x-
7:1	RESERVED	Write 0s for future compatibility. Reads returns 0	RW	0x00
0	EMPTYWRITEBUFFERSTATUS	Stores the empty status of the write buffer 0x0: Write buffer is not empty. 0x1: Write buffer is empty.	R	0x1

Table 10-435. Register Call Summary for Register GPMC_STATUS

GPMC Functional Description

- NAND Device Command and Address Phase Control: [0]
- NAND Device-Ready Pin: [1][2]
- Ready Pin Monitored by Software Polling: [3]

GPMC Register Manual

- GPMC Register Summary: [4]

Table 10-436. GPMC_CONFIG1_i

Address Offset	0x0000 0060 + (0x0000 0030 * i)	**Index**	i = 0 to 7
Physical Address	0x5000 0060 + (0x0000 0030 * i)	**Instance**	GPMC
Description	The configuration register 1 sets signal control parameters per chip-select.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WRAPBURST	READMULTIPLE	READTYPE	WRITEMULTIPLE	WRITETYPE	CLKACTIVATIONTIME			ATTACHEDDEVICEPAGELENGTH		WAITREADMONITORING	WAITWRITEMONITORING	RESERVED		WAITMONITORINGTIME		WAITPINSELECT		RESERVED		DEVICESIZE		DEVICETYPE		MUXADDDATA		RESERVED			TIMEPARAGRANULARITY	RESERVED		GPMCFCLKDIVIDER	

Bits	Field Name	Description	Type	Reset
31	WRAPBURST	Enables the wrapping burst capability. Must be set if the attached device is configured in wrapping burst 0x0: Synchronous wrapping burst not supported 0x1: Synchronous wrapping burst supported	RW	0x0
30	READMULTIPLE	Selects the read single or multiple access 0x0: Single access 0x1: Multiple access (burst if synchronous, page if asynchronous)	RW	0x0
29	READTYPE	Selects the read mode operation 0x0: Read asynchronous 0x1: Read synchronous	RW	0x0
28	WRITEMULTIPLE	Selects the write single or multiple access 0x0: Single access 0x1: Multiple access (burst if synchronous, considered as single if asynchronous)	RW	0x0
27	WRITETYPE	Selects the write mode operation 0x0: Write asynchronous 0x1: Write synchronous	RW	0x0

Bits	Field Name	Description	Type	Reset
26:25	CLKACTIVATIONTIME	Output GPMC_CLK activation time 0x0: First rising edge of GPMC_CLK at start access time 0x1: First rising edge of GPMC_CLK one GPMC_FCLK cycle after start access time 0x2: First rising edge of GPMC_CLK two GPMC_FCLK cycles after start access time 0x3: Reserved	RW	0x0
24:23	ATTACHEDDEVICEPAGELENGTH	Specifies the attached device page (burst) length 0x0: 4 words 0x1: 8 words 0x2: 16 words 0x3: Reserved (1 word = interface size)	RW	0x0
22	WAITREADMONITORING	Selects the Wait monitoring configuration for Read accesses (Reset value is *bootwaiten* input pin sampled at device reset) 0x0: Wait pin is not monitored for read accesses. 0x1: Wait pin is monitored for read accesses.	RW	0x-
21	WAITWRITEMONITORING	Selects the Wait monitoring configuration for Write accesses 0x0: Wait pin is not monitored for write accesses. 0x1: Wait pin is monitored for write accesses.	RW	0x0
20	RESERVED	Write 0s for future compatibility. Read returns 0.	RW	0x0
19:18	WAITMONITORINGTIME	Selects input pin Wait monitoring time 0x0: Wait pin is monitored with valid data. 0x1: Wait pin is monitored one GPMC_CLK cycle before valid data. 0x2: Wait pin is monitored two GPMC_CLK cycle before valid data. 0x3: Reserved	RW	0x0
17:16	WAITPINSELECT	Selects the input wait pin for this chip-select (The reset value is HW fixed to 0x0 for CS0-CS7) 0x0: Wait input pin is WAIT0. 0x1: Wait input pin is WAIT1. 0x2, 0x3: Reserved	RW	0x0
15:14	RESERVED	Write 0s for future compatibility. Reads returns 0	RW	0x0
13:12	DEVICESIZE	Selects the device size attached (Reset value is *bootdevicesize* input pin sampled at device reset for CS0 and 0x1 for CS1 to CS7) 0x0: 8 bit 0x1: 16 bit 0x2: Reserved 0x3: Reserved	RW	0x-
11:10	DEVICETYPE	Selects the attached device type 0x0: NOR flash-like, asynchronous and synchronous devices 0x1: Reserved 0x2: NAND flash-like devices, stream mode 0x3: Reserved	RW	0x0

Bits	Field Name	Description	Type	Reset
9:8	MUXADDDATA	Enables the address and data multiplexed protocol (Reset value is *cs0muxdevice* input pin sampled at device reset for CS0 and 0 for CS1-CS7) 0x0: Nonmultiplexed attached device 0x1: AAD-multiplexed protocol device 0x2: Address and data multiplexed attached device 0x3: Reserved	RW	0x-
7:5	RESERVED	Write 0s for future compatibility. Read returns 0s.	RW	0x0
4	TIMEPARAGRANULARITY	Signals timing latencies scalar factor (RD/WRCYCLETIME, RD/WRACCESSTIME, PAGEBURSTACCESSTIME, CSONTIME, CSRD/WROFFTIME, ADVONTIME, ADVRD/WROFFTIME, OEONTIME, OEOFFTIME, WEONTIME, WEOFFTIME, CYCLE2CYCLEDELAY, BUSTURNAROUND, TIMEOUTSTARTVALUE, WRDATAONADMUXBUS) 0x0: x1 latencies 0x1: x2 latencies	RW	0x0
3:2	RESERVED	Write 0s for future compatibility. Read returns 0s.	RW	0x0
1:0	GPMCFCLKDIVIDER	Divides the GPMC_FCLK clock 0x0: GPMC_CLK frequency = GPMC_FCLK frequency 0x1: GPMC_CLK frequency = GPMC_FCLK frequency / 2 0x2: GPMC_CLK frequency = GPMC_FCLK frequency / 3 0x3: GPMC_CLK frequency = GPMC_FCLK frequency /4	RW	0x0

Table 10-437. Register Call Summary for Register GPMC_CONFIG1_i

GPMC Environment
- GPMC Signals: [0]

GPMC Functional Description
- GPMC Clock Configuration: [1][2]
- L3 Interconnect Interface: [3][4]
- GPMC I/O Configuration Setting: [5][6][7][8][9][10][11][12]
- GPMC CS0 Default Configuration at Device Reset: [13][14][15][16]
- Supported Devices: [17]
- Access Size Adaptation and Device Width: [18]
- Address/Data-Multiplexing Interface: [19]
- Wait Pin Monitoring Control: [20][21][22][23][24][25][26][27][28][29][30][31][32][33][34][35][36]
- Read Cycle Time and Write Cycle Time (RDCYCLETIME / WRCYCLETIME): [37]
- GPMC_CLK: [38][39]
- GPMC_CLK and Control Signals Setup and Hold: [40]
- Access Time (RDACCESSTIME / WRACCESSTIME): [41]
- Page Burst Access Time (PAGEBURSTACCESSTIME): [42]
- NOR Access Description: [43][44]
- Access on Address/Data Multiplexed Devices: [45][46]
- Access on Address/Address/Data-Multiplexed Devices: [47][48]
- Synchronous Access Description: [49][50]
- Synchronous Multiple (Burst) Read (4-, 8-, 16-Word16 Burst With Wraparound Capability): [51][52][53]
- Page and Burst Support: [54][55]
- System Burst vs External Device Burst Support: [56][57][58]
- pSRAM Access Specificities: [59]
- Chip-Select Configuration for NAND Interfacing in Byte or Word Stream Mode: [60][61][62][63][64][65][66][67][68][69][70][71][72][73][74][75][76][77]
- NAND Device-Ready Pin: [78][79]

Table 10-437. Register Call Summary for Register GPMC_CONFIG1_i (continued)

GPMC Basic Programming Model

- GPMC Configuration in NOR Mode: [80][81][82][83][84][85][86][87][88][89][90][91][92][93][94][95]
- GPMC Configuration in NAND Mode: [96][97][98][99]
- Set Memory Access: [100][101][102][103][104]
- GPMC Timing Parameters: [105][106][107][108][109][110][111][112][113]
- NAND Flash Interface Timing Parameters Formulas: [114]
- Synchronous NOR Flash Timing Parameters Formulas: [115][116][117][118][119][120][121][122][123][124][125]
- Asynchronous NOR Flash Timing Parameters Formulas: [126]

GPMC Register Manual

- GPMC Register Summary: [127]

Table 10-438. GPMC_CONFIG2_i

Address Offset	0x0000 0064 + (0x0000 0030 * i)	**Index**	i = 0 to 7
Physical Address	0x5000 0064 + (0x0000 0030 * i)	**Instance**	GPMC
Description	CS signal timing parameter configuration		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								CSWROFFTIME								RESERVED			CSRDOFFTIME					CSEXTRADELAY	RESERVED					CSONTIME			

Bits	Field Name	Description	Type	Reset
31:21	RESERVED	Write 0s for future compatibility. Read returns 0s.	RW	0x000
20:16	CSWROFFTIME	CS i deassertion time from start cycle time for write accesses 0x00: 0 GPMC_FCLK cycle 0x01: 1 GPMC_FCLK cycle ... 0x1F: 31 GPMC_FCLK cycles	RW	0x10
15:13	RESERVED	Write 0s for future compatibility. Read returns 0s.	RW	0x0
12:8	CSRDOFFTIME	CS i de-assertion time from start cycle time for read accesses 0x00: 0 GPMC_FCLK cycle 0x01: 1 GPMC_FCLK cycle ... 0x1F: 31 GPMC_FCLK cycles	RW	0x10
7	CSEXTRADELAY	CS i Add extra half-GPMC_FCLK cycle 0x0: CS i Timing control signal is not delayed 0x1: CS i Timing control signal is delayed of half GPMC_FCLK clock cycle	RW	0x0
6:4	RESERVED	Write 0s for future compatibility. Read returns 0s.	RW	0x0
3:0	CSONTIME	CS i assertion time from start cycle time 0x0: 0 GPMC_FCLK cycle 0x1: 1 GPMC_FCLK cycle ... 0xF: 15 GPMC_FCLK cycles	RW	0x1

Table 10-439. Register Call Summary for Register GPMC_CONFIG2_i

GPMC Functional Description

- nCS: Chip-Select Signal Control Assertion/Deassertion Time (CSONTIME / CSRDOFFTIME / CSWROFFTIME / CSEXTRADELAY): [0][1][2][3]
- Access on Address/Data Multiplexed Devices: [4][5]
- Synchronous Single Read: [6][7]
- Synchronous Multiple (Burst) Write: [8][9]

GPMC Basic Programming Model

- GPMC Timing Parameters: [10][11][12][13]

GPMC Register Manual

- GPMC Register Summary: [14]

Table 10-440. GPMC_CONFIG3_i

Address Offset	0x0000 0068 + (0x0000 0030 * i)	**Index**	i = 0 to 7
Physical Address	0x5000 0068 + (0x0000 0030 * i)	**Instance**	GPMC
Description	nADV signal timing parameter configuration		
Type	RW		

31	30 29 28	27	26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7	6 5 4	3 2 1 0
RESERVED	ADVAADMUXWROFFTIME	RESERVED	ADVAADMUXRDOFFTIME	RESERVED	ADVWROFFTIME	RESERVED	ADVRDOFFTIME	ADVEXTRADELAY	ADVAADMUXONTIME	ADVONTIME

Bits	Field Name	Description	Type	Reset
31	RESERVED	Write 0s for future compatibility. Read returns 0s.	RW	0
30:28	ADVAADMUXWROFFTIME	nADV deassertion for first address phase when using the AAD-multiplexed protocol 0x0: 0 GPMC_FCLK cycle 0x1: 1 GPMC_FCLK cycle ... 0x7: 7 GPMC_FCLK cycles	RW	0x2
27	RESERVED	Write 0s for future compatibility. Read returns 0s.	RW	0
26:24	ADVAADMUXRDOFFTIME	nADV assertion for first address phase when using the AAD-multiplexed protocol 0x0: 0 GPMC_FCLK cycle 0x1: 1 GPMC_FCLK cycle ... 0x7: 7 GPMC_FCLK cycles	RW	0x2
23:21	RESERVED	Write 0s for future compatibility. Read returns 0s.	RW	0x0
20:16	ADVWROFFTIME	nADV deassertion time from start cycle time for write accesses 0x00: 0 GPMC_FCLK cycle 0x01: 1 GPMC_FCLK cycle ... 0x1F: 31 GPMC_FCLK cycles	RW	0x06
15:13	RESERVED	Write 0s for future compatibility. Read returns 0s.	RW	0x0
12:8	ADVRDOFFTIME	nADV deassertion time from start cycle time for read accesses 0x00: 0 GPMC_FCLK cycle 0x01: 1 GPMC_FCLK cycle ... 0x1F: 31 GPMC_FCLK cycles	RW	0x05

Bits	Field Name	Description	Type	Reset
7	ADVEXTRADELAY	nADV add extra half-GPMC_FCLK cycle 0x0: nADV timing control signal is not delayed 0x1: nADV timing control signal is delayed of half GPMC_FCLK clock cycle	RW	0
6:4	ADVAADMUXONTIME	nADV assertion for first address phase when using the AAD-multiplexed protocol 0x0: 0 GPMC_FCLK cycle 0x1: 1 GPMC_FCLK cycle ... 0x7: 7 GPMC_FCLK cycles	RW	0x1
3:0	ADVONTIME	nADV assertion time from start cycle time 0x0: 0 GPMC_FCLK cycle 0x1: 1 GPMC_FCLK cycle ... 0xF: 15 GPMC_FCLK cycles	RW	0x4

Table 10-441. Register Call Summary for Register GPMC_CONFIG3_i

GPMC Functional Description
- nADV/ALE: Address Valid/Address Latch Enable Signal Control Assertion/Deassertion Time (ADVONTIME / ADVRDOFFTIME / ADVWROFFTIME / ADVEXTRADELAY/ADVAADMUXONTIME/ADVAADMUXRDOFFTIME/ADVAADMUXWROFFTIME): [0][1][2][3][4][5][6]
- Access on Address/Data Multiplexed Devices: [7][8]
- Access on Address/Address/Data-Multiplexed Devices: [9][10][11][12][13][14][15][16][17][18]
- Synchronous Single Read: [19][20][21][22][23][24][25]
- Synchronous Multiple (Burst) Write: [26][27][28][29][30][31]

GPMC Basic Programming Model
- GPMC Timing Parameters: [32][33][34][35][36][37][38]

GPMC Register Manual
- GPMC Register Summary: [39]

Table 10-442. GPMC_CONFIG4_i

Address Offset	0x0000 006C + (0x0000 0030 * i)	**Index**	i = 0 to 7
Physical Address	0x5000 006C + (0x0000 0030 * i)	**Instance**	GPMC
Description	nWE and nOE signals timing parameter configuration		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED			WEOFFTIME					WEEXTRADELAY	RESERVED			WEONTIME				OEAADMUX OFFTIME			OEOFFTIME					OEEXTRADELAY	OEAADMUX ONTIME			OEONTIME			

Bits	Field Name	Description	Type	Reset
31:29	RESERVED	Write 0s for future compatibility. Read returns 0s.	RW	0x0
28:24	WEOFFTIME	nWE deassertion time from start cycle time 0x00: 0 GPMC_FCLK cycle 0x01: 1 GPMC_FCLK cycle ... 0x1F: 31 GPMC_FCLK cycles	RW	0x10

Bits	Field Name	Description	Type	Reset
23	WEEXTRADELAY	nWE add extra half-GPMC_FCLK cycle 0x0: nWE timing control signal is not delayed 0x1: nWE timing control signal is delayed of half-GPMC_FCLK clock cycle	RW	0
22:20	RESERVED	Write 0s for future compatibility. Read returns 0s.	RW	0x0
19:16	WEONTIME	nWE assertion time from start cycle time 0x0: 0 GPMC_FCLK cycle 0x1: 1 GPMC_FCLK cycle ... 0xF: 15 GPMC_FCLK cycles	RW	0x5
15:13	OEAADMUX OFFTIME	nOE deassertion time for the first address phase in an AAD-multiplexed access 0x0: 0 GPMC_FCLK cycle 0x1: 1 GPMC_FCLK cycle ... 0x7: 7 GPMC_FCLK cycles	RW	0x3
12:8	OEOFFTIME	nOE deassertion time from start cycle time 0x00: 0 GPMC_FCLK cycle 0x01: 1 GPMC_FCLK cycle ... 0x1F: 31 GPMC_FCLK cycles	RW	0x10
7	OEEXTRADELAY	nOE add extra half-GPMC_FCLK cycle 0x0: nOE timing control signal is not delayed 0x1: nOE timing control signal is delayed of half-GPMC_FCLK clock cycle	RW	0
6:4	OEAADMUX ONTIME	nOE assertion time for the first address phase in an AAD-mux access 0x0: 0 GPMC_FCLK cycle 0x1: 1 GPMC_FCLK cycle ... 0x7: 7 GPMC_FCLK cycles	RW	0x1
3:0	OEONTIME	nOE assertion time from start cycle time 0x0: 0 GPMC_FCLK cycle 0x1: 1 GPMC_FCLK cycle ... 0xF: 15 GPMC_FCLK cycles	RW	0x6

Table 10-443. Register Call Summary for Register GPMC_CONFIG4_i

GPMC Functional Description

- nOE/nRE: Output Enable/Read Enable Signal Control Assertion/Deassertion Time (OEONTIME / OEOFFTIME / OEEXTRADELAY / OEAADMUXONTIME / OEAADMUXOFFTIME): [0][1][2]
- nWE: Write Enable Signal Control Assertion/Deassertion Time (WEONTIME / WEOFFTIME / WEEXTRADELAY): [3][4][5]
- Access on Address/Data Multiplexed Devices: [6][7][8][9]
- Access on Address/Address/Data-Multiplexed Devices: [10][11][12][13]
- Synchronous Single Read: [14][15][16][17][18]
- Synchronous Multiple (Burst) Write: [19][20][21][22][23][24]
- Chip-Select Configuration for NAND Interfacing in Byte or Word Stream Mode: [25]

GPMC Basic Programming Model

- GPMC Timing Parameters: [26][27][28][29][30][31][32][33]

GPMC Register Manual

- GPMC Register Summary: [34]

Table 10-444. GPMC_CONFIG5_i

Address Offset	0x0000 0070 + (0x0000 0030 * i)	**Index**	i = 0 to 7
Physical Address	0x5000 0070 + (0x0000 0030 * i)	**Instance**	GPMC
Description	RdAccessTime and CycleTime timing parameters configuration		
Type	RW		

31 30 29 28	27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
RESERVED	PAGEBURSTACCESSTIME	RESERVED / RDACCESSTIME	RESERVED / WRCYCLETIME	RESERVED / RDCYCLETIME

Bits	Field Name	Description	Type	Reset
31:28	RESERVED	Write 0s for future compatibility. Read returns 0s.	RW	0x0
27:24	PAGEBURSTACCESSTIME	Delay between successive words in a multiple access 0x0: 0 GPMC_FCLK cycle 0x1: 1 GPMC_FCLK cycle ... 0xF: 15 GPMC_FCLK cycles	RW	0x1
23:21	RESERVED	Write 0s for future compatibility. Read returns 0s.	RW	0x0
20:16	RDACCESSTIME	Delay between start cycle time and first data valid 0x00: 0 GPMC_FCLK cycle 0x01: 1 GPMC_FCLK cycle ... 0x1F: 31 GPMC_FCLK cycles	RW	0x0F
15:13	RESERVED	Write 0s for future compatibility. Reads returns 0	RW	0x0
12:8	WRCYCLETIME	Total write cycle time 0x00: 0 GPMC_FCLK cycle 0x01: 1 GPMC_FCLK cycle ... 0x1F: 31 GPMC_FCLK cycles	RW	0x11
7:5	RESERVED	Write 0s for future compatibility. Read returns 0s.	RW	0x0
4:0	RDCYCLETIME	Total read cycle time 0x00: 0 GPMC_FCLK cycle 0x01: 1 GPMC_FCLK cycle ... 0x1F: 31 GPMC_FCLK cycles	RW	0x11

Table 10-445. Register Call Summary for Register GPMC_CONFIG5_i

GPMC Functional Description

- Read Cycle Time and Write Cycle Time (RDCYCLETIME / WRCYCLETIME): [0][1]
- Access Time (RDACCESSTIME / WRACCESSTIME): [2]
- Access Time on Read Access: [3][4]
- Page Burst Access Time (PAGEBURSTACCESSTIME): [5]
- Access on Address/Data Multiplexed Devices: [6][7][8]
- Synchronous Single Read: [9][10]
- Synchronous Multiple (Burst) Read (4-, 8-, 16-Word16 Burst With Wraparound Capability): [11][12][13]
- Synchronous Multiple (Burst) Write: [14][15][16][17]
- Asynchronous Multiple (Page Mode) Read Operation on Nonmultiplexed Device: [18][19][20][21]

GPMC Basic Programming Model

- GPMC Timing Parameters: [22][23][24][25]

Table 10-445. Register Call Summary for Register GPMC_CONFIG5_i (continued)

GPMC Register Manual

- GPMC Register Summary: [26]

Table 10-446. GPMC_CONFIG6_i

Address Offset	0x0000 0074 + (0x0000 0030 * i)	**Index**	i = 0 to7
Physical Address	0x5000 0074 + (0x0000 0030 * i)	**Instance**	GPMC
Description	WrAccessTime, WrDataOnADmuxBus, Cycle2Cycle and BusTurnAround parameters configuration		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED			WRACCESSTIME					RESERVED				WRDATAONADMUXBUS				RESERVED				CYCLE2CYCLEDELAY				CYCLE2CYCLESAMECSEN	CYCLE2CYCLEDIFFCSEN	RESERVED		BUSTURNAROUND			

Bits	Field Name	Description	Type	Reset
31	RESERVED	TI Internal use - Do not modify.	RW	1
30:29	RESERVED	Write 0s for future compatibility. Read returns 0s.	RW	0x0
28:24	WRACCESSTIME	Delay from start access time to the GPMC_FCLK rising edge corresponding the GPMC_CLK rising edge used by the attached memory for the first data capture 0x00: 0 GPMC_FCLK cycle 0x01: 1 GPMC_FCLK cycle ... 0x1F: 31 GPMC_FCLK cycles	RW	0x0F
23:20	RESERVED	Write 0s for future compatibility. Read returns 0s.	RW	0x0
19:16	WRDATAONADMUXBUS	Specifies on which GPMC_FCLK rising edge the first data of the write is driven in the add/data mux bus	RW	0x7
15:12	RESERVED	Write 0s for future compatibility. Read returns 0s.	RW	0x0
11:8	CYCLE2CYCLEDELAY	Chip-select high pulse delay between successive accesses 0x0: 0 GPMC_FCLK cycle 0x1: 1 GPMC_FCLK cycle ... 0xF: 15 GPMC_FCLK cycles	RW	0x0
7	CYCLE2CYCLESAMECSEN	Add CYCLE2CYCLEDELAY between successive accesses to the same chip-select (any access type) 0x0: No delay between the two accesses 0x1: Add CYCLE2CYCLEDELAY	RW	0
6	CYCLE2CYCLEDIFFCSEN	Add CYCLE2CYCLEDELAY between successive accesses to a different chip-select (any access type) 0x0: No delay between the two accesses 0x1: Add CYCLE2CYCLEDELAY	RW	0x0
5:4	RESERVED	Write 0s for future compatibility. Reads return 0.	RW	0x0

Bits	Field Name	Description	Type	Reset
3:0	BUSTURNAROUND	Bus turnaround latency between successive accesses to the same chip-select (read to write) or to a different chip-select (read to read and read to write) 0x0: 0 GPMC_FCLK cycle 0x1: 1 GPMC_FCLK cycle ... 0xF: 15 GPMC_FCLK cycles	RW	0x0

Table 10-447. Register Call Summary for Register GPMC_CONFIG6_i

GPMC Functional Description
- Wait Pin Monitoring Control: [0][1][2][3][4][5]
- Read Cycle Time and Write Cycle Time (RDCYCLETIME / WRCYCLETIME): [6][7]
- Access Time (RDACCESSTIME / WRACCESSTIME): [8]
- Access Time on Write Access: [9]
- Access on Address/Data Multiplexed Devices: [10][11][12]
- Access on Address/Address/Data-Multiplexed Devices: [13]
- Synchronous Single Write: [14]
- Synchronous Multiple (Burst) Write: [15][16][17]

GPMC Basic Programming Model
- GPMC Timing Parameters: [18][19][20][21][22][23]

GPMC Register Manual
- GPMC Register Summary: [24]

Table 10-448. GPMC_CONFIG7_i

Address Offset	0x0000 0078 + (0x0000 0030 * i)	**Index**	i = 0 to 7
Physical Address	0x5000 0078 + (0x0000 0030 * i)	**Instance**	GPMC
Description	CS address mapping configuration		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																				MASKADDRESS				RESERVED	CSVALID	BASEADDRESS					

Bits	Field Name	Description	Type	Reset
31:12	RESERVED	Write 0s for future compatibility. Read returns 0s.	RW	0x00000
11:8	MASKADDRESS	CS mask address. 0x0000: Chip-select size of 256 MiB 0x1000: Chip-select size of 128 MiB 0x1100: Chip-select size of 64 MiB 0x1110: Chip-select size of 32 MiB 0x1111: Chip-select size of 16 MiB Other values must be avoided as they create holes in the chip-select address space.	RW	0xF
7	RESERVED	Write 0s for future compatibility. Read returns 0.	RW	0x0
6	CSVALID	CS enable 0x0: CS disabled 0x1: CS enabled	RW	See (1)
5:0	BASEADDRESS	CSi base address where i = 0 to 7 (16-MiB minimum granularity) bits [5:0] corresponds to A29, A28, A27, A26, A25, and A24. See Figure 10-18	RW	0x00

(1) Reset value is 0x1 for CS0 and 0x0 for CS1 to CS7

Table 10-449. Register Call Summary for Register GPMC_CONFIG7_i

GPMC Functional Description • Chip-Select Base Address and Region Size: [0][1][2] • Chip-Select Configuration for NAND Interfacing in Byte or Word Stream Mode: [3]
GPMC Basic Programming Model • GPMC Configuration in NOR Mode: [4][5][6] • GPMC Configuration in NAND Mode: [7][8][9] • GPMC Timing Parameters: [10]
GPMC Register Manual • GPMC Register Summary: [11]

Table 10-450. GPMC_NAND_COMMAND_i

Address Offset	0x0000 007C + (0x0000 0030 * i)	**Index**	i = 0 to 7
Physical Address	0x5000 007C + (0x0000 0030 * i)	**Instance**	GPMC
Description	This register is not a true register, only an address location.		
Type	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPMC_NAND_COMMAND																															

Bits	Field Name	Description	Type	Reset
31:0	GPMC_NAND_COMMAND	This register is not a true register, only an address location. Writing data at the GPMC_NAND_COMMAND_i location places the data as the NAND command value on the bus, using a regular asynchronous write access.	W	0x-

Table 10-451. Register Call Summary for Register GPMC_NAND_COMMAND_i

GPMC Functional Description • NAND Device Command and Address Phase Control: [0][1][2][3][4] • Command Latch Cycle: [5] • General Facts About the Engine Configuration: [6]
GPMC Register Manual • GPMC Register Summary: [7] • GPMC Register Descriptions: [8]

Table 10-452. GPMC_NAND_ADDRESS_i

Address Offset	0x0000 0080 + (0x0000 0030 * i)	**Index**	i = 0 to 7
Physical Address	0x5000 0080 + (0x0000 0030 * i)	**Instance**	GPMC
Description	This register is not a true register, only an address location.		
Type	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPMC_NAND_ADDRESS																															

Bits	Field Name	Description	Type	Reset
31:0	GPMC_NAND_ADDRESS	This register is not a true register, only an address location. Writing data at the GPMC_NAND_ADDRESS_i location places the data as the NAND partial address value on the bus, using a regular asynchronous write access.	W	0x-

Table 10-453. Register Call Summary for Register GPMC_NAND_ADDRESS_i

GPMC Functional Description
• NAND Device Command and Address Phase Control: [0][1][2][3][4]
• Address Latch Cycle: [5]
• General Facts About the Engine Configuration: [6]
GPMC Register Manual
• GPMC Register Summary: [7]
• GPMC Register Descriptions: [8]

Table 10-454. GPMC_NAND_DATA_i

Address Offset	0x0000 0084 + (0x0000 0030 * i)	**Index**	i = 0 to 7
Physical Address	0x5000 0084 + (0x0000 0030 * i)	**Instance**	GPMC
Description	This register is not a true register,only an address location.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GPMC_NAND_DATA																															

Bits	Field Name	Description	Type	Reset
31:0	GPMC_NAND_DATA	This register is not a true register, only an address location. Reading data from the GPMC_NAND_DATA_i location or from any location in the associated chip-select memory region activates an asynchronous read access.	W	0x-

Table 10-455. Register Call Summary for Register GPMC_NAND_DATA_i

GPMC Functional Description
• NAND Device Data Read and Write Phase Control in Stream Mode: [0][1][2][3]
• General Facts About the Engine Configuration: [4]
• Optimizing NAND Access Using the Prefetch and Write-Posting Engine: [5]
GPMC Register Manual
• GPMC Register Summary: [6]
• GPMC Register Descriptions: [7][8][9]

Table 10-456. GPMC_PREFETCH_CONFIG1

Address Offset	0x0000 01E0		
Physical Address	0x5000 01E0	**Instance**	GPMC
Description	Prefetch engine configuration 1		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	CYCLEOPTIMIZATION			ENABLEOPTIMIZEDACCESS	ENGINECSSELECTOR			PFPWENROUNDROBIN	RESERVED			PFPWWEIGHTEDPRIO				RESERVED	FIFOTHRESHOLD							ENABLEENGINE	RESERVED	WAITPINSELECTOR		SYNCHROMODE	DMAMODE	RESERVED	ACCESSMODE

Bits	Field Name	Description	Type	Reset
31	RESERVED	Write 0s for future compatibility. Read returns 0.	RW	0x0
30:28	CYCLEOPTIMIZATION	Define the number of GPMC_FCLK cycles to be subtracted from RDCYCLETIME, WRCYCLETIME, RDACCESSTIME, CSRDOFFTIME, CSWROFFTIME, ADVRDOFFTIME, ADVWROFFTIME, OEOFFTIME, WEOFFTIME 0x0: 0 GPMC_FCLK cycle 0x1: 1 GPMC_FCLK cycle ... 0x7: 7 GPMC_FCLK cycles	RW	0x0
27	ENABLEOPTIMIZEDACCESS	Enables access cycle optimization 0x0: Access cycle optimization is disabled. 0x1: Access cycle optimization is enabled.	RW	0x0
26:24	ENGINECSSELECTOR	Selects the chip-select where Prefetch Postwrite engine is active 0x0: CS0 0x1: CS1 0x2: CS2 0x3: CS3 0x4: CS4 0x5: CS5 0x6: CS6 0x7: CS7	RW	0x0
23	PFPWENROUNDROBIN	Enables the PFPW RoundRobin arbitration 0x0: Prefetch Postwrite engine round robin arbitration is disabled. 0x1: Prefetch Postwrite engine round robin arbitration is enabled.	RW	0x0
22:20	RESERVED	Write 0s for future compatibility. Read returns 0s.	RW	0x0
19:16	PFPWWEIGHTEDPRIO	When an arbitration occurs between a DMA and a PFPW engine access, the DMA is always serviced. If the PFPWEnRoundRobin is enabled, 0x0: The next access is granted to the PFPW engine. 0x1: The next two accesses are granted to the PFPW engine. ..., 0xF: The next 16 accesses are granted to the PFPW engine.	RW	0x0
15	RESERVED	Write 0s for future compatibility. Read returns 0.	RW	0x0
14:8	FIFOTHRESHOLD	Selects the maximum number of bytes read from the FIFO or written to the FIFO by the host on a DMA or interrupt request 0x00: 0 byte 0x01: 1 byte ... 0x40: 64 bytes	RW	0x40
7	ENABLEENGINE	Enables the Prefetch Postwite engine 0x0: Prefetch Postwrite engine is disabled. 0x1: Prefetch Postwrite engine is enabled.	RW	0x0
6	RESERVED	Write 0s for future compatibility. Read returns 0.	RW	0x0
5:4	WAITPINSELECTOR	Select which wait pin edge detector should start the engine in synchronized mode 0x0: Selects Wait0 EdgeDetection 0x1: Selects Wait1 EdgeDetection 0x2, 0x3: Reserved	RW	0x0
3	SYNCHROMODE	Selects when the engine starts the access to chip-select 0x0: Engine starts the access to chip-select as soon as STARTENGINE is set 0x1: Engine starts the access to chip-select as soon as STARTENGINE is set AND wait to nonwait edge detection on the selected wait pin	RW	0x0

Bits	Field Name	Description	Type	Reset
2	DMAMODE	Selects interrupt synchronization or DMA request synchronization 0x0: Interrupt synchronization is enabled. Only interrupt line is activated on FIFO threshold crossing. 0x1: DMA request synchronization is enabled. A DMA request protocol is used.	RW	0x0
1	RESERVED	Write 0s for future compatibility. Read returns 0.	RW	0x0
0	ACCESSMODE	Selects prefetch read or write-posting accesses 0x0: Prefetch read mode 0x1: Write-posting mode	RW	0x0

Table 10-457. Register Call Summary for Register GPMC_PREFETCH_CONFIG1

GPMC Functional Description
- GPMC Interrupt Requests: [0]
- Prefetch and Write-Posting Engine: [1][2]
- General Facts About the Engine Configuration: [3][4]
- Prefetch Mode: [5][6][7][8][9][10][11][12][13][14][15][16]
- FIFO Control in Prefetch Mode: [17]
- Write-Posting Mode: [18][19][20][21][22][23][24][25][26]
- FIFO Control in Write-Posting Mode: [27]
- Optimizing NAND Access Using the Prefetch and Write-Posting Engine: [28][29][30]
- Interleaved Accesses Between Prefetch and Write-Posting Engine and Other Chip-Selects: [31][32]

GPMC Basic Programming Model
- GPMC Configuration in NAND Mode: [33][34][35][36][37][38][39][40][41][42]

GPMC Register Manual
- GPMC Register Summary: [43]

Table 10-458. GPMC_PREFETCH_CONFIG2

Address Offset	0x0000 01E4		
Physical Address	0x5000 01E4	**Instance**	GPMC
Description	Prefetch engine configuration 2		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																TRANSFERCOUNT															

Bits	Field Name	Description	Type	Reset
31:14	RESERVED	Write 0s for future compatibility. Read returns 0s.	RW	0x00000
13:0	TRANSFERCOUNT	Selects the number of bytes to be read or written by the engine to the selected chip-select 0x0000: 0 byte 0x0001: 1 byte ... 0x2000: 8 Kbytes	RW	0x0000

Table 10-459. Register Call Summary for Register GPMC_PREFETCH_CONFIG2

GPMC Functional Description
- Prefetch Mode: [0][1]
- Write-Posting Mode: [2][3]

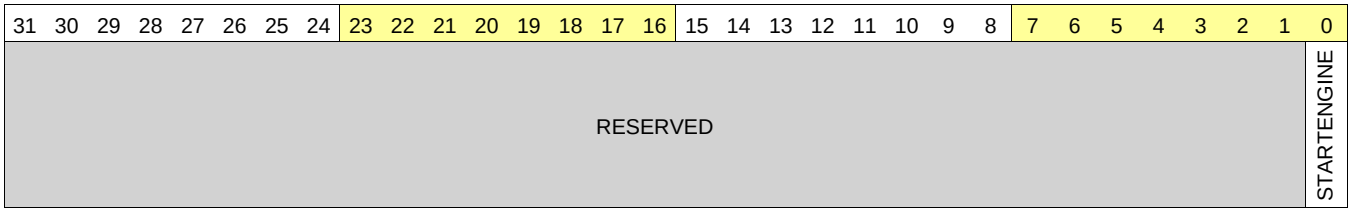
GPMC Basic Programming Model
- GPMC Configuration in NAND Mode: [4]

Table 10-459. Register Call Summary for Register GPMC_PREFETCH_CONFIG2 (continued)

GPMC Register Manual • GPMC Register Summary: [5]

Table 10-460. GPMC_PREFETCH_CONTROL

Address Offset	0x0000 01EC		
Physical Address	0x5000 01EC	**Instance**	GPMC
Description	Prefetch engine control		
Type	RW		



Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Write 0s for future compatibility. Read returns 0s.	RW	0x00000000
0	STARTENGINE	Resets the FIFO pointer and starts the engine Read 0x0: Engine is stopped. Write 0x0: Stops the engine Read 0x1: Engine is running. Write 0x1: Resets the FIFO pointer to 0x0 in prefetch mode and 0x40 in postwrite mode and starts the engine	RW	0x0

Table 10-461. Register Call Summary for Register GPMC_PREFETCH_CONTROL

GPMC Functional Description • General Facts About the Engine Configuration: [0] • Prefetch Mode: [1][2][3] • Write-Posting Mode: [4][5][6]
GPMC Basic Programming Model • GPMC Configuration in NAND Mode: [7][8]
GPMC Register Manual • GPMC Register Summary: [9]

Table 10-462. GPMC_PREFETCH_STATUS

Address Offset	0x0000 01F0		
Physical Address	0x5000 01F0	**Instance**	GPMC
Description	Prefetch engine status		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								FIFOPOINTER								RESERVED								FIFOTHRESHOLDSTATUS							
																								COUNTVALUE							

Bits	Field Name	Description	Type	Reset
31	RESERVED	Write 0s for future compatibility. Read returns 0.	RW	0x0
30:24	FIFOPOINTER	Number of available bytes to be read or number of free empty byte places to be written 0x00: 0 byte available to be read or 0 free empty place to be written ... 0x40: 64 bytes available to be read or 64 empty places to be written	R	0x00
23:17	RESERVED	Write 0s for future compatibility. Read returns 0s.	RW	0x00
16	FIFOTHRESHOLDSTATUS	Set when FIFOPointer exceeds FIFOThreshold value 0x0: FIFOPointer smaller or equal to FIFOThreshold. Writing to this bit has no effect. 0x1: FIFOPointer greater than FIFOThreshold. Writing to this bit has no effect.	R	0x0
15:14	RESERVED	Write 0s for future compatibility. Read returns 0s.	RW	0x0
13:0	COUNTVALUE	Number of remaining bytes to be read or to be written by the engine according to the TransferCount value 0x0000: 0 byte remaining to be read or to be written 0x0001: 1 byte remaining to be read or to be written ... 0x2000: 8 KiB remaining to be read or to be written	R	0x0000

Table 10-463. Register Call Summary for Register GPMC_PREFETCH_STATUS

GPMC Functional Description

- General Facts About the Engine Configuration: [0][1]
- FIFO Control in Prefetch Mode: [2][3][4]
- FIFO Control in Write-Posting Mode: [5][6]

GPMC Register Manual

- GPMC Register Summary: [7]
- GPMC Register Descriptions: [8][9]

Table 10-464. GPMC_ECC_CONFIG

Address Offset	0x0000 01F4		
Physical Address	0x5000 01F4	**Instance**	GPMC
Description	ECC configuration		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED															ECCALGORITHM	RESERVED		ECCBCHTSEL		ECCWRAPMODE				ECC16B	ECCTOPSECTOR			ECCCS			ECCENABLE

Bits	Field Name	Description	Type	Reset
31:17	RESERVED	Write 0s for future compatibility. Read returns 0s.	RW	0x0000
16	ECCALGORITHM	ECC algorithm used 0x0: Hamming code 0x1: BCH code	RW	0x0
15:14	RESERVED	Write 0s for future compatibility. Read returns 0s.	RW	0x0
13:12	ECCBCHTSEL	Error correction capability used for BCH 0x0: Up to 4 bits error correction (t = 4) 0x1: Up to 8 bits error correction (t = 8) 0x2: Up to 16 bits error correction (t = 16) 0x3: Reserved	RW	0x1
11:8	ECCWRAPMODE	Spare area organization definition for the BCH algorithm. See the BCH syndrome/parity calculator module functional specification for more details	RW	0x0
7	ECC16B	Selects an ECC calculated on 16 columns 0x0: ECC calculated on 8 columns 0x1: ECC calculated on 16 columns	RW	0x0
6:4	ECCTOPSECTOR	Number of sectors to process with the BCH algorithm 0x0: 1 sector (512-kB page) 0x1: 2 sectors ... 0x3: 4 sectors (2-kB page) ... 0x7: 8 sectors (4-kB page)	RW	0x3
3:1	ECCCS	Selects the CS where ECC is computed 0x0: CS0 0x1: CS1 0x2: CS2 0x3: CS3 Other: Reserved	RW	0x0
0	ECCENABLE	Enables the ECC feature 0x0: ECC disabled 0x1: ECC enabled	RW	0x0

Table 10-465. Register Call Summary for Register GPMC_ECC_CONFIG

GPMC Functional Description

- ECC Calculator: [0][1]
- Hamming Code: [2][3][4][5][6][7]

Table 10-465. Register Call Summary for Register GPMC_ECC_CONFIG (continued)

GPMC Basic Programming Model • GPMC Configuration in NAND Mode: [8][9][10][11][12][13]
GPMC Register Manual • GPMC Register Summary: [14] • GPMC Register Descriptions: [15][16][17]

Table 10-466. GPMC_ECC_CONTROL

Address Offset	0x0000 01F8		
Physical Address	0x5000 01F8	**Instance**	GPMC
Description	ECC control		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							ECCCLEAR	RESERVED				ECCPOINTER			

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Write 0s for future compatibility. Read returns 0s.	RW	0x000000
8	ECCCLEAR	Clear all ECC result registers Reads return 0. Write 0x1 to this field clears all ECC result registers. Write 0x0 is ignored.	RW	0x0
7:4	RESERVED	Write 0s for future compatibility. Read returns 0s.	RW	0x0
3:0	ECCPOINTER	Selects ECC result register (Reads to this field give the dynamic position of the ECC pointer - Writes to this field select the ECC result register where the first ECC computation will be stored.); Other enums: writing other values disables the ECC engine (ECCENABLE bit of GPMC_ECC_CONFIG set to 0) 0x0: Writing 0x0 disables the ECC engine (ECCENABLE bit of GPMC_ECC_CONFIG set to 0) 0x1: ECC result register 1 selected 0x2: ECC result register 2 selected 0x3: ECC result register 3 selected 0x4: ECC result register 4 selected 0x5: ECC result register 5 selected 0x6: ECC result register 6 selected 0x7: ECC result register 7 selected 0x8: ECC result register 8 selected 0x9: ECC result register 9 selected	RW	0x0

Table 10-467. Register Call Summary for Register GPMC_ECC_CONTROL

GPMC Functional Description • Hamming Code: [0][1][2][3]
GPMC Basic Programming Model • GPMC Configuration in NAND Mode: [4][5]
GPMC Register Manual • GPMC Register Summary: [6]

Table 10-468. GPMC_ECC_SIZE_CONFIG

Address Offset	0x0000 01FC		
Physical Address	0x5000 01FC	**Instance**	GPMC
Description	ECC size		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								ECCSIZE1								RESERVED				ECCSIZE0				RESERVED								ECC9RESULTSIZE	ECC8RESULTSIZE	ECC7RESULTSIZE	ECC6RESULTSIZE	ECC5RESULTSIZE	ECC4RESULTSIZE	ECC3RESULTSIZE	ECC2RESULTSIZE	ECC1RESULTSIZE

Bits	Field Name	Description	Type	Reset
31:30	RESERVED	Write 0s for future compatibility. Read returns 3.	RW	0x3
29:22	ECCSIZE1	Defines Hamming code ECC size 1 in bytes 0x00: 2 bytes 0x01: 4 bytes 0x02: 6 bytes 0x03: 8 bytes ... 0xFF: 512 bytes For BCH code ECC, the size 1 is programmed directly with the number of nibbles. For details, see Section 10.3.4.12.3.2.2.3, *Wrapping Modes*.	RW	0xFF
21:20	RESERVED	Write 0s for future compatibility. Read returns 3.	RW	0x3
19:12	ECCSIZE0	Defines Hamming code ECC size 0 in bytes 0x00: 2 bytes 0x01: 4 bytes 0x02: 6 bytes 0x03: 8 bytes ... 0xFF: 512 bytes For BCH code ECC, the size 0 is programmed directly with the number of nibbles. For details, see Section 10.3.4.12.3.2.2.3, *Wrapping Modes*.	RW	0xFF
11:9	RESERVED	Write 0s for future compatibility. Read returns 0s.	RW	0x0
8	ECC9RESULTSIZE	Selects ECC size for ECC 9 result register 0x0: ECCSIZE0 selected 0x1: ECCSIZE1 selected	RW	0x0
7	ECC8RESULTSIZE	Selects ECC size for ECC 8 result register 0x0: ECCSIZE0 selected 0x1: ECCSIZE1 selected	RW	0x0
6	ECC7RESULTSIZE	Selects ECC size for ECC 7 result register 0x0: ECCSIZE0 selected 0x1: ECCSIZE1 selected	RW	0x0
5	ECC6RESULTSIZE	Selects ECC size for ECC 6 result register 0x0: ECCSIZE0 selected 0x1: ECCSIZE1 selected	RW	0x0
4	ECC5RESULTSIZE	Selects ECC size for ECC 5 result register 0x0: ECCSIZE0 selected 0x1: ECCSIZE1 selected	RW	0x0

Bits	Field Name	Description	Type	Reset
3	ECC4RESULTSIZE	Selects ECC size for ECC 4 result register 0x0: ECCSIZE0 selected 0x1: ECCSIZE1 selected	RW	0x0
2	ECC3RESULTSIZE	Selects ECC size for ECC 3 result register 0x0: ECCSIZE0 selected 0x1: ECCSIZE1 selected	RW	0x0
1	ECC2RESULTSIZE	Selects ECC size for ECC 2 result register 0x0: ECCSIZE0 selected 0x1: ECCSIZE1 selected	RW	0x0
0	ECC1RESULTSIZE	Selects ECC size for ECC 1 result register 0x0: ECCSIZE0 selected 0x1: ECCSIZE1 selected	RW	0x0

Table 10-469. Register Call Summary for Register GPMC_ECC_SIZE_CONFIG

GPMC Functional Description
- Hamming Code: [0][1][2][3]
- BCH Code: [4][5][6]

GPMC Basic Programming Model
- GPMC Configuration in NAND Mode: [7][8]

GPMC Register Manual
- GPMC Register Summary: [9]

Table 10-470. GPMC_ECCj_RESULT

Address Offset	0x0000 0200 + (0x0000 0004 * (j - 1))	**Index**	j = 1 to 9
Physical Address	0x5000 0200 + (0x0000 0004 * j)	**Instance**	GPMC
Description	ECC result register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				P2048O	P1024O	P512O	P256O	P128O	P64O	P32O	P16O	P8O	P4O	P2O	P1O	RESERVED				P2048E	P1024E	P512E	P256E	P128E	P64E	P32E	P16E	P8E	P4E	P2E	P1E

Bits	Field Name	Description	Type	Reset
31:28	RESERVED	Write 0s for future compatibility. Read returns 0s.	RW	0x0
27	P2048O	Odd row parity bit 2048, only used for ECC computed on 512 bytes	R	0x0
26	P1024O	Odd row parity bit 1024	R	0x0
25	P512O	Odd row parity bit 512	R	0x0
24	P256O	Odd row parity bit 256	R	0x0
23	P128O	Odd row parity bit 128	R	0x0
22	P64O	Odd row parity bit 64	R	0x0
21	P32O	Odd row parity bit 32	R	0x0
20	P16O	Odd row parity bit 16	R	0x0
19	P8O	Odd row parity bit 8	R	0x0
18	P4O	Odd Column Parity bit 4	R	0x0
17	P2O	Odd Column Parity bit 2	R	0x0
16	P1O	Odd Column Parity bit 1	R	0x0
15:12	RESERVED	Write 0s for future compatibility. Read returns 0s.	RW	0x0

Bits	Field Name	Description	Type	Reset
11	P2048E	Even row parity bit 2048, only used for ECC computed on 512 bytes	R	0x0
10	P1024E	Even row parity bit 1024	R	0x0
9	P512E	Even row parity bit 512	R	0x0
8	P256E	Even row parity bit 256	R	0x0
7	P128E	Even row parity bit 128	R	0x0
6	P64E	Even row parity bit 64	R	0x0
5	P32E	Even row parity bit 32	R	0x0
4	P16E	Even row parity bit 16	R	0x0
3	P8E	Even row parity bit 8	R	0x0
2	P4E	Even column parity bit 4	R	0x0
1	P2E	Even column parity bit 2	R	0x0
0	P1E	Even column parity bit 1	R	0x0

Table 10-471. Register Call Summary for Register GPMC_ECCj_RESULT

GPMC Functional Description
- Hamming Code: [0][1][2][3][4][5][6][7]

GPMC Register Manual
- GPMC Register Summary: [8]

Table 10-472. GPMC_BCH_RESULT0_i

Address Offset	0x0000 0240 + (0x0000 0010 * i)	**Index**	i = 0 to 7
Physical Address	0x5000 0240 + (0x0000 0010 * i)	**Instance**	GPMC
Description	BCH ECC result (bits 0 to 31)		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BCH_RESULT_0																															

Bits	Field Name	Description	Type	Reset
31:0	BCH_RESULT_0	BCH ECC result (bits 0 to 31)	RW	0x00000000

Table 10-473. Register Call Summary for Register GPMC_BCH_RESULT0_i

GPMC Functional Description
- BCH Code: [0]

GPMC Register Manual
- GPMC Register Summary: [1]

Table 10-474. GPMC_BCH_RESULT1_i

Address Offset	0x0000 0244 + (0x0000 0010 * i)	**Index**	i = 0 to 7
Physical Address	0x5000 0244 + (0x0000 0010 * i)	**Instance**	GPMC
Description	BCH ECC result (bits 32 to 63)		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BCH_RESULT_1																															

Bits	Field Name	Description	Type	Reset
31:0	BCH_RESULT_1	BCH ECC result (bits 32 to 63)	RW	0x00000000

Table 10-475. Register Call Summary for Register GPMC_BCH_RESULT1_i

GPMC Functional Description • BCH Code: [0]
GPMC Register Manual • GPMC Register Summary: [1]

Table 10-476. GPMC_BCH_RESULT2_i

Address Offset	0x0000 0248 + (0x0000 0010 * i)	**Index**	i = 0 to 7
Physical Address	0x5000 0248 + (0x0000 0010 * i)	**Instance**	GPMC
Description	BCH ECC result (bits 64 to 95)		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BCH_RESULT_2																															

Bits	Field Name	Description	Type	Reset
31:0	BCH_RESULT_2	BCH ECC result (bits 64 to 95)	RW	0x00000000

Table 10-477. Register Call Summary for Register GPMC_BCH_RESULT2_i

GPMC Functional Description • BCH Code: [0]
GPMC Register Manual • GPMC Register Summary: [1]

Table 10-478. GPMC_BCH_RESULT3_i

Address Offset	0x0000 024C + (0x0000 0010 * i)	**Index**	i = 0 to 7
Physical Address	0x5000 024C + (0x0000 0010 * i)	**Instance**	GPMC
Description	BCH ECC result (bits 96 to 127)		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BCH_RESULT_3																															

Bits	Field Name	Description	Type	Reset
31:0	BCH_RESULT_3	BCH ECC result (bits 96 to 127)	RW	0x00000000

Table 10-479. Register Call Summary for Register GPMC_BCH_RESULT3_i

GPMC Functional Description • BCH Code: [0]
GPMC Register Manual • GPMC Register Summary: [1]

Table 10-480. GPMC_BCH_RESULT4_i

Address Offset	0x0000 0300 + (0x0000 0010 * i)	**Index**	i = 0 to 7
Physical Address	0x5000 0300 + (0x0000 0010 * i)	**Instance**	GPMC
Description	BCH ECC result (bits 128 to 159)		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BCH_RESULT_4																															

Bits	Field Name	Description	Type	Reset
31:0	BCH_RESULT_4	BCH ECC result (bits 128 to 159)	RW	0x00000000

Table 10-481. Register Call Summary for Register GPMC_BCH_RESULT4_i

GPMC Register Manual
- GPMC Register Summary: [0]

Table 10-482. GPMC_BCH_RESULT5_i

Address Offset	0x0000 0304 + (0x0000 0010 * i)	**Index**	i = 0 to 7
Physical Address	0x5000 0304 + (0x0000 0010 * i)	**Instance**	GPMC
Description	BCH ECC result (bits 160 to 191)		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BCH_RESULT_5																															

Bits	Field Name	Description	Type	Reset
31:0	BCH_RESULT_5	BCH ECC result (bits 160 to 191)	RW	0x00000000

Table 10-483. Register Call Summary for Register GPMC_BCH_RESULT5_i

GPMC Register Manual
- GPMC Register Summary: [0]

Table 10-484. GPMC_BCH_RESULT6_i

Address Offset	0x0000 0308 + (0x0000 0010 * i)	**Index**	i = 0 to 7
Physical Address	0x5000 0308 + (0x0000 0010 * i)	**Instance**	GPMC
Description	BCH ECC result (bits 192 to 207)		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																BCH_RESULT_6															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Write 0s for future compatibility. Read returns 0s.	R	0x0000
15:0	BCH_RESULT_6	BCH ECC result (bits 192 to 207)	RW	0x0000

Table 10-485. Register Call Summary for Register GPMC_BCH_RESULT6_i

GPMC Register Manual
- GPMC Register Summary: [0]

Table 10-486. GPMC_BCH_SWDATA

Address Offset	0x0000 02D0		
Physical Address	0x5000 02D0	**Instance**	GPMC
Description	This register is used to directly pass data to the BCH ECC calculator without accessing the actual NAND flash interface.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																BCH_DATA															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Write 0s for future compatibility. Read returns 0s.	R	0x0000
15:0	BCH_DATA	Data to be included in the BCH calculation Only bits 0 to 7 are considered if the calculator is configured to use 8-bit data (GPMC_ECC_CONFIG[7] ECC16B = 0)	RW	0x0000

Table 10-487. Register Call Summary for Register GPMC_BCH_SWDATA

GPMC Register Manual
- GPMC Register Summary: [0]

10.4 Error Location Module

10.4.1 Error Location Module Overview

When reading from NAND flash memories, some level of error-correction is required. In the case of NAND modules with no internal correction capability, sometimes referred to as *bare NANDs*, the correction process is delegated to the memory controller.

The general-purpose memory controller (GPMC) probes data read from an external NAND flash and uses this to compute checksum-like information, called syndrome polynomials, on a per-block basis. Each syndrome polynomial gives a status of the read operations for a full block, including 512 bytes of data, parity bits, and an optional spare-area data field, with a maximum block size of 1023 bytes. Computation is based on a Bose-Chaudhuri-Hocquenghem (BCH) algorithm. The error-location module (ELM) extracts error addresses from these syndrome polynomials.

Based on the syndrome polynomial value, the ELM can detect errors, compute the number of errors, and give the location of each error bit. The actual data is not required to complete the error-correction algorithm. Errors can be reported anywhere in the NAND flash block, including in the parity bits.

The maximum acceptable number of errors that can be corrected depends on a programmable configuration parameter. 4-, 8-, and 16-bit error-correction levels are supported. The ELM relies on a static and fixed definition of the generator polynomial for each error-correction level that corresponds to the generator polynomials defined in the GPMC (there are three fixed polynomial for the three correction error levels). A larger number of errors than the programmed error-correction level may be detected, but the ELM cannot correct them all. The offending block is then tagged as *uncorrectable* in the associated computation exit status register. If the computation is successful, that is, if the number of errors detected does not exceed the maximum value authorized for the chosen correction capability, the exit status register contains the information on the number of detected errors.

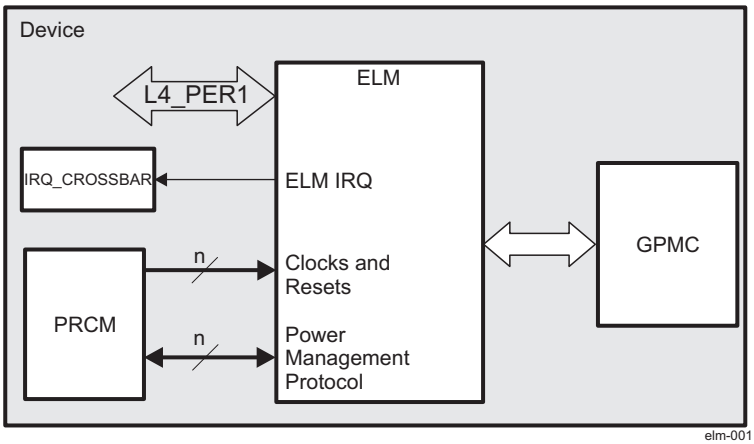
When the error-location process completes, an interrupt is triggered to inform the software that its status can be checked. The number of detected errors and their locations in the NAND block can be retrieved from the module through register accesses.

The ELM has the following features:

- 4, 8 and 16 bits per 512-byte block error-location based on BCH algorithms
- Eight simultaneous processing contexts
- Page-based and continuous modes
- Interrupt generation on error-location process completion:
  - When the full page has been processed in page mode
  - For each syndrome polynomial in continuous mode

Figure 10-63 shows the ELM overview.

Figure 10-63. ELM Overview



10.4.2 ELM Integration

The ELM extracts error addresses from generated syndrome polynomials.

The ELM is used with the GPMC. Syndrome polynomials generated on-the-fly when reading a NAND flash page and stored in GPMC registers are passed to the ELM. A host processor can then correct the data block by flipping the bits to which the ELM error-location outputs point.

Figure 10-64 shows the integration of the ELM subsystem in the device.

Figure 10-64. ELM Integration

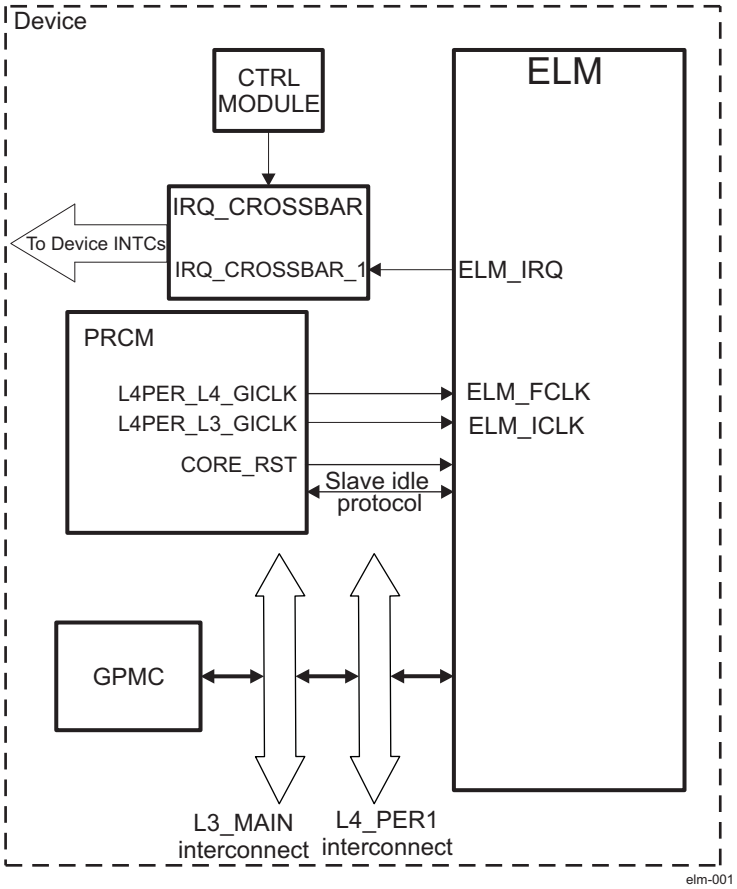


Table 10-488 through Table 10-490 summarize the integration of the module in the device.

Table 10-488. ELM Integration Attributes

Module Instance	Attributes		
	Power Domain	Wake-Up Capability	Interconnect
ELM	PD_COREAON	No	L4_PER1

Table 10-489. ELM Clocks and Resets

Clocks				
Module Instance	**Destination Signal Name**	**Source Signal Name**	**Source**	**Description**
ELM	ELM_FCLK	L4PER_L4_GICLK	PRCM	Functional clock
	ELM_ICLK	L4PER_L3_GICLK	PRCM	Interface clock
Resets				
Module Instance	**Destination Signal Name**	**Source Signal Name**	**Source**	**Description**
ELM	ELM_RST	L4PER_RST	PRCM	Module hardware reset

Table 10-490. ELM Hardware Requests

Interrupt Requests				
Module Instance	**Source Signal Name**	**IRQ_CROSSBAR**	**Default Mapping**	**Description**
ELM	ELM_IRQ	IRQ_CROSSBAR_1	DSP1_IRQ_32 DSP2_IRQ_32 EVE_IRQ_0	BCH error-location module interrupt

NOTE: The "**Default Mapping**" column in Table 10-490 *ELM Hardware Requests* shows the default mapping of module IRQ source signals. These IRQ source signals can also be mapped to other lines of each device Interrupt controller through the IRQ_CROSSBAR module. For more information about the IRQ_CROSSBAR module, see Section 13.4.6.4, *IRQ_CROSSBAR Module Functional Description*, in Chapter 13, *Control Module*. For more information about the device interrupt controllers, see Chapter 12, *Interrupt Controllers*.

10.4.3 ELM Functional Description

The ELM is designed around the error-location engine, which handles the computation based on the input syndrome polynomials.

The ELM maps the error-location engine to a standard interconnect interface by using a set of registers to control inputs and outputs.

10.4.3.1 ELM Software Reset

To perform a software reset, set the ELM_SYSCONFIG[1] SOFTRESET bit to 1. The ELM_SYSSTATUS[0] RESETDONE bit indicates that the software reset is complete when its value is 1. When the software reset completes, the ELM_SYSCONFIG[1] SOFTRESET bit is automatically reset.

10.4.3.2 ELM Power Management

Table 10-491 describes the power-management features available to the ELM.

NOTE:

- For information about source clock gating and a description of the sleep/wake-up transitions, see the Section 3.6.4.5, *CD_L4_PER1 Clock Domain*, in Chapter 3, *Power, Reset, and Clock Management*.
- For a general description of EnaWakeUp, IdleMode, ClockActivity, and StandbyMode features, see Section 3.6, *Clock Management Functional Description*, in Chapter 3, *Power, Reset, and Clock Management*

Table 10-491. Local Power-Management Features

Feature	Registers	Description
Clock autogating	ELM_SYSCONFIG[0] AUTOGATING	This bit allows a local power optimization inside the module by gating the ELM_FCLK clock upon the interface activity.
Slave idle modes	ELM_SYSCONFIG[4:3] SIDLEMODE	Force-idle, no-idle, and smart-idle modes are available.
Clock activity	ELM_SYSCONFIG[8] CLOCKACTIVITY	The clock can be switched off or maintained during the wake-up period.
Master standby modes	N/A	
Global wake-up enable	N/A	
Wake-up sources enable	N/A	

CAUTION

The PRCM module has no hardware means of reading CLOCKACTIVITY settings. Thus, software must ensure consistent programming between the ELM CLOCKACTIVITY and ELM clock PRCM control bits. For a description of the ClockActivity feature, see Chapter 3, *Power, Reset, and Clock Management*.

10.4.3.3 ELM Interrupt Requests

Table 10-492 lists the event flags, and their masks, that can cause module interrupts.

Table 10-492. Events

Event Flag	Event Mask	Description
ELM_IRQSTATUS[8] PAGE_VALID	ELM_IRQENABLE[8] PAGE_MASK	Page interrupt
ELM_IRQSTATUS[7] LOC_VALID_7	ELM_IRQENABLE[7] LOCATION_MASK_7	Error-location interrupt for syndrome polynomial 7
ELM_IRQSTATUS[6] LOC_VALID_6	ELM_IRQENABLE[6] LOCATION_MASK_6	Error-location interrupt for syndrome polynomial 6
ELM_IRQSTATUS[5] LOC_VALID_5	ELM_IRQENABLE[5] LOCATION_MASK_5	Error-location interrupt for syndrome polynomial 5
ELM_IRQSTATUS[4] LOC_VALID_4	ELM_IRQENABLE[4] LOCATION_MASK_4	Error-location interrupt for syndrome polynomial 4
ELM_IRQSTATUS[3] LOC_VALID_3	ELM_IRQENABLE[3] LOCATION_MASK_3	Error-location interrupt for syndrome polynomial 3
ELM_IRQSTATUS[2] LOC_VALID_2	ELM_IRQENABLE[2] LOCATION_MASK_2	Error-location interrupt for syndrome polynomial 2
ELM_IRQSTATUS[1] LOC_VALID_1	ELM_IRQENABLE[1] LOCATION_MASK_1	Error-location interrupt for syndrome polynomial 1
ELM_IRQSTATUS[0] LOC_VALID_0	ELM_IRQENABLE[0] LOCATION_MASK_0	Error-location interrupt for syndrome polynomial 0

10.4.3.4 Processing Initialization

ELM_LOCATION_CONFIG global setting parameters must be set before using the error-location engine. The ELM_LOCATION_CONFIG[1:0] ECC_BCH_LEVEL bit field defines the error-correction level used (4-,8-, or 16-bit error correction). The ELM_LOCATION_CONFIG[26:16] ECC_SIZE bit field defines the maximum buffer length beyond which the engine processing no longer looks for errors.

Software can choose to use the ELM in continuous mode or page mode. If all ELM_PAGE_CTRL[i] SECTOR_i bits (i is the syndrome polynomial number, where i = 0 to 7) are reset, continuous mode is used. In any other case, page mode is implicitly selected.

- Continuous mode: Each syndrome polynomial is processed independently. Results for a syndrome can be retrieved and acknowledged at any time, regardless of the status of the other seven processing contexts.
- Page mode: Syndrome polynomials are grouped into atomic entities: only one page can be processed at any given time, even if all eight contexts are not used for this page. Unused contexts are lost and cannot be affected to any other processing. The full page must be acknowledged and cleared before moving to the next page.

For completion interrupts to be generated correctly, all ELM_IRQENABLE[i] LOCATION_MASK_i bits (where i = 0 to 7) must be forced to 0 when in page mode, and set to 1 in continuous mode. Additionally, the ELM_IRQENABLE[8] PAGE_MASK bit must be set to 1 when in page mode.

Software initiates error-location processing by writing a syndrome polynomial into one of the eight possible register sets. Each of these register sets includes seven registers: ELM_SYNDROME_FRAGMENT_0_i to ELM_SYNDROME_FRAGMENT_6_i. The first six registers can be written in any order, but ELM_SYNDROME_FRAGMENT_6_i must be written last because it includes the validity bit, which instructs the ELM that this syndrome polynomial must be processed (the ELM_SYNDROME_FRAGMENT_6_i[16] SYNDROME_VALID bit).

As soon as one validity bit is asserted (ELM_SYNDROME_FRAGMENT_6_i[16] SYNDROME_VALID = 0x1, where i = 0 to 7), error-location processing can start for the corresponding syndrome polynomial. The associated ELM_LOCATION_STATUS_i and ELM_ERROR_LOCATION_0_i to ELM_ERROR_LOCATION_15_i registers (where i = 0 to 7) are not reset. Software must not consider them until the corresponding ELM_IRQSTATUS[i] LOC_VALID_i bit is set.

10.4.3.5 Processing Sequence

While the error-location engine is busy processing one syndrome polynomial, further syndrome polynomials can be written. They are processed when the current processing completes.

The engine completes early when:

- No error is detected; that is, when the ELM_LOCATION_STATUS_i[8] ECC_CORRECTABLE bit is set to 1 and the ELM_LOCATION_STATUS_i[4:0] ECC_NB_ERRORS bit field is set to 0x0.
- Too many errors are detected; that is, when the ELM_LOCATION_STATUS_i[8] ECC_CORRECTABLE bit is set to 0 while the ELM_LOCATION_STATUS_i[4:0] ECC_NB_ERRORS bit field is set with the value output by the error-location engine. The reported number of errors is not ensured if ECC_CORRECTABLE is 0.

If the engine completes early, the associated error-location registers ELM_ERROR_LOCATION_0_i to ELM_ERROR_LOCATION_15_i (where i = 0 to 7) are not updated.

In all other cases, the engine goes through the entire error-location process. Each time an error location is found, it is logged in the associated ECC_ERROR_LOCATION bit field. The first error detected is logged in the ELM_ERROR_LOCATION_0_i[12:0] ECC_ERROR_LOCATION bit field; the second is logged in the ELM_ERROR_LOCATION_1_i[12:0] ECC_ERROR_LOCATION bit field, and so on.

Table 10-493 describes the ELM_LOCATION_STATUS_i value decoding.

Table 10-493. ELM_LOCATION_STATUS_i Value Decoding

ECC_CORRECTABLE Value	ECC_NB_ERRORS Value	Status	Number of Errors Detected	Action Required
1	0	OK	0	None
1	≠ 0	OK	ECC_NB_ERRORS	Correct the data buffer read based on the ELM_ERROR_LOCATION_0_i to ELM_ERROR_LOCATION_15_i results.
0	Any	Failed	Unknown	Software-dependent

10.4.3.6 Processing Completion

When the processing for a given syndrome polynomial completes, its ELM_SYNDROME_FRAGMENT_6_i[16] SYNDROME_VALID bit is reset. It must not be set again until the exit status registers, ELM_LOCATION_STATUS_i (where i = 0 to 7) for this processing are checked. Failure to comply with this rule leads to potential loss of the first polynomial process data output.

The error-location engine signals the process completion to the ELM. When this event is detected, the corresponding ELM_IRQSTATUS[i] LOC_VALID_i bit (where i = 0 to 7) is set. The processing exit status is available from the associated ELM_LOCATION_STATUS_i register, and error locations are stored in order in the ECC_ERROR_LOCATION bit fields. Software must read only valid error-location registers based on the number of errors detected and located.

Immediately after the error-location engine completes, a new syndrome polynomial can be processed, if any is available, as reported by the ELM_SYNDROME_FRAGMENT_6_i[16] SYNDROME_VALID bit, depending on the configured error-correction level. If several syndrome polynomials are available, a round-robin arbitration is used to select one for processing.

In continuous mode (that is, all bits in ELM_PAGE_CTRL are reset), an interrupt is triggered whenever a ELM_IRQSTATUS[i] LOC_VALID_i bit is asserted. Software must read the ELM_IRQSTATUS register to determine which polynomial is processed and retrieve the exit status and error locations (ELM_LOCATION_STATUS_i and ELM_ERROR_LOCATION_0_i to ELM_ERROR_LOCATION_15_i). When done, software must clear the corresponding ELM_IRQSTATUS[i] LOC_VALID_i bit by setting it to 1. Other status bits must be set to 0 so that other interrupts are not unintentionally cleared. When using this mode, the ELM_IRQSTATUS[8] PAGE_VALID interrupt is never triggered.

In page mode, the module does not trigger interrupts for the processing completion of each polynomial because the ELM_IRQENABLE[i] LOCATION_MASK_i bits are cleared. A page is defined using the ELM_PAGE_CTRL register. Each SECTOR_i bit set means the corresponding polynomial i is part of the page processing. A page is fully processed when all tagged polynomials have been processed, as logged in the ELM_IRQSTATUS[i] LOC_VALID_i bits. The module triggers an ELM_IRQSTATUS[8] PAGE_VALID interrupt whenever it detects that the full page has been processed. To make sure the next page can be correctly processed, all status bits in the ELM_IRQSTATUS register must be cleared by using a single atomic-write access.

NOTE: Do not modify page setting parameters in the ELM_PAGE_CTRL register unless the engine is idle, no polynomial input is valid, and all interrupts have been cleared.

Because no polynomial-level interrupt is triggered in page mode, polynomials cleared in the ELM_PAGE_CTRL[i] SECTOR_i bits (where i = 0 to 7) are processed as usual, but are essentially ignored. Software must manually poll the ELM_IRQSTATUS bits to check for their status.

10.4.4 ELM Basic Programming Model

10.4.4.1 ELM Low-Level Programming Model

10.4.4.1.1 Processing Initialization

Table 10-494. ELM Processing Initialization

Step	Register/Bit Field/Programming Model	Value
Resets the module	ELM_SYSCONFIG[1] SOFTRESET	0x1
Wait until reset is done.	ELM_SYSSTATUS[0] RESETDONE	0x1
Configure the slave interface power management.	ELM_SYSCONFIG[4:3] SIDLEMODE	Set value.
Defines the error-correction level used	ELM_LOCATION_CONFIG[1:0] ECC_BCH_LEVEL	Set value.
Defines the maximum buffer length	ELM_LOCATION_CONFIG[26:16] ECC_SIZE	Set value.
Sets the ELM in continuous mode or page mode	ELM_PAGE_CTRL	Set value.
If continuous mode is used	All ELM_PAGE_CTRL[i] SECTOR_i (where i = 0 to 7)	0x0
Enables interrupt for syndrome polynomial i	ELM_IRQENABLE[i] LOCATION_MASK_i	0x1

Table 10-494. ELM Processing Initialization (continued)

Step	Register/Bit Field/Programming Model	Value
else (page mode is used)	One syndrome polynomial i is set ELM_PAGE_CTRL[i] SECTOR_i (where i = 0 to 7)	0x1
Disable all interrupts for syndrome polynomial and enable PAGE_MASK interrupt.	All ELM_IRQENABLE[i] LOCATION_MASK_i = 0x0 and ELM_IRQENABLE[8] PAGE_MASK = 0x1	Set value.
endif		Set value.
Set the input syndrome polynomial i.	ELM_SYNDROME_FRAGMENT_0_i	Set value.
	ELM_SYNDROME_FRAGMENT_1_i	Set value.
	ELM_SYNDROME_FRAGMENT_5_i	Set value.
	ELM_SYNDROME_FRAGMENT_6_i	Set value.
Initiates the computation process	ELM_SYNDROME_FRAGMENT_6_i[16] SYNDROME_VALID	0x1

10.4.4.1.2 Read Results

The engine goes through the entire error-location process and results can be read. Table 10-495 and Table 10-496 describe the processing completion for continuous and page modes, respectively.

Table 10-495. ELM Processing Completion for Continuous Mode

Step	Register/Bit Field/Programming Model	Value
Wait until process is complete for syndrome polynomial i: Wait until the ELM_IRQ interrupt is generated, or poll the status register.		
Read for which i the error-location process is complete.	ELM_IRQSTATUS[i] LOC_VALID_i	0x1
if the process fails (too many errors)	ELM_LOCATION_STATUS_i[8] ECC_CORRECTABLE	0x0
It is software dependant.		
else (process successful, the engine completes)	ELM_LOCATION_STATUS_i[8] ECC_CORRECTABLE	0x1
Read the number of errors.	ELM_LOCATION_STATUS_i[4:0] ECC_NB_ERRORS	
Read the error-location bit addresses for syndrome polynomial i of the ECC_NB_ERRORS first registers. Software must correct errors in the data buffer.	ELM_ERROR_LOCATION_0_i[12:0] ECC_ERROR_LOCATION	
	ELM_ERROR_LOCATION_1_i[12:0] ECC_ERROR_LOCATION	
	...	
	ELM_ERROR_LOCATION_15_i[12:0] ECC_ERROR_LOCATION	
endif		
Clear the corresponding i interrupt.	ELM_IRQSTATUS[i] LOC_VALID_i	0x1

A new syndrome polynomial can be processed after the end of processing (ELM_SYNDROME_FRAGMENT_6_i[16] SYNDROME_VALID = 0x0) and after the exit status register check (ELM_LOCATION_STATUS_i).

Table 10-496. ELM Processing Completion for Page Mode

Step	Register/Bit Field/Programming Model	Value
Wait until process is complete for syndrome polynomial i: Wait until the ELM_IRQ interrupt is generated, or poll the status register.		
Wait for page completed interrupt: All error locations are valid.	ELM_IRQSTATUS[8] PAGE_VALID	0x1
Repeat the following actions the necessary number of times. That is, once for each valid defined block in the page.		

Table 10-496. ELM Processing Completion for Page Mode (continued)

Step	Register/Bit Field/Programming Model	Value
Read the process exit status.	ELM_LOCATION_STATUS_i[8] ECC_CORRECTABLE	
if the process fails (too many errors)	ELM_LOCATION_STATUS_i[8] ECC_CORRECTABLE	0x0
It is software dependent.		
else (process successful, the engine completes)	ELM_LOCATION_STATUS_i[8] ECC_CORRECTABLE	0x1
Read the number of errors.	ELM_LOCATION_STATUS_i[4:0] ECC_NB_ERRORS	
Read the error-location bit addresses for syndrome polynomial i of the ECC_NB_ERRORS first registers.	ELM_ERROR_LOCATION_0_i[12:0] ECC_ERROR_LOCATION	
	ELM_ERROR_LOCATION_1_i[12:0] ECC_ERROR_LOCATION	
	...	
	ELM_ERROR_LOCATION_15_i[12:0] ECC_ERROR_LOCATION	
endif		
End Repeat		
Clear the ELM_IRQSTATUS register.	ELM_IRQSTATUS	0x1FF

Next page can be correctly processed after a page is fully processed, when all tagged polynomials have been processed (ELM_IRQSTATUS[i] LOC_VALID_i = 0x1 for all syndrome polynomials i used in the page).

10.4.4.2 Use Case: ELM Used in Continuous Mode

In this example, the ELM is programmed for an 8-bit error-correction capability in continuous mode (see Table 10-497). After reading a 528-byte NAND flash sector (512B data plus 16B spare area) with a 16-bit interface, a nonzero polynomial syndrome is reported from the GPMC (polynomial syndrome 0 is used in the ELM):

- P = 0x0A16ABE115E44F767BFB0D0980

Table 10-497. Use Case: Continuous Mode

Step	Register/Bit Field/Programming Model	Value
Resets the module	ELM_SYSCONFIG[1] SOFTRESET	0x1
Wait until reset is done.	ELM_SYSSTATUS[0] RESETDONE	0x1
Configure the slave interface power management: Smart idle is used.	ELM_SYSCONFIG[4:3] SIDLEMODE	0x2
Defines the error-correction level used: 8 bits	ELM_LOCATION_CONFIG[1:0] ECC_BCH_LEVEL	0x1
Defines the maximum buffer length: 528 bytes (2 × 528 = 1056)	ELM_LOCATION_CONFIG[26:16] ECC_SIZE	0x420
Sets the ELM in continuous mode	ELM_PAGE_CTRL	0
Enables interrupt for syndrome polynomial 0	ELM_IRQENABLE[0] LOCATION_MASK_0	0x1
Set the input syndrome polynomial 0.	ELM_SYNDROME_FRAGMENT_0_i (where i = 0)	0xFB0D0980
	ELM_SYNDROME_FRAGMENT_1_i (where i = 0)	0xE44F767B
	ELM_SYNDROME_FRAGMENT_2_i (where i = 0)	0x16ABE115
	ELM_SYNDROME_FRAGMENT_3_i (where i = 0)	0x0000000A
Initiates the computation process	ELM_SYNDROME_FRAGMENT_6_i[16] SYNDROME_VALID (where i = 0)	0x1
Wait until process is complete for syndrome polynomial 0: IRQ_ELM is generated or poll the status register.		
Read that error-location process is complete for syndrome polynomial 0.	ELM_IRQSTATUS[0] LOC_VALID_0	0x1
Read the process exit status: All errors were successfully located.	ELM_LOCATION_STATUS_i[8] ECC_CORRECTABLE (where i = 0)	0x1

Table 10-497. Use Case: Continuous Mode (continued)

Step	Register/Bit Field/Programming Model	Value
Read the number of errors: Four errors detected.	ELM_LOCATION_STATUS_i[4:0] ECC_NB_ERRORS (where i = 0)	0x4
Read the error-location bit addresses for syndrome polynomial 0 of the first four registers: Errors are located in the data buffer at decimal addresses 431, 1062, 1909, 3452.	ELM_ERROR_LOCATION_0_i (where i = 0)	0x1AF
	ELM_ERROR_LOCATION_1_i (where i = 0)	0x426
	ELM_ERROR_LOCATION_2_i (where i = 0)	0x775
	ELM_ERROR_LOCATION_3_i (where i = 0)	0xD7C
Clear the corresponding interrupt for polynomial 0.	ELM_IRQSTATUS[0] LOC_VALID_0	0x1

The NAND flash data in the sector are seen as a polynomial of degree 4223 (number of bits in a 528 byte buffer minus 1), with each data bit being a coefficient in the polynomial. When reading from a NAND flash using the GPMC module, computation of the polynomial syndrome assumes that the first NAND word read at address 0x0 contains the highest-order coefficient in the message. Furthermore, in the 16-bit NAND word, bits are ordered from bit 7 to bit 0, and then from bit 15 to bit 8. Based on this convention, an address table of the data buffer can be built. NAND memory addresses in Table 10-498 are given in decimal format.

Table 10-498. 16-Bit NAND Sector Buffer Address Map

NAND Memory Address	Message Bit Addresses in Memory Word															
	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
0	4215	4214	4213	4212	4211	4210	4209	4208	4223	4222	4221	4220	4219	4218	4217	4216
1	4175	4174	4173	4172	4171	4170	4169	4168	4183	4182	4181	4180	4179	4178	4177	4176
...																
47	3463	3462	3461	3460	3459	3458	3457	3456	3471	3470	3469	3468	3467	3466	3465	3464
48	3447	3446	3445	3444	3443	3442	3441	3440	3455	3454	3453	3452	3451	3450	3449	3448
49	3431	3430	3429	3428	3427	3426	3425	3424	3439	3438	3437	3436	3435	3434	3433	3432
50	3415	3414	3413	3412	3411	3410	3409	3408	3423	3422	3421	3420	3419	3418	3417	3416
...																
255	135	134	133	132	131	130	129	128	143	142	141	140	139	138	137	136
256	119	118	117	116	115	114	113	112	127	126	125	124	123	122	121	120
257	103	102	101	100	99	98	97	96	111	110	109	108	107	106	105	104
258	87	86	85	84	83	82	81	80	95	94	93	92	91	90	89	88
259	71	70	69	68	67	66	65	64	79	78	77	76	75	74	73	72
260	55	54	53	52	51	50	49	48	63	62	61	60	59	58	57	56
261	39	38	37	36	35	34	33	32	47	46	45	44	43	42	41	40
262	23	22	21	20	19	18	17	16	31	30	29	28	27	26	25	24
263	7	6	5	4	3	2	1	0	15	14	13	12	11	10	9	8

The table can now be used to determine which bits in the buffer were incorrect and must be flipped. In this example, the first bit to be flipped is bit 4 from the 49th byte read from memory. It is up to the processor to correctly map this word to the copied buffer and flip this bit. The same process must be repeated for all detected errors.

10.4.4.3 Use Case: ELM Used in Page Mode

In this example, the ELM module is programmed for an 16-bit error-correction capability in page mode (see Table 10-499). After reading a 528-byte NAND flash sector (512B data plus 16B spare area) with a 16-bit interface, four non-zero polynomial syndromes are reported from the GPMC (polynomial syndrome 0, 1, 2, and 3 are used in the ELM):

- P0 = 0xE8B0 12ADDB5A318E05BE B0693DB28330B5CC A329AA05E0B718EF
- P1 = 0xBAD0 49A0D932C22E6669 0948DF08BE093336 79C6BA10E5F935EB

- P2 = 0x69D9 B86ABCD5EC3697FA A6498FEE54556EA0 1579EF7D60BA3189
- P3 = 0x0

Table 10-499. Use Case: Page Mode

Step	Register/Bit Field/Programming Model	Value
Resets the module	ELM_SYSCONFIG[1] SOFTRESET	0x1
Wait until reset is done.	ELM_SYSSTATUS[0] RESETDONE	0x1
Configure the slave interface power management: Smart idle is used.	ELM_SYSCONFIG[4:3] SIDLEMODE	0x2
Defines the error-correction level used: 16 bits	ELM_LOCATION_CONFIG[1:0] ECC_BCH_LEVEL	0x2
Defines the maximum buffer length: 528 bytes	ELM_LOCATION_CONFIG[26:16] ECC_SIZE	0x420
Sets the ELM in page mode (four blocks in a page)	ELM_PAGE_CTRL[0] SECTOR_0	0x1
	ELM_PAGE_CTRL[1] SECTOR_1	0x1
	ELM_PAGE_CTRL[2] SECTOR_2	0x1
	ELM_PAGE_CTRL[3] SECTOR_3	0x1
Disable all interrupts for syndrome polynomial and enable PAGE_MASK interrupt.	ELM_IRQENABLE	0x100
Set the input syndrome polynomial 0.	ELM_SYNDROME_FRAGMENT_0_i (where i = 0)	0xE0B718EF
	ELM_SYNDROME_FRAGMENT_1_i (where i = 0)	0xA329AA05
	ELM_SYNDROME_FRAGMENT_2_i (where i = 0)	0x8330B5CC
	ELM_SYNDROME_FRAGMENT_3_i (where i = 0)	0xB0693DB2
	ELM_SYNDROME_FRAGMENT_4_i (where i = 0)	0x318E05BE
	ELM_SYNDROME_FRAGMENT_5_i (where i = 0)	0x12ADDB5A
	ELM_SYNDROME_FRAGMENT_6_i (where i = 0)	0xE8B0
Set the input syndrome polynomial 1.	ELM_SYNDROME_FRAGMENT_0_i (where i = 1)	0xE5F935EB
	ELM_SYNDROME_FRAGMENT_1_i (where i = 1)	0x79C6BA10
	ELM_SYNDROME_FRAGMENT_2_i ((where i = 1)	0xBE093336
	ELM_SYNDROME_FRAGMENT_3_i (where i = 1)	0x0948DF08
	ELM_SYNDROME_FRAGMENT_4_i (where i = 1)	0xC22E6669
	ELM_SYNDROME_FRAGMENT_5_i (where i = 1)	0x49A0D932
	ELM_SYNDROME_FRAGMENT_6_i (where i = 1)	0xBAD0
Set the input syndrome polynomial 2.	ELM_SYNDROME_FRAGMENT_0_i (where i = 2)	0x60BA3189
	ELM_SYNDROME_FRAGMENT_1_i (where i = 2)	0x1579EF7D
	ELM_SYNDROME_FRAGMENT_2_i (where i = 2)	0x54556EA0
	ELM_SYNDROME_FRAGMENT_3_i (where i = 2)	0xA6498FEE
	ELM_SYNDROME_FRAGMENT_4_i (where i = 2)	0xEC3697FA
	ELM_SYNDROME_FRAGMENT_5_i (where i = 2)	0xB86ABCD5
	ELM_SYNDROME_FRAGMENT_6_i (where i = 2)	0x69D9
Set the input syndrome polynomial 3.	ELM_SYNDROME_FRAGMENT_0_i (where i = 3)	0x0
	ELM_SYNDROME_FRAGMENT_1_i (where i = 3)	0x0
	ELM_SYNDROME_FRAGMENT_2_i (where i = 3)	0x0
	ELM_SYNDROME_FRAGMENT_3_i (where i = 3)	0x0
	ELM_SYNDROME_FRAGMENT_4_i (where i = 3)	0x0
	ELM_SYNDROME_FRAGMENT_5_i (where i = 3)	0x0
	ELM_SYNDROME_FRAGMENT_6_i (where i = 3)	0x0
Initiates the computation process for syndrome polynomial 0	ELM_SYNDROME_FRAGMENT_6_i[16] SYNDROME_VALID (where i = 0)	0x1
Initiates the computation process for syndrome polynomial 1	ELM_SYNDROME_FRAGMENT_6_i[16] SYNDROME_VALID (where i = 1)	0x1
Initiates the computation process for syndrome polynomial 2	ELM_SYNDROME_FRAGMENT_6_i[16] SYNDROME_VALID (where i = 2)	0x1

Table 10-499. Use Case: Page Mode (continued)

Step	Register/Bit Field/Programming Model	Value
Initiates the computation process for syndrome polynomial 3	ELM_SYNDROME_FRAGMENT_6_i[16] SYNDROME_VALID (where i = 3)	0x1
Wait until process is complete for syndrome polynomial 0, 1, 2, and 3: Wait until the ELM_IRQ interrupt is generated or poll the status register.		
Wait for page completed interrupt: All error locations are valid.	ELM_IRQSTATUS[8] PAGE_VALID	0x1
Read the process exit status for syndrome polynomial 0: All errors were successfully located.	ELM_LOCATION_STATUS_i[8] ECC_CORRECTABLE (where i = 0)	0x1
Read the process exit status for syndrome polynomial 1: All errors were successfully located.	ELM_LOCATION_STATUS_i[8] ECC_CORRECTABLE (where i = 1)	0x1
Read the process exit status for syndrome polynomial 2: All errors were successfully located.	ELM_LOCATION_STATUS_i[8] ECC_CORRECTABLE (where i = 2)	0x1
Read the process exit status for syndrome polynomial 3: All errors were successfully located.	ELM_LOCATION_STATUS_i[8] ECC_CORRECTABLE (where i = 3)	0x1
Read the number of errors for syndrome polynomial 0: four errors detected.	ELM_LOCATION_STATUS_i[4:0] ECC_NB_ERRORS (where i = 0)	0x4
Read the number of errors for syndrome polynomial 1: two errors detected.	ELM_LOCATION_STATUS_i[4:0] ECC_NB_ERRORS (where i = 1)	0x2
Read the number of errors for syndrome polynomial 2: one error detected.	ELM_LOCATION_STATUS_i[4:0] ECC_NB_ERRORS (where i = 2)	0x1
Read the number of errors for syndrome polynomial 3: no errors detected.	ELM_LOCATION_STATUS_i[4:0] ECC_NB_ERRORS (where i = 3)	0x0
Read the error-location bit addresses for syndrome polynomial 0 of the first four registers:	ELM_ERROR_LOCATION_0_i (where i = 0)	0x1FE
	ELM_ERROR_LOCATION_1_i (where i = 0)	0x617
	ELM_ERROR_LOCATION_2_i (where i = 0)	0x650
	ELM_ERROR_LOCATION_3_i (where i = 0)	0xA83
Read the error-location bit addresses for syndrome polynomial 1 of the first two registers:	ELM_ERROR_LOCATION_0_i (where i = 1)	0x4
	ELM_ERROR_LOCATION_1_i (where i = 1)	0x1036
Read the errors location bit addresses for syndrome polynomial 2 of the first registers:	ELM_ERROR_LOCATION_0_i (where i = 1)	0x3E8
Clear the ELM_IRQSTATUS register.	ELM_IRQSTATUS	0x1FF

10.4.5 ELM Register Manual

10.4.5.1 ELM Instance Summary

Table 10-500summarizes the ELM instance.

Table 10-500. ELM Instance Summary

Module Name	Base Address	Size
ELM	0x4807 8000	4 KiB

10.4.5.2 ELM Registers

10.4.5.2.1 ELM Register Summary

Table 10-501 summarizes the ELM register mapping.

Table 10-501. ELM Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	Physical Address ELM
ELM_REVISION	R	32	0x0000 0000	0x4807 8000
ELM_SYSCONFIG	RW	32	0x0000 0010	0x4807 8010
ELM_SYSSTATUS	R	32	0x0000 0014	0x4807 8014
ELM_IRQSTATUS	RW	32	0x0000 0018	0x4807 8018
ELM_IRQENABLE	RW	32	0x0000 001C	0x4807 801C
ELM_LOCATION_CONFIG	RW	32	0x0000 0020	0x4807 8020
ELM_PAGE_CTRL	RW	32	0x0000 0080	0x4807 8080
ELM_SYNDROME_FRAGMENT_0_i (1)	RW	32	0x0000 0400 + (0x40 * i)	0x4807 8400 + (0x40 * i)
ELM_SYNDROME_FRAGMENT_1_i (1)	RW	32	0x0000 0404 + (0x40 * i)	0x4807 8404 + (0x40 * i)
ELM_SYNDROME_FRAGMENT_2_i (1)	RW	32	0x0000 0408 + (0x40 * i)	0x4807 8408 + (0x40 * i)
ELM_SYNDROME_FRAGMENT_3_i (1)	RW	32	0x0000 040C + (0x40 * i)	0x4807 840C + (0x40 * i)
ELM_SYNDROME_FRAGMENT_4_i (1)	RW	32	0x0000 0410 + (0x40 * i)	0x4807 8410 + (0x40 * i)
ELM_SYNDROME_FRAGMENT_5_i (1)	RW	32	0x0000 0414 + (0x40 * i)	0x4807 8414 + (0x40 * i)
ELM_SYNDROME_FRAGMENT_6_i (1)	RW	32	0x0000 0418 + (0x40 * i)	0x4807 8418 + (0x40 * i)
ELM_LOCATION_STATUS_i (1)	R	32	0x0000 0800 + (0x100 * i)	0x4807 8800 + (0x100 * i)
ELM_ERROR_LOCATION_0_i (1)	R	32	0x0000 0880 + (0x100 * i)	0x4807 8880 + (0x100 * i)
ELM_ERROR_LOCATION_1_i (1)	R	32	0x0000 0884 + (0x100 * i)	0x4807 8884 + (0x100 * i)
ELM_ERROR_LOCATION_2_i (1)	R	32	0x0000 0888 + (0x100 * i)	0x4807 8888 + (0x100 * i)
ELM_ERROR_LOCATION_3_i (1)	R	32	0x0000 088C + (0x100 * i)	0x4807 888C + (0x100 * i)
ELM_ERROR_LOCATION_4_i (1)	R	32	0x0000 0890 + (0x100 * i)	0x4807 8890 + (0x100 * i)
ELM_ERROR_LOCATION_5_i (1)	R	32	0x0000 0894 + (0x100 * i)	0x4807 8894 + (0x100 * i)
ELM_ERROR_LOCATION_6_i (1)	R	32	0x0000 0898 + (0x100 * i)	0x4807 8898 + (0x100 * i)
ELM_ERROR_LOCATION_7_i (1)	R	32	0x0000 089C + (0x100 * i)	0x4807 889C + (0x100 * i)
ELM_ERROR_LOCATION_8_i (1)	R	32	0x0000 08A0 + (0x100 * i)	0x4807 88A0 + (0x100 * i)
ELM_ERROR_LOCATION_9_i (1)	R	32	0x0000 08A4 + (0x100 * i)	0x4807 88A4 + (0x100 * i)
ELM_ERROR_LOCATION_10_i (1)	R	32	0x0000 08A8 + (0x100 * i)	0x4807 88A8 + (0x100 * i)
ELM_ERROR_LOCATION_11_i (1)	R	32	0x0000 08AC + (0x100 * i)	0x4807 88AC + (0x100 * i)
ELM_ERROR_LOCATION_12_i (1)	R	32	0x0000 08B0 + (0x100 * i)	0x4807 88B0 + (0x100 * i)
ELM_ERROR_LOCATION_13_i (1)	R	32	0x0000 08B4 + (0x100 * i)	0x4807 88B4 + (0x100 * i)
ELM_ERROR_LOCATION_14_i (1)	R	32	0x0000 08B8 + (0x100 * i)	0x4807 88B8 + (0x100 * i)
ELM_ERROR_LOCATION_15_i (1)	R	32	0x0000 08BC + (0x100 * i)	0x4807 88BC + (0x100 * i)

(1) i = 0 to 7 for ELM

10.4.5.2.2 ELM Register Description

Table 10-502 through Table 10-562 describe the individual ELM registers.

Table 10-502. ELM_REVISION

Address Offset	0x0000 0000		
Physical Address	0x4807 8000	**Instance**	ELM
Description	This register contains the IP revision code. (A write or reset of to this register has no effect.)		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REVISION																															

Bits	Field Name	Description	Type	Reset
31:0	REVISION	IP Revision (TI internal data)	R	0x-

Table 10-503. Register Call Summary for Register ELM_REVISION

Error Location Module
- ELM Register Summary: [0]

Table 10-504. ELM_SYSCONFIG

Address Offset	0x0000 0010		
Physical Address	0x4807 8010	**Instance**	ELM
Description	This register allows controlling various parameters of the OCP interface.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							CLOCKACTIVITYOCP	RESERVED			SIDLEMODE		RESERVED	SOFTRESET	AUTOGATING

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reserved	R	0x000000
8	CLOCKACTIVITYOCP	OCP clock activity when module is in IDLE mode (during wake-up mode period) 0x0: OCP clock can be switched off. 0x1: OCP clock is maintained during wake-up period.	RW	0
7:5	RESERVED	Reserved	R	0x0
4:3	SIDLEMODE	Slave interface power management (IDLE req/ack control) 0x0: Force-idle. IDLE request is acknowledged unconditionally and immediately (Default *Dumb* mode for safety) 0x1: No-idle. IDLE request is never acknowledged. 0x2: Smart-idle. The acknowledgment to an IDLE request is given based on the internal activity. 0x3: Reserved - do not use	RW	0x2

Bits	Field Name	Description	Type	Reset
2	RESERVED	Reserved	R	0
1	SOFTRESET	Module software reset This bit is automatically reset by hardware (during reads, it always returns 0). It has same effect as the OCP hardware reset. 0x0: Normal mode 0x1: Start soft reset sequence.	RW	0
0	AUTOGATING	Internal OCP clock gating strategy (no module visible effect other than saving power) 0x0: OCP clock is free-running. 0x1: Automatic internal OCP clock gating strategy is applied based on the OCP interface activity.	RW	1

Table 10-505. Register Call Summary for Register ELM_SYSCONFIG

Error Location Module

- ELM Software Reset: [0][1]
- ELM Power Management: [2][3][4]
- Processing Initialization: [5][6]
- Use Case: ELM Used in Continuous Mode: [7][8]
- Use Case: ELM Used in Page Mode: [9][10]
- ELM Register Summary: [11]

Table 10-506. ELM_SYSSTATUS

Address Offset	0x0000 0014		
Physical Address	0x4807 8014	**Instance**	ELM
Description	Internal reset monitoring (OCP domain) Undefined since: From hardware perspective, the reset state is 0. From software user perspective, when the accessible module is 1.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															RESETDONE

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved	R	0x0000 0000
0	RESETDONE	Internal reset monitoring (OCP domain) Undefined since: From hardware perspective, the reset state is 0. From software user perspective, when the accessible module is 1. Read 0x0: Reset is ongoing. Read 0x1: Reset is done (completed).	R	1

Table 10-507. Register Call Summary for Register ELM_SYSSTATUS

Error Location Module

- ELM Software Reset: [0]
- Processing Initialization: [1]
- Use Case: ELM Used in Continuous Mode: [2]
- Use Case: ELM Used in Page Mode: [3]
- ELM Register Summary: [4]

Table 10-508. ELM_IRQSTATUS

Address Offset	0x0000 0018		
Physical Address	0x4807 8018	**Instance**	ELM
Description	Interrupt status. This register doubles as a status register for the error-location processes.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							PAGE_VALID	LOC_VALID_7	LOC_VALID_6	LOC_VALID_5	LOC_VALID_4	LOC_VALID_3	LOC_VALID_2	LOC_VALID_1	LOC_VALID_0

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reserved	R	0x000000
8	PAGE_VALID	Error-location status for a full page, based on the mask definition Read 0x0: Error locations invalid for all polynomials enabled in the ECC_INTERRUPT_MASK register Read 0x1: All error locations valid Write 0x0: No effect Write 0x1: Clear interrupt	RW	0
7	LOC_VALID_7	Error-location status for syndrome polynomial 7 Read 0x0: No syndrome processed or process in progress Read 0x1: Error-location process completed Write 0x0: No effect Write 0x1: Clear interrupt	RW W1toClr	0
6	LOC_VALID_6	Error-location status for syndrome polynomial 6	RW W1toClr	0
5	LOC_VALID_5	Error-location status for syndrome polynomial 5	RW W1toClr	0
4	LOC_VALID_4	Error-location status for syndrome polynomial 4	RW W1toClr	0
3	LOC_VALID_3	Error-location status for syndrome polynomial 3	RW W1toClr	0
2	LOC_VALID_2	Error-location status for syndrome polynomial 2	RW W1toClr	0
1	LOC_VALID_1	Error-location status for syndrome polynomial 1	RW W1toClr	0
0	LOC_VALID_0	Error-location status for syndrome polynomial 0	RW W1toClr	0

Table 10-509. Register Call Summary for Register ELM_IRQSTATUS

Error Location Module

- ELM Interrupt Requests: [0][1][2][3][4][5][6][7][8]
- Processing Initialization: [9]
- Processing Completion: [10][11][12][13][14][15][16][17][18]
- Read Results: [19][20][21][22][23][24]
- Use Case: ELM Used in Continuous Mode: [25][26]
- Use Case: ELM Used in Page Mode: [27][28][29]
- ELM Register Summary: [30]

Table 10-510. ELM_IRQENABLE

Address Offset	0x0000 001C		
Physical Address	0x4807 801C	**Instance**	ELM
Description	Interrupt enable		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							PAGE_MASK	LOCATION_MASK_7	LOCATION_MASK_6	LOCATION_MASK_5	LOCATION_MASK_4	LOCATION_MASK_3	LOCATION_MASK_2	LOCATION_MASK_1	LOCATION_MASK_0

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reserved	R	0x000000
8	PAGE_MASK	Page interrupt mask bit 0: Disable interrupt 1: Enable interrupt	RW	0
7	LOCATION_MASK_7	Error-location interrupt mask bit for syndrome polynomial 7	RW	0
6	LOCATION_MASK_6	Error-location interrupt mask bit for syndrome polynomial 6	RW	0
5	LOCATION_MASK_5	Error-location interrupt mask bit for syndrome polynomial 5	RW	0
4	LOCATION_MASK_4	Error-location interrupt mask bit for syndrome polynomial 4	RW	0
3	LOCATION_MASK_3	Error-location interrupt mask bit for syndrome polynomial 3	RW	0
2	LOCATION_MASK_2	Error-location interrupt mask bit for syndrome polynomial 2	RW	0
1	LOCATION_MASK_1	Error-location interrupt mask bit for syndrome polynomial 1	RW	0
0	LOCATION_MASK_0	Error-location interrupt mask bit for syndrome polynomial 0 0: Disable interrupt 1: Enable interrupt	RW	0

Table 10-511. Register Call Summary for Register ELM_IRQENABLE

Error Location Module

- ELM Interrupt Requests: [0][1][2][3][4][5][6][7][8]
- Processing Initialization: [9][10]
- Processing Completion: [11]
- Processing Initialization: [12][13][14]
- Use Case: ELM Used in Continuous Mode: [15]
- Use Case: ELM Used in Page Mode: [16]
- ELM Register Summary: [17]

Table 10-512. ELM_LOCATION_CONFIG

Address Offset	0x0000 0020		
Physical Address	0x4807 8020	**Instance**	ELM
Description	ECC algorithm parameters		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								ECC_SIZE								RESERVED														ECC_BCH_LEVEL	

Bits	Field Name	Description	Type	Reset
31:27	RESERVED	Reserved	R	0x00
26:16	ECC_SIZE	Maximum size of the buffers for which the error-location engine is used, in number of nibbles (4-bit entities)	RW	0x000
15:2	RESERVED	Reserved	R	0x0000
1:0	ECC_BCH_LEVEL	Error correction level 0x0: 4 bits 0x1: 8 bits 0x2: 16 bits 0x3: Reserved	RW	0x0

Table 10-513. Register Call Summary for Register ELM_LOCATION_CONFIG

Error Location Module

- Processing Initialization: [0][1][2]
- Processing Initialization: [3][4]
- Use Case: ELM Used in Continuous Mode: [5][6]
- Use Case: ELM Used in Page Mode: [7][8]
- ELM Register Summary: [9]

Table 10-514. ELM_PAGE_CTRL

Address Offset	0x0000 0080		
Physical Address	0x4807 8080	**Instance**	ELM
Description	Page definition		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								SECTOR_7	SECTOR_6	SECTOR_5	SECTOR_4	SECTOR_3	SECTOR_2	SECTOR_1	SECTOR_0

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Reserved	R	0x000000
7	SECTOR_7	Set to 1 if syndrome polynomial 7 is part of the page in page mode. Must be 0 in continuous mode.	RW	0
6	SECTOR_6	Set to 1 if syndrome polynomial 6 is part of the page in page mode. Must be 0 in continuous mode.	RW	0

Bits	Field Name	Description	Type	Reset
5	SECTOR_5	Set to 1 if syndrome polynomial 5 is part of the page in page mode. Must be 0 in continuous mode.	RW	0
4	SECTOR_4	Set to 1 if syndrome polynomial 4 is part of the page in page mode. Must be 0 in continuous mode.	RW	0
3	SECTOR_3	Set to 1 if syndrome polynomial 3 is part of the page in page mode. Must be 0 in continuous mode.	RW	0
2	SECTOR_2	Set to 1 if syndrome polynomial 2 is part of the page in page mode. Must be 0 in continuous mode.	RW	0
1	SECTOR_1	Set to 1 if syndrome polynomial 1 is part of the page in page mode. Must be 0 in continuous mode.	RW	0
0	SECTOR_0	Set to 1 if syndrome polynomial 0 is part of the page in page mode. Must be 0 in continuous mode.	RW	0

Table 10-515. Register Call Summary for Register ELM_PAGE_CTRL

Error Location Module

- Processing Initialization: [0]
- Processing Completion: [1][2][3][4]
- Processing Initialization: [5][6][7]
- Use Case: ELM Used in Continuous Mode: [8]
- Use Case: ELM Used in Page Mode: [9][10][11][12]
- ELM Register Summary: [13]

Table 10-516. ELM_SYNDROME_FRAGMENT_0_i

Address Offset	0x0000 0400 + (0x40 * i)	**Index**	i = 0 to 7
Physical Address	0x4807 8400 + (0x40 * i)	**Instance**	ELM
Description	Input syndrome polynomial bits 0 to 31.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SYNDROME_0																															

Bits	Field Name	Description	Type	Reset
31:0	SYNDROME_0	Syndrome bits 0 to 31	RW	0x0000 0000

Table 10-517. Register Call Summary for Register ELM_SYNDROME_FRAGMENT_0_i

Error Location Module

- Processing Initialization: [0]
- Processing Initialization: [1]
- Use Case: ELM Used in Continuous Mode: [2]
- Use Case: ELM Used in Page Mode: [3][4][5][6]
- ELM Register Summary: [7]

Table 10-518. ELM_SYNDROME_FRAGMENT_1_i

Address Offset	0x0000 0404 + (0x40 * i)	**Index**	i = 0 to 7
Physical Address	0x4807 8404 + (0x40 * i)	**Instance**	ELM
Description	Input syndrome polynomial bits 32 to 63.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SYNDROME_1																															

Bits	Field Name	Description	Type	Reset
31:0	SYNDROME_1	Syndrome bits 32 to 63	RW	0x0000 0000

Table 10-519. Register Call Summary for Register ELM_SYNDROME_FRAGMENT_1_i

Error Location Module

- Processing Initialization: [0]
- Use Case: ELM Used in Continuous Mode: [1]
- Use Case: ELM Used in Page Mode: [2][3][4][5]
- ELM Register Summary: [6]

Table 10-520. ELM_SYNDROME_FRAGMENT_2_i

Address Offset	0x0000 0408 + (0x40 * i)	**Index**	i = 0 to 7
Physical Address	0x4807 8408 + (0x40 * i)	**Instance**	ELM
Description	Input syndrome polynomial bits 64 to 95.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SYNDROME_2																															

Bits	Field Name	Description	Type	Reset
31:0	SYNDROME_2	Syndrome bits 64 to 95	RW	0x0000 0000

Table 10-521. Register Call Summary for Register ELM_SYNDROME_FRAGMENT_2_i

Error Location Module

- Use Case: ELM Used in Continuous Mode: [0]
- Use Case: ELM Used in Page Mode: [1][2][3][4]
- ELM Register Summary: [5]

Table 10-522. ELM_SYNDROME_FRAGMENT_3_i

Address Offset	0x0000 040C + (0x40 * i)	**Index**	i = 0 to 7
Physical Address	0x4807 840C + (0x40 * i)	**Instance**	ELM
Description	Input syndrome polynomial bits 96 to 127		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SYNDROME_3																															

Bits	Field Name	Description	Type	Reset
31:0	SYNDROME_3	Syndrome bits 96 to 127	RW	0x0000 0000

Table 10-523. Register Call Summary for Register ELM_SYNDROME_FRAGMENT_3_i

Error Location Module

- Use Case: ELM Used in Continuous Mode: [0]
- Use Case: ELM Used in Page Mode: [1][2][3][4]
- ELM Register Summary: [5]

Table 10-524. ELM_SYNDROME_FRAGMENT_4_i

Address Offset	0x0000 0410 + (0x40 * i)	**Index**	i = 0 to 7
Physical Address	0x4807 8410 + (0x40 * i)	**Instance**	ELM
Description	Input syndrome polynomial bits 128 to 159.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SYNDROME_4																															

Bits	Field Name	Description	Type	Reset
31:0	SYNDROME_4	Syndrome bits 128 to 159	RW	0x0000 0000

Table 10-525. Register Call Summary for Register ELM_SYNDROME_FRAGMENT_4_i

Error Location Module

- Use Case: ELM Used in Page Mode: [0][1][2][3]
- ELM Register Summary: [4]

Table 10-526. ELM_SYNDROME_FRAGMENT_5_i

Address Offset	0x0000 0414 + (0x40 * i)	**Index**	i = 0 to 7
Physical Address	0x4807 8414 + (0x40 * i)	**Instance**	ELM
Description	Input syndrome polynomial bits 160 to 191.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SYNDROME_5																															

Bits	Field Name	Description	Type	Reset
31:0	SYNDROME_5	Syndrome bits 160 to 191	RW	0x0000 0000

Table 10-527. Register Call Summary for Register ELM_SYNDROME_FRAGMENT_5_i

Error Location Module

- Processing Initialization: [0]
- Use Case: ELM Used in Page Mode: [1][2][3][4]
- ELM Register Summary: [5]

Table 10-528. ELM_SYNDROME_FRAGMENT_6_i

Address Offset	0x0000 0418 + (0x40 * i)	**Index**	i = 0 to 7
Physical Address	0x4807 8418 + (0x40 * i)	**Instance**	ELM
Description	Input syndrome polynomial bits 192 to 207.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED															SYNDROME_VALID	SYNDROME_6															

Bits	Field Name	Description	Type	Reset
31:17	RESERVED	Reserved	R	0x0000
16	SYNDROME_VALID	Syndrome valid bit 0x0: This syndrome polynomial must not be processed. 0x1: This syndrome polynomial must be processed.	RW	0
15:0	SYNDROME_6	Syndrome bits 192 to 207	RW	0x0000

Table 10-529. Register Call Summary for Register ELM_SYNDROME_FRAGMENT_6_i

Error Location Module

- Processing Initialization: [0][1][2][3]
- Processing Completion: [4][5]
- Processing Initialization: [6][7]
- Read Results: [8]
- Use Case: ELM Used in Continuous Mode: [9]
- Use Case: ELM Used in Page Mode: [10][11][12][13][14][15][16][17]
- ELM Register Summary: [18]

Table 10-530. ELM_LOCATION_STATUS_i

Address Offset	0x0000 0800 + (0x100 * i)	**Index**	i = 0 to 7
Physical Address	0x4807 8800 + (0x100 * i)	**Instance**	ELM
Description	Exit status for the syndrome polynomial processing		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							ECC_CORRECTABLE	RESERVED			ECC_NB_ERRORS				

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reserved	R	0x000000
8	ECC_CORRECTABLE	Error-location process exit status 0x0: ECC error-location process failed. Number of errors and error locations are invalid. 0x1: All errors were successfully located. Number of errors and error locations are valid.	R	0
7:5	RESERVED	Reserved	R	0x0
4:0	ECC_NB_ERRORS	Number of errors detected and located	R	0x00

Table 10-531. Register Call Summary for Register ELM_LOCATION_STATUS_i

Error Location Module

- Processing Initialization: [0]
- Processing Sequence: [1][2][3][4][5]
- Processing Completion: [6][7][8]
- Read Results: [9][10][11][12][13][14][15][16]
- Use Case: ELM Used in Continuous Mode: [17][18]
- Use Case: ELM Used in Page Mode: [19][20][21][22][23][24][25][26]
- ELM Register Summary: [27]

Table 10-532. ELM_ERROR_LOCATION_0_i

Address Offset	0x0000 0880 + (0x100 * i)	**Index**	i = 0 to 7
Physical Address	0x4807 8880 + (0x100 * i)	**Instance**	ELM
Description	Error-location register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								ECC_ERROR_LOCATION							

Bits	Field Name	Description	Type	Reset
31:13	RESERVED	Reserved	R	0x00000
12:0	ECC_ERROR_LOCATION	Error-location bit address	R	0x0000

Table 10-533. Register Call Summary for Register ELM_ERROR_LOCATION_0_i

Error Location Module

- Processing Initialization: [0]
- Processing Sequence: [1][2][3]
- Processing Completion: [4]
- Read Results: [5][6]
- Use Case: ELM Used in Continuous Mode: [7]
- Use Case: ELM Used in Page Mode: [8][9][10]
- ELM Register Summary: [11]

Table 10-534. ELM_ERROR_LOCATION_1_i

Address Offset	0x0000 0884 + (0x100 * i)	**Index**	i = 0 to 7
Physical Address	0x4807 8884 + (0x100 * i)	**Instance**	ELM
Description	Error-location register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																ECC_ERROR_LOCATION															

Bits	Field Name	Description	Type	Reset
31:13	RESERVED	Reserved	R	0x00000
12:0	ECC_ERROR_LOCATION	Error-location bit address	R	0x0000

Table 10-535. Register Call Summary for Register ELM_ERROR_LOCATION_1_i

Error Location Module

- Processing Sequence: [0]
- Read Results: [1][2]
- Use Case: ELM Used in Continuous Mode: [3]
- Use Case: ELM Used in Page Mode: [4][5]
- ELM Register Summary: [6]

Table 10-536. ELM_ERROR_LOCATION_2_i

Address Offset	0x0000 0888 + (0x100 * i)	**Index**	i = 0 to 7
Physical Address	0x4807 8888 + (0x100 * i)	**Instance**	ELM
Description	Error-location register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																ECC_ERROR_LOCATION															

Bits	Field Name	Description	Type	Reset
31:13	RESERVED	Reserved	R	0x00000
12:0	ECC_ERROR_LOCATION	Error-location bit address	R	0x0000

Table 10-537. Register Call Summary for Register ELM_ERROR_LOCATION_2_i

Error Location Module

- Use Case: ELM Used in Continuous Mode: [0]
- Use Case: ELM Used in Page Mode: [1]
- ELM Register Summary: [2]

Table 10-538. ELM_ERROR_LOCATION_3_i

Address Offset	0x0000 088C + (0x100 * i)	**Index**	i = 0 to 7
Physical Address	0x4807 888C + (0x100 * i)	**Instance**	ELM
Description	Error-location register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								ECC_ERROR_LOCATION							

Bits	Field Name	Description	Type	Reset
31:13	RESERVED	Reserved	R	0x00000
12:0	ECC_ERROR_LOCATION	Error-location bit address	R	0x0000

Table 10-539. Register Call Summary for Register ELM_ERROR_LOCATION_3_i

Error Location Module

- Use Case: ELM Used in Continuous Mode: [0]
- Use Case: ELM Used in Page Mode: [1]
- ELM Register Summary: [2]

Table 10-540. ELM_ERROR_LOCATION_4_i

Address Offset	0x0000 0890 + (0x100 * i)	**Index**	i = 0 to 7
Physical Address	0x4807 8890 + (0x100 * i)	**Instance**	ELM
Description	Error-location register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								ECC_ERROR_LOCATION							

Bits	Field Name	Description	Type	Reset
31:13	RESERVED	Reserved	R	0x00000
12:0	ECC_ERROR_LOCATION	Error-location bit address	R	0x0000

Table 10-541. Register Call Summary for Register ELM_ERROR_LOCATION_4_i

Error Location Module

- ELM Register Summary: [0]

Table 10-542. ELM_ERROR_LOCATION_5_i

Address Offset	0x0000 0894 + (0x100 * i)	**Index**	i = 0 to 7
Physical Address	0x4807 8894 + (0x100 * i)	**Instance**	ELM
Description	Error-location register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								ECC_ERROR_LOCATION							

Bits	Field Name	Description	Type	Reset
31:13	RESERVED	Reserved	R	0x00000
12:0	ECC_ERROR_LOCATION	Error-location bit address	R	0x0000

Table 10-543. Register Call Summary for Register ELM_ERROR_LOCATION_5_i

Error Location Module

- ELM Register Summary: [0]

Table 10-544. ELM_ERROR_LOCATION_6_i

Address Offset	0x0000 0898 + (0x100 * i)	**Index**	i = 0 to 7
Physical Address	0x4807 8898 + (0x100 * i)	**Instance**	ELM
Description	Error-location register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								ECC_ERROR_LOCATION							

Bits	Field Name	Description	Type	Reset
31:13	RESERVED	Reserved	R	0x00000
12:0	ECC_ERROR_LOCATION	Error-location bit address	R	0x0000

Table 10-545. Register Call Summary for Register ELM_ERROR_LOCATION_6_i

Error Location Module

- ELM Register Summary: [0]

Table 10-546. ELM_ERROR_LOCATION_7_i

Address Offset	0x0000 089C + (0x100 * i)	**Index**	i = 0 to 7
Physical Address	0x4807 889C + (0x100 * i)	**Instance**	ELM
Description	Error-location register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								ECC_ERROR_LOCATION							

Bits	Field Name	Description	Type	Reset
31:13	RESERVED	Reserved	R	0x00000
12:0	ECC_ERROR_LOCATION	Error-location bit address	R	0x0000

Table 10-547. Register Call Summary for Register ELM_ERROR_LOCATION_7_i

Error Location Module

- ELM Register Summary: [0]

Table 10-548. ELM_ERROR_LOCATION_8_i

Address Offset	0x0000 08A0 + (0x100 * i)	**Index**	i = 0 to 7
Physical Address	0x4807 88A0 + (0x100 * i)	**Instance**	ELM
Description	Error-location register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																ECC_ERROR_LOCATION															

Bits	Field Name	Description	Type	Reset
31:13	RESERVED	Reserved	R	0x00000
12:0	ECC_ERROR_LOCATION	Error-location bit address	R	0x0000

Table 10-549. Register Call Summary for Register ELM_ERROR_LOCATION_8_i

Error Location Module

- ELM Register Summary: [0]

Table 10-550. ELM_ERROR_LOCATION_9_i

Address Offset	0x0000 08A4 + (0x100 * i)	**Index**	i = 0 to 7
Physical Address	0x4807 88A4 + (0x100 * i)	**Instance**	ELM
Description	Error-location register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																ECC_ERROR_LOCATION															

Bits	Field Name	Description	Type	Reset
31:13	RESERVED	Reserved	R	0x00000
12:0	ECC_ERROR_LOCATION	Error-location bit address	R	0x0000

Table 10-551. Register Call Summary for Register ELM_ERROR_LOCATION_9_i

Error Location Module

- ELM Register Summary: [0]

Table 10-552. ELM_ERROR_LOCATION_10_i

Address Offset	0x0000 08A8 + (0x100 * i)	**Index**	i = 0 to 7
Physical Address	0x4807 88A8 + (0x100 * i)	**Instance**	ELM
Description	Error-location register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																ECC_ERROR_LOCATION															

Bits	Field Name	Description	Type	Reset
31:13	RESERVED	Reserved	R	0x00000
12:0	ECC_ERROR_LOCATION	Error-location bit address	R	0x0000

Table 10-553. Register Call Summary for Register ELM_ERROR_LOCATION_10_i

Error Location Module

- ELM Register Summary: [0]

Table 10-554. ELM_ERROR_LOCATION_11_i

Address Offset	0x0000 08AC + (0x100 * i)	**Index**	i = 0 to 7
Physical Address	0x4807 88AC + (0x100 * i)	**Instance**	ELM
Description	Error-location register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								ECC_ERROR_LOCATION							

Bits	Field Name	Description	Type	Reset
31:13	RESERVED	Reserved	R	0x00000
12:0	ECC_ERROR_LOCATION	Error-location bit address	R	0x0000

Table 10-555. Register Call Summary for Register ELM_ERROR_LOCATION_11_i

Error Location Module

- ELM Register Summary: [0]

Table 10-556. ELM_ERROR_LOCATION_12_i

Address Offset	0x0000 08B0 + (0x100 * i)	**Index**	i = 0 to 7
Physical Address	0x4807 88B0 + (0x100 * i)	**Instance**	ELM
Description	Error-location register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								ECC_ERROR_LOCATION							

Bits	Field Name	Description	Type	Reset
31:13	RESERVED	Reserved	R	0x00000
12:0	ECC_ERROR_LOCATION	Error-location bit address	R	0x0000

Table 10-557. Register Call Summary for Register ELM_ERROR_LOCATION_12_i

Error Location Module

- ELM Register Summary: [0]

Table 10-558. ELM_ERROR_LOCATION_13_i

Address Offset	0x0000 08B4 + (0x100 * i)	**Index**	i = 0 to 7
Physical Address	0x4807 88B4 + (0x100 * i)	**Instance**	ELM
Description	Error-location register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								ECC_ERROR_LOCATION							

Bits	Field Name	Description	Type	Reset
31:13	RESERVED	Reserved	R	0x00000
12:0	ECC_ERROR_LOCATION	Error-location bit address	R	0x0000

Table 10-559. Register Call Summary for Register ELM_ERROR_LOCATION_13_i

Error Location Module

- ELM Register Summary: [0]

Table 10-560. ELM_ERROR_LOCATION_14_i

Address Offset	0x0000 08B8 + (0x100 * i)	**Index**	i = 0 to 7
Physical Address	0x4807 88B8 + (0x100 * i)	**Instance**	ELM
Description	Error-location register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								ECC_ERROR_LOCATION							

Bits	Field Name	Description	Type	Reset
31:13	RESERVED	Reserved	R	0x00000
12:0	ECC_ERROR_LOCATION	Error-location bit address	R	0x0000

Table 10-561. Register Call Summary for Register ELM_ERROR_LOCATION_14_i

Error Location Module

- ELM Register Summary: [0]

Table 10-562. ELM_ERROR_LOCATION_15_i

Address Offset	0x0000 08BC + (0x100 * i)	**Index**	i = 0 to 7
Physical Address	0x4807 88BC + (0x100 * i)	**Instance**	ELM
Description	Error-location register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								ECC_ERROR_LOCATION							

Bits	Field Name	Description	Type	Reset
31:13	RESERVED	Reserved	R	0x00000
12:0	ECC_ERROR_LOCATION	Error-location bit address	R	0x0000

Table 10-563. Register Call Summary for Register ELM_ERROR_LOCATION_15_i

Error Location Module

- Processing Initialization: [0]
- Processing Sequence: [1][2]
- Processing Completion: [3]
- Read Results: [4][5]
- ELM Register Summary: [6]

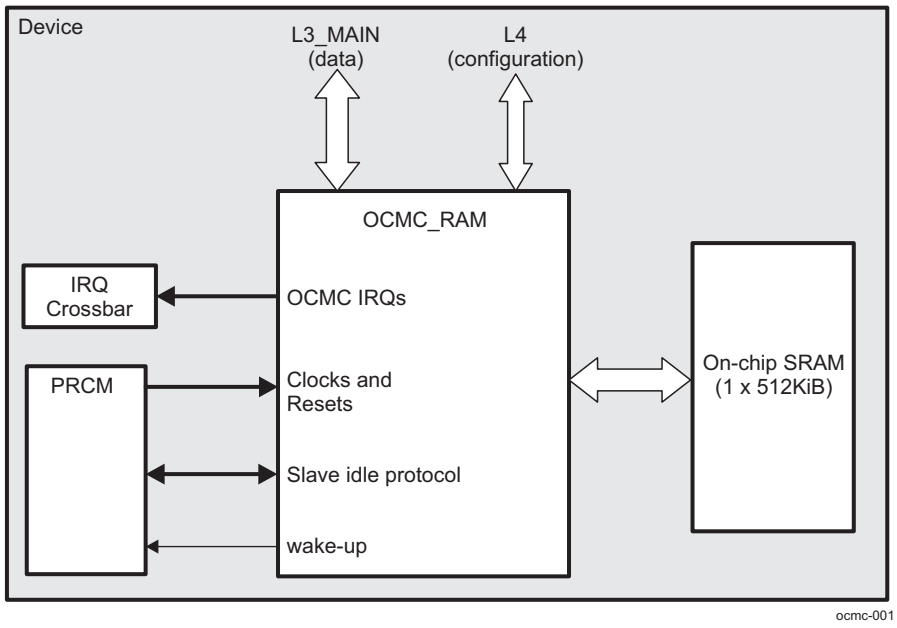
10.5 On-Chip Memory (OCM) Subsystem

10.5.1 OCM Subsystem Overview

The OCM subsystem consists of one OCM Controller (OCMC) that is associated with the on-chip RAM. This is the OCMC_RAM with 512 KiB of dedicated memory space. The OCM controller is also introduced in Section 10.1.4, *OCM Overview* of Section 10.1, *Memory Subsystem Overview*.

Figure 10-65 shows theOCMC_RAM controller.

Figure 10-65. OCMC_RAM Overview



10.5.2 OCM Subsystem Integration

Figure 10-66 shows the integration of the OCMC_RAM in the device.

Figure 10-66. OCMC_RAM Integration

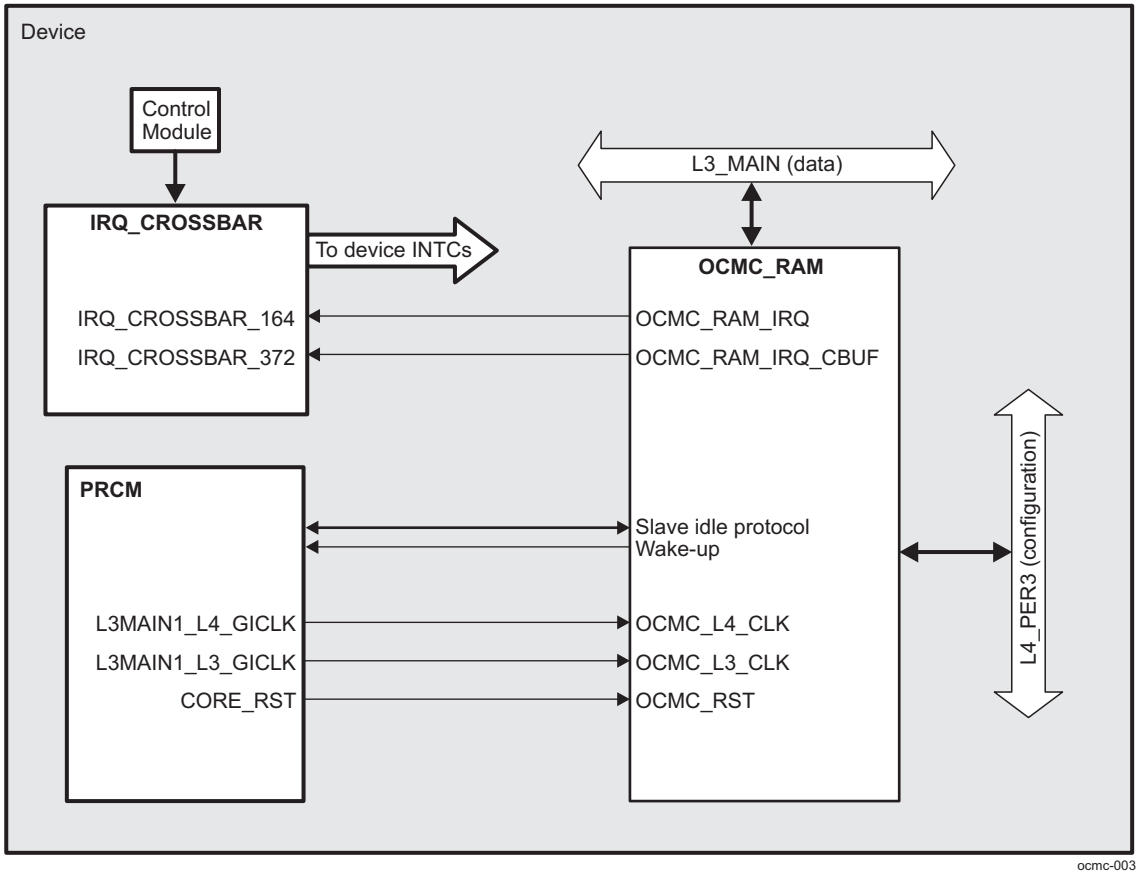


Table 10-564 through Table 10-566 summarize the integration of the OCMC_RAM in the device.

Table 10-564. OCMC_RAM Integration Attributes

Module Instance	Attributes		
	Power Domain	Wake-Up Capability	Interconnect
OCMC_RAM	PD_COREAON	Yes	L3_MAIN
			L4_PER3

Table 10-565. OCMC_RAM Clocks and Resets

Clocks				
Module Instance	**Destination Signal Name**	**Source Signal Name**	**Source**	**Description**
OCMC_RAM	OCMC_L3_CLK	L3MAIN1_L3_GICLK	PRCM	Clock used to drive and receive data over the L3 data bus. This is also the processing clock of the OCM Controller and the SRAM. With this clock all internal data transfers are clocked.
	OCMC_L4_CLK	L3MAIN1_L4_GICLK	PRCM	Clock used to drive and receive data over the L4_PER3 configuration bus. This clock should run at half the OCMC_L3_CLK clock rate.
Resets				
Module Instance	**Destination Signal Name**	**Source Signal Name**	**Source**	**Description**
OCMC_RAM	OCMC_RST	CORE_RST	PRCM	Reset signal for the OCMC_RAM

Table 10-566. OCMC_RAM Hardware Requests

Interrupt Requests				
Module Instance	**Source Signal Name**	**Destination IRQ_CROSSBAR Input**	**Default Mapping**	**Description**
OCMC_RAM	OCMC_RAM_IRQ	IRQ_CROSSBAR_164	-	First OCMC_RAM interrupt request. This IRQ source signal is not mapped by default to any device INTC.
	OCMC_RAM_IRQ_CBUF	IRQ_CROSSBAR_372	-	Second OCMC_RAM interrupt request. This IRQ source signal is not mapped by default to any device INTC.

NOTE: The "**Default Mapping**" column in Table 10-566 *OCMC_RAM Hardware Requests* shows the default mapping of module IRQ source signals. These IRQ source signals can also be mapped to other lines of each device Interrupt controller through the IRQ_CROSSBAR module.
For more information about the IRQ_CROSSBAR module, see Section 13.4.6.4, *IRQ_CROSSBAR Module Functional Description*, in Chapter 13, *Control Module*.
For more information about the device interrupt controllers, see Chapter 12, *Interrupt Controllers*.

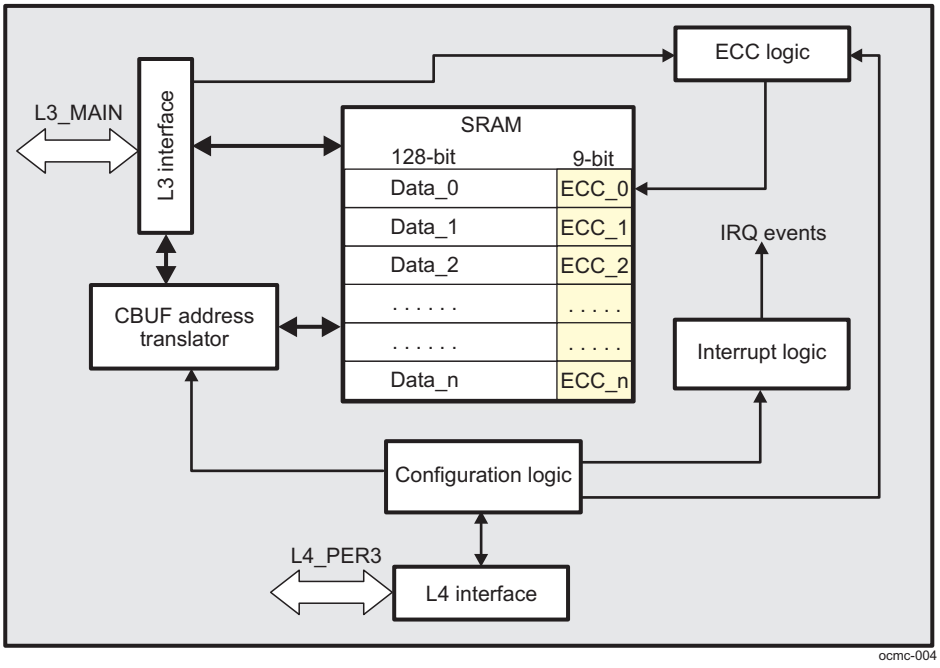
NOTE: For description of the interrupt sources, see Section 10.5.3.4, *Interrupt Requests*.

10.5.3 OCM Subsystem Functional Desctiption

10.5.3.1 Block Diagram

Figure 10-67 shows the OCMC block diagram.

Figure 10-67. OCMC Block Diagram



The module is composed by the following main blocks:

- L3 interface used for data transactions
- L4 interface used for configuration
- CBUF address translator which converts the L3_MAIN VBUF addresses into SRAM addresses
- ECC logic to support single error correction and double error detection
- SRAM used for storring data and the corresponding for each 128-bit word ECC
- Interrupt logic used for generating interrupt requests
- Configuration logic in which reside the registers used for configuring the OCM controller operation modes

10.5.3.2 Resets

The OCMC_RST is the reset signal for the OCM controller which asynchronously resets the whole internal logic of the controller, including all configuration registers. It does not reset the SRAM. In addition, the OCMC_SYSCONFIG_RST[0] SW_RST bit provides a software way to reset the OCM controller. In this case all of its internal logic is reset except the configuration registers accessible through the L4_PER3.

10.5.3.3 Clock Management

The OCM controller complies with the PRCM slave idle protocol. The OCMC_L3_CLK and OCMC_L4_CLK clocks are gated based on the values loaded in the OCMC_SYSCONFIG_PM[3:2] IDLEMODE bit field.

10.5.3.4 Interrupt Requests

The OCM controller generates two interrupt requests, OCMC_RAM_IRQ and OCMC_RAM_IRQ_CBUF.

The OCMC_RAM_IRQ line is associated with the following registers:

- INTR0_STATUS_RAW_SET - interrupt raw status register
- INTR0_STATUS_ENABLED_CLEAR - interrupt status register
- INTR0_ENABLE_SET - interrupt enable register
- INTR0_ENABLE_CLEAR - interrupt disable register

The OCMC_RAM_IRQ_CBUF line is associated with the following registers:

- INTR1_STATUS_RAW_SET - interrupt raw status register
- INTR1_STATUS_ENABLED_CLEAR - interrupt status register
- INTR1_ENABLE_SET - interrupt enable (event unmask) register
- INTR1_ENABLE_CLEAR - interrupt disable (event mask) register

Both the register groups previously described have identical bits. When for example, only one event occurs one or the two IRQ lines will be asserted depending on the masks applied to the INTR0_ENABLE_SET and INTR1_ENABLE_SET registers. When for example, a short frame detection event occurs and if both the INTR0_ENABLE_SET[14] CBUF_SHORT_FRAME_DETECT_FOUND and INTR1_ENABLE_SET[14] CBUF_SHORT_FRAME_DETECT_FOUND bits are set to 0x1, then both the OCMC_RAM_IRQ and OCMC_RAM_IRQ_CBUF lines are asserted. If only the INTR0_ENABLE_SET[14] CBUF_SHORT_FRAME_DETECT_FOUND bit is set to 0x1, then only the OCMC_RAM_IRQ line is asserted and the OCMC_RAM_IRQ_CBUF line remains inactive (deasserted). Thus depending on the mask applied one IRQ line of the OCM controller can be associated only with CBUF events for example, and the other one IRQ line can be associated with ECC events. In other words, two unique interrupts can be provided.

Table 10-567 lists the interrupt events which can assert the two interrupt lines of the OCM controller.

The OCM controller asserts each of its interrupt lines only if the interrupts are enabled by setting to 0x1 the corresponding bits in the INTR0_ENABLE_SET/INTR1_ENABLE_SET register. These interrupts can be disabled by setting to 0x1 the corresponding bits in the INTR0_ENABLE_CLEAR/INTR1_ENABLE_CLEAR register. After the interrupt has been serviced the corresponding status flag must be cleared by software. This is done by setting to 0x1 the corresponding bit in the INTR0_STATUS_ENABLED_CLEAR/INTR1_STATUS_ENABLED_CLEAR register which also

clears the corresponding bit in the INTR0_STATUS_RAW_SET/INTR1_STATUS_RAW_SET register. The status flags in the INTR0_STATUS_RAW_SET/INTR1_STATUS_RAW_SET register are set even if the corresponding interrupt is disabled as opposed to those in the INTR0_STATUS_ENABLED_CLEAR/INTR1_STATUS_ENABLED_CLEAR register, which are set only if the corresponding interrupt is enabled. An interrupt is also generated by the OCM controller, if certain bit in the INTR0_STATUS_RAW_SET/INTR1_STATUS_RAW_SET register is set to 0x1 and the corresponding interrupt is enabled through the INTR0_ENABLE_SET/INTR1_ENABLE_SET register. This is useful when user software debugging is performed. Additionally, even if interrupts are not enabled, certain status bit in the INTR0_STATUS_RAW_SET/INTR1_STATUS_RAW_SET register can be cleared by setting to 0x1 the corresponding bit in the INTR0_STATUS_ENABLED_CLEAR/INTR1_STATUS_ENABLED_CLEAR register.

For additional details regarding the CBUF related events, see Section 10.5.3.11, *CBUF Mode Error Handling*.

Table 10-567. OCMC_RAM Events

Event Flag	Event Mask	Description
INTR0_STATUS_RAW_SET[14] CBUF_SHORT_FRAME_DETECT_FOUND/INTR1_STATUS_RAW_SET[14] CBUF_SHORT_FRAME_DETECT_FOUND and INTR0_STATUS_ENABLED_CLEAR[14] CBUF_SHORT_FRAME_DETECT_FOUND/INTR1_STATUS_ENABLED_CLEAR[14] CBUF_SHORT_FRAME_DETECT_FOUND	INTR0_ENABLE_SET[14] CBUF_SHORT_FRAME_DETECT_FOUND/INTR1_ENABLE_SET[14] CBUF_SHORT_FRAME_DETECT_FOUND and INTR0_ENABLE_CLEAR[14] CBUF_SHORT_FRAME_DETECT_FOUND/INTR1_ENABLE_CLEAR[14] CBUF_SHORT_FRAME_DETECT_FOUND	This bit indicates a short frame detection. It is set if at least one of the bits in the STATUS_CBUF_SHORT_FRAME_DETECT register is set to 0x1.
INTR0_STATUS_RAW_SET[13] CBUF_UNDERFLOW_ERR_FOUND/INTR1_STATUS_RAW_SET[13] CBUF_UNDERFLOW_ERR_FOUND and INTR0_STATUS_ENABLED_CLEAR[13] CBUF_UNDERFLOW_ERR_FOUND/INTR1_STATUS_ENABLED_CLEAR[13] CBUF_UNDERFLOW_ERR_FOUND	INTR0_ENABLE_SET[13] CBUF_UNDERFLOW_ERR_FOUND/INTR1_ENABLE_SET[13] CBUF_UNDERFLOW_ERR_FOUND and INTR0_ENABLE_CLEAR[13] CBUF_UNDERFLOW_ERR_FOUND/INTR1_ENABLE_CLEAR[13] CBUF_UNDERFLOW_ERR_FOUND	Indicates CBUF underflow detction. This bit is set if at least one of the bits in the STATUS_CBUF_UNDERFLOW register is set to 0x1.
INTR0_STATUS_RAW_SET[12] CBUF_OVERFLOW_WRAP_ERR_FOUND/INTR1_STATUS_RAW_SET[12] CBUF_OVERFLOW_WRAP_ERR_FOUND and INTR0_STATUS_ENABLED_CLEAR[12] CBUF_OVERFLOW_WRAP_ERR_FOUND/INTR1_STATUS_ENABLED_CLEAR[12] CBUF_OVERFLOW_WRAP_ERR_FOUND	INTR0_ENABLE_SET[12] CBUF_OVERFLOW_WRAP_ERR_FOUND/INTR1_ENABLE_SET[12] CBUF_OVERFLOW_WRAP_ERR_FOUND and INTR0_ENABLE_CLEAR[12] CBUF_OVERFLOW_WRAP_ERR_FOUND/INTR1_ENABLE_CLEAR[12] CBUF_OVERFLOW_WRAP_ERR_FOUND	Indicates Cbuf_Overflow_Wrap event
INTR0_STATUS_RAW_SET[11] CBUF_OVERFLOW_MID_ERR_FOUND/INTR1_STATUS_RAW_SET[11] CBUF_OVERFLOW_MID_ERR_FOUND and INTR0_STATUS_ENABLED_CLEAR[11] CBUF_OVERFLOW_MID_ERR_FOUND/INTR1_STATUS_ENABLED_CLEAR[11] CBUF_OVERFLOW_MID_ERR_FOUND	INTR0_ENABLE_SET[11] CBUF_OVERFLOW_MID_ERR_FOUND/INTR1_ENABLE_SET[11] CBUF_OVERFLOW_MID_ERR_FOUND and INTR0_ENABLE_CLEAR[11] CBUF_OVERFLOW_MID_ERR_FOUND/INTR1_ENABLE_CLEAR[11] CBUF_OVERFLOW_MID_ERR_FOUND	Indicates Cbuf_Overflow_Mid event
INTR0_STATUS_RAW_SET[10] CBUF_READ_SEQUENCE_ERR_FOUND/INTR1_STATUS_RAW_SET[10] CBUF_READ_SEQUENCE_ERR_FOUND and INTR0_STATUS_ENABLED_CLEAR[10] CBUF_READ_SEQUENCE_ERR_FOUND/INTR1_STATUS_ENABLED_CLEAR[10] CBUF_READ_SEQUENCE_ERR_FOUND	INTR0_ENABLE_SET[10] CBUF_READ_SEQUENCE_ERR_FOUND/INTR1_ENABLE_SET[10] CBUF_READ_SEQUENCE_ERR_FOUND and INTR0_ENABLE_CLEAR[10] CBUF_READ_SEQUENCE_ERR_FOUND/INTR1_ENABLE_CLEAR[10] CBUF_READ_SEQUENCE_ERR_FOUND	This flag indicates that at least one CBUF read address is not incrementing in raster scan order(1).

(1) The LAST_ILLEGAL_OCMC_ADDR register stores the last illegal L3_MAIN address caused the assertion of this event.

Table 10-567. OCMC_RAM Events (continued)

Event Flag	Event Mask	Description
INTR0_STATUS_RAW_SET[9] CBUF_VBUF_READ_START_ERR_FOUND/INTR1_STATUS_RAW_SET[9] CBUF_VBUF_READ_START_ERR_FOUND and INTR0_STATUS_ENABLED_CLEAR[9] CBUF_VBUF_READ_START_ERR_FOUND/INTR1_STATUS_ENABLED_CLEAR[9] CBUF_VBUF_READ_START_ERR_FOUND	INTR0_ENABLE_SET[9] CBUF_VBUF_READ_START_ERR_FOUND/INTR1_ENABLE_SET[9] CBUF_VBUF_READ_START_ERR_FOUND and INTR0_ENABLE_CLEAR[9] CBUF_VBUF_READ_START_ERR_FOUND/INTR1_ENABLE_CLEAR[9] CBUF_VBUF_READ_START_ERR_FOUND	This flag indicates when at least one of the CBUF read accesses does not start at the VBUF start address(1).
INTR0_STATUS_RAW_SET[8] CBUF_READ_OUT_OF_RANGE_ERR_FOUND /INTR1_STATUS_RAW_SET[8] CBUF_READ_OUT_OF_RANGE_ERR_FOUND and INTR0_STATUS_ENABLED_CLEAR[8] CBUF_READ_OUT_OF_RANGE_ERR_FOUND /INTR1_STATUS_ENABLED_CLEAR[8] CBUF_READ_OUT_OF_RANGE_ERR_FOUND	INTR0_ENABLE_SET[8] CBUF_READ_OUT_OF_RANGE_ERR_FOUND /INTR1_ENABLE_SET[8] CBUF_READ_OUT_OF_RANGE_ERR_FOUND and INTR0_ENABLE_CLEAR[8] CBUF_READ_OUT_OF_RANGE_ERR_FOUND /INTR1_ENABLE_CLEAR[8] CBUF_READ_OUT_OF_RANGE_ERR_FOUND	This flag indicates when at least one of the CBUF read addresses is out of the CBUF address range(1).
INTR0_STATUS_RAW_SET[7] CBUF_WRITE_SEQUENCE_ERR_FOUND/INTR1_STATUS_RAW_SET[7] CBUF_WRITE_SEQUENCE_ERR_FOUND and INTR0_STATUS_ENABLED_CLEAR[7] CBUF_WRITE_SEQUENCE_ERR_FOUND/INTR1_STATUS_ENABLED_CLEAR[7] CBUF_WRITE_SEQUENCE_ERR_FOUND	INTR0_ENABLE_SET[7] CBUF_WRITE_SEQUENCE_ERR_FOUND/INTR1_ENABLE_SET[7] CBUF_WRITE_SEQUENCE_ERR_FOUND and INTR0_ENABLE_CLEAR[7] CBUF_WRITE_SEQUENCE_ERR_FOUND/INTR1_ENABLE_CLEAR[7] CBUF_WRITE_SEQUENCE_ERR_FOUND	This flag indicates that at least one CBUF write address is not incrementing in raster scan order(1).
INTR0_STATUS_RAW_SET[6] CBUF_VBUF_WRITE_START_ERR_FOUND/INTR1_STATUS_RAW_SET[6] CBUF_VBUF_WRITE_START_ERR_FOUND and INTR0_STATUS_ENABLED_CLEAR[6] CBUF_VBUF_WRITE_START_ERR_FOUND/INTR1_STATUS_ENABLED_CLEAR[6] CBUF_VBUF_WRITE_START_ERR_FOUND	INTR0_ENABLE_SET[6] CBUF_VBUF_WRITE_START_ERR_FOUND/INTR1_ENABLE_SET[6] CBUF_VBUF_WRITE_START_ERR_FOUND and INTR0_ENABLE_CLEAR[6] CBUF_VBUF_WRITE_START_ERR_FOUND/INTR1_ENABLE_CLEAR[6] CBUF_VBUF_WRITE_START_ERR_FOUND	This flag indicates when at least one of the CBUF write accesses does not start at the VBUF start address(1).
INTR0_STATUS_RAW_SET[5] CBUF_WR_OUT_OF_RANGE_ERR_FOUND/INTR1_STATUS_RAW_SET[5] CBUF_WR_OUT_OF_RANGE_ERR_FOUND and INTR0_STATUS_ENABLED_CLEAR[5] CBUF_WR_OUT_OF_RANGE_ERR_FOUND/INTR1_STATUS_ENABLED_CLEAR[5] CBUF_WR_OUT_OF_RANGE_ERR_FOUND	INTR0_ENABLE_SET[5] CBUF_WR_OUT_OF_RANGE_ERR_FOUND/INTR1_ENABLE_SET[5] CBUF_WR_OUT_OF_RANGE_ERR_FOUND and INTR0_ENABLE_CLEAR[5] CBUF_WR_OUT_OF_RANGE_ERR_FOUND/INTR1_ENABLE_CLEAR[5] CBUF_WR_OUT_OF_RANGE_ERR_FOUND	This flag indicates when at least one of the CBUF write addresses is out of the CBUF address range(1).
INTR0_STATUS_RAW_SET[4] CBUF_VIRTUAL_ADDR_ERR_FOUND/INTR1_STATUS_RAW_SET[4] CBUF_VIRTUAL_ADDR_ERR_FOUND and INTR0_STATUS_ENABLED_CLEAR[4] CBUF_VIRTUAL_ADDR_ERR_FOUND/INTR1_STATUS_ENABLED_CLEAR[4] CBUF_VIRTUAL_ADDR_ERR_FOUND	INTR0_ENABLE_SET[4] CBUF_VIRTUAL_ADDR_ERR_FOUND/INTR1_ENABLE_SET[4] CBUF_VIRTUAL_ADDR_ERR_FOUND and INTR0_ENABLE_CLEAR[4] CBUF_VIRTUAL_ADDR_ERR_FOUND/INTR1_ENABLE_CLEAR[4] CBUF_VIRTUAL_ADDR_ERR_FOUND	This flag indicates when a virtual address error is detected. This is a general interrupt event which is not associated with any specific CBUF(1).
INTR0_STATUS_RAW_SET[3] OUT_OF_RANGE_ERR_FOUND/INTR1_STATUS_RAW_SET[3] OUT_OF_RANGE_ERR_FOUND and INTR0_STATUS_ENABLED_CLEAR[3] OUT_OF_RANGE_ERR_FOUND/INTR1_STATUS_ENABLED_CLEAR[3] OUT_OF_RANGE_ERR_FOUND	INTR0_ENABLE_SET[3] OUT_OF_RANGE_ERR_FOUND/INTR1_ENABLE_SET[3] OUT_OF_RANGE_ERR_FOUND and INTR0_ENABLE_CLEAR[3] OUT_OF_RANGE_ERR_FOUND/INTR1_ENABLE_CLEAR[3] OUT_OF_RANGE_ERR_FOUND	General interrupt for an access made with illegal VBUF address(1).

Table 10-567. OCMC_RAM Events (continued)

Event Flag	Event Mask	Description
INTR0_STATUS_RAW_SET[2] ADDR_ERR_FOUND/INTR1_STATUS_RAW_SET[2] ADDR_ERR_FOUND and INTR0_STATUS_ENABLED_CLEAR[2] ADDR_ERR_FOUND/INTR1_STATUS_ENABLED_CLEAR[2] ADDR_ERR_FOUND	INTR0_ENABLE_SET[2] ADDR_ERR_FOUND/INTR1_ENABLE_SET[2] ADDR_ERR_FOUND and INTR0_ENABLE_CLEAR[2] ADDR_ERR_FOUND/INTR1_ENABLE_CLEAR[2] ADDR_ERR_FOUND	This status flag is set when the value of the STATUS_ERROR_CNT[23:20] ADDR_ERROR_CNT bit field reaches the threshold counter value configured through the CFG_OCMC_ECC_ERROR[23:20] CFG_ADDR_ERR_CNT_MAX bit field. The status remains asserted until the counter is not cleared. This is done by setting to 0x1 the CFG_OCMC_ECC_CLEAR_HIST[2] CLEAR_ADDR_ERR_CNT bit. If the counter is not cleared another interrupt is re-issued.
INTR0_STATUS_RAW_SET[1] DED_ERR_FOUND/INTR1_STATUS_RAW_SET[1] DED_ERR_FOUND and INTR0_STATUS_ENABLED_CLEAR[1] DED_ERR_FOUND/INTR1_STATUS_ENABLED_CLEAR[1] DED_ERR_FOUND	INTR0_ENABLE_SET[1] DED_ERR_FOUND/INTR1_ENABLE_SET[1] DED_ERR_FOUND and INTR0_ENABLE_CLEAR[1] DED_ERR_FOUND/INTR1_ENABLE_CLEAR[1] DED_ERR_FOUND	This status flag is set when the value of the STATUS_ERROR_CNT[19:16] DED_ERROR_CNT bit field reaches the threshold counter value configured through the CFG_OCMC_ECC_ERROR[19:16] CFG_DED_CNT_MAX bit field. The status remains asserted until the counter is not cleared. This is done by setting to 0x1 the CFG_OCMC_ECC_CLEAR_HIST[1] CLEAR_DED_ERR_CNT bit. If the counter is not cleared another interrupt is re-issued.
INTR0_STATUS_RAW_SET[0] SEC_ERR_FOUND/INTR1_STATUS_RAW_SET[0] SEC_ERR_FOUND and INTR0_STATUS_ENABLED_CLEAR[0] SEC_ERR_FOUND/INTR1_STATUS_ENABLED_CLEAR[0] SEC_ERR_FOUND	INTR0_ENABLE_SET[0] SEC_ERR_FOUND/INTR1_ENABLE_SET[0] SEC_ERR_FOUND and INTR0_ENABLE_CLEAR[0] SEC_ERR_FOUND/INTR1_ENABLE_CLEAR[0] SEC_ERR_FOUND	This status flag is set when the value of the STATUS_ERROR_CNT[15:0] SEC_ERROR_CNT bit field reaches the threshold counter value configured through the CFG_OCMC_ECC_ERROR[15:0] CFG_SEC_CNT_MAX bit field. The status remains asserted until the counter is not cleared. This is done by setting to 0x1 the CFG_OCMC_ECC_CLEAR_HIST[0] CLEAR_SEC_ERR_CNT bit. If the counter is not cleared another interrupt is re-issued.

10.5.3.5 OCM Subsystem Memory Regions

The SRAM associated with the OCMC_RAM is accessible through the L3_MAIN interconnect. The start address is 0x4030 0000 and the end address is 0x4037 FFFF. That is address space of 512KiB. The configuration registers of the OCMC_RAM are accessible through the L4_PER3 starting at address 0x4880 4000.

10.5.3.6 OCM Controller Modes Of Operation

The OCM controller supports four modes of operation. Each of these modes is selected through the CFG_OCMC_ECC[2:0] CFG_OCMC_MODE bit field. The four supported modes are the following:

- Non-ECC mode (Data Access) - Accesses to the SRAM are non-ECC-enabled and an ECC is not calculated.
- Non-ECC mode (Code Access) - The L3_MAIN address is mapped to the SRAM memory space where the ECC code is stored. This mode allows read and write access to the 9-bit ECC word associated with each 128-bit data word. This mode is used for test purposes.
- Full-ECC mode - Accesses to the SRAM are ECC-enabled and a 9-bit ECC is calcualted for each 128 bits of data.
- Block-ECC mode - A 9-bit ECC is calcualted only for a 128KiB block of the SRAM. Accesses outside the 128KiB ECC-enabled block are non-ECC data acceses. In other words, a 9-bit ECC will be calculated not for each 128-bit data word of the whole SRAM space, but for each 128-bit data word within the boundaries of this 128KiB block of the SRAM. The selection of a 128KiB ECC-enabled block is done using bits[19:0] of the CFG_OCMC_ECC_MEM_BLK register. Each bit specifies which 128KiB block of the SRAM is ECC-enabled. 0x1 is the active value for each bit. In addition, more than one 128KiB ECC-enabled block can be selected.

The default mode of operation for the OCM controller is the non-ECC data access mode.

10.5.3.7 ECC Associated FIFOs

There are three FIFOs used when ECC mode is enabled. Each FIFO is four level deep. The FIFOs are the following:

- Address FIFO for single errors also referred to as SEC FIFO
- Address FIFO for double errors also referred to as DED FIFO
- Address FIFO for address errors also referred to as ADDRERR FIFO

The SEC FIFO stores the SRAM addresses at which a single error is detected. The FIFO is able to store up to four unique addresses of the single errors occured. If there are more than four single errors associated with unique addresses, then only the first four addresses are stored in the FIFO. In case of occurrence of multiple correctable errors, the addresses are stored in the order the errors occurred. The SEC FIFO can be optionally configured to store all addresses of the single errors occured including also the repeated addresses. This is done by setting to 0x0 the CFG_OCMC_ECC_ERROR[24] CFG_DISCARD_DUP_ADDR bit. The SEC FIFO can be cleared by setting to 0x1 the CFG_OCMC_ECC_CLEAR_HIST[0] CLEAR_SEC_ERR_CNT bit or by reding FIFO's content one by one through the STATUS_SEC_ERROR_TRACE[17:0] ADDRESS_128BIT bit field which points to the SEC FIFO. In addition, the STATUS_SEC_ERROR_TRACE[18] VALID bit shows whether the FIFO is empty or not. A value of 0x1 means that the SEC FIFO is not empty and valid address can be read.

The DED FIFO stores the SRAM addresses when a double error is detected. The FIFO is able to store up to four unique addresses, but it can also be configured to store the repeated addresses by setting to 0x0 the CFG_OCMC_ECC_ERROR[24] CFG_DISCARD_DUP_ADDR bit. The STATUS_DED_ERROR_TRACE[17:0] ADDRESS_128BIT bit field points to the DED FIFO. The STATUS_DED_ERROR_TRACE[18] VALID bit shows whether the FIFO is empty or not. The DED FIFO can be cleared by setting to 0x1 the CFG_OCMC_ECC_CLEAR_HIST[1] CLEAR_DED_ERR_CNT bit or by reading its content one by one through the STATUS_DED_ERROR_TRACE[17:0] ADDRESS_128BIT.

The ADDRERR FIFO stores the SRAM addresses where the single error occured is an addres error. The STATUS_ADDR_TRANSLATION_ERROR_TRACE[17:0] ADDRESS_128BIT bit field points to the ADDRERR FIFO. The STATUS_ADDR_TRANSLATION_ERROR_TRACE[18] VALID bit shows whether the FIFO is empty or not. The ADDRERR FIFO can be cleared by setting to 0x1 the CFG_OCMC_ECC_CLEAR_HIST[2] CLEAR_ADDR_ERR_CNT bit or by reading its content one by one through the STATUS_ADDR_TRANSLATION_ERROR_TRACE[17:0] ADDRESS_128BIT pointer. The ADDRERR FIFO is also able to store up to four unique addresses and can also be configured to store repeated addresses by setting to 0x0 the CFG_OCMC_ECC_ERROR[24] CFG_DISCARD_DUP_ADDR bit.

10.5.3.8 ECC Counters And Corrected Bit Distribution Register

There are three counters which are used to count different types of errors occured when the ECC mode is enabled. The counters are the following:

- SEC Counter - STATUS_ERROR_CNT[15:0] SEC_ERROR_CNT. This is the counter for the single errors occured. It keeps track of all single errors. That is, even for those with error addresses that have already been logged.
- DED Counter - STATUS_ERROR_CNT[19:16] DED_ERROR_CNT. This is the counter for the double error detections.
- ADDRERR Counter - STATUS_ERROR_CNT[23:20] ADDR_ERROR_CNT. This is the counter for the address errors found when a single error occurs. That is, when the single error is an addres error.

When a single error in the 128-bit data word occurs and this error is corrected, the OCM controller indicates for the corrected bit of this 128-bit word by setting to 0x1 a corresponding bit in the corrected bit distribution register. This is a 128-bit register composed by the following registers:

- STATUS_SEC_ERROR_DISTR_0[31:0] SEC_BIT_ERROR_FOUND - Bits [31:0] of the corrected 128-bit data word.
- STATUS_SEC_ERROR_DISTR_1[31:0] SEC_BIT_ERROR_FOUND - Bits [63:32] of the corrected 128-bit data word.
- STATUS_SEC_ERROR_DISTR_2[31:0] SEC_BIT_ERROR_FOUND - Bits [95:64] of the corrected 128-bit data word.
- STATUS_SEC_ERROR_DISTR_3[31:0] SEC_BIT_ERROR_FOUND - Bits [127:96] of the corrected 128-bit data word.

When a single error in the ECC itself occurs, the OCM controller indicates for this error by setting to 0x1 a corresponding bit in the STATUS_SEC_ERROR_DISTR_4[7:0] SEC_ECC_CODE_ERROR_FOUND bit field. The parity bit is not associated with this bit field.

10.5.3.9 ECC Support

The ECC mode is enabled through the CFG_OCMC_ECC[2:0] CFG_OCMC_MODE bit field. When enabled a 9-bit Hamming ECC is calculated and stored for each consecutive 128-bit block of the SRAM. The ECC is calculated based on a codeword constructed by concatenating the 128 bits of data with the address bits A21 through A4 of the L3_MAIN. The ECC generated is Hamming(155,146) code and has a Hamming distance of 4. The OCM controller uses this code to validate the content of the SRAM, to correct a single bit error that occurs within the 128-bit boundary or to determine if a non-correctable error has occurred within the 128-bit boundary.

During write transactions and when ECC is enabled, for every 128-bit data word, the ECC is calculated and stored in a 9-bit field of the SRAM associated only with the address where that 128-bit data word is written. If an initiator performs a write transaction which is less than 128 bits or it is non-128-bit aligned transaction, then the content of all 128 bits is read, the new sub-quanta of data is inserted and the ECC is calculated based on all 128 bits.

During read transactions and when ECC is enabled, the ECC is calcualated based on the memory address (bits A21 through A4) and the 128-bit data word read from the SRAM. This ECC is then compared to the ECC stored at the address when the data was written to the SRAM. If the two ECCs are matching then the data is transferred to the requesting initiator without further exceptions. If the two ECCs are not matching then a check is made to determine if the error is correctable or not. If the error is not correctable, that is a double error, then the address of the erroneous word is stored in the DED FIFO, the STATUS_ERROR_CNT[19:16] DED_ERROR_CNT counter is incremented by 1 and the double error flag is asserted. The double error flag is indicated by the INTR0_STATUS_RAW_SET[1] DED_ERR_FOUND/INTR1_STATUS_RAW_SET[1] DED_ERR_FOUND bits, but its assertion depends on the threshold configured through the CFG_OCMC_ECC_ERROR[19:16] CFG_DED_CNT_MAX bit field. For more information see, Table 10-567, *OCMC_RAM Events*.

In case of a single error, the OCM controller first determines whether the erroneous bit is in the data, address or the ECC itself. In this case the following applies:

- If the single error is located in the data read, then the erroneous bit is corrected and the data is passed to the requesting initiator. The erroneous bit is corrected also in the SRAM location by the auto re-write

feature of the OCM controller if this feature is enabled by setting to 0x1 the CFG_OCMC_ECC[3] CFG_ECC_SEC_AUTO_CORRECT bit field. In addition, the address of the corrected data word is pushed to the SEC FIFO and the STATUS_ERROR_CNT[15:0] SEC_ERROR_CNT counter is incremented by 1. The single error flag is asserted too, but only if the value of the STATUS_ERROR_CNT[15:0] SEC_ERROR_CNT counter reaches the threshold configured by the CFG_OCMC_ECC_ERROR[15:0] CFG_SEC_CNT_MAX bit field. The single error flag is indicated by the INTR0_STATUS_RAW_SET[0] SEC_ERR_FOUND/INTR1_STATUS_RAW_SET[0] SEC_ERR_FOUND bits. For more information see, Table 10-567, *OCMC_RAM Events*.

- If the single error is located in the address portion of the quanta, then the data is passed to the requesting initiator unchanged and the STATUS_ERROR_CNT[23:20] ADDR_ERROR_CNT and STATUS_ERROR_CNT[15:0] SEC_ERROR_CNT counters are incremented by 1. The address associated with this error is stored in the ADDRERR FIFO and the address error flag is asserted. This flag is indicated by the INTR0_STATUS_RAW_SET[2] ADDR_ERR_FOUND/INTR1_STATUS_RAW_SET[2] ADDR_ERR_FOUND bits, but its assertion depends on the threshold configured through the CFG_OCMC_ECC_ERROR[23:20] CFG_ADDR_ERR_CNT_MAX bit field. For more information see, Table 10-567, *OCMC_RAM Events*.
- If the single error is located in the ECC itself, then the data is passed to the initiator unchanged but the STATUS_ERROR_CNT[15:0] SEC_ERROR_CNT counter is incremented by 1 and the address associated with this error is stored in the SEC FIFO.

Table 10-568 summarizes the actions taken by the OCM controller for the different error types.

Table 10-568. OCMC Error Handling In Case Of Different Error Types

Error Type	Data Returned	SRAM Data/ECC Update	SEC Counter	SEC FIFO	DED Counter	DED FIFO	ADDRERR Counter	ADDRERR FIFO	Error Bit Distribution
Single data error	Corrected	Data corrected/ECC re-generated	Incremented by 1	Address written	-	-	-	-	The content of the corrected bit distribution register is updated
Single address error	Unchanged	ECC re-generated	Incremented by 1	-	-	-	Incremented by 1	Address written	-
Single ECC error	Unchanged	ECC re-generated	Incremented by 1	Address written	-	-	-	-	The content of the STATUS_SEC_ERROR_DISTR_4[7:0] SEC_ECC_CODE_ERROR_FOUND bit field is updated
Double error	Unchanged	-	-	-	Incremented by 1	Address written	-	-	-

If ECC has to be used, before performing SRAM read/write operations the following steps should be performed:

- Configuring the ECC registers
- Enabling the ECC
- Initializing the SRAM with data
- Clearing the status flags

Table 10-569 describes in detail the steps to be performed.

Table 10-569. OCMC ECC Configuration

Step	Associated Register/ Bit Field	Value
(Optional) Enable the error response on L3_MAIN for a non-correctable error	CFG_OCMC_ECC[4] CFG_ECC_ERR_SRESP_EN	0x1
(Optional) Enable data auto correction in case of a single error	CFG_OCMC_ECC[3] CFG_ECC_SEC_AUTO_CORRECT	0x1
Enable the memory blocks to be ECC protected, if ECC block mode is used	CFG_OCMC_ECC_MEM_BLK[19:0] CFG_ECC_ENABLED_128K_BLK	0x-

Table 10-569. OCMC ECC Configuration (continued)

Step	Associated Register/ Bit Field	Value
(Optional) Configure the address errors count needed for triggering an interrupt	CFG_OCMC_ECC_ERROR[23:20] CFG_ADDR_ERR_CNT_MAX	0x-
(Optional) Configure the DED count needed for triggering an interrupt	CFG_OCMC_ECC_ERROR[19:16] CFG_DED_CNT_MAX	0x-
(Optional) Configure the SEC count needed for triggering an interrupt	CFG_OCMC_ECC_ERROR[15:0] CFG_SEC_CNT_MAX	0x-
Enable ECC	CFG_OCMC_ECC[2:0] CFG_OCMC_MODE	0x2 for full ECC or 0x3 for block ECC mode
Initialize the memory with data		
Clear the error counters in the STATUS_ERROR_CNT register	CFG_OCMC_ECC_CLEAR_HIST[2:0]	0x7
Clear the SEC bit distribution history from the following registers:	CFG_OCMC_ECC_CLEAR_HIST[3] CLEAR_SEC_BIT_DISTR	0x1
• STATUS_SEC_ERROR_DISTR_0		
• STATUS_SEC_ERROR_DISTR_1		
• STATUS_SEC_ERROR_DISTR_2		
• STATUS_SEC_ERROR_DISTR_3		
• STATUS_SEC_ERROR_DISTR_4		

10.5.3.10 Circular Buffer (CBUF) Support

The OCM controller provides up to 12 programmable circular buffers that are mapped to virtual video frames to support sliced based on-the-fly video frame processing. Each circular buffer must be programmed with the following:

- Unique virtual frame start address - configured through CBUF_i_VBUF_START_ADDR[31:4] ADDR.
- Unique virtual frame end address - configured through CBUF_i_VBUF_END_ADDR[31:4] ADDR.
- SRAM start address - configured through CBUF_i_OCMC_START_ADDR[31:4] ADDR.
- SRAM size allocated for that circular buffer - configured through CBUF_i_OCMC_BUF_SIZE[19:4] BUF_SIZE.

The circular buffer mode is enabled when the CFG_OCMC_CBUF_EN[0] CBUF_MODE_EN bit is set to 0x1. In addition, to enable certain circular buffer the corresponding bit in the CFG_OCMC_CBUF_EN[27:16] bit field must be set to 0x1. For example, to enable CBUF_0 both the CFG_OCMC_CBUF_EN[0] CBUF_MODE_EN and CFG_OCMC_CBUF_EN[16] CBUF_EN_0 bits must be set to 0x1.

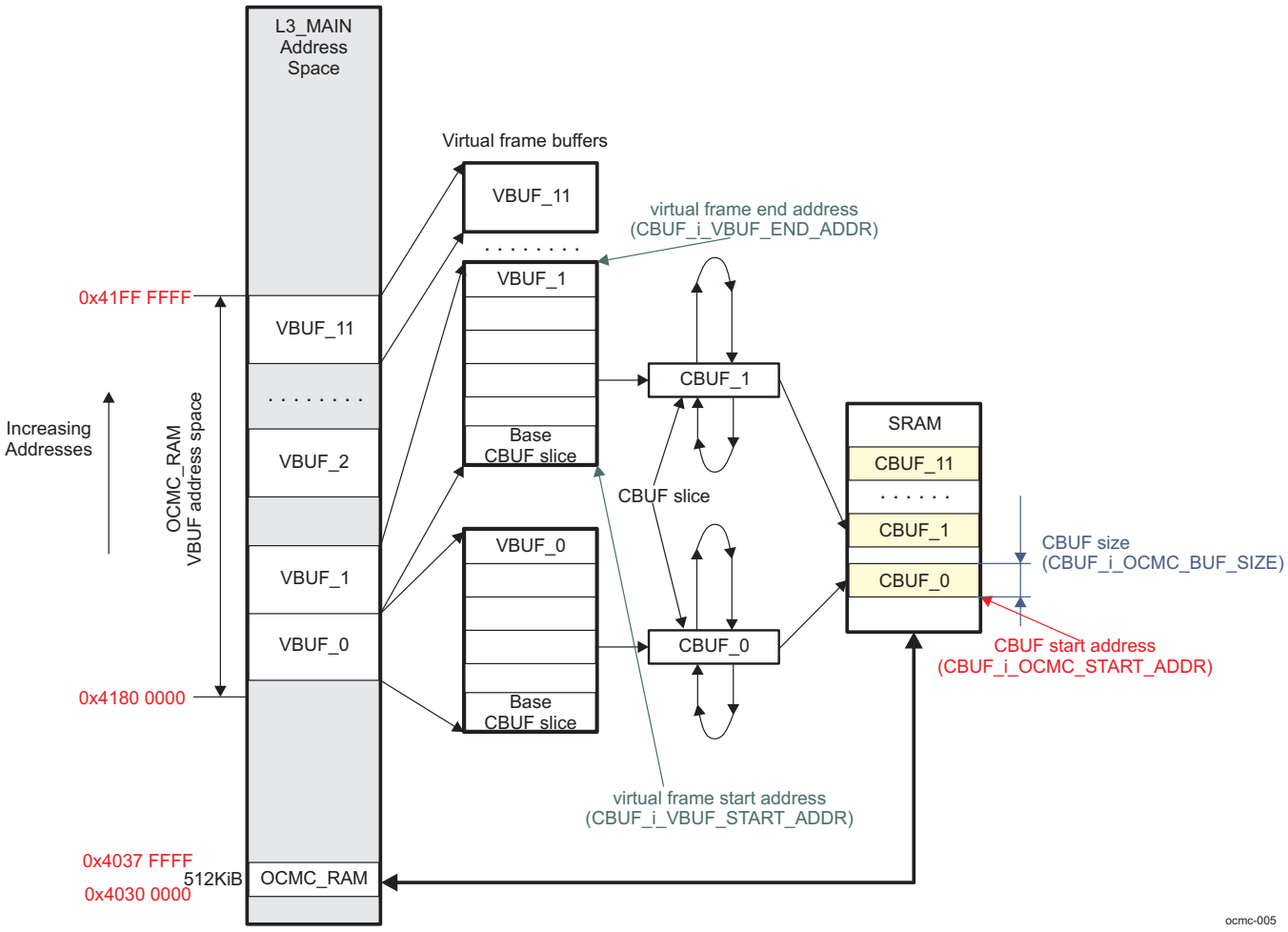
The data transfers are made to the virtual frame addresses. The OCM controller translates these virtual addresses to SRAM addresses allocated for the CBUF. The address translation is done on-the-fly without leading to any additional latency and therefore has no performance impact on the overall operation of the OCM controller.

The L3_MAIN address range associated with the virtual frame addresses is the following:

- For OCMC_RAM the start address is 0x4180 0000 and the end address is 0x41FF FFFF.

When an initiator performs an access to these L3_MAIN addresses, these virtual frame addresses will be translated automatically by the OCM controller into SRAM addresses. Figure 10-68 shows the address mapping of the virtual frame addresses to CBUF addresses.

Figure 10-68. VBUF to CBUF Address Mapping



It must be taken into account that the start address of a virtual frame always begins at the start of the CBUF. Since the virtual frame buffer size is not necessarily to be a multiple of the CBUF size, the last VBUF write location may not be the last line of the CBUF. As the CBUF read and write pointers are independently managed the initiators that are writing to and reading from the CBUF should be coordinated in order to avoid overflow and underflow conditions.

10.5.3.11 CBUF Mode Error Handling

For each L3_MAIN write or read access associated with the virtual frame buffer, the OCM controller performs various address error checks to prevent illegal CBUF accesses from causing false overflow and underflow conditions. Once the L3_MAIN VBUF address is determined as valid, the OCM controller further checks for overflow and underflow conditions to properly handle the access to the CBUF during each one of these conditions and to notify for error condition an external host.

For more information regarding all CBUF events see, Table 10-567, *OCMC_RAM Events*.

10.5.3.11.1 VBUF Address Not Mapped to a CBUF Memory Space

The L3_MAIN VBUF address should be mapped to a valid CBUF region. If this address is mapped to none of the configured CBUF regions, then the OCM controller generates a virtual address error or out of range CBUF address error event. It is possible both virtual address error and out of range CBUF address error to be generated. The L3_MAIN VBUF address caused these two events can be read through the LAST_ILLEGAL_OCMC_ADDR[31:0] ADDR bit field.

Virtual address error is generated when the L3_MAIN VBUF address is lower than the virtual frame start address. This is indicated by asserting the INTR0_STATUS_RAW_SET[4] CBUF_VIRTUAL_ADDR_ERR_FOUND/INTR1_STATUS_RAW_SET[4] CBUF_VIRTUAL_ADDR_ERR_FOUND bit.

The out of range CBUF address error is generated when the L3_MAIN VBUF address is greater than the virtual frame end address. This error is an indication of an unexpected change in the frame size or a long frame and should be handled immediately to prevent further memory corruption. The out of range CBUF address error can be caused when performing write or read access. The out of range CBUF write address error is indicated by the STATUS_CBUF_WR_OUT_OF_RANGE_ERR[11:0] CBUF_ERR bit field. The out of range CBUF read address error is indicated by the STATUS_CBUF_RD_OUT_OF_RANGE_ERROR[11:0] CBUF_ERR bit field. Each bit in these registers is associated only with one CBUF. Bit 0 corresponds to CBUF_0, bit 1 corresponds to CBUF_1 and so on. A value of 0x1 for each bit means that there is an error (CBUF address is out of CBUF range) for the corresponding CBUF. Writing 0x1 for each bit clears the status.

The INTR0_STATUS_RAW_SET[5] CBUF_WR_OUT_OF_RANGE_ERR_FOUND/INTR1_STATUS_RAW_SET[5] CBUF_WR_OUT_OF_RANGE_ERR_FOUND bit is asserted if at least one of the bits in the STATUS_CBUF_WR_OUT_OF_RANGE_ERR[11:0] CBUF_ERR bit field is set to 0x1.

The INTR0_STATUS_RAW_SET[8] CBUF_READ_OUT_OF_RANGE_ERR_FOUND/INTR1_STATUS_RAW_SET[8] CBUF_READ_OUT_OF_RANGE_ERR_FOUND bit is asserted if at least one of the bits in the STATUS_CBUF_RD_OUT_OF_RANGE_ERROR[11:0] CBUF_ERR bit field is set to 0x1.

10.5.3.11.2 VBUF Access Not Starting At The Base Address

After (hardware of software) reset or on a new frame start condition, a frame write or read access for any VBUF must start from its virtual frame start address (CBUF_i_VBUF_START_ADDR[31:4] ADDR). If this is not met the OCM controller ignors all accesses which don't start at address CBUF_i_VBUF_START_ADDR[31:4] ADDR and also generates an interrupt. The STATUS_CBUF_WR_VBUF_START_ERR[11:0] CBUF_ERR register contains the status bits of this interrupt which are associated with frame write access. The STATUS_CBUF_VBUF_RD_START_ERROR[11:0] CBUF_ERR register contains the status bits associated with frame read access. Each bit in these two registers is associated only with one CBUF. A value of 0x1 for each bit means that the read/write access does not start at the VBUF start address.

While this error condition exists, no updates to the CBUF channel status registers are made. An exception to this constraint is when the last read/write access is still within the base CBUF slice. The enforcing of a frame starting at VBUF starting address is not done in this case to allow random access within the same CBUF. However, this will be flagged and handled as a short frame occurrence.

The INTR0_STATUS_RAW_SET[6] CBUF_VBUF_WRITE_START_ERR_FOUND/INTR1_STATUS_RAW_SET[6] CBUF_VBUF_WRITE_START_ERR_FOUND bit is asserted if at least one of the bits in the STATUS_CBUF_WR_VBUF_START_ERR[11:0] CBUF_ERR bit field is set to 0x1.

The INTR0_STATUS_RAW_SET[9] CBUF_VBUF_READ_START_ERR_FOUND/INTR1_STATUS_RAW_SET[9] CBUF_VBUF_READ_START_ERR_FOUND bit is asserted if at least one of the bits in the STATUS_CBUF_VBUF_RD_START_ERROR[11:0] CBUF_ERR bit field is set to 0x1.

10.5.3.11.3 Illegal Address Change Between Two Same Type Accesses

The OCM controller requires both read and write accesses to a virtual frame buffer to be performed in the positive raster scan order, that is, successive write or read accesses should be to a higher address except on a new frame start for which the address jumps back down to the beginning of the virtual frame buffer. The OCM controller checks to make sure that a virtual address meets one of the following conditions:

- It is within the same CBUF slice as the last access of the same type (read or write)
- It is within the next CBUF slice of the last access of the same type (read or write)
- The virtual address is to the VBUF frame start address

Based on these constraints, the OCM controller generates an write/read address sequence error interrupt and invalidates the access, that is, writes are disabled and reads return unknown data, if the following two conditions are met:

- The current access address is not to a current or next CBUF slice space
- The current access address is not to the base of the virtual frame buffer (the first CBUF slice)

The random accesses within the current CBUF slice are permited to allow some variations in the line width of the captured video which often happens with weak video signal conditions.

To avoid illegal CBUF slice backward switching, the modules capturing the input video and sending it to the CBUF of the OCM controller should limit the maximum width of a line width to be less than the stride size (in pixels). This guarantees that a start of a new line does not point back to the previous CBUF slice.

If a write or read address sequence error event is detected on an access, the OCM controller will simply invalidate the accesses and will not update the CBUF status registers.

The bits in the STATUS_CBUF_WR_ADDR_SEQ_ERROR[11:0] CBUF_ERR bit field indicate write address sequence error event.

The bits in the STATUS_CBUF_RD_ADDR_SEQ_ERROR[11:0] CBUF_ERR bit field indicate read address sequence error event.

In both the registers each bit is associated only with one CBUF.

The INTR0_STATUS_RAW_SET[7] CBUF_WRITE_SEQUENCE_ERR_FOUND/INTR1_STATUS_RAW_SET[7] CBUF_WRITE_SEQUENCE_ERR_FOUND bit is asserted if at least one of the bits in the STATUS_CBUF_WR_ADDR_SEQ_ERROR[11:0] CBUF_ERR bit field is set to 0x1.

The INTR0_STATUS_RAW_SET[10] CBUF_READ_SEQUENCE_ERR_FOUND/INTR1_STATUS_RAW_SET[10] CBUF_READ_SEQUENCE_ERR_FOUND bit is asserted if at least one of the bits in the STATUS_CBUF_RD_ADDR_SEQ_ERROR[11:0] CBUF_ERR bit field is set to 0x1.

10.5.3.11.4 Illegal Frame SIze (Short Frame Detection)

If a frame write access for any VBUF begins from its virtual frame start address without a CBUF software reset and the last write address is not the virtual frame end address or is in the last CBUF slice, then the previous frame is considered to be short. This includes the case where a new frame write happens twice without a new frame read in between, which is an indication of an extremely short frame. When a short frame event is detected, the OCM controller generates a short frame detection interrupt and temporarily disables overflow checking on the write access, including also the current access, until a frame read happens in order to avoid false overflow indication on a suspended read side following the short frame detection.

If the last write address is in the last CBUF slice the short frame detection can be ignored if the CFG_OCMC_CBUF_ERR_HANDLER[1] SHORT_FRAME_PREV_EOF_SEL bit is set to 0x1.

The STATUS_CBUF_SHORT_FRAME_DETECT[11:0] CBUF_ERR bit field has status flags to indicate a short frame detection for each CBUF. When a short frame condition occurs the corresponding bit is set to 0x1.

The INTR0_STATUS_RAW_SET[14] CBUF_SHORT_FRAME_DETECT_FOUND/INTR1_STATUS_RAW_SET[14] CBUF_SHORT_FRAME_DETECT_FOUND bit is asserted if at least one of the bits in the STATUS_CBUF_SHORT_FRAME_DETECT[11:0] CBUF_ERR bit field is set to 0x1.

In addition, the short frame detection enabling/disabling is controlled by the CFG_OCMC_CBUF_ERR_HANDLER[0] SHORT_FRAME_DETECT_CHECK_EN bit.

10.5.3.11.5 CBUF Overflow

For each valid CBUF write access, the write address is compared to the last read address for an overflow condition. If an overflow is detected, the corresponding CBUF overflow status bit in the STATUS_CBUF_OVERFLOW_MID or STATUS_CBUF_OVERFLOW_WRAP registers is set to 0x1 and an interrupt is generated. This interrupt condition exists until the corresponding overflow status bit is cleared.

There are two types of overflow conditions referred to as Cbuf_overflow_wrap and Cbuf_overflow_mid.

Cbuf_overflow_wrap occurs in case at least one of the following two:

- VBUF write address – VBUF frame start address < CBUF size
- VBUF write address – VBUF frame start address > VBUF last read address – Current VBUF start address

Cbuf_overflow_mid occurs in case at least one of the following two:

- VBUF write address – VBUF frame start address > CBUF size
- VBUF write address – VBUF last read address > CBUF size

If an overflow condition is detected on a CBUF write access, the OCM controller generates a CBUF overflow (Cbuf_overflow_wrap or Cbuf_overflow_mid) interrupt and enter an overflow error handling state. In this state, the OCM controller do the following in the default error handling configuration:

- When a Cbuf_Overflow_Wrap condition occurs (write buffer wraparound with read from previous frame still remaining), writes are disabled but reads are serviced normally. This mechanism preserves the back end of the previous frame and discards the new frame which is already showing signs of overflowing. Writes will be re-enabled automatically on write to the virtual frame start address which indicates the next frame after the overflowed frame has begun. The overflow (and underflow) check for this CBUF will be disabled until detection of write to/read from the virtual frame start address. The module which reads should not start reading the next frame (which is being discarded) or stop reading the frame if the read of this new frame has already started. This mechanism allows both write and read sides to re-synchronize on the next frame after one-frame drop.
- When a Cbuf_Overflow_Mid condition occurs, writes and reads are serviced normally allowing the read side to catch up to the write side. The overflow (and underflow) check will be disabled until first a write to the virtual frame start address and then a read from the virtual frame start address is detected. This mechanism allows the frame with a “momentary” overflow condition to be saved. The reader side has an option to stop reading and discarding the data by just resetting its CBUF read pointer at the next frame start, if the read side is too far back.

These steps are taken to minimize the frame loss due to an overflow condition. To support the previously described behavior but to allow software to override the default error handling behavior the following controls are available:

- CFG_OCMC_CBUF_ERR_HANDLER[7:6] OVERFLOW_CHECK_REENABLE_SEL
- CFG_OCMC_CBUF_ERR_HANDLER[5:4] OVERFLOW_WRITE_HANDLER_SEL

In addition, the overflow check can be enabled or disabled through the CFG_OCMC_CBUF_ERR_HANDLER[2] OVERFLOW_ERR_CHECK_EN bit.

10.5.3.11.6 CBUF Underflow

According to the CBUF operation a read request can only be made when there are enough number of video lines written to the CBUF. Therefore, there should not be any underflow conditions due to normal processing speed variations. An underflow condition occurs when the VBUF read address is greater than the last VBUF write address. To detect an abnormal failure in the read/write mechanism which could cause an underflow condition, each valid CBUF read access is checked for an underflow condition and if detected, an underflow interrupt is generated for the selected CBUF. The STATUS_CBUF_UNDERFLOW[11:0] CBUF_ERR bit field has status flags indicating when underflow occurs. This interrupt condition exists until the corresponding underflow status bit in the STATUS_CBUF_UNDERFLOW register is cleared.

If an underflow is detected on a CBUF read access, the OCM controller generates an underflow interrupt and continues to perform reads and writes normally, but the return data will not be correct while the underflow condition exists.

To avoid a false underflow detection at the end of a frame (due to external decoders sending short last line or due to interlaced input video having different number of lines in even/odd fields), the CFG_OCMC_CBUF_ERR_HANDLER[8] UNDERFLOW_LAST_CBUF_SLICE_DISABLE bit can be set to 0x1 which disables the underflow checking in the last CBUF slice.

The INTR0_STATUS_RAW_SET[13] CBUF_UNDERFLOW_ERR_FOUND/INTR1_STATUS_RAW_SET[13] CBUF_UNDERFLOW_ERR_FOUND bit is asserted if at least one of the bits in the STATUS_CBUF_UNDERFLOW[11:0] CBUF_ERR bit field is set to 0x1.

In addition, the underflow check can be enabled or disabled through the CFG_OCMC_CBUF_ERR_HANDLER[3] UNDERFLOW_ERR_CHECK_EN bit.

10.5.3.12 Status Reporting

The OCM controller keeps a history of last valid write and read addresses for each CBUF. The CBUF_k_LAST_WR_ADDR register stores the address for the last valid write and the CBUF_k_LAST_RD_ADDR register stores the address for the last valid read access.

The OCMC_ECC generates also a general VBUF mode error event for an access made with illegal VBUF address. The INTR0_STATUS_RAW_SET[3] OUT_OF_RANGE_ERR_FOUND/INTR1_STATUS_RAW_SET[3] OUT_OF_RANGE_ERR_FOUND bit indicates for this event.

10.5.4 OCM Subsystem Register Manual

10.5.4.1 OCM Subsystem Instance Summary

Table 10-570. OCM Subsystem Instance Summary

Module Name	Module Base Address	Size
OCMC_RAM	0x4880 4000	4KiB

10.5.4.2 OCM Subsystem Registers

10.5.4.2.1 OCM Subsystem Register Summary

Table 10-571. OCM Subsystem Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	OCMC_RAM Base Address
OCMC_ECC_PID	R	32	0x0000 0000	0x4880 4000
OCMC_SYSCONFIG_PM	RW	32	0x0000 0004	0x4880 4004
OCMC_SYSCONFIG_RST	RW	32	0x0000 0008	0x4880 4008
OCMC_MEM_SIZE_READ	R	32	0x0000 000C	0x4880 400C
INTR0_STATUS_RAW_SET	RW	32	0x0000 0040	0x4880 4040
INTR0_STATUS_ENABLED_CLEAR	RW	32	0x0000 0044	0x4880 4044
INTR0_ENABLE_SET	RW	32	0x0000 0048	0x4880 4048
INTR0_ENABLE_CLEAR	RW	32	0x0000 004C	0x4880 404C
OCMC_INTR0_EOI	RW	32	0x0000 0050	0x4880 4050
INTR1_STATUS_RAW_SET	RW	32	0x0000 0060	0x4880 4060
INTR1_STATUS_ENABLED_CLEAR	RW	32	0x0000 0064	0x4880 4064
INTR1_ENABLE_SET	RW	32	0x0000 0068	0x4880 4068

Table 10-571. OCM Subsystem Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	OCMC_RAM Base Address
INTR1_ENABLE_CLEAR	RW	32	0x0000 006C	0x4880 406C
OCMC_INTR1_EOI	RW	32	0x0000 0070	0x4880 4070
CFG_OCMC_ECC	RW	32	0x0000 0080	0x4880 4080
CFG_OCMC_ECC_MEM_BLK	RW	32	0x0000 0084	0x4880 4084
CFG_OCMC_ECC_ERROR	RW	32	0x0000 0088	0x4880 4088
CFG_OCMC_ECC_CLEAR_HIST	RW	32	0x0000 008C	0x4880 408C
STATUS_ERROR_CNT	R	32	0x0000 0090	0x4880 4090
STATUS_SEC_ERROR_TRACE	R	32	0x0000 0094	0x4880 4094
STATUS_DED_ERROR_TRACE	R	32	0x0000 0098	0x4880 4098
STATUS_ADDR_TRANSLATION_ERROR_TRACE	R	32	0x0000 009C	0x4880 409C
STATUS_SEC_ERROR_DISTR_0	R	32	0x0000 00A0	0x4880 40A0
STATUS_SEC_ERROR_DISTR_1	R	32	0x0000 00A4	0x4880 40A4
STATUS_SEC_ERROR_DISTR_2	R	32	0x0000 00A8	0x4880 40A8
STATUS_SEC_ERROR_DISTR_3	R	32	0x0000 00AC	0x4880 40AC
STATUS_SEC_ERROR_DISTR_4	R	32	0x0000 00B0	0x4880 40B0
CFG_OCMC_CBUF_EN	RW	32	0x0000 0200	0x4880 4200
CFG_OCMC_CBUF_RESET	RW	32	0x0000 0204	0x4880 4204
CFG_OCMC_CBUF_ERR_HANDLER	RW	32	0x0000 0208	0x4880 4208
STATUS_CBUF_WR_OUT_OF_RANGE_ERR	RW	32	0x0000 020C	0x4880 420C
STATUS_CBUF_WR_VBUF_START_ERR	RW	32	0x0000 0210	0x4880 4210
STATUS_CBUF_WR_ADDR_SEQ_ERROR	RW	32	0x0000 0214	0x4880 4214
STATUS_CBUF_RD_OUT_OF_RANGE_ERROR	RW	32	0x0000 0218	0x4880 4218
STATUS_CBUF_VBUF_RD_START_ERROR	RW	32	0x0000 021C	0x4880 421C
STATUS_CBUF_RD_ADDR_SEQ_ERROR	RW	32	0x0000 0220	0x4880 4220
STATUS_CBUF_OVERFLOW_MID	RW	32	0x0000 0224	0x4880 4224
STATUS_CBUF_OVERFLOW_WRAP	RW	32	0x0000 0228	0x4880 4228
STATUS_CBUF_UNDERFLOW	RW	32	0x0000 022C	0x4880 422C
STATUS_CBUF_SHORT_FRAME_DETECT	RW	32	0x0000 0230	0x4880 4230
CBUF_i_VBUF_START_ADDR(1)	RW	32	0x0000 0240 + (i*16)	0x4880 4240 + (i*16)
CBUF_i_VBUF_END_ADDR(1)	RW	32	0x0000 0244 + (i*16)	0x4880 4244 + (i*16)
CBUF_i_OCMC_START_ADDR(1)	RW	32	0x0000 0248 + (i*16)	0x4880 4248 + (i*16)
CBUF_i_OCMC_BUF_SIZE(1)	RW	32	0x0000 024C + (i*16)	0x4880 424C + (i*16)
CBUF_k_LAST_WR_ADDR(2)	R	32	0x0000 0300 + (k*8)	0x4880 4300 + (k*8)
CBUF_k_LAST_RD_ADDR(2)	R	32	0x0000 0304 + (k*8)	0x4880 4304 + (k*8)
LAST_ILLEGAL_OCMC_ADDR	R	32	0x0000 0360	0x4880 4360

(1) i = 0 to 11
(2) k = 0 to 11

10.5.4.2.2 OCM Subsystem Register Description

Table 10-572. OCMC_ECC_PID

Address Offset	0x0000 0000		
Physical Address	0x4880 4000	**Instance**	OCMC_RAM
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REVISION																															

Bits	Field Name	Description	Type	Reset
31:0	REVISION	TI internal data	R	0x-

Table 10-573. Register Call Summary for Register OCMC_ECC_PID

On-Chip Memory (OCM) Subsystem

- OCM Subsystem Register Summary: [2]

Table 10-574. OCMC_SYSCONFIG_PM

Address Offset	0x0000 0004		
Physical Address	0x4880 4004	**Instance**	OCMC_RAM
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												IDLEMODE		RESERVED	

Bits	Field Name	Description	Type	Reset
31:4	RESERVED		R	0x0
3:2	IDLEMODE	Configuration of the local target state management mode. By definition, target can handle read/write transaction as long as it is out of IDLE state. 0x0: Force-idle mode: local target's idle state follows (acknowledges) the system's idle requests unconditionally, that is, regardless of the IP module's internal requirements. Backup mode, for debug only. 0x1: No-idle mode: local target never enters idle state. Backup mode, for debug only 0x2: Smart-idle mode: local target's idle state eventually follows (acknowledges) the system's idle requests, depending on the IP module's internal requirements. IP module shall not generate IRQ-request-related wakeup events. 0x3: Smart-idle wakeup-capable mode: local target's idle state eventually follows (acknowledges) the system's idle requests, depending on the IP module's internal requirements. IP module may generate IRQ-request-related wakeup events when in idle state.	RW	0x2
1:0	RESERVED		R	0x0

Table 10-575. Register Call Summary for Register OCMC_SYSCONFIG_PM

On-Chip Memory (OCM) Subsystem
- Clock Management: [0]
- OCM Subsystem Register Summary: [3]

Table 10-576. OCMC_SYSCONFIG_RST

Address Offset	0x0000 0008		
Physical Address	0x4880 4008	**Instance**	OCMC_RAM
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															SW_RST

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	SW_RST	Software reset of the OCM controller configuration and history logic (does not reset L4 interface) 0x0: Normal operation (OCM controller is not reset) 0x1: Reset the OCM controller (except its registers). This bit must be set back to 0x0 to resume the normal operation of the OCM Controller.	RW	0x0

Table 10-577. Register Call Summary for Register OCMC_SYSCONFIG_RST

On-Chip Memory (OCM) Subsystem
- Resets: [0]
- OCM Subsystem Register Summary: [3]

Table 10-578. OCMC_MEM_SIZE_READ

Address Offset	0x0000 000C		
Physical Address	0x4880 400C	**Instance**	OCMC_RAM
Description	This register provides the status of the OCM Controller configuration.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																VBUF_ADDR_MSB				RESERVED		MEM_CBUF_ENABLE	MEM_ECC_ENABLE	RESERVED					MEM_SIZE_128K_CNT		

Bits	Field Name	Description	Type	Reset
31:17	RESERVED		R	0x0
16:12	VBUF_ADDR_MSB	This bit field returns the MSB bit of the valid VBUF address range. The default value of 23 means that the valid VBUF address range is from 0x8000 0000 to 0x80FF FFFF	R	0x-
11:10	RESERVED		R	0x0
9	MEM_CBUF_ENABLE	Indicates whether CBUF is supported or not. 0x0: CBUF not supported 0x1: CBUF supported	R	0x-
8	MEM_ECC_ENABLE	Indicates whether ECC is supported or not. 0x0: ECC not supported 0x1: ECC supported	R	0x-
7:5	RESERVED		R	0x0
4:0	MEM_SIZE_128K_CNT	This bit field indicates how many 128KiB memory blocks are present in the SRAM. Access beyond the memory size reported in the MEM_SIZE_128K_CNT bit field results in an address error interrupt. 0x1: One 128KiB memory block 0x2: Two 128KiB memory blocks ... 0x14: 20 memory blocks of 128KiB	R	0x-

Table 10-579. Register Call Summary for Register OCMC_MEM_SIZE_READ

On-Chip Memory (OCM) Subsystem

- OCM Subsystem Register Summary: [2]
- OCM Subsystem Register Description: [3][4]

Table 10-580. INTR0_STATUS_RAW_SET

Address Offset	0x0000 0040		
Physical Address	0x4880 4040	**Instance**	OCMC_RAM
Description	This register contains the raw interrupt status. Read indicates RAW interrupt status (0=inactive, 1=active). Writing 1 will SET the corresponding raw status bit (soft interrupt set). Writing 0 has no effect.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																CBUF_SHORT_FRAME_DETECT_FOUND	CBUF_UNDERFLOW_ERR_FOUND	CBUF_OVERFLOW_WRAP_ERR_FOUND	CBUF_OVERFLOW_MID_ERR_FOUND	CBUF_READ_SEQUENCE_ERR_FOUND	CBUF_VBUF_READ_START_ERR_FOUND	CBUF_READ_OUT_OF_RANGE_ERR_FOUND	CBUF_WRITE_SEQUENCE_ERR_FOUND	CBUF_VBUF_WRITE_START_ERR_FOUND	CBUF_WR_OUT_OF_RANGE_ERR_FOUND	CBUF_VIRTUAL_ADDR_ERR_FOUND	OUT_OF_RANGE_ERR_FOUND	ADDR_ERR_FOUND	DED_ERR_FOUND	SEC_ERR_FOUND

Bits	Field Name	Description	Type	Reset
31:15	RESERVED		R	0x0
14	CBUF_SHORT_FRAME_DETECT_FOUND	CBUF detected short frame.	RW	0x0
13	CBUF_UNDERFLOW_ERR_FOUND		RW	0x0
12	CBUF_OVERFLOW_WRAP_ERR_FOUND		RW	0x0
11	CBUF_OVERFLOW_MID_ERR_FOUND		RW	0x0
10	CBUF_READ_SEQUENCE_ERR_FOUND		RW	0x0
9	CBUF_VBUF_READ_START_ERR_FOUND		RW	0x0
8	CBUF_READ_OUT_OF_RANGE_ERR_FOUND		RW	0x0
7	CBUF_WRITE_SEQUENCE_ERR_FOUND		RW	0x0
6	CBUF_VBUF_WRITE_START_ERR_FOUND		RW	0x0
5	CBUF_WR_OUT_OF_RANGE_ERR_FOUND		RW	0x0
4	CBUF_VIRTUAL_ADDR_ERR_FOUND		RW	0x0
3	OUT_OF_RANGE_ERR_FOUND		RW	0x0
2	ADDR_ERR_FOUND		RW	0x0
1	DED_ERR_FOUND		RW	0x0
0	SEC_ERR_FOUND		RW	0x0

Table 10-581. Register Call Summary for Register INTR0_STATUS_RAW_SET

On-Chip Memory (OCM) Subsystem

- Interrupt Requests: [0][1][2][3][4][5][6][7][8][9][10][11][12][13][14][15][16][17][18][19]
- ECC Support: [20][21][22]
- VBUF Address Not Mapped to a CBUF Memory Space: [23][24][25]
- VBUF Access Not Starting At The Base Address: [26][27]
- Illegal Address Change Between Two Same Type Accesses: [28][29]
- Illegal Frame SIze (Short Frame Detection): [30]
- CBUF Underflow: [31]
- Status Reporting: [32]
- OCM Subsystem Register Summary: [35]

Table 10-582. INTR0_STATUS_ENABLED_CLEAR

Address Offset	0x0000 0044		
Physical Address	0x4880 4044	**Instance**	OCMC_RAM
Description	Read indicates ENABLED interrupt status (0=inactive, 1=active). Writing 1 will CLEAR the corresponding enabled status bit. Writing 0 has no effect.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																	CBUF_SHORT_FRAME_DETECT_FOUND	CBUF_UNDERFLOW_ERR_FOUND	CBUF_OVERFLOW_WRAP_ERR_FOUND	CBUF_OVERFLOW_MID_ERR_FOUND	CBUF_READ_SEQUENCE_ERR_FOUND	CBUF_VBUF_READ_START_ERR_FOUND	CBUF_READ_OUT_OF_RANGE_ERR_FOUND	CBUF_WRITE_SEQUENCE_ERR_FOUND	CBUF_VBUF_WRITE_START_ERR_FOUND	CBUF_WR_OUT_OF_RANGE_ERR_FOUND	CBUF_VIRTUAL_ADDR_ERR_FOUND	OUT_OF_RANGE_ERR_FOUND	ADDR_ERR_FOUND	DED_ERR_FOUND	SEC_ERR_FOUND

Bits	Field Name	Description	Type	Reset
31:15	RESERVED		R	0x0
14	CBUF_SHORT_FRAME_DETECT_FOUND	CBUF detected short frame	RW W1toClr	0x0
13	CBUF_UNDERFLOW_ERR_FOUND		RW W1toClr	0x0
12	CBUF_OVERFLOW_WRAP_ERR_FOUND		RW W1toClr	0x0
11	CBUF_OVERFLOW_MID_ERR_FOUND		RW W1toClr	0x0
10	CBUF_READ_SEQUENCE_ERR_FOUND		RW W1toClr	0x0
9	CBUF_VBUF_READ_START_ERR_FOUND		RW W1toClr	0x0
8	CBUF_READ_OUT_OF_RANGE_ERR_FOUND		RW W1toClr	0x0
7	CBUF_WRITE_SEQUENCE_ERR_FOUND		RW W1toClr	0x0
6	CBUF_VBUF_WRITE_START_ERR_FOUND		RW W1toClr	0x0
5	CBUF_WR_OUT_OF_RANGE_ERR_FOUND		RW W1toClr	0x0
4	CBUF_VIRTUAL_ADDR_ERR_FOUND		RW W1toClr	0x0
3	OUT_OF_RANGE_ERR_FOUND		RW W1toClr	0x0
2	ADDR_ERR_FOUND		RW W1toClr	0x0
1	DED_ERR_FOUND		RW W1toClr	0x0
0	SEC_ERR_FOUND		RW W1toClr	0x0

Table 10-583. Register Call Summary for Register INTR0_STATUS_ENABLED_CLEAR

On-Chip Memory (OCM) Subsystem

- Interrupt Requests: [0][1][2][3][4][5][6][7][8][9][10][11][12][13][14][15][16][17][18]
- OCM Subsystem Register Summary: [21]

Table 10-584. INTR0_ENABLE_SET

Address Offset	0x0000 0048		
Physical Address	0x4880 4048	**Instance**	OCMC_RAM
Description	Read indicates interrupt enable (0=disabled, 1=enabled) Writing 1 will set the corresponding interrupt enable bit. Writing 0 has no effect. Interrupt_enable_set		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																	CBUF_SHORT_FRAME_DETECT_FOUND	CBUF_UNDERFLOW_ERR_FOUND	CBUF_OVERFLOW_WRAP_ERR_FOUND	CBUF_OVERFLOW_MID_ERR_FOUND	CBUF_READ_SEQUENCE_ERR_FOUND	CBUF_VBUF_READ_START_ERR_FOUND	CBUF_READ_OUT_OF_RANGE_ERR_FOUND	CBUF_WRITE_SEQUENCE_ERR_FOUND	CBUF_VBUF_WRITE_START_ERR_FOUND	CBUF_WR_OUT_OF_RANGE_ERR_FOUND	CBUF_VIRTUAL_ADDR_ERR_FOUND	OUT_OF_RANGE_ERR_FOUND	ADDR_ERR_FOUND	DED_ERR_FOUND	SEC_ERR_FOUND

Bits	Field Name	Description	Type	Reset
31:15	RESERVED		R	0x0
14	CBUF_SHORT_FRAME_DETECT_FOUND	CBUF detected short frame	RW	0x0
13	CBUF_UNDERFLOW_ERR_FOUND		RW	0x0
12	CBUF_OVERFLOW_WRAP_ERR_FOUND		RW	0x0
11	CBUF_OVERFLOW_MID_ERR_FOUND		RW	0x0
10	CBUF_READ_SEQUENCE_ERR_FOUND		RW	0x0
9	CBUF_VBUF_READ_START_ERR_FOUND		RW	0x0
8	CBUF_READ_OUT_OF_RANGE_ERR_FOUND		RW	0x0
7	CBUF_WRITE_SEQUENCE_ERR_FOUND		RW	0x0
6	CBUF_VBUF_WRITE_START_ERR_FOUND		RW	0x0
5	CBUF_WR_OUT_OF_RANGE_ERR_FOUND		RW	0x0
4	CBUF_VIRTUAL_ADDR_ERR_FOUND		RW	0x0
3	OUT_OF_RANGE_ERR_FOUND		RW	0x0
2	ADDR_ERR_FOUND		RW	0x0
1	DED_ERR_FOUND		RW	0x0
0	SEC_ERR_FOUND		RW	0x0

Table 10-585. Register Call Summary for Register INTR0_ENABLE_SET

On-Chip Memory (OCM) Subsystem

- Interrupt Requests: [0][1][2][3][4][5][6][7][8][9][10][11][12][13][14][15][16][17][18][19][20]
- OCM Subsystem Register Summary: [23]

Table 10-586. INTR0_ENABLE_CLEAR

Address Offset	0x0000 004C		
Physical Address	0x4880 404C	**Instance**	OCMC_RAM
Description	Read indicates interrupt enable (0=disabled, 1=enabled) Writing 1 will clear interrupt enabled. Writing 0 has no effect. Interrupt_enable_clear		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																CBUF_SHORT_FRAME_DETECT_FOUND	CBUF_UNDERFLOW_ERR_FOUND	CBUF_OVERFLOW_WRAP_ERR_FOUND	CBUF_OVERFLOW_MID_ERR_FOUND	CBUF_READ_SEQUENCE_ERR_FOUND	CBUF_VBUF_READ_START_ERR_FOUND	CBUF_READ_OUT_OF_RANGE_ERR_FOUND	CBUF_WRITE_SEQUENCE_ERR_FOUND	CBUF_VBUF_WRITE_START_ERR_FOUND	CBUF_WR_OUT_OF_RANGE_ERR_FOUND	CBUF_VIRTUAL_ADDR_ERR_FOUND	OUT_OF_RANGE_ERR_FOUND	ADDR_ERR_FOUND	DED_ERR_FOUND	SEC_ERR_FOUND	

Bits	Field Name	Description	Type	Reset
31:15	RESERVED		R	0x0
14	CBUF_SHORT_FRAME_DETECT_FOUND	CBUF detected short frame	RW W1toClr	0x0
13	CBUF_UNDERFLOW_ERR_FOUND		RW W1toClr	0x0
12	CBUF_OVERFLOW_WRAP_ERR_FOUND		RW W1toClr	0x0
11	CBUF_OVERFLOW_MID_ERR_FOUND		RW W1toClr	0x0
10	CBUF_READ_SEQUENCE_ERR_FOUND		RW W1toClr	0x0
9	CBUF_VBUF_READ_START_ERR_FOUND		RW W1toClr	0x0
8	CBUF_READ_OUT_OF_RANGE_ERR_FOUND		RW W1toClr	0x0
7	CBUF_WRITE_SEQUENCE_ERR_FOUND		RW W1toClr	0x0
6	CBUF_VBUF_WRITE_START_ERR_FOUND		RW W1toClr	0x0
5	CBUF_WR_OUT_OF_RANGE_ERR_FOUND		RW W1toClr	0x0
4	CBUF_VIRTUAL_ADDR_ERR_FOUND		RW W1toClr	0x0

Bits	Field Name	Description	Type	Reset
3	OUT_OF_RANGE_ERR_FOUND		RW W1toClr	0x0
2	ADDR_ERR_FOUND		RW W1toClr	0x0
1	DED_ERR_FOUND		RW W1toClr	0x0
0	SEC_ERR_FOUND		RW W1toClr	0x0

Table 10-587. Register Call Summary for Register INTR0_ENABLE_CLEAR

On-Chip Memory (OCM) Subsystem

- Interrupt Requests: [0][1][2][3][4][5][6][7][8][9][10][11][12][13][14][15][16]
- OCM Subsystem Register Summary: [19]

Table 10-588. OCMC_INTR0_EOI

Address Offset	0x0000 0050		
Physical Address	0x4880 4050	**Instance**	OCMC_RAM
Description	This register contains the EOI vector.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															EOI_VECTOR

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	EOI_VECTOR		RW	0x0

Table 10-589. Register Call Summary for Register OCMC_INTR0_EOI

On-Chip Memory (OCM) Subsystem

- OCM Subsystem Register Summary: [2]

Table 10-590. INTR1_STATUS_RAW_SET

Address Offset	0x0000 0060		
Physical Address	0x4880 4060	**Instance**	OCMC_RAM
Description	This register contains the raw interrupt status. Read indicates RAW interrupt status (0=inactive, 1=active). Writing 1 will SET the corresponding raw status bit (soft interrupt set). Writing 0 has no effect.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																CBUF_SHORT_FRAME_DETECT_FOUND	CBUF_UNDERFLOW_ERR_FOUND	CBUF_OVERFLOW_WRAP_ERR_FOUND	CBUF_OVERFLOW_MID_ERR_FOUND	CBUF_READ_SEQUENCE_ERR_FOUND	CBUF_VBUF_READ_START_ERR_FOUND	CBUF_READ_OUT_OF_RANGE_ERR_FOUND	CBUF_WRITE_SEQUENCE_ERR_FOUND	CBUF_VBUF_WRITE_START_ERR_FOUND	CBUF_WR_OUT_OF_RANGE_ERR_FOUND	CBUF_VIRTUAL_ADDR_ERR_FOUND	OUT_OF_RANGE_ERR_FOUND	ADDR_ERR_FOUND	DED_ERR_FOUND	SEC_ERR_FOUND								

Bits	Field Name	Description	Type	Reset
31:15	RESERVED		R	0x0
14	CBUF_SHORT_FRAME_DETECT_FOUND	CBUF detected short frame	RW	0x0
13	CBUF_UNDERFLOW_ERR_FOUND		RW	0x0
12	CBUF_OVERFLOW_WRAP_ERR_FOUND		RW	0x0
11	CBUF_OVERFLOW_MID_ERR_FOUND		RW	0x0
10	CBUF_READ_SEQUENCE_ERR_FOUND		RW	0x0
9	CBUF_VBUF_READ_START_ERR_FOUND		RW	0x0
8	CBUF_READ_OUT_OF_RANGE_ERR_FOUND		RW	0x0
7	CBUF_WRITE_SEQUENCE_ERR_FOUND		RW	0x0
6	CBUF_VBUF_WRITE_START_ERR_FOUND		RW	0x0
5	CBUF_WR_OUT_OF_RANGE_ERR_FOUND		RW	0x0
4	CBUF_VIRTUAL_ADDR_ERR_FOUND		RW	0x0
3	OUT_OF_RANGE_ERR_FOUND		RW	0x0
2	ADDR_ERR_FOUND		RW	0x0
1	DED_ERR_FOUND		RW	0x0
0	SEC_ERR_FOUND		RW	0x0

Table 10-591. Register Call Summary for Register INTR1_STATUS_RAW_SET

On-Chip Memory (OCM) Subsystem

- Interrupt Requests: [0][1][2][3][4][5][6][7][8][9][10][11][12][13][14][15][16][17][18][19]
- ECC Support: [20][21][22]
- VBUF Address Not Mapped to a CBUF Memory Space: [23][24][25]
- VBUF Access Not Starting At The Base Address: [26][27]
- Illegal Address Change Between Two Same Type Accesses: [28][29]
- Illegal Frame SIze (Short Frame Detection): [30]
- CBUF Underflow: [31]
- Status Reporting: [32]
- OCM Subsystem Register Summary: [35]

Table 10-592. INTR1_STATUS_ENABLED_CLEAR

Address Offset	0x0000 0064		
Physical Address	0x4880 4064	**Instance**	OCMC_RAM
Description	Read indicates ENABLED interrupt status (0=inactive, 1=active). Writing 1 will CLEAR the corresponding enabled status bit. Writing 0 has no effect.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																	CBUF_SHORT_FRAME_DETECT_FOUND	CBUF_UNDERFLOW_ERR_FOUND	CBUF_OVERFLOW_WRAP_ERR_FOUND	CBUF_OVERFLOW_MID_ERR_FOUND	CBUF_READ_SEQUENCE_ERR_FOUND	CBUF_VBUF_READ_START_ERR_FOUND	CBUF_READ_OUT_OF_RANGE_ERR_FOUND	CBUF_WRITE_SEQUENCE_ERR_FOUND	CBUF_VBUF_WRITE_START_ERR_FOUND	CBUF_WR_OUT_OF_RANGE_ERR_FOUND	CBUF_VIRTUAL_ADDR_ERR_FOUND	OUT_OF_RANGE_ERR_FOUND	ADDR_ERR_FOUND	DED_ERR_FOUND	SEC_ERR_FOUND

Bits	Field Name	Description	Type	Reset
31:15	RESERVED		R	0x0
14	CBUF_SHORT_FRAME_DETECT_FOUND	CBUF detected short frame	RW W1toClr	0x0
13	CBUF_UNDERFLOW_ERR_FOUND		RW W1toClr	0x0
12	CBUF_OVERFLOW_WRAP_ERR_FOUND		RW W1toClr	0x0
11	CBUF_OVERFLOW_MID_ERR_FOUND		RW W1toClr	0x0
10	CBUF_READ_SEQUENCE_ERR_FOUND		RW W1toClr	0x0
9	CBUF_VBUF_READ_START_ERR_FOUND		RW W1toClr	0x0
8	CBUF_READ_OUT_OF_RANGE_ERR_FOUND		RW W1toClr	0x0
7	CBUF_WRITE_SEQUENCE_ERR_FOUND		RW W1toClr	0x0

Bits	Field Name	Description	Type	Reset
6	CBUF_VBUF_WRITE_START_ERR_FOUND		RW W1toClr	0x0
5	CBUF_WR_OUT_OF_RANGE_ERR_FOUND		RW W1toClr	0x0
4	CBUF_VIRTUAL_ADDR_ERR_FOUND		RW W1toClr	0x0
3	OUT_OF_RANGE_ERR_FOUND		RW W1toClr	0x0
2	ADDR_ERR_FOUND		RW W1toClr	0x0
1	DED_ERR_FOUND		RW W1toClr	0x0
0	SEC_ERR_FOUND		RW W1toClr	0x0

Table 10-593. Register Call Summary for Register INTR1_STATUS_ENABLED_CLEAR

On-Chip Memory (OCM) Subsystem

- Interrupt Requests: [0][1][2][3][4][5][6][7][8][9][10][11][12][13][14][15][16][17][18]
- OCM Subsystem Register Summary: [21]

Table 10-594. INTR1_ENABLE_SET

Address Offset	0x0000 0068		
Physical Address	0x4880 4068	**Instance**	OCMC_RAM
Description	Read indicates interrupt enable (0=disabled, 1=enabled) Writing 1 will set the corresponding interrupt enable bit. Writing 0 has no effect. Interrupt_enable_set		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																	CBUF_SHORT_FRAME_DETECT_FOUND	CBUF_UNDERFLOW_ERR_FOUND	CBUF_OVERFLOW_WRAP_ERR_FOUND	CBUF_OVERFLOW_MID_ERR_FOUND	CBUF_READ_SEQUENCE_ERR_FOUND	CBUF_VBUF_READ_START_ERR_FOUND	CBUF_READ_OUT_OF_RANGE_ERR_FOUND	CBUF_WRITE_SEQUENCE_ERR_FOUND	CBUF_VBUF_WRITE_START_ERR_FOUND	CBUF_WR_OUT_OF_RANGE_ERR_FOUND	CBUF_VIRTUAL_ADDR_ERR_FOUND	OUT_OF_RANGE_ERR_FOUND	ADDR_ERR_FOUND	DED_ERR_FOUND	SEC_ERR_FOUND

Bits	Field Name	Description	Type	Reset
31:15	RESERVED		R	0x0
14	CBUF_SHORT_FRAME_DETECT_FOUND	CBUF detected short frame	RW	0x0
13	CBUF_UNDERFLOW_ERR_FOUND		RW	0x0
12	CBUF_OVERFLOW_WRAP_ERR_FOUND		RW	0x0

Bits	Field Name	Description	Type	Reset
11	CBUF_OVERFLOW_MID_ERR_FOUND		RW	0x0
10	CBUF_READ_SEQUENCE_ERR_FOUND		RW	0x0
9	CBUF_VBUF_READ_START_ERR_FOUND		RW	0x0
8	CBUF_READ_OUT_OF_RANGE_ERR_FOUND		RW	0x0
7	CBUF_WRITE_SEQUENCE_ERR_FOUND		RW	0x0
6	CBUF_VBUF_WRITE_START_ERR_FOUND		RW	0x0
5	CBUF_WR_OUT_OF_RANGE_ERR_FOUND		RW	0x0
4	CBUF_VIRTUAL_ADDR_ERR_FOUND		RW	0x0
3	OUT_OF_RANGE_ERR_FOUND		RW	0x0
2	ADDR_ERR_FOUND		RW	0x0
1	DED_ERR_FOUND		RW	0x0
0	SEC_ERR_FOUND		RW	0x0

Table 10-595. Register Call Summary for Register INTR1_ENABLE_SET

On-Chip Memory (OCM) Subsystem

- Interrupt Requests: [0][1][2][3][4][5][6][7][8][9][10][11][12][13][14][15][16][17][18][19]
- OCM Subsystem Register Summary: [22]

Table 10-596. INTR1_ENABLE_CLEAR

Address Offset	0x0000 006C		
Physical Address	0x4880 406C	**Instance**	OCMC_RAM
Description	Read indicates interrupt enable (0=disabled, 1=enabled) Writing 1 will clear interrupt enabled. Writing 0 has no effect. Interrupt_enable_clear		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																CBUF_SHORT_FRAME_DETECT_FOUND	CBUF_UNDERFLOW_ERR_FOUND	CBUF_OVERFLOW_WRAP_ERR_FOUND	CBUF_OVERFLOW_MID_ERR_FOUND	CBUF_READ_SEQUENCE_ERR_FOUND	CBUF_VBUF_READ_START_ERR_FOUND	CBUF_READ_OUT_OF_RANGE_ERR_FOUND	CBUF_WRITE_SEQUENCE_ERR_FOUND	CBUF_VBUF_WRITE_START_ERR_FOUND	CBUF_WR_OUT_OF_RANGE_ERR_FOUND	CBUF_VIRTUAL_ADDR_ERR_FOUND	OUT_OF_RANGE_ERR_FOUND	ADDR_ERR_FOUND	DED_ERR_FOUND	SEC_ERR_FOUND

Bits	Field Name	Description	Type	Reset
31:15	RESERVED		R	0x0
14	CBUF_SHORT_FRAME_DETECT_FOUND	CBUF detected short frame	RW W1toClr	0x0
13	CBUF_UNDERFLOW_ERR_FOUND		RW W1toClr	0x0
12	CBUF_OVERFLOW_WRAP_ERR_FOUND		RW W1toClr	0x0
11	CBUF_OVERFLOW_MID_ERR_FOUND		RW W1toClr	0x0
10	CBUF_READ_SEQUENCE_ERR_FOUND		RW W1toClr	0x0
9	CBUF_VBUF_READ_START_ERR_FOUND		RW W1toClr	0x0
8	CBUF_READ_OUT_OF_RANGE_ERR_FOUND		RW W1toClr	0x0
7	CBUF_WRITE_SEQUENCE_ERR_FOUND		RW W1toClr	0x0
6	CBUF_VBUF_WRITE_START_ERR_FOUND		RW W1toClr	0x0
5	CBUF_WR_OUT_OF_RANGE_ERR_FOUND		RW W1toClr	0x0
4	CBUF_VIRTUAL_ADDR_ERR_FOUND		RW W1toClr	0x0
3	OUT_OF_RANGE_ERR_FOUND		RW W1toClr	0x0
2	ADDR_ERR_FOUND		RW W1toClr	0x0
1	DED_ERR_FOUND		RW W1toClr	0x0
0	SEC_ERR_FOUND		RW W1toClr	0x0

Table 10-597. Register Call Summary for Register INTR1_ENABLE_CLEAR

On-Chip Memory (OCM) Subsystem

- Interrupt Requests: [0][1][2][3][4][5][6][7][8][9][10][11][12][13][14][15][16]
- OCM Subsystem Register Summary: [19]

Table 10-598. OCMC_INTR1_EOI

Address Offset	0x0000 0070		
Physical Address	0x4880 4070	**Instance**	OCMC_RAM
Description	This register contains the EOI vector.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															EOI_VECTOR

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	EOI_VECTOR		RW	0x0

Table 10-599. Register Call Summary for Register OCMC_INTR1_EOI

On-Chip Memory (OCM) Subsystem

- OCM Subsystem Register Summary: [2]

Table 10-600. CFG_OCMC_ECC

Address Offset	0x0000 0080		
Physical Address	0x4880 4080	**Instance**	OCMC_RAM
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																										CFG_ECC_OPT_NON_ECC_READ	CFG_ECC_ERR_SRESP_EN	CFG_ECC_SEC_AUTO_CORRECT	CFG_OCMC_MODE		

Bits	Field Name	Description	Type	Reset
31:6	RESERVED		R	0x0
5	CFG_ECC_OPT_NON_ECC_READ	Optimize read latency for non-ECC read. Returns the data one cycle faster if the read access is from a non-ECC enabled space. 0x0: Disable 0x1: Enable	RW	0x0
4	CFG_ECC_ERR_SRESP_EN	ECC non-correctable error SRESP enable. Enables ERR return on L3 OCP SRESP when a non-correctable data (DED) or address error is detected. 0x0: Disable 0x1: Enable	RW	0x0
3	CFG_ECC_SEC_AUTO_CORRECT	SEC error auto correction mode. Enables the OCM Controller to automatically update the wrong data word with the corrected word. 0x0: Disable 0x1: Enable (If the OCM Controller is performing a read-modify operation for a sub-128b write to an ECC enabled memory, the error found during the read phase will be corrected always regardless of the value of this bit)	RW	0x0
2:0	CFG_OCMC_MODE	OCM Controller memory access modes. 0x0: Non-ECC mode (data access) 0x1: Non-ECC mode (code access) 0x2: Full ECC enabled mode 0x3: Block ECC enabled mode 0x4-0x7: Reserved (internally defaults to 0x0 mode)	RW	0x0

Table 10-601. Register Call Summary for Register CFG_OCMC_ECC

On-Chip Memory (OCM) Subsystem

- OCM Controller Modes Of Operation: [0]
- ECC Support: [1][2][3][4][5]
- OCM Subsystem Register Summary: [8]

Table 10-602. CFG_OCMC_ECC_MEM_BLK

Address Offset	0x0000 0084		
Physical Address	0x4880 4084	**Instance**	OCMC_RAM
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								CFG_ECC_ENABLED_128K_BLK																									

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19:0	CFG_ECC_ENABLED_128K_BLK	ECC memory block enable bits. The active level of each bit is 0x1. Bit [0] -> Address offset range 0x0 to 0x1FFFF Bit [1] -> Address offset range 0x20000 to 0x3FFFF ... Bit [19] -> Address offset range 0x260000 to 0x27FFFF	RW	0x0

Table 10-603. Register Call Summary for Register CFG_OCMC_ECC_MEM_BLK

On-Chip Memory (OCM) Subsystem

- OCM Controller Modes Of Operation: [0]
- ECC Support: [1]
- OCM Subsystem Register Summary: [4]

Table 10-604. CFG_OCMC_ECC_ERROR

Address Offset	0x0000 0088		
Physical Address	0x4880 4088	**Instance**	OCMC_RAM
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED							CFG_DISCARD_DUP_ADDR	CFG_ADDR_ERR_CNT_MAX								CFG_DED_CNT_MAX								CFG_SEC_CNT_MAX							

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24	CFG_DISCARD_DUP_ADDR	Do not save duplicate error address. This bit applies to the SEC, DED and ADDRERR FIFOs. 0x0: Save the duplicated addresses 0x1: Save only the unique addresses	RW	0x0
23:20	CFG_ADDR_ERR_CNT_MAX	Number of ADDR errors to trigger an interrupt (The value configured must be > 0 to generate an interrupt).	RW	0x1
19:16	CFG_DED_CNT_MAX	Number of DED errors to trigger an interrupt (The value configured must be > 0 to generate an interrupt).	RW	0x1
15:0	CFG_SEC_CNT_MAX	Number of SEC error to trigger an interrupt (The value configured must be > 0 to generate an interrupt).	RW	0x1

Table 10-605. Register Call Summary for Register CFG_OCMC_ECC_ERROR

On-Chip Memory (OCM) Subsystem
- Interrupt Requests: [0][1][2]
- ECC Associated FIFOs: [3][4][5]
- ECC Support: [6][7][8][9][10][11]
- OCM Subsystem Register Summary: [14]

Table 10-606. CFG_OCMC_ECC_CLEAR_HIST

Address Offset	0x0000 008C		
Physical Address	0x4880 408C	**Instance**	OCMC_RAM
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												CLEAR_SEC_BIT_DISTR	CLEAR_ADDR_ERR_CNT	CLEAR_DED_ERR_CNT	CLEAR_SEC_ERR_CNT

Bits	Field Name	Description	Type	Reset
31:4	RESERVED		R	0x0
3	CLEAR_SEC_BIT_DISTR	Clear stored single error correction (SEC) bit distribution history. Returns 0 on read. 0x0: Reserved (not used) 0x1: Cleares the following registers: • STATUS_SEC_ERROR_DISTR_0 • STATUS_SEC_ERROR_DISTR_1 • STATUS_SEC_ERROR_DISTR_2 • STATUS_SEC_ERROR_DISTR_3 • STATUS_SEC_ERROR_DISTR_4	RW	0x0
2	CLEAR_ADDR_ERR_CNT	Clear stored address error history. Returns 0 on read. 0x0: Reserved (not used) 0x1: Clears the STATUS_ERROR_CNT[23:20] ADDR_ERROR_CNT bit field and the ADDRERR FIFO	RW	0x0

Bits	Field Name	Description	Type	Reset
1	CLEAR_DED_ERR_CNT	Clear stored double error detection (DED) history. Returns 0 on read. 0x0: Reserved (not used) 0x1: Clears the STATUS_ERROR_CNT[19:16] DED_ERROR_CNT bit field and the DED FIFO	RW	0x0
0	CLEAR_SEC_ERR_CNT	Clear stored single error correction history. Returns 0 on read. 0x0: Reserved (not used) 0x1: Clears the STATUS_ERROR_CNT[15:0] SEC_ERROR_CNT bit field and the SEC FIFO	RW	0x0

Table 10-607. Register Call Summary for Register CFG_OCMC_ECC_CLEAR_HIST

On-Chip Memory (OCM) Subsystem
- Interrupt Requests: [0][1][2]
- ECC Associated FIFOs: [3][4][5]
- ECC Support: [6][7]
- OCM Subsystem Register Summary: [10]
- OCM Subsystem Register Description: [11][12][13]

Table 10-608. STATUS_ERROR_CNT

Address Offset	0x0000 0090		
Physical Address	0x4880 4090	**Instance**	OCMC_RAM
Description	OCM Controller error status		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								ADDR_ERROR_CNT				DED_ERROR_CNT				SEC_ERROR_CNT															

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:20	ADDR_ERROR_CNT	Counter for the address errors found. This bit field is reset when 0x1 is written to the CFG_OCMC_ECC_CLEAR_HIST[2] CLEAR_ADDR_ERR_CNT bit.	R	0x0
19:16	DED_ERROR_CNT	Counter for the double error detections. This bit field is reset when 0x1 is written to the CFG_OCMC_ECC_CLEAR_HIST[1] CLEAR_DED_ERR_CNT bit.	R	0x0
15:0	SEC_ERROR_CNT	Counter for the single errors occured. This bit field is reset when 0x1 is written to the CFG_OCMC_ECC_CLEAR_HIST[0] CLEAR_SEC_ERR_CNT bit.	R	0x0

Table 10-609. Register Call Summary for Register STATUS_ERROR_CNT

On-Chip Memory (OCM) Subsystem

- Interrupt Requests: [0][1][2]
- ECC Counters And Corrected Bit Distribution Register: [3][4][5]
- ECC Support: [6][7][8][9][10][11][12]
- OCM Subsystem Register Summary: [15]
- OCM Subsystem Register Description: [16][17][18]

Table 10-610. STATUS_SEC_ERROR_TRACE

Address Offset	0x0000 0094		
Physical Address	0x4880 4094	**Instance**	OCMC_RAM
Description	SEC error 128-bit memory address		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED													VALID	ADDRESS_128BIT																	

Bits	Field Name	Description	Type	Reset
31:19	RESERVED		R	0x0
18	VALID	SEC FIFO valid addres indication. 0x0: The SEC FIFO is empty 0x1: There is a valid address in the SEC FIFO	R	0x0
17:0	ADDRESS_128BIT	SEC error 128-bit memory address (Read from the SEC error address trace fifo)	R	0x0

Table 10-611. Register Call Summary for Register STATUS_SEC_ERROR_TRACE

On-Chip Memory (OCM) Subsystem

- ECC Associated FIFOs: [0][1]
- OCM Subsystem Register Summary: [4]

Table 10-612. STATUS_DED_ERROR_TRACE

Address Offset	0x0000 0098		
Physical Address	0x4880 4098	**Instance**	OCMC_RAM
Description	DED error 128-bit memory address		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED													VALID	ADDRESS_128BIT																	

Bits	Field Name	Description	Type	Reset
31:19	RESERVED		R	0x0
18	VALID	DED FIFO valid addres indication. 0x0: The DED FIFO is empty 0x1: There is a valid address in the DED FIFO	R	0x0
17:0	ADDRESS_128BIT	DED error 128-bit memory address (Read from the DED error address trace fifo)	R	0x0

Table 10-613. Register Call Summary for Register STATUS_DED_ERROR_TRACE

On-Chip Memory (OCM) Subsystem
- ECC Associated FIFOs: [0][1][2]
- OCM Subsystem Register Summary: [5]

Table 10-614. STATUS_ADDR_TRANSLATION_ERROR_TRACE

Address Offset	0x0000 009C		
Physical Address	0x4880 409C	**Instance**	OCMC_RAM
Description	ADDR error 128-bit memory address		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED													VALID	ADDRESS_128BIT																	

Bits	Field Name	Description	Type	Reset
31:19	RESERVED		R	0x0
18	VALID	ADDRERR FIFO valid addres indication. 0x0: The ADDRERR FIFO is empty 0x1: There is a valid address in the ADDRERR FIFO	R	0x0
17:0	ADDRESS_128BIT	ADDR error 128-bit memory address (Read from the ADDR error address trace fifo)	R	0x0

Table 10-615. Register Call Summary for Register STATUS_ADDR_TRANSLATION_ERROR_TRACE

On-Chip Memory (OCM) Subsystem
- ECC Associated FIFOs: [0][1][2]
- OCM Subsystem Register Summary: [5]

Table 10-616. STATUS_SEC_ERROR_DISTR_0

Address Offset	0x0000 00A0		
Physical Address	0x4880 40A0	**Instance**	OCMC_RAM
Description	SEC data error bit distribution status [31:0]		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SEC_BIT_ERROR_FOUND																															

Bits	Field Name	Description	Type	Reset
31:0	SEC_BIT_ERROR_FOUND	1 in a bit position means that an SEC error was found at that bit position and corrected	R	0x0

Table 10-617. Register Call Summary for Register STATUS_SEC_ERROR_DISTR_0

On-Chip Memory (OCM) Subsystem
- ECC Counters And Corrected Bit Distribution Register: [0]
- ECC Support: [1]
- OCM Subsystem Register Summary: [4]
- OCM Subsystem Register Description: [5]

Table 10-618. STATUS_SEC_ERROR_DISTR_1

Address Offset	0x0000 00A4		
Physical Address	0x4880 40A4	**Instance**	OCMC_RAM
Description	SEC data error bit distribution status [63:32]		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SEC_BIT_ERROR_FOUND																															

Bits	Field Name	Description	Type	Reset
31:0	SEC_BIT_ERROR_FOUND	1 in a bit position means that an SEC error was found at that bit position and corrected	R	0x0

Table 10-619. Register Call Summary for Register STATUS_SEC_ERROR_DISTR_1

On-Chip Memory (OCM) Subsystem

- ECC Counters And Corrected Bit Distribution Register: [0]
- ECC Support: [1]
- OCM Subsystem Register Summary: [4]
- OCM Subsystem Register Description: [5]

Table 10-620. STATUS_SEC_ERROR_DISTR_2

Address Offset	0x0000 00A8		
Physical Address	0x4880 40A8	**Instance**	OCMC_RAM
Description	SEC data error bit distribution status [95:64]		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SEC_BIT_ERROR_FOUND																															

Bits	Field Name	Description	Type	Reset
31:0	SEC_BIT_ERROR_FOUND	1 in a bit position means that an SEC error was found at that bit position and corrected	R	0x0

Table 10-621. Register Call Summary for Register STATUS_SEC_ERROR_DISTR_2

On-Chip Memory (OCM) Subsystem

- ECC Counters And Corrected Bit Distribution Register: [0]
- ECC Support: [1]
- OCM Subsystem Register Summary: [4]
- OCM Subsystem Register Description: [5]

Table 10-622. STATUS_SEC_ERROR_DISTR_3

Address Offset	0x0000 00AC		
Physical Address	0x4880 40AC	**Instance**	OCMC_RAM
Description	SEC data error bit distribution status [127:96]		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SEC_BIT_ERROR_FOUND																															

Bits	Field Name	Description	Type	Reset
31:0	SEC_BIT_ERROR_FOUND	1 in a bit position means that an SEC error was found at that bit position and corrected	R	0x0

Table 10-623. Register Call Summary for Register STATUS_SEC_ERROR_DISTR_3

On-Chip Memory (OCM) Subsystem

- ECC Counters And Corrected Bit Distribution Register: [0]
- ECC Support: [1]
- OCM Subsystem Register Summary: [4]
- OCM Subsystem Register Description: [5]

Table 10-624. STATUS_SEC_ERROR_DISTR_4

Address Offset	0x0000 00B0		
Physical Address	0x4880 40B0	**Instance**	OCMC_RAM
Description	SEC ecc code error bit distribution status [7:0]		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								SEC_ECC_CODE_ERROR_FOUND							

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x0
7:0	SEC_ECC_CODE_ERROR_FOUND	ECC Code (excluding the parity bit) error distribution [7:0]. For each bit: 0x0: SEC error not found 0x1: SEC error found In the corresponding bit location	R	0x0

Table 10-625. Register Call Summary for Register STATUS_SEC_ERROR_DISTR_4

On-Chip Memory (OCM) Subsystem

- ECC Counters And Corrected Bit Distribution Register: [0]
- ECC Support: [1][2]
- OCM Subsystem Register Summary: [5]
- OCM Subsystem Register Description: [6]

Table 10-626. CFG_OCMC_CBUF_EN

Address Offset	0x0000 0200		
Physical Address	0x4880 4200	**Instance**	OCMC_RAM
Description	CBUF mode enable register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				CBUF_EN_11	CBUF_EN_10	CBUF_EN_9	CBUF_EN_8	CBUF_EN_7	CBUF_EN_6	CBUF_EN_5	CBUF_EN_4	CBUF_EN_3	CBUF_EN_2	CBUF_EN_1	CBUF_EN_0	RESERVED														NEW_FRAME_SEL	CBUF_DEBUG_EN	CBUF_MODE_EN

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27	CBUF_EN_11	CBUF 11 enable. 0x0: Disable 0x1: Enable	RW	0x0
26	CBUF_EN_10	CBUF 10 enable. 0x0: Disable 0x1: Enable	RW	0x0
25	CBUF_EN_9	CBUF 9 enable. 0x0: Disable 0x1: Enable	RW	0x0
24	CBUF_EN_8	CBUF 8 enable. 0x0: Disable 0x1: Enable	RW	0x0
23	CBUF_EN_7	CBUF 7 enable. 0x0: Disable 0x1: Enable	RW	0x0
22	CBUF_EN_6	CBUF 6 enable. 0x0: Disable 0x1: Enable	RW	0x0
21	CBUF_EN_5	CBUF 5 enable. 0x0: Disable 0x1: Enable	RW	0x0
20	CBUF_EN_4	CBUF 4 enable. 0x0: Disable 0x1: Enable	RW	0x0
19	CBUF_EN_3	CBUF 3 enable. 0x0: Disable 0x1: Enable	RW	0x0
18	CBUF_EN_2	CBUF 2 enable. 0x0: Disable 0x1: Enable	RW	0x0
17	CBUF_EN_1	CBUF 1 enable. 0x0: Disable 0x1: Enable	RW	0x0

Bits	Field Name	Description	Type	Reset
16	CBUF_EN_0	CBUF 0 enable. 0x0: Disable 0x1: Enable	RW	0x0
15:3	RESERVED		R	0x0
2	NEW_FRAME_SEL	CBUF New Frame Event Definition Select. 0x0: New frame event flag is set when a VBUF access is made to the base address of the VBUF 0x1: New frame event flag is set when a VBUF access is made to the base CBUF slice address range of the VBUF	RW	0x0
1	CBUF_DEBUG_EN	CBUF Debug Enable Mode. 0x0: Default Normal mode. All CBUF accesses with MReqDebug=1 are rejected. 0x1: Debug mode. MReqDebug Interconnect qualifier is ignored.	RW	0x0
0	CBUF_MODE_EN	CBUF Mode Enable. 0x0: Disable all CBUF address translation 0x1: Enable CBUF address translation	RW	0x0

Table 10-627. Register Call Summary for Register CFG_OCMC_CBUF_EN

On-Chip Memory (OCM) Subsystem
- Circular Buffer (CBUF) Support: [0][1][2][3]
- OCM Subsystem Register Summary: [6]

Table 10-628. CFG_OCMC_CBUF_RESET

Address Offset	0x0000 0204		
Physical Address	0x4880 4204	**Instance**	OCMC_RAM
Description	Writing 1 to bit n will set a reset bit to clear the corresponding CBUF_n address translation logic. Sliding CBUF frame tracking will be cleared so that the CBUF now points to the base of the virtual frame buffer. Normally, a reset is not required since the CBUF logic will clear itself when a VBUF access is to the base of the virtual frame.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																				CBUF_RESET_11	CBUF_RESET_10	CBUF_RESET_9	CBUF_RESET_8	CBUF_RESET_7	CBUF_RESET_6	CBUF_RESET_5	CBUF_RESET_4	CBUF_RESET_3	CBUF_RESET_2	CBUF_RESET_1	CBUF_RESET_0

Bits	Field Name	Description	Type	Reset
31:12	RESERVED		R	0x0
11	CBUF_RESET_11	cbuf_reset_11	RW W1toClr	0x0
10	CBUF_RESET_10	cbuf_reset_10	RW W1toClr	0x0
9	CBUF_RESET_9	cbuf_reset_9	RW W1toClr	0x0
8	CBUF_RESET_8	cbuf_reset_8	RW W1toClr	0x0
7	CBUF_RESET_7	cbuf_reset_7	RW W1toClr	0x0

Bits	Field Name	Description	Type	Reset
6	CBUF_RESET_6	cbuf_reset_6	RW W1toClr	0x0
5	CBUF_RESET_5	cbuf_reset_5	RW W1toClr	0x0
4	CBUF_RESET_4	cbuf_reset_4	RW W1toClr	0x0
3	CBUF_RESET_3	cbuf_reset_3	RW W1toClr	0x0
2	CBUF_RESET_2	cbuf_reset_2	RW W1toClr	0x0
1	CBUF_RESET_1	cbuf_reset_1	RW W1toClr	0x0
0	CBUF_RESET_0	cbuf_reset_0	RW W1toClr	0x0

Table 10-629. Register Call Summary for Register CFG_OCMC_CBUF_RESET

On-Chip Memory (OCM) Subsystem
- OCM Subsystem Register Summary: [2]

Table 10-630. CFG_OCMC_CBUF_ERR_HANDLER

Address Offset	0x0000 0208		
Physical Address	0x4880 4208	**Instance**	OCMC_RAM
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							UNDERFLOW_LAST_CBUF_SLICE_DISABLE	OVERFLOW_CHECK_REENABLE_SEL		OVERFLOW_WRITE_HANDLER_SEL		UNDERFLOW_ERR_CHECK_EN	OVERFLOW_ERR_CHECK_EN	SHORT_FRAME_PREV_EOF_SEL	SHORT_FRAME_DETECT_CHECK_EN

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	UNDERFLOW_LAST_CBUF_SLICE_DISABLE	0x0: Check underflow even when read is from the last CBUF slice 0x1: Disable underflow check when read is from the last CBUF slice	RW	0x0

Bits	Field Name	Description	Type	Reset
7:6	OVERFLOW_CHECK_REENABLE_SEL	Overflow check re-enable selection. 0x0: Overflow check is disabled until next wtire to or read from virtual frame start address is detected 0x1: Overflow check is disabled until next write to virtual frame start address is detected 0x2: Overflow check is disabled until next read from virtual frame start address is detected 0x3: Overflow check is re-enabled immediately	RW	0x0
5:4	OVERFLOW_WRITE_HANDLER_SEL	Overflow write handler selection. 0x0: Writes disabled only on CBUF_overflow_wrap cases until next write to virtual frame start address is detected 0x1: Writes disabled on all overflow cases until next write to virtual frame start address is detected 0x2: Writes serviced with CBUF pointer updated even on overflow condition 0x3: Reserved	RW	0x0
3	UNDERFLOW_ERR_CHECK_EN	Underflow chek enable. 0x0: Underflow check enabled 0x1: Underflow check disabled	RW	0x0
2	OVERFLOW_ERR_CHECK_EN	Overflow chek enable. 0x0: Overflow check enabled 0x1: Overflow check disabled	RW	0x0
1	SHORT_FRAME_PREV_EOF_SEL	0x0: previous frame EOF history is set if the last write address is equal to the VBUF frame end address 0x1: previous frame EOF history is set if the last write address is in the Last CBUF slice	RW	0x0
0	SHORT_FRAME_DETECT_CHECK_EN	Short frame detection enable. 0x0: Detection enabled 0x1: Detection disabled	RW	0x0

Table 10-631. Register Call Summary for Register CFG_OCMC_CBUF_ERR_HANDLER

On-Chip Memory (OCM) Subsystem

- Illegal Frame SIze (Short Frame Detection): [0][1]
- CBUF Overflow: [2][3][4]
- CBUF Underflow: [5][6]
- OCM Subsystem Register Summary: [9]

Table 10-632. STATUS_CBUF_WR_OUT_OF_RANGE_ERR

Address Offset	0x0000 020C		
Physical Address	0x4880 420C	**Instance**	OCMC_RAM
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																				CBUF_ERR											

Bits	Field Name	Description	Type	Reset
31:12	RESERVED		R	0x0
11:0	CBUF_ERR	Indicates that the CBUF write address is out of the CBUF range. Each bit indicates the error of this type for CBUF[n], where n = 0 to 11. Writing 0x1 to bit [n] clears the status of CBUF[n]. Reading 0x1 from bit [n] means that the status is set.	RW W1toClr	0x0

Table 10-633. Register Call Summary for Register STATUS_CBUF_WR_OUT_OF_RANGE_ERR

On-Chip Memory (OCM) Subsystem

- VBUF Address Not Mapped to a CBUF Memory Space: [0][1]
- OCM Subsystem Register Summary: [4]

Table 10-634. STATUS_CBUF_WR_VBUF_START_ERR

Address Offset	0x0000 0210		
Physical Address	0x4880 4210	**Instance**	OCMC_RAM
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																				CBUF_ERR											

Bits	Field Name	Description	Type	Reset
31:12	RESERVED		R	0x0
11:0	CBUF_ERR	CBUF write is not to the base address at vbuf access start. Each bit indicates the error of this type for CBUF[n], where n = 0 to 11. Writing 0x1 to bit [n] clears the status of CBUF[n]. Reading 0x1 from bit [n] means that the status is set.	RW W1toClr	0x0

Table 10-635. Register Call Summary for Register STATUS_CBUF_WR_VBUF_START_ERR

On-Chip Memory (OCM) Subsystem

- VBUF Access Not Starting At The Base Address: [0][1]
- OCM Subsystem Register Summary: [4]

Table 10-636. STATUS_CBUF_WR_ADDR_SEQ_ERROR

Address Offset	0x0000 0214		
Physical Address	0x4880 4214	**Instance**	OCMC_RAM
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																				CBUF_ERR											

Bits	Field Name	Description	Type	Reset
31:12	RESERVED		R	0x0
11:0	CBUF_ERR	CBUF address is not incrementing in raster scan order. Each bit indicates the error of this type for CBUF[n], where n = 0 to 11. Writing 0x1 to bit [n] clears the status of CBUF[n]. Reading 0x1 from bit [n] means that the status is set.	RW W1toClr	0x0

Table 10-637. Register Call Summary for Register STATUS_CBUF_WR_ADDR_SEQ_ERROR

On-Chip Memory (OCM) Subsystem

- Illegal Address Change Between Two Same Type Accesses: [0][1]
- OCM Subsystem Register Summary: [4]

Table 10-638. STATUS_CBUF_RD_OUT_OF_RANGE_ERROR

Address Offset	0x0000 0218		
Physical Address	0x4880 4218	**Instance**	OCMC_RAM
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																				CBUF_ERR											

Bits	Field Name	Description	Type	Reset
31:12	RESERVED		R	0x0
11:0	CBUF_ERR	Indicates that the CBUF read address is out of the CBUF range. Each bit indicates the error of this type for CBUF[n], where n = 0 to 11. Writing 0x1 to bit [n] clears the status of CBUF[n]. Reading 0x1 from bit [n] means that the status is set.	RW W1toClr	0x0

Table 10-639. Register Call Summary for Register STATUS_CBUF_RD_OUT_OF_RANGE_ERROR

On-Chip Memory (OCM) Subsystem

- VBUF Address Not Mapped to a CBUF Memory Space: [0][1]
- OCM Subsystem Register Summary: [4]

Table 10-640. STATUS_CBUF_VBUF_RD_START_ERROR

Address Offset	0x0000 021C		
Physical Address	0x4880 421C	**Instance**	OCMC_RAM
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																				CBUF_ERR											

Bits	Field Name	Description	Type	Reset
31:12	RESERVED		R	0x0
11:0	CBUF_ERR	CBUF read is not from the base address at VBUF access start. Each bit indicates the error of this type for CBUF[n], where n = 0 to 11. Writing 0x1 to bit [n] clears the status of CBUF[n]. Reading 0x1 from bit [n] means that the status is set.	RW W1toClr	0x0

Table 10-641. Register Call Summary for Register STATUS_CBUF_VBUF_RD_START_ERROR

On-Chip Memory (OCM) Subsystem

- VBUF Access Not Starting At The Base Address: [0][1]
- OCM Subsystem Register Summary: [4]

Table 10-642. STATUS_CBUF_RD_ADDR_SEQ_ERROR

Address Offset	0x0000 0220		
Physical Address	0x4880 4220	**Instance**	OCMC_RAM
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																				CBUF_ERR											

Bits	Field Name	Description	Type	Reset
31:12	RESERVED		R	0x0
11:0	CBUF_ERR	CBUF read address is not incrementing in raster scan order. Each bit indicates the error of this type for CBUF[n], where n = 0 to 11. Writing 0x1 to bit [n] clears the status of CBUF[n]. Reading 0x1 from bit [n] means that the status is set.	RW W1toClr	0x0

Table 10-643. Register Call Summary for Register STATUS_CBUF_RD_ADDR_SEQ_ERROR

On-Chip Memory (OCM) Subsystem

- Illegal Address Change Between Two Same Type Accesses: [0][1]
- OCM Subsystem Register Summary: [4]

Table 10-644. STATUS_CBUF_OVERFLOW_MID

Address Offset	0x0000 0224		
Physical Address	0x4880 4224	**Instance**	OCMC_RAM
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																				CBUF_ERR											

Bits	Field Name	Description	Type	Reset
31:12	RESERVED		R	0x0
11:0	CBUF_ERR	CBUF overflow condition detected in the middle of a frame. Each bit indicates the error of this type for CBUF[n], where n = 0 to 11. Writing 0x1 to bit [n] clears the status of CBUF[n]. Reading 0x1 from bit [n] means that the status is set.	RW W1toClr	0x0

Table 10-645. Register Call Summary for Register STATUS_CBUF_OVERFLOW_MID

On-Chip Memory (OCM) Subsystem

- CBUF Overflow: [0]
- OCM Subsystem Register Summary: [3]

Table 10-646. STATUS_CBUF_OVERFLOW_WRAP

Address Offset	0x0000 0228		
Physical Address	0x4880 4228	**Instance**	OCMC_RAM
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																				CBUF_ERR											

Bits	Field Name	Description	Type	Reset
31:12	RESERVED		R	0x0
11:0	CBUF_ERR	CBUF overflow condition detected during buffer switching. Each bit indicates the error of this type for CBUF[n], where n = 0 to 11. Writing 0x1 to bit [n] clears the status of CBUF[n]. Reading 0x1 from bit [n] means that the status is set.	RW W1toClr	0x0

Table 10-647. Register Call Summary for Register STATUS_CBUF_OVERFLOW_WRAP

On-Chip Memory (OCM) Subsystem

- CBUF Overflow: [0]
- OCM Subsystem Register Summary: [3]

Table 10-648. STATUS_CBUF_UNDERFLOW

Address Offset	0x0000 022C		
Physical Address	0x4880 422C	**Instance**	OCMC_RAM
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																				CBUF_ERR											

Bits	Field Name	Description	Type	Reset
31:12	RESERVED		R	0x0
11:0	CBUF_ERR	CBUF underflow condition detected. Each bit indicates the error of this type for CBUF[n], where n = 0 to 11. Writing 0x1 to bit [n] clears the status of CBUF[n]. Reading 0x1 from bit [n] means that the status is set.	RW W1toClr	0x0

Table 10-649. Register Call Summary for Register STATUS_CBUF_UNDERFLOW

On-Chip Memory (OCM) Subsystem

- Interrupt Requests: [0]
- CBUF Underflow: [1][2][3]
- OCM Subsystem Register Summary: [6]

Table 10-650. STATUS_CBUF_SHORT_FRAME_DETECT

Address Offset	0x0000 0230		
Physical Address	0x4880 4230	**Instance**	OCMC_RAM
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																				CBUF_ERR											

Bits	Field Name	Description	Type	Reset
31:12	RESERVED		R	0x0
11:0	CBUF_ERR	CBUF short frame detected. Each bit indicates the error of this type for CBUF[n], where n = 0 to 11. Writing 0x1 to bit [n] clears the status of CBUF[n]. Reading 0x1 from bit [n] means that the status is set.	RW W1toClr	0x0

Table 10-651. Register Call Summary for Register STATUS_CBUF_SHORT_FRAME_DETECT

On-Chip Memory (OCM) Subsystem

- Interrupt Requests: [0]
- Illegal Frame SIze (Short Frame Detection): [1][2]
- OCM Subsystem Register Summary: [5]

Table 10-652. CBUF_i_VBUF_START_ADDR

Address Offset	0x0000 0240 + (i*16); i = 0 to 11		
Physical Address	0x4880 4240 + (i*16)	**Instance**	OCMC_RAM
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								ADDR																				RESERVED			

Bits	Field Name	Description	Type	Reset
31:24	RESERVED	The virtual address range is determined by the OCMC_MEM_SIZE_READ[16:12] VBUF_ADDR_MSB bit field. For default value of 23, the valid VBUF address is 0x80xx_xxxx. Writing to this field above the MSB bit returned by VBUF_ADDR_MSB will be ignored and reading will return all zeroes except for bit[31] = 1.	R	0x80
23:4	ADDR	Virtual frame start address for this CBUF - bits [23:4]	RW	0x0
3:0	RESERVED		R	0x0

Table 10-653. Register Call Summary for Register CBUF_i_VBUF_START_ADDR

On-Chip Memory (OCM) Subsystem
- Circular Buffer (CBUF) Support: [0]
- VBUF Access Not Starting At The Base Address: [1][2]
- OCM Subsystem Register Summary: [5]

Table 10-654. CBUF_i_VBUF_END_ADDR

Address Offset	0x0000 0244 + (i*16); i = 0 to 11		
Physical Address	0x4880 4244 + (i*16)	**Instance**	OCMC_RAM
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								ADDR																				RESERVED			

Bits	Field Name	Description	Type	Reset
31:24	RESERVED	The virtual address range is determined by the OCMC_MEM_SIZE_READ[16:12] VBUF_ADDR_MSB bit field. For default value of 23, the valid VBUF address is 0x80xx_xxxx. Writing to this field above the MSB bit returned by VBUF_ADDR_MSB will be ignored and reading will return all zeroes except for bit[31] = 1.	R	0x80
23:4	ADDR	Virtual frame end address for this CBUF - bits [23:4]	RW	0x0
3:0	RESERVED		R	0x0

Table 10-655. Register Call Summary for Register CBUF_i_VBUF_END_ADDR

On-Chip Memory (OCM) Subsystem
- Circular Buffer (CBUF) Support: [0]
- OCM Subsystem Register Summary: [3]

Table 10-656. CBUF_i_OCMC_START_ADDR

Address Offset	0x0000 0248 + (i*16); i = 0 to 11		
Physical Address	0x4880 4248 + (i*16)	**Instance**	OCMC_RAM
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								ADDR																				RESERVED			

Bits	Field Name	Description	Type	Reset
31:22	RESERVED		R	0x0
21:4	ADDR	SRAM start address for this CBUF - bits [21:4]	RW	0x0
3:0	RESERVED		R	0x0

Table 10-657. Register Call Summary for Register CBUF_i_OCMC_START_ADDR

On-Chip Memory (OCM) Subsystem
- Circular Buffer (CBUF) Support: [0]
- OCM Subsystem Register Summary: [3]

Table 10-658. CBUF_i_OCMC_BUF_SIZE

Address Offset	0x0000 024C + (i*16); i = 0 to 11		
Physical Address	0x4880 424C + (i*16)	**Instance**	OCMC_RAM
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								BUF_SIZE																RESERVED							

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19:4	BUF_SIZE	SRAM size allocated for this CBUF - bits [19:4]	RW	0x0
3:0	RESERVED		R	0x0

Table 10-659. Register Call Summary for Register CBUF_i_OCMC_BUF_SIZE

On-Chip Memory (OCM) Subsystem
- Circular Buffer (CBUF) Support: [0]
- OCM Subsystem Register Summary: [3]

Table 10-660. CBUF_k_LAST_WR_ADDR

Address Offset	0x0000 0300 + (k*8); k = 0 to 11		
Physical Address	0x4880 4300 + (k*8)	**Instance**	OCMC_RAM
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR																															

Bits	Field Name	Description	Type	Reset
31:0	ADDR	Last virtual write address accessing CBUF	R	0x0

Table 10-661. Register Call Summary for Register CBUF_k_LAST_WR_ADDR

On-Chip Memory (OCM) Subsystem
- Status Reporting: [0]
- OCM Subsystem Register Summary: [3]

Table 10-662. CBUF_k_LAST_RD_ADDR

Address Offset	0x0000 0304 + (k*8); k = 0 to 11		
Physical Address	0x4880 4304 + (k*8)	**Instance**	OCMC_RAM
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR																															

Bits	Field Name	Description	Type	Reset
31:0	ADDR	Last virtual read address accessing CBUF	R	0x0

Table 10-663. Register Call Summary for Register CBUF_k_LAST_RD_ADDR

On-Chip Memory (OCM) Subsystem
- Status Reporting: [0]
- OCM Subsystem Register Summary: [3]

Table 10-664. LAST_ILLEGAL_OCMC_ADDR

Address Offset	0x0000 0360		
Physical Address	0x4880 4360	**Instance**	OCMC_RAM
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ADDR																															

Bits	Field Name	Description	Type	Reset
31:0	ADDR	Last Illegal OCMC Address. This register returns the OCMC L3_MAIN address of the last access that was invalidated due to an OUT_OF_RANGE_ERR_FOUND (non-VBUF address) error or any one of the CBUF related access errors (including any write access disabled during overflow error handling).	R	0x0

Table 10-665. Register Call Summary for Register LAST_ILLEGAL_OCMC_ADDR

On-Chip Memory (OCM) Subsystem
- Interrupt Requests: [0]
- VBUF Address Not Mapped to a CBUF Memory Space: [1]
- OCM Subsystem Register Summary: [4]

Enhanced DMA

This chapter describes the Enhanced Direct Memory Access (EDMA) controller.

Topic **Page**

11.1 EDMA Module Overview

The enhanced direct memory access module, also called EDMA, performs high-performance data transfers between two slave points, memories and peripheral devices without microprocessor unit (MPU) or digital signal processor (DSP) support during transfer. EDMA transfer is programmed through a logical EDMA channel, which allows the transfer to be optimally tailored to the requirements of the application.

The EDMA can also perform transfers between external memories and between device subsystems internal memories, with some performance loss caused by resource sharing between the read and write ports.

EDMA controller is based on two major principal blocks:

- EDMA third-party channel controller (EDMA_TPCC)
- EDMA third-party transfer controller (EDMA_TPTC)

The **TPCC** is a high flexible Channel Controller. It serves as an user interface and an event interface for the EDMA controller. The EDMA_TPCC serves to prioritize incoming software requests or events from peripherals and submits transfer requests (TRs) to the transfer controller.

The **TPTC** performs read and write transfers by EDMA ports to the slave peripherals as programmed in the "Active" and "Pending" set of the registers. The transfer controllers are responsible for data movement and issue read/write commands to the source and destination addresses that are programmed for a given transfer in the EDMA_TPCC.

The SoC integrates the following EDMA instances:

- One system-level EDMA
- One DSP internal EDMA (per DSP)
- One EVE internal EDMA (per EVE)

Each of these EDMA modules consists of:

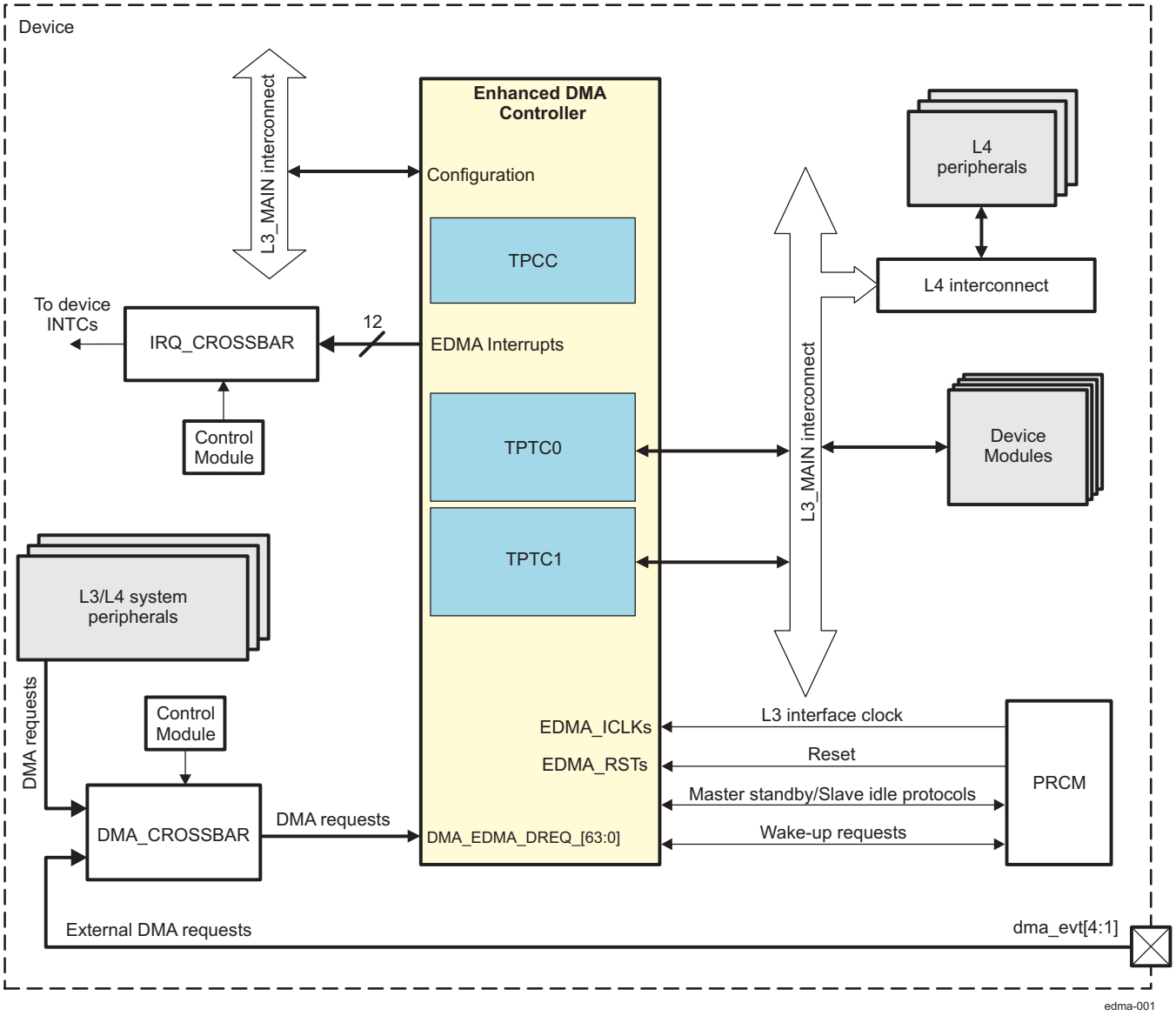
- One TPCC instance
- Two TPTC instances

NOTE: All EDMA modules in the SoC are functionally identical. Note that some of the configuration parameters may be different for the various EDMA instances (see Section 11.1.2, *EDMA Controllers Configuration*).

This chapter is mostly focused on describing the system-level EDMA module (in terms of configuration and integration in the SoC). For details on DSPx_EDMA / EVEx_EDMA integration, see their respective chapters.

Figure 11-1 shows an overview of the EDMA module.

Figure 11-1. EDMA Module Overview



The device CPUs can configure the EDMA controller blocks through the L3_MAIN interconnect.

11.1.1 EDMA Features

The **EDMA_TPCC** channel controller has following features:

- Fully orthogonal transfer description:
  - Three transfer dimensions.
  - A-synchronized transfers: one-dimension serviced per event.
  - AB-synchronized transfers: two-dimensions serviced per event.
  - Independent indexes on source and destination.
  - Chaining feature allows a 3-D transfer based on a single event.
- Flexible transfer definition:
  - Increment or FIFO transfer addressing modes.
  - Linking mechanism allows automatic PaRAM set update.

  - Chaining allows multiple transfers to execute with one event.
- Interrupt generation for the following:
  - Transfer completion.
  - Error conditions.
- Debug visibility:
  - Queue water marking/threshold.
  - Error and status recording to facilitate debug.
- 64 DMA request channels:
  - Event synchronization.
  - Manual synchronization (CPU(s) write to event set registers EDMA_TPCC_ESR and EDMA_TPCC_ESRH).
  - Chain synchronization (completion of one transfer triggers another transfer).
- Eight QDMA channels:
  - QDMA channels trigger automatically upon writing to a parameter RAM (PaRAM) set entry.
  - Support for programmable QDMA channel to PaRAM mapping.
- 512 PaRAM sets:
  - Each PaRAM set can be used for a DMA channel, QDMA channel, or link set.
- Two transfer controllers/event queues.
- 16 event entries per event queue.
- Memory protection support:
  - Proxy memory protection for TR submission.
  - Active memory protection for accesses to PaRAM and registers.

The **EDMA_TPTC** transfer controller has the following features:

- Two transfer controllers (TC).
- 128-bit wide read and write ports per TC.
- Up to four in-flight transfer requests (TRs).
- Programmable priority level.
- Supports two-dimensional transfers with independent indexes on source and destination (EDMA_TPCC manages the 3rd dimension).
- Support for increment or constant addressing mode transfers.
- Interrupt and error support.
- Memory-Mapped Register (MMR) bit fields are fixed position in 32-bit MMR regardless of endianness.

EDMA controller uses the shared MMU1 module for transfering to and from DSP module. This provides several benefits including:

- Protection of Host CPU memory regions from accidental corruption by EDMA TPTCs.
- Direct allocation of buffers in user space without the need for translation between CPU and DSP applications utilizing EDMA TPTCs.

Accesses by the EDMA TPTCs (both TPTC0 and TPTC1) may optionally be routed through the MMU1.

The TPTC0 and TPTC1 routing allows EDMA transfer controller to be used to perform transfers using only the virtual addresses of the associated buffers.

For more information about MMU1 module refer to Chapter 15 *Memory Management Units*.

NOTE: The device DSP integrated EDMA controller instances (DSP_EDMA_TPCC, DSP_EDMA_TPTC0 and DSP_EDMA_TPTC1) are functionally identical with the device EDMA controller instances (EDMA_TPCC, EDMA_TPTC0 and EDMA_TPTC1). The only difference is that the DSP_EDMA instances are located at different physical addresses.

For detail information about DSPSS refer to Chapter 4 *DSP Subsystem*.

11.1.2 EDMA Controllers Configuration

Table 11-1 summarizes the configuration for each of the EDMA channel controllers present on the SoC.

Table 11-1. EDMA Channel Controllers Configuration

Parameter	SYS_EDMA CC Configuration	DSPx_EDMA CC Configuration	EVEx_EDMA CC Configuration
Number of DMA channels (NUM_DMACH)	64	64	16
Number of QDMA channels (NUM_QDMACH)	8	8	8
Number of interrupt channels (NUM_INTCH)	64	64	16
Number of PaRAM set entries (NUM_PARAMENTRY)	512	128	128
Number of event queues (NUM_EVQUE)	2	2	2
Number of transfer controllers (NUM_TC)	2	2	2
Memory protection existence (MPEXIST)	Yes	Yes	Yes
Number of memory protection and shadow regions (NUM_REGIONS)	8	8	8
Channel mapping existence (CHMAPEXIST)	Yes	Yes	Yes

Table 11-2 summarizes the configuration of each of the EDMA transfer controllers present on the SoC.

Table 11-2. EDMA Transfer Controllers Configuration

Parameter	SYS_EDMA TC0 / TC1 Configuration	DSPx_EDMA TC0 / TC1 Configuration	EVEx_EDMA TC0 / TC1 Configuration
Data FIFO size (FIFOSIZE)	1024 bytes	2048 bytes	2048 bytes
Bus width (BUSBYTE)	16 bytes	16 bytes	16 bytes
Number of destination FIFO register sets (DSTREGDEPTH)	4 entries	4 entries	4 entries
Default burst size (DBS)	Defined by CTRL_CORE_CONTROL_IO_1 register	Defined by DSP_SYS_BUS_CONFIG register	Defined by EVE_BUS_CONFIG register

11.2 EDMA Controller Environment

The EDMA controller supports external DMA requests through the dma_evt[4:1] pins (see Table 11-3). A logical channel can be configured to respond to an external synchronization request.

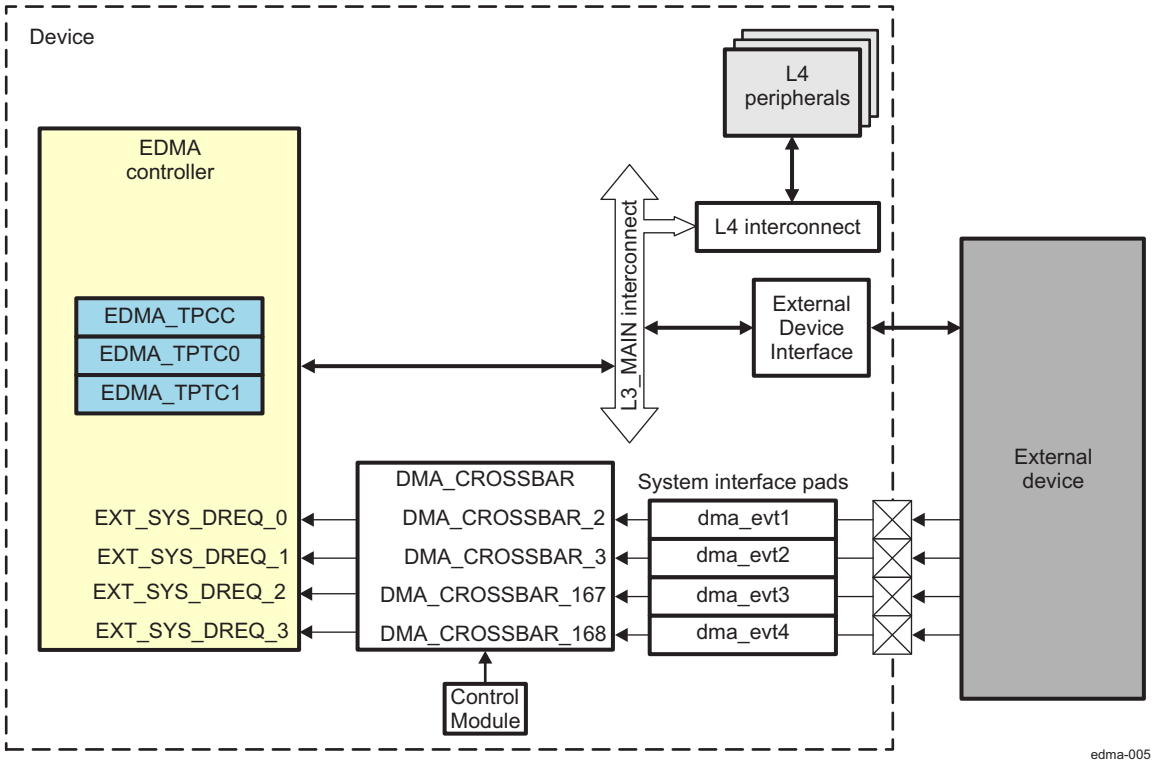
Table 11-3. External EDMA Request Signals

Pin Name	DMA_CROSSBAR Input	Signal Name	I/O[1]	Description	Module Reset Value
dma_evt1	DMA_CROSSBAR_2	EXT_SYS_DREQ_0	I	External DMA request 0 (system expansion)	Z
dma_evt2	DMA_CROSSBAR_3	EXT_SYS_DREQ_1	I	External DMA request 1 (system expansion)	Z
dma_evt3	DMA_CROSSBAR_167	EXT_SYS_DREQ_2	I	External DMA request 2 (system expansion)	Z
dma_evt4	DMA_CROSSBAR_168	EXT_SYS_DREQ_3	I	External DMA request 3 (system expansion)	Z

(1) I = Input, O = Output

Figure 11-2 shows an example of how to use the external hardware DMA request pins in the EDMA environment.

Figure 11-2. Example of External DMA Requests Use



An external device can use the external DMA request pins to start a logical channel transfer. The transfer can be a memory-to-memory transfer in which the source memory is in the external device.

By default, the external DMA request signals are not available on external pins after a cold reset. For more information about multiplexing out the two signal lines to pins, refer to Chapter 13, *Control Module*.

All 64 DMA request lines are transition sensitive.

11.3 EDMA Controller Integration

This section describes the integration of the module in the device, including information about clocks, resets, and hardware requests.

Figure 11-3 shows the EDMA controller integration.

Figure 11-3. EDMA Controller Integration

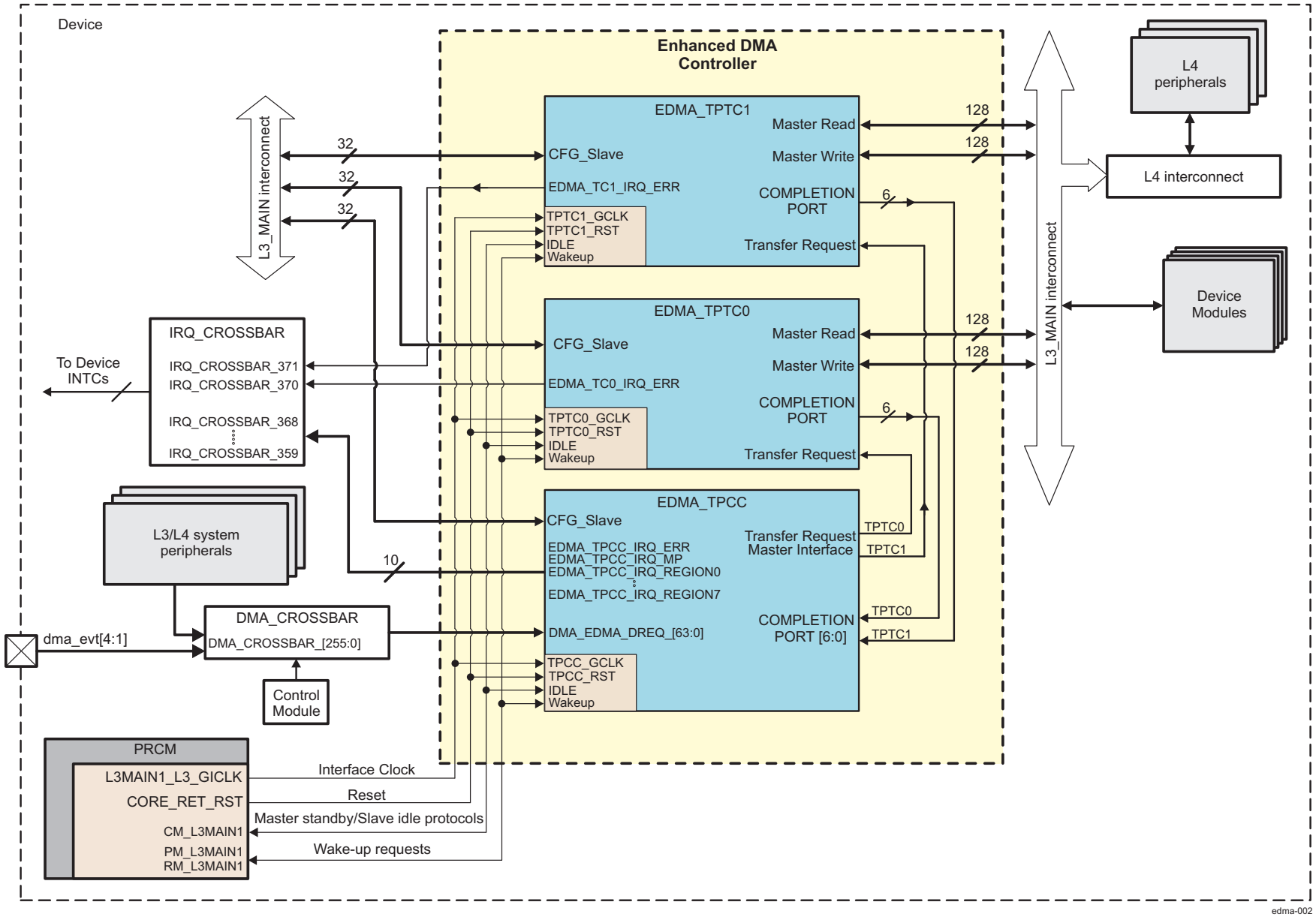


Table 11-4 through Table 11-6 summarize the integration of the module in the device.

Table 11-4. EDMA Integration Attributes

Module Instance	Attributes	
	Power Domain	Interconnect
EDMA_TPCC	PD_COREAON	L3_MAIN
EDMA_TPTC0		
EDMA_TPTC1		

Table 11-5. EDMA Clocks and Resets

Clocks				
Module Instance	**Destination Signal Name**	**Source Signal Name**	**Source**	**Description**
EDMA_TPCC	EDMA_TPCC_GCLK	L3MAIN1_L3_GICLK	PRCM	Interface clock. It supports the configuration port. For information about PRCM clock gating and management, see Chapter 3, *Power, Reset, and Clock Management*.
EDMA_TPTC0	EDMA_TPTC0_GCLK			
EDMA_TPTC1	EDMA_TPTC1_GCLK			
Resets				
Module Instance	**Destination Signal Name**	**Source Signal Name**	**Source**	**Description**
EDMA_TPCC	EDMA_TPCC_RST	CORE_RET_RST	PRCM	Hardware retention reset. It initializes all internal logic of the EDMA contrller modules, all global registers, and some of the per-channel registers, implemented in flip-flops. However, all remaining per-channel registers are memory-based, and, therefore, are not reset (have undefined values). Thus, when programming a channel for the first time, all bits that have undefined reset values must be configured before enabling the channel. For information about PRCM reset sources and distribution, see Chapter 3, *Power, Reset, and Clock Management*.
EDMA_TPTC0	EDMA_TPTC0_RST			
EDMA_TPTC1	EDMA_TPTC1_RST			

Table 11-6. EDMA Hardware Requests

Interrupt Requests				
Module Instance	**Source Signal Name**	**Destination IRQ_CROSSBAR INPUT**	**Default mapping**	**Description**
EDMA_TPCC	EDMA_TPCC_IRQ_ERR	IRQ_CROSSBAR_359	-	TPCC error interrupt. This IRQ source signal is not mapped by default to any device INTC.
	EDMA_TPCC_IRQ_MP	IRQ_CROSSBAR_360	-	TPCC memory protection interrupt. This IRQ source signal is not mapped by default to any device INTC.
	EDMA_TPCC_IRQ_REGION0	IRQ_CROSSBAR_361	-	TPCC Region 0 interrupt. This IRQ source signal is not mapped by default to any device INTC.

Table 11-6. EDMA Hardware Requests (continued)

	EDMA_TPCC_IRQ_REGION1	IRQ_CROSSBAR_362	-	TPCC Region 1 interrupt. This IRQ source signal is not mapped by default to any device INTC.
	EDMA_TPCC_IRQ_REGION2	IRQ_CROSSBAR_363	-	TPCC Region 2 interrupt. This IRQ source signal is not mapped by default to any device INTC.
	EDMA_TPCC_IRQ_REGION3	IRQ_CROSSBAR_364	-	TPCC Region 3 interrupt. This IRQ source signal is not mapped by default to any device INTC.
	EDMA_TPCC_IRQ_REGION4	IRQ_CROSSBAR_365	-	TPCC Region 4 interrupt. This IRQ source signal is not mapped by default to any device INTC.
	EDMA_TPCC_IRQ_REGION5	IRQ_CROSSBAR_366	-	TPCC Region 5 interrupt. This IRQ source signal is not mapped by default to any device INTC.
	EDMA_TPCC_IRQ_REGION6	IRQ_CROSSBAR_367	-	TPCC Region 6 interrupt. This IRQ source signal is not mapped by default to any device INTC.
	EDMA_TPCC_IRQ_REGION7	IRQ_CROSSBAR_368	-	TPCC Region 7 interrupt. This IRQ source signal is not mapped by default to any device INTC.
EDMA_TPTC0	EDMA_TC0_IRQ_ERR	IRQ_CROSSBAR_370	-	TPTC0 error interrupt. This IRQ source signal is not mapped by default to any device INTC.
EDMA_TPTC1	EDMA_TC1_IRQ_ERR	IRQ_CROSSBAR_371	-	TPTC1 error interrupt. This IRQ source signal is not mapped by default to any device INTC.

NOTE: The "**Default Mapping**" column in Table 11-6 *EDMA Hardware Requests* shows the default mapping of module IRQ source signals. These IRQ source signals can also be mapped to other lines of each device Interrupt controller through the IRQ_CROSSBAR or DMA_CROSSBAR modules.
For more information about the IRQ_CROSSBAR and DMA_CROSSBAR modules, see sections: Section 13.4.6.4*IRQ_CROSSBAR Module Functional Description* and Section 13.4.6.5 *DMA_CROSSBAR Module Functional Description*, in Chapter 13 *Control Module*.
For more information about the device interrupt controllers, see Chapter 12 *Interrupt Controllers*.

NOTE: For a description of the interrupt source, see Section 11.4.9, *EDMA interrupts*.

11.3.1 DMA Requests to the EDMA Controller

Table 11-7 lists the default DMA sources for the EDMA controller. In addition, the EDMA inputs (DMA_EDMA_DREQ_[63:0]) can alternatively be sourced through the associated DMA_CROSSBAR from one of the 256 multiplexed device DMA sources listed in Table 11-8. The CTRL_CORE_DMA_EDMA_DREQ_y_z registers (where *y* and *z* are indexes of EDMA input lines) in the Control Module are used to select between the default DMA sources and the multiplexed DMA sources.

Table 11-7. EDMA Default Request Mapping

DMA Request Line	DMA_ CROSSBAR Instance Number	DMA_CROSSBAR Configuration Register	DMA_ CROSSBAR Default Input Index	Default DMA Source Name	Default DMA Source Description
DMA_EDMA_DREQ_0	1	CTRL_CORE_DMA_EDMA_DREQ_0_1[7:0]	1	Reserved	Reserved
DMA_EDMA_DREQ_1	2	CTRL_CORE_DMA_EDMA_DREQ_0_1[23:16]	2	EXT_SYS_DREQ_0	External DMA request 0 (system expansion)
DMA_EDMA_DREQ_2	3	CTRL_CORE_DMA_EDMA_DREQ_2_3[7:0]	3	EXT_SYS_DREQ_1	External DMA request 1 (system expansion)
DMA_EDMA_DREQ_3	4	CTRL_CORE_DMA_EDMA_DREQ_2_3[23:16]	4	GPMC_DREQ	GPMC data transmit request from prefetch engine
DMA_EDMA_DREQ_4	5	CTRL_CORE_DMA_EDMA_DREQ_4_5[7:0]	5	Reserved	Reserved
DMA_EDMA_DREQ_5	6	CTRL_CORE_DMA_EDMA_DREQ_4_5[23:16]	6	DISPC_DREQ	Frame update request
DMA_EDMA_DREQ_6	7	CTRL_CORE_DMA_EDMA_DREQ_6_7[7:0]	7	CT_TBR_DREQ	DEBUG subsystem CT_TBR request
DMA_EDMA_DREQ_7	8	CTRL_CORE_DMA_EDMA_DREQ_6_7[23:16]	8	Reserved	Reserved
DMA_EDMA_DREQ_8	9	CTRL_CORE_DMA_EDMA_DREQ_8_9[7:0]	9	ISS_DREQ_1	ISS DMA request 1(1)
DMA_EDMA_DREQ_9	10	CTRL_CORE_DMA_EDMA_DREQ_8_9[23:16]	10	ISS_DREQ_2	ISS DMA request 2(1)
DMA_EDMA_DREQ_10	11	CTRL_CORE_DMA_EDMA_DREQ_10_11[7:0]	11	Reserved	Reserved
DMA_EDMA_DREQ_11	12	CTRL_CORE_DMA_EDMA_DREQ_10_11[23:16]	12	ISS_DREQ_3	ISS DMA request 3(1)
DMA_EDMA_DREQ_12	13	CTRL_CORE_DMA_EDMA_DREQ_12_13[7:0]	13	ISS_DREQ_4	ISS DMA request 4(1)
DMA_EDMA_DREQ_13	14	CTRL_CORE_DMA_EDMA_DREQ_12_13[23:16]	14	Reserved	Reserved
DMA_EDMA_DREQ_14	15	CTRL_CORE_DMA_EDMA_DREQ_14_15[7:0]	15	MCSPI3_DREQ_TX0	McSPI3 transmit request channel 0
DMA_EDMA_DREQ_15	16	CTRL_CORE_DMA_EDMA_DREQ_14_15[23:16]	16	MCSPI3_DREQ_RX0	McSPI3 receive request channel 0
DMA_EDMA_DREQ_16	17	CTRL_CORE_DMA_EDMA_DREQ_16_17[7:0]	17	Reserved	Reserved
DMA_EDMA_DREQ_17	18	CTRL_CORE_DMA_EDMA_DREQ_16_17[23:16]	18	Reserved	Reserved
DMA_EDMA_DREQ_18	19	CTRL_CORE_DMA_EDMA_DREQ_18_19[7:0]	19	Reserved	Reserved
DMA_EDMA_DREQ_19	20	CTRL_CORE_DMA_EDMA_DREQ_18_19[23:16]	20	Reserved	Reserved
DMA_EDMA_DREQ_20	21	CTRL_CORE_DMA_EDMA_DREQ_20_21[7:0]	21	Reserved	Reserved
DMA_EDMA_DREQ_21	22	CTRL_CORE_DMA_EDMA_DREQ_20_21[23:16]	22	Reserved	Reserved
DMA_EDMA_DREQ_22	23	CTRL_CORE_DMA_EDMA_DREQ_22_23[7:0]	23	MCSPI3_DREQ_TX1	McSPI3 transmit request channel 1
DMA_EDMA_DREQ_23	24	CTRL_CORE_DMA_EDMA_DREQ_22_23[23:16]	24	MCSPI3_DREQ_RX1	McSPI3 receive request channel 1
DMA_EDMA_DREQ_24	25	CTRL_CORE_DMA_EDMA_DREQ_24_25[7:0]	25	Reserved	Reserved
DMA_EDMA_DREQ_25	26	CTRL_CORE_DMA_EDMA_DREQ_24_25[23:16]	26	Reserved	Reserved
DMA_EDMA_DREQ_26	27	CTRL_CORE_DMA_EDMA_DREQ_26_27[7:0]	27	I2C1_DREQ_TX	I2C1 transmit request
DMA_EDMA_DREQ_27	28	CTRL_CORE_DMA_EDMA_DREQ_26_27[23:16]	28	I2C1_DREQ_RX	I2C1 receive request
DMA_EDMA_DREQ_28	29	CTRL_CORE_DMA_EDMA_DREQ_28_29[7:0]	29	I2C2_DREQ_TX	I2C2 transmit request
DMA_EDMA_DREQ_29	30	CTRL_CORE_DMA_EDMA_DREQ_28_29[23:16]	30	I2C2_DREQ_RX	I2C2 receive request

(1) ISS is not supported on the DRA78x family of devices.

Table 11-7. EDMA Default Request Mapping (continued)

DMA Request Line	DMA_ CROSSBAR Instance Number	DMA_CROSSBAR Configuration Register	DMA_ CROSSBAR Default Input Index	Default DMA Source Name	Default DMA Source Description
DMA_EDMA_DREQ_30	31	CTRL_CORE_DMA_EDMA_DREQ_30_31[7:0]	31	Reserved	Reserved
DMA_EDMA_DREQ_31	32	CTRL_CORE_DMA_EDMA_DREQ_30_31[23:16]	32	Reserved	Reserved
DMA_EDMA_DREQ_32	33	CTRL_CORE_DMA_EDMA_DREQ_32_33[7:0]	33	Reserved	Reserved
DMA_EDMA_DREQ_33	34	CTRL_CORE_DMA_EDMA_DREQ_32_33[23:16]	34	Reserved	Reserved
DMA_EDMA_DREQ_34	35	CTRL_CORE_DMA_EDMA_DREQ_34_35[7:0]	35	MCSPI1_DREQ_TX0	McSPI1 transmit request channel 0
DMA_EDMA_DREQ_35	36	CTRL_CORE_DMA_EDMA_DREQ_34_35[23:16]	36	MCSPI1_DREQ_RX0	McSPI1 receive request channel 0
DMA_EDMA_DREQ_36	37	CTRL_CORE_DMA_EDMA_DREQ_36_37[7:0]	37	MCSPI1_DREQ_TX1	McSPI1 transmit request channel 1
DMA_EDMA_DREQ_37	38	CTRL_CORE_DMA_EDMA_DREQ_36_37[23:16]	38	MCSPI1_DREQ_RX1	McSPI1 receive request channel 1
DMA_EDMA_DREQ_38	39	CTRL_CORE_DMA_EDMA_DREQ_38_39[7:0]	39	MCSPI1_DREQ_TX2	McSPI1 transmit request channel 2
DMA_EDMA_DREQ_39	40	CTRL_CORE_DMA_EDMA_DREQ_38_39[23:16]	40	MCSPI1_DREQ_RX2	McSPI1 receive request channel 2
DMA_EDMA_DREQ_40	41	CTRL_CORE_DMA_EDMA_DREQ_40_41[7:0]	41	MCSPI1_DREQ_TX3	McSPI1 transmit request channel 3
DMA_EDMA_DREQ_41	42	CTRL_CORE_DMA_EDMA_DREQ_40_41[23:16]	42	MCSPI1_DREQ_RX3	McSPI1 receive request channel 3
DMA_EDMA_DREQ_42	43	CTRL_CORE_DMA_EDMA_DREQ_42_43[7:0]	43	MCSPI2_DREQ_TX0	McSPI2 transmit request channel 0
DMA_EDMA_DREQ_43	44	CTRL_CORE_DMA_EDMA_DREQ_42_43[23:16]	44	MCSPI2_DREQ_RX0	McSPI2 receive request channel 0
DMA_EDMA_DREQ_44	45	CTRL_CORE_DMA_EDMA_DREQ_44_45[7:0]	45	MCSPI2_DREQ_TX1	McSPI2 transmit request channel 1
DMA_EDMA_DREQ_45	46	CTRL_CORE_DMA_EDMA_DREQ_44_45[23:16]	46	MCSPI2_DREQ_RX1	McSPI2 receive request channel 1
DMA_EDMA_DREQ_46	47	CTRL_CORE_DMA_EDMA_DREQ_46_47[7:0]	47	Reserved	Reserved
DMA_EDMA_DREQ_47	48	CTRL_CORE_DMA_EDMA_DREQ_46_47[23:16]	48	Reserved	Reserved
DMA_EDMA_DREQ_48	49	CTRL_CORE_DMA_EDMA_DREQ_48_49[7:0]	49	UART1_DREQ_TX	UART1 transmit request
DMA_EDMA_DREQ_49	50	CTRL_CORE_DMA_EDMA_DREQ_48_49[23:16]	50	UART1_DREQ_RX	UART1 receive request
DMA_EDMA_DREQ_50	51	CTRL_CORE_DMA_EDMA_DREQ_50_51[7:0]	51	UART2_DREQ_TX	UART2 transmit request
DMA_EDMA_DREQ_51	52	CTRL_CORE_DMA_EDMA_DREQ_50_51[23:16]	52	UART2_DREQ_RX	UART2 receive request
DMA_EDMA_DREQ_52	53	CTRL_CORE_DMA_EDMA_DREQ_52_53[7:0]	53	UART3_DREQ_TX	UART3 transmit request
DMA_EDMA_DREQ_53	54	CTRL_CORE_DMA_EDMA_DREQ_52_53[23:16]	54	UART3_DREQ_RX	UART3 receive request
DMA_EDMA_DREQ_54	55	CTRL_CORE_DMA_EDMA_DREQ_54_55[7:0]	55	Reserved	Reserved
DMA_EDMA_DREQ_55	56	CTRL_CORE_DMA_EDMA_DREQ_54_55[23:16]	56	Reserved	Reserved
DMA_EDMA_DREQ_56	57	CTRL_CORE_DMA_EDMA_DREQ_56_57[7:0]	57	Reserved	Reserved
DMA_EDMA_DREQ_57	58	CTRL_CORE_DMA_EDMA_DREQ_56_57[23:16]	58	Reserved	Reserved
DMA_EDMA_DREQ_58	59	CTRL_CORE_DMA_EDMA_DREQ_58_59[7:0]	59	Reserved	Reserved
DMA_EDMA_DREQ_59	60	CTRL_CORE_DMA_EDMA_DREQ_58_59[23:16]	60	Reserved	Reserved
DMA_EDMA_DREQ_60	61	CTRL_CORE_DMA_EDMA_DREQ_60_61[7:0]	61	MMC_DREQ_TX	MMC transmit request
DMA_EDMA_DREQ_61	62	CTRL_CORE_DMA_EDMA_DREQ_60_61[23:16]	62	MMC_DREQ_RX	MMC receive request

Table 11-7. EDMA Default Request Mapping (continued)

DMA Request Line	DMA_ CROSSBAR Instance Number	DMA_CROSSBAR Configuration Register	DMA_ CROSSBAR Default Input Index	Default DMA Source Name	Default DMA Source Description
DMA_EDMA_DREQ_62	63	CTRL_CORE_DMA_EDMA_DREQ_62_63[7:0]	63	UART5_DREQ_TX	UART5 transmit request
DMA_EDMA_DREQ_63	64	CTRL_CORE_DMA_EDMA_DREQ_62_63[23:16]	64	UART5_DREQ_RX	UART5 receive request

11.3.2 Mapping of DMA Requests to DMA_CROSSBAR Inputs

NOTE: The information about the DMA_CROSSBAR module, see in the Section 13.4.6.5*DMA_CROSSBAR Module Functional Description* from the *Control Module* in the device TRM.

Table 11-8 lists the DMA_CROSSBAR inputs request mapping.

NOTE: All DREQs from column “Device Module DREQs” can also be mapped to an input of the device ESM using dedicated DMA_CROSSBAR register. For more information, see *IRQ_CROSSBAR Registers Associated With The Device ESM* in *Control Module*

NOTE: ESM is not supported on the DRA78x family of devices.

Table 11-8. Connection of The Device DREQs to The DMA_CROSSBAR Inputs

DMA_CROSSBAR Input	Device Module DREQs	Description
DMA_CROSSBAR_0	Reserved	Reserved
DMA_CROSSBAR_1	Reserved	Reserved
DMA_CROSSBAR_2	EXT_SYS_DREQ_0	External DMA request 0 (system expansion) - coming from SOC IOs. Level sensitive only
DMA_CROSSBAR_3	EXT_SYS_DREQ_1	External DMA request 1 (system expansion) - coming from SOC IOs. Level sentitive only
DMA_CROSSBAR_4	GPMC_DREQ	GPMC request from prefetch engine
DMA_CROSSBAR_5	Reserved	Reserved
DMA_CROSSBAR_6	DISPC_DREQ	The line trigger signal to synchronize a memory to memory logical channel in the DMA is generated by the Display Controller IP.
DMA_CROSSBAR_7	CT_TBR_DREQ	DMA request coming from CT_TBR in DEBUGSS (used to be External DMA request 2 - coming from SOC IOs)
DMA_CROSSBAR_8	Reserved	Reserved
DMA_CROSSBAR_9	ISS_DREQ_1	ISS DMA request 1[(1)]
DMA_CROSSBAR_10	ISS_DREQ_2	ISS DMA request 2[(1)]
DMA_CROSSBAR_11	Reserved	Reserved
DMA_CROSSBAR_12	ISS_DREQ_3	ISS DMA request 3[(1)]
DMA_CROSSBAR_13	ISS_DREQ_4	ISS DMA request 4[(1)]
DMA_CROSSBAR_14	Reserved	Reserved
DMA_CROSSBAR_15	MCSPI3_DREQ_TX0	McSPI module 3 - transmit request channel 0
DMA_CROSSBAR_16	MCSPI3_DREQ_RX0	McSPI module 3 - receive request channel 0
DMA_CROSSBAR_17 to DMA_CROSSBAR_22	Reserved	Reserved
DMA_CROSSBAR_23	MCSPI3_DREQ_TX1	McSPI module 3 - transmit request channel 1
DMA_CROSSBAR_24	MCSPI3_DREQ_RX1	McSPI module 3 - receive request channel 1
DMA_CROSSBAR_25	Reserved	Reserved
DMA_CROSSBAR_26	Reserved	Reserved
DMA_CROSSBAR_27	I2C1_DREQ_TX	I2C module 1 - transmit request
DMA_CROSSBAR_28	I2C1_DREQ_RX	I2C module 1 - receive request
DMA_CROSSBAR_29	I2C2_DREQ_TX	I2C module 2 - transmit request
DMA_CROSSBAR_30	I2C2_DREQ_RX	I2C module 2 - receive request
DMA_CROSSBAR_31 to DMA_CROSSBAR_34	Reserved	Reserved
DMA_CROSSBAR_35	MCSPI1_DREQ_TX0	McSPI module 1 - transmit request channel 0
DMA_CROSSBAR_36	MCSPI1_DREQ_RX0	McSPI module 1 - receive request channel 0
DMA_CROSSBAR_37	MCSPI1_DREQ_TX1	McSPI module 1 - transmit request channel 1
DMA_CROSSBAR_38	MCSPI1_DREQ_RX1	McSPI module 1 - receive request channel 1
DMA_CROSSBAR_39	MCSPI1_DREQ_TX2	McSPI module 1 - transmit request channel 2
DMA_CROSSBAR_40	MCSPI1_DREQ_RX2	McSPI module 1 - receive request channel 2
DMA_CROSSBAR_41	MCSPI1_DREQ_TX3	McSPI module 1 - transmit request channel 3
DMA_CROSSBAR_42	MCSPI1_DREQ_RX3	McSPI module 1 - receive request channel 3
DMA_CROSSBAR_43	MCSPI2_DREQ_TX0	McSPI module 2 - transmit request channel 0
DMA_CROSSBAR_44	MCSPI2_DREQ_RX0	McSPI module 2 - receive request channel 0
DMA_CROSSBAR_45	MCSPI2_DREQ_TX1	McSPI module 2 - transmit request channel 1
DMA_CROSSBAR_46	MCSPI2_DREQ_RX1	McSPI module 2 - receive request channel 1
DMA_CROSSBAR_47 to DMA_CROSSBAR_48	Reserved	Reserved
DMA_CROSSBAR_49	UART1_DREQ_TX	UART module 1 - transmit request

(1) ISS and CRC are not supported on the DRA78x family of devices.

Table 11-8. Connection of The Device DREQs to The DMA_CROSSBAR Inputs (continued)

DMA_CROSSBAR Input	Device Module DREQs	Description
DMA_CROSSBAR_50	UART1_DREQ_RX	UART module 1 - receive request
DMA_CROSSBAR_51	UART2_DREQ_TX	UART module 2 - transmit request
DMA_CROSSBAR_52	UART2_DREQ_RX	UART module 2 - receive request
DMA_CROSSBAR_53	UART3_DREQ_TX	UART module 3 - transmit request
DMA_CROSSBAR_54	UART3_DREQ_RX	UART module 3 - receive request
DMA_CROSSBAR_55 to DMA_CROSSBAR_60	Reserved	Reserved
DMA_CROSSBAR_61	MMC_DREQ_TX	MMC/SD1 transmit request
DMA_CROSSBAR_62	MMC_DREQ_RX	MMC/SD1 receive request
DMA_CROSSBAR_63 to DMA_CROSSBAR_69	Reserved	Reserved
DMA_CROSSBAR_70	MCSPI4_DREQ_TX0	McSPI module 4 - transmit request channel 0
DMA_CROSSBAR_71	MCSPI4_DREQ_RX0	McSPI module 4 - receive request channel 0
DMA_CROSSBAR_72 to DMA_CROSSBAR_127	Reserved	Reserved
DMA_CROSSBAR_128	McASP1_DREQ_RX	McASP1 receive event
DMA_CROSSBAR_129	McASP1_DREQ_TX	McASP1 transmit event
DMA_CROSSBAR_130 to DMA_CROSSBAR_157	Reserved	Reserved
DMA_CROSSBAR_158	DCAN_DREQ_IF1	DCAN IF1 Event
DMA_CROSSBAR_159	DCAN_DREQ_IF2	DCAN IF2 Event
DMA_CROSSBAR_160	DCAN_DREQ_IF3	DCAN IF3 Event
DMA_CROSSBAR_161	MCAN_DREQ_TX	MCAN TX DMA Event
DMA_CROSSBAR_162	MCAN_DREQ_RX_FE1	MCAN RX Filter Event 1
DMA_CROSSBAR_163	MCAN_DREQ_RX_FE2	MCAN RX Filter Event 2
DMA_CROSSBAR_164 to DMA_CROSSBAR_166	Reserved	Reserved
DMA_CROSSBAR_167	EXT_SYS_DREQ_2	External DMA request 2 (system expansion) - coming from SOC IOs. Level sensitive only
DMA_CROSSBAR_168	EXT_SYS_DREQ_3	External DMA request 3 (system expansion) - coming from SOC IOs. Level sentitive only
DMA_CROSSBAR_169	MCSPI2_DREQ_TX2	McSPI module 2 - transmit request channel 2
DMA_CROSSBAR_170	MCSPI2_DREQ_RX2	McSPI module 2 - receive request channel 2
DMA_CROSSBAR_171	MCSPI2_DREQ_TX3	McSPI module 2 - transmit request channel 3
DMA_CROSSBAR_172	MCSPI2_DREQ_RX3	McSPI module 2 - receive request channel 3
DMA_CROSSBAR_173	MCSPI3_DREQ_TX2	McSPI module 3 - transmit request channel 2
DMA_CROSSBAR_174	MCSPI3_DREQ_RX2	McSPI module 3 - receive request channel 2
DMA_CROSSBAR_175	MCSPI3_DREQ_TX3	McSPI module 3 - transmit request channel 3
DMA_CROSSBAR_176	MCSPI3_DREQ_RX3	McSPI module 3 - receive request channel 3
DMA_CROSSBAR_177	MCSPI4_DREQ_TX1	McSPI module 4 - transmit request channel 1
DMA_CROSSBAR_178	MCSPI4_DREQ_RX1	McSPI module 4 - receive request channel 1
DMA_CROSSBAR_179	MCSPI4_DREQ_TX2	McSPI module 4 - transmit request channel 2
DMA_CROSSBAR_180	MCSPI4_DREQ_RX2	McSPI module 4 - receive request channel 2
DMA_CROSSBAR_181	MCSPI4_DREQ_TX3	McSPI module 4 - transmit request channel 3
DMA_CROSSBAR_182	MCSPI4_DREQ_RX3	McSPI module 4 - receive request channel 3
DMA_CROSSBAR_183 to DMA_CROSSBAR_186	Reserved	Reserved
DMA_CROSSBAR_187	GPIO1_DREQ_EVT	GPIO module 1 - event/interrupt 1
DMA_CROSSBAR_188	GPIO2_DREQ_EVT	GPIO module 2 - event/interrupt 1
DMA_CROSSBAR_189	GPIO3_DREQ_EVT	GPIO module 3 - event/interrupt 1

Table 11-8. Connection of The Device DREQs to The DMA_CROSSBAR Inputs (continued)

DMA_CROSSBAR Input	Device Module DREQs	Description
DMA_CROSSBAR_190	GPIO4_DREQ_EVT	GPIO module 4 - event/interrupt 1
DMA_CROSSBAR_191 to DMA_CROSSBAR_194	Reserved	Reserved
DMA_CROSSBAR_195	PWMSS_DREQ_ePWM0_EVT	eHRPWM0 event/interrupt
DMA_CROSSBAR_196 to DMA_CROSSBAR_197	Reserved	Reserved
DMA_CROSSBAR_198	PWMSS_DREQ_eQEP0_EVT	eQEP0 event/interrupt
DMA_CROSSBAR_199 to DMA_CROSSBAR_200	Reserved	Reserved
DMA_CROSSBAR_201	PWMSS_DREQ_eCAP0_EVT	eCAP0 event/interrupt
DMA_CROSSBAR_202 to DMA_CROSSBAR_203	Reserved	Reserved
DMA_CROSSBAR_204	CRC_DREQ_EVT0	CRC DMA event 0(1)
DMA_CROSSBAR_205	TSC_ADC_DREQ_FIFO0	ADC FIFO0 DMA request
DMA_CROSSBAR_206	TSC_ADC_DREQ_FIFO1	ADC FIFO1 DMA request
DMA_CROSSBAR_207	CRC_DREQ_EVT1	CRC DMA event 1(1)
DMA_CROSSBAR_208	CRC_DREQ_EVT2	CRC DMA event 2(1)
DMA_CROSSBAR_209	CRC_DREQ_EVT3	CRC DMA event 3(1)
DMA_CROSSBAR_210	RTI1_DREQ_EVT0	RTI DMA Event 0
DMA_CROSSBAR_211	RTI1_DREQ_EVT1	RTI DMA Event 1
DMA_CROSSBAR_212	RTI1_DREQ_EVT2	RTI DMA Event 2
DMA_CROSSBAR_213	RTI1_DREQ_EVT3	RTI DMA Event 3
DMA_CROSSBAR_214	RTI2_DREQ_EVT0	RTI DMA Event 0
DMA_CROSSBAR_215	RTI2_DREQ_EVT1	RTI DMA Event 1
DMA_CROSSBAR_216	RTI2_DREQ_EVT2	RTI DMA Event 2
DMA_CROSSBAR_217	RTI2_DREQ_EVT3	RTI DMA Event 3
DMA_CROSSBAR_218	RTI3_DREQ_EVT0	RTI DMA Event 0
DMA_CROSSBAR_219	RTI3_DREQ_EVT1	RTI DMA Event 1
DMA_CROSSBAR_220	RTI3_DREQ_EVT2	RTI DMA Event 2
DMA_CROSSBAR_221	RTI3_DREQ_EVT3	RTI DMA Event 3
DMA_CROSSBAR_222	RTI4_DREQ_EVT0	RTI DMA Event 0
DMA_CROSSBAR_223	RTI4_DREQ_EVT1	RTI DMA Event 1
DMA_CROSSBAR_224	RTI4_DREQ_EVT2	RTI DMA Event 2
DMA_CROSSBAR_225	RTI4_DREQ_EVT3	RTI DMA Event 3
DMA_CROSSBAR_226	RTI5_DREQ_EVT0	RTI DMA Event 0
DMA_CROSSBAR_227	RTI5_DREQ_EVT1	RTI DMA Event 1
DMA_CROSSBAR_228	RTI5_DREQ_EVT2	RTI DMA Event 2
DMA_CROSSBAR_229	RTI5_DREQ_EVT3	RTI DMA Event 3
DMA_CROSSBAR_230 to DMA_CROSSBAR_231	Reserved	Reserved
DMA_CROSSBAR_232	McASP2_DREQ_RX	McASP2 receive event. **Note: Only available on EDMA DREQ_54, DSP1_EDMA DREQ_16 and DSP2_EDMA DREQ_16 (connectivity restriction).**
DMA_CROSSBAR_233	McASP2_DREQ_TX	McASP2 transmit event. **Note: Only available on EDMA DREQ_55, DSP1_EDMA DREQ_17 and DSP2_EDMA DREQ_17 (connectivity restriction).**
DMA_CROSSBAR_234	McASP3_DREQ_RX	McASP3 receive event. **Note: Only available on EDMA DREQ_56, DSP1_EDMA DREQ_18 and DSP2_EDMA DREQ_18 (connectivity restriction).**

Table 11-8. Connection of The Device DREQs to The DMA_CROSSBAR Inputs (continued)

DMA_CROSSBAR Input	Device Module DREQs	Description
DMA_CROSSBAR_235	McASP3_DREQ_TX	McASP3 transmit event. **Note: Only available on EDMA DREQ_57, DSP1_EDMA DREQ_19 and DSP2_EDMA DREQ_19 (connectivity restriction).**
DMA_CROSSBAR_236 to DMA_CROSSBAR_240	Reserved	Reserved

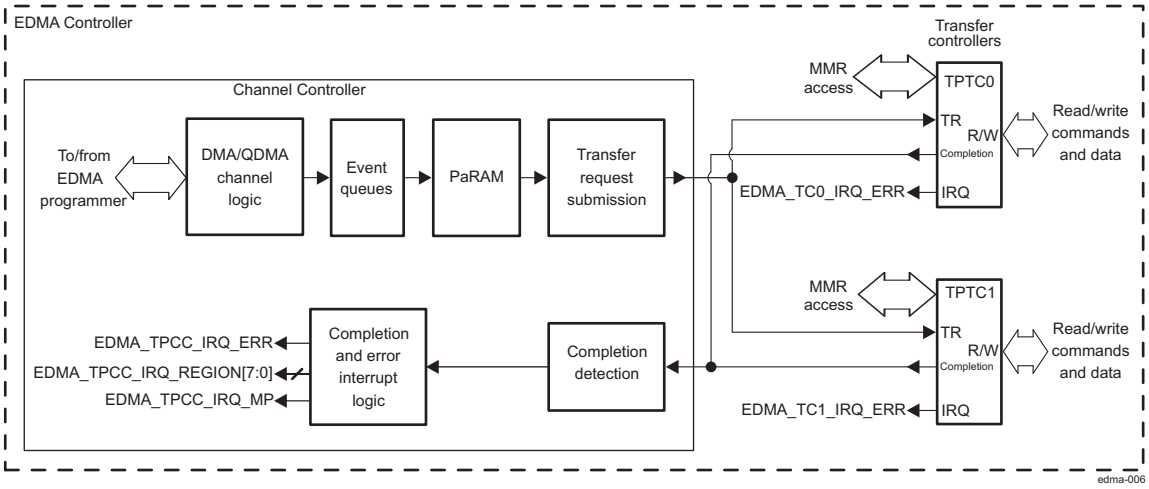
11.4 EDMA Controller Functional Description

This chapter discusses the architecture of the EDMA controller.

11.4.1 Block Diagram

Figure 11-4 shows the functional block diagram of the EDMA controller.

Figure 11-4. EDMA Controller Block Diagram



11.4.1.1 Third-Party Channel Controller

The TPCC is the EDMA transfer scheduler responsible for scheduling, arbitrating, and issuing user programmed transfers to the two TPTCs.

Figure 11-5 shows a functional block diagram of the EDMA channel controller (EDMA_TPCC).

The main blocks of the EDMA_TPCC are as follows:

- Parameter RAM (PaRAM): The PaRAM maintains parameter sets for channel and reload parameter sets. The PaRAM must be written with the transfer context for the desired channels and link parameter sets. EDMA_TPCC processes and sets based on a trigger event and submits a transfer request (TR) to the transfer controllers.
- EDMA event and interrupt processing registers: Allows mapping of events to parameter sets, enable/disable events, enable/disable interrupt conditions, and clearing interrupts.
- Completion detection: The completion detect block detects completion of transfers by the EDMA_TPTCs or slave peripherals. The completion of transfers can be used optionally to chain trigger new transfers or to assert interrupts.
- Event queues: Event queues form the interface between the event detection logic and the transfer request submission logic.
- Memory protection registers: Memory protection registers define the accesses (privilege level and requestor(s)) that are allowed to access the DMA channel shadow region view(s) and regions of PaRAM.

Other functions include the following:

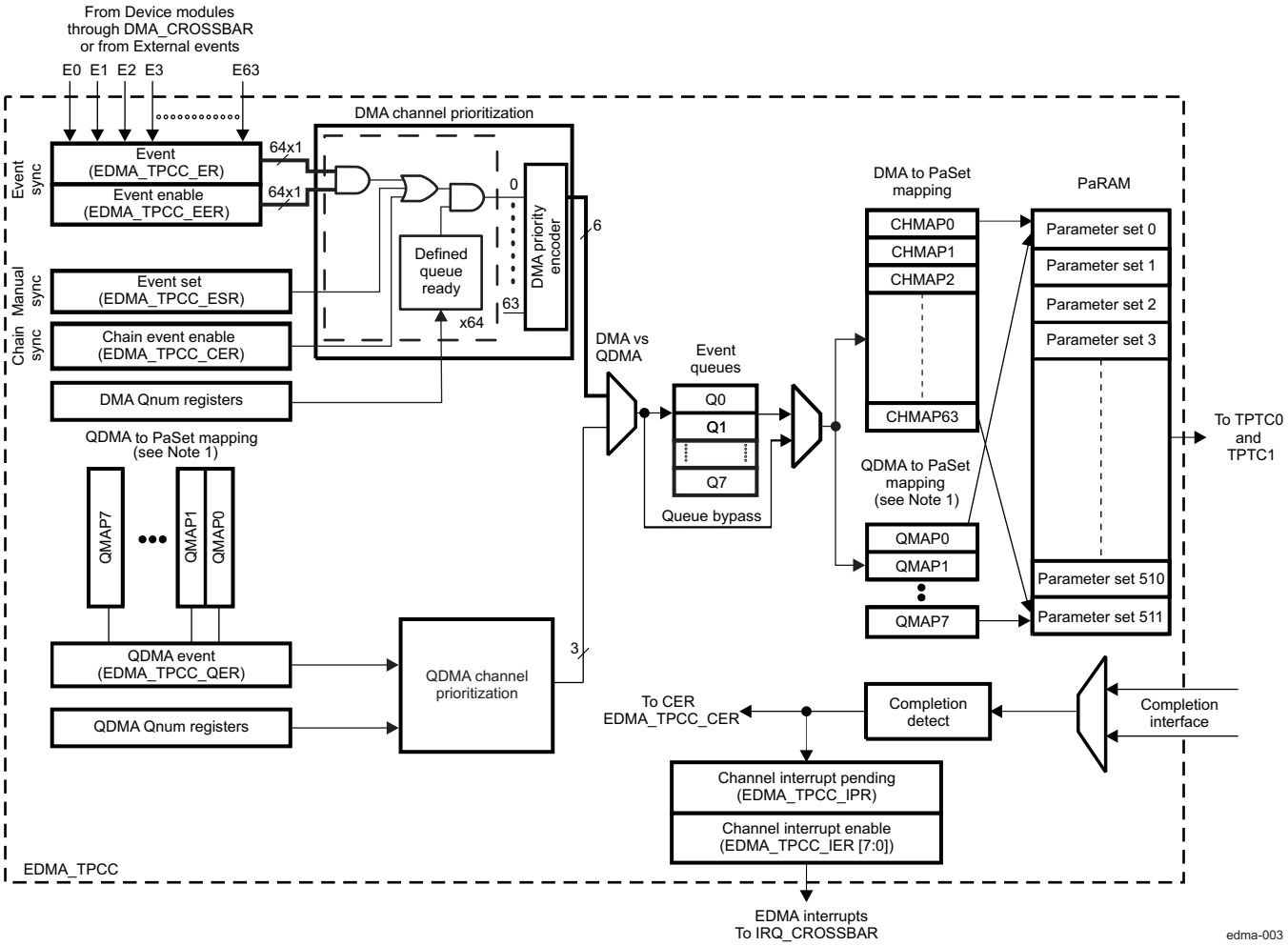
- Region registers: Region registers allow DMA resources (DMA channels and interrupts) to be assigned to unique regions that different EDMA programmers own (for example, DSPs).
- Debug registers: Debug registers allow debug visibility by providing registers to read the queue status, controller status, and missed event status.

The EDMA_TPCC includes two channel types: DMA channels (64 channels) and QDMA channels (8 channels).

Each channel is associated with a given event queue/transfer controller and with a given PaRAM set. The main difference between a DMA channel and a QDMA channel is the method that the system uses to trigger transfers.

Figure 11-5 is a block diagram of the EDMA_TPCC.

Figure 11-5. EDMA Channel Controller Block Diagram



(1) Although it is depicted twice in Figure 11-5, there is only one physical register set for the QDMA to PaRAM set mapping block.

The EDMA_TPCC supports up to 64 DMA channels and up to 8 QDMA channels. These channels are identical, except for how they are triggered:

- DMA channels are triggered by external events (such as McSPI modules TX event and McSPI modules RX event) by the event set registers EDMA_TPCC_ESR and EDMA_TPCC_ESRH, or through chaining register EDMA_TPCC_CER.
- QDMA channels are triggered automatically (auto-triggered) by the CPU. QDMAs allow a minimum number of linear writes to be issued to the TPCC to force a series of transfers to occur.

The TPCC arbitrates among pending DMA and QDMA events with a fixed [64:1] and [8:1] priority encoder for these events, respectively (a low channel number corresponds to a high priority).

DMA events are always higher priority than QDMA events. The higher-priority event is placed in the event queue to await submission to the transfer controllers, which occurs at the earliest opportunity. Each event queue is serviced in FIFO order, with a maximum of 16 queued events per event queue. If more than one TPTC is ready to be programmed with a transmission request (TR), the event queues are serviced with fixed priority: Q0 is higher than Q1. When an event is ready to be queued and the event queue and the TC channel are empty, the event bypasses the event queue and goes directly to the PaRAM processing logic for submission to the appropriate TC. If the transfer request TR bus or PaRAM processing are busy, the bypass path is not used. The bypass is not used to dequeue for a higher-priority event.

Events are extracted from the event queue when the EDMA_TPTC is available for a new TR to be programmed into the EDMA_TPTC (signaled with the empty signal, indicating an empty program register set). As an event is extracted from the event queue, the associated PaRAM entry is processed and submitted to the TPTC as a TR. The TPCC updates the appropriate counts and addresses in the PaRAM entry in anticipation of the next trigger event for that PaRAM entry.

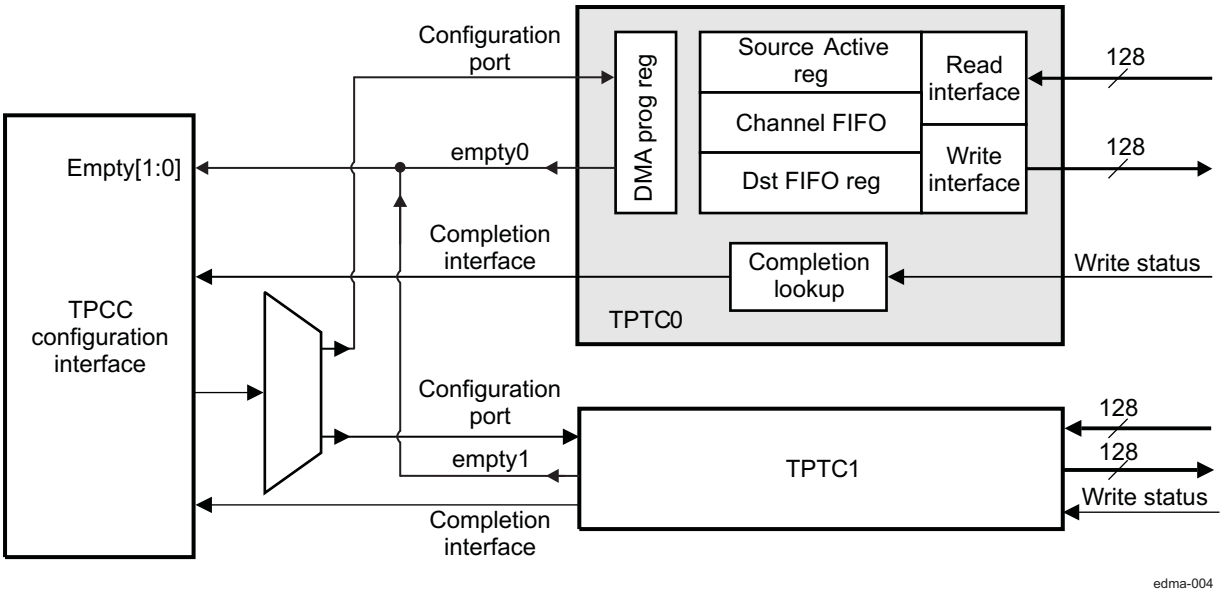
The EDMA_TPCC also has an error detection logic that causes an error interrupt generation on various error conditions (for example: missed events EDMA_TPCC_EMR and EDMA_TPCC_EMRH registers, exceeding event queue thresholds in EDMA_TPCC_CCERR register, etc.).

11.4.1.2 Third-Party Transfer Controller

The TPTC module is the EDMA transfer engine that generates transfers as programmed in dedicated working registers, using two dedicated master ports: a read-only port and a write-only port.

Figure 11-6 shows a functional block diagram and of the EDMA transfer controller (EDMA_TPTC) and its connection to the EDMA_TPCC.

Figure 11-6. TPTC Block Diagram



NOTE: The port data bus width of the instances of the TPTC is fixed at 128 bits.

Two instances of the EDMA_TPTC generate concurrent traffic on the L3_MAIN interconnect. Each TC controller consists of the following components:

- DMA Program Register Set: Stores the context for the DMA transfer that is loaded into the active register set when the current active register set completes. The CPU or TPCC programs the Program Register Set, not the active register set. For typical standalone operation, the CPU programs the Program Register while the TC services the Active register set. The Program Register set includes ownership control such that CPU software and the EDMA stay synchronized relative to one another.
- Source Active Register Set : Stores the context (src/dst/cnt/etc) for the DMA Transfer Request (TR) in

progress in the Read Controller. The Active register set is split into independent Source and Destination, because the source interconnect controller and the distant interconnect controller operate independently of one another.

- Destination FIFO Register Set: Stores the context (src/dst/cnt/etc) for the DMA Transfer Request (TR) in progress, or pending, in the Write Controller. The pending register must allow the source controller to begin processing a new TR while the distant register set processes the previous TR.
- Channel FIFO: Temporary holding buffer for in-flight data. The read return data of the source peripheral is stored in the Data FIFO, and then is written to the destination peripheral by the write command/data bus.
- Read Controller/Interconnect Read Interface: The Interconnect read interface issues optimally sized read commands to the source peripheral, based on a burst size of 128 bytes and available landing space in the channel FIFO.
- Write controller/Interconnect Write interface: The local interconnect write interface issues optimally sized write commands to the destination peripheral, based on a burst size of 128 bytes and available data in the channel FIFO.
- Completion interface: sends completion codes to the EDMA_TPCC when a transfer completes and generates interrupts and chained events in the TPCC module.
- Configuration port: Slave interface that provides read/write access to program registers and read access to all memory-mapped TPTC registers.

When one EDMA_TPTC module is idle and receive its first TR, DMA program register set receives the TR, where it transitions to the DMA source active set and the destination FIFO register set immediately. The second TR (if pending from EDMA_TPCC) is loaded into the DMA program set, ensuring it can start as soon as possible when the active transfer completes. As soon as the current active set is exhausted, the TR is loaded from the DMA program register set into the DMA source active register set as well as to the appropriate entry in the destination FIFO register set.

The read controller issues read commands controlled by the rules of command fragmentation and optimization. These are issued only when the data FIFO has space available for the data read. When sufficient data is in the data FIFO, the write controller starts issuing a write command again following the rules for command fragmentation and optimization.

Depending on the number of entries, the read controller can process up to two or four transfer requests ahead of the destination subject to the amount of free data FIFO.

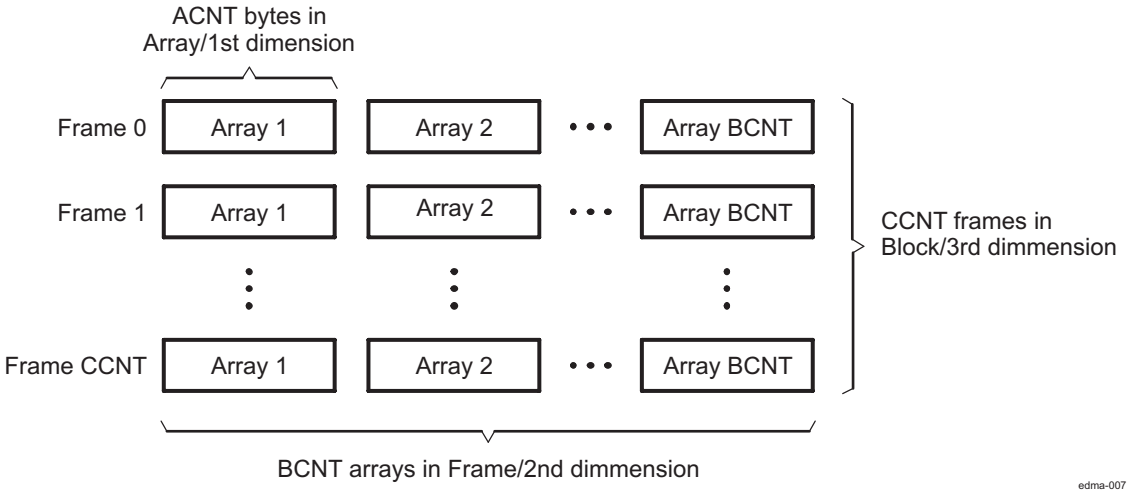
11.4.2 Types of EDMA controller Transfers

An EDMA transfer is always defined in terms of three dimensions. Figure 11-7 shows the three dimensions used by EDMA controller transfers. These three dimensions are defined as:

- 1st Dimension or Array (A): The 1st dimension in a transfer consists of EDMA_TPCC_ABCNT_n[15:0] ACNT contiguous bytes.
- 2nd Dimension or Frame (B): The 2nd dimension in a transfer consists of EDMA_TPCC_ABCNT_n[31:16] BCNT arrays of ACNT bytes. Each array transfer in the 2nd dimension is separated from each other by an index programmed using bit-fields EDMA_TPCC_BIDX_n[15:0] SBIDX or EDMA_TPCC_BIDX_n[31:16] DBIDX.
- 3rd Dimension or Block (C): The 3rd dimension in a transfer consists of CCNT frames of BCNT arrays of ACNT bytes. The Count for 3rd Dimension is defined in register EDMA_TPCC_CCNT_n[15:0] CCNT. Each transfer in the 3rd dimension is separated from the previous by an index programmed using EDMA_TPCC_CIDX_n[15:0] SCIDX or EDMA_TPCC_CIDX_n[31:16] DCIDX.

NOTE: The reference point for the index depends on the synchronization type. The amount of data transferred upon receipt of a trigger/synchronization event is controlled by the synchronization types (EDMA_TPCC_OPT_n[2] SYNCDIM bit). For these three dimensions, only two synchronization types are supported: A-synchronized transfers and AB-synchronized transfers.

Figure 11-7. Definition of ACNT, BCNT, and CCNT



11.4.2.1 A-Synchronized Transfers

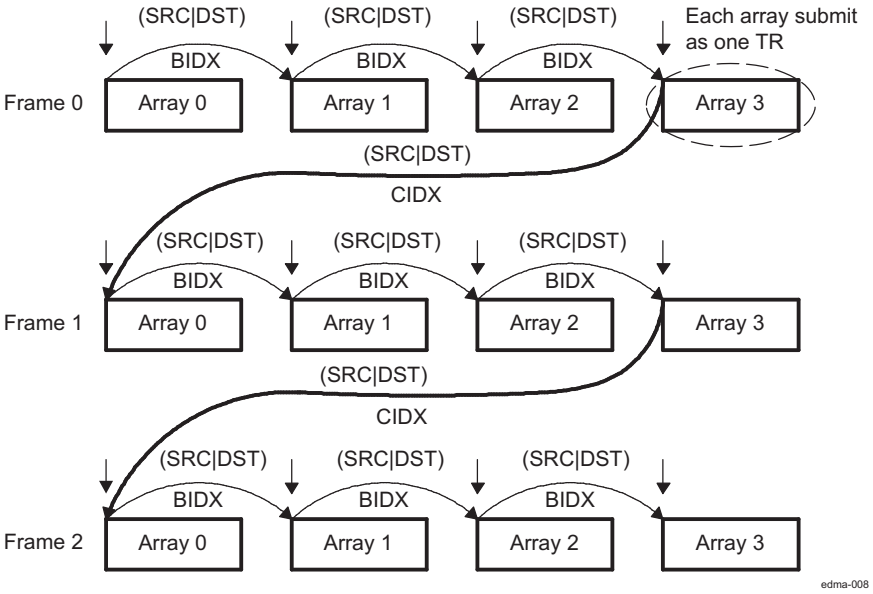
In an A-synchronized transfer, each EDMA sync event initiates the transfer of the 1st dimension of EDMA_TPCC_ABCNT_n[15:0] ACNT bytes, or one array of ACNT bytes. Each event/TR packet conveys the transfer information for one array only. Thus, BCNT × CCNT events are needed to completely service a PaRAM set.

Arrays are always separated by EDMA_TPCC_BIDX_n[15:0]SBIDX and EDMA_TPCC_BIDX_n[31:16] DBIDX, as shown in Figure 11-8, where the start address of Array N is equal to the start address of Array N – 1 plus source (SRC) or destination (DST) in EDMA_TPCC_BIDX_n register.

Frames are always separated by EDMA_TPCC_CIDX_n[15:0] SCIDX and EDMA_TPCC_CIDX_n[31:16] DCIDX. For A-synchronized transfers, after the frame is exhausted, the address is updated by adding SRCCIDX/DSTCIDX to the beginning address of the last array in the frame. As in Figure 11-8, SRCCIDX / DSTCIDX is the difference between the start of Frame 0 Array 3 to the start of Frame 1 Array 0.

Figure 11-8 shows an A-synchronized transfer of 3 (CCNT) frames of 4 (BCNT) arrays of n (ACNT) bytes. In this example, a total of 12 sync events (BCNT × CCNT) exhaust a PaRAM set. See Figure 11-8 for details on parameter set updates.

Figure 11-8. A-Synchronized Transfers (ACNT = n, BCNT = 4, CCNT = 3)



11.4.2.2 AB-Synchronized Transfers

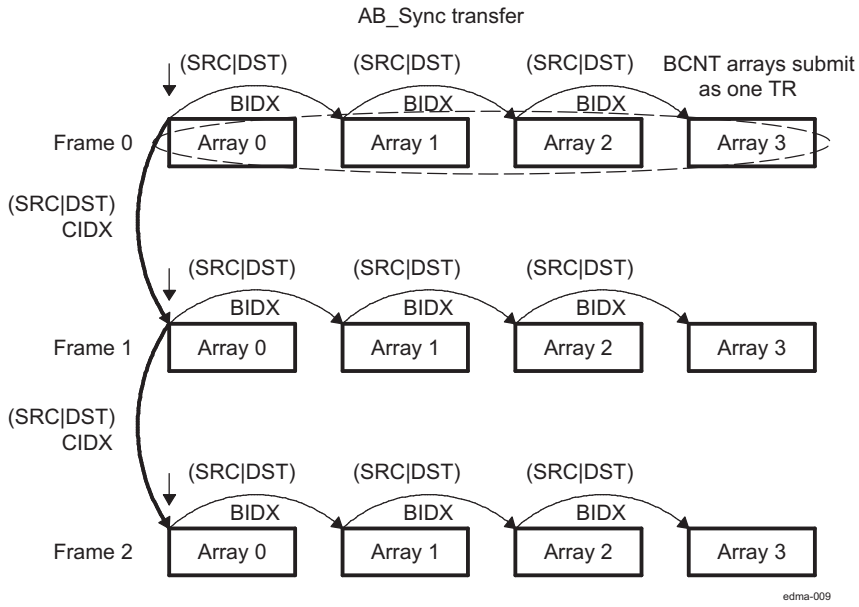
In a AB-synchronized transfer, each EDMA sync event initiates the transfer of 2 dimensions or one frame. Each event/TR packet conveys information for one entire frame of BCNT_n arrays of ACNT_n bytes. Thus, EDMA_TPCC_CCNT_n events are needed to completely service a PaRAM set.

Arrays are always separated by EDMA_TPCC_BIDX_n[15:0] SBIDX and EDMA_TPCC_BIDX_n[31:16] DBIDX as shown in Figure 11-9. Frames are always separated by SRCCIDX and DSTCIDX.

Note that for AB-synchronized transfers, after a TR for the frame is submitted, the address update is to add EDMA_TPCC_CIDX_n[15:0] SCIDX / EDMA_TPCC_CIDX_n[31:16] DCIDX to the beginning address of the beginning array in the frame. This is different from A-synchronized transfers where the address is updated by adding SRCCIDX/DSTCIDX to the start address of the last array in the frame. See Section 11.4.3.6 *Parameter Set Updates* for details on parameter set updates.

Figure 11-9 shows an AB-synchronized transfer of 3 (CCNT) frames of 4 (BCNT) arrays of *n* (ACNT) bytes. In this example, a total of 3 sync events (CCNT) exhaust a PaRAM set; that is, a total of 3 transfers of 4 arrays each completes the transfer.

Figure 11-9. AB-Synchronized Transfers (ACNT = n, BCNT = 4, CCNT = 3)



NOTE: ABC-synchronized transfers are not directly supported. It can be logically achieved by chaining between multiple AB-synchronized transfers.

11.4.3 Parameter RAM (PaRAM)

The EDMA controller is a RAM-based architecture. The transfer context (source/destination addresses, count, indexes, etc.) for DMA or QDMA channels is programmed in a parameter RAM table in EDMA_TPCC. The PaRAM table is segmented into multiple PaRAM sets. Each PaRAM set includes eight four-byte PaRAM set entries (32-bytes total per PaRAM set), which includes typical DMA transfer parameters such as source address, destination address, transfer counts, indexes, options, etc.

The PaRAM structure supports flexible ping-pong, circular buffering, channel chaining, and auto-reloading (linking).

The contents of the PaRAM include the following:

- 512 PaRAM sets
- 64 channels that are direct mapped and can be used as link or QDMA sets if not used for DMA channels

- 64 channels remain for link or QDMA sets

By default, all channels map to PaRAM set to 0, they should be remapped before use by EDMA_TPCC_DCHMAPN_m and EDMA_TPCC_QCHMAPN_j registers.

Table 11-9. EDMA Parameter RAM Contents

PaRAM Set Number	Base Address	Parameters[1]
0	EDMA Base Address + 4000h to EDMA Base Address + 401Fh	PaRAM set 0
1	EDMA Base Address + 4020h to EDMA Base Address + 403Fh	PaRAM set 1
2	EDMA Base Address + 4040h to EDMA Base Address + 405Fh	PaRAM set 2
3	EDMA Base Address + 4060h to EDMA Base Address + 407Fh	PaRAM set 3
4	EDMA Base Address + 4080h to EDMA Base Address + 409Fh	PaRAM set 4
5	EDMA Base Address + 40A0h to EDMA Base Address + 40BFh	PaRAM set 5
6	EDMA Base Address + 40C0h to EDMA Base Address + 40DFh	PaRAM set 6
7	EDMA Base Address + 40E0h to EDMA Base Address + 40FFh	PaRAM set 7
8	EDMA Base Address + 4100h to EDMA Base Address + 411Fh	PaRAM set 8
9	EDMA Base Address + 4120h to EDMA Base Address + 413Fh	PaRAM set 9
...	...	...
63	EDMA Base Address + 47E0h to EDMA Base Address + 47FFh	PaRAM set 63
64	EDMA Base Address + 4800h to EDMA Base Address + 481Fh	PaRAM set 64
65	EDMA Base Address + 4820h to EDMA Base Address + 483Fh	PaRAM set 65
...	...	...
510	EDMA Base Address + 7FC0h to EDMA Base Address + 7FDFh	PaRAM set 510
511	EDMA Base Address + 7FE0h to EDMA Base Address + 7FFFh	PaRAM set 511

(1) The device has 8 QDMA channels that can be mapped to any parameter set number from 0 to 511.

11.4.3.1 PaRAM

Each parameter set of PaRAM is organized into eight 32-bit words or 32 bytes, as shown in Figure 11-10 and described in Table 11-10. Each PaRAM set consists of 16-bit and 32-bit parameters.

Figure 11-10. PaRAM Set

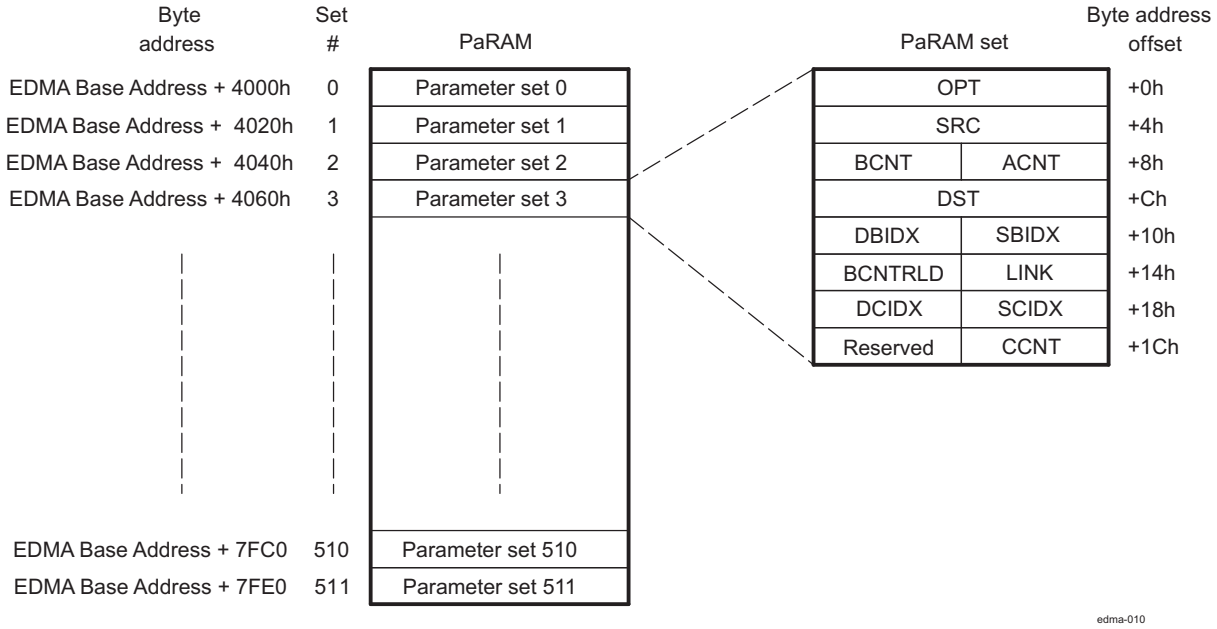


Table 11-10. EDMA Channel Parameter Description

Offset Address (bytes)	Acronym	Parameter	Description
0h	OPT	Channel Options EDMA_TPCC_OPT_n register	Transfer configuration options
4h	SRC	Channel Source Address EDMA_TPCC_SRC_n register	The byte address from which data is transferred
8h[1]	ACNT	Count for 1st Dimension EDMA_TPCC_ABCNT_n[15:0] ACNT bit-field.	Unsigned value specifying the number of contiguous bytes within an array (first dimension of the transfer). Valid values range from 1 to 65 535.
	BCNT	Count for 2nd Dimension EDMA_TPCC_ABCNT_n[31:16] BCNT bit-field.	Unsigned value specifying the number of arrays in a frame, where an array is ACNT bytes. Valid values range from 1 to 65 535.
Ch	DST	Channel Destination Address EDMA_TPCC_DST_n register	The byte address to which data is transferred
10h[1]	SBIDX	Source BCNT Index EDMA_TPCC_BIDX_n[15:0] SBIDX bit-field.	Signed value specifying the byte address offset between source arrays within a frame (2nd dimension). Valid values range from –32 768 and 32 767.
	DBIDX	Destination BCNT Index EDMA_TPCC_BIDX_n[31:16] DBIDX bit-field.	Signed value specifying the byte address offset between destination arrays within a frame (2nd dimension). Valid values range from –32 768 and 32 767.
14h[1]	LINK	Link Address EDMA_TPCC_LNK_n[15:0] LINK bit-field	The PaRAM address containing the PaRAM set to be linked (copied from) when the current PaRAM set is exhausted. A value of FFFFh specifies a null link.
	BCNTRLD	BCNT Reload EDMA_TPCC_LNK_n[31:16] BCNTRLD bit-field	The count value used to reload BCNT when BCNT decrements to 0 (TR is submitted for the last array in 2nd dimension). Only relevant in A-synchronized transfers.
18h[1]	SCIDX	Source CCNT index. EDMA_TPCC_CIDX_n[15:0] SCIDX bit-field.	Signed value specifying the byte address offset between frames within a block (3rd dimension). Valid values range from –32 768 and 32 767. A-synchronized transfers: The byte address offset from the beginning of the last source array in a frame to the beginning of the first source array in the next frame. AB-synchronized transfers: The byte address offset from the beginning of the first source array in a frame to the beginning of the first source array in the next frame.
	DCIDX	Destination CCNT index. EDMA_TPCC_CIDX_n[31:16] DCIDX bit-field.	Signed value specifying the byte address offset between frames within a block (3rd dimension). Valid values range from –32 768 and 32 767. A-synchronized transfers: The byte address offset from the beginning of the last destination array in a frame to the beginning of the first destination array in the next frame. AB-synchronized transfers: The byte address offset from the beginning of the first destination array in a frame to the beginning of the first destination array in the next frame.
1Ch	CCNT	Count for 3rd Dimension. EDMA_TPCC_CCNT_n[15:0] CCNT bit-field.	Unsigned value specifying the number of frames in a block, where a frame is BCNT arrays of ACNT bytes. Valid values range from 1 to 65 535.
	Reserved	Reserved	Reserved. Always write 0 to this bit; writes of 1 to this bit are not supported and attempts to do so may result in undefined behavior.

(1) If OPT, SRC, or DST is the trigger word for a QDMA transfer then it is required to do a 32-bit access to that field. Furthermore, it is recommended to perform only 32-bit accesses on the parameter RAM for best code compatibility. For example, switching the endianness of the processor will swap addresses of the 16-bit fields, but 32-bit accesses avoid the issue entirely.

11.4.3.2 EDMA Channel PaRAM Set Entry Fields

11.4.3.2.1 Channel Options Parameter (OPT)

For detailed information about the channel options parameter, see the EDMA_TPCC_OPT_n register description in Section 11.7.2.2.1, *EDMA_TPCC Register Description*.

11.4.3.2.2 Channel Source Address (SRC)

The 32-bit source address parameter specifies the starting byte address of the source. For SAM in increment mode, there are no alignment restrictions imposed by EDMA. For SAM in constant addressing mode, it must program the source address to be aligned to a 256-bit aligned address (5 LSBs of address must be 0). If this rule is not observed, the EDMA_TPTC returns an error. Refer to Section 11.4.12.3 *Error Generation* for additional details.

11.4.3.2.3 Channel Destination Address (DST)

The 32-bit destination address parameter specifies the starting byte address of the destination. For DAM in increment mode, there are no alignment restrictions imposed by EDMA. For DAM in constant addressing mode, it must program the destination address to be aligned to a 256-bit aligned address (5 LSBs of address must be 0). If this rule is not observed, the EDMA_TPTC returns an error. Refer to Section 11.4.12.3 *Error Generation* for additional details.

11.4.3.2.4 Count for 1st Dimension (ACNT)

EDMA_TPCC_ABCNT_n[15:0] ACNT represents the number of bytes within the 1st dimension of a transfer. ACNT is a 16-bit unsigned value with valid values between 0 and 65 535. Therefore, the maximum number of bytes in an array is 65 535 bytes (64K – 1 bytes). ACNT must be greater than or equal to 1 for a TR to be submitted to EDMA_TPTC. A transfer with ACNT equal to 0 is considered either a null or dummy transfer. A dummy or null transfer generates a completion code depending on the settings of the completion bit fields in EDMA_TPCC_OPT_n.

Refer to Section 11.4.3.5 *Dummy Versus Null Transfer Comparison* and Section 11.4.5.3 *Dummy or Null Completion* for details on dummy/null completion conditions.

11.4.3.2.5 Count for 2nd Dimension (BCNT)

EDMA_TPCC_ABCNT_n[15:0] BCNT is a 16-bit unsigned value that specifies the number of arrays of length ACNT. For normal operation, valid values for BCNT are between 1 and 65 535. Therefore, the maximum number of arrays in a frame is 65 535 (64K – 1 arrays). A transfer with BCNT equal to 0 is considered either a null or dummy transfer. A dummy or null transfer generates a completion code depending on the settings of the completion bit fields in EDMA_TPCC_OPT_n.

Refer to Section 11.4.3.5 *Dummy Versus Null Transfer Comparison* and Section 11.4.5.3 *Dummy or Null Completion* for details on dummy/null completion conditions.

11.4.3.2.6 Count for 3rd Dimension (CCNT)

EDMA_TPCC_CCNT_n[15:0] CCNT is a 16-bit unsigned value that specifies the number of frames in a block. Valid values for CCNT are between 1 and 65 535. Therefore, the maximum number of frames in a block is 65 535 (64K – 1 frames). A transfer with CCNT equal to 0 is considered either a null or dummy transfer. A dummy or null transfer generates a completion code depending on the settings of the completion bit fields in EDMA_TPCC_OPT_n.

A CCNT value of 0 is considered either a null or dummy transfer.

Refer to Section 11.4.3.5 *Dummy Versus Null Transfer Comparison* and Section 11.4.5.3 *Dummy or Null Completion* for details on dummy/null completion conditions.

11.4.3.2.7 BCNT Reload (BCNTRLD)

EDMA_TPCC_LNK_n[31:16] BCNTRLD is a 16-bit unsigned value used to reload the EDMA_TPCC_ABCNT_n[15:0] BCNT field once the last array in the 2nd dimension is transferred. This field is only used for A-synchronized transfers. In this case, the EDMA_TPCC decrements the BCNT value by 1 on each TR submission. When BCNT reaches 0, the EDMA_TPCC decrements CCNT and uses the BCNTRLD value to reinitialize the BCNT value.

For AB-synchronized transfers, the EDMA_TPCC submits the BCNT in the TR and the EDMA_TPTC decrements BCNT appropriately. For AB-synchronized transfers, BCNTRLD is not used.

11.4.3.2.8 Source B Index (SBIDX)

EDMA_TPCC_BIDX_n[15:0] SBIDX is a 16-bit signed value (2s complement) used for source address modification between each array in the 2nd dimension. Valid values for EDMA_TPCC_BIDX_n[15:0] SBIDX are between –32 768 and 32 767. It provides a byte address offset from the beginning of the source array to the beginning of the next source array. It applies to both A-synchronized and AB-synchronized transfers. Some examples:

- EDMA_TPCC_BIDX_n[15:0] SBIDX = 0000h (0): no address offset from the beginning of an array to the beginning of the next array. All arrays are fixed to the same beginning address.
- EDMA_TPCC_BIDX_n[15:0] SBIDX = 0003h (+3): the address offset from the beginning of an array to the beginning of the next array in a frame is 3 bytes. For example, if the current array begins at address 1000h, the next array begins at 1003h.
- EDMA_TPCC_BIDX_n[15:0] SBIDX = FFFFh (–1): the address offset from the beginning of an array to the beginning of the next array in a frame is –1 byte. For example, if the current array begins at address 5054h, the next array begins at 5053h.

11.4.3.2.9 Destination B Index (DBIDX)

EDMA_TPCC_BIDX_n[31:16] DBIDX is a 16-bit signed value (2s complement) used for destination address modification between each array in the 2nd dimension. Valid values for EDMA_TPCC_BIDX_n[31:16] DBIDX are between –32 768 and 32 767. It provides a byte address offset from the beginning of the destination array to the beginning of the next destination array within the current frame. It applies to both A-synchronized and AB-synchronized transfers. Refer to Section 11.4.3.2.8 *Source B Index (SBIDX)* for examples.

11.4.3.2.10 Source C Index (SCIDX)

EDMA_TPCC_CIDX_n[15:0] SCIDX is a 16-bit signed value (2s complement) used for source address modification in the 3rd dimension. Valid values for EDMA_TPCC_CIDX_n[15:0] SCIDX are between –32 768 and 32 767. It provides a byte address offset from the beginning of the current array (pointed to by SRC address) to the beginning of the first source array in the next frame. It applies to both A-synchronized and AB-synchronized transfers.

NOTE: When SCIDX is applied, the current array in an A-synchronized transfer is the last array in the frame (Figure 11-8), while the current array in an AB-synchronized transfer is the first array in the frame (Figure 11-9).

11.4.3.2.11 Destination C Index (DCIDX)

EDMA_TPCC_CIDX_n[31:16] DCIDX is a 16-bit signed value (2s complement) used for destination address modification in the 3rd dimension. Valid values are between –32 768 and 32 767. It provides a byte address offset from the beginning of the current array (pointed to by DST address) to the beginning of the first destination array TR in the next frame. It applies to both A-synchronized and AB-synchronized transfers.

NOTE: When DCIDX is applied, the current array in an A-synchronized transfer is the last array in the frame (Figure 11-8), while the current array in a AB-synchronized transfer is the first array in the frame (Figure 11-9).

11.4.3.2.12 Link Address (LINK)

The EDMA_TPCC provides a mechanism, called linking, to reload the current PaRAM set upon its natural termination (that is, after the count fields are decremented to 0) with a new PaRAM set. The 16-bit parameter EDMA_TPCC_LNK_n[15:0] LINK specifies the byte address offset in the PaRAM from which the EDMA_TPCC loads/reloads the next PaRAM set during linking.

It must program the link address to point to a valid aligned 32-byte PaRAM set. The 5 LSBs of the LINK field should be cleared to 0.

The EDMA_TPCC ignores the upper 2 bits of the LINK entry, allowing the flexibility of programming the link address as either an absolute/literal byte address or use the PaRAM-base-relative offset address. Therefore, if it use the literal address with a range from 4000h to 7FFFh, it will be treated as a PaRAM-base-relative value of 0000h to 3FFFh.

It should check that the programed value in the EDMA_TPCC_LNK_n[15:0] LINK field is correctly, so that link update is requested from a PaRAM address that falls in the range of the available PaRAM addresses on the device.

Value of FFFFh in EDMA_TPCC_LNK_n[15:0] LINK bit-field is referred to as a NULL link that should cause the EDMA_TPCC to perform an internal write of 0 to all entries of the current PaRAM set, except for the EDMA_TPCC_LNK_n[15:0] LINK field is set to FFFFh. Also, see Section 11.4.5 *Completion of a DMA Transfer* for details on terminating a transfer.

11.4.3.3 Null PaRAM Set

A null PaRAM set is defined as a PaRAM set where all count fields (EDMA_TPCC_ABCNT_n[15:0] ACNT, EDMA_TPCC_ABCNT_n[31:16] BCNT, and EDMA_TPCC_CCNT_n[15:0] CCNT) are cleared to 0. If a PaRAM set associated with a channel is a NULL set, then when serviced by the EDMA_TPCC, the bit corresponding to the channel is set in the associated event missed register (EDMA_TPCC_EMR, EDMA_TPCC_EMRH, or EDMA_TPCC_QEMR). This bit remains set in the associated secondary event register (EDMA_TPCC_SER, EDMA_TPCC_SERH, or EDMA_TPCC_QSER).
This implies that any future events on the same channel are ignored by the EDMA_TPCC and it is required to clear the bit in EDMA_TPCC_SER, EDMA_TPCC_SERH, or EDMA_TPCC_QSER for the channel. This is considered an error condition, since events are not expected on a channel that is configured as a null transfer.

11.4.3.4 Dummy PaRAM Set

A dummy PaRAM set is defined as a PaRAM set where at least one of the count fields (EDMA_TPCC_ABCNT_n[15:0] ACNT, EDMA_TPCC_ABCNT_n[31:16] BCNT, or EDMA_TPCC_CCNT_n[15:0] CCNT) is cleared to 0 and at least one of the count fields is nonzero.

If a PaRAM set associated with a channel is a dummy set, then when serviced by the EDMA_TPCC, it will not set the bit corresponding to the channel (DMA/QDMA) in the event missed register (EDMA_TPCC_EMR, EDMA_TPCC_EMRH, or EDMA_TPCC_QEMR) and the secondary event register (EDMA_TPCC_SER, EDMA_TPCC_SERH, or EDMA_TPCC_QSER) bit gets cleared similar to a normal transfer. Future events on that channel are serviced. A dummy transfer is a legal transfer of 0 bytes.

11.4.3.5 Dummy Versus Null Transfer Comparison

There are some differences in the way the EDMA_TPCC logic treats a dummy versus a null transfer request. A null transfer request is an error condition, but a dummy transfer is a legal transfer of 0 bytes. A null transfer causes an error bit (E*n*) in EDMA_TPCC_EMR to get set and the E*n* bit in EDMA_TPCC_SER remains set, essentially preventing any further transfers on that channel without clearing the associated error registers.

Table 11-11 summarizes the conditions and effects of null and dummy transfer requests.

Table 11-11. Dummy and Null Transfer Request

Feature	Null TR	Dummy TR
EDMA_TPCC_EMR / EDMA_TPCC_EMRH / EDMA_TPCC_QEMR is set	Yes	No
EDMA_TPCC_SER / EDMA_TPCC_SERH / EDMA_TPCC_QSER remains set	Yes	No
Link update (STATIC = 0 in EDMA_TPCC_OPT_n)	Yes	Yes
EDMA_TPCC_QER is set	Yes	Yes
EDMA_TPCC_IPR / EDMA_TPCC_IPRH, EDMA_TPCC_CER / EDMA_TPCC_CERH is set using early completion	Yes	Yes

11.4.3.6 Parameter Set Updates

When a TR is submitted for a given DMA/QDMA channel and its corresponding PaRAM set, the EDMA_TPCC is responsible for updating the PaRAM set in anticipation of the next trigger event. For events that are not final, this includes address and count updates; for final events, this includes the link update.

The specific PaRAM set entries that are updated depend on the channel's synchronization type (A-synchronized or B-synchronized) and the current state of the PaRAM set. A B-update refers to the decrementing of EDMA_TPCC_ABCNT_n[31:16] BCNT in the case of A-synchronized transfers after the submission of successive TRs. A C-update refers to the decrementing of CCNT in the case of A-synchronized transfers after BCNT TRs for EDMA_TPCC_ABCNT_n[15:0] ACNT byte transfers have submitted. For AB-synchronized transfers, a C-update refers to the decrementing of EDMA_TPCC_CCNT_n[15:0] CCNT after submission of every transfer request.

Refer to Table 11-12 for details and conditions on the parameter updates. A link update occurs when the PaRAM set is exhausted, as described in Section 11.4.3.7 *Linking Transfers*.

After the TR is read from the PaRAM (and is in process of being submitted to EDMA_TPTC), the following fields are updated if needed:

- A-synchronized: BCNT, CCNT, SRC, DST.
- AB-synchronized: CCNT, SRC, DST.

The following fields are not updated (except for during linking, where all fields are overwritten by the link PaRAM set):

- A-synchronized: EDMA_TPCC_ABCNT_n[15:0] ACNT, EDMA_TPCC_LNK_n[31:16] BCNTRLD, EDMA_TPCC_BIDX_n[15:0] SBIDX, EDMA_TPCC_BIDX_n[31:16] DBIDX, EDMA_TPCC_CIDX_n[15:0] SCIDX, EDMA_TPCC_CIDX_n[31:16] DCIDX, EDMA_TPCC_OPT_n, EDMA_TPCC_LNK_n[15:0]LINK.
- AB-synchronized: EDMA_TPCC_ABCNT_n[15:0] ACNT, EDMA_TPCC_ABCNT_n[31:16] BCNT, EDMA_TPCC_LNK_n[31:16] BCNTRLD, EDMA_TPCC_BIDX_n[15:0] SBIDX, EDMA_TPCC_BIDX_n[31:16] DBIDX, EDMA_TPCC_CIDX_n[15:0] SCIDX, EDMA_TPCC_CIDX_n[31:16] DCIDX, EDMA_TPCC_OPT_n, EDMA_TPCC_LNK_n[15:0]LINK.

NOTE: PaRAM updates only pertain to the information that is needed to properly submit the next transfer request to the EDMA_TPTC. Updates that occur while data is moved within a transfer request are tracked within the transfer controller, and is detailed in Section 11.4.12 *EDMA Transfer Controller (EDMA_TPTC)*. For A-synchronized transfers, the EDMA_TPCC always submits a TRP for EDMA_TPCC_ABCNT_n[15:0] ACNT bytes (EDMA_TPCC_ABCNT_n[31:16] BCNT = 1 and EDMA_TPCC_CCNT_n[15:0] CCNT = 1). For AB-synchronized transfers, the EDMA_TPCC always submits a TRP for EDMA_TPCC_ABCNT_n[15:0] ACNT bytes of BCNT arrays (EDMA_TPCC_CCNT_n[15:0] CCNT = 1). The EDMA_TPTC is responsible for updating source and destination addresses within the array based on EDMA_TPCC_ABCNT_n[15:0] ACNT and EDMA_TPCC_OPT_n[10:8] FWID. For AB-synchronized transfers, the EDMA_TPTC is also responsible to update source and destination addresses between arrays based on EDMA_TPCC_BIDX_n[15:0] SBIDX and EDMA_TPCC_BIDX_n[31:16] DBIDX.

Table 11-12 shows the details of parameter updates that occur within EDMA_TPCC for A-synchronized and AB-synchronized transfers.

Table 11-12. Parameter Updates in EDMA_TPCC (for Non-Null, Non-Dummy PaRAM Set)

	A-Synchronized Transfer			AB-Synchronized Transfer		
	B-Update	C-Update	Link Update	B-Update	C-Update	Link Update
Condition:	BCNT > 1	BCNT == 1 && CCNT > 1	BCNT == 1 && CCNT == 1	N/A	EDMA_TPCC_CCNT_n[15:0] CCNT > 1	EDMA_TPCC_CCNT_n[15:0] CCNT == 1
SRC	+= SBIDX	+= SCIDX	= Link.EDMA_TPCC_SRC_n	in EDMA_TPTC	+= SCIDX	= Link.EDMA_TPCC_SRC_n
DST	+= DBIDX	+= DCIDX	= Link.EDMA_TPCC_DST_n	in EDMA_TPTC	+= DCIDX	= Link.EDMA_TPCC_DST_n
ACNT	None	None	= Link.EDMA_TPCC_ABCNT_n[15:0] ACNT	None	None	= Link.EDMA_TPCC_ABCNT_n[15:0] ACNT
BCNT	–= 1	= BCNTRLD	= Link.EDMA_TPCC_ABCNT_n[31:16] BCNT	in EDMA_TPTC	N/A	= Link.EDMA_TPCC_ABCNT_n[31:16] BCNT
CCNT	None	–= 1	= Link.EDMA_TPCC_CCNT_n[15:0] CCNT	in EDMA_TPTC	-=1	= Link.EDMA_TPCC_CCNT_n[15:0] CCNT
SBIDX	None	None	= Link.EDMA_TPCC_BIDX_n[15:0] SBIDX	in EDMA_TPTC	None	= Link.EDMA_TPCC_BIDX_n[15:0] SBIDX
DBIDX	None	None	= Link.EDMA_TPCC_BIDX_n[31:16] DBIDX	None	None	= Link.EDMA_TPCC_BIDX_n[31:16] DBIDX
SCIDX	None	None	= Link.EDMA_TPCC_BIDX_n[15:0] SBIDX	in EDMA_TPTC	None	= Link.EDMA_TPCC_BIDX_n[15:0] SBIDX
DCIDX	None	None	= Link.EDMA_TPCC_BIDX_n[31:16] DBIDX	None	None	= Link.EDMA_TPCC_BIDX_n[31:16] DBIDX
LINK	None	None	= Link.EDMA_TPCC_LNK_n[15:0]LINK	None	None	= Link.EDMA_TPCC_LNK_n[15:0]LINK
BCNTRLD	None	None	= Link.EDMA_TPCC_LNK_n[31:16] BCNTRLD	None	None	= Link.EDMA_TPCC_LNK_n[31:16] BCNTRLD
OPT[1]	None	None	= LINK.EDMA_TPCC_OPT_n	None	None	= LINK.EDMA_TPCC_OPT_n

(1) In all cases, no updates occur if EDMA_TPCC_OPT_n[3] STATIC == 1 for the current PaRAM set.

NOTE: The EDMA_TPCC includes no special hardware to detect when an indexed address update calculation overflows/underflows. The address update will wrap across boundaries as programmed by the user. It should ensure that no transfer is allowed to cross internal port boundaries between peripherals. A single TR must target a single source/destination slave endpoint.

11.4.3.7 Linking Transfers

The EDMA_TPCC provides a mechanism known as linking, which allows the entire PaRAM set to be reloaded from a location within the PaRAM memory map (for both DMA and QDMA channels). Linking is especially useful for maintaining ping-pong buffers, circular buffering, and repetitive/continuous transfers with no CPU intervention. Upon completion of a transfer, the current transfer parameters are reloaded with the parameter set pointed that the 16-bit link address field of the current parameter set points to. Linking only occurs when the EDMA_TPCC_OPT_n[3] STATIC bit is cleared.

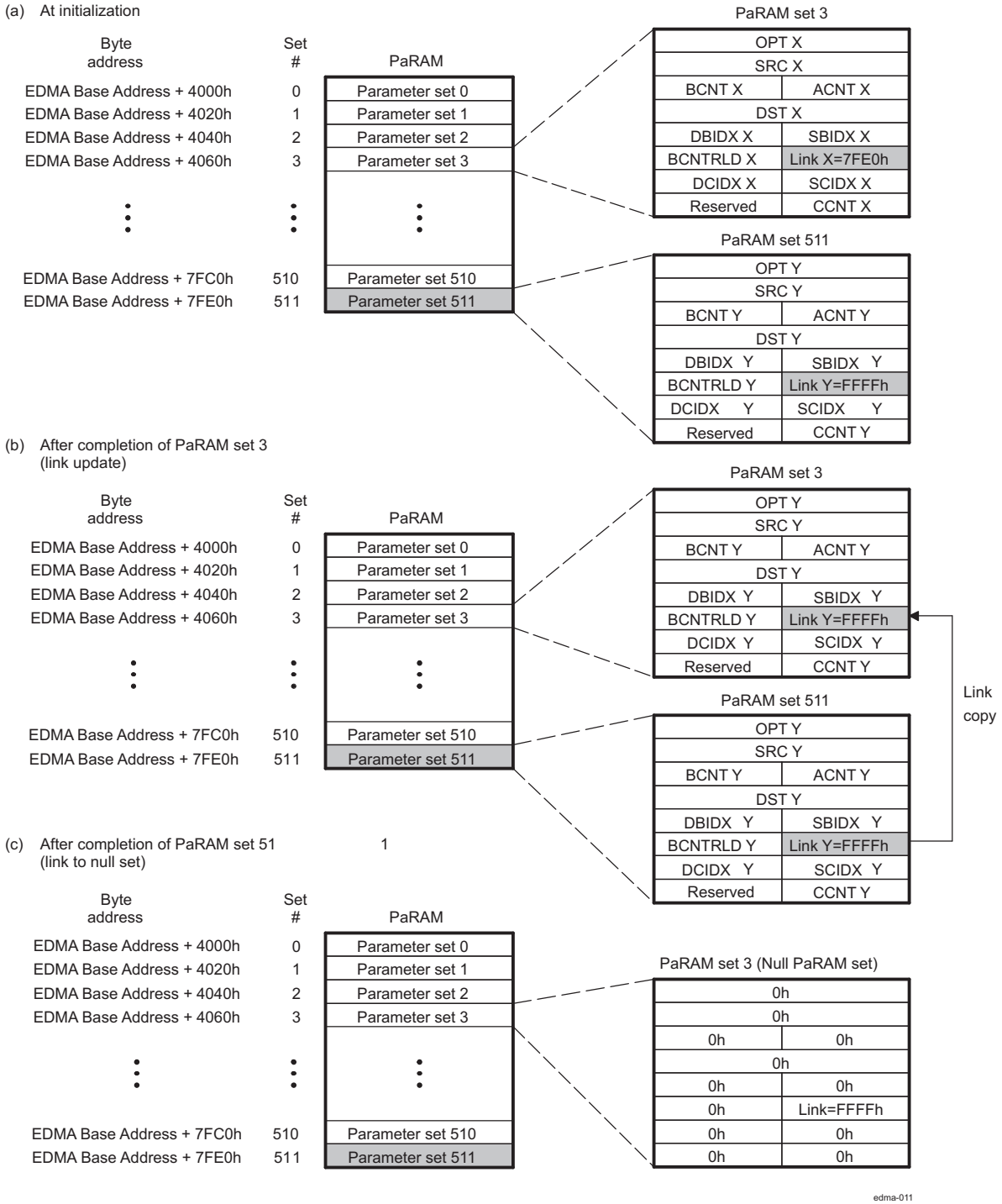
NOTE: It should always link a transfer (EDMA or QDMA) to another useful transfer. If it must terminate a transfer, then link the transfer to a NULL parameter set. Refer to Section 11.4.3.3 *Null PaRAM Set*.

The link update occurs after the current PaRAM set event parameters have been exhausted. An event's parameters are exhausted when the EDMA channel controller has submitted all of the transfers that are associated with the PaRAM set.

A link update occurs for null and dummy transfers depending on the state of the EDMA_TPCC_OPT_n[3] STATIC bit and the EDMA_TPCC_LNK_n[15:0] LINK field. In both cases (null or dummy), if the value of EDMA_TPCC_LNK_n[15:0] LINK is FFFFh, then a null PaRAM set (with all 0s and EDMA_TPCC_LNK_n[15:0] LINK set to FFFFh) is written to the current PaRAM set. Similarly, if EDMA_TPCC_LNK_n[15:0] LINK is set to a value other than FFFFh, then the appropriate PaRAM location that EDMA_TPCC_LNK_n[15:0] LINK points to is copied to the current PaRAM set.

Once the channel completion conditions are met for an event, the transfer parameters that are located at the link address are loaded into the current DMA or QDMA channel's associated parameter set. This indicates that the EDMA_TPCC reads the entire set (eight words) from the PaRAM set specified by EDMA_TPCC_LNK_n[15:0] LINK and writes all eight words to the PaRAM set that is associated with the current channel. Figure 11-11 shows an example of a linked transfer.

Figure 11-11. Linked Transfer

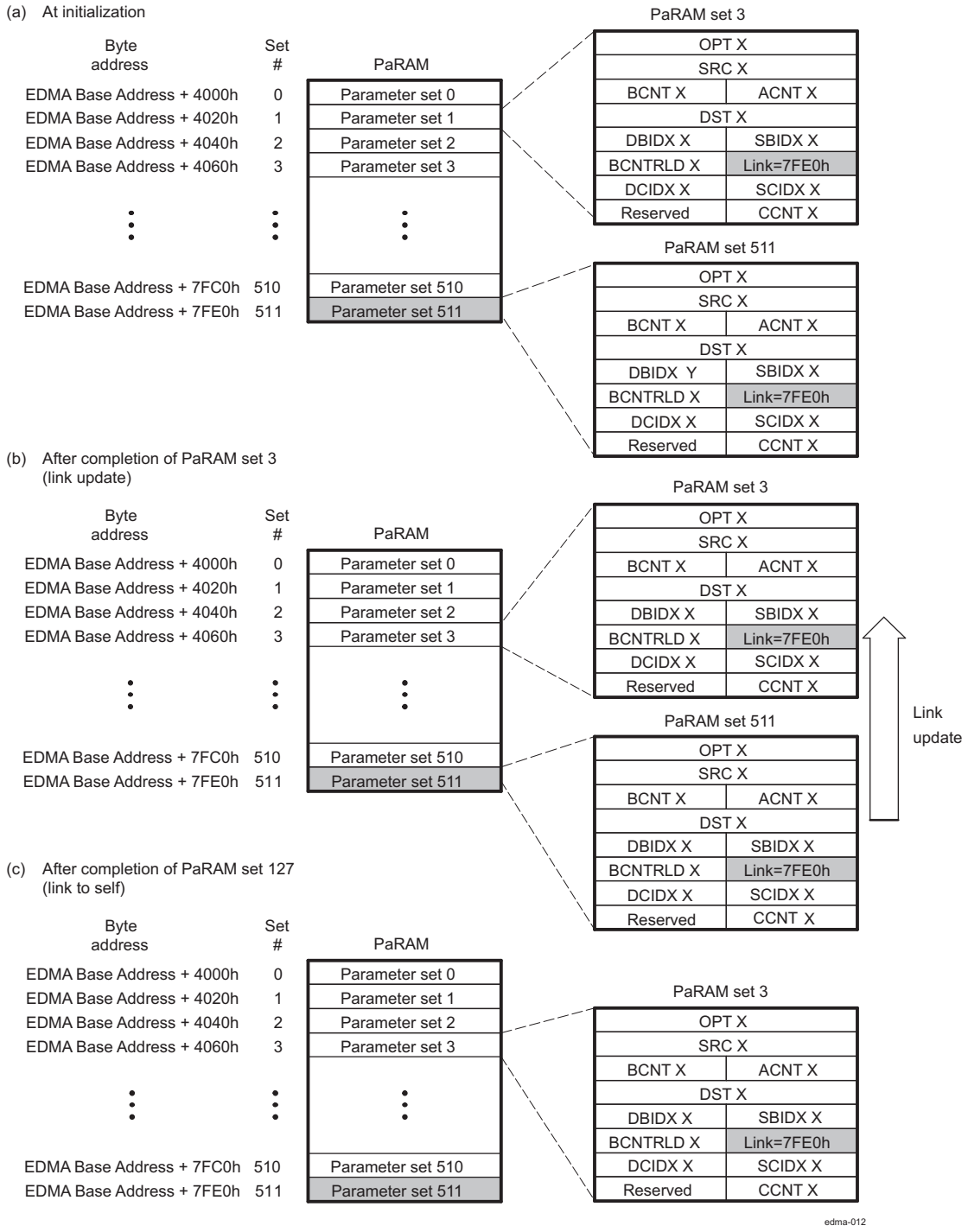


Any PaRAM set in the PaRAM can be used as a link/reload parameter set. The PaRAM sets associated with peripheral synchronization events (refer to Section 11.4.6 *Event, Channel, and PaRAM Mapping*) only use for linking if the corresponding events are disabled.

If a PaRAM set location is defined as a QDMA channel PaRAM set (by EDMA_TPCC_QCHMAPN_j register), then copying the link PaRAM set into the current QDMA channel PaRAM set is recognized as a trigger event. It is latched in EDMA_TPCC_QER because a write to the trigger word was performed. This feature is used to create a linked list of transfers using a single QDMA channel and multiple PaRAM sets. Refer to Section 11.4.4.2 *QDMA Channels*.

Linking to itself replicates the behavior of auto-initialization, thus facilitating the use of circular buffering and repetitive transfers. After an EDMA channel exhausts its current PaRAM set, it reloads all of the parameter set entries from another PaRAM set, which is initialized with values that are identical to the original PaRAM set. Figure 11-12 shows an example of a linked to self transfer. Here, the PaRAM set 511 has the link field pointing to the address of parameter set 511 (linked to self).

Figure 11-12. Link-to-Self Transfer



NOTE: If the in EDMA_TPCC_OPT_n[3] STATIC bit is set for a PaRAM set, then link updates are not performed.

11.4.3.8 Constant Addressing Mode Transfers/Alignment Issues

If either EDMA_TPCC_OPT_n[0] SAM or EDMA_TPCC_OPT_n[1] DAM is set (constant addressing mode), then the source or destination address must be aligned to a 256-bit aligned address, respectively, and the corresponding EDMA_TPCC_BIDX_n is an even multiple of 32 bytes (256 bits). The EDMA_TPCC does not recognize errors here, but the EDMA_TPTC asserts an error if this is not true. Refer to Section 11.4.12.3 *Error Generation*.

NOTE: The constant addressing (CONST) mode has limited applicability. The EDMA is configured for the constant addressing mode (EDMA_TPCC_OPT_n[0] SAM / EDMA_TPCC_OPT_n[1] DAM = 1) only if the transfer source or destination (on-chip memory, off-chip memory controllers, slave peripherals) support the constant addressing mode. If the constant addressing mode is not supported, the similar logical transfer can be achieved using the increment (INCR) mode (EDMA_TPCC_OPT_n[0] SAM / EDMA_TPCC_OPT_n[1] DAM =0) by appropriately programming the count and indices values.

11.4.3.9 Element Size

The EDMA controller does not use element-size and element-indexing. Instead, all transfers are defined in terms of all three dimensions: EDMA_TPCC_ABCNT_n[15:0] ACNT, EDMA_TPCC_ABCNT_n[31:16] BCNT, and EDMA_TPCC_CCNT_n[15:0] CCNT. An element-indexed transfer is logically achieved by programming EDMA_TPCC_ABCNT_n[15:0] ACNT to the size of the element and EDMA_TPCC_ABCNT_n[31:16] BCNT to the number of elements that need to be transferred. For example: If there are 16-bit audio data and 256 audio samples that must be transferred to a serial port, therefore the EDMA_TPCC_ABCNT_n[15:0] ACNT = 2 (2 bytes) and EDMA_TPCC_ABCNT_n[31:16] BCNT = 256.

11.4.4 Initiating a DMA Transfer

There are multiple ways to initiate a programmed data transfer using the EDMA_TPCC channel controller. Transfers on DMA channels are initiated by three sources.

They are listed as follows:

- **Event-triggered transfer request** (this is the typical usage of EDMA controller): A peripheral, system, or externally-generated event triggers a transfer request.
- **Manually-triggered transfer request:**The CPU manually triggers a transfer by writing a 1 to the corresponding bit in the event set registers (EDMA_TPCC_ESR / EDMA_TPCC_ESRH).
- **Chain-triggered transfer request:** A transfer is triggered on the completion of another transfer or sub-transfer.

Transfers on QDMA channels are initiated by two sources. They are as follows:

- **Auto-triggered transfer request:** Writing to the programmed trigger word triggers a transfer.
- **Link-triggered transfer requests:** Writing to the trigger word triggers the transfer when linking occurs.

11.4.4.1 DMA Channels

11.4.4.1.1 Event-Triggered Transfer Request

When an event is asserted from a peripheral or device pins, it gets latched in the corresponding bit of the event register (EDMA_TPCC_ER[31:0] E*n* = 1). For more information about peripheral events to EDMA events mapping, refer to *the device data manual*.
If the corresponding event in the event enable register (EDMA_TPCC_EER) is enabled (EDMA_TPCC_EER[31:0] E*n* = 1), then the EDMA_TPCC prioritizes and queues the event in the appropriate event queue. When the event reaches the head of the queue, it is evaluated for submission as a transfer request to the transfer controller.

If the PaRAM set is valid (not a NULL set), then a transfer request packet (TRP) is submitted to the EDMA_TPTC and the EDMA_TPCC_ER[31:0] E*n* bit is cleared. At this point, a new event can be safely received by the EDMA_TPCC.

If the PaRAM set associated with the channel is a NULL set (see Section 11.4.3.3 *Null PaRAM Set*), then no transfer request (TR) is submitted and the corresponding EDMA_TPCC_ER[31:0] E*n* bit is cleared and simultaneously the corresponding channel bit is set in the event miss register (EDMA_TPCC_EMR[31:0] E*n* = 1) to indicate that the event was discarded due to a null TR being serviced. Good programming practices should include cleaning the event missed error before re-triggering the DMA channel.

When an event is received, the corresponding event bit in the event register is set (EDMA_TPCC_ER[31:0] E*n* = 1), regardless of the state of EDMA_TPCC_EER[31:0] E*n*. If the event is disabled when an external event is received (EDMA_TPCC_ER[31:0] E*n* = 1 and EDMA_TPCC_EER[31:0] E*n* = 0), the EDMA_TPCC_ER[31:0] E*n* bit remains set. If the event is subsequently enabled (EDMA_TPCC_EER[31:0] E*n* = 1), then the pending event is processed by the EDMA_TPCC and the TR is processed/submitted, after which the EDMA_TPCC_ER[31:0] E*n* bit is cleared.

If an event is being processed (prioritized or is in the event queue) and another sync event is received for the same channel prior to the original being cleared (EDMA_TPCC_ER[31:0] E*n* != 0), then the second event is registered as a missed event in the corresponding bit of the event missed register (EDMA_TPCC_EMR[31:0] E*n* = 1).

11.4.4.1.2 Manually-Triggered Transfer Request

The CPU or any peripheral device module initiates a DMA transfer by writing to the event set register EDMA_TPCC_ESR. Writing a 1 to an event bit in the EDMA_TPCC_ESR results in the event being prioritized/queued in the appropriate event queue, regardless of the state of the EDMA_TPCC_EER[31:0] E*n* bit. When the event reaches the head of the queue, it is evaluated for submission as a transfer request to the transfer controller.

As in the event-triggered transfers, if the PaRAM set associated with the channel is valid (it is not a null set) then the TR is submitted to the associated EDMA_TPTC and the channel can be triggered again.

If the PaRAM set associated with the channel is a NULL set (see Section 11.4.3.3 *Null PaRAM Set*), then no transfer request (TR) is submitted and the corresponding EDMA_TPCC_ER[31:0] E*n* bit is cleared and simultaneously the corresponding channel bit is set in the event miss register EDMA_TPCC_EMR[31:0] E*n* = 1 to indicate that the event was discarded due to a null TR being serviced. Good programming practices should include clearing the event missed error before re-triggering the DMA channel.

If an event is being processed (prioritized or is in the event queue) and the same channel is manually set by a write to the corresponding channel bit of the event set register EDMA_TPCC_ESR[31:0] E*n* = 1 prior to the original being cleared EDMA_TPCC_ESR[31:0] E*n* = 0, then the second event is registered as a missed event in the corresponding bit of the event missed register EDMA_TPCC_EMR[31:0] E*n* = 1.

11.4.4.1.3 Chain-Triggered Transfer Request

Chaining is a mechanism by which the completion of one transfer automatically sets the event for another channel. When a chained completion code is detected, the value of which is dictated by the transfer completion code EDMA_TPCC_OPT_n[17:12] TCC of the PaRAM set associated with the channel, it results in the corresponding bit in the chained event register EDMA_TPCC_CER to be set EDMA_TPCC_CER[31:0] E[TCC] = 1).

Once a bit is set in EDMA_TPCC_CER, the EDMA_TPCC prioritizes and queues the event in the appropriate event queue. When the event reaches the head of the queue, it is evaluated for submission as a transfer request to the transfer controller.

As in the event-triggered transfers, if the PaRAM set associated with the channel is valid (it is not a null set) then the TR is submitted to the associated EDMA_TPTC and the channel can be triggered again.

If the PaRAM set associated with the channel is a NULL set (see Section 11.4.3.3 *Null PaRAM Set*), then no transfer request (TR) is submitted and the corresponding EDMA_TPCC_CER[31:0] E*n* bit is cleared and simultaneously the corresponding channel bit is set in the event miss register EDMA_TPCC_EMR[31:0] E*n* = 1 to indicate that the event was discarded due to a null TR being serviced. In this case, the error condition must be cleared before the DMA channel can be re-triggered. Good programming practices might include clearing the event missed error before re-triggering the DMA channel.

If a chaining event is being processed (prioritized or queued) and another chained event is received for the same channel prior to the original being cleared EDMA_TPCC_CER[31:0] E*n* != 0), then the second chained event is registered as a missed event in the corresponding channel bit of the event missed register EDMA_TPCC_EMR[31:0] E*n* = 1.

NOTE: Chained event registers EDMA_TPCC_CER, event registers EDMA_TPCC_ER, and event set registers EDMA_TPCC_ESR operate independently. An event E*n* can be triggered by any of the trigger sources (event-triggered, manually-triggered, or chain-triggered).

11.4.4.2 QDMA Channels

11.4.4.2.1 Auto-triggered and Link-Triggered Transfer Request

QDMA-based transfer requests are issued when a QDMA event gets latched in the QDMA event register EDMA_TPCC_QER[31:0] E*n* = 1. A bit corresponding to a QDMA channel is set in the QDMA event register EDMA_TPCC_QER when the following occurs:

- A CPU (or any device module) write occurs to a PaRAM address that is defined as a QDMA channel trigger word (programmed in the QDMA channel mapping register EDMA_TPCC_QCHMAPN_j for the particular QDMA channel and the QDMA channel is enabled via the QDMA event enable register EDMA_TPCC_QEER[31:0] E*n* = 1.
- EDMA_TPCC performs a link update on a PaRAM set address that is configured as a QDMA channel matches EDMA_TPCC_QCHMAPN_j settings and the corresponding channel is enabled via the QDMA event enable register EDMA_TPCC_QEER[31:0] E*n* = 1.

Once a bit is set in EDMA_TPCC_QER, the EDMA_TPCC prioritizes and queues the event in the appropriate event queue. When the event reaches the head of the queue, it is evaluated for submission as a transfer request to the transfer controller.

As in the event-triggered transfers, if the PaRAM set associated with the channel is valid (it is not a null set) then the TR is submitted to the associated EDMA_TPTC and the channel can be triggered again.

If a bit is already set in EDMA_TPCC_QER[31:0] E*n* = 1 and a second QDMA event for the same QDMA channel occurs prior to the original being cleared, the second QDMA event gets captured in the QDMA event miss register EDMA_TPCC_QEMR[7:0] E*n* = 1.

11.4.4.3 Comparison Between DMA and QDMA Channels

The primary difference between DMA and QDMA channels is the event/channel synchronization.

QDMA events are either auto-triggered or link triggered. Auto-triggering allows QDMA channels to be triggered by CPU(s) with a minimum number of linear writes to PaRAM. Link triggering allows a linked list of transfers to be executed, using a single QDMA PaRAM set and multiple link PaRAM sets.

A QDMA transfer is triggered when a CPU (or other device modules) writes to the trigger word of the QDMA channel parameter set (auto-triggered) or when the EDMA_TPCC performs a link update on a PaRAM set that has been mapped to a QDMA channel (link triggered).

NOTE: The CPUs triggered (manually triggered) DMA channels, in addition to writing to the PaRAM set, it is required to write to the event set register EDMA_TPCC_ESR to kick-off the transfer.

QDMA channels are typically for cases where a single event accomplishes a complete transfer since the CPU (or other device modules) must reprogram some portion of the QDMA PaRAM set in order to re-trigger the channel. QDMA transfers are programmed with EDMA_TPCC_ABCNT_n[31:0] BCNT =1 and EDMA_TPCC_CCNT_n[15:0] CCNT = 1 for A-synchronized transfers, and EDMA_TPCC_CCNT_n[15:0] CCNT = 1 for AB-synchronized transfers.

Additionally, since linking is also supported (if EDMA_TPCC_OPT_n[3] STATIC = 0) for QDMA transfers, it allows to initiate a linked list of QDMAs, so when EDMA_TPCC copies over a link PaRAM set (including the write to the trigger word), the current PaRAM set mapped to the QDMA channel automatically recognizes as a valid QDMA event and initiate another set of transfers as specified by the linked set.

11.4.5 Completion of a DMA Transfer

A parameter set for a given channel is complete when the required number of transfer requests is submitted (based on receiving the number of synchronization events). The expected number of TRs for a non-null/non-dummy transfer is shown in Table 11-13 for both synchronization types along with state of the PaRAM set prior to the final TR being submitted. When the counts (EDMA_TPCC_ABCNT_n[31:0] BCNT and/or EDMA_TPCC_CCNT_n[15:0] CCNT) are this value, the next TR results in:

- Final chaining or interrupt codes sent by the transfer controllers (instead of intermediate).
- Link updates (linking to either null or another valid link set).

Table 11-13. Expected Number of Transfers for Non-Null Transfer

Sync Mode	Counts at time 0	Total # Transfers	Counts prior to final TR
A-synchronized	ACNT BCNT CCNT	(BCNT × CCNT) TRs of ACNT bytes each	EDMA_TPCC_ABCNT_n[31:0] BCNT == 1 && EDMA_TPCC_CCNT_n[15:0] CCNT == 1
AB-synchronized	ACNT BCNT CCNT	CCNT TRs for ACNT × BCNT bytes each	EDMA_TPCC_CCNT_n[15:0] CCNT == 1

The PaRAM OPT field must program with a specific transfer completion code TCC or EDMA_TPCC_OPT_n[17:12] TCC along with the other EDMA_TPCC_OPT_n fields ([22] TCCHEN, [20] TCINTEN, [23] ITCCHEN, and [21] ITCINTEN bits) to indicate whether the completion code is to be used for generating a chained event or/and for generating an interrupt upon completion of a transfer.

The specific EDMA_TPCC_OPT_n[17:12] TCC value (6-bit binary value) programmed dictates which of the 64-bits in the chain event register EDMA_TPCC_CER [TCC] and/or interrupt pending register EDMA_TPCC_IPR [TCC] is set.

It can selectively program whether the transfer controller sends back completion codes on completion of the final transfer request (TR) of a parameter set EDMA_TPCC_OPT_n[22] TCCHEN or EDMA_TPCC_OPT_n[20] TCINTEN, for all but the final transfer request (TR) of a parameter set EDMA_TPCC_OPT_n[23] ITCCHEN or EDMA_TPCC_OPT_n[21] ITCINTEN), or for all TRs of a parameter set (both). Refer to Section 11.4.8 *Chaining EDMA Channels* for details on chaining (intermediate/final chaining) and Section 11.4.9 *EDMA Interrupts* for details on intermediate/final interrupt completion.

A completion detection interface exists between the EDMA channel controller and transfer controller(s). This interface sends back information from the transfer controller to the channel controller to indicate that a specific transfer is completed. Completion of a transfer is used for generating chained events and/or generating interrupts to the CPU(s).

All DMA/QDMA PaRAM sets must also specify a link address value. For repetitive transfers such as ping-pong buffers, the link address value must point to another predefined PaRAM set. Alternatively, a non-repetitive transfer must set the link address value to the null link value. The null link value is defined as FFFFh. Refer to Section 11.4.3.7 *Linking Transfers* for more details.

NOTE: Any incoming events that are mapped to a null PaRAM set results in an error condition. The error condition must clear before the corresponding channel is used again. Refer to Section 11.4.3.5 *Dummy Versus Null Transfer Comparison*.

There are three ways the EDMA_TPCC gets updated/informed about a transfer completion: normal completion, early completion, and dummy/null completion. This applies to both chained events and completion interrupt generation.

11.4.5.1 Normal Completion

In normal completion mode EDMA_TPCC_OPT_n[11] TCCMODE = 0, the transfer or sub-transfer is considered to be complete when the EDMA channel controller receives the completion codes from the EDMA transfer controller. In this mode, the completion code to the channel controller is posted by the transfer controller after it receives a signal from the destination peripheral. Normal completion is typically used to generate an interrupt to inform the CPU that a set of data is ready for processing.

11.4.5.2 Early Completion

In early completion mode EDMA_TPCC_OPT_n[11] TCCMODE = 1, the transfer is considered to be complete when the EDMA channel controller submits the transfer request (TR) to the EDMA transfer controller. In this mode, the channel controller generates the completion code internally. Early completion is typically useful for chaining, as it allows subsequent transfers to be chained-triggered while the previous transfer is still in progress within the transfer controller, maximizing the overall throughput of the set of the transfers.

11.4.5.3 Dummy or Null Completion

This is a variation of early completion. Dummy or null completion is associated with a dummy set Section 11.4.3.4 or null set Section 11.4.3.3. In both cases, the EDMA channel controller does not submit the associated transfer request to the EDMA transfer controller(s). However, if the set (dummy/null) has the OPT field programmed to return completion code (intermediate/final interrupt/chaining completion), then it sets the appropriate bits in the interrupt pending registers EDMA_TPCC_IPR and EDMA_TPCC_IPRH or chained event register EDMA_TPCC_CER and EDMA_TPCC_CERH. The internal early completion path is used by the channel controller to return the completion codes internally (that is, EDMA_TPCC generates the completion code).

11.4.6 Event, Channel, and PaRAM Mapping

Several of the 64 DMA channels are tied to a specific hardware event, thus allowing events from device peripherals or external hardware (via the dma_evt[4:1] pins) to trigger transfers. A DMA channel typically requests a data transfer when it receives its event (apart from manually-triggered, chain-triggered, and other transfers). The amount of data transferred per synchronization event depends on the channel's configuration (EDMA_TPCC_ABCNT_n[15:0] ACNT, EDMA_TPCC_ABCNT_n[31:16] BCNT, EDMA_TPCC_CCNT_n[15:0] CCNT, etc.) and the synchronization type (A-synchronized or AB-synchronized).

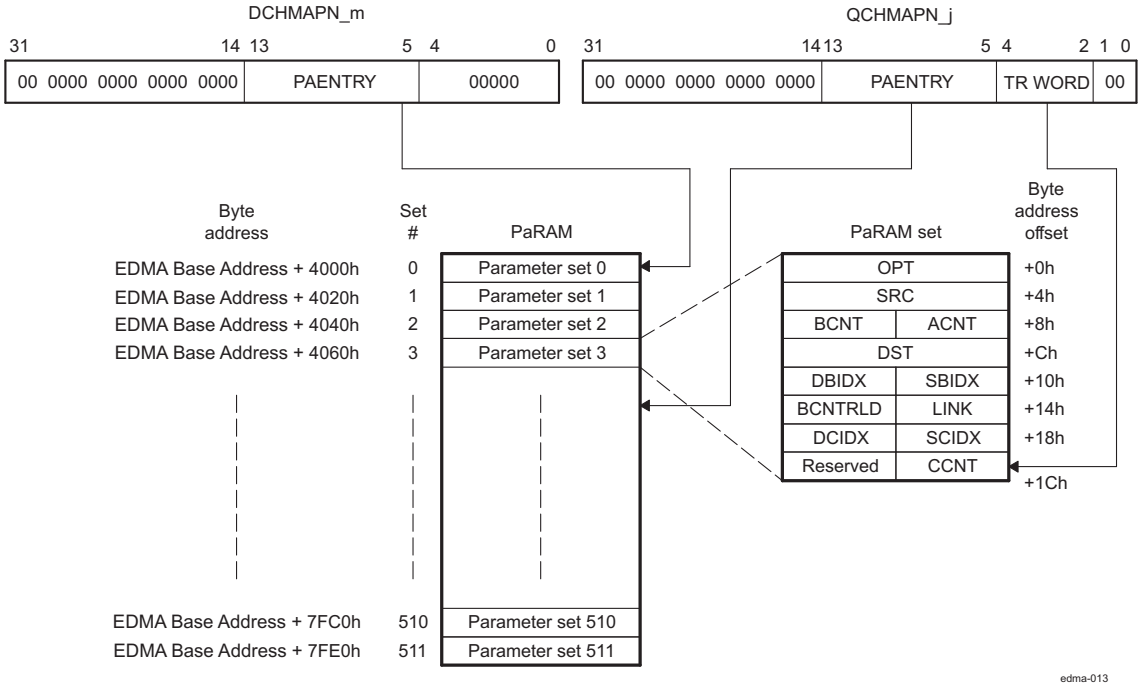
The association of an event to a channel is fixed within the EDMA Channel Controller, that is, each DMA channel has one specific event associated with it. The EDMA event crossbar can be used to select which level events (of which there are more than 64) are mapped to the 64 input events to the EDMA Channel Controller. The default mapping and event crossbar mapping are defined in Section 11.3.1, *EDMA Requests to the EDMA Controller*. The event crossbar mapping is controlled by the device Control Modules registers refer to Section 13.4.6.4 *IRQ_CROSSBAR Module Functional Description*, in Chapter 13 *Control Module*.

In an application, if a channel does not use the associated synchronization event or if it does not have an associated synchronization event (unused), that channel can be used for manually-triggered or chained-triggered transfers, for linking/reloading, or as a QDMA channel.

11.4.6.1 DMA Channel to PaRAM Mapping

The mapping between the DMA channel numbers and the PaRAM sets is programmable (see Table 11-9). The DMA channel mapping registers EDMA_TPCC_DCHMAPN_m in the EDMA_TPCC provide programmability that allows the DMA channels to be mapped to any of the PaRAM sets in the PaRAM memory map. Figure 11-13 illustrates the use of EDMA_TPCC_DCHMAPN_m. There is one EDMA_TPCC_DCHMAPN_m register per channel.

Figure 11-13. DMA Channel and QDMA Channel to PaRAM Mapping



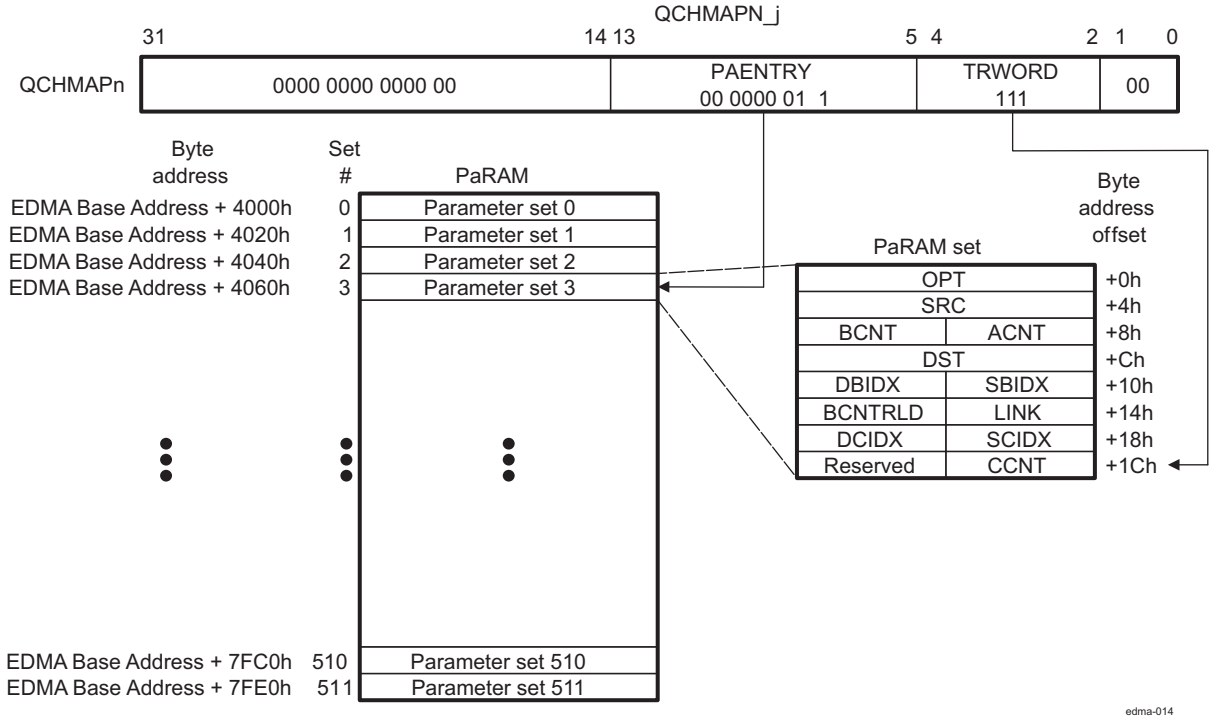
11.4.6.2 QDMA Channel to PaRAM Mapping

The mapping between the QDMA channels and the PaRAM sets is programmable. The QDMA channel mapping register EDMA_TPCC_QCHMAPN_j in the EDMA_TPCC allows to map the QDMA channels to any of the PaRAM sets in the PaRAM memory map. Figure 11-14 illustrates the use of EDMA_TPCC_QCHMAPN_j.

EDMA_TPCC_QCHMAPN_j[4:2] TRWORD bit-field allows to program the trigger word in the PaRAM set for the QDMA channel. A trigger word is one of the eight words in the PaRAM set. For a QDMA transfer to occur, a valid TR synchronization event for EDMA_TPCC is a write to the trigger word in the PaRAM set pointed to by EDMA_TPCC_QCHMAPN_j for a particular QDMA channel. By default, QDMA channels are mapped to PaRAM set 0.

It must appropriately re-map PaRAM set 0 before use.

Figure 11-14. QDMA Channel to PaRAM Mapping



11.4.7 EDMA Channel Controller Regions

The EDMA channel controller divides its address space into eight regions. Individual channel resources are assigned to a specific region, where each region is typically assigned to a specific device module uses the EDMA controller.

Application software can use regions or to ignore them altogether. It can be used active memory protection in conjunction with regions so that only a specific device module which uses the EDMA (for example, privilege identification) or privilege level (for example, user vs. supervisor) is allowed access to a given region, and thus to a given DMA or QDMA channel. This allows robust system-level DMA code where each EDMA initiator only modifies the state of the assigned resources. Memory protection is described in Section 11.4.10 *Memory Protection*.

11.4.7.1 Region Overview

The EDMA channel controller memory-mapped registers are divided in three main categories:

1. Global registers
2. Global region channel registers
3. Shadow region channel registers

The global registers are located at a single/fixed location in the EDMA_TPCC memory map. These registers control EDMA resource mapping and provide debug visibility and error tracking information.

The channel registers (including DMA, QDMA, and interrupt registers) are accessible via the global channel region address range, or in the shadow *n* channel region address range(s). For example, the event enable register EDMA_TPCC_EER is visible at the global address of EDMA Base Address + 1020h or region addresses of EDMA Base Address + 2020h for region 0, EDMA Base Address + 2220h for region 1, … EDMA Base Address + 2E20h for region 7.

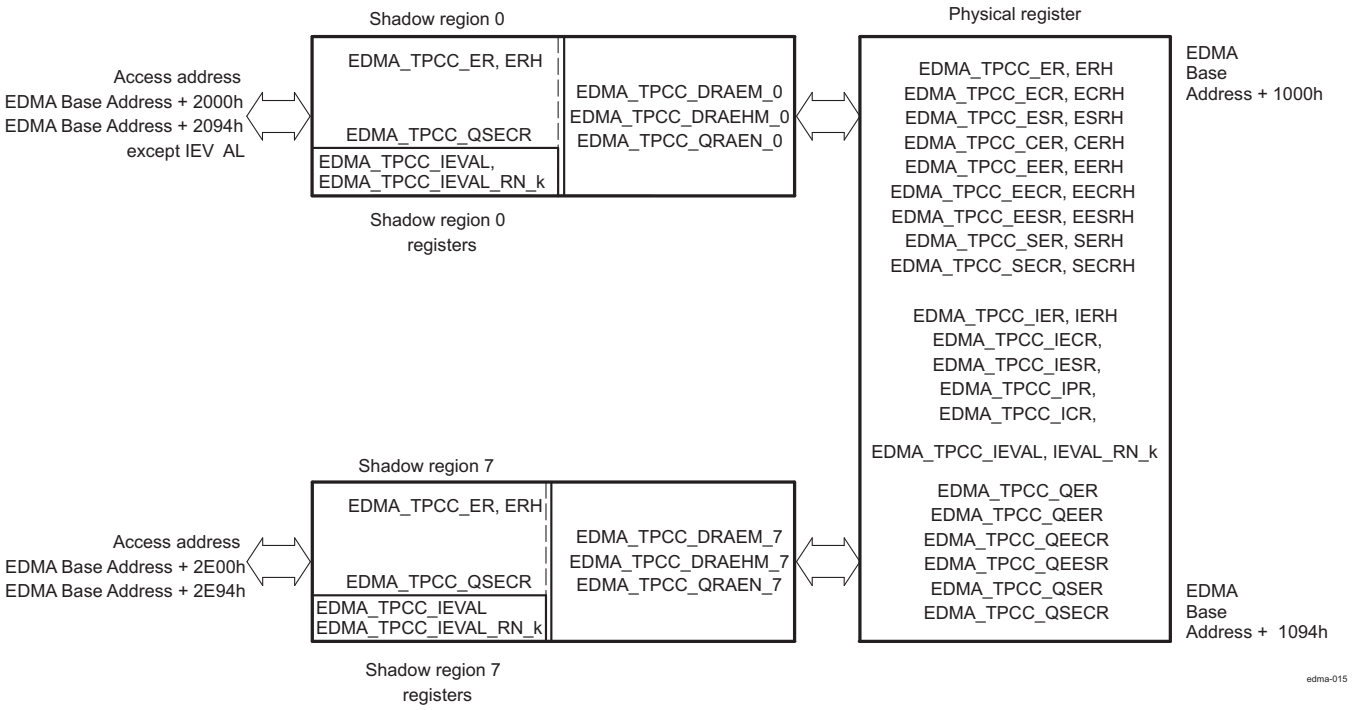
The DMA region access enable registers EDMA_TPCC_DRAEM_k and the QDMA region access enable registers EDMA_TPCC_QRAEN_k control the underlying control register bits that are accessible via the shadow region address space (except for EDMA_TPCC_IEVAL and EDMA_TPCC_IEVAL_RN_k registers). Table 11-14 lists the registers in the shadow region memory map. Refer to Table 11-30 *EDMA_TPCC register mapping summary* for the complete global and shadow region memory maps.

Table 11-14. Shadow Region Registers

EDMA_TPCC_DRAEM_k	EDMA_TPCC_DRAEHM_k	EDMA_TPCC_QRAEN_k
EDMA_TPCC_ER	EDMA_TPCC_ERH	EDMA_TPCC_QER
EDMA_TPCC_ECR	EDMA_TPCC_ECRH	EDMA_TPCC_QEER
EDMA_TPCC_ESR	EDMA_TPCC_ESRH	EDMA_TPCC_QEECR
EDMA_TPCC_CER	EDMA_TPCC_CERH	EDMA_TPCC_QEESR
EDMA_TPCC_EER	EDMA_TPCC_EERH	
EDMA_TPCC_EECR	EDMA_TPCC_EECRH	
EDMA_TPCC_EESR	EDMA_TPCC_EESRH	
EDMA_TPCC_SER	EDMA_TPCC_SERH	
EDMA_TPCC_SECR	EDMA_TPCC_SECRH	
EDMA_TPCC_IER	EDMA_TPCC_IERH	
EDMA_TPCC_IECR	EDMA_TPCC_IECRH	
EDMA_TPCC_IESR	EDMA_TPCC_IESRH	
EDMA_TPCC_IPR	EDMA_TPCC_IPRH	
EDMA_TPCC_ICR	EDMA_TPCC_ICRH	
Register not affected by DRAE\DRAEH		
EDMA_TPCC_IEVAL		
EDMA_TPCC_IEVAL_RN_k		

Figure 11-15 illustrates the conceptual view of the regions.

Figure 11-15. Shadow Region Registers



11.4.7.2 Channel Controller Regions

There are eight EDMA shadow regions (and associated memory maps). Associated with each shadow region are a set of registers defining which channels and interrupt completion codes belong to that region. These registers are user-programmed per region to assign ownership of the DMA/QDMA channels to a region.

- EDMA_TPCC_DRAEM_k and EDMA_TPCC_DRAEHM_k: One register pair exists for each of the shadow regions. The number of bits in each register pair matches the number of DMA channels (64 DMA channels). These registers need to be programmed to assign ownership of DMA channels and interrupt (or EDMA_TPCC_OPT_n[17:12] TCC codes) to the respective region. Accesses to DMA and interrupt registers via the shadow region address view are filtered through the DRAEM/DRAEHM pair. A value of 1 in the corresponding EDMA_TPCC_DRAEM_k[31:0] / EDMA_TPCC_DRAEHM_k[31:0] bit implies that the corresponding DMA interrupt channel is accessible; a value of 0 in the corresponding EDMA_TPCC_DRAEM_k[31:0] / EDMA_TPCC_DRAEHM_k[31:0] bit forces writes to be discarded and returns a value of 0 for reads.
- EDMA_TPCC_QRAEN_k: One register exists for every region. The number of bits in each register matches the number of QDMA channels (4 QDMA channels). These registers must be programmed to assign ownership of QDMA channels to the respective region. To enable a channel in a shadow region using shadow region 0 EDMA_TPCC_QEER, the corresponding bits in QRAE must be set or writing into EDMA_TPCC_QEESR there will be no the desired effect.
- EDMA_TPCC_MPPAN_k and EDMA_TPCC_MPPAG: One register exists for every region. This register defines the privilege level, requestor, and types of accesses allowed to a region's memory-mapped registers.

It is typical for an application to have a unique assignment of QDMA/DMA channels (and, therefore, a given bit position) to a given region.

The use of shadow regions allows restricted access to EDMA resources (DMA channels, QDMA channels, TCC, interrupts) by tasks in a system by setting or clearing bits in the EDMA_TPCC_DRAEM_k / EDMA_TPCC_QRAEN_k registers.

If exclusive access to any given channel / TCC code is required for a region, then only that region's EDMA_TPCC_DRAEM_k / EDMA_TPCC_QRAEN_k have the associated bit set.

Example 11-1. Resource Pool Division Across Two Regions

This example illustrates a resource pool division across two regions, assuming region 0 must be allocated 16 DMA channels (0-15) and 1 QDMA channel (0) and 32 TCC codes (0-15 and 48-63).

Region 1 needs to be allocated 16 DMA channels (16-32) and the remaining 7 QDMA channels (1-7) and TCC codes (16-47).

EDMA_TPCC_DRAEM_k should be equal to the OR of the bits that are required for the DMA channels and the TCC codes:

```
Region 0: DRAEHM, DRAEM = 0xFFFF0000, 0x0000FFFF QRAEN = 0x0000001
Region 1: DRAEHM, DRAEM = 0x0000FFFF, 0xFFFF0000 QRAEN = 0x00000FE
```

11.4.7.3 Region Interrupts

In addition to the EDMA_TPCC global completion interrupt, there is an additional completion interrupt line that is associated with every shadow region. Along with the interrupt enable register EDMA_TPCC_IER, DRAEM acts as a secondary interrupt enable for the respective shadow region interrupts. Refer to Table 11-6 *Hardware Request* for more information about EDMA Interrupts.

11.4.8 Chaining EDMA Channels

The channel chaining capability for the EDMA allows the completion of an EDMA channel transfer to trigger another EDMA channel transfer. The purpose is to allow the ability to chain several events through one event occurrence.

Chaining is different from linking (Section 11.4.3.7 *Linking Transfers*). The EDMA link feature reloads the current channel parameter set with the linked parameter set. The EDMA chaining feature does not modify or update any channel parameter set. It provides a synchronization event to the chained channel (see Section 11.4.4.1.3 *Chain-Triggered Transfer Request*).

Chaining is achieved at either final transfer completion or intermediate transfer completion, or both, of the current channel. Consider a channel *m* (DMA/QDMA) required to chain to channel *n*. Channel number *n* (0-63) needs to be programmed into the EDMA_TPCC_OPT_n[17:12] TCC bit-field of channel *m* channel options parameter (OPT) set.

- If final transfer completion chaining EDMA_TPCC_OPT_n[22] TCCHEN = 1 is enabled, the chain-triggered event occurs after the submission of the last transfer request of channel *m* is either submitted or completed (depending on early or normal completion).

- If intermediate transfer completion chaining EDMA_TPCC_OPT_n[23] ITCCHEN = 1 is enabled, the chain-triggered event occurs after every transfer request, except the last of channel *m* is either submitted or completed (depending on early or normal completion).
- If both final and intermediate transfer completion chaining (EDMA_TPCC_OPT_n[22] TCCHEN = 1 and EDMA_TPCC_OPT_n[23] ITCCHEN = 1) are enabled, then the chain-trigger event occurs after every transfer request is submitted or completed (depending on early or normal completion).

Table 11-15 illustrates the number of chain event triggers occurring in different synchronized scenarios. Consider channel 31 programmed with EDMA_TPCC_ABCNT_n[15:0] ACNT = 3, EDMA_TPCC_ABCNT_n[31:16] BCNT = 4, EDMA_TPCC_CCNT_n[15:0] CCNT = 5, and EDMA_TPCC_OPT_n[17:12] TCC = 30.

Table 11-15. Chain Event Triggers

	(Number of chained event triggers on channel 30)	
Options	**A-Synchronized**	**AB-Synchronized**
EDMA_TPCC_OPT_n[22] TCCHEN = 1, EDMA_TPCC_OPT_n[23] ITCCHEN = 0	1 (Owing to the last TR)	1 (Owing to the last TR)
EDMA_TPCC_OPT_n[22] TCCHEN = 0, EDMA_TPCC_OPT_n[23] ITCCHEN = 1	19 (Owing to all but the last TR)	4 (Owing to all but the last TR)
EDMA_TPCC_OPT_n[22] TCCHEN = 1, EDMA_TPCC_OPT_n[23] ITCCHEN = 1	20 (Owing to a total of 20 TRs)	5 (Owing to a total of 5 TRs)

11.4.9 EDMA Interrupts

The EDMA interrupts are divided into 2 categories: transfer completion interrupts and error interrupts.

There are nine region interrupts, eight shadow regions and one global region. The transfer completion interrupts are listed in Table 11-16. The transfer completion interrupts and the error interrupts from the transfer controllers are all routed to the device interrupt controllers INTCs through the inputs of the IRQ_CROSSBAR module.

Table 11-16. EDMA Transfer Completion Interrupts

Name	Description
EDMA_TPCC_INT0	EDMA_TPCC Transfer Completion Interrupt Shadow Region 0
EDMA_TPCC_INT1	EDMA_TPCC Transfer Completion Interrupt Shadow Region 1
EDMA_TPCC_INT2	EDMA_TPCC Transfer Completion Interrupt Shadow Region 2
EDMA_TPCC_INT3	EDMA_TPCC Transfer Completion Interrupt Shadow Region 3
EDMA_TPCC_INT4	EDMA_TPCC Transfer Completion Interrupt Shadow Region 4
EDMA_TPCC_INT5	EDMA_TPCC Transfer Completion Interrupt Shadow Region 5
EDMA_TPCC_INT6	EDMA_TPCC Transfer Completion Interrupt Shadow Region 6
EDMA_TPCC_INT7	EDMA_TPCC Transfer Completion Interrupt Shadow Region 7

Table 11-17. EDMA Error Interrupts

Name	Description
EDMA_TPCC_ERRINT	EDMA_TPCC Error Interrupt
EDMA_TPCC_MPINT	EDMA_TPCC Memory Protection Interrupt
EDMA_TC0_ERRINT	TC0 Error Interrupt
EDMA_TC1_ERRINT	TC1 Error Interrupt

11.4.9.1 Transfer Completion Interrupts

The EDMA_TPCC is responsible for generating transfer completion interrupts to the CPU(s) (and other EDMA masters). The EDMA generates a single completion interrupt per shadow region, as well as one for the global region on behalf of all 64 channels. The various control registers and bit fields facilitate EDMA interrupt generation.

The software architecture must either use the global interrupt or the shadow interrupts, but not both.

The transfer completion code EDMA_TPCC_OPT_n[17:12] TCC value is directly mapped to the bits of the interrupt pending register EDMA_TPCC_IPR / EDMA_TPCC_IPRH.

For example, if EDMA_TPCC_OPT_n[17:12] TCC = 10 0001b, EDMA_TPCC_IPRH[1] is set after transfer completion, and results in interrupt generation to the CPU(s) if the completion interrupt is enabled for the CPU. See Section 11.4.9.1.1 *Enabling Transfer Completion Interrupts* for details about enabling EDMA transfer completion interrupts.

When a completion code is returned (as a result of early or normal completions), the corresponding bit in EDMA_TPCC_IPR / EDMA_TPCC_IPRH registers is set if transfer completion interrupt (final/intermediate) is enabled in the channel options parameter (OPT) for a PaRAM set associated with the transfer.

Table 11-18. Transfer Complete Code (TCC) to EDMA_TPCC Interrupt Mapping

TCC values in EDMA_TPCC_OPT_n[17:12] TCC (EDMA_TPCC_OPT_n[20] TCINTEN / EDMA_TPCC_OPT_n[21] ITCINTEN = 1)	EDMA_TPCC_IPR Bit Set	TCC values in EDMA_TPCC_OPT_n[17:12] TCC (EDMA_TPCC_OPT_n[20] TCINTEN / EDMA_TPCC_OPT_n[21] ITCINTEN = 1)	EDMA_TPCC_IPRH Bit Set[1]
0	EDMA_TPCC_IPR[0]	20h	EDMA_TPCC_IPR[32] / EDMA_TPCC_IPRH[0]
1	EDMA_TPCC_IPR[1]	21h	EDMA_TPCC_IPR[33] / EDMA_TPCC_IPRH[1]
2h	EDMA_TPCC_IPR[2]	22h	EDMA_TPCC_IPR[34] / EDMA_TPCC_IPRH[2]
3h	EDMA_TPCC_IPR[3]	23h	EDMA_TPCC_IPR[35] / EDMA_TPCC_IPRH[3]
4h	EDMA_TPCC_IPR[4]	24h	EDMA_TPCC_IPR[36] / EDMA_TPCC_IPRH[4]
...	...	...	...
1Eh	EDMA_TPCC_IPR[30]	3Eh	EDMA_TPCC_IPR[62] / EDMA_TPCC_IPRH[30]
1Fh	EDMA_TPCC_IPR[31]	3Fh	EDMA_TPCC_IPR[63] / EDMA_TPCC_IPRH[31]

(1) Bit fields EDMA_TPCC_IPR [32-63] correspond to bits 0 to 31 in EDMA_TPCC_IPRH, respectively.

The transfer completion code (TCC) can program to any value for a DMA/QDMA channel. A direct relation between the channel number and the transfer completion code value does not need to exist. This allows multiple channels having the same transfer completion code value to cause a CPU to execute the same interrupt service routine (ISR) for different channels.

If the channel is used in the context of a shadow region and it intends for the shadow region interrupt to be asserted, then ensure that the bit corresponding to the TCC code is enabled in EDMA_TPCC_IER / EDMA_TPCC_IERH and in the corresponding shadow region's DMA region access registers (EDMA_TPCC_DRAEM_k / EDMA_TPCC_DRAEHM_k).

Interrupt generation can be enabled at either final transfer completion or intermediate transfer completion, or both. Consider channel *m* as an example.

- If the final transfer interrupt (EDMA_TPCC_OPT_n[20] TCINTEN = 1) is enabled, the interrupt occurs after the last transfer request of channel *m* is either submitted or completed (depending on early or normal completion).

- If the intermediate transfer interrupt (EDMA_TPCC_OPT_n[21] ITCINTEN = 1) is enabled, the interrupt occurs after every transfer request, except the last TR of channel *m* is either submitted or completed (depending on early or normal completion).
- If both final and intermediate transfer completion interrupts (EDMA_TPCC_OPT_n[20] TCINTEN = 1, and EDMA_TPCC_OPT_n[21] ITCINTEN = 1) are enabled, then the interrupt occurs after every transfer request is submitted or completed (depending on early or normal completion).

Table 11-19 shows the number of interrupts that occur in different synchronized scenarios. Consider channel 31, programmed with ABCNT_n[15:0] ACNT = 3, EDMA_TPCC_ABCNT_n[31:16] BCNT = 4, EDMA_TPCC_CCNT_n[15:0]CCNT = 5, and EDMA_TPCC_OPT_n[17:12] TCC = 30.

Table 11-19. Number of Interrupts

Options	A-Synchronized	AB-Synchronized
EDMA_TPCC_OPT_n[20] TCINTEN = 1, EDMA_TPCC_OPT_n[21] ITCINTEN = 0	1 (Last TR)	1 (Last TR)
EDMA_TPCC_OPT_n[20] TCINTEN = 0, EDMA_TPCC_OPT_n[21] ITCINTEN = 1	19 (All but the last TR)	4 (All but the last TR)
EDMA_TPCC_OPT_n[20] TCINTEN = 1, EDMA_TPCC_OPT_n[21] ITCINTEN = 1	20 (All TRs)	5 (All TRs)

11.4.9.1.1 Enabling Transfer Completion Interrupts

For the EDMA channel controller to assert a transfer completion to the external environment, the interrupts must be enabled in the EDMA_TPCC. This is in addition to setting up the EDMA_TPCC_OPT_n[20] TCINTEN and EDMA_TPCC_OPT_n[21] ITCINTEN bits of the associated PaRAM set.

The EDMA channel controller has interrupt enable registers EDMA_TPCC_IER / EDMA_TPCC_IERH and each bit location in EDMA_TPCC_IER / EDMA_TPCC_IERH serves as a primary enable for the corresponding interrupt pending registers EDMA_TPCC_IPR / EDMA_TPCC_IPRH.

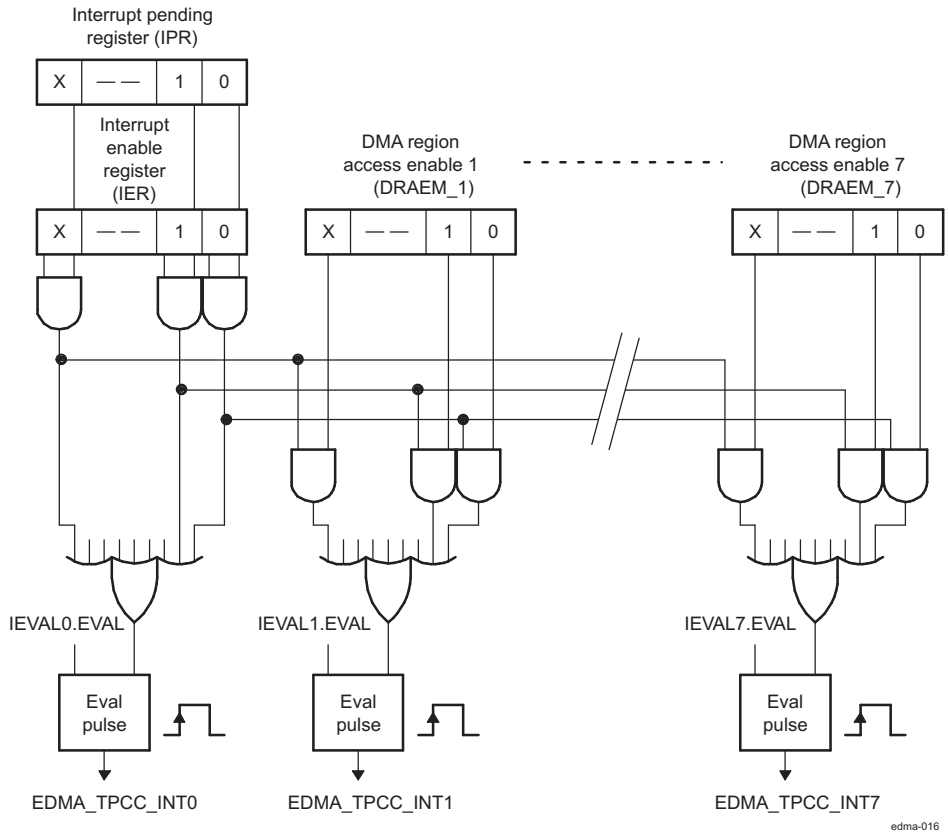
All of the interrupt registers (EDMA_TPCC_IER, EDMA_TPCC_IESR, EDMA_TPCC_IECR, and EDMA_TPCC_IPR) are either manipulated from the global DMA channel region, or by the DMA channel shadow regions. The shadow regions provide a view to the same set of physical registers that are in the global region.

The EDMA channel controller has a hierarchical completion interrupt scheme that uses a single set of interrupt pending registers EDMA_TPCC_IPR / EDMA_TPCC_IPRH and single set of interrupt enable registers EDMA_TPCC_IER / EDMA_TPCC_IERH. The programmable DMA region access enable registers EDMA_TPCC_DRAEM_k / EDMA_TPCC_DRAEHM_k provides a second level of interrupt masking. The global region interrupt output is gated based on the enable mask that is provided by EDMA_TPCC_IER / EDMA_TPCC_IERH, see Figure 11-16

The region interrupt outputs are gated by EDMA_TPCC_IER and the specific EDMA_TPCC_DRAEM_k / EDMA_TPCC_DRAEHM_k associated with the region.

Figure 11-16 shows the Interrupt diagram of the EDMA controller.

Figure 11-16. Interrupt Diagram



The EDMA_TPCC generates the transfer completion interrupts that are associated with each shadow region, the following conditions must be true:

- EDMA_TPCC_INT0: (EDMA_TPCC_IPR[0] E0 & EDMA_TPCC_IER[0] E0 & EDMA_TPCC_DRAEM_k.DRAEM_0[0] E0) | (EDMA_TPCC_IPR[1] E1 & EDMA_TPCC_IER[1] E1 & EDMA_TPCC_DRAEM_k.DRAEM_0[1] E1) | …|(EDMA_TPCC_IPRH[31] E63 & EDMA_TPCC_IERH[31] E63 & EDMA_TPCC_DRAEHM_k.DRAEHM_0[31] E63)
- EDMA_TPCC_INT1: (EDMA_TPCC_IPR[0] E0 & EDMA_TPCC_IER[0] E0 & EDMA_TPCC_DRAEM_k.DRAEM_1[0] E0) | (EDMA_TPCC_IPR[1] E1 & EDMA_TPCC_IER[1] E1 & EDMA_TPCC_DRAEM_k.DRAEM_1[1] E1) | …| (EDMA_TPCC_IPRH[31] E63 & EDMA_TPCC_IERH[31] E63 & EDMA_TPCC_DRAEHM_k.DRAEHM_1[31] E63)
- EDMA_TPCC_INT2: (EDMA_TPCC_IPR[0] E0 & EDMA_TPCC_IER[0] E0 & EDMA_TPCC_DRAEM_k.DRAEM_2[0] E0) | (EDMA_TPCC_IPR[1] E1 & EDMA_TPCC_IER[1] E1 & EDMA_TPCC_DRAEM_k.DRAEM_2[1] E1) | …|(EDMA_TPCC_IPRH[31] E63 & EDMA_TPCC_IERH[31] E63 & EDMA_TPCC_DRAEHM_k.DRAEHM_2[31] E63)....
- Up to EDMA_TPCC_INT7: (EDMA_TPCC_IPR[0] E0 & EDMA_TPCC_IER[0] E0 & EDMA_TPCC_DRAEM_k.DRAEM_7[0] E0) | (EDMA_TPCC_IPR[1] E1 & EDMA_TPCC_IER[1] E1 & EDMA_TPCC_DRAEM_k.DRAEM_7[1] E1) | …|(EDMA_TPCC_IPRH[31] E63 & EDMA_TPCC_IERH[31] E63 & EDMA_TPCC_DRAEHM_k.DRAEHM_7[31] E63)

NOTE: The EDMA_TPCC_DRAEM_k / EDMA_TPCC_DRAEHM_k for all regions are expected to be set up at system initialization and to remain static for an extended period of time. The interrupt enable registers are used for dynamic enable/disable of individual interrupts.

Because there is no relation between the EDMA_TPCC_OPT_n[17:12] TCC value and the DMA/QDMA channel, it is possible, the DMA channel 0 to have the EDMA_TPCC_OPT_n[17:12] TCC = 63 in its associated PaRAM set. This mean that if a transfer completion interrupt is enabled (EDMA_TPCC_OPT_n[20] TCINTEN or EDMA_TPCC_OPT_n[21] ITCINTEN is set), then based on the TCC value, EDMA_TPCC_IPRH[31] E63 is set up on completion. For proper channel operations and interrupt generation using the shadow region map - program the EDMA_TPCC_DRAEM_k / EDMA_TPCC_DRAEHM_k that is associated with the shadow region to have read/write access to both bit 0 (corresponding to channel 0) and bit 63 (corresponding to EDMA_TPCC_IPRH bit that is set upon completion).

11.4.9.1.2 Clearing Transfer Completion Interrupts

Transfer completion interrupts that are latched to the interrupt pending registers (EDMA_TPCC_IPR / EDMA_TPCC_IPRH) are cleared by writing a 1 to the corresponding bit in the interrupt pending clear register (EDMA_TPCC_ICR / EDMA_TPCC_ICRH). For example, a write of 1 to EDMA_TPCC_ICR[0] E0 clears a pending interrupt in EDMA_TPCC_IPR[0] E0.

If an incoming transfer completion code TCC (EDMA_TPCC_OPT_n[17:12] TCC) gets latched to a bit in EDMA_TPCC_IPR / EDMA_TPCC_IPRH, then additional bits that get set due to a subsequent transfer completion does not result in asserting the EDMA_TPCC completion interrupt. In order for the completion interrupt to be pulsed, the required transition is from a state where no enabled interrupts are set to a state where at least one enabled interrupt is set.

11.4.9.2 EDMA Interrupt Servicing

Upon completion of a transfer (early or normal completion), the EDMA channel controller sets the appropriate bit in the interrupt pending registers (EDMA_TPCC_IPR / EDMA_TPCC_IPRH), as the transfer completion codes specify. If the completion interrupts are appropriately enabled, then the CPU enters the interrupt service routine (ISR) when the completion interrupt is asserted.

After servicing the interrupt, the ISR should clear the corresponding bit in EDMA_TPCC_IPR/EDMA_TPCC_IPRH, thereby enabling recognition of future interrupts. The EDMA_TPCC only asserts additional completion interrupts when all EDMA_TPCC_IPR / EDMA_TPCC_IPRH bits clear.

When one interrupt is serviced many other transfer completions may result in additional bits being set in EDMA_TPCC_IPR / EDMA_TPCC_IPRH, thereby resulting in additional interrupts. Each of the bits in EDMA_TPCC_IPR / EDMA_TPCC_IPRH may need different types of service therefore, the ISR must check all pending interrupts and continue until all of the posted interrupts are serviced appropriately.

Examples of pseudo code for a CPU interrupt service routine for an EDMA_TPCC completion interrupt are shown in Example 11-2 and Example 11-3.

The ISR routine in Example 11-2 is more exhaustive and incurs a higher latency.

Example 11-2. Interrupt Servicing

The pseudo code:

1. Reads the interrupt pending register EDMA_TPCC_IPR / EDMA_TPCC_IPRH.
2. Performs the operations needed.
3. Writes to the interrupt pending clear register EDMA_TPCC_ICR / EDMA_TPCC_ICRH to clear the corresponding EDMA_TPCC_IPR / EDMA_TPCC_IPRH bit(s).
4. Reads EDMA_TPCC_IPR / EDMA_TPCC_IPRH again:
   a. If EDMA_TPCC_IPR / EDMA_TPCC_IPRH is not equal to 0, repeat from step 2 (implies occurrence of new event between step 2 to step 4).

Example 11-2. Interrupt Servicing (continued)

b. If EDMA_TPCC_IPR / EDMA_TPCC_IPRH is equal to 0, assure that all of the enabled interrupts are inactive.

NOTE: An event may occur during step 4 while the EDMA_TPCC_IPR / EDMA_TPCC_IPRH bits are read as 0 and the application is still in the interrupt service routine. If this happens, a new interrupt is recorded in the device interrupt controller and a new interrupt generates as soon as the application exits in the interrupt service routine.

Example 11-3 is less rigorous, with less burden on the software in polling for set interrupt bits, but can occasionally cause a race condition as mentioned above.

Example 11-3. Interrupt Servicing

If any enabled and pending (possibly lower priority) interrupts are left, force the interrupt logic to reassert the interrupt pulse by setting the EDMA_TPCC_IEVAL[0] EVAL bit in the interrupt evaluation register.

The pseudo code is as follows:

1. Enters ISR.
2. Reads EDMA_TPCC_IPR / EDMA_TPCC_IPRH.
3. For the condition that is set in EDMA_TPCC_IPR / EDMA_TPCC_IPRH:
   a. Service interrupt as the application requires.
   b. Clear the bit for serviced conditions (others may still be set, and other transfers may have resulted in returning the TCC to EDMA_TPCC after step 2).
4. Reads EDMA_TPCC_IPR / EDMA_TPCC_IPRH prior to exiting the ISR:
   a. If EDMA_TPCC_IPR / EDMA_TPCC_IPRH is equal to 0, then exit the ISR.
   b. If EDMA_TPCC_IPR / EDMA_TPCC_IPRH is not equal to 0, then set EDMA_TPCC_IEVAL so that upon exit of ISR, a new interrupt triggers if any enabled interrupts are still pending.

11.4.9.3 Interrupt Evaluation Operations

The EDMA_TPCC has interrupt evaluate registers EDMA_TPCC_IEVAL that exist in the global region and in each shadow region. The registers in the shadow region are the only registers in the DMA channel shadow region memory map that are not affected by the settings for the DMA region access enable registers EDMA_TPCC_DRAEM_k / EDMA_TPCC_DRAEHM_k. Writing a 1 to the EDMA_TPCC_IEVAL[0] EVAL bit in the registers that are associated with a particular shadow region results in pulsing the associated region interrupt (global or shadow), if any enabled interrupt (via EDMA_TPCC_IER / EDMA_TPCC_IERH) is still pending EDMA_TPCC_IPR / EDMA_TPCC_IPRH. This register assures that the CPU does not miss the interrupts (or the EDMA master associated with the shadow region) if the software architecture chooses not to use all interrupts. Refer to Example 11-3 about the use of EDMA_TPCC_IEVAL in the EDMA interrupt service routine (ISR).

Similarly an error evaluation register EDMA_TPCC_EEVAL exists in the global region. Writing a 1 to the EDMA_TPCC_EEVAL[0] EVAL bit causes the pulsing of the error interrupt if any pending errors are in EDMA_TPCC_EMR / EDMA_TPCC_EMRH, EDMA_TPCC_QEMR, or EDMA_TPCC_CCERR. See Section 11.4.9.4 *Error Interrupts* for additional information regarding error interrupts.

NOTE: While using EDMA_TPCC_IEVAL for shadow region completion interrupts, check that the EDMA_TPCC_IEVAL operated upon is from that particular shadow region memory map.

11.4.9.4 Error Interrupts

The EDMA_TPCC error registers provide the capability to differentiate error conditions (event missed, threshold exceed, etc.). Additionally, setting the error bits in these registers results in asserting the EDMA_TPCC error interrupt. If the EDMA_TPCC error interrupt is enabled in the device interrupt controller(s), then it allows the CPU(s) to handle the error conditions.

The EDMA_TPCC has a single error interrupt (EDMA_TPCC_ERRINT) that is asserted for all EDMA_TPCC error conditions. There are four conditions that cause the error interrupt:

- DMA missed events: for all 64 DMA channels. DMA missed events are latched in the event missed registers EDMA_TPCC_EMR / EDMA_TPCC_EMRH.
- QDMA missed events: for all 8 QDMA channels. QDMA missed events are latched in the QDMA event missed register EDMA_TPCC_QEMR.
- Threshold exceed: for all event queues. These are latched in EDMA_TPCC error register EDMA_TPCC_CCERR.
- TCC error: for outstanding transfer requests that are expected to return completion code EDMA_TPCC_OPT_n[22] TCCHEN or EDMA_TPCC_OPT_n[23] TCINTEN bit is set to 1, exceeding the maximum limit of 63. This is also latched in the EDMA_TPCC error register EDMA_TPCC_CCERR.

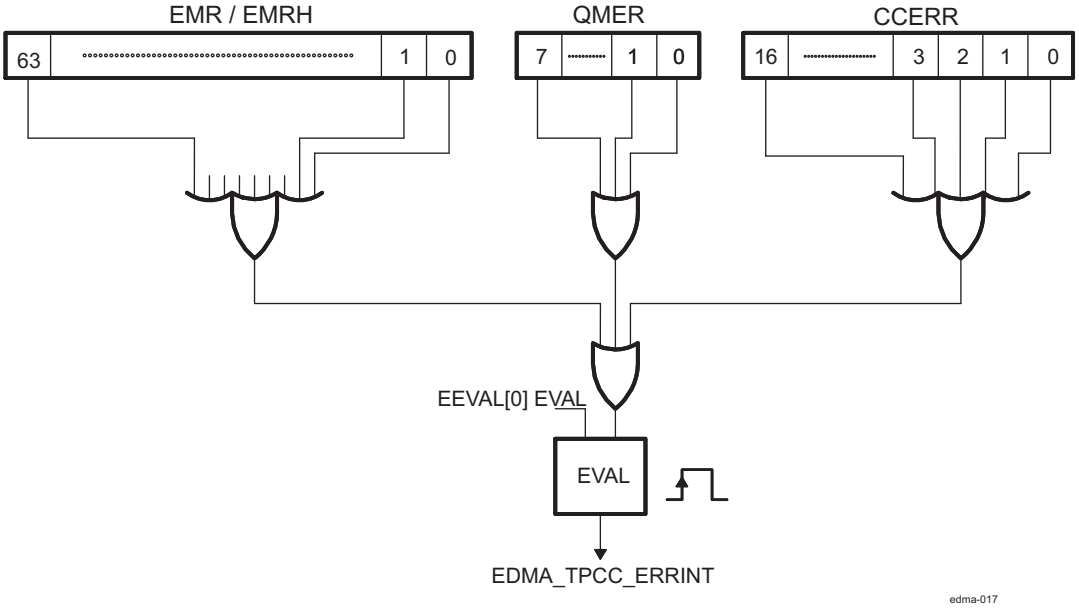
Figure 11-17 illustrates the EDMA_TPCC error interrupt generation operation.

If any of the bits are set in the error registers due to any error condition, the EDMA_TPCC_ERRINT is always asserted, as there are no enables for masking these error events. Similar to transfer completion interrupts (EDMA_TPCC_INT), the error interrupt also only pulses when the error interrupt condition transitions from no errors being set to at least one error being set. If additional error events are latched prior to the original error bits clearing, the EDMA_TPCC does not generate additional interrupt.

To reduce the burden on the software, there is an error evaluate register EDMA_TPCC_EEVAL that allows re-evaluation of pending set error events/bits, similar to the interrupt evaluate register EDMA_TPCC_IEVAL. Unlike the EDMA_TPCC_IEVAL functionality, the EDMA_TPCC_EEVAL register must be written with ‘1’ after any error interrupts are serviced (even when all pending errors are cleared) in order for subsequent errors to trigger a new interrupt.

NOTE: It is good practice to enable the error interrupt in the device interrupt controller and to associate an interrupt service routine with it to address the various error conditions appropriately. Doing so puts less burden on the software (polling for error status), it provides a good debug mechanism for unexpected error conditions.

Figure 11-17. Error Interrupt Operation



11.4.10 Memory Protection

The EDMA channel controller supports two kinds of memory protection: active and proxy.

11.4.10.1 Active Memory Protection

Active memory protection is a feature that allows or prevents read and write accesses to the EDMA_TPCC registers. Active memory protection is achieved by a set of memory protection permissions attribute EDMA_TPCC_MPPAN_k registers.

The EDMA_TPCC register map is divided into three categories:

- a global region.
- a global channel region.
- eight shadow regions.

Each shadow region consists of the respective shadow region registers and the associated PaRAM. For more detailed information regarding the contents of a shadow region, refer to Table 11-30 *EDMA_TPCC Registers Mapping Summary*.

Each of the eight shadow regions has an associated EDMA_TPCC_MPPAN_k registers that defines the specific requestor(s) and types of requests that are allowed to the regions resources.

The global channel region is also protected with a memory-mapped register EDMA_TPCC_MPPAG. The EDMA_TPCC_MPPAG applies to the global region and to the global channel region, except the other EDMA_TPCC_MPPAN_k registers themselves.

Table 11-20 shows the accesses that are allowed or not allowed to the EDMA_TPCC_MPPAG and EDMA_TPCC_MPPAN_k. The active memory protection uses the EDMA_TPCC_OPT_n[31] PRIV and EDMA_TPCC_OPT_n[27:24] PRIVID attributes of the EDMA peripheral modules. TheEDMA_TPCC_OPT_n[31] PRIV is the privilege level (i.e., user vs. supervisor).

The EDMA_TPCC_OPT_n[27:24] PRIVID refers to a privilege ID with a number that is associated with an EDMA peripheral modules.

Table 11-20. Allowed Accesses

Access	Supervisor	User
Read	Yes	Yes
Write	Yes	No

Table 11-21 describes the EDMA_TPCC_MPPAN_k register mapping for the shadow regions (which includes shadow region registers and PaRAM addresses).

The region-based EDMA_TPCC_MPPAN_k registers are used to protect accesses to the DMA shadow regions and the associated region PaRAM. Because there are eight regions, there are eight EDMA_TPCC_MPPAN_k region registers (MPPA[0-7]).

Table 11-21. MPPA Registers to Region Assignment

Register	Registers Protect	Address Range	PaRAM Protect[1]	Address Range
EDMA_TPCC_MPPAG	Global Range	0000h-1FFCh	N/A	N/A
EDMA_TPCC_MPPAN_k. MPPAN_0	DMA Shadow 0	2000h-21FCh	1st octant	4000h-47FCh
MPPAN_1	DMA Shadow 1	2200h-23FCh	2nd octant	4800h-4FFCh
MPPAN_2	DMA Shadow 2	2400h-25FCh	3rd octant	5000h-57FCh
MPPAN_3	DMA Shadow 3	2600h-27FCh	4th octant	5800h-5FFCh
MPPAN_4	DMA Shadow 4	2800h-29FCh	5th octant	6000h-67FCh
MPPAN_5	DMA Shadow 5	2A00h-2BFCh	6th octant	6800h-6FFCh
MPPAN_6	DMA Shadow 6	2C00h-2DFCh	7th octant	7000h-77FCh
MPPAN_7	DMA Shadow 7	2E00h-2FFCh	8th octant	7800h-7FFCh

(1) The PARAM region is divided into 8 regions referred to as an octant.

Example *Access denied.*

Write access to shadow region 7's event enable set register EDMA_TPCC_EESR:

1. The original value of the event enable register EDMA_TPCC_EER at address offset 0x1020 is 0x0.
2. The EDMA_TPCC_MPPAN_k. EDMA_TPCC_MPPAN_7[7] NS is set to prevent user level accesses (EDMA_TPCC_MPPAN_k. EDMA_TPCC_MPPAN_7[1] UW = 0, EDMA_TPCC_MPPAN_k. EDMA_TPCC_MPPAN_7[2] UR = 0), but it allows supervisor level accesses (EDMA_TPCC_MPPAN_k. EDMA_TPCC_MPPAN_7[4] SW = 1, EDMA_TPCC_MPPAN_k. EDMA_TPCC_MPPAN_7[5] SR = 1) with a privilege ID of 0. (EDMA_TPCC_MPPAN_k. EDMA_TPCC_MPPAN_7[10] AID0 = 1).
3. EDMA peripheral modules with a privilege ID of 0 attempts to perform a user-level write of a value of 0xFF00FF00 to shadow region 7's event enable set register EDMA_TPCC_EESR at address offset 0x2E30.

NOTE: The EDMA_TPCC_EER is a read-only register and the only way that write to it is by writing to the EDMA_TPCC_EESR. There is only one physical register for EDMA_TPCC_EER, EDMA_TPCC_EESR, etc. and that the shadow regions only provide to the same physical set.

4. Since the EDMA_TPCC_MPPAN_k. EDMA_TPCC_MPPAN_7[1] UW = 0, though the privilege ID of the write access is set to 0, the access is not allowed and the EDMA_TPCC_EER is not written too.

Table 11-22. Example Access Denied

Register	Value	Description
EDMA_TPCC_EER (offset 0x1020)	0x0000 0000	Value in EDMA_TPCC_EER to begin with.
EDMA_TPCC_EESR (offset 0x2E30)	0xFF00 FF00 ↓	Value attempted to be written to shadow region 7's EDMA_TPCC_EESR. This is done by an EDMA connected device module with a privilege level of User and Privilege ID of 0.
EDMA_TPCC_MPPAN_k (offset 0x082C)	0x0000 04B0	Memory Protection Filter EDMA_TPCC_MPPAN_k[10] AID0 = 1, EDMA_TPCC_MPPAN_k[1] UW = 0, EDMA_TPCC_MPPAN_k[2] UR = 0, EDMA_TPCC_MPPAN_k[4] SW = 1, EDMA_TPCC_MPPAN_k[5] SR = 1.
	X	Access Denied
EDMA_TPCC_EER (offset 0x1020)	0x0000 0000	Final value of EDMA_TPCC_EER

Example *Access Allowed*

Write access to shadow region 7's event enable set register EDMA_TPCC_EESR:

1. The original value of the event enable register EDMA_TPCC_EER at address offset 0x1020 is 0x0.
2. The EDMA_TPCC_MPPAN_k.EDMA_TPCC_MPPAN_7 is set to allow user-level accesses (EDMA_TPCC_MPPAN_k.EDMA_TPCC_MPPAN_7[1] UW = 1, EDMA_TPCC_MPPAN_k. EDMA_TPCC_MPPAN_7[2] UR = 1) and supervisor-level accesses (EDMA_TPCC_MPPAN_k. EDMA_TPCC_MPPAN_7[4] SW = 1, EDMA_TPCC_MPPAN_k. EDMA_TPCC_MPPAN_7[5] SR = 1) with a privilege ID of 0. (EDMA_TPCC_MPPAN_k. EDMA_TPCC_MPPAN_7[10] AID0 = 1).
3. EDMA peripheral modules with a privilege ID of 0, attempts to perform a user-level write of a value of 0xABCD0123 to shadow region 7's event enable set register EDMA_TPCC_EESR at address offset 0x2E30.

NOTE: The EDMA_TPCC_EER is a read-only register and the only way that write to it is by writing to the EDMA_TPCC_EESR. There is only one physical register for EDMA_TPCC_EER, EDMA_TPCC_EESR, etc. and that the shadow regions only provide to the same physical set.

4. Since the EDMA_TPCC_MPPAN_k. EDMA_TPCC_MPPAN_7[1] UW = 1 and EDMA_TPCC_MPPAN_k. MPPAN_7[10] AID0 = 1, the user-level write access is allowed.
5. The accesse to shadow region registers are masked by their respective EDMA_TPCC_DRAEM_k

register. In this example, the EDMA_TPCC_DRAEM_k. EDMA_TPCC_DRAEM_7 is set of 0x9FF00FC2.

6. The value finally written to EDMA_TPCC_EER is 0x8BC00102.

Table 11-23. Example Access Allowed

Register	Value	Description
EDMA_TPCC_EER (offset 0x1020)	0x0000 0000	Value in EER to begin with.
EDMA_TPCC_EESR (offset 0x2E30)	0xFF00 FF00	Value attempted to be written to shadow region 7's EESR. This is done by an EDMA peripheral module with a privilege level of User and Privilege ID of 0.
EDMA_TPCC_MPPAN_k. EDMA_TPCC_MPPAN_7 (offset 0x082C)	0x0000 04B3	Memory Protection Filter EDMA_TPCC_MPPAN_k[10] AID = 1, EDMA_TPCC_MPPAN_k. EDMA_TPCC_MPPAN_7[1] UW = 1, EDMA_TPCC_MPPAN_k. EDMA_TPCC_MPPAN_7[2] UR = 1, EDMA_TPCC_MPPAN_k. EDMA_TPCC_MPPAN_7[4] SW = 1, EDMA_TPCC_MPPAN_k. EDMA_TPCC_MPPAN_7[5] SR = 1.
	√ ↓	Access allowed.
EDMA_TPCC_DRAEM_k. EDMA_TPCC_DRAEM_7 (offset 0x0378)	0x9FF0 0FC2 ↓	DMA Region Access Enable Filter
EDMA_TPCC_EESR (offset 0x2E30)	0x8BC0 0102 ↓	Value written to shadow region 7's EESR. This is done by an EDMA peripheral module with a privilege level of User and a Privilege ID of 0.
EDMA_TPCC_EER (offset 0x1020)	↓ 0xBC0 0102	Final value of EER.

11.4.10.2 Proxy Memory Protection

Proxy memory protection allows an EDMA transfer programmed by a given peripheral module connected to EDMA, to have its permissions travel with the transfer through the EDMA_TPTC. The permissions travel along with the read transactions to the source and the write transactions to the destination endpoints. The EDMA_TPCC_OPT_n[31] PRIV bit and EDMA_TPCC_OPT_n[27:24] PRIVID bit is set with the peripheral module's PRIV value and PRIVID values, respectively, when any part of the PaRAM set is written.

The EDMA_TPCC_OPT_n[31] PRIV is the privilege level (i.e., user vs. supervisor). The EDMA_TPCC_OPT_n[27:24] PRIVID refers to a privilege ID with a number that is associated with an peripheral module connected to EDMA.

These options are part of the TR that are submitted to the transfer controller. The transfer controller uses the above values on their respective read and write command bus so that the target endpoints can perform memory protection checks based on these values.

Consider a parameter set that is programmed by a CPU in user privilege level for a simple transfer with the source buffer on an L2 page and the destination buffer on an L1D page. The EDMA_TPCC_OPT_n[31] PRIV is 0 for user-level and the CPU has a EDMA_TPCC_OPT_n[27:24] PRIVID to 0.

The PaRAM set is shown in Figure 11-18.

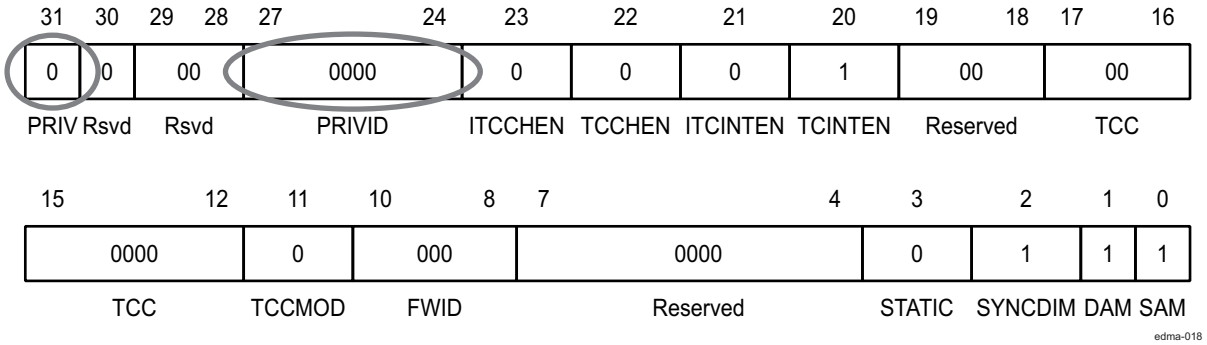
Figure 11-18. PaRAM Set Content for Proxy Memory Protection Example

(a) EDMA Parameters

Parameter Contents		Parameter	
0010 0007h		Channel Options Parameter (OPT)	
009F 0000h		Channel Source Address (SRC)	
0001h	0004h	Count for 2nd Dimension (BCNT)	Count for 1st Dimension (ACNT)
00F0 7800h		Channel Destination Address (DST)	
0001h	0001h	Destination BCNT Index (DBIDX)	Source BCNT Index (SBIDX)
0000h	FFFFh	BCNT Reload (BCNTRLD)	Link Address (LINK)
0001h	1000h	Destination CCNT Index (DCIDX)	Source CCNT Index (SCIDX)
0000h	0001h	Reserved	Count for 3rd Dimension (CCNT)

Figure 11-19. Channel Options Parameter (OPT) Example

(b) Channel Options Parameter (OPT_n) Content



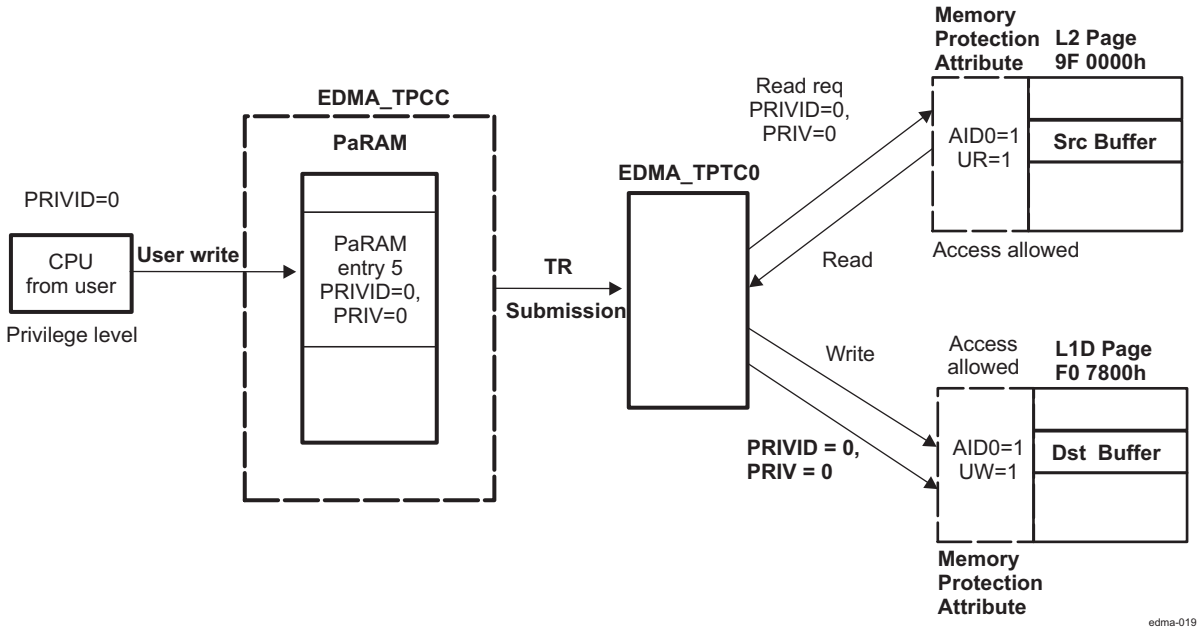
The EDMA_TPCC_OPT_n[31] PRIV and EDMA_TPCC_OPT_n[27:24] PRIVID information travels along with the read and write requests that are issued to the source and destination memories.

For example, if the access attributes that are associated with the L2 page with the source buffer only allow supervisor read, write accesses EDMA_TPCC_MPPAN_k[4] SW and EDMA_TPCC_MPPAN_k[5] SR, the user-level read request above is refused. Similarly, if the access attributes that are associated with the L1D page with the destination buffer only allow supervisor read and write accesses (EDMA_TPCC_MPPAN_k[4] SW, EDMA_TPCC_MPPAN_k[5] SR), the user-level write request above is refused. For the transfer to succeed, the source and destination pages must have user-read and user-write permissions, respectively, along with allowing accesses from a PRIVID = 0.

Because the privilege level and privilege identification travel with the read and write requests, EDMA acts as a proxy.

Figure 11-20 illustrates the propagation of EDMA_TPCC_OPT_n[31] PRIV and EDMA_TPCC_OPT_n[27:24] PRIVID at the boundaries of all the interacting entities (CPU, EDMA_TPCC, EDMA_TPTCs, and slave memories).

Figure 11-20. Proxy Memory Protection Example



11.4.11 Event Queue(s)

Event queues are a part of the EDMA channel controller. Event queues form the interface between the event detection logic in the EDMA_TPCC and the transfer request (TR) submission logic of the EDMA_TPCC. Each queue is 16 entries deep. Each event queue can queue a maximum of 16 events. If there are more than 16 events, then the events that cannot find a place in the event queue remain set in the associated event register and the CPU does not stall.

There are two event queues for the device: Queue0, Queue1. Events in Queue0 result in submission of its associated transfer requests (TRs) to TC0. The transfer requests that are associated with events in Queue1 are submitted to TC1.

An event that wins prioritization against other DMA and/or QDMA pending events is placed at the tail of the appropriate event queue. Each event queue is serviced in FIFO order. Once the event reaches the head of its queue and the corresponding transfer controller is ready to receive another TR, the event is de-queued and the PaRAM set corresponding to the de-queued event is processed and submitted as a transfer request packet (TRP) to the associated EDMA transfer controller.

Queue0 has highest priority and Queue1 has the lowest priority, if Queue0 and Queue1 both have at least one event entry and if both TC0 and TC1 can accept transfer requests, then the event in Queue0 is de-queued first and its associated PaRAM set is processed and submitted as a transfer request (TR) to TC0.

Refer to Section 11.4.11.4 for system-level performance considerations. All of the event entries in all of the event queues are software readable (not writeable) by accessing the event entry registers EDMA_TPCC_Q0E_p and EDMA_TPCC_Q1E_p. Each event entry register characterizes the queued event in terms of the type of event (manual, event, chained or auto-triggered) and the event number. Refer to Section 11.7.2.2.1 *EDMA_TPCC Register Description* for EDMA_TPCC_Q0E_p / EDMA_TPCC_Q1E_p descriptions of the bit fields.

11.4.11.1 DMA/QDMA Channel to Event Queue Mapping

Each of the 64 DMA channels and eight QDMA channels are programmed independently to map to a specific queue, using the DMA queue number register EDMA_TPCC_DMAQNUMN_k and the QDMA queue number register EDMA_TPCC_QDMAQNUM. The mapping of DMA/QDMA channels is critical to achieving the desired performance level for the EDMA and most importantly, in meeting real-time deadlines. Refer to Section 11.4.11.4 *System-level Performance Considerations*.

NOTE: If an event is ready to be queued and both the event queue and the EDMA transfer controller that is associated to the event queue are empty, then the event bypasses the event queue, and moves the PaRAM processing logic, and eventually to the transfer request submission logic for submission to the EDMA_TPTC. In this case, the event is not logged in the event queue status registers.

11.4.11.2 Queue RAM Debug Visibility

There are two event queues and each queue has 16 entries. These 16 entries are managed in a circular FIFO. There is a queue status register EDMA_TPCC_QSTATN_i associated with each queue. These along with all of the 16 entries per queue can be read via registers EDMA_TPCC_QSTATN_i and Q0E_p / Q1E_p, respectively.

These registers provide user visibility.

The event queue entry register (Q*x*E*y* Q0E_p / Q1E_p) uniquely identifies the specific event type (event-triggered, manually-triggered, chain-triggered, and QDMA events) along with the event number (for all DMA/QDMA event channels) that are in the queue or have been de-queued (passed through the queue).

Each of the 16 entries in the event queue are read using the EDMA_TPCC memory-mapped register. To see the history of the last 16 TRs that have been processed by the EDMA on a given queue, read the event queue registers. This provides user/software visibility and is helpful for debugging real-time issues (typically post-mortem), involving multiple events and event sources.

The queue status register (QSTAT*n* EDMA_TPCC_QSTATN_i) includes fields for the start pointer EDMA_TPCC_QSTATN_i[3:0] STRTPTR which provides the offset to the head entry of an event. It also includes a field called EDMA_TPCC_QSTATN_i[12:8] NUMVAL that provides the total number of valid entries residing in the event queue at a given instance of time. The EDMA_TPCC_QSTATN_i[3:0] STRTPTR is used to index appropriately into the 16 event entries. EDMA_TPCC_QSTATN_i[12:8] NUMVAL number of entries starting from STRTPTR are indicative of events still queued in the respective queue. The remaining entry must be read to determine what's already de-queued and submitted to the associated transfer controller.

11.4.11.3 Queue Resource Tracking

The EDMA_TPCC event queue includes watermarking/threshold logic that allows to keep track of maximum usage of all event queues. This is useful for debugging real-time deadline violations that may result from head-of-line blocking on a given EDMA event queue.

The maximum number of events are programed that the queue up in an event queue by programming the threshold value (between 0 to 15) in the queue watermark threshold A register EDMA_TPCC_QWMTHRA. The maximum queue usage is recorded actively in the watermark EDMA_TPCC_QSTATN_i[20:16] WM field of the queue status register, that keeps getting updated based on a comparison of number of valid entries, which is also visible in the EDMA_TPCC_QSTATN_i[12:8] NUMVAL bit and the maximum number of entries.

If the queue usage is exceeded, this status is visible in the EDMA_TPCC registers: the QTHRXCD*n* bits in the channel controller error register EDMA_TPCC_CCERR[7:0] and the EDMA_TPCC_QSTATN_i[24] THRXCD bit, where *n* stands for the event queue number. Any bits that are set in EDMA_TPCC_CCERR also generate an EDMA_TPCC error interrupt.

11.4.11.4 Performance Considerations

The main system bus infrastructure (L3) arbitrates bus requests from all of the masters (TCs, CPU(S), and other bus masters) to the shared slave resources (peripherals and memories).

The priorities of transfer requests (read and write commands) from the EDMA transfer controllers with respect to other masters within the device IRQ_CROSSBAR are programmed using the Control Module registers. The EDMA_TPCC_QUEPRI register has no affect.

Therefore, the priority of unloading queues has a secondary affect compared to the priority of the transfers as they are executed by the EDMA_TPTC (dictated by the priority set using the Control Module registers, refer to Section 13.5 *Control Module Register Manual* in Chapter 13 *Control Module* chapter).

11.4.12 EDMA Transfer Controller (EDMA_TPTC)

The EDMA channel controller is the user-interface of the EDMA and the EDMA transfer controller (EDMA_TPTC) is the data movement engine of the EDMA controller. The EDMA_TPCC submits transfer requests (TR) to the EDMA_TPTC and the EDMA_TPTC performs the data transfers dictated by the TR, so the EDMA_TPTC is a slave to the EDMA_TPCC.

11.4.12.1 Architecture Details

11.4.12.1.1 Command Fragmentation

The TC read and write controllers in conjunction with the source and destination register sets are responsible for issuing optimally-sized reads and writes to the slave endpoints. An optimally-sized command is defined by the transfer controller default burst size (DBS), which is defined in Section 11.4.12.5 *EDMA_TPTC Configuration*.

The EDMA_TPTC attempts to issue the largest possible command size as limited by the DBS value or the EDMA_TPCC_ABCNT_n[15:0] ACNT and EDMA_TPCC_ABCNT_n[31:16] BCNT value of the TR. EDMA_TPTC obeys the following rules:

- The read/write controllers always issue commands less than or equal to the DBS value.
- The first command of a 1D transfer command always aligns the address of subsequent commands to the DBS value.

Table 11-24 lists the TR segmentation rules that are followed by the EDMA_TPTC. In summary, if the EDMA_TPCC_ABCNT_n[15:0] ACNT value is larger than the DBS value, then the EDMA_TPTC breaks the EDMA_TPCC_ABCNT_n[15:0] ACNT array into DBS-sized commands to the source/destination addresses. Each EDMA_TPCC_ABCNT_n[31:16] BCNT number of arrays are then serviced in succession.

For BCNT arrays of ACNT bytes (that is, a 2D transfer), if the EDMA_TPCC_ABCNT_n[15:0] ACNT value is less than or equal to the DBS value, then the TR may be optimized into a 1D-transfer in order to maximize efficiency. The optimization takes place if the EDMA_TPTC recognizes that the 2D-transfer is organized as a single dimension (EDMA_TPCC_ABCNT_n[15:0] ACNT == EDMA_TPCC_BIDX_n) and the ACNT value is a power of 2.

Table 11-24 lists conditions in which the optimizations are performed.

Table 11-24. Read/Write Command Optimization Rules

ACNT ≤ DBS	ACNT is power of 2	BIDX = ACNT	BCNT ≤ 1023	SAM/DAM = Increment	Description
Yes	Yes	Yes	Yes	Yes	Optimized
No	x	x	x	x	Not Optimized
x	No	x	x	x	Not Optimized
x	x	No	x	x	Not Optimized
x	x	x	No	x	Not Optimized
x	x	x	x	No	Not Optimized

11.4.12.1.2 TR Pipelining

TR pipelining refers to the ability of the source active set to proceed ahead of the destination active set. Essentially, the reads for a given TR may already be in progress while the writes of a previous TR may not have completed.

The number of outstanding TRs is limited by the number of destination FIFO register entries.

TR pipelining is useful for maintaining throughput on back-to-back small TRs. It minimizes the startup overhead because reads start in the background of a previous TR writes.

Example 11-4. Command Fragmentation (DBS = 64)

The pseudo code:

1. EDMA_TPTCn_PCNT[15:0] ACNT = 8, EDMA_TPTCn_PCNT[31:16] BCNT = 8, EDMA_TPTCn_PBIDX[15:0] SBIDX = 8, EDMA_TPTCn_PBIDX[31:16] DBIDX = 10, EDMA_TPTCn_PSRC[31:0] SADDR = 64, EDMA_TPTCn_SADST[31:0] DADDR = 191

   Read Controller: This is optimized from a 2D-transfer to a 1D-transfer such that the read side is equivalent to EDMA_TPTCn_PCNT[15:0] ACNT = 64, EDMA_TPTCn_PCNT[31:16] BCNT = 1.

   Cmd0 = 64 byte

   Write Controller: Because DBIDX != ACNT, it is not optimized.

   Cmd0 = 8 byte, Cmd1 = 8 byte, Cmd2 = 8 byte, Cmd3 = 8 byte, Cmd4 = 8 byte, Cmd5 = 8 byte, Cmd6 = 8 byte, Cmd7 = 8 byte.

2. EDMA_TPTCn_PCNT[15:0] ACNT=128, EDMA_TPTCn_PCNT[31:16] BCNT = 1, EDMA_TPTCn_PSRC[31:0] SADDR = 63, EDMA_TPTCn_SADST[31:0] DADDR = 513

   Read Controller: Read address is not aligned.

   Cmd0 = 1 byte, (now the SADDR is aligned to 64 for the next command)

   Cmd1 = 64 bytes

   Cmd2 = 63 bytes

   Write Controller: The write address is also not aligned.

   Cmd0 = 63 bytes, (now the DADDR is aligned to 64 for the next command)

   Cmd1 = 64 bytes

   Cmd2 = 1 byte

11.4.12.1.3 Performance Tuning

By default, reads are as issued as fast as possible. In some cases, the reads issued by the EDMA_TPTC could fill the available command buffering for a slave, delaying other (potentially higher priority) masters from successfully submitting commands to that slave. The rate at which read commands are issued by the EDMA_TPTC is controlled by the EDMA_TPTCn_RDRATE register. The EDMA_TPTCn_RDRATE register defines the number of cycles that the EDMA_TPTC read controller waits before issuing subsequent commands for a given TR, thus minimizing the chance of the EDMA_TPTC consuming all available slave resources. The EDMA_TPTCn_RDRATE[2:0] RDRATE value must be set to a relatively small value if the transfer controller is targeted for high priority transfers and to a higher value if the transfer controller is targeted for low priority transfers.

In contrast, the Write Interface does not have any performance turning knobs because writes always have an interval between commands as write commands are submitted along with the associated write data.

11.4.12.2 Memory Protection

The transfer controller plays an important role in handling proxy memory protection. There are two access properties associated with a transfer: for instance, the privilege id (system-wide identification assigned to a master) of the master initiating the transfer, and the privilege level (user versus supervisor) used to program the transfer. This information is maintained in the PaRAM set when it is programmed in the channel controller. When a TR is submitted to the transfer controller, this information is made available to the EDMA_TPTC and used by the EDMA_TPTC while issuing read and write commands. The read or write commands have the same privilege identification, and privilege level as that programmed in the EDMA transfer in the channel controller.

11.4.12.3 Error Generation

Errors are generated if enabled under three conditions:

- EDMA_TPTC detection of an error signaled by the source or destination address.
- Attempt to read or write to an invalid address in the configuration memory map.

- Detection of a constant addressing mode TR violating the constant addressing mode transfer rules (the source/destination addresses and source/destination indexes must be aligned to 32 bytes).

Either or all error types may be disabled. If an error bit is set and enabled, the error interrupt for the concerned transfer controller is generated.

11.4.12.4 Debug Features

The DMA program register set, DMA source active register set, and the destination FIFO register set are used to derive a brief history of TRs serviced through the transfer controller.

Additionally, the EDMA_TPTC status register EDMA_TPTCn_TCSTAT has dedicated bit fields to indicate the ongoing activity within different parts of the transfer controller:

- The EDMA_TPTCn_TCSTAT[1] SRCACTV bit indicates whether the source active set is active.
- The EDMA_TPTCn_TCSTAT[6:4] DSTACTV bit indicates the number of TRs resident in the destination register active set at a given instance.
- The EDMA_TPTCn_TCSTAT[0] PROGBUSY bit indicates whether a valid TR is present in the DMA program set.

NOTE: If the TRs are in progression, it must realize that there is a chance that the values read from the EDMA_TPTC status registers will be inconsistent since the EDMA_TPTC changes the values of these registers due to ongoing activities.

It is recommended that to ensure no additional submission of TRs to the EDMA_TPTC in order to facilitate ease of debug.

11.4.12.4.1 Destination FIFO Register Pointer

The destination FIFO register pointer is implemented as a circular buffer with the start pointer being EDMA_TPTCn_TCSTAT[12:11] DFSTRTPTR and a buffer depth of usually 2 or 4. The EDMA_TPTC maintains two important status details in EDMA_TPTCn_TCSTAT that are used during advanced debugging, if necessary. The EDMA_TPTCn_TCSTAT[12:11] DFSTRTPTR is a start pointer, the index to the head of the destination FIFO register. The EDMA_TPTCn_TCSTAT[6:4] DSTACTV is a counter for the number of valid (occupied) entries. These registers are used to get a brief history of transfers.

Examples of some register field values and their interpretation:

- EDMA_TPTCn_TCSTAT[12:11] DFSTRTPTR = 0x0 and EDMA_TPTCn_TCSTAT[6:4] DSTACTV = 0x0 implies that no TRs are stored in the destination FIFO register.
- EDMA_TPTCn_TCSTAT[12:11] DFSTRTPTR = 0x1 and EDMA_TPTCn_TCSTAT[6:4] DSTACTV = 0x2 implies that two TRs are present. The first pending TR is read from the destination FIFO register entry 1 and the second pending TR is read from the destination FIFO register entry 2.
- EDMA_TPTCn_TCSTAT[12:11] DFSTRTPTR = 0x3 and EDMA_TPTCn_TCSTAT[6:4] DSTACTV = 0x2 implies that two TRs are present. The first pending TR is read from the destination FIFO register entry 3 and the second pending TR is read from the destination FIFO register entry 0.

11.4.12.5 EDMA_TPTC Configuration

Table 11-25 provides the configuration of the individual EDMA transfer controllers present on the device. The DBS for each transfer controller is defined by Control Module register (CTRL_CORE_CONTROL_IO_1) settings.

Table 11-25. EDMA Transfer Controller Configurations

Name	TC0	TC1
EDMA_TPTCn_TCCFG[2:0] FIFOSIZE	1024 bytes	1024 bytes
EDMA_TPTCn_TCCFG[5:4] BUSWIDTH	16 bytes	16 bytes
EDMA_TPTCn_TCCFG[9:8] DSTREGDEPTH	4 entries	4 entries

Table 11-25. EDMA Transfer Controller Configurations (continued)

Name	TC0	TC1
DBS	Defined by CTRL_CORE_CONTROL_IO_1[9:8] TC0_DEFAULT_BURST_SIZE	Defined by CTRL_CORE_CONTROL_IO_1[13:12] TC1_DEFAULT_BURST_SIZE

11.4.13 Event Dataflow

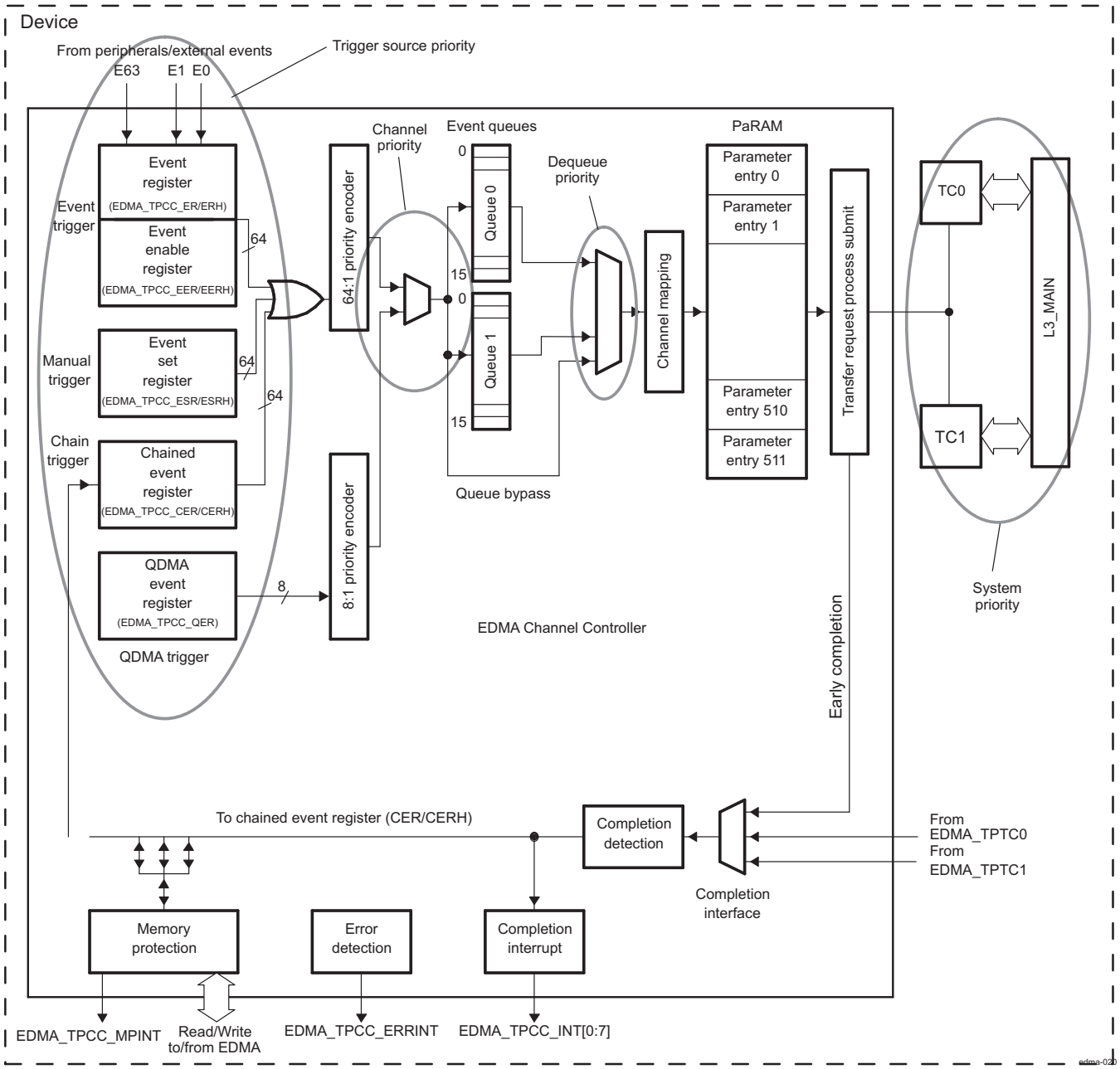
This section summarizes the data flow of a single event, from the time the event is latched to the channel controller to the time the transfer completion code is returned. The following steps list the sequence of EDMA_TPCC activity:

1. Event is asserted from an external source (peripheral or external interrupt). This also is similar for a manually-triggered, chained-triggered, or QDMA-triggered event. The event is latched into the EDMA_TPCC_ER[31:0]E*n* / EDMA_TPCC_ERH[31:0] E*n* (or EDMA_TPCC_CER[31:0] E*n* / EDMA_TPCC_CERH[31:0] E*n*, EDMA_TPCC_ESR[31:0] E*n* / EDMA_TPCC_ESRH[31:0] E*n*, EDMA_TPCC_QER[7:0] E*n*) bit.
2. Once an event is prioritized and queued into the appropriate event queue, the EDMA_TPCC_SER[31:0] E*n* \ EDMA_TPCC_SERH[31:0] E*n* (or EDMA_TPCC_QSER[7:0] E*n*) bit is set to inform the event prioritization / processing logic to disregard this event since it is already in the queue. Alternatively, if the transfer controller and the event queue are empty, then the event bypasses the queue.
3. The EDMA_TPCC processing and the submission logic evaluates the appropriate PaRAM set and determines whether it is a non-null and non-dummy transfer request (TR).
4. The EDMA_TPCC clears the EDMA_TPCC_ER[31:0] E*n*/ EDMA_TPCC_ERH[31:0] E*n* (or EDMA_TPCC_CER[31:0] E*n* / EDMA_TPCC_CERH[31:0] E*n*, EDMA_TPCC_ESR[31:0]E*n* / EDMA_TPCC_ESRH[31:0] E*n*, EDMA_TPCC_QER[31:0] E*n*) bit and the EDMA_TPCC_SER[31:0] E*n*/EDMA_TPCC_SERH[31:0] E*n* bit as soon as it determines the TR is non-null. In the case of a null set, the EDMA_TPCC_SER[31:0] E*n*/ EDMA_TPCC_SERH[31:0] E*n* bit remains set. It submits the non-null/non-dummy TR to the associated transfer controller. If the TR was programmed for early completion, the EDMA_TPCC immediately sets the interrupt pending register (EDMA_TPCC_IPR[31:0] I[TCC] / EDMA_TPCC_IPRH[31:0] I[TCC] - 32).
5. If the TR was programmed for normal completion, the EDMA_TPCC sets the interrupt pending register (EDMA_TPCC_IPR[31:0] I[TCC] / EDMA_TPCC_IPRH[31:0] I[TCC]) when the EDMA_TPTC informs the EDMA_TPCC about completion of the transfer (returns transfer completion codes).
6. The EDMA_TPCC programs the associated EDMA_TPTC's Program Register Set with the TR.
7. The TR is then passed to the Source Active set and the DST FIFO Register Set, if both the register sets are available.
8. The Read Controller processes the TR by issuing read commands to the source slave endpoint. The Read Data lands in the Data FIFO of the EDMA_TPTC*n*.
9. As soon as sufficient data is available, the Write Controller begins processing the TR by issuing write commands to the destination slave endpoint.
10. This continues until the TR completes and the EDMA_TPTC*n* then signals completion status to the EDMA_TPCC.

11.4.14 EDMA controller Prioritization

The EDMA controller has many implementation rules to deal with concurrent events/channels, transfers, etc. The following subsections detail various arbitration details whenever there might be occurrence of concurrent activity. Figure 11-21 shows the different places EDMA priorities come into play.

Figure 11-21. EDMA Prioritization



11.4.14.1 Channel Priority

The EDMA event registers EDMA_TPCC_ER and EDMA_TPCC_ERH capture up to 64 events, the QDMA event register EDMA_TPCC_QER captures QDMA events for all QDMA channels therefore, it is possible for events to occur simultaneously on the DMA/QDMA event inputs. For events arriving simultaneously, the event associated with the lowest channel number is prioritized for submission to the event queues (for DMA events, channel 0 has the highest priority and channel 63 has the lowest priority, for QDMA events, channel 0 has the highest priority and channel 7 has the lowest priority). This mechanism only sorts simultaneous events for submission to the event queues.

If a DMA and QDMA event occurs simultaneously, the DMA event always has prioritization against the QDMA event for submission to the event queues.

11.4.14.2 Trigger Source Priority

If a EDMA channel is associated with more than one trigger source (event trigger, manual trigger, and chain trigger), and if multiple events are set simultaneously for the same channel (EDMA_TPCC_ER[31:0] E*n* = 1, EDMA_TPCC_ESR[31:0] E*n* = 1 , EDMA_TPCC_CER[31:0] E*n* = 1) , then the EDMA_TPCC always services these events in the following priority order: event trigger (via EDMA_TPCC_ER) is higher priority than chain trigger (via EDMA_TPCC_CER) and chain trigger is higher priority than manual trigger (via EDMA_TPCC_ESR).

This implies that if for channel 0, both EDMA_TPCC_ER[0] E0 = 1 and EDMA_TPCC_CER[0] E0 = 1 at the same time, then the EDMA_TPCC_ER[0] E0 event is always queued before the EDMA_TPCC_CER[0] E0 event.

11.4.14.3 Dequeue Priority

The priority of the associated transfer request (TR) is further mitigated by which event queue is being used for event submission (dictated by EDMA_TPCC_DMAQNUMN_k and EDMA_TPCC_QDMAQNUM). For submission of a TR to the transfer request, events need to be de-queued from the event queues. Queue 0 has the highest dequeue priority and queue 1 the lowest.

11.4.15 EDMA Power, Reset and Clock Management

11.4.15.1 Clock and Power Management

The EDMA channel controller and transfer controller are clocked from L3MIAN1_L3_GICLK interface clock. The EDMA system runs at the L3 clock frequency.

The Auto clock gating for the EDMA_TPCC module is controlled by the EDMA_TPCC_CLKGDIS[0] CLKGDIS bit at the module level.

The L3MIAN1_L3_GICLK interface clock to EDMA controller's modules are controlled by the following registers in the PRCM module:

- PRCM.CM_L3MAIN1_TPCC_CLKCTRL - manages EDMA_TPCC module clock.
- PRCM.CM_L3MAIN1_TPTC1_CLKCTRL - manages EDMA_TPTC0 module clock.
- PRCM.CM_L3MAIN1_TPTC2_CLKCTRL - manages EDMA_TPTC1 module clock.

EDMA_TPCC and EDMA_TPTC0 and EDMA_TPTC1 modules have wakeup depandaces to several device modules. The wakeup dependency based on EDMA modules service requests are controlled by registers in PRCM module:

- PRCM.PM_L3MAIN1_TPCC_WKDEP - controls wakeup dependency based on TPCC service requests.
- PRCM.PM_L3MAIN1_TPTC1_WKDEP - controls wakeup dependency based on TPTC0 service requests.
- PRCM.PM_L3MAIN1_TPTC2_WKDEP - controls wakeup dependency based on TPTC1 service requests.

  The EDMA_TPCC, EDMA_TPTC0 and EDMA_TPTC1 can be placed in reduced-power modes to conserve power during periods of low activity. The power management of the peripheral is controlled by the PRCM module. The PRCM acts as a master controller for power management for all peripherals on the device.

  The EDMA controller can be idled on receiving a clock stop request from the PRCM. The requests to EDMA_TPCC and EDMA_TPTC0 and EDMA_TPTC1 are separate. In general, it should be verified that there are no pending activities in the EDMA controller

NOTE: When EDMA controler modules no longer require the interface clock, software can disable it at the PRCM level by configuring the MODULEMODE bit field (PRCM.CM_L3MAIN1_TPCC_CLKCTRL[1:0], PRCM.CM_L3MAIN1_TPTC1_CLKCTRL[1:0], PRCM.CM_L3MAIN1_TPTC2_CLKCTRL[1:0]) in the PRCM registers. The clock is effectively cut, provided the other modules that receive it do not require it.

At the PRCM level, when all the conditions to shut off the L3MIAN1_L3_GICLK clock are met the PRCM module automatically launches a hardware handshake protocol to ensure EDMA modules are ready to have this clock switched off. Namely, the PRCM module asserts an IDLE request to the EDMA modules. For more information, refer to Chapter 3, *Power, Reset, and Clock Management*.

11.4.15.2 Reset Considerations

A hardware resets the EDMA (EDMA_TPCC, EDMA_TPTC0 and EDMA_TPTC1) and the EDMA configuration registers. The PaRAM memory contents are undefined after device reset and it should not rely on parameters to be reset to a known state. The PaRAM entry must be initialized to a desired value before it is used. The EDMA modules are reset by CORE_RET_RST reset signal from PRCM.

11.4.16 Emulation Considerations

During debug when using the emulator, the CPU(s) may be halted on an execute packet boundary for single-stepping, benchmarking, profiling, or other debug purposes. During an emulation halt, the EDMA channel controller and transfer controller operations continue. Events continue to be latched and processed and transfer requests continue to be submitted and serviced.

Since EDMA is involved in servicing multiple master and slave peripherals, it is not feasible to have an independent behavior of the EDMA for emulation halts. EDMA functionality would be coupled with the peripherals it is servicing, which might have different behavior during emulation halts.

11.5 EDMA Transfer Examples

The EDMA channel controller performs a variety of transfers depending on the parameter configuration. The following sections provide a description and PaRAM configuration for some typical use case scenarios.

11.5.1 Block Move Example

The most basic transfer performed by the EDMA is a block move. During device operation it is often necessary to transfer a block of data from one location to another, usually between on-chip and off-chip memory.

In this example, a section of data is to be copied from external memory to internal L2 SRAM as shown in Figure 11-22.

The source address for the transfer is set to the start of the data block in external memory, and the destination address is set to the start of the data block in L2. If the data block is less than 64K bytes, the PaRAM configuration shown in Figure 11-23 holds true with the synchronization type set to A-synchronized and indexes cleared to 0. If the amount of data is greater than 64K bytes, EDMA_TPCC_ABCNT_n[31:16] BCNT and the B-indexes need to be set appropriately with the synchronization type set to AB-synchronized. The EDMA_TPCC_OPT_n[3] STATIC bit is set to prevent linking.

This transfer example may also be set up using QDMA. For successive transfer submissions, of a similar nature, the number of cycles used to submit the transfer are fewer depending on the number of changing transfer parameters. The QDMA trigger word must be programed to be the highest numbered offset in the PaRAM set that undergoes change.

Figure 11-23 shows the parameters Block Move transfer.

Figure 11-22. Block Move Example

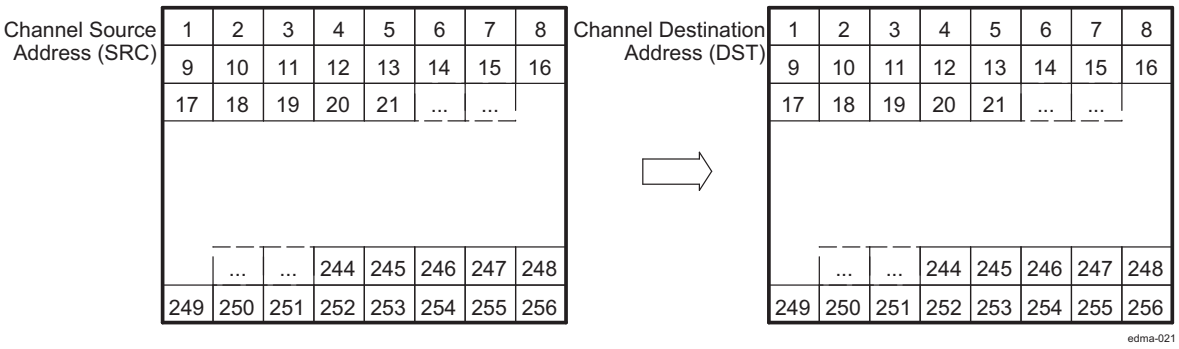


Figure 11-23. Block Move Example PaRAM Configuration

(a) EDMA Parameters

Parameter Contents	
0010 0008h	
Channel Source Address (SRC)	
0001h	0100h
Channel Destination Address (DST)	
0000h	0000h
0000h	FFFFh
0000h	0000h
0000h	0001h

Parameter	
Channel Options Parameter (OPT)	
Channel Source Address (SRC)	
Count for 2nd Dimension (BCNT)	Count for 1st Dimension (ACNT)
Channel Destination Address (DST)	
Destination BCNT Index (DBIDX)	Source BCNT Index (SBIDX)
BCNT Reload (BCNTRLD)	Link Address (LINK)
Destination CCNT Index (DCIDX)	Source CCNT Index (SCIDX)
Reserved	Count for 3rd Dimension (CCNT)

(b)Channel Options Parameter (OPT) Content

- EDMA_TPCC_OPT_n[3] STATIC = 0x1
- EDMA_TPCC_OPT_n[20] TCINTEN = 0x1

11.5.2 Subframe Extraction Example

The EDMA can efficiently extract a small frame of data from a larger frame of data. By performing a 2D-to-1D transfer, the EDMA retrieves a portion of data for the CPU to process. In this example, a 640 × 480-pixel frame of video data is stored in external memory. Each pixel is represented by a 16-bit halfword. The CPU extracts a 16 × 12-pixel subframe of the image for processing. To facilitate more efficient processing time by the CPU, the EDMA places the subframe in internal L2 SRAM. Figure 11-24 shows the transfer of a subframe from external memory to L2.

The same PaRAM entry options are used for QDMA channels, as well as DMA channels. The EDMA_TPCC_OPT_n[3] STATIC bit is set to prevent linking. For successive transfers, only changed parameters need to be programmed before triggering the channel.

Figure 11-25 shows the parameters for Subframe Extraction transfer.

Figure 11-24. Subframe Extraction Transfer

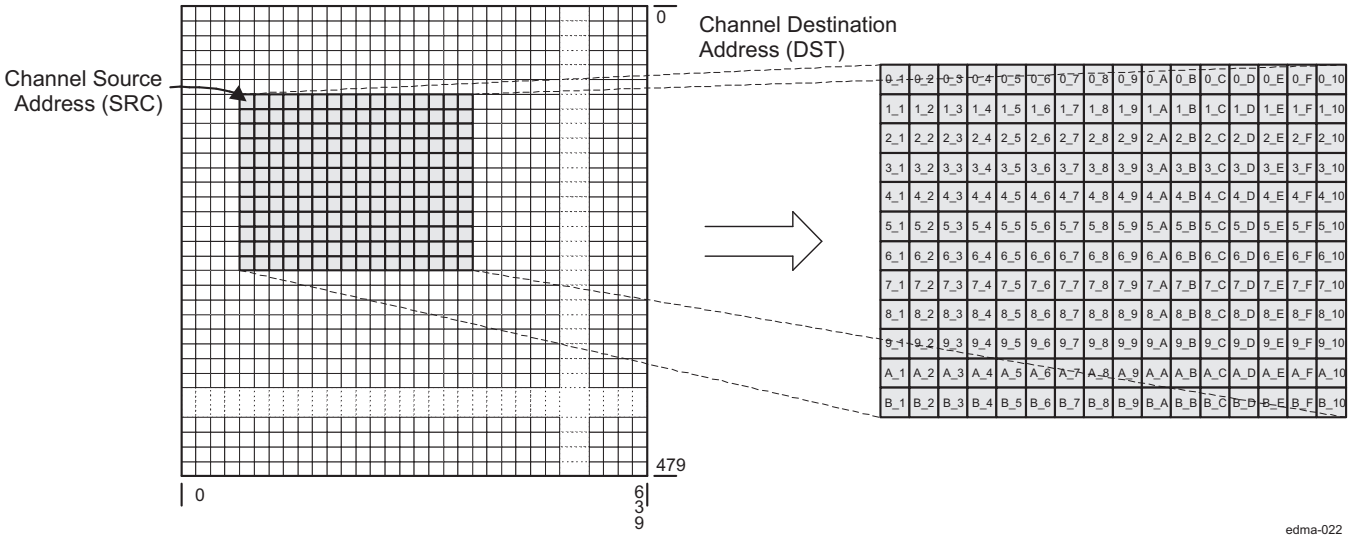


Figure 11-25. Subframe Extraction Example PaRAM Configuration

(a) EDMA Parameters

Parameter Contents	
0010 000Ch	
Channel Source Address (SRC)	
000Ch	0020h
Channel Destination Address (DST)	
0020h	0500h
0000h	FFFFh
0000h	0000h
0000h	0001h

Parameter	
Channel Options Parameter (OPT)	
Channel Source Address (SRC)	
Count for 2nd Dimension (BCNT)	Count for 1st Dimension (ACNT)
Channel Destination Address (DST)	
Destination BCNT Index (DBIDX)	Source BCNT Index (SBIDX)
BCNT Reload (BCNTRLD)	Link Address (LINK)
Destination CCNT Index (DCIDX)	Source CCNT Index (SCIDX)
Reserved	Count for 3rd Dimension (CCNT)

(b)Channel Options Parameter (OPT) Content

- EDMA_TPCC_OPT_n[2] SYNCDIM = 0x1
- EDMA_TPCC_OPT_n[3] STATIC = 0x1
- EDMA_TPCC_OPT_n[20] TCINTEN = 0x1

11.5.3 Data Sorting Example

Many applications require the use of multiple data arrays, it is often desirable to have the arrays arranged such that the first elements of each array are adjacent, the second elements are adjacent, and so on. Often this is not how the data is presented to the device. Either data is transferred via a peripheral with the data arrays arriving one after the other or the arrays are located in memory with each array occupying a portion of contiguous memory spaces. For these instances, the EDMA can reorganize the data into the desired format.

To determine the parameter set values, the following need to be considered:

- ACNT - Program this to be the size in bytes of an element.
- BCNT - Program this to be the number of elements in a frame.
- CCNT - Program this to be the number of frames.
- SBIDX - Program this to be the size of the element or ACNT.
- DBIDX - CCNT × ACNT
- SCIDX - ACNT × BCNT
- DCIDX - ACNT

The synchronization type needs to be AB-synchronized and the EDMA_TPCC_OPT_n[3] STATIC bit is 0 to allow updates to the parameter set. It is advised to use normal EDMA channels for sorting.

It is not possible to sort this with a single trigger event. Instead, the channel can be programmed to be chained to itself. After BCNT elements get sorted, intermediate chaining could be used to trigger the channel again causing the transfer of the next BCNT elements and so on. Figure 11-27 shows the parameter set programming for this transfer, assuming channel 0 and an element size of 4 bytes.

Figure 11-26 shows the Data Sorting transfer

Figure 11-26. Data Sorting Example

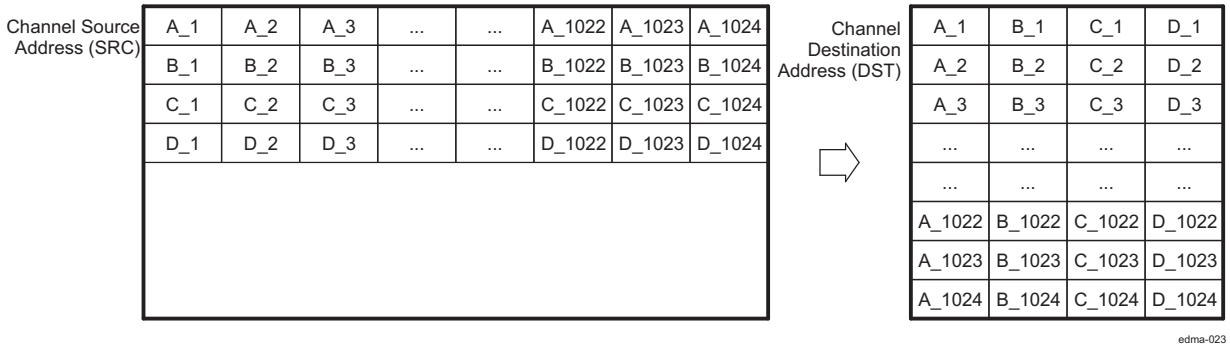


Figure 11-27. Data Sorting Example PaRAM Configuration

(a) EDMA Parameters

Parameter Contents	
0090 0004h	
Channel Source Address (SRC)	
0400h	0004h
Channel Destination Address (DST)	
0010h	0001h
0000h	FFFFh
0001h	1000h
0000h	0004h

Parameter	
Channel Options Parameter (OPT)	
Channel Source Address (SRC)	
Count for 2nd Dimension (BCNT)	Count for 1st Dimension (ACNT)
Channel Destination Address (DST)	
Destination BCNT Index (DSTBIDX)	Source BCNT Index (SRCBIDX)
BCNT Reload (BCNTRLD)	Link Address (LINK)
Destination CCNT Index (DSTCIDX)	Source CCNT Index (SRCCIDX)
Reserved	Count for 3rd Dimension (CCNT)

(b) Channel Options Parameter (OPT) Content

- EDMA_TPCC_OPT_n[2] SYNCDIM = 0x1
- EDMA_TPCC_OPT_n[20] TCINTEN = 0x1
- EDMA_TPCC_OPT_n[23] ITCCHEN = 0x1

11.5.4 Peripheral Servicing Example

The EDMA channel controller also services peripherals in the background of CPU operation, without requiring any CPU intervention. Through proper initialization of the EDMA channels, they can be configured to continuously service on-chip and off-chip peripherals throughout the device operation. Each event available to the EDMA has its own dedicated channel, and all channels operate simultaneously. The only requirements are to use the proper channel for a particular transfer and to enable the channel event in the event enable register EDMA_TPCC_EER. When programming an EDMA channel to service a peripheral, it is necessary to know how data is to be presented to the processor. Data is always provided with some kind of synchronization event as either one element per event (non-bursting) or multiple elements per event (bursting).

11.5.4.1 Non-bursting Peripherals

Non-bursting peripherals include the on-chip multichannel audio serial port (McASP) and many external devices, such as codecs. Regardless of the peripheral, the EDMA channel configuration is the same.

The McASP transmit and receive data streams are treated independently by the EDMA. The transmit and receive data streams can have completely different counts, data sizes, and formats.

To transfer the incoming data stream to its proper location in DDR memory, the EDMA channel must be set up for a 1D-to-1D transfer with A-synchronization. Because an event (AREVT) is generated for every word as it arrives, it is necessary to have the EDMA issue the transfer request for each element individually. Figure 11-29 shows the parameters for this transfer. The source address of the EDMA channel is set to the data port address (DAT) for McASP, and the destination address is set to the start of the data block in DDR. Because the address of serializer buffer is fixed, the source B index is cleared to 0 (no modification) and the destination B index is set to 0x2 (increment).

Based on the premise that serial data is typically a high priority, the EDMA channel should be programmed to be on queue 0.

Figure 11-28 shows servicing incoming McASP data.

Figure 11-28. Servicing Incoming McASP Data Example

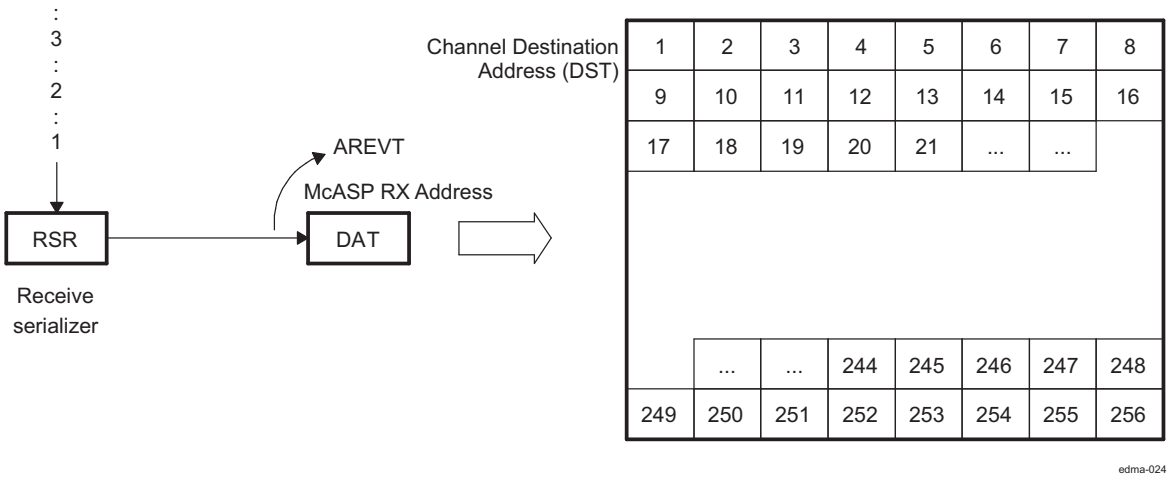


Figure 11-29. Servicing Incoming McASP Data Example PaRAM Configuration

(a) EDMA Parameters

Parameter Contents	
0010 0000h	
McASP RX Address	
0100h	0001h
Channel Destination Address (DST)	
0001h	0000h
0000h	FFFFh
0000h	0000h
0000h	0004h

Parameter	
Channel Options Parameter (OPT)	
Channel Source Address (SRC)	
Count for 2nd Dimension (BCNT)	Count for 1st Dimension (ACNT)
Channel Destination Address (DST)	
Destination BCNT Index (DBIDX)	Source BCNT Index (SBIDX)
BCNT Reload (BCNTRLD)	Link Address (LINK)
Destination CCNT Index (DCIDX)	Source CCNT Index (SCIDX)
Reserved	Count for 3rd Dimension (CCNT)

(b) Channel Options Parameter (OPT) Content

- EDMA_TPCC_OPT_n[20] TCINTEN = 0x1

11.5.4.2 Bursting Peripherals

Higher bandwidth applications require that multiple data elements be presented to the processor core for every synchronization event. This frame of data can either be from multiple sources that are working simultaneously or from a single high-throughput peripheral that streams data to/from the processor.

In this example, a port is receiving a video frame from a camera and presenting it to the DSP one array at a time. The video image is 640 × 480 pixels, with each pixel represented by a 16-bit element. The image is to be stored in external memory.

To transfer data from an external peripheral to an external buffer one array at a time based on EVT*n* , channel *n* must be configured. Due to the nature of the data (a video frame made up of arrays of pixels) the destination is essentially a 2D entity. Figure 11-31 shows the parameters to service the incoming data with a 1D-to-2D transfer using AB-synchronization. The source address is set to the location of the video framer peripheral, and the destination address is set to the start of the data buffer. Because the input address is static, the BDIX_n[15:0] SBIDX is 0 (no modification to the source address). The destination is made up of arrays of contiguous, linear elements; therefore, the EDMA_TPCC_BIDX_n[31:16] DBIDX is set to pixel size, 2 bytes. EDMA_TPCC_ABCNT_n[15:0] ANCT is equal to the pixel size, 2 bytes. EDMA_TPCC_ABCNT_n[31:16] BCNT is set to the number of pixels in an array, 640. EDMA_TPCC_CCNT_n[15:0] CCNT is equal to the total number of arrays in the block, 480. EDMA_TPCC_CIDX_n[15:0] SCIDX is 0 because the source address undergoes no increment. The EDMA_TPCC_CIDX_n[31:16] DCIDX is equal to the difference between the starting addresses of each array. Because one pixel is 16 bits (2 bytes), EDMA_TPCC_CIDX_n[31:16] DCIDX is equal to 640 × 2.

Figure 11-30 shows Bursting Peripherals Transfer.

Figure 11-30. Servicing Peripheral Burst Example

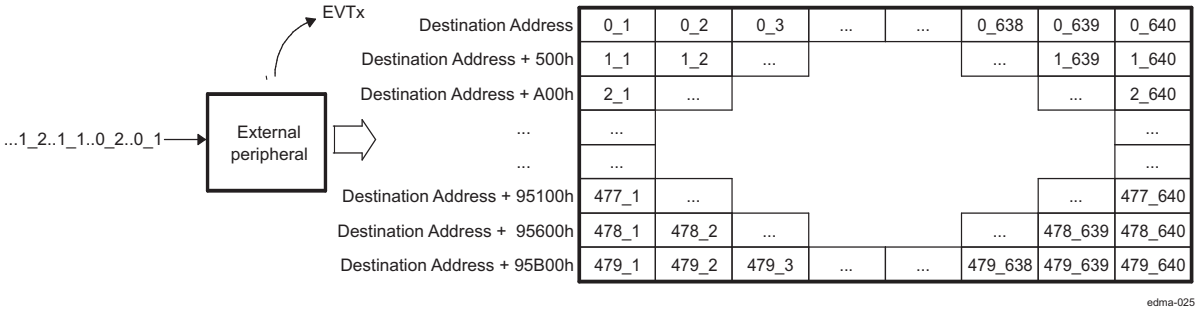


Figure 11-31. Servicing Peripheral Burst Example PaRAM Configuration

(a) EDMA Parameters

Parameter Contents	
0010 0004h	
Channel Source Address	
0280h	0002h
Channel Destination Address	
0002h	0000h
0000h	FFFFh
0500h	0000h
0000h	01E0h

Parameter	
Channel Options Parameter (OPT)	
Channel Source Address (SRC)	
Count for 2nd Dimension (BCNT)	Count for 1st Dimension (ACNT)
Channel Destination Address (DST)	
Destination BCNT Index (DSTBIDX)	Source BCNT Index (SRCBIDX)
BCNT Reload (BCNTRLD)	Link Address (LINK)
Destination CCNT Index (DSTCIDX)	Source CCNT Index (SRCCIDX)
Reserved	Count for 3rd Dimension (CCNT)

(b) Channel Options Parameter (OPT) Content

- EDMA_TPCC_OPT_n[2] SYNCDIM = 0x1
- EDMA_TPCC_OPT_n[20] TCINTEN = 0x1

11.5.4.3 Continuous Operation

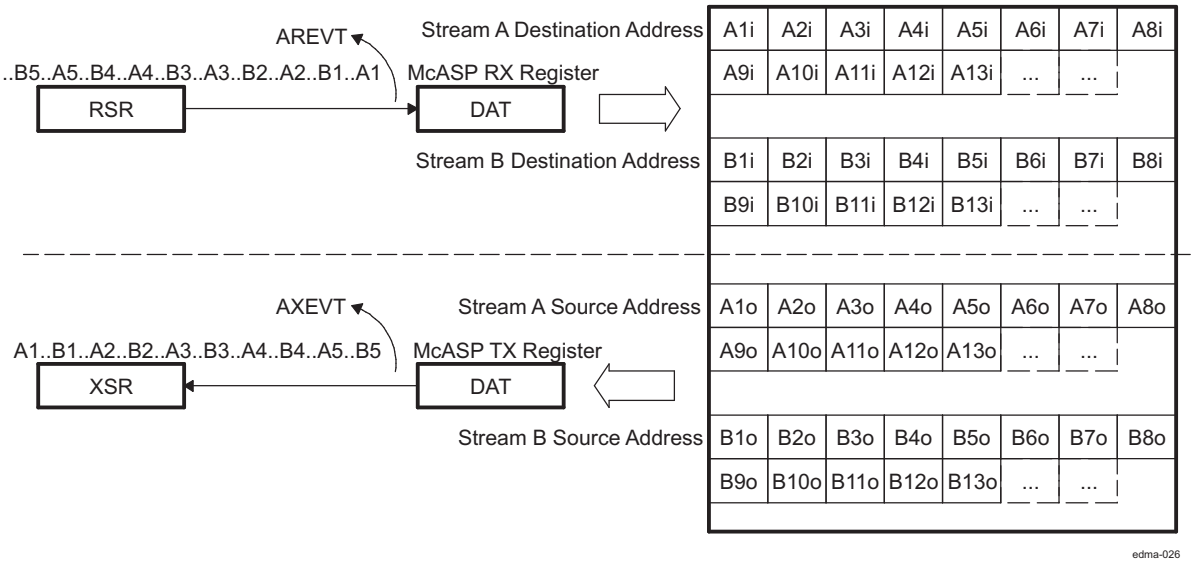
Configuring an EDMA channel to receive a single frame of data is useful, and is applicable to some systems. A majority of the time, however, data is going to be continuously transmitted and received throughout the entire operation of the processor. In this case, it is necessary to implement some form of linking such that the EDMA channels continuously reload the necessary parameter sets. In this example, McASP is configured to transmit and receive data on a T1 array. To simplify the example, only two channels are active for both transmit and receive data streams. Each channel receives packets of 128 elements. The packets are transferred from the serial port to internal memory and from internal memory to the serial port, as shown in Figure 11-32.

The McASP generates AREVT for every element received and generates AXEVT for every element transmitted. To service the data streams, them DMA channels associated with the McASP must be setup for 1D-to-1D transfers with A-synchronization.

Figure 11-33 shows the parameter entries for the channel for these transfers. To service the McASP continuously throughout DSP operation, the channels must be linked to a duplicate PaRAM set in the PaRAM. After all frames have been transferred, the EDMA channels reload and continue.

Figure 11-34 shows the reload parameters for the channel.

Figure 11-32. Servicing Continuous McASP Data Example



11.5.4.3.1 Receive Channel

EDMA channel 15 services the incoming data stream of McASP. The source address is set to that of the receive serializer buffer, and the destination address is set to the first element of the data block. Because there are two data channels being serviced, A and B, they are to be located separately within the L2 SRAM.

To facilitate continuous operation, a copy of the PaRAM set for the channel is placed in PaRAM set 64. The LINK option is set and the link address is provided in the PaRAM set. Upon exhausting the channel 15 parameter set, the parameters located at the link address are loaded into the channel 15 parameter set and operation continues. This function continues throughout device operation until halted by the CPU.

11.5.4.3.2 Transmit Channel

EDMA channel 12 services the outgoing data stream of McASP. In this case the destination address needs no update, hence, the parameter set changes accordingly. Linking is also used to allow continuous operation by the EDMA channel, with duplicate PaRAM set entries at PaRAM set 65.

Figure 11-33. Servicing Continuous McASP Data Example PaRAM Configuration

(a) EDMA Parameters for Receive Channel (PaRAM Set 15) being Linked to PaRAM Set 64

Parameter Contents	
0010 0000h	
McASP RX Register	
0080h	0001h
Channel Destination Address (DST)	
0001h	0000h
0080h	4800h
0000h	0000h
0000h	FFFFh

Parameter	
Channel Options Parameter (OPT)	
Channel Source Address (SRC)	
Count for 2nd Dimension (BCNT)	Count for 1st Dimension (ACNT)
Channel Destination Address (DST)	
Destination BCNT Index (DBIDX)	Source BCNT Index (SBIDX)
BCNT Reload (BCNTRLD)	Link Address (LINK)
Destination CCNT Index (DCIDX)	Source CCNT Index (SCIDX)
Reserved	Count for 3rd Dimension (CCNT)

(b) Channel Options Parameter (OPT) Content for Receive Channel (PaRAM Set 15)

- EDMA_TPCC_OPT_n[20] TCINTEN = 0x1

(c) EDMA Parameters for Transmit Channel (PaRAM Set 12) being Linked to PaRAM Set 65

Parameter Contents	
0010 1000h	
Channel Source Address (SRC)	
0080h	0001h
McASP TX Register	
0000h	0001h
0080h	4860h
0000h	0000h
0000h	FFFFh

Parameter	
Channel Options Parameter (OPT)	
Channel Source Address (SRC)	
Count for 2nd Dimension (BCNT)	Count for 1st Dimension (ACNT)
Channel Destination Address (DST)	
Destination BCNT Index (DBIDX)	Source BCNT Index (SBIDX)
BCNT Reload (BCNTRLD)	Link Address (LINK)
Destination CCNT Index (DCIDX)	Source CCNT Index (SCIDX)
Reserved	Count for 3rd Dimension (CCNT)

(d) Channel Options Parameter (OPT) Content for Transmit Channel (PaRAM Set 12)

- EDMA_TPCC_OPT_n[20] TCINTEN = 0x1

Figure 11-34. Servicing Continuous McASP Data Example Reload PaRAM Configuration

(a) EDMA Reload Parameters (PaRAM Set 64) for Receive Channel

Parameter Contents	
0010 0000h	
McASP RX Register	
0080h	0001h
Channel Destination Address (DST)	
0001h	0000h
0080h	4800h
0000h	0000h
0000h	FFFFh

Parameter	
Channel Options Parameter (OPT)	
Channel Source Address (SRC)	
Count for 2nd Dimension (BCNT)	Count for 1st Dimension (ACNT)
Channel Destination Address (DST)	
Destination BCNT Index (DBIDX)	Source BCNT Index (SBIDX)
BCNT Reload (BCNTRLD)	Link Address (LINK)
Destination CCNT Index (DCIDX)	Source CCNT Index (SCIDX)
Reserved	Count for 3rd Dimension (CCNT)

(b) Channel Options Parameter (OPT) Content for Receive Channel (PaRAM Set 64)

- EDMA_TPCC_OPT_n[20] TCINTEN = 0x1

(c) EDMA Reload Parameters (PaRAM Set 65) for Transmit Channel

Parameter Contents	
0010 1000h	
Channel Source Address (SRC)	
0080h	0001h
McASP TX Register	
0000h	0001h
0080h	4860h
0000h	0000h
0000h	FFFFh

Parameter	
Channel Options Parameter (OPT)	
Channel Source Address (SRC)	
Count for 2nd Dimension (BCNT)	Count for 1st Dimension (ACNT)
Channel Destination Address (DST)	
Destination BCNT Index (DBIDX)	Source BCNT Index (SBIDX)
BCNT Reload (BCNTRLD)	Link Address (LINK)
Destination CCNT Index (DCIDX)	Source CCNT Index (SCIDX)
Reserved	Count for 3rd Dimension (CCNT)

(d) Channel Options Parameter (OPT) Content for Transmit Channel (PaRAM Set 65)

- EDMA_TPCC_OPT_n[20] TCINTEN = 0x1

11.5.4.4 Ping-Pong Buffering

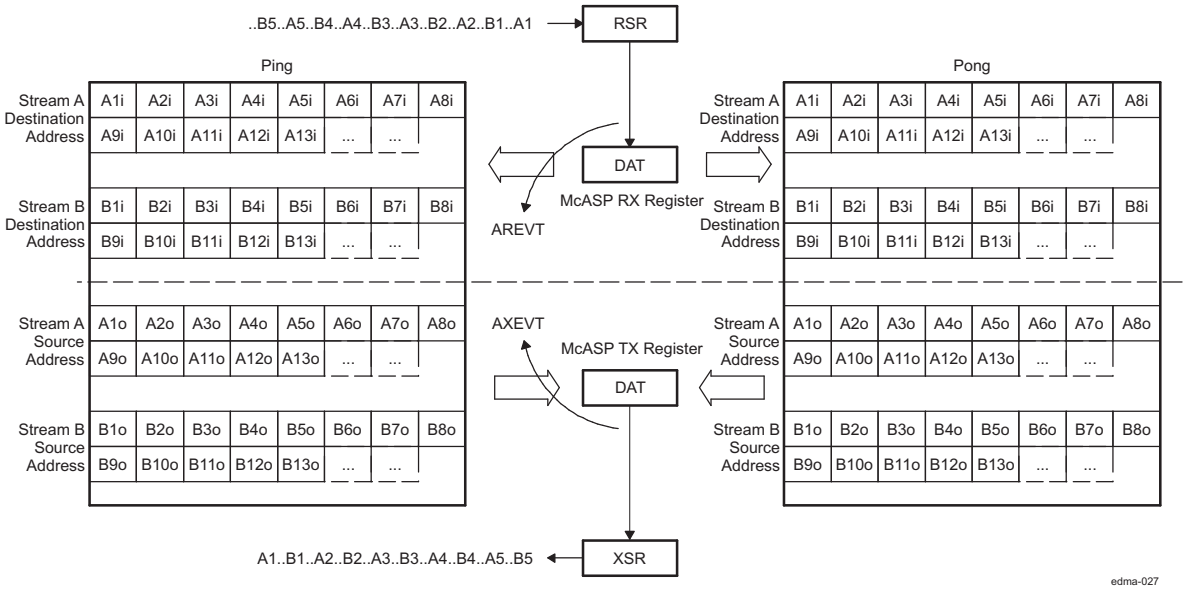
Although the previous configuration allows the EDMA to service a peripheral continuously, it presents a number of restrictions to the CPU. Because the input and output buffers are continuously being filled/emptied, the CPU must match the pace of the EDMA very closely to process the data. The EDMA receive data must always be placed in memory before the CPU accesses it, and the CPU must provide the output data before the EDMA transfers it. Though not impossible, this is an unnecessary challenge. It is particularly difficult in a 2-level cache scheme.

Ping-pong buffering is a simple technique that allows the CPU activity to be distanced from the EDMA activity. This means that there are multiple (usually two) sets of data buffers for all incoming and outgoing data streams. While the EDMA transfers the data into and out of the ping buffers, the CPU manipulates the data in the pong buffers. When both CPU and EDMA activity completes, they switch. The EDMA then writes over the old input data and transfers the new output data. Figure 11-35 shows the ping-pong scheme for this example.

To change the continuous operation example, such that a ping-pong buffering scheme is used, the EDMA channels need only a moderate change. Instead of one parameter set, there are two; one for transferring data to/from the ping buffers and one for transferring data to/from the pong buffers. As soon as one transfer completes, the channel loads the PaRAM set for the other and the data transfers continue. Figure 11-36 shows the EDMA channel configuration required.

Each channel has two parameter sets, ping and pong. The EDMA channel is initially loaded with the ping parameters (Figure 11-37). The link address for the ping set is set to the PaRAM offset of the pong parameter set (Figure 11-37). The link address for the pong set is set to the PaRAM offset of the ping parameter set (Figure 11-38). The channel options, count values, and index values are all identical between the ping and pong parameters for each channel. The only differences are the link address provided and the address of the data buffer.

Figure 11-35. Ping-Pong Buffering for McASP Data Example



11.5.4.4.1 Synchronization with the CPU

To utilize the ping-pong buffering technique, the system must signal the CPU when to begin to access the new data set. After the CPU finishes processing an input buffer (ping), it waits for the EDMA to complete before switching to the alternate (pong) buffer. In this example, both channels provide their channel numbers as their report word and set the EDMA_TPCC_OPT_n[20] TCINTEN bit to generate an interrupt after completion. When channel 15 fills an input buffer, the E15 bit in the interrupt pending register

EDMA_TPCC_IPR is set; when channel 12 empties an output buffer, the E12 bit in EDMA_TPCC_IPR is set. The CPU must manually clear these bits. With the channel parameters set, the CPU polls EDMA_TPCC_IPR to determine when to switch. The EDMA and CPU could alternatively be configured such that the channel completion interrupts the CPU. By doing this, the CPU could service a background task while waiting for the EDMA to complete.

Figure 11-36. Ping-Pong Buffering for McASP Example PaRAM Configuration

(a) EDMA Parameters for Channel 15 (Using PaRAM Set 15 Linked to Pong Set 64)

Parameter Contents	
0010 D000h	
McASP RX Register	
0080h	0001h
Channel Destination Address (DST)	
0001h	0000h
0080h	4800h
0000h	0000h
0000h	0001h

Parameter	
Channel Options Parameter (OPT)	
Channel Source Address (SRC)	
Count for 2nd Dimension (BCNT)	Count for 1st Dimension (ACNT)
Channel Destination Address (DST)	
Destination BCNT Index (DBIDX)	Source BCNT Index (SBIDX)
BCNT Reload (BCNTRLD)	Link Address (LINK)
Destination CCNT Index (DCIDX)	Source CCNT Index (SCIDX)
Reserved	Count for 3rd Dimension (CCNT)

(b) Channel Options Parameter (OPT) Content for Channel 15

- EDMA_TPCC_OPT_n[20] TCINTEN = 0x1

(c) EDMA Parameters for Channel 12 (Using PaRAM Set 12 Linked to Pong Set 65)

Parameter Contents	
0010 C000h	
Channel Source Address (SRC)	
0080h	0001h
McASP TX Register	
0000h	0001h
0080h	4840h
0000h	0000h
0000h	0001h

Parameter	
Channel Options Parameter (OPT)	
Channel Source Address (SRC)	
Count for 2nd Dimension (BCNT)	Count for 1st Dimension (ACNT)
Channel Destination Address (DST)	
Destination BCNT Index (DBIDX)	Source BCNT Index (SBIDX)
BCNT Reload (BCNTRLD)	Link Address (LINK)
Destination CCNT Index (DCIDX)	Source CCNT Index (SCIDX)
Reserved	Count for 3rd Dimension (CCNT)

(d) Channel Options Parameter (OPT) Content for Channel 12

- EDMA_TPCC_OPT_n[20] TCINTEN = 0x1

Figure 11-37. Ping-Pong Buffering for McASP Example Pong PaRAM Configuration

(a) EDMA Pong Parameters for Channel 15 at Set 64 Linked to Set 65

Parameter Contents	
0010 D000h	
McASP RX Register	
0080h	0001h
Channel Destination Address (DST)	
0001h	0000h
0080h	4820h
0000h	0000h
0000h	0001h

Parameter	
Channel Options Parameter (OPT)	
Channel Source Address (SRC)	
Count for 2nd Dimension (BCNT)	Count for 1st Dimension (ACNT)
Channel Destination Address (DST)	
Destination BCNT Index (DBIDX)	Source BCNT Index (SBIDX)
BCNT Reload (BCNTRLD)	Link Address (LINK)
Destination CCNT Index (DCIDX)	Source CCNT Index (SCIDX)
Reserved	Count for 3rd Dimension (CCNT)

(b) EDMA Pong Parameters for Channel 12 at Set 66 Linked to Set 67

Parameter Contents	
0010 C000h	
Channel Source Address (SRC)	
0080h	0001h
McASP TX Register	
0000h	0001h
0080h	4860h
0000h	0000h
0000h	0001h

Parameter	
Channel Options Parameter (OPT)	
Channel Source Address (SRC)	
Count for 2nd Dimension (BCNT)	Count for 1st Dimension (ACNT)
Channel Destination Address (DST)	
Destination BCNT Index (DBIDX)	Source BCNT Index (SBIDX)
BCNT Reload (BCNTRLD)	Link Address (LINK)
Destination CCNT Index (DCIDX)	Source CCNT Index (SCIDX)
Reserved	Count for 3rd Dimension (CCNT)

Figure 11-38. Ping-Pong Buffering for McASP Example Ping PaRAM Configuration

(a) EDMA Ping Parameters for Channel 15 at Set 65 Linked to Set 64

Parameter Contents	
0010 D000h	
McASP RX Register	
0080h	0001h
Channel Destination Address (DST)	
0001h	0000h
0080h	4800h
0000h	0000h
0000h	0001h

Parameter	
Channel Options Parameter (OPT)	
Channel Source Address (SRC)	
Count for 2nd Dimension (BCNT)	Count for 1st Dimension (ACNT)
Channel Destination Address (DST)	
Destination BCNT Index (DBIDX)	Source BCNT Index (SBIDX)
BCNT Reload (BCNTRLD)	Link Address (LINK)
Destination CCNT Index (DCIDX)	Source CCNT Index (SCIDX)
Reserved	Count for 3rd Dimension (CCNT)

(b) EDMA Ping Parameters for Channel 12 at Set 67 Linked to Set 66

Parameter Contents	
0010 C000h	
Channel Source Address (SRC)	
0080h	0001h
McASP TX Register	
0000h	0001h
0080h	4840h
0000h	0000h
0000h	0001h

Parameter	
Channel Options Parameter (OPT)	
Channel Source Address (SRC)	
Count for 2nd Dimension (BCNT)	Count for 1st Dimension (ACNT)
Channel Destination Address (DST)	
Destination BCNT Index (DBIDX)	Source BCNT Index (SBIDX)
BCNT Reload (BCNTRLD)	Link Address (LINK)
Destination CCNT Index (DCIDX)	Source CCNT Index (SCIDX)
Reserved	Count for 3rd Dimension (CCNT)

11.5.4.5 Transfer Chaining Examples

The following examples explain the intermediate transfer complete chaining function.

11.5.4.5.1 Servicing Input/Output FIFOs with a Single Event

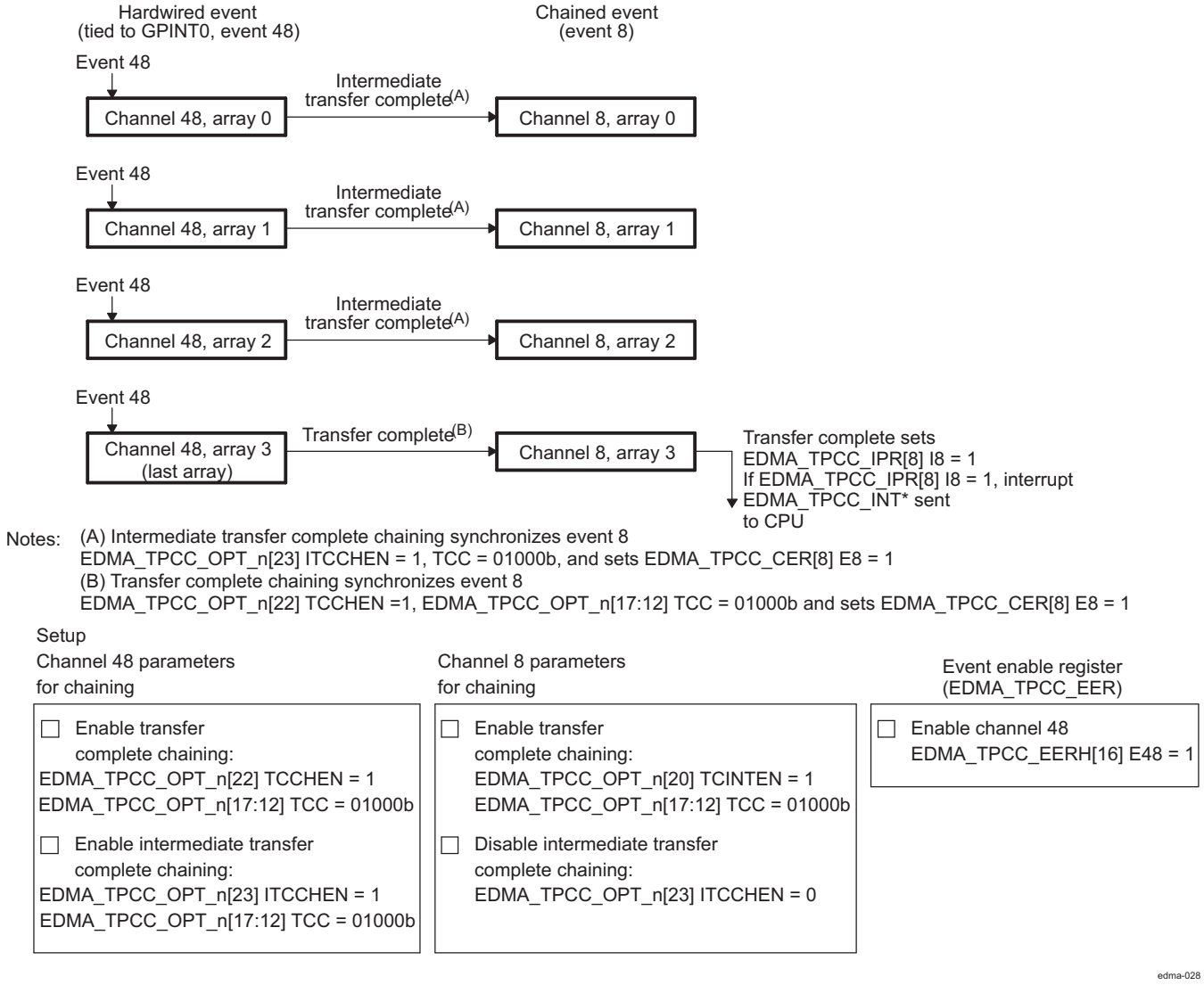
Many systems require the use of a pair of external FIFOs that must be serviced at the same rate. One FIFO buffers data input, and the other buffers data output. The EDMA channels that service these FIFOs can be set up for AB-synchronized transfers. While each FIFO is serviced with a different set of parameters, both can be signaled from a single event.

For example, an external interrupt pin can be tied to the status flags of one of the FIFOs. When this event arrives, the EDMA needs to perform servicing for both the input and output streams. Without the intermediate transfer complete chaining feature this would require two events, and thus two external interrupt pins. The intermediate transfer complete chaining feature allows the use of a single external event (for example, a GPIO event). Figure 11-39 shows the EDMA setup and illustration for this example.

A GPIO event (in this case, GPINT0) triggers an array transfer. Upon completion of each intermediate array transfer of channel 48, intermediate transfer complete chaining sets the E8 bit (specified by TCC of 8) in the chained event register EDMA_TPCC_CER and provides a synchronization event to channel 8. Upon completion of the last array transfer of channel 48, transfer complete chaining—not intermediate transfer complete chaining—sets the E8 bit in EDMA_TPCC_CER (specified by EDMA_TPCC_OPT_n[11] TCCMODE: TCC) and provides a synchronization event to channel 8. The completion of channel 8 sets the I8 bit (specified by EDMA_TPCC_OPT_n[11] TCCMODE: TCC) in the interrupt pending register EDMA_TPCC_IPR, which can generate an interrupt to the CPU, if the I8 bit in the interrupt enable register EDMA_TPCC_IER is set.

Figure 11-39 shows the Intermediate Transfer Completion Chaining Example.

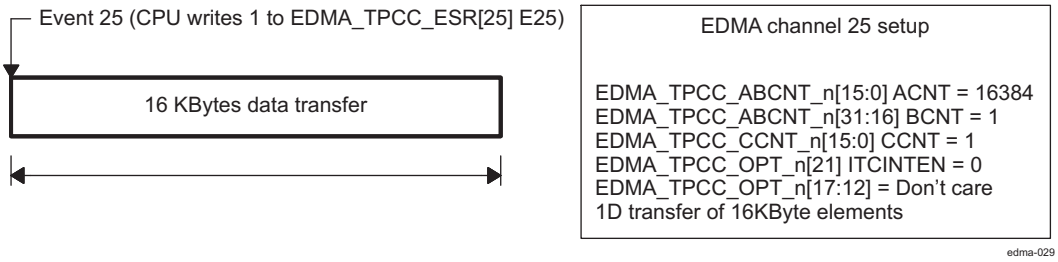
Figure 11-39. Intermediate Transfer Completion Chaining Example



11.5.4.5.2 Breaking Up Large Transfers with Intermediate Chaining

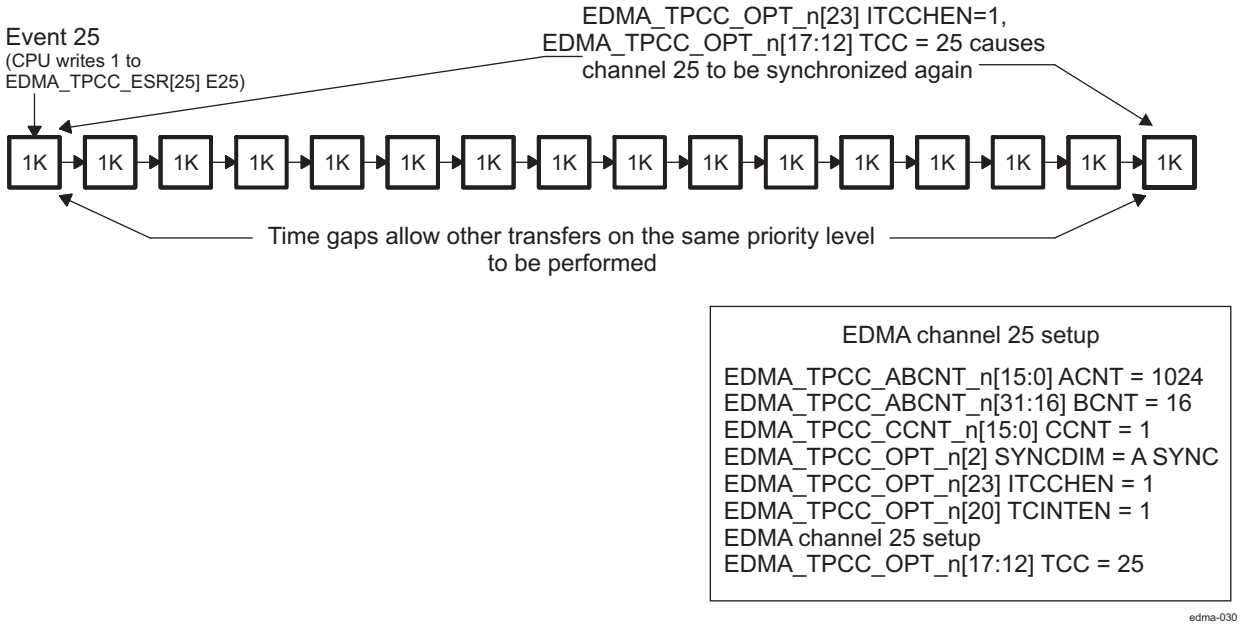
Another feature of intermediate transfer chaining EDMA_TPCC_OPT_n[23] ITCCHEN is for breaking up large transfers. A large transfer may lock out other transfers of the same priority level for the duration of the transfer. For example, a large transfer on queue 0 from the internal memory to the external memory using the EMIF may starve other EDMA transfers on the same queue. In addition, this large high-priority transfer may prevent the EMIF for a long duration to service other lower priority transfers. When a large transfer is considered to be high priority, it should be split into multiple smaller transfers. Figure 11-40 shows the EDMA setup and illustration of an example single large block transfer.

Figure 11-40. Single Large Block Transfer Example



The intermediate transfer chaining enable EDMA_TPCC_OPT_n[23] ITCCHEN provides a method to break up a large transfer into smaller transfers. For example, to move a single large block of memory (16K bytes), the EDMA performs an A-synchronized transfer. The element count is set to a reasonable value, where reasonable derives from the amount of time it would take to move this smaller amount of data. Assume 1 Kbyte is a reasonable small transfer in this example. The EDMA is set up to transfer 16 arrays of 1 Kbyte elements, for a total of 16K byte elements. The EDMA_TPCC_OPT_n[17:12] TCC field in the channel options parameter (OPT) is set to the same value as the channel number and EDMA_TPCC_OPT_n[23] ITCCHEN are set. In this example, EDMA channel 25 is used and EDMA_TPCC_OPT_n[17:12] TCC is also set to 25. The EDMA_TPCC_OPT_n[20] TCINTEN may also be set to trigger interrupt 25 when the last 1 Kbyte array is transferred. The CPU starts the EDMA transfer by writing to the appropriate bit of the event set register EDMA_TPCC_ESR[25] E25. The EDMA transfers the first 1 Kbyte array. Upon completion of the first array, intermediate transfer complete code chaining generates a synchronization event to channel 25, a value specified by the EDMA_TPCC_OPT_n[17:12] TCC field. This intermediate transfer completion chaining event causes EDMA channel 25 to transfer the next 1 Kbyte array. This process continues until the transfer parameters are exhausted, at which point the EDMA has completed the 16K byte transfer. This method breaks up a large transfer into smaller packets, thus providing natural time slices in the transfer such that other events may be processed. Figure 11-41 shows the EDMA setup and illustration of the broken up smaller packet transfers.

Figure 11-41. Smaller Packet Data Transfers Example



11.5.5 Setting Up an EDMA Transfer

The following list provides a quick guide for the typical steps involved in setting up a transfer.

Step 1. Initiating a DMA/QDMA channel

1. Determine the type of channel (QDMA or DMA) to be used.
2. Channel mapping
   i. If using a QDMA channel, program the EDMA_TPCC_QCHMAPN_j with the parameter set number to which the channel maps and the trigger word.
   ii. If using a DMA channel, program the EDMA_TPCC_DCHMAPN_m with the parameter set number to which the channel maps.
3. If the channel is being used in the context of a shadow region, ensure the EDMA_TPCC_DRAEM_k / EDMA_TPCC_DRAEHM_k for the region is properly set up to allow read write accesses to bits in the event registers and interrupt registers in the Shadow region memory map. The subsequent steps in this process should be done using the respective shadow region registers. (Shadow region descriptions and usage are provided in Section 11.4.7.1.)
4. Determine the type of triggering used.
   i. If external events are used for triggering (DMA channels), enable the respective event in EDMA_TPCC_EER / EDMA_TPCC_EERH by writing into EDMA_TPCC_EESR / EDMA_TPCC_EESRH.
   ii. If QDMA Channel is used, enable the channel in EDMA_TPCC_QEER by writing into EDMA_TPCC_QEESR.
5. Queue setup
   i. If a QDMA channel is used, set up the EDMA_TPCC_QDMAQNUM to map the channel to the respective event queue.
   ii. If a DMA channel is used, set up the EDMA_TPCC_DMAQNUMN_k to map the event to the respective event queue.

Step 2. Parameter set setup

1. Program the PaRAM set number associated with the channel. Note that

NOTE: If it is a QDMA channel, the PaRAM entry that is configured as trigger word is written to last. Alternatively, enable the QDMA channel (step 1-b-ii above) just before the write to the trigger word.

Step 3. Interrupt setup

1. Enable the interrupt in the EDMA_TPCC_IER / EDMA_TPCC_IERH by writing into EDMA_TPCC_IESR / EDMA_TPCC_IESRH.
2. Ensure that the EDMA_TPCC completion interrupt (either the global or the shadow region interrupt) is enabled properly in the device interrupt controller.
3. Ensure the EDMA_TPCC completion interrupt (this refers to either the Global interrupt or the shadow region interrupt) is enabled properly in the Device Interrupt controller.
4. Set up the interrupt controller properly to receive the expected EDMA interrupt.

Step 4. Initiate transfer

1. This step is highly dependent on the event trigger source:
   i. If the source is an external event coming from a peripheral, the peripheral will be enabled to start generating relevant EDMA events that can be latched to the EDMA_TPCC_ER transfer.
   ii. For QDMA events, writes to the trigger word (step 2-a above) will initiate the transfer.
   iii. Manually triggered transfers will be initiated by writes to the Event Set Registers EDMA_TPCC_ESR / EDMA_TPCC_ESRH.
   iv. Chained-trigger events initiate when a previous transfer returns a transfer completion code equal to the chained channel number.

Step 5. Wait for completion

1. If the interrupts are enabled as mentioned in step 3 above, then the EDMA_TPCC will generate a completion interrupt to the CPU whenever transfer completion results in setting the corresponding bits in the interrupt pending register EDMA_TPCC_IPR / EDMA_TPCC_IPRH. The set bits must be cleared in the EDMA_TPCC_IPR / EDMA_TPCC_IPRH by writing to corresponding bit in EDMA_TPCC_ICR / EDMA_TPCC_ICRH.
2. If polling for completion (interrupts not enabled in the device controller), then the application code can wait on the expected bits to be set in the EDMA_TPCC_IPR / EDMA_TPCC_IPRH. Again, the set bits in the EDMA_TPCC_IPR / EDMA_TPCC_IPRH must be manually cleared via EDMA_TPCC_ICR / EDMA_TPCC_ICRH before the next set of transfers is performed for the same transfer completion code values.

11.6 EDMA Debug Checklist and Programming Tips

This section lists some tips to keep in mind while debugging applications using the EDMA controller.

11.6.1 EDMA Debug Checklist

Table 11-26 provides some common issues and their probable causes and resolutions.

Table 11-26. Debug Checklist

Issue	Description/Solution
The transfer associated with the channel does not happen. The channel does not get serviced.	The EDMA_TPCC may not service a transfer request, even though the associated PaRAM set is programmed appropriately. Check for the following: 1) Verify that events are enabled, i.e., if an external/peripheral event is latched in Event Registers EDMA_TPCC_ER / EDMA_TPCC_ERH, check that the event is enabled in the Event Enable Registers EDMA_TPCC_EER / EDMA_TPCC_EERH. Similarly, for QDMA channels, check that QDMA events are appropriately enabled in the QDMA Event Enable Register EDMA_TPCC_QEER. 2) Verify that the DMA or QDMA Secondary Event Register EDMA_TPCC_SER / EDMA_TPCC_SERH / EDMA_TPCC_QSER bits corresponding to the particular event or channel are not set.
The Secondary Event Registers bits are set, not allowing additional transfers to occur on a channel.	It is possible that a trigger event was received when the parameter set associated with the channel/event was a NULL set for a previous transfer on the channel. This is typical in two cases: 1) QDMA channels: Typically if the parameter set is non-static and expected to be terminated by a NULL set (i.e., EDMA_TPCC_OPT_n[3] STATIC = 0x0, EDMA_TPCC_LNK_n[15:0] LINK = 0xFFFF), the parameter set is updated with a NULL set after submission of the last TR. Because QDMA channels are auto-triggered, this update caused the generation of an event. An event generated for a NULL set causes an error condition and results in setting the bits corresponding to the QDMA channel in the EDMA_TPCC_QEMR and EDMA_TPCC_QSER. This will disable further prioritization of the channel. 2) DMA channels used in a continuous mode: The peripheral may be set up to continuously generate infinite events (for instance, in case of McASP, every time the data shifts out from the DXR register, it generates an XEVT). The parameter set may be programmed to expect only a finite number of events and to be terminated by a NULL link. After the expected number of events, the parameter set is reloaded with a NULL parameter set. Because the peripheral will generate additional events, an error condition is set in the EDMA_TPCC_SER[31:0] E*n* and EDMA_TPCC_EMR[31:0] E*n* set, preventing further event prioritization. Check the number of events received is limited to the expected number of events for which the parameter set is programmed, or check the bits corresponding to particular channel or event are not set in the Secondary event registers (EDMA_TPCC_SER / EDMA_TPCC_SERH / EDMA_TPCC_QSER) and Event Missed Registers (EDMA_TPCC_EMR / EDMA_TPCC_EMRH / EDMA_TPCC_QEMR) before trying to perform subsequent transfers for the event/channel.
Completion interrupts are not asserted, or no further interrupts are received after the first completion interrupt.	Check the following: 1) The interrupt generation is enabled in the EDMA_TPCC_OPT_n of the associated PaRAM set (EDMA_TPCC_OPT_n[20] TCINTEN = 0x1 and/or EDMA_TPCC_OPT_n[20] ITCINTEN = 0x1). 2) The interrupts are enabled in the EDMA Channel Controller, via the Interrupt Enable Registers (EDMA_TPCC_IER / EDMA_TPCC_IERH). 3) The corresponding interrupts are enabled in the device interrupt controller. 4) The set interrupts are cleared in the interrupt pending registers (EDMA_TPCC_IPR / EDMA_TPCC_IPRH) before exiting the transfer completion interrupt service routine (ISR). See Section 11.4.9.1.2 *Clearing Transfer Complerion Interrupts* for details on writing EDMA ISRs. 5) If working with shadow region interrupts, make sure that the DMA Region Access registers (EDMA_TPCC_DRAEM_k / EDMA_TPCC_DRAEHM_k) are set up properly, because the EDMA_TPCC_DRAEM_k / EDMA_TPCC_DRAEHM_k registers act as secondary enables for shadow region completion interrupts, along with the EDMA_TPCC_IER / EDMA_TPCC_IERH registers. If working with shadow region interrupts, make sure that the bits corresponding to the transfer completion code EDMA_TPCC_OPT_n[17:12] TCC value are also enabled in the EDMA_TPCC_DRAEM_k / EDMA_TPCC_DRAEHM_k registers. For instance, if the PaRAM set associated with Channel 0 returns a completion code of 63 EDMA_TPCC_OPT_n[17:12] TCC = 63, ensure that EDMA_TPCC_DRAEHM_k[31] E63 is also set for a shadow region completion interrupt because the interrupt pending register bit set will be EDMA_TPCC_IPRH[31] I63 (not EDMA_TPCC_IPR[0] I0).

11.6.2 EDMA Programming Tips

1. For several registers, the setting and clearing of bits needs to be done via separate dedicated registers.

   For example, the Event Register (EDMA_TPCC_ER / EDMA_TPCC_ERH) can only be cleared by writing a 1 to the corresponding bits in the Event Clear Registers (EDMA_TPCC_ECR / EDMA_TPCC_ECRH). Similarly, the Event Enable Register (EDMA_TPCC_EER / EDMA_TPCC_EERH) bits can only be set with writing of 0x1 to the Event Enable Set Registers (EDMA_TPCC_EESR / EDMA_TPCC_EESRH) and cleared with writing of 0x1 to the corresponding bits in the Event Enable Clear Register (EDMA_TPCC_EECR / EDMA_TPCC_EECRH).
2. Writes to the shadow region memory maps are governed by region access registers (EDMA_TPCC_DRAE / EDMA_TPCC_DRAEHM_k / EDMA_TPCC_QRAEN_k). If the appropriate channels are not enabled in these registers, read/write access to the shadow region memory map is not enabled.
3. When working with shadow region completion interrupts, ensure that the DMA Region Access Registers (EDMA_TPCC_DRAEM_k / EDMA_TPCC_DRAEHM_k) for every region are set in a mutually exclusive way (unless it is a requirement for an application). If there is an overlap in the allocated channels and transfer completion codes (setting of Interrupt Pending Register bits) in the region resource allocation, it results in multiple shadow region completion interrupts.

   For example, if EDMA_TPCC_DRAEM_k.DRAEM_0[0] E0 and EDMA_TPCC_DRAEM_k.DRAEM_1[0] E0 are both set, then on completion of a transfer that returns a TCC = 0x0, they will generate both shadow region 0 and 1 completion interrupts.
4. While programming a non-dummy parameter set, ensure the EDMA_TPCC_CCNT_n[15:0] CCNT is not left to zero.
5. Enable the EDMA_TPCC error interrupt in the device controller and attach an interrupt service routine (ISR) to ensure that error conditions are not missed in an application and are appropriately addressed with the ISR.
6. Depending on the application, it can want to break large transfers into smaller transfers and use self-chaining to prevent starvation of other events in an event queue.
7. In applications where a large transfer is broken into sets of small transfers using chaining or other methods, it chooses to use the early chaining option to reduce the time between the sets of transfers and increase the throughput.

   However, keep in mind that with early completion, all data might have not been received at the end point when completion is reported because the EDMA_TPCC internally signals completion when the TR is submitted to the EDMA_TPTC, potentially before any data has been transferred.
8. The event queue entries can be observed to determine the last few events if there is a system failure (provided the entries were not bypassed).

11.7 EDMA Register Manual

11.7.1 EDMA Instance Summary

Table 11-27 shows the L3_MAIN base address and address space for the EDMA module instances.

Table 11-27. EDMA Instance Summary

Module Name	Base Address	Size
SYS_EDMA_TPCC	0x4330 0000	1 MB
SYS_EDMA_TPTC0	0x4340 0000	1 MB
SYS_EDMA_TPTC1	0x4350 0000	1 MB
DSP1_EDMA_TPCC	0x40D1 0000	32 KB
DSP2_EDMA_TPCC	0x4151 0000	32 KB
DSP1_EDMA_TPTC0	0x40D0 5000	4 KB
DSP2_EDMA_TPTC0	0x4150 5000	4 KB
DSP1_EDMA_TPTC1	0x40D0 6000	4 KB
DSP2_EDMA_TPTC1	0x4150 6000	4 KB
EVE_EDMA_TPCC	0x420A 0000	32 KB
EVE_EDMA_TPTC0	0x4208 6000	4 KB
EVE_EDMA_TPTC1	0x4208 7000	4 KB

Table 11-28 lists the base addresses for DSP internal (private) access to its embedded TPCC / TPTC modules.

Table 11-28. DSP EDMA Instance Summary (Private Access)

Module Name	Base Address	Size
DSP_EDMA_TPCC	0x01D1 0000	32 KB
DSP_EDMA_TPTC0	0x01D0 5000	4 KB
DSP_EDMA_TPTC1	0x01D0 6000	4 KB

Table 11-29 lists the base addresses for EVE internal (private) access to its embedded TPCC / TPTC modules.

Table 11-29. EVE EDMA Instance Summary (Private Access)

Module Name	Base Address (EVE Private Access)	Size
EVE_EDMA_TPCC	0x400A 0000	32 KB
EVE_EDMA_TPTC0	0x4008 6000	4 KB
EVE_EDMA_TPTC1	0x4008 7000	4 KB

11.7.2 EDMA Registers

11.7.2.1 EDMA Register Summary

Table 11-30 through Table 11-39 summarize the EDMA_TPCC, EDMA_TPTC0 and EDMA_TPTC1 registers.

NOTE: All EDMA modules in the SoC are functionally identical. Note that some of the configuration parameters may be different for the various EDMA instances (see Section 11.1.2, *EDMA Controllers Configuration*).

Table 11-30. System EDMA_TPCC Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	SYS_EDMA_TPCC Physical Address (L3_MAIN Access)
EDMA_TPCC_PID	R	32	0x0000 0000	0x4330 0000
EDMA_TPCC_CCCFG	R	32	0x0000 0004	0x4330 0004
EDMA_TPCC_CLKGDIS	RW	32	0x0000 00FC	0x4330 00FC
EDMA_TPCC_DCHMAPN_m (1)	RW	32	0x0000 0100 + (0x4 * m)	0x4330 0100 + (0x4 * m)
EDMA_TPCC_QCHMAPN_j (2)	RW	32	0x0000 0200 + (0x4 * j)	0x4330 0200 + (0x4 * j)
EDMA_TPCC_DMAQNUMN_k (3)	RW	32	0x0000 0240 + (0x4 * k)	0x4330 0240 + (0x4 * k)
EDMA_TPCC_QDMAQNUM	RW	32	0x0000 0260	0x4330 0260
EDMA_TPCC_QUETCMAP	RW	32	0x0000 0280	0x4330 0280
EDMA_TPCC_QUEPRI	RW	32	0x0000 0284	0x4330 0284
EDMA_TPCC_EMR	R	32	0x0000 0300	0x4330 0300
EDMA_TPCC_EMRH	R	32	0x0000 0304	0x4330 0304
EDMA_TPCC_EMCR	W	32	0x0000 0308	0x4330 0308
EDMA_TPCC_EMCRH	W	32	0x0000 030C	0x4330 030C
EDMA_TPCC_QEMR	R	32	0x0000 0310	0x4330 0310
EDMA_TPCC_QEMCR	W	32	0x0000 0314	0x4330 0314
EDMA_TPCC_CCERR	R	32	0x0000 0318	0x4330 0318
EDMA_TPCC_CCERRCLR	W	32	0x0000 031C	0x4330 031C
EDMA_TPCC_EEVAL	W	32	0x0000 0320	0x4330 0320
EDMA_TPCC_DRAEM_k (3)	RW	32	0x0000 0340 + (0x8 * k)	0x4330 0340 + (0x8 * k)
EDMA_TPCC_DRAEHM_k (3)	RW	32	0x0000 0344 + (0x8 * k)	0x4330 0344 + (0x8 * k)
EDMA_TPCC_QRAEN_k (3)	RW	32	0x0000 0380 + (0x4 * k)	0x4330 0380 + (0x4 * k)
EDMA_TPCC_Q0E_p (4)	R	32	0x0000 0400 + (0x4 *pl)	0x4330 0400 + (0x4 * p)
EDMA_TPCC_Q1E_p (4)	R	32	0x0000 0440 + (0x4 * p)	0x4330 0440 + (0x4 * p)
EDMA_TPCC_QSTATN_i (5)	R	32	0x0000 0600 + (0x4 * i)	0x4330 0600 + (0x4 * i)
EDMA_TPCC_QWMTHRA	RW	32	0x0000 0620	0x4330 0620
EDMA_TPCC_QWMTHRB	RW	32	0x0000 0624	0x4330 0624
EDMA_TPCC_CCSTAT	R	32	0x0000 0640	0x4330 0640
EDMA_TPCC_AETCTL	RW	32	0x0000 0700	0x4330 0700
EDMA_TPCC_AETSTAT	R	32	0x0000 0704	0x4330 0704
EDMA_TPCC_AETCMD	W	32	0x0000 0708	0x4330 0708
EDMA_TPCC_MPFAR	R	32	0x0000 0800	0x4330 0800
EDMA_TPCC_MPFSR	R	32	0x0000 0804	0x4330 0804
EDMA_TPCC_MPFCR	W	32	0x0000 0808	0x4330 0808
EDMA_TPCC_MPPAG	RW	32	0x0000 080C	0x4330 080C
EDMA_TPCC_MPPAN_k (3)	RW	32	0x0000 0810 + (0x4 * k)	0x4330 0810 + (0x4 * k)
EDMA_TPCC_ER	R	32	0x0000 1000	0x4330 1000
EDMA_TPCC_ERH	R	32	0x0000 1004	0x4330 1004
EDMA_TPCC_ECR	W	32	0x0000 1008	0x4330 1008
EDMA_TPCC_ECRH	W	32	0x0000 100C	0x4330 100C
EDMA_TPCC_ESR	W	32	0x0000 1010	0x4330 1010
EDMA_TPCC_ESRH	W	32	0x0000 1014	0x4330 1014
EDMA_TPCC_CER	R	32	0x0000 1018	0x4330 1018

(1) m = 0 to 63 for SYS_EDMA_TPCC
(2) j = 0 to 7 for SYS_EDMA_TPCC
(3) k = 0 to 7 for SYS_EDMA_TPCC
(4) p = 0 to 15 for SYS_EDMA_TPCC
(5) i = 0 to 1 for SYS_EDMA_TPCC

Table 11-30. System EDMA_TPCC Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	SYS_EDMA_TPCC Physical Address (L3_MAIN Access)
EDMA_TPCC_CERH	R	32	0x0000 101C	0x4330 101C
EDMA_TPCC_EER	R	32	0x0000 1020	0x4330 1020
EDMA_TPCC_EERH	R	32	0x0000 1024	0x4330 1024
EDMA_TPCC_EECR	W	32	0x0000 1028	0x4330 1028
EDMA_TPCC_EECRH	W	32	0x0000 102C	0x4330 102C
EDMA_TPCC_EESR	W	32	0x0000 1030	0x4330 1030
EDMA_TPCC_EESRH	W	32	0x0000 1034	0x4330 1034
EDMA_TPCC_SER	R	32	0x0000 1038	0x4330 1038
EDMA_TPCC_SERH	R	32	0x0000 103C	0x4330 103C
EDMA_TPCC_SECR	W	32	0x0000 1040	0x4330 1040
EDMA_TPCC_SECRH	W	32	0x0000 1044	0x4330 1044
EDMA_TPCC_IER	R	32	0x0000 1050	0x4330 1050
EDMA_TPCC_IERH	R	32	0x0000 1054	0x4330 1054
EDMA_TPCC_IECR	W	32	0x0000 1058	0x4330 1058
EDMA_TPCC_IECRH	W	32	0x0000 105C	0x4330 105C
EDMA_TPCC_IESR	W	32	0x0000 1060	0x4330 1060
EDMA_TPCC_IESRH	W	32	0x0000 1064	0x4330 1064
EDMA_TPCC_IPR	R	32	0x0000 1068	0x4330 1068
EDMA_TPCC_IPRH	R	32	0x0000 106C	0x4330 106C
EDMA_TPCC_ICR	W	32	0x0000 1070	0x4330 1070
EDMA_TPCC_ICRH	W	32	0x0000 1074	0x4330 1074
EDMA_TPCC_IEVAL	W	32	0x0000 1078	0x4330 1078
EDMA_TPCC_QER	R	32	0x0000 1080	0x4330 1080
EDMA_TPCC_QEER	R	32	0x0000 1084	0x4330 1084
EDMA_TPCC_QEECR	W	32	0x0000 1088	0x4330 1088
EDMA_TPCC_QEESR	W	32	0x0000 108C	0x4330 108C
EDMA_TPCC_QSER	R	32	0x0000 1090	0x4330 1090
EDMA_TPCC_QSECR	W	32	0x0000 1094	0x4330 1094
EDMA_TPCC_ER_RN_k (3)	R	32	0x0000 2000 + (0x200 * k)	0x4330 2000 + (0x200 * k)
EDMA_TPCC_ERH_RN_k (3)	R	32	0x0000 2004 + (0x200 * k)	0x4330 2004 + (0x200 * k)
EDMA_TPCC_ECR_RN_k (3)	W	32	0x0000 2008 + (0x200 * k)	0x4330 2008 + (0x200 * k)
EDMA_TPCC_ECRH_RN_k (3)	W	32	0x0000 200C + (0x200 * k)	0x4330 200C + (0x200 * k)
EDMA_TPCC_ESR_RN_k (3)	W	32	0x0000 2010 + (0x200 * k)	0x4330 2010 + (0x200 * k)
EDMA_TPCC_ESRH_RN_k (3)	W	32	0x0000 2014 + (0x200 * k)	0x4330 2014 + (0x200 * k)
EDMA_TPCC_CER_RN_k (3)	R	32	0x0000 2018 + (0x200 * k)	0x4330 2018 + (0x200 * k)
EDMA_TPCC_CERH_RN_k (3)	R	32	0x0000 201C + (0x200 * k)	0x4330 201C + (0x200 * k)
EDMA_TPCC_EER_RN_k (3)	R	32	0x0000 2020 + (0x200 * k)	0x4330 2020 + (0x200 * k)
EDMA_TPCC_EERH_RN_k (3)	R	32	0x0000 2024 + (0x200 * k)	0x4330 2024 + (0x200 * k)
EDMA_TPCC_EECR_RN_k (3)	W	32	0x0000 2028 + (0x200 * k)	0x4330 2028 + (0x200 * k)
EDMA_TPCC_EECRH_RN_k (3)	W	32	0x0000 202C + (0x200 * k)	0x4330 202C + (0x200 * k)
EDMA_TPCC_EESR_RN_k (3)	W	32	0x0000 2030 + (0x200 * k)	0x4330 2030 + (0x200 * k)
EDMA_TPCC_EESRH_RN_k (3)	W	32	0x0000 2034 + (0x200 * k)	0x4330 2034 + (0x200 * k)
EDMA_TPCC_SER_RN_k (3)	R	32	0x0000 2038 + (0x200 * k)	0x4330 2038 + (0x200 * k)
EDMA_TPCC_SERH_RN_k (3)	R	32	0x0000 203C + (0x200 * k)	0x4330 203C + (0x200 * k)
EDMA_TPCC_SECR_RN_k (3)	W	32	0x0000 2040 + (0x200 * k)	0x4330 2040 + (0x200 * k)
EDMA_TPCC_SECRH_RN_k (3)	W	32	0x0000 2044 + (0x200 * k)	0x4330 2044 + (0x200 * k)

Table 11-30. System EDMA_TPCC Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	SYS_EDMA_TPCC Physical Address (L3_MAIN Access)
EDMA_TPCC_IER_RN_k [3]	R	32	0x0000 2050 + (0x200 * k)	0x4330 2050 + (0x200 * k)
EDMA_TPCC_IERH_RN_k [3]	R	32	0x0000 2054 + (0x200 * k)	0x4330 2054 + (0x200 * k)
EDMA_TPCC_IECR_RN_k [3]	W	32	0x0000 2058 + (0x200 * k)	0x4330 2058 + (0x200 * k)
EDMA_TPCC_IECRH_RN_k [3]	W	32	0x0000 205C + (0x200 * k)	0x4330 205C + (0x200 * k)
EDMA_TPCC_IESR_RN_k [3]	W	32	0x0000 2060 + (0x200 * k)	0x4330 2060 + (0x200 * k)
EDMA_TPCC_IESRH_RN_k [3]	W	32	0x0000 2064 + (0x200 * k)	0x4330 2064 + (0x200 * k)
EDMA_TPCC_IPR_RN_k [3]	R	32	0x0000 2068 + (0x200 * k)	0x4330 2068 + (0x200 * k)
EDMA_TPCC_IPRH_RN_k [3]	R	32	0x0000 206C + (0x200 * k)	0x4330 206C + (0x200 * k)
EDMA_TPCC_ICR_RN_k [3]	W	32	0x0000 2070 + (0x200 * k)	0x4330 2070 + (0x200 * k)
EDMA_TPCC_ICRH_RN_k [3]	W	32	0x0000 2074 + (0x200 * k)	0x4330 2074 + (0x200 * k)
EDMA_TPCC_IEVAL_RN_k [3]	W	32	0x0000 2078 + (0x200 * k)	0x4330 2078 + (0x200 * k)
EDMA_TPCC_QER_RN_k [3]	R	32	0x0000 2080 + (0x200 * k)	0x4330 2080 + (0x200 * k)
EDMA_TPCC_QEER_RN_k [3]	R	32	0x0000 2084 + (0x200 * k)	0x4330 2084 + (0x200 * k)
EDMA_TPCC_QEECR_RN_k [3]	W	32	0x0000 2088 + (0x200 * k)	0x4330 2088 + (0x200 * k)
EDMA_TPCC_QEESR_RN_k [3]	W	32	0x0000 208C + (0x200 * k)	0x4330 208C + (0x200 * k)
EDMA_TPCC_QSER_RN_k [3]	R	32	0x0000 2090 + (0x200 * k)	0x4330 2090 + (0x200 * k)
EDMA_TPCC_QSECR_RN_k [3]	W	32	0x0000 2094 + (0x200 * k)	0x4330 2094 + (0x200 * k)
EDMA_TPCC_OPT_n [6]	RW	32	0x0000 4000 + (0x20 * n)	0x4330 4000 + (0x20 * n)
EDMA_TPCC_SRC_n [6]	RW	32	0x0000 4004 + (0x20 * n)	0x4330 4004 + (0x20 * n)
EDMA_TPCC_ABCNT_n [6]	RW	32	0x0000 4008 + (0x20 * n)	0x4330 4008 + (0x20 * n)
EDMA_TPCC_DST_n [6]	RW	32	0x0000 400C + (0x20 * n)	0x4330 400C + (0x20 * n)
EDMA_TPCC_BIDX_n [6]	RW	32	0x0000 4010 + (0x20 * n)	0x4330 4010 + (0x20 * n)
EDMA_TPCC_LNK_n [6]	RW	32	0x0000 4014 + (0x20 * n)	0x4330 4014 + (0x20 * n)
EDMA_TPCC_CIDX_n [6]	RW	32	0x0000 4018 + (0x20 * n)	0x4330 4018 + (0x20 * n)
EDMA_TPCC_CCNT_n [6]	RW	32	0x0000 401C + (0x20 * n)	0x4330 401C + (0x20 * n)

[6] n = 0 to 512 for SYS_EDMA_TPCC

Table 11-31. DSP EDMA_TPCC Registers Mapping Summary (L3_MAIN Access)

Register Name	Type	Register Width (Bits)	Address Offset	DSP1_EDMA_TPCC Physical Address (L3_MAIN Access)	DSP2_EDMA_TPCC Physical Address (L3_MAIN Access)
EDMA_TPCC_PID	R	32	0x0000 0000	0x40D1 0000	0x4151 0000
EDMA_TPCC_CCCFG	R	32	0x0000 0004	0x40D1 0004	0x4151 0004
EDMA_TPCC_CLKGDIS	RW	32	0x0000 00FC	0x40D1 00FC	0x4151 00FC
EDMA_TPCC_DCHMAPN_m [1]	RW	32	0x0000 0100 + (0x4 * m)	0x40D1 0100 + (0x4 * m)	0x4151 0100 + (0x4 * m)
EDMA_TPCC_QCHMAPN_j [2]	RW	32	0x0000 0200 + (0x4 * j)	0x40D1 0200 + (0x4 * j)	0x4151 0200 + (0x4 * j)
EDMA_TPCC_DMAQNUMN_k [3]	RW	32	0x0000 0240 + (0x4 * k)	0x40D1 0240 + (0x4 * k)	0x4151 0240 + (0x4 * k)
EDMA_TPCC_QDMAQNUM	RW	32	0x0000 0260	0x40D1 0260	0x4151 0260
EDMA_TPCC_QUETCMAP	RW	32	0x0000 0280	0x40D1 0280	0x4151 0280
EDMA_TPCC_QUEPRI	RW	32	0x0000 0284	0x40D1 0284	0x4151 0284
EDMA_TPCC_EMR	R	32	0x0000 0300	0x40D1 0300	0x4151 0300
EDMA_TPCC_EMRH	R	32	0x0000 0304	0x40D1 0304	0x4151 0304

[1] m = 0 to 63 for DSPx_EDMA_TPCC
[2] j = 0 to 7 for DSPx_EDMA_TPCC
[3] k = 0 to 7 for DSPx_EDMA_TPCC

Table 11-31. DSP EDMA_TPCC Registers Mapping Summary (L3_MAIN Access) (continued)

Register Name	Type	Register Width (Bits)	Address Offset	DSP1_EDMA_TPCC Physical Address (L3_MAIN Access)	DSP2_EDMA_TPCC Physical Address (L3_MAIN Access)
EDMA_TPCC_EMCR	W	32	0x0000 0308	0x40D1 0308	0x4151 0308
EDMA_TPCC_EMCRH	W	32	0x0000 030C	0x40D1 030C	0x4151 030C
EDMA_TPCC_QEMR	R	32	0x0000 0310	0x40D1 0310	0x4151 0310
EDMA_TPCC_QEMCR	W	32	0x0000 0314	0x40D1 0314	0x4151 0314
EDMA_TPCC_CCERR	R	32	0x0000 0318	0x40D1 0318	0x4151 0318
EDMA_TPCC_CCERRCLR	W	32	0x0000 031C	0x40D1 031C	0x4151 031C
EDMA_TPCC_EEVAL	W	32	0x0000 0320	0x40D1 0320	0x4151 0320
EDMA_TPCC_DRAEM_k (3)	RW	32	0x0000 0340 + (0x8 * k)	0x40D1 0340 + (0x8 * k)	0x4151 0340 + (0x8 * k)
EDMA_TPCC_DRAEHM_k (3)	RW	32	0x0000 0344 + (0x8 * k)	0x40D1 0344 + (0x8 * k)	0x4151 0344 + (0x8 * k)
EDMA_TPCC_QRAEN_k (3)	RW	32	0x0000 0380 + (0x4 * k)	0x40D1 0380 + (0x4 * k)	0x4151 0380 + (0x4 * k)
EDMA_TPCC_Q0E_p (4)	R	32	0x0000 0400 + (0x4 *pl)	0x40D1 0400 + (0x4 * p)	0x4151 0400 + (0x4 * p)
EDMA_TPCC_Q1E_p (4)	R	32	0x0000 0440 + (0x4 * p)	0x40D1 0440 + (0x4 * p)	0x4151 0440 + (0x4 * p)
EDMA_TPCC_QSTATN_i (5)	R	32	0x0000 0600 + (0x4 * i)	0x40D1 0600 + (0x4 * i)	0x4151 0600 + (0x4 * i)
EDMA_TPCC_QWMTHRA	RW	32	0x0000 0620	0x40D1 0620	0x4151 0620
EDMA_TPCC_QWMTHRB	RW	32	0x0000 0624	0x40D1 0624	0x4151 0624
EDMA_TPCC_CCSTAT	R	32	0x0000 0640	0x40D1 0640	0x4151 0640
EDMA_TPCC_AETCTL	RW	32	0x0000 0700	0x40D1 0700	0x4151 0700
EDMA_TPCC_AETSTAT	R	32	0x0000 0704	0x40D1 0704	0x4151 0704
EDMA_TPCC_AETCMD	W	32	0x0000 0708	0x40D1 0708	0x4151 0708
EDMA_TPCC_MPFAR	R	32	0x0000 0800	0x40D1 0800	0x4151 0800
EDMA_TPCC_MPFSR	R	32	0x0000 0804	0x40D1 0804	0x4151 0804
EDMA_TPCC_MPFCR	W	32	0x0000 0808	0x40D1 0808	0x4151 0808
EDMA_TPCC_MPPAG	RW	32	0x0000 080C	0x40D1 080C	0x4151 080C
EDMA_TPCC_MPPAN_k (3)	RW	32	0x0000 0810 + (0x4 * k)	0x40D1 0810 + (0x4 * k)	0x4151 0810 + (0x4 * k)
EDMA_TPCC_ER	R	32	0x0000 1000	0x40D1 1000	0x4151 1000
EDMA_TPCC_ERH	R	32	0x0000 1004	0x40D1 1004	0x4151 1004
EDMA_TPCC_ECR	W	32	0x0000 1008	0x40D1 1008	0x4151 1008
EDMA_TPCC_ECRH	W	32	0x0000 100C	0x40D1 100C	0x4151 100C
EDMA_TPCC_ESR	W	32	0x0000 1010	0x40D1 1010	0x4151 1010
EDMA_TPCC_ESRH	W	32	0x0000 1014	0x40D1 1014	0x4151 1014
EDMA_TPCC_CER	R	32	0x0000 1018	0x40D1 1018	0x4151 1018
EDMA_TPCC_CERH	R	32	0x0000 101C	0x40D1 101C	0x4151 101C
EDMA_TPCC_EER	R	32	0x0000 1020	0x40D1 1020	0x4151 1020
EDMA_TPCC_EERH	R	32	0x0000 1024	0x40D1 1024	0x4151 1024
EDMA_TPCC_EECR	W	32	0x0000 1028	0x40D1 1028	0x4151 1028
EDMA_TPCC_EECRH	W	32	0x0000 102C	0x40D1 102C	0x4151 102C
EDMA_TPCC_EESR	W	32	0x0000 1030	0x40D1 1030	0x4151 1030
EDMA_TPCC_EESRH	W	32	0x0000 1034	0x40D1 1034	0x4151 1034
EDMA_TPCC_SER	R	32	0x0000 1038	0x40D1 1038	0x4151 1038
EDMA_TPCC_SERH	R	32	0x0000 103C	0x40D1 103C	0x4151 103C

(4) p = 0 to 15 for DSPx_EDMA_TPCC
(5) i = 0 to 1 for DSPx_EDMA_TPCC

Table 11-31. DSP EDMA_TPCC Registers Mapping Summary (L3_MAIN Access) (continued)

Register Name	Type	Register Width (Bits)	Address Offset	DSP1_EDMA_TPCC Physical Address (L3_MAIN Access)	DSP2_EDMA_TPCC Physical Address (L3_MAIN Access)
EDMA_TPCC_SECR	W	32	0x0000 1040	0x40D1 1040	0x4151 1040
EDMA_TPCC_SECRH	W	32	0x0000 1044	0x40D1 1044	0x4151 1044
EDMA_TPCC_IER	R	32	0x0000 1050	0x40D1 1050	0x4151 1050
EDMA_TPCC_IERH	R	32	0x0000 1054	0x40D1 1054	0x4151 1054
EDMA_TPCC_IECR	W	32	0x0000 1058	0x40D1 1058	0x4151 1058
EDMA_TPCC_IECRH	W	32	0x0000 105C	0x40D1 105C	0x4151 105C
EDMA_TPCC_IESR	W	32	0x0000 1060	0x40D1 1060	0x4151 1060
EDMA_TPCC_IESRH	W	32	0x0000 1064	0x40D1 1064	0x4151 1064
EDMA_TPCC_IPR	R	32	0x0000 1068	0x40D1 1068	0x4151 1068
EDMA_TPCC_IPRH	R	32	0x0000 106C	0x40D1 106C	0x4151 106C
EDMA_TPCC_ICR	W	32	0x0000 1070	0x40D1 1070	0x4151 1070
EDMA_TPCC_ICRH	W	32	0x0000 1074	0x40D1 1074	0x4151 1074
EDMA_TPCC_IEVAL	W	32	0x0000 1078	0x40D1 1078	0x4151 1078
EDMA_TPCC_QER	R	32	0x0000 1080	0x40D1 1080	0x4151 1080
EDMA_TPCC_QEER	R	32	0x0000 1084	0x40D1 1084	0x4151 1084
EDMA_TPCC_QEECR	W	32	0x0000 1088	0x40D1 1088	0x4151 1088
EDMA_TPCC_QEESR	W	32	0x0000 108C	0x40D1 108C	0x4151 108C
EDMA_TPCC_QSER	R	32	0x0000 1090	0x40D1 1090	0x4151 1090
EDMA_TPCC_QSECR	W	32	0x0000 1094	0x40D1 1094	0x4151 1094
EDMA_TPCC_ER_RN_k (3)	R	32	0x0000 2000 + (0x200 * k)	0x40D1 2000 + (0x200 * k)	0x4151 2000 + (0x200 * k)
EDMA_TPCC_ERH_RN_k (3)	R	32	0x0000 2004 + (0x200 * k)	0x40D1 2004 + (0x200 * k)	0x4151 2004 + (0x200 * k)
EDMA_TPCC_ECR_RN_k (3)	W	32	0x0000 2008 + (0x200 * k)	0x40D1 2008 + (0x200 * k)	0x4151 2008 + (0x200 * k)
EDMA_TPCC_ECRH_RN_k (3)	W	32	0x0000 200C + (0x200 * k)	0x40D1 200C + (0x200 * k)	0x4151 200C + (0x200 * k)
EDMA_TPCC_ESR_RN_k (3)	W	32	0x0000 2010 + (0x200 * k)	0x40D1 2010 + (0x200 * k)	0x4151 2010 + (0x200 * k)
EDMA_TPCC_ESRH_RN_k (3)	W	32	0x0000 2014 + (0x200 * k)	0x40D1 2014 + (0x200 * k)	0x4151 2014 + (0x200 * k)
EDMA_TPCC_CER_RN_k (3)	R	32	0x0000 2018 + (0x200 * k)	0x40D1 2018 + (0x200 * k)	0x4151 2018 + (0x200 * k)
EDMA_TPCC_CERH_RN_k (3)	R	32	0x0000 201C + (0x200 * k)	0x40D1 201C + (0x200 * k)	0x4151 201C + (0x200 * k)
EDMA_TPCC_EER_RN_k (3)	R	32	0x0000 2020 + (0x200 * k)	0x40D1 2020 + (0x200 * k)	0x4151 2020 + (0x200 * k)
EDMA_TPCC_EERH_RN_k (3)	R	32	0x0000 2024 + (0x200 * k)	0x40D1 2024 + (0x200 * k)	0x4151 2024 + (0x200 * k)
EDMA_TPCC_EECR_RN_k (3)	W	32	0x0000 2028 + (0x200 * k)	0x40D1 2028 + (0x200 * k)	0x4151 2028 + (0x200 * k)
EDMA_TPCC_EECRH_RN_k (3)	W	32	0x0000 202C + (0x200 * k)	0x40D1 202C + (0x200 * k)	0x4151 202C + (0x200 * k)
EDMA_TPCC_EESR_RN_k (3)	W	32	0x0000 2030 + (0x200 * k)	0x40D1 2030 + (0x200 * k)	0x4151 2030 + (0x200 * k)
EDMA_TPCC_EESRH_RN_k (3)	W	32	0x0000 2034 + (0x200 * k)	0x40D1 2034 + (0x200 * k)	0x4151 2034 + (0x200 * k)
EDMA_TPCC_SER_RN_k (3)	R	32	0x0000 2038 + (0x200 * k)	0x40D1 2038 + (0x200 * k)	0x4151 2038 + (0x200 * k)
EDMA_TPCC_SERH_RN_k (3)	R	32	0x0000 203C + (0x200 * k)	0x40D1 203C + (0x200 * k)	0x4151 203C + (0x200 * k)

Table 11-31. DSP EDMA_TPCC Registers Mapping Summary (L3_MAIN Access) (continued)

Register Name	Type	Register Width (Bits)	Address Offset	DSP1_EDMA_TPCC Physical Address (L3_MAIN Access)	DSP2_EDMA_TPCC Physical Address (L3_MAIN Access)
EDMA_TPCC_SECR_RN_k [3]	W	32	0x0000 2040 + (0x200 * k)	0x40D1 2040 + (0x200 * k)	0x4151 2040 + (0x200 * k)
EDMA_TPCC_SECRH_RN_k [3]	W	32	0x0000 2044 + (0x200 * k)	0x40D1 2044 + (0x200 * k)	0x4151 2044 + (0x200 * k)
EDMA_TPCC_IER_RN_k [3]	R	32	0x0000 2050 + (0x200 * k)	0x40D1 2050 + (0x200 * k)	0x4151 2050 + (0x200 * k)
EDMA_TPCC_IERH_RN_k [3]	R	32	0x0000 2054 + (0x200 * k)	0x40D1 2054 + (0x200 * k)	0x4151 2054 + (0x200 * k)
EDMA_TPCC_IECR_RN_k [3]	W	32	0x0000 2058 + (0x200 * k)	0x40D1 2058 + (0x200 * k)	0x4151 2058 + (0x200 * k)
EDMA_TPCC_IECRH_RN_k [3]	W	32	0x0000 205C + (0x200 * k)	0x40D1 205C + (0x200 * k)	0x4151 205C + (0x200 * k)
EDMA_TPCC_IESR_RN_k [3]	W	32	0x0000 2060 + (0x200 * k)	0x40D1 2060 + (0x200 * k)	0x4151 2060 + (0x200 * k)
EDMA_TPCC_IESRH_RN_k [3]	W	32	0x0000 2064 + (0x200 * k)	0x40D1 2064 + (0x200 * k)	0x4151 2064 + (0x200 * k)
EDMA_TPCC_IPR_RN_k [3]	R	32	0x0000 2068 + (0x200 * k)	0x40D1 2068 + (0x200 * k)	0x4151 2068 + (0x200 * k)
EDMA_TPCC_IPRH_RN_k [3]	R	32	0x0000 206C + (0x200 * k)	0x40D1 206C + (0x200 * k)	0x4151 206C + (0x200 * k)
EDMA_TPCC_ICR_RN_k [3]	W	32	0x0000 2070 + (0x200 * k)	0x40D1 2070 + (0x200 * k)	0x4151 2070 + (0x200 * k)
EDMA_TPCC_ICRH_RN_k [3]	W	32	0x0000 2074 + (0x200 * k)	0x40D1 2074 + (0x200 * k)	0x4151 2074 + (0x200 * k)
EDMA_TPCC_IEVAL_RN_k [3]	W	32	0x0000 2078 + (0x200 * k)	0x40D1 2078 + (0x200 * k)	0x4151 2078 + (0x200 * k)
EDMA_TPCC_QER_RN_k [3]	R	32	0x0000 2080 + (0x200 * k)	0x40D1 2080 + (0x200 * k)	0x4151 2080 + (0x200 * k)
EDMA_TPCC_QEER_RN_k [3]	R	32	0x0000 2084 + (0x200 * k)	0x40D1 2084 + (0x200 * k)	0x4151 2084 + (0x200 * k)
EDMA_TPCC_QEECR_RN_k [3]	W	32	0x0000 2088 + (0x200 * k)	0x40D1 2088 + (0x200 * k)	0x4151 2088 + (0x200 * k)
EDMA_TPCC_QEESR_RN_k [3]	W	32	0x0000 208C + (0x200 * k)	0x40D1 208C + (0x200 * k)	0x4151 208C + (0x200 * k)
EDMA_TPCC_QSER_RN_k [3]	R	32	0x0000 2090 + (0x200 * k)	0x40D1 2090 + (0x200 * k)	0x4151 2090 + (0x200 * k)
EDMA_TPCC_QSECR_RN_k [3]	W	32	0x0000 2094 + (0x200 * k)	0x40D1 2094 + (0x200 * k)	0x4151 2094 + (0x200 * k)
EDMA_TPCC_OPT_n [6]	RW	32	0x0000 4000 + (0x20 * n)	0x40D1 4000 + (0x20 * n)	0x4151 4000 + (0x20 * n)
EDMA_TPCC_SRC_n [6]	RW	32	0x0000 4004 + (0x20 * n)	0x40D1 4004 + (0x20 * n)	0x4151 4004 + (0x20 * n)
EDMA_TPCC_ABCNT_n [6]	RW	32	0x0000 4008 + (0x20 * n)	0x40D1 4008 + (0x20 * n)	0x4151 4008 + (0x20 * n)
EDMA_TPCC_DST_n [6]	RW	32	0x0000 400C + (0x20 * n)	0x40D1 400C + (0x20 * n)	0x4151 400C + (0x20 * n)
EDMA_TPCC_BIDX_n [6]	RW	32	0x0000 4010 + (0x20 * n)	0x40D1 4010 + (0x20 * n)	0x4151 4010 + (0x20 * n)
EDMA_TPCC_LNK_n [6]	RW	32	0x0000 4014 + (0x20 * n)	0x40D1 4014 + (0x20 * n)	0x4151 4014 + (0x20 * n)
EDMA_TPCC_CIDX_n [6]	RW	32	0x0000 4018 + (0x20 * n)	0x40D1 4018 + (0x20 * n)	0x4151 4018 + (0x20 * n)
EDMA_TPCC_CCNT_n [6]	RW	32	0x0000 401C + (0x20 * n)	0x40D1 401C + (0x20 * n)	0x4151 401C + (0x20 * n)

[6] n = 0 to 127 for DSPx_EDMA_TPCC

Table 11-32. DSP EDMA_TPCC Registers Mapping Summary (Private Access)

Register Name	Type	Register Width (Bits)	Address Offset	DSP_EDMA_TPCC Physical Address (DSP Private Access)
EDMA_TPCC_PID	R	32	0x0000 0000	0x01D1 0000
EDMA_TPCC_CCCFG	R	32	0x0000 0004	0x01D1 0004
EDMA_TPCC_CLKGDIS	RW	32	0x0000 00FC	0x01D1 00FC
EDMA_TPCC_DCHMAPN_m (1)	RW	32	0x0000 0100 + (0x4 * m)	0x01D1 0100 + (0x4 * m)
EDMA_TPCC_QCHMAPN_j (2)	RW	32	0x0000 0200 + (0x4 * j)	0x01D1 0200 + (0x4 * j)
EDMA_TPCC_DMAQNUMN_k (3)	RW	32	0x0000 0240 + (0x4 * k)	0x01D1 0240 + (0x4 * k)
EDMA_TPCC_QDMAQNUM	RW	32	0x0000 0260	0x01D1 0260
EDMA_TPCC_QUETCMAP	RW	32	0x0000 0280	0x01D1 0280
EDMA_TPCC_QUEPRI	RW	32	0x0000 0284	0x01D1 0284
EDMA_TPCC_EMR	R	32	0x0000 0300	0x01D1 0300
EDMA_TPCC_EMRH	R	32	0x0000 0304	0x01D1 0304
EDMA_TPCC_EMCR	W	32	0x0000 0308	0x01D1 0308
EDMA_TPCC_EMCRH	W	32	0x0000 030C	0x01D1 030C
EDMA_TPCC_QEMR	R	32	0x0000 0310	0x01D1 0310
EDMA_TPCC_QEMCR	W	32	0x0000 0314	0x01D1 0314
EDMA_TPCC_CCERR	R	32	0x0000 0318	0x01D1 0318
EDMA_TPCC_CCERRCLR	W	32	0x0000 031C	0x01D1 031C
EDMA_TPCC_EEVAL	W	32	0x0000 0320	0x01D1 0320
EDMA_TPCC_DRAEM_k (3)	RW	32	0x0000 0340 + (0x8 * k)	0x01D1 0340 + (0x8 * k)
EDMA_TPCC_DRAEHM_k (3)	RW	32	0x0000 0344 + (0x8 * k)	0x01D1 0344 + (0x8 * k)
EDMA_TPCC_QRAEN_k (3)	RW	32	0x0000 0380 + (0x4 * k)	0x01D1 0380 + (0x4 * k)
EDMA_TPCC_Q0E_p (4)	R	32	0x0000 0400 + (0x4 *pl)	0x01D1 0400 + (0x4 * p)
EDMA_TPCC_Q1E_p (4)	R	32	0x0000 0440 + (0x4 * p)	0x01D1 0440 + (0x4 * p)
EDMA_TPCC_QSTATN_i (5)	R	32	0x0000 0600 + (0x4 * i)	0x01D1 0600 + (0x4 * i)
EDMA_TPCC_QWMTHRA	RW	32	0x0000 0620	0x01D1 0620
EDMA_TPCC_QWMTHRB	RW	32	0x0000 0624	0x01D1 0624
EDMA_TPCC_CCSTAT	R	32	0x0000 0640	0x01D1 0640
EDMA_TPCC_AETCTL	RW	32	0x0000 0700	0x01D1 0700
EDMA_TPCC_AETSTAT	R	32	0x0000 0704	0x01D1 0704
EDMA_TPCC_AETCMD	W	32	0x0000 0708	0x01D1 0708
EDMA_TPCC_MPFAR	R	32	0x0000 0800	0x01D1 0800
EDMA_TPCC_MPFSR	R	32	0x0000 0804	0x01D1 0804
EDMA_TPCC_MPFCR	W	32	0x0000 0808	0x01D1 0808
EDMA_TPCC_MPPAG	RW	32	0x0000 080C	0x01D1 080C
EDMA_TPCC_MPPAN_k (3)	RW	32	0x0000 0810 + (0x4 * k)	0x01D1 0810 + (0x4 * k)
EDMA_TPCC_ER	R	32	0x0000 1000	0x01D1 1000
EDMA_TPCC_ERH	R	32	0x0000 1004	0x01D1 1004
EDMA_TPCC_ECR	W	32	0x0000 1008	0x01D1 1008
EDMA_TPCC_ECRH	W	32	0x0000 100C	0x01D1 100C
EDMA_TPCC_ESR	W	32	0x0000 1010	0x01D1 1010
EDMA_TPCC_ESRH	W	32	0x0000 1014	0x01D1 1014
EDMA_TPCC_CER	R	32	0x0000 1018	0x01D1 1018

(1) m = 0 to 63 for DSP_EDMA_TPCC
(2) j = 0 to 7 for DSP_EDMA_TPCC
(3) k = 0 to 7 for DSP_EDMA_TPCC
(4) p = 0 to 15 for DSP_EDMA_TPCC
(5) i = 0 to 1 for DSP_EDMA_TPCC

Table 11-32. DSP EDMA_TPCC Registers Mapping Summary (Private Access) (continued)

Register Name	Type	Register Width (Bits)	Address Offset	DSP_EDMA_TPCC Physical Address (DSP Private Access)
EDMA_TPCC_CERH	R	32	0x0000 101C	0x01D1 101C
EDMA_TPCC_EER	R	32	0x0000 1020	0x01D1 1020
EDMA_TPCC_EERH	R	32	0x0000 1024	0x01D1 1024
EDMA_TPCC_EECR	W	32	0x0000 1028	0x01D1 1028
EDMA_TPCC_EECRH	W	32	0x0000 102C	0x01D1 102C
EDMA_TPCC_EESR	W	32	0x0000 1030	0x01D1 1030
EDMA_TPCC_EESRH	W	32	0x0000 1034	0x01D1 1034
EDMA_TPCC_SER	R	32	0x0000 1038	0x01D1 1038
EDMA_TPCC_SERH	R	32	0x0000 103C	0x01D1 103C
EDMA_TPCC_SECR	W	32	0x0000 1040	0x01D1 1040
EDMA_TPCC_SECRH	W	32	0x0000 1044	0x01D1 1044
EDMA_TPCC_IER	R	32	0x0000 1050	0x01D1 1050
EDMA_TPCC_IERH	R	32	0x0000 1054	0x01D1 1054
EDMA_TPCC_IECR	W	32	0x0000 1058	0x01D1 1058
EDMA_TPCC_IECRH	W	32	0x0000 105C	0x01D1 105C
EDMA_TPCC_IESR	W	32	0x0000 1060	0x01D1 1060
EDMA_TPCC_IESRH	W	32	0x0000 1064	0x01D1 1064
EDMA_TPCC_IPR	R	32	0x0000 1068	0x01D1 1068
EDMA_TPCC_IPRH	R	32	0x0000 106C	0x01D1 106C
EDMA_TPCC_ICR	W	32	0x0000 1070	0x01D1 1070
EDMA_TPCC_ICRH	W	32	0x0000 1074	0x01D1 1074
EDMA_TPCC_IEVAL	W	32	0x0000 1078	0x01D1 1078
EDMA_TPCC_QER	R	32	0x0000 1080	0x01D1 1080
EDMA_TPCC_QEER	R	32	0x0000 1084	0x01D1 1084
EDMA_TPCC_QEECR	W	32	0x0000 1088	0x01D1 1088
EDMA_TPCC_QEESR	W	32	0x0000 108C	0x01D1 108C
EDMA_TPCC_QSER	R	32	0x0000 1090	0x01D1 1090
EDMA_TPCC_QSECR	W	32	0x0000 1094	0x01D1 1094
EDMA_TPCC_ER_RN_k (3)	R	32	0x0000 2000 + (0x200 * k)	0x01D1 2000 + (0x200 * k)
EDMA_TPCC_ERH_RN_k (3)	R	32	0x0000 2004 + (0x200 * k)	0x01D1 2004 + (0x200 * k)
EDMA_TPCC_ECR_RN_k (3)	W	32	0x0000 2008 + (0x200 * k)	0x01D1 2008 + (0x200 * k)
EDMA_TPCC_ECRH_RN_k (3)	W	32	0x0000 200C + (0x200 * k)	0x01D1 200C + (0x200 * k)
EDMA_TPCC_ESR_RN_k (3)	W	32	0x0000 2010 + (0x200 * k)	0x01D1 2010 + (0x200 * k)
EDMA_TPCC_ESRH_RN_k (3)	W	32	0x0000 2014 + (0x200 * k)	0x01D1 2014 + (0x200 * k)
EDMA_TPCC_CER_RN_k (3)	R	32	0x0000 2018 + (0x200 * k)	0x01D1 2018 + (0x200 * k)
EDMA_TPCC_CERH_RN_k (3)	R	32	0x0000 201C + (0x200 * k)	0x01D1 201C + (0x200 * k)
EDMA_TPCC_EER_RN_k (3)	R	32	0x0000 2020 + (0x200 * k)	0x01D1 2020 + (0x200 * k)
EDMA_TPCC_EERH_RN_k (3)	R	32	0x0000 2024 + (0x200 * k)	0x01D1 2024 + (0x200 * k)
EDMA_TPCC_EECR_RN_k (3)	W	32	0x0000 2028 + (0x200 * k)	0x01D1 2028 + (0x200 * k)
EDMA_TPCC_EECRH_RN_k (3)	W	32	0x0000 202C + (0x200 * k)	0x01D1 202C + (0x200 * k)
EDMA_TPCC_EESR_RN_k (3)	W	32	0x0000 2030 + (0x200 * k)	0x01D1 2030 + (0x200 * k)
EDMA_TPCC_EESRH_RN_k (3)	W	32	0x0000 2034 + (0x200 * k)	0x01D1 2034 + (0x200 * k)
EDMA_TPCC_SER_RN_k (3)	R	32	0x0000 2038 + (0x200 * k)	0x01D1 2038 + (0x200 * k)
EDMA_TPCC_SERH_RN_k (3)	R	32	0x0000 203C + (0x200 * k)	0x01D1 203C + (0x200 * k)
EDMA_TPCC_SECR_RN_k (3)	W	32	0x0000 2040 + (0x200 * k)	0x01D1 2040 + (0x200 * k)
EDMA_TPCC_SECRH_RN_k (3)	W	32	0x0000 2044 + (0x200 * k)	0x01D1 2044 + (0x200 * k)

Table 11-32. DSP EDMA_TPCC Registers Mapping Summary (Private Access) (continued)

Register Name	Type	Register Width (Bits)	Address Offset	DSP_EDMA_TPCC Physical Address (DSP Private Access)
EDMA_TPCC_IER_RN_k [3]	R	32	0x0000 2050 + (0x200 * k)	0x01D1 2050 + (0x200 * k)
EDMA_TPCC_IERH_RN_k [3]	R	32	0x0000 2054 + (0x200 * k)	0x01D1 2054 + (0x200 * k)
EDMA_TPCC_IECR_RN_k [3]	W	32	0x0000 2058 + (0x200 * k)	0x01D1 2058 + (0x200 * k)
EDMA_TPCC_IECRH_RN_k [6]	W	32	0x0000 205C + (0x200 * k)	0x01D1 205C + (0x200 * k)
EDMA_TPCC_IESR_RN_k [3]	W	32	0x0000 2060 + (0x200 * k)	0x01D1 2060 + (0x200 * k)
EDMA_TPCC_IESRH_RN_k [3]	W	32	0x0000 2064 + (0x200 * k)	0x01D1 2064 + (0x200 * k)
EDMA_TPCC_IPR_RN_k [3]	R	32	0x0000 2068 + (0x200 * k)	0x01D1 2068 + (0x200 * k)
EDMA_TPCC_IPRH_RN_k [3]	R	32	0x0000 206C + (0x200 * k)	0x01D1 206C + (0x200 * k)
EDMA_TPCC_ICR_RN_k [3]	W	32	0x0000 2070 + (0x200 * k)	0x01D1 2070 + (0x200 * k)
EDMA_TPCC_ICRH_RN_k [3]	W	32	0x0000 2074 + (0x200 * k)	0x01D1 2074 + (0x200 * k)
EDMA_TPCC_IEVAL_RN_k [3]	W	32	0x0000 2078 + (0x200 * k)	0x01D1 2078 + (0x200 * k)
EDMA_TPCC_QER_RN_k [3]	R	32	0x0000 2080 + (0x200 * k)	0x01D1 2080 + (0x200 * k)
EDMA_TPCC_QEER_RN_k [3]	R	32	0x0000 2084 + (0x200 * k)	0x01D1 2084 + (0x200 * k)
EDMA_TPCC_QEECR_RN_k [3]	W	32	0x0000 2088 + (0x200 * k)	0x01D1 2088 + (0x200 * k)
EDMA_TPCC_QEESR_RN_k [3]	W	32	0x0000 208C + (0x200 * k)	0x01D1 208C + (0x200 * k)
EDMA_TPCC_QSER_RN_k [3]	R	32	0x0000 2090 + (0x200 * k)	0x01D1 2090 + (0x200 * k)
EDMA_TPCC_QSECR_RN_k [3]	W	32	0x0000 2094 + (0x200 * k)	0x01D1 2094 + (0x200 * k)
EDMA_TPCC_OPT_n [7]	RW	32	0x0000 4000 + (0x20 * n)	0x01D1 4000 + (0x20 * n)
EDMA_TPCC_SRC_n [7]	RW	32	0x0000 4004 + (0x20 * n)	0x01D1 4004 + (0x20 * n)
EDMA_TPCC_ABCNT_n [7]	RW	32	0x0000 4008 + (0x20 * n)	0x01D1 4008 + (0x20 * n)
EDMA_TPCC_DST_n [7]	RW	32	0x0000 400C + (0x20 * n)	0x01D1 400C + (0x20 * n)
EDMA_TPCC_BIDX_n [7]	RW	32	0x0000 4010 + (0x20 * n)	0x01D1 4010 + (0x20 * n)
EDMA_TPCC_LNK_n [7]	RW	32	0x0000 4014 + (0x20 * n)	0x01D1 4014 + (0x20 * n)
EDMA_TPCC_CIDX_n [7]	RW	32	0x0000 4018 + (0x20 * n)	0x01D1 4018 + (0x20 * n)
EDMA_TPCC_CCNT_n [7]	RW	32	0x0000 401C + (0x20 * n)	0x01D1 401C + (0x20 * n)

(6) k = 0 to 7 for DSPx_EDMA_TPCC
(7) n = 0 to 127 for DSP_EDMA_TPCC

Table 11-33. EVE EDMA_TPCC Registers Mapping Summary (L3_MAIN Access)

Register Name	Type	Register Width (Bits)	Address Offset	EVE_EDMA_TPCC Physical Address (L3_MAIN Access)
EDMA_TPCC_PID	R	32	0x0000 0000	0x420A 0000
EDMA_TPCC_CCCFG	R	32	0x0000 0004	0x420A 0004
EDMA_TPCC_CLKGDIS	RW	32	0x0000 00FC	0x420A 00FC
EDMA_TPCC_DCHMAPN_m [1]	RW	32	0x0000 0100 + (0x4 * m)	0x420A 0100 + (0x4 * m)
EDMA_TPCC_QCHMAPN_j [2]	RW	32	0x0000 0200 + (0x4 * j)	0x420A 0200 + (0x4 * j)
EDMA_TPCC_DMAQNUMN_k [3]	RW	32	0x0000 0240 + (0x4 * k)	0x420A 0240 + (0x4 * k)
EDMA_TPCC_QDMAQNUM	RW	32	0x0000 0260	0x420A 0260
EDMA_TPCC_QUETCMAP	RW	32	0x0000 0280	0x420A 0280
EDMA_TPCC_QUEPRI	RW	32	0x0000 0284	0x420A 0284
EDMA_TPCC_EMR	R	32	0x0000 0300	0x420A 0300
EDMA_TPCC_EMRH	R	32	0x0000 0304	0x420A 0304
EDMA_TPCC_EMCR	W	32	0x0000 0308	0x420A 0308

(1) m = 0 to 15 for EVE_EDMA_TPCC
(2) j = 0 to 7 for EVE_EDMA_TPCC
(3) k = 0 to 7 for EVE_EDMA_TPCC

Table 11-33. EVE EDMA_TPCC Registers Mapping Summary (L3_MAIN Access) (continued)

Register Name	Type	Register Width (Bits)	Address Offset	EVE_EDMA_TPCC Physical Address (L3_MAIN Access)
EDMA_TPCC_EMCRH	W	32	0x0000 030C	0x420A 030C
EDMA_TPCC_QEMR	R	32	0x0000 0310	0x420A 0310
EDMA_TPCC_QEMCR	W	32	0x0000 0314	0x420A 0314
EDMA_TPCC_CCERR	R	32	0x0000 0318	0x420A 0318
EDMA_TPCC_CCERRCLR	W	32	0x0000 031C	0x420A 031C
EDMA_TPCC_EEVAL	W	32	0x0000 0320	0x420A 0320
EDMA_TPCC_DRAEM_k (3)	RW	32	0x0000 0340 + (0x8 * k)	-
EDMA_TPCC_DRAEHM_k (3)	RW	32	0x0000 0344 + (0x8 * k)	-
EDMA_TPCC_QRAEN_k (3)	RW	32	0x0000 0380 + (0x4 * k)	0x420A 0380 + (0x4 * k)
EDMA_TPCC_Q0E_p (4)	R	32	0x0000 0400 + (0x4 * p)	-
EDMA_TPCC_Q1E_p (4)	R	32	0x0000 0440 + (0x4 *pl)	-
EDMA_TPCC_QSTATN_i (5)	R	32	0x0000 0600 + (0x4 * i)	0x420A 0600 + (0x4 * i)
EDMA_TPCC_QWMTHRA	RW	32	0x0000 0620	0x420A 0620
EDMA_TPCC_QWMTHRB	RW	32	0x0000 0624	0x420A 0624
EDMA_TPCC_CCSTAT	R	32	0x0000 0640	0x420A 0640
EDMA_TPCC_AETCTL	RW	32	0x0000 0700	0x420A 0700
EDMA_TPCC_AETSTAT	R	32	0x0000 0704	0x420A 0704
EDMA_TPCC_AETCMD	W	32	0x0000 0708	0x420A 0708
EDMA_TPCC_MPFAR	R	32	0x0000 0800	0x420A 0800
EDMA_TPCC_MPFSR	R	32	0x0000 0804	0x420A 0804
EDMA_TPCC_MPFCR	W	32	0x0000 0808	0x420A 0808
EDMA_TPCC_MPPAG	RW	32	0x0000 080C	0x420A 080C
EDMA_TPCC_MPPAN_k (3)	RW	32	0x0000 0810 + (0x4 * k)	0x420A 0810 + (0x4 * k)
EDMA_TPCC_ER	R	32	0x0000 1000	0x420A 1000
EDMA_TPCC_ERH	R	32	0x0000 1004	0x420A 1004
EDMA_TPCC_ECR	W	32	0x0000 1008	0x420A 1008
EDMA_TPCC_ECRH	W	32	0x0000 100C	0x420A 100C
EDMA_TPCC_ESR	W	32	0x0000 1010	0x420A 1010
EDMA_TPCC_ESRH	W	32	0x0000 1014	0x420A 1014
EDMA_TPCC_CER	R	32	0x0000 1018	0x420A 1018
EDMA_TPCC_CERH	R	32	0x0000 101C	0x420A 101C
EDMA_TPCC_EER	R	32	0x0000 1020	0x420A 1020
EDMA_TPCC_EERH	R	32	0x0000 1024	0x420A 1024
EDMA_TPCC_EECR	W	32	0x0000 1028	0x420A 1028
EDMA_TPCC_EECRH	W	32	0x0000 102C	0x420A 102C
EDMA_TPCC_EESR	W	32	0x0000 1030	0x420A 1030
EDMA_TPCC_EESRH	W	32	0x0000 1034	0x420A 1034
EDMA_TPCC_SER	R	32	0x0000 1038	0x420A 1038
EDMA_TPCC_SERH	R	32	0x0000 103C	0x420A 103C
EDMA_TPCC_SECR	W	32	0x0000 1040	0x420A 1040
EDMA_TPCC_SECRH	W	32	0x0000 1044	0x420A 1044
EDMA_TPCC_IER	R	32	0x0000 1050	0x420A 1050
EDMA_TPCC_IERH	R	32	0x0000 1054	0x420A 1054
EDMA_TPCC_IECR	W	32	0x0000 1058	0x420A 1058

(4) p = 0 to 1 for EVE_EDMA_TPCC
(5) i = 0 to 1 for EVE_EDMA_TPCC

Table 11-33. EVE EDMA_TPCC Registers Mapping Summary (L3_MAIN Access) (continued)

Register Name	Type	Register Width (Bits)	Address Offset	EVE_EDMA_TPCC Physical Address (L3_MAIN Access)
EDMA_TPCC_IECRH	W	32	0x0000 105C	0x420A 105C
EDMA_TPCC_IESR	W	32	0x0000 1060	0x420A 1060
EDMA_TPCC_IESRH	W	32	0x0000 1064	0x420A 1064
EDMA_TPCC_IPR	R	32	0x0000 1068	0x420A 1068
EDMA_TPCC_IPRH	R	32	0x0000 106C	0x420A 106C
EDMA_TPCC_ICR	W	32	0x0000 1070	0x420A 1070
EDMA_TPCC_ICRH	W	32	0x0000 1074	0x420A 1074
EDMA_TPCC_IEVAL	W	32	0x0000 1078	0x420A 1078
EDMA_TPCC_QER	R	32	0x0000 1080	0x420A 1080
EDMA_TPCC_QEER	R	32	0x0000 1084	0x420A 1084
EDMA_TPCC_QEECR	W	32	0x0000 1088	0x420A 1088
EDMA_TPCC_QEESR	W	32	0x0000 108C	0x420A 108C
EDMA_TPCC_QSER	R	32	0x0000 1090	0x420A 1090
EDMA_TPCC_QSECR	W	32	0x0000 1094	0x420A 1094
EDMA_TPCC_ER_RN_k (3)	R	32	0x0000 2000 + (0x200 * k)	0x420A 2000 + (0x200 * k)
EDMA_TPCC_ERH_RN_k (3)	R	32	0x0000 2004 + (0x200 * k)	0x420A 2004 + (0x200 * k)
EDMA_TPCC_ECR_RN_k (3)	W	32	0x0000 2008 + (0x200 * k)	0x420A 2008 + (0x200 * k)
EDMA_TPCC_ECRH_RN_k (3)	W	32	0x0000 200C + (0x200 * k)	0x420A 200C + (0x200 * k)
EDMA_TPCC_ESR_RN_k (3)	W	32	0x0000 2010 + (0x200 * k)	0x420A 2010 + (0x200 * k)
EDMA_TPCC_ESRH_RN_k (3)	W	32	0x0000 2014 + (0x200 * k)	0x420A 2014 + (0x200 * k)
EDMA_TPCC_CER_RN_k (3)	R	32	0x0000 2018 + (0x200 * k)	0x420A 2018 + (0x200 * k)
EDMA_TPCC_CERH_RN_k (3)	R	32	0x0000 201C + (0x200 * k)	0x420A 201C + (0x200 * k)
EDMA_TPCC_EER_RN_k (3)	R	32	0x0000 2020 + (0x200 * k)	0x420A 2020 + (0x200 * k)
EDMA_TPCC_EERH_RN_k (3)	R	32	0x0000 2024 + (0x200 * k)	0x420A 2024 + (0x200 * k)
EDMA_TPCC_EECR_RN_k (3)	W	32	0x0000 2028 + (0x200 * k)	0x420A 2028 + (0x200 * k)
EDMA_TPCC_EECRH_RN_k (3)	W	32	0x0000 202C + (0x200 * k)	0x420A 202C + (0x200 * k)
EDMA_TPCC_EESR_RN_k (3)	W	32	0x0000 2030 + (0x200 * k)	0x420A 2030 + (0x200 * k)
EDMA_TPCC_EESRH_RN_k (3)	W	32	0x0000 2034 + (0x200 * k)	0x420A 2034 + (0x200 * k)
EDMA_TPCC_SER_RN_k (3)	R	32	0x0000 2038 + (0x200 * k)	0x420A 2038 + (0x200 * k)
EDMA_TPCC_SERH_RN_k (3)	R	32	0x0000 203C + (0x200 * k)	0x420A 203C + (0x200 * k)
EDMA_TPCC_SECR_RN_k (3)	W	32	0x0000 2040 + (0x200 * k)	0x420A 2040 + (0x200 * k)
EDMA_TPCC_SECRH_RN_k (3)	W	32	0x0000 2044 + (0x200 * k)	0x420A 2044 + (0x200 * k)
EDMA_TPCC_IER_RN_k (3)	R	32	0x0000 2050 + (0x200 * k)	0x420A 2050 + (0x200 * k)
EDMA_TPCC_IERH_RN_k (3)	R	32	0x0000 2054 + (0x200 * k)	0x420A 2054 + (0x200 * k)
EDMA_TPCC_IECR_RN_k (3)	W	32	0x0000 2058 + (0x200 * k)	0x420A 2058 + (0x200 * k)
EDMA_TPCC_IECRH_RN_k (3)	W	32	0x0000 205C + (0x200 * k)	0x420A 205C + (0x200 * k)
EDMA_TPCC_IESR_RN_k (3)	W	32	0x0000 2060 + (0x200 * k)	0x420A 2060 + (0x200 * k)
EDMA_TPCC_IESRH_RN_k (3)	W	32	0x0000 2064 + (0x200 * k)	0x420A 2064 + (0x200 * k)
EDMA_TPCC_IPR_RN_k (3)	R	32	0x0000 2068 + (0x200 * k)	0x420A 2068 + (0x200 * k)
EDMA_TPCC_IPRH_RN_k (3)	R	32	0x0000 206C + (0x200 * k)	0x420A 206C + (0x200 * k)
EDMA_TPCC_ICR_RN_k (3)	W	32	0x0000 2070 + (0x200 * k)	0x420A 2070 + (0x200 * k)
EDMA_TPCC_ICRH_RN_k (3)	W	32	0x0000 2074 + (0x200 * k)	0x420A 2074 + (0x200 * k)
EDMA_TPCC_IEVAL_RN_k (3)	W	32	0x0000 2078 + (0x200 * k)	0x420A 2078 + (0x200 * k)
EDMA_TPCC_QER_RN_k (3)	R	32	0x0000 2080 + (0x200 * k)	0x420A 2080 + (0x200 * k)
EDMA_TPCC_QEER_RN_k (3)	R	32	0x0000 2084 + (0x200 * k)	0x420A 2084 + (0x200 * k)
EDMA_TPCC_QEECR_RN_k (3)	W	32	0x0000 2088 + (0x200 * k)	0x420A 2088 + (0x200 * k)

Table 11-33. EVE EDMA_TPCC Registers Mapping Summary (L3_MAIN Access) (continued)

Register Name	Type	Register Width (Bits)	Address Offset	EVE_EDMA_TPCC Physical Address (L3_MAIN Access)
EDMA_TPCC_QEESR_RN_k (3)	W	32	0x0000 208C + (0x200 * k)	0x420A 208C + (0x200 * k)
EDMA_TPCC_QSER_RN_k (3)	R	32	0x0000 2090 + (0x200 * k)	0x420A 2090 + (0x200 * k)
EDMA_TPCC_QSECR_RN_k (3)	W	32	0x0000 2094 + (0x200 * k)	0x420A 2094 + (0x200 * k)
EDMA_TPCC_OPT_n (6)	RW	32	0x0000 4000 + (0x20 * n)	0x420A 4000 + (0x20 * n)
EDMA_TPCC_SRC_n (6)	RW	32	0x0000 4004 + (0x20 * n)	0x420A 4004 + (0x20 * n)
EDMA_TPCC_ABCNT_n (6)	RW	32	0x0000 4008 + (0x20 * n)	0x420A 4008 + (0x20 * n)
EDMA_TPCC_DST_n (6)	RW	32	0x0000 400C + (0x20 * n)	0x420A 400C + (0x20 * n)
EDMA_TPCC_BIDX_n (6)	RW	32	0x0000 4010 + (0x20 * n)	0x420A 4010 + (0x20 * n)
EDMA_TPCC_LNK_n (6)	RW	32	0x0000 4014 + (0x20 * n)	0x420A 4014 + (0x20 * n)
EDMA_TPCC_CIDX_n (6)	RW	32	0x0000 4018 + (0x20 * n)	0x420A 4018 + (0x20 * n)
EDMA_TPCC_CCNT_n (6)	RW	32	0x0000 401C + (0x20 * n)	0x420A 401C + (0x20 * n)

(6) n = 0 to 127 for EVE_EDMA_TPCC

NOTE: The value for "n" is from 0 to 1 in the Table 11-34. It corresponds of the Transfer Controller (EDMA_TPTC0 and EDMA_TPTC1) instances in the device.

Table 11-34. System EDMA_TPTC Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	SYS_EDMA_TPTC0 Physical Address	SYS_EDMA_TPTC1 Physical Address
EDMA_TPTCn_PID	R	32	0x0000 0000	0x4340 0000	0x4350 0000
EDMA_TPTCn_TCCFG	R	32	0x0000 0004	0x4340 0004	0x4350 0004
EDMA_TPTCn_TCSTAT	R	32	0x0000 0100	0x4340 0100	0x4350 0100
EDMA_TPTCn_INTSTAT	R	32	0x0000 0104	0x4340 0104	0x4350 0104
EDMA_TPTCn_INTEN	RW	32	0x0000 0108	0x4340 0108	0x4350 0108
EDMA_TPTCn_INTCLR	W	32	0x0000 010C	0x4340 010C	0x4350 010C
EDMA_TPTCn_INTCMD	W	32	0x0000 0110	0x4340 0110	0x4350 0110
EDMA_TPTCn_ERRSTAT	R	32	0x0000 0120	0x4340 0120	0x4350 0120
EDMA_TPTCn_ERREN	RW	32	0x0000 0124	0x4340 0124	0x4350 0124
EDMA_TPTCn_ERRCLR	W	32	0x0000 0128	0x4340 0128	0x4350 0128
EDMA_TPTCn_ERRDET	R	32	0x0000 012C	0x4340 012C	0x4350 012C
EDMA_TPTCn_ERRCMD	W	32	0x0000 0130	0x4340 0130	0x4350 0130
EDMA_TPTCn_RDRATE	RW	32	0x0000 0140	0x4340 0140	0x4350 0140
EDMA_TPTCn_POPT	RW	32	0x0000 0200	0x4340 0200	0x4350 0200
EDMA_TPTCn_PSRC	RW	32	0x0000 0204	0x4340 0204	0x4350 0204
EDMA_TPTCn_PCNT	RW	32	0x0000 0208	0x4340 0208	0x4350 0208
EDMA_TPTCn_PDST	RW	32	0x0000 020C	0x4340 020C	0x4350 020C
EDMA_TPTCn_PBIDX	RW	32	0x0000 0210	0x4340 0210	0x4350 0210
EDMA_TPTCn_PMPPRXY	R	32	0x0000 0214	0x4340 0214	0x4350 0214
EDMA_TPTCn_SAOPT	R	32	0x0000 0240	0x4340 0240	0x4350 0240
EDMA_TPTCn_SASRC	R	32	0x0000 0244	0x4340 0244	0x4350 0244
EDMA_TPTCn_SACNT	R	32	0x0000 0248	0x4340 0248	0x4350 0248
EDMA_TPTCn_SADST	R	32	0x0000 024C	0x4340 024C	0x4350 024C
EDMA_TPTCn_SABIDX	R	32	0x0000 0250	0x4340 0250	0x4350 0250
EDMA_TPTCn_SAMPPRXY	R	32	0x0000 0254	0x4340 0254	0x4350 0254

Table 11-34. System EDMA_TPTC Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	SYS_EDMA_TPTC0 Physical Address	SYS_EDMA_TPTC1 Physical Address
EDMA_TPTCn_SACNTRLD	R	32	0x0000 0258	0x4340 0258	0x4350 0258
EDMA_TPTCn_SASRCBREF	R	32	0x0000 025C	0x4340 025C	0x4350 025C
EDMA_TPTCn_SADSTBREF	R	32	0x0000 0260	0x4340 0260	0x4350 0260
EDMA_TPTCn_DFCNTRLD	R	32	0x0000 0280	0x4340 0280	0x4350 0280
EDMA_TPTCn_DFSRCBREF	R	32	0x0000 0284	0x4340 0284	0x4350 0284
EDMA_TPTCn_DFDSTBREF	R	32	0x0000 0288	0x4340 0288	0x4350 0288
EDMA_TPTCn_DFOPTi(1)	R	32	0x0000 0300 + (0x40 * i)	0x4340 0300 + (0x40 * i)	0x4350 0300 + (0x40 * i)
EDMA_TPTCn_DFSRCi (1)	R	32	0x0000 0304 + (0x40 * i)	0x4340 0304 + (0x40 * i)	0x4350 0304 + (0x40 * i)
EDMA_TPTCn_DFCNTi (1)	R	32	0x0000 0308 + (0x40 * i)	0x4340 0308 + (0x40 * i)	0x4350 0308 + (0x40 * i)
EDMA_TPTCn_DFDSTi (1)	R	32	0x0000 030C + (0x40 * i)	0x4340 030C + (0x40 * i)	0x4350 030C + (0x40 * i)
EDMA_TPTCn_DFBIDXi (1)	R	32	0x0000 0310 + (0x40 * i)	0x4340 0310 + (0x40 * i)	0x4350 0310 + (0x40 * i)
EDMA_TPTCn_DFMPPRXYi (1)	R	32	0x0000 0314 + (0x40 * i)	0x4340 0314 + (0x40 * i)	0x4350 0314 + (0x40 * i)

(1) i = 0 to 1 for SYS_EDMA_TPTC0 and SYS_EDMA_TPTC1

Table 11-35. DSP EDMA_TPTC0 Registers Mapping Summary (L3_MAIN Access)

Register Name	Type	Register Width (Bits)	Address Offset	DSP1_EDMA_TPTC0 Physical Address (L3_MAIN Access)	DSP2_EDMA_TPTC0 Physical Address (L3_MAIN Access)
EDMA_TPTCn_PID	R	32	0x0000 0000	0x40D0 5000	0x4150 5000
EDMA_TPTCn_TCCFG	R	32	0x0000 0004	0x40D0 5004	0x4150 5004
EDMA_TPTCn_TCSTAT	R	32	0x0000 0100	0x40D0 5100	0x4150 5100
EDMA_TPTCn_INTSTAT	R	32	0x0000 0104	0x40D0 5104	0x4150 5104
EDMA_TPTCn_INTEN	RW	32	0x0000 0108	0x40D0 5108	0x4150 5108
EDMA_TPTCn_INTCLR	W	32	0x0000 010C	0x40D0 510C	0x4150 510C
EDMA_TPTCn_INTCMD	W	32	0x0000 0110	0x40D0 5110	0x4150 5110
EDMA_TPTCn_ERRSTAT	R	32	0x0000 0120	0x40D0 5120	0x4150 5120
EDMA_TPTCn_ERREN	RW	32	0x0000 0124	0x40D0 5124	0x4150 5124
EDMA_TPTCn_ERRCLR	W	32	0x0000 0128	0x40D0 5128	0x4150 5128
EDMA_TPTCn_ERRDET	R	32	0x0000 012C	0x40D0 512C	0x4150 512C
EDMA_TPTCn_ERRCMD	W	32	0x0000 0130	0x40D0 5130	0x4150 5130
EDMA_TPTCn_RDRATE	RW	32	0x0000 0140	0x40D0 5140	0x4150 5140
EDMA_TPTCn_POPT	RW	32	0x0000 0200	0x40D0 5200	0x4150 5200
EDMA_TPTCn_PSRC	RW	32	0x0000 0204	0x40D0 5204	0x4150 5204
EDMA_TPTCn_PCNT	RW	32	0x0000 0208	0x40D0 5208	0x4150 5208
EDMA_TPTCn_PDST	RW	32	0x0000 020C	0x40D0 520C	0x4150 520C
EDMA_TPTCn_PBIDX	RW	32	0x0000 0210	0x40D0 5210	0x4150 5210
EDMA_TPTCn_PMPPRXY	R	32	0x0000 0214	0x40D0 5214	0x4150 5214
EDMA_TPTCn_SAOPT	R	32	0x0000 0240	0x40D0 5240	0x4150 5240
EDMA_TPTCn_SASRC	R	32	0x0000 0244	0x40D0 5244	0x4150 5244
EDMA_TPTCn_SACNT	R	32	0x0000 0248	0x40D0 5248	0x4150 5248
EDMA_TPTCn_SADST	R	32	0x0000 024C	0x40D0 524C	0x4150 524C
EDMA_TPTCn_SABIDX	R	32	0x0000 0250	0x40D0 5250	0x4150 5250

Table 11-35. DSP EDMA_TPTC0 Registers Mapping Summary (L3_MAIN Access) (continued)

Register Name	Type	Register Width (Bits)	Address Offset	DSP1_EDMA_TPTC0 Physical Address (L3_MAIN Access)	DSP2_EDMA_TPTC0 Physical Address (L3_MAIN Access)
EDMA_TPTCn_SAMPPRXY	R	32	0x0000 0254	0x40D0 5254	0x4150 5254
EDMA_TPTCn_SACNTRLD	R	32	0x0000 0258	0x40D0 5258	0x4150 5258
EDMA_TPTCn_SASRCBREF	R	32	0x0000 025C	0x40D0 525C	0x4150 525C
EDMA_TPTCn_SADSTBREF	R	32	0x0000 0260	0x40D0 5260	0x4150 5260
EDMA_TPTCn_DFCNTRLD	R	32	0x0000 0280	0x40D0 5280	0x4150 5280
EDMA_TPTCn_DFSRCBREF	R	32	0x0000 0284	0x40D0 5284	0x4150 5284
EDMA_TPTCn_DFDSTBREF	R	32	0x0000 0288	0x40D0 5288	0x4150 5288
EDMA_TPTCn_DFOPTi(1)	R	32	0x0000 0300 + (0x40 * i)	0x40D0 5300 + (0x40 * i)	0x4150 5300 + (0x40 * i)
EDMA_TPTCn_DFSRCi (1)	R	32	0x0000 0304 + (0x40 * i)	0x40D0 5304 + (0x40 * i)	0x4150 5304 + (0x40 * i)
EDMA_TPTCn_DFCNTi (1)	R	32	0x0000 0308 + (0x40 * i)	0x40D0 5308 + (0x40 * i)	0x4150 5308 + (0x40 * i)
EDMA_TPTCn_DFDSTi (1)	R	32	0x0000 030C + (0x40 * i)	0x40D0 530C + (0x40 * i)	0x4150 530C + (0x40 * i)
EDMA_TPTCn_DFBIDXi (1)	R	32	0x0000 0310 + (0x40 * i)	0x40D0 5310 + (0x40 * i)	0x4150 5310 + (0x40 * i)
EDMA_TPTCn_DFMPPRXYi (1)	R	32	0x0000 0314 + (0x40 * i)	0x40D0 5314 + (0x40 * i)	0x4150 5314 + (0x40 * i)

(1) i = 0 to 1 for DSP1_EDMA_TPTC0 and DSP2_EDMA_TPTC0

Table 11-36. DSP EDMA_TPTC1 Registers Mapping Summary (L3_MAIN Access)

Register Name	Type	Register Width (Bits)	Address Offset	DSP1_EDMA_TPTC1 Physical Address (L3_MAIN Access)	DSP2_EDMA_TPTC1 Physical Address (L3_MAIN Access)
EDMA_TPTCn_PID	R	32	0x0000 0000	0x40D0 6000	0x4150 6000
EDMA_TPTCn_TCCFG	R	32	0x0000 0004	0x40D0 6004	0x4150 6004
EDMA_TPTCn_TCSTAT	R	32	0x0000 0100	0x40D0 6100	0x4150 6100
EDMA_TPTCn_INTSTAT	R	32	0x0000 0104	0x40D0 6104	0x4150 6104
EDMA_TPTCn_INTEN	RW	32	0x0000 0108	0x40D0 6108	0x4150 6108
EDMA_TPTCn_INTCLR	W	32	0x0000 010C	0x40D0 610C	0x4150 610C
EDMA_TPTCn_INTCMD	W	32	0x0000 0110	0x40D0 6110	0x4150 6110
EDMA_TPTCn_ERRSTAT	R	32	0x0000 0120	0x40D0 6120	0x4150 6120
EDMA_TPTCn_ERREN	RW	32	0x0000 0124	0x40D0 6124	0x4150 6124
EDMA_TPTCn_ERRCLR	W	32	0x0000 0128	0x40D0 6128	0x4150 6128
EDMA_TPTCn_ERRDET	R	32	0x0000 012C	0x40D0 612C	0x4150 612C
EDMA_TPTCn_ERRCMD	W	32	0x0000 0130	0x40D0 6130	0x4150 6130
EDMA_TPTCn_RDRATE	RW	32	0x0000 0140	0x40D0 6140	0x4150 6140
EDMA_TPTCn_POPT	RW	32	0x0000 0200	0x40D0 6200	0x4150 6200
EDMA_TPTCn_PSRC	RW	32	0x0000 0204	0x40D0 6204	0x4150 6204
EDMA_TPTCn_PCNT	RW	32	0x0000 0208	0x40D0 6208	0x4150 6208
EDMA_TPTCn_PDST	RW	32	0x0000 020C	0x40D0 620C	0x4150 620C
EDMA_TPTCn_PBIDX	RW	32	0x0000 0210	0x40D0 6210	0x4150 6210
EDMA_TPTCn_PMPPRXY	R	32	0x0000 0214	0x40D0 6214	0x4150 6214
EDMA_TPTCn_SAOPT	R	32	0x0000 0240	0x40D0 6240	0x4150 6240
EDMA_TPTCn_SASRC	R	32	0x0000 0244	0x40D0 6244	0x4150 6244
EDMA_TPTCn_SACNT	R	32	0x0000 0248	0x40D0 6248	0x4150 6248
EDMA_TPTCn_SADST	R	32	0x0000 024C	0x40D0 624C	0x4150 624C

Table 11-36. DSP EDMA_TPTC1 Registers Mapping Summary (L3_MAIN Access) (continued)

Register Name	Type	Register Width (Bits)	Address Offset	DSP1_EDMA_TPTC1 Physical Address (L3_MAIN Access)	DSP2_EDMA_TPTC1 Physical Address (L3_MAIN Access)
EDMA_TPTCn_SABIDX	R	32	0x0000 0250	0x40D0 6250	0x4150 6250
EDMA_TPTCn_SAMPPRXY	R	32	0x0000 0254	0x40D0 6254	0x4150 6254
EDMA_TPTCn_SACNTRLD	R	32	0x0000 0258	0x40D0 6258	0x4150 6258
EDMA_TPTCn_SASRCBREF	R	32	0x0000 025C	0x40D0 625C	0x4150 625C
EDMA_TPTCn_SADSTBREF	R	32	0x0000 0260	0x40D0 6260	0x4150 6260
EDMA_TPTCn_DFCNTRLD	R	32	0x0000 0280	0x40D0 6280	0x4150 6280
EDMA_TPTCn_DFSRCBREF	R	32	0x0000 0284	0x40D0 6284	0x4150 6284
EDMA_TPTCn_DFDSTBREF	R	32	0x0000 0288	0x40D0 6288	0x4150 6288
EDMA_TPTCn_DFOPTi(1)	R	32	0x0000 0300 + (0x40 * i)	0x40D0 6300 + (0x40 * i)	0x4150 6300 + (0x40 * i)
EDMA_TPTCn_DFSRCi (1)	R	32	0x0000 0304 + (0x40 * i)	0x40D0 6304 + (0x40 * i)	0x4150 6304 + (0x40 * i)
EDMA_TPTCn_DFCNTi (1)	R	32	0x0000 0308 + (0x40 * i)	0x40D0 6308 + (0x40 * i)	0x4150 6308 + (0x40 * i)
EDMA_TPTCn_DFDSTi (1)	R	32	0x0000 030C + (0x40 * i)	0x40D0 630C + (0x40 * i)	0x4150 630C + (0x40 * i)
EDMA_TPTCn_DFBIDXi (1)	R	32	0x0000 0310 + (0x40 * i)	0x40D0 6310 + (0x40 * i)	0x4150 6310 + (0x40 * i)
EDMA_TPTCn_DFMPPRXYi (1)	R	32	0x0000 0314 + (0x40 * i)	0x40D0 6314 + (0x40 * i)	0x4150 6314 + (0x40 * i)

(1) i = 0 to 1 for DSP1_EDMA_TPTC1 and DSP2_EDMA_TPTC1

Table 11-37. DSP EDMA_TPTC0 and EDMA_TPTC1 Registers Mapping Summary (Private Access)

Register Name	Type	Register Width (Bits)	Address Offset	DSP_EDMA_TPTC0 Physical Address (DSP Private Access)	DSP_EDMA_TPTC1 Physical Address (DSP Private Access)
EDMA_TPTCn_PID	R	32	0x0000 0000	0x01D0 5000	0x01D0 6000
EDMA_TPTCn_TCCFG	R	32	0x0000 0004	0x01D0 5004	0x01D0 6004
EDMA_TPTCn_TCSTAT	R	32	0x0000 0100	0x01D0 5100	0x01D0 6100
EDMA_TPTCn_INTSTAT	R	32	0x0000 0104	0x01D0 5104	0x01D0 6104
EDMA_TPTCn_INTEN	RW	32	0x0000 0108	0x01D0 5108	0x01D0 6108
EDMA_TPTCn_INTCLR	W	32	0x0000 010C	0x01D0 510C	0x01D0 610C
EDMA_TPTCn_INTCMD	W	32	0x0000 0110	0x01D0 5110	0x01D0 6110
EDMA_TPTCn_ERRSTAT	R	32	0x0000 0120	0x01D0 5120	0x01D0 6120
EDMA_TPTCn_ERREN	RW	32	0x0000 0124	0x01D0 5124	0x01D0 6124
EDMA_TPTCn_ERRCLR	W	32	0x0000 0128	0x01D0 5128	0x01D0 6128
EDMA_TPTCn_ERRDET	R	32	0x0000 012C	0x01D0 512C	0x01D0 612C
EDMA_TPTCn_ERRCMD	W	32	0x0000 0130	0x01D0 5130	0x01D0 6130
EDMA_TPTCn_RDRATE	RW	32	0x0000 0140	0x01D0 5140	0x01D0 6140
EDMA_TPTCn_POPT	RW	32	0x0000 0200	0x01D0 5200	0x01D0 6200
EDMA_TPTCn_PSRC	RW	32	0x0000 0204	0x01D0 5204	0x01D0 6204
EDMA_TPTCn_PCNT	RW	32	0x0000 0208	0x01D0 5208	0x01D0 6208
EDMA_TPTCn_PDST	RW	32	0x0000 020C	0x01D0 520C	0x01D0 620C
EDMA_TPTCn_PBIDX	RW	32	0x0000 0210	0x01D0 5210	0x01D0 6210
EDMA_TPTCn_PMPPRXY	R	32	0x0000 0214	0x01D0 5214	0x01D0 6214
EDMA_TPTCn_SAOPT	R	32	0x0000 0240	0x01D0 5240	0x01D0 6240
EDMA_TPTCn_SASRC	R	32	0x0000 0244	0x01D0 5244	0x01D0 6244
EDMA_TPTCn_SACNT	R	32	0x0000 0248	0x01D0 5248	0x01D0 6248

Table 11-37. DSP EDMA_TPTC0 and EDMA_TPTC1 Registers Mapping Summary (Private Access) (continued)

Register Name	Type	Register Width (Bits)	Address Offset	DSP_EDMA_TPTC0 Physical Address (DSP Private Access)	DSP_EDMA_TPTC1 Physical Address (DSP Private Access)
EDMA_TPTCn_SADST	R	32	0x0000 024C	0x01D0 524C	0x01D0 624C
EDMA_TPTCn_SABIDX	R	32	0x0000 0250	0x01D0 5250	0x01D0 6250
EDMA_TPTCn_SAMPPRXY	R	32	0x0000 0254	0x01D0 5254	0x01D0 6254
EDMA_TPTCn_SACNTRLD	R	32	0x0000 0258	0x01D0 5258	0x01D0 6258
EDMA_TPTCn_SASRCBREF	R	32	0x0000 025C	0x01D0 525C	0x01D0 625C
EDMA_TPTCn_SADSTBREF	R	32	0x0000 0260	0x01D0 5260	0x01D0 6260
EDMA_TPTCn_DFCNTRLD	R	32	0x0000 0280	0x01D0 5280	0x01D0 6280
EDMA_TPTCn_DFSRCBREF	R	32	0x0000 0284	0x01D0 5284	0x01D0 6284
EDMA_TPTCn_DFDSTBREF	R	32	0x0000 0288	0x01D0 5288	0x01D0 6288
EDMA_TPTCn_DFOPTi(1)	R	32	0x0000 0300 + (0x40 * i)	0x01D0 5300 + (0x40 * i)	0x01D0 6300 + (0x40 * i)
EDMA_TPTCn_DFSRCi (1)	R	32	0x0000 0304 + (0x40 * i)	0x01D0 5304 + (0x40 * i)	0x01D0 6304 + (0x40 * i)
EDMA_TPTCn_DFCNTi (1)	R	32	0x0000 0308 + (0x40 * i)	0x01D0 5308 + (0x40 * i)	0x01D0 6308 + (0x40 * i)
EDMA_TPTCn_DFDSTi (1)	R	32	0x0000 030C + (0x40 * i)	0x01D0 530C + (0x40 * i)	0x01D0 630C + (0x40 * i)
EDMA_TPTCn_DFBIDXi (1)	R	32	0x0000 0310 + (0x40 * i)	0x01D0 5310 + (0x40 * i)	0x01D0 6310 + (0x40 * i)
EDMA_TPTCn_DFMPPRXYi (1)	R	32	0x0000 0314 + (0x40 * i)	0x01D0 5314 + (0x40 * i)	0x01D0 6314 + (0x40 * i)

(1) i = 0 to 1 for DSP_EDMA_TPTC0 and DSP_EDMA_TPTC1

Table 11-38. EVE EDMA_TPTC0 Registers Mapping Summary (L3_MAIN Access)

Register Name	Type	Register Width (Bits)	Address Offset	EVE_EDMA_TPTC0 Physical Address (L3_MAIN Access)
EDMA_TPTCn_PID	R	32	0x0000 0000	-
EDMA_TPTCn_TCCFG	R	32	0x0000 0004	-
EDMA_TPTCn_TCSTAT	R	32	0x0000 0100	0x4208 6100
EDMA_TPTCn_INTSTAT	R	32	0x0000 0104	0x4208 6104
EDMA_TPTCn_INTEN	RW	32	0x0000 0108	0x4208 6108
EDMA_TPTCn_INTCLR	W	32	0x0000 010C	0x4208 610C
EDMA_TPTCn_INTCMD	W	32	0x0000 0110	0x4208 6110
EDMA_TPTCn_ERRSTAT	R	32	0x0000 0120	0x4208 6120
EDMA_TPTCn_ERREN	RW	32	0x0000 0124	0x4208 6124
EDMA_TPTCn_ERRCLR	W	32	0x0000 0128	0x4208 6128
EDMA_TPTCn_ERRDET	R	32	0x0000 012C	0x4208 612C
EDMA_TPTCn_ERRCMD	W	32	0x0000 0130	0x4208 6130
EDMA_TPTCn_RDRATE	RW	32	0x0000 0140	0x4208 6140
EDMA_TPTCn_POPT	RW	32	0x0000 0200	0x4208 6200
EDMA_TPTCn_PSRC	RW	32	0x0000 0204	0x4208 6204
EDMA_TPTCn_PCNT	RW	32	0x0000 0208	0x4208 6208
EDMA_TPTCn_PDST	RW	32	0x0000 020C	0x4208 620C
EDMA_TPTCn_PBIDX	RW	32	0x0000 0210	0x4208 6210
EDMA_TPTCn_PMPPRXY	R	32	0x0000 0214	0x4208 6214
EDMA_TPTCn_SAOPT	R	32	0x0000 0240	0x4208 6240

Table 11-38. EVE EDMA_TPTC0 Registers Mapping Summary (L3_MAIN Access) (continued)

Register Name	Type	Register Width (Bits)	Address Offset	EVE_EDMA_TPTC0 Physical Address (L3_MAIN Access)
EDMA_TPTCn_SASRC	R	32	0x0000 0244	0x4208 6244
EDMA_TPTCn_SACNT	R	32	0x0000 0248	0x4208 6248
EDMA_TPTCn_SADST	R	32	0x0000 024C	-
EDMA_TPTCn_SABIDX	R	32	0x0000 0250	0x4208 6250
EDMA_TPTCn_SAMPPRXY	R	32	0x0000 0254	0x4208 6254
EDMA_TPTCn_SACNTRLD	R	32	0x0000 0258	0x4208 6258
EDMA_TPTCn_SASRCBREF	R	32	0x0000 025C	0x4208 625C
EDMA_TPTCn_SADSTBREF	R	32	0x0000 0260	0x4208 6260
EDMA_TPTCn_DFCNTRLD	R	32	0x0000 0280	0x4208 6280
EDMA_TPTCn_DFSRCBREF	R	32	0x0000 0284	0x4208 6284
EDMA_TPTCn_DFDSTBREF	R	32	0x0000 0288	-
EDMA_TPTCn_DFOPTi(1)	R	32	0x0000 0300 + (0x40 * i)	0x4208 6300 + (0x40 * i)
EDMA_TPTCn_DFSRCi (1)	R	32	0x0000 0304 + (0x40 * i)	0x4208 6304 + (0x40 * i)
EDMA_TPTCn_DFCNTi (1)	R	32	0x0000 0308 + (0x40 * i)	0x4208 6308 + (0x40 * i)
EDMA_TPTCn_DFDSTi (1)	R	32	0x0000 030C + (0x40 * i)	0x4208 630C + (0x40 * i)
EDMA_TPTCn_DFBIDXi (1)	R	32	0x0000 0310 + (0x40 * i)	0x4208 6310 + (0x40 * i)
EDMA_TPTCn_DFMPPRXYi (1)	R	32	0x0000 0314 + (0x40 * i)	0x4208 6314 + (0x40 * i)

(1) i = 0 to 1 for EVE_EDMA_TPTC0

Table 11-39. EVE EDMA_TPTC1 Registers Mapping Summary (L3_MAIN Access)

Register Name	Type	Register Width (Bits)	Address Offset	EVE_EDMA_TPTC1 Physical Address (L3_MAIN Access)
EDMA_TPTCn_PID	R	32	0x0000 0000	-
EDMA_TPTCn_TCCFG	R	32	0x0000 0004	-
EDMA_TPTCn_TCSTAT	R	32	0x0000 0100	0x4208 7100
EDMA_TPTCn_INTSTAT	R	32	0x0000 0104	0x4208 7104
EDMA_TPTCn_INTEN	RW	32	0x0000 0108	0x4208 7108
EDMA_TPTCn_INTCLR	W	32	0x0000 010C	0x4208 710C
EDMA_TPTCn_INTCMD	W	32	0x0000 0110	0x4208 7110
EDMA_TPTCn_ERRSTAT	R	32	0x0000 0120	0x4208 7120
EDMA_TPTCn_ERREN	RW	32	0x0000 0124	0x4208 7124
EDMA_TPTCn_ERRCLR	W	32	0x0000 0128	0x4208 7128
EDMA_TPTCn_ERRDET	R	32	0x0000 012C	0x4208 712C
EDMA_TPTCn_ERRCMD	W	32	0x0000 0130	0x4208 7130
EDMA_TPTCn_RDRATE	RW	32	0x0000 0140	0x4208 7140
EDMA_TPTCn_POPT	RW	32	0x0000 0200	0x4208 7200
EDMA_TPTCn_PSRC	RW	32	0x0000 0204	0x4208 7204
EDMA_TPTCn_PCNT	RW	32	0x0000 0208	0x4208 7208
EDMA_TPTCn_PDST	RW	32	0x0000 020C	0x4208 720C
EDMA_TPTCn_PBIDX	RW	32	0x0000 0210	0x4208 7210
EDMA_TPTCn_PMPPRXY	R	32	0x0000 0214	0x4208 7214
EDMA_TPTCn_SAOPT	R	32	0x0000 0240	0x4208 7240
EDMA_TPTCn_SASRC	R	32	0x0000 0244	0x4208 7244
EDMA_TPTCn_SACNT	R	32	0x0000 0248	0x4208 7248
EDMA_TPTCn_SADST	R	32	0x0000 024C	-

Table 11-39. EVE EDMA_TPTC1 Registers Mapping Summary (L3_MAIN Access) (continued)

Register Name	Type	Register Width (Bits)	Address Offset	EVE_EDMA_TPTC1 Physical Address (L3_MAIN Access)
EDMA_TPTCn_SABIDX	R	32	0x0000 0250	0x4208 7250
EDMA_TPTCn_SAMPPRXY	R	32	0x0000 0254	0x4208 7254
EDMA_TPTCn_SACNTRLD	R	32	0x0000 0258	0x4208 7258
EDMA_TPTCn_SASRCBREF	R	32	0x0000 025C	0x4208 725C
EDMA_TPTCn_SADSTBREF	R	32	0x0000 0260	0x4208 7260
EDMA_TPTCn_DFCNTRLD	R	32	0x0000 0280	0x4208 7280
EDMA_TPTCn_DFSRCBREF	R	32	0x0000 0284	0x4208 7284
EDMA_TPTCn_DFDSTBREF	R	32	0x0000 0288	-
EDMA_TPTCn_DFOPTi(1)	R	32	0x0000 0300 + (0x40 * i)	0x4208 7300 + (0x40 * i)
EDMA_TPTCn_DFSRCi (1)	R	32	0x0000 0304 + (0x40 * i)	0x4208 7304 + (0x40 * i)
EDMA_TPTCn_DFCNTi (1)	R	32	0x0000 0308 + (0x40 * i)	0x4208 7308 + (0x40 * i)
EDMA_TPTCn_DFDSTi (1)	R	32	0x0000 030C + (0x40 * i)	0x4208 730C + (0x40 * i)
EDMA_TPTCn_DFBIDXi (1)	R	32	0x0000 0310 + (0x40 * i)	0x4208 7310 + (0x40 * i)
EDMA_TPTCn_DFMPPRXYi (1)	R	32	0x0000 0314 + (0x40 * i)	0x4208 7314 + (0x40 * i)

(1) i = 0 to 1 for EVE_EDMA_TPTC1

11.7.2.2 EDMA Register Description

11.7.2.2.1 EDMA_TPCC Register Description

Table 11-40 through Table 11-264 describe the EDMA_TPCC module registers.

Table 11-40. EDMA_TPCC_PID

Address Offset	0x0000 0000		
Physical Address	0x4330 0000 0x40D1 0000 0x4151 0000 0x01D1 0000 0x420A 0000	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE_EDMA_TPCC
Description	Peripheral ID Register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REVISION																															

Bits	Field Name	Description	Type	Reset
31:0	REVISION	IP revision	R	TI internal data

Table 11-41. Register Call Summary for Register EDMA_TPCC_PID

EDMA Register Manual
- EDMA Register Summary: [0][1][2][3]

Table 11-42. EDMA_TPCC_CCCFG

Address Offset	0x0000 0004		
Physical Address	0x4330 0004 0x40D1 0004 0x4151 0004 0x01D1 0004 0x420A 0004	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE_EDMA_TPCC
Description	CC Configuration Register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED						MPEXIST	CHMAPEXIST	RESERVED		NUMREGN		RESERVED	NUMTC			RESERVED	NUMPAENTRY			RESERVED	NUMINTCH			RESERVED	NUMQDMACH			RESERVED	NUMDMACH		

Bits	Field Name	Description	Type	Reset
31:26	RESERVED	Reads return 0's	R	0x0
25	MPEXIST	Memory Protection Existence 0x0: No memory protection 0x1: Memory Protection logic included	R	See Table 11-1
24	CHMAPEXIST	Channel Mapping Existence 0x0: No Channel mapping 0x1: Channel mapping logic included	R	See Table 11-1
23:22	RESERVED	Reads return 0's	R	0x0
21:20	NUMREGN	Number of MP and Shadow regions 0x0: 0 Regions 0x1: 2 Regions 0x2: 4 Regions 0x3: 8 Regions	R	See Table 11-1
19	RESERVED	Reads return 0's	R	0x0
18:16	NUMTC	Number of Queues/Number of TCs 0x0: 1 TC/Event Queue 0x1: 2 TC/Event Queue 0x2: 3 TC/Event Queue 0x3: 4 TC/Event Queue 0x4: 5 TC/Event Queue 0x5: 6 TC/Event Queue 0x6: 7 TC/Event Queue 0x7: 8 TC/Event Queue	R	See Table 11-1
15	RESERVED	Reads return 0's	R	0x0
14:12	NUMPAENTRY	Number of PaRAM entries 0x0: 16 entries 0x1: 32 entries 0x2: 64 entries 0x3: 128 entries 0x4: 256 entries 0x5: 512 entries	R	See Table 11-1
11	RESERVED	Reads return 0's	R	0x0

Bits	Field Name	Description	Type	Reset
10:8	NUMINTCH	Number of Interrupt Channels 0x1: 8 Interrupt channels 0x2: 16 Interrupt channels 0x3: 32 Interrupt channels 0x4: 64 Interrupt channels	R	See Table 11-1
7	RESERVED	reads return 0's	R	0x0
6:4	NUMQDMACH	Number of QDMA Channels 0x0: No QDMA Channels 0x1: 2 QDMA Channels 0x2: 4 QDMA Channels 0x3: 6 QDMA Channels 0x4: 8 QDMA Channels	R	See Table 11-1
3	RESERVED	reads return 0's	R	0x0
2:0	NUMDMACH	Number of DMA Channels 0x0: No DMA Channels 0x1: 4 DMA Channels 0x2: 8 DMA Channels 0x3: 16 DMA Channels 0x4: 32 DMA Channels 0x5: 64 DMA Channels	R	See Table 11-1

Table 11-43. Register Call Summary for Register EDMA_TPCC_CCCFG

EDMA Register Manual
- EDMA Register Summary: [0][1][2][3]

Table 11-44. EDMA_TPCC_CLKGDIS

Address Offset	0x0000 00FC		
Physical Address	0x4330 00FC 0x40D1 00FC 0x4151 00FC 0x01D1 00FC 0x420A 00FC	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE_EDMA_TPCC
Description	Auto Clock Gate Disable		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															CLKGDIS

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved	R	0x0
0	CLKGDIS	Auto Clock Gate Disable	RW	0x0

Table 11-45. Register Call Summary for Register EDMA_TPCC_CLKGDIS

EDMA Controller Functional Description
- Clock and Power Management: [0]

EDMA Register Manual
- EDMA Register Summary: [1][2][3][4]

Table 11-46. EDMA_TPCC_DCHMAPN_m

Address Offset	0x0000 0100 + (0x4 * m)		
Physical Address	0x4330 0100 + (0x4 * m) 0x40D1 0100 + (0x4 * m) 0x4151 0100 + (0x4 * m) 0x01D1 0100 + (0x4 * m) 0x420A 0100 + (0x4 * m)	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE_EDMA_TPCC
Description	DMA Channel N Mapping Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																PAENTRY								RESERVED							

Bits	Field Name	Description	Type	Reset
31:14	RESERVED	Reserved	R	0x0
13:5	PAENTRY	PaRAM Entry number for DMA Channel N.	RW	0x0
4:0	RESERVED	Reserved	R	0x0

Table 11-47. Register Call Summary for Register EDMA_TPCC_DCHMAPN_m

EDMA Controller Functional Description
- Parameter RAM (PaRAM): [0]
- DMA Channel to PaRAM Mapping: [1][2][3]

EDMA Transfer Examples
- Setting Up an EDMA Transfer: [4]

EDMA Register Manual
- EDMA Register Summary: [5][6][7][8]

Table 11-48. EDMA_TPCC_QCHMAPN_j

Address Offset	0x0000 0200 + (0x4 * j)		
Physical Address	0x4330 0200 + (0x4 * j) 0x40D1 0200 + (0x4 * j) 0x4151 0200 + (0x4 * j) 0x01D1 0200 + (0x4 * j) 0x420A 0200 + (0x4 * j)	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE_EDMA_TPCC
Description	QDMA Channel N Mapping Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																PAENTRY								TRWORD					RESERVED		

Bits	Field Name	Description	Type	Reset
31:14	RESERVED	Reserved	R	0x0
13:5	PAENTRY	PaRAM Entry number for QDMA Channel N.	RW	0x0
4:2	TRWORD	TRWORD points to the specific trigger word of the PaRAM Entry defined by PAENTRY. A write to the trigger word results in a QDMA Event being recognized.	RW	0x0
1:0	RESERVED	Reserved	R	0x0

Table 11-49. Register Call Summary for Register EDMA_TPCC_QCHMAPN_j

EDMA Controller Functional Description • Parameter RAM (PaRAM): [0] • Linking Transfers: [1] • QDMA Channels: [2][3] • QDMA Channel to PaRAM Mapping: [4][5][6][7]
EDMA Transfer Examples • Setting Up an EDMA Transfer: [8]
EDMA Register Manual • EDMA Register Summary: [9][10][11][12]

Table 11-50. EDMA_TPCC_DMAQNUMN_k

Address Offset	0x0000 0240 + (0x4 * k)		
Physical Address	0x4330 0240 + (0x4 * k) 0x40D1 0240 + (0x4 * k) 0x4151 0240 + (0x4 * k) 0x01D1 0240 + (0x4 * k) 0x420A 0240 + (0x4 * k)	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE_EDMA_TPCC
Description	DMA Queue Number Register n Contains the Event queue number to be used for the corresponding DMA Channel.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	E7			RESERVED	E6			RESERVED	E5			RESERVED	E4			RESERVED	E3			RESERVED	E2			RESERVED	E1			RESERVED	E0		

Bits	Field Name	Description	Type	Reset
31	RESERVED	Reserved	R	0x0
30:28	E7	DMA Queue Number for event #7	RW	0x0
27	RESERVED	Reserved	R	0x0
26:24	E6	DMA Queue Number for event #6	RW	0x0
23	RESERVED	Reserved	R	0x0
22:20	E5	DMA Queue Number for event #5	RW	0x0
19	RESERVED	Reserved	R	0x0
18:16	E4	DMA Queue Number for event #4	RW	0x0
15	RESERVED	Reserved	R	0x0
14:12	E3	DMA Queue Number for event #3	RW	0x0
11	RESERVED	Reserved	R	0x0
10:8	E2	DMA Queue Number for event #2	RW	0x0
7	RESERVED	Reserved	R	0x0
6:4	E1	DMA Queue Number for event #1	RW	0x0
3	RESERVED	Reserved	R	0x0
2:0	E0	DMA Queue Number for event #0	RW	0x0

Table 11-51. Register Call Summary for Register EDMA_TPCC_DMAQNUMN_k

EDMA Controller Functional Description • DMA/QDMA Channel to Event Queue Mapping: [0] • Dequeue Priority: [1]

Table 11-51. Register Call Summary for Register EDMA_TPCC_DMAQNUMN_k (continued)

EDMA Transfer Examples • Setting Up an EDMA Transfer: [2]
EDMA Register Manual • EDMA Register Summary: [3][4][5][6]

Table 11-52. EDMA_TPCC_QDMAQNUM

Address Offset	0x0000 0260		
Physical Address	0x4330 0260 0x40D1 0260 0x4151 0260 0x01D1 0260 0x420A 0260	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE_EDMA_TPCC
Description	QDMA Queue Number Register Contains the Event queue number to be used for the corresponding QDMA Channel.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	E7			RESERVED	E6			RESERVED	E5			RESERVED	E4			RESERVED	E3			RESERVED	E2			RESERVED	E1			RESERVED	E0		

Bits	Field Name	Description	Type	Reset
31	RESERVED		R	0x0
30:28	E7	QDMA Queue Number for event #7	RW	0x0
27	RESERVED		R	0x0
26:24	E6	QDMA Queue Number for event #6	RW	0x0
23	RESERVED		R	0x0
22:20	E5	QDMA Queue Number for event #5	RW	0x0
19	RESERVED		R	0x0
18:16	E4	QDMA Queue Number for event #4	RW	0x0
15	RESERVED		R	0x0
14:12	E3	QDMA Queue Number for event #3	RW	0x0
11	RESERVED		R	0x0
10:8	E2	QDMA Queue Number for event #2	RW	0x0
7	RESERVED		R	0x0
6:4	E1	QDMA Queue Number for event #1	RW	0x0
3	RESERVED		R	0x0
2:0	E0	QDMA Queue Number for event #0	RW	0x0

Table 11-53. Register Call Summary for Register EDMA_TPCC_QDMAQNUM

EDMA Controller Functional Description • DMA/QDMA Channel to Event Queue Mapping: [0] • Dequeue Priority: [1]
EDMA Transfer Examples • Setting Up an EDMA Transfer: [2]
EDMA Register Manual • EDMA Register Summary: [3][4][5][6]

Table 11-54. EDMA_TPCC_QUETCMAP

Address Offset	0x0000 0280		
Physical Address	0x4330 0280 0x40D1 0280 0x4151 0280 0x01D1 0280 0x420A 0280	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE_EDMA_TPCC
Description	Queue to TC Mapping		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								TCNUMQ1				RESERVED	TCNUMQ0		

Bits	Field Name	Description	Type	Reset
31:7	RESERVED	Reserved	R	0x0
6:4	TCNUMQ1	TC Number for Queue N: Defines the TC number that Event Queue N TRs are written to.	RW	0x1
3	RESERVED	Reserved	R	0x0
2:0	TCNUMQ0	TC Number for Queue N: Defines the TC number that Event Queue N TRs are written to.	RW	0x0

Table 11-55. Register Call Summary for Register EDMA_TPCC_QUETCMAP

EDMA Register Manual

- EDMA Register Summary: [0][1][2][3]

Table 11-56. EDMA_TPCC_QUEPRI

Address Offset	0x0000 0284		
Physical Address	0x4330 0284 0x40D1 0284 0x4151 0284 0x01D1 0284 0x420A 0284	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE_EDMA_TPCC
Description	Queue Priority		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								PRIQ1				RESERVED	PRIQ0		

Bits	Field Name	Description	Type	Reset
31:7	RESERVED	Reserved	R	0x0
6:4	PRIQ1	Priority Level for Queue 1 Dictates the priority level used for the OPTIONS field programmation for Qn TRs. Sets the priority used for TC read and write commands.	RW	0x0
3	RESERVED	Reserved	R	0x0
2:0	PRIQ0	Priority Level for Queue 0 Dictates the priority level used for the OPTIONS field programmation for Qn TRs. Sets the priority used for TC read and write commands.	RW	0x0

Table 11-57. Register Call Summary for Register EDMA_TPCC_QUEPRI

EDMA Controller Functional Description

- Performance Considerations: [0]

EDMA Register Manual

- EDMA Register Summary: [1][2][3][4]

Table 11-58. EDMA_TPCC_EMR

Address Offset	0x0000 0300		
Physical Address	0x4330 0300 0x40D1 0300 0x4151 0300 0x01D1 0300 0x420A 0300	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE_EDMA_TPCC
Description	Event Missed Register: The Event Missed register is set if 2 events are received without the first event being cleared or if a Null TR is serviced. Chained events (CER), Set Events (ESR), and normal events (ER) are treated individually. If any bit in the EMR register is set (and all errors (including EDMA_TPCC_QEMR / EDMA_TPCC_CCERR) were previously clear), then an error will be signaled with TPCC error interrupt.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E31	E30	E29	E28	E27	E26	E25	E24	E23	E22	E21	E20	E19	E18	E17	E16	E15	E14	E13	E12	E11	E10	E9	E8	E7	E6	E5	E4	E3	E2	E1	E0

Bits	Field Name	Description	Type	Reset
31	E31	Event Missed #31	R	0x0
30	E30	Event Missed #30	R	0x0
29	E29	Event Missed #29	R	0x0
28	E28	Event Missed #28	R	0x0
27	E27	Event Missed #27	R	0x0
26	E26	Event Missed #26	R	0x0
25	E25	Event Missed #25	R	0x0
24	E24	Event Missed #24	R	0x0
23	E23	Event Missed #23	R	0x0
22	E22	Event Missed #22	R	0x0
21	E21	Event Missed #21	R	0x0
20	E20	Event Missed #20	R	0x0
19	E19	Event Missed #19	R	0x0
18	E18	Event Missed #18	R	0x0
17	E17	Event Missed #17	R	0x0
16	E16	Event Missed #16	R	0x0
15	E15	Event Missed #15	R	0x0
14	E14	Event Missed #14	R	0x0
13	E13	Event Missed #13	R	0x0
12	E12	Event Missed #12	R	0x0
11	E11	Event Missed #11	R	0x0
10	E10	Event Missed #10	R	0x0
9	E9	Event Missed #9	R	0x0
8	E8	Event Missed #8	R	0x0
7	E7	Event Missed #7	R	0x0
6	E6	Event Missed #6	R	0x0
5	E5	Event Missed #5	R	0x0

Bits	Field Name	Description	Type	Reset
4	E4	Event Missed #4	R	0x0
3	E3	Event Missed #3	R	0x0
2	E2	Event Missed #2	R	0x0
1	E1	Event Missed #1	R	0x0
0	E0	Event Missed #0	R	0x0

Table 11-59. Register Call Summary for Register EDMA_TPCC_EMR

EDMA Controller Functional Description
- Third-Party Channel Controller: [0]
- Null PaRAM Set: [1]
- Dummy PaRAM Set: [2]
- Dummy Versus Null Transfer Comparison: [3][4]
- DMA Channels: [5][6][7][8][9][10]
- Interrupt Evaluation Operations: [11]
- Error Interrupts: [12]

EDMA Debug Checklist and Programming Tips
- EDMA Debug Checklist: [13][14]

EDMA Register Manual
- EDMA Register Summary: [15][16][17][18]
- EDMA Register Description: [19][20][21][22][23][24][25][26][27][28][29][30][31]

Table 11-60. EDMA_TPCC_EMRH

Address Offset	0x0000 0304		
Physical Address	0x4330 0304 0x40D1 0304 0x4151 0304 0x01D1 0304 0x420A 0304	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE_EDMA_TPCC
Description	Event Missed Register (High Part): The Event Missed register is set if 2 events are received without the first event being cleared or if a Null TR is serviced. Chained events (CER), Set Events (ESR), and normal events (ER) are treated individually. If any bit in the EMR register is set (and all errors (including EDMA_TPCC_QEMR / EDMA_TPCC_CCERR) were previously clear), then an error will be signaled with TPCC error interrupt.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E63	E62	E61	E60	E59	E58	E57	E56	E55	E54	E53	E52	E51	E50	E49	E48	E47	E46	E45	E44	E43	E42	E41	E40	E39	E38	E37	E36	E35	E34	E33	E32

Bits	Field Name	Description	Type	Reset
31	E63	Event Missed #63	R	0x0
30	E62	Event Missed #62	R	0x0
29	E61	Event Missed #61	R	0x0
28	E60	Event Missed #60	R	0x0
27	E59	Event Missed #59	R	0x0
26	E58	Event Missed #58	R	0x0
25	E57	Event Missed #57	R	0x0
24	E56	Event Missed #56	R	0x0
23	E55	Event Missed #55	R	0x0
22	E54	Event Missed #54	R	0x0
21	E53	Event Missed #53	R	0x0
20	E52	Event Missed #52	R	0x0

Bits	Field Name	Description	Type	Reset
19	E51	Event Missed #51	R	0x0
18	E50	Event Missed #50	R	0x0
17	E49	Event Missed #49	R	0x0
16	E48	Event Missed #48	R	0x0
15	E47	Event Missed #47	R	0x0
14	E46	Event Missed #46	R	0x0
13	E45	Event Missed #45	R	0x0
12	E44	Event Missed #44	R	0x0
11	E43	Event Missed #43	R	0x0
10	E42	Event Missed #42	R	0x0
9	E41	Event Missed #41	R	0x0
8	E40	Event Missed #40	R	0x0
7	E39	Event Missed #39	R	0x0
6	E38	Event Missed #38	R	0x0
5	E37	Event Missed #37	R	0x0
4	E36	Event Missed #36	R	0x0
3	E35	Event Missed #35	R	0x0
2	E34	Event Missed #34	R	0x0
1	E33	Event Missed #33	R	0x0
0	E32	Event Missed #32	R	0x0

Table 11-61. Register Call Summary for Register EDMA_TPCC_EMRH

EDMA Controller Functional Description
- Third-Party Channel Controller: [0]
- Null PaRAM Set: [1]
- Dummy PaRAM Set: [2]
- Dummy Versus Null Transfer Comparison: [3]
- Interrupt Evaluation Operations: [4]
- Error Interrupts: [5]

EDMA Debug Checklist and Programming Tips
- EDMA Debug Checklist: [6]

EDMA Register Manual
- EDMA Register Summary: [7][8][9][10]
- EDMA Register Description: [11][12]

Table 11-62. EDMA_TPCC_EMCR

Address Offset	0x0000 0308		
Physical Address	0x4330 0308 0x40D1 0308 0x4151 0308 0x01D1 0308 0x420A 0308	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE_EDMA_TPCC
Description	Event Missed Clear Register: CPU write of '1' to the EDMA_TPCC_EMCR.En bit causes the EDMA_TPCC_EMR.En bit to be cleared. CPU write of '0' has no effect.. All error bits must be cleared before additional error interrupts will be asserted by CC.		
Type	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E31	E30	E29	E28	E27	E26	E25	E24	E23	E22	E21	E20	E19	E18	E17	E16	E15	E14	E13	E12	E11	E10	E9	E8	E7	E6	E5	E4	E3	E2	E1	E0

Bits	Field Name	Description	Type	Reset
31	E31	Event Missed Clear #31	W	0x0
30	E30	Event Missed Clear #30	W	0x0
29	E29	Event Missed Clear #29	W	0x0
28	E28	Event Missed Clear #28	W	0x0
27	E27	Event Missed Clear #27	W	0x0
26	E26	Event Missed Clear #26	W	0x0
25	E25	Event Missed Clear #25	W	0x0
24	E24	Event Missed Clear #24	W	0x0
23	E23	Event Missed Clear #23	W	0x0
22	E22	Event Missed Clear #22	W	0x0
21	E21	Event Missed Clear #21	W	0x0
20	E20	Event Missed Clear #20	W	0x0
19	E19	Event Missed Clear #19	W	0x0
18	E18	Event Missed Clear #18	W	0x0
17	E17	Event Missed Clear #17	W	0x0
16	E16	Event Missed Clear #16	W	0x0
15	E15	Event Missed Clear #15	W	0x0
14	E14	Event Missed Clear #14	W	0x0
13	E13	Event Missed Clear #13	W	0x0
12	E12	Event Missed Clear #12	W	0x0
11	E11	Event Missed Clear #11	W	0x0
10	E10	Event Missed Clear #10	W	0x0
9	E9	Event Missed Clear #9	W	0x0
8	E8	Event Missed Clear #8	W	0x0
7	E7	Event Missed Clear #7	W	0x0
6	E6	Event Missed Clear #6	W	0x0
5	E5	Event Missed Clear #5	W	0x0
4	E4	Event Missed Clear #4	W	0x0
3	E3	Event Missed Clear #3	W	0x0
2	E2	Event Missed Clear #2	W	0x0
1	E1	Event Missed Clear #1	W	0x0
0	E0	Event Missed Clear #0	W	0x0

Table 11-63. Register Call Summary for Register EDMA_TPCC_EMCR

EDMA Register Manual
- EDMA Register Summary: [0][1][2][3]
- EDMA Register Description: [4][5]

Table 11-64. EDMA_TPCC_EMCRH

Address Offset	0x0000 030C		
Physical Address	0x4330 030C 0x40D1 030C 0x4151 030C 0x01D1 030C 0x420A 030C	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE_EDMA_TPCC
Description	Event Missed Clear Register (High Part): CPU write of '1' to the EDMA_TPCC_EMCR.En bit causes the EDMA_TPCC_EMR.En bit to be cleared. CPU write of '0' has no effect. All error bits must be cleared before additional error interrupts will be asserted by CC.		
Type	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E63	E62	E61	E60	E59	E58	E57	E56	E55	E54	E53	E52	E51	E50	E49	E48	E47	E46	E45	E44	E43	E42	E41	E40	E39	E38	E37	E36	E35	E34	E33	E32

Bits	Field Name	Description	Type	Reset
31	E63	Event Missed Clear #63	W	0x0
30	E62	Event Missed Clear #62	W	0x0
29	E61	Event Missed Clear #61	W	0x0
28	E60	Event Missed Clear #60	W	0x0
27	E59	Event Missed Clear #59	W	0x0
26	E58	Event Missed Clear #58	W	0x0
25	E57	Event Missed Clear #57	W	0x0
24	E56	Event Missed Clear #56	W	0x0
23	E55	Event Missed Clear #55	W	0x0
22	E54	Event Missed Clear #54	W	0x0
21	E53	Event Missed Clear #53	W	0x0
20	E52	Event Missed Clear #52	W	0x0
19	E51	Event Missed Clear #51	W	0x0
18	E50	Event Missed Clear #50	W	0x0
17	E49	Event Missed Clear #49	W	0x0
16	E48	Event Missed Clear #48	W	0x0
15	E47	Event Missed Clear #47	W	0x0
14	E46	Event Missed Clear #46	W	0x0
13	E45	Event Missed Clear #45	W	0x0
12	E44	Event Missed Clear #44	W	0x0
11	E43	Event Missed Clear #43	W	0x0
10	E42	Event Missed Clear #42	W	0x0
9	E41	Event Missed Clear #41	W	0x0
8	E40	Event Missed Clear #40	W	0x0
7	E39	Event Missed Clear #39	W	0x0
6	E38	Event Missed Clear #38	W	0x0
5	E37	Event Missed Clear #37	W	0x0
4	E36	Event Missed Clear #36	W	0x0
3	E35	Event Missed Clear #35	W	0x0
2	E34	Event Missed Clear #34	W	0x0
1	E33	Event Missed Clear #33	W	0x0
0	E32	Event Missed Clear #32	W	0x0

Table 11-65. Register Call Summary for Register EDMA_TPCC_EMCRH

EDMA Register Manual

- EDMA Register Summary: [0][1][2][3]

Table 11-66. EDMA_TPCC_QEMR

Address Offset	0x0000 0310		
Physical Address	0x4330 0310	**Instance**	SYS_EDMA_TPCC
	0x40D1 0310		DSP1_EDMA_TPCC
	0x4151 0310		DSP2_EDMA_TPCC
	0x01D1 0310		DSP_EDMA_TPCC
	0x420A 0310		EVE_EDMA_TPCC

Table 11-66. EDMA_TPCC_QEMR (continued)

Description	QDMA Event Missed Register: The QDMA Event Missed register is set if 2 QDMA events are detected without the first event being cleared or if a Null TR is serviced. If any bit in the EDMA_TPCC_QEMR register is set (and all errors (including EDMA_TPCC_EMR / EDMA_TPCC_CCERR) were previously clear), then an error will be signaled with TPCC error interrupt.
Type	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								E7	E6	E5	E4	E3	E2	E1	E0

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Reserved	R	0x0
7	E7	Event Missed #7	R	0x0
6	E6	Event Missed #6	R	0x0
5	E5	Event Missed #5	R	0x0
4	E4	Event Missed #4	R	0x0
3	E3	Event Missed #3	R	0x0
2	E2	Event Missed #2	R	0x0
1	E1	Event Missed #1	R	0x0
0	E0	Event Missed #0	R	0x0

Table 11-67. Register Call Summary for Register EDMA_TPCC_QEMR

EDMA Controller Functional Description
- Null PaRAM Set: [0]
- Dummy PaRAM Set: [1]
- Dummy Versus Null Transfer Comparison: [2]
- QDMA Channels: [3]
- Interrupt Evaluation Operations: [4]
- Error Interrupts: [5]

EDMA Debug Checklist and Programming Tips
- EDMA Debug Checklist: [6][7]

EDMA Register Manual
- EDMA Register Summary: [8][9][10][11]
- EDMA Register Description: [12][13][14][15][16][17][18][19][20][21][22][23][24][25]

Table 11-68. EDMA_TPCC_QEMCR

Address Offset	0x0000 0314		
Physical Address	0x4330 0314 0x40D1 0314 0x4151 0314 0x01D1 0314 0x420A 0314	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE_EDMA_TPCC
Description	QDMA Event Missed Clear Register: CPU write of '1' to the EDMA_TPCC_QEMCR.En bit causes the EDMA_TPCC_QEMR.En bit to be cleared. CPU write of '0' has no effect. All error bits must be cleared before additional error interrupts will be asserted by CC.		
Type	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								E7	E6	E5	E4	E3	E2	E1	E0

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Reserved	R	0x0
7	E7	Event Missed Clear #7	W	0x0
6	E6	Event Missed Clear #6	W	0x0
5	E5	Event Missed Clear #5	W	0x0
4	E4	Event Missed Clear #4	W	0x0
3	E3	Event Missed Clear #3	W	0x0
2	E2	Event Missed Clear #2	W	0x0
1	E1	Event Missed Clear #1	W	0x0
0	E0	Event Missed Clear #0	W	0x0

Table 11-69. Register Call Summary for Register EDMA_TPCC_QEMCR

EDMA Register Manual

- EDMA Register Summary: [0][1][2][3]
- EDMA Register Description: [4]

Table 11-70. EDMA_TPCC_CCERR

Address Offset	0x0000 0318		
Physical Address	0x4330 0318 0x40D1 0318 0x4151 0318 0x01D1 0318 0x420A 0318	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE_EDMA_TPCC
Description	CC Error Register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED															TCERR	RESERVED								QTHRXCD7	QTHRXCD6	QTHRXCD5	QTHRXCD4	QTHRXCD3	QTHRXCD2	QTHRXCD1	QTHRXCD0

Bits	Field Name	Description	Type	Reset
31:17	RESERVED	Reserved	R	0x0
16	TCERR	Transfer Completion Code Error 0x0: Total number of allowed TCCs outstanding has not been reached. 0x1: Total number of allowed TCCs has been reached. TCERR can be cleared by writing a '1' to corresponding bit in EDMA_TPCC_CCERRCLR register. If any bit in the EDMA_TPCC_CCERR register is set (and all errors were previously clear), then an error will be signaled with TPCC error interrupt.	R	0x0
15:8	RESERVED	Reserved	R	0x0
7	QTHRXCD7	Queue Threshold Error for Q7 0x0: Watermark/threshold has not been exceeded. 0x1: Watermark/threshold has been exceeded. QTHRXCD7 can be cleared by writing a '1' to corresponding bit in EDMA_TPCC_CCERRCLR register. If any bit in the EDMA_TPCC_CCERR register is set (and all errors (including EDMA_TPCC_EMR/EDMA_TPCC_QEMR) were previously clear), then an error will be signaled with the TPCC error interrupt.	R	0x0

Bits	Field Name	Description	Type	Reset
6	QTHRXCD6	Queue Threshold Error for Q6 0x0 : Watermark/threshold has not been exceeded. 0x1 : Watermark/threshold has been exceeded. QTHRXCD6 can be cleared by writing a '1' to corresponding bit in EDMA_TPCC_CCERRCLR register. If any bit in the EDMA_TPCC_CCERR register is set (and all errors (including EDMA_TPCC_EMR/EDMA_TPCC_QEMR) were previously clear), then an error will be signaled with the TPCC error interrupt.	R	0x0
5	QTHRXCD5	Queue Threshold Error for Q5 0x0 : Watermark/threshold has not been exceeded. 0x1 : Watermark/threshold has been exceeded. QTHRXCD5 can be cleared by writing a '1' to corresponding bit in EDMA_TPCC_CCERRCLR register. If any bit in the EDMA_TPCC_CCERR register is set (and all errors (including EDMA_TPCC_EMR/EDMA_TPCC_QEMR) were previously clear), then an error will be signaled with the TPCC error interrupt.	R	0x0
4	QTHRXCD4	Queue Threshold Error for Q4 0x0: Watermark/threshold has not been exceeded. 0x1: Watermark/threshold has been exceeded. QTHRXCD4 can be cleared by writing a '1' to corresponding bit in EDMA_TPCC_CCERRCLR register. If any bit in the EDMA_TPCC_CCERR register is set (and all errors (including EDMA_TPCC_EMR/EDMA_TPCC_QEMR) were previously clear), then an error will be signaled with the TPCC error interrupt.	R	0x0
3	QTHRXCD3	Queue Threshold Error for Q3 0x0: Watermark/threshold has not been exceeded. 0x1 : Watermark/threshold has been exceeded. QTHRXCD3 can be cleared by writing a '1' to corresponding bit in EDMA_TPCC_CCERRCLR register. If any bit in the EDMA_TPCC_CCERR register is set (and all errors (including EDMA_TPCC_EMR/EDMA_TPCC_QEMR) were previously clear), then an error will be signaled with the TPCC error interrupt.	R	0x0
2	QTHRXCD2	Queue Threshold Error for Q2 0x0: Watermark/threshold has not been exceeded. 0x1: Watermark/threshold has been exceeded. QTHRXCD2 can be cleared by writing a '1' to corresponding bit in EDMA_TPCC_CCERRCLR register. If any bit in the EDMA_TPCC_CCERR register is set (and all errors (including EDMA_TPCC_EMR/EDMA_TPCC_QEMR) were previously clear), then an error will be signaled with the TPCC error interrupt.	R	0x0
1	QTHRXCD1	Queue Threshold Error for Q1 0x0: Watermark/threshold has not been exceeded. 0x1: Watermark/threshold has been exceeded. QTHRXCD1 can be cleared by writing a '1' to corresponding bit in EDMA_TPCC_CCERRCLR register. If any bit in the EDMA_TPCC_CCERR register is set (and all errors (including EDMA_TPCC_EMR/EDMA_TPCC_QEMR) were previously clear), then an error will be signaled with the TPCC error interrupt.	R	0x0

Bits	Field Name	Description	Type	Reset
0	QTHRXCD0	Queue Threshold Error for Q0: 0x0: Watermark/threshold has not been exceeded. 0x1: Watermark/threshold has been exceeded. QTHRXCD0 can be cleared by writing a '1' to corresponding bit in EDMA_TPCC_CCERRCLR register. If any bit in the EDMA_TPCC_CCERR register is set (and all errors (including EDMA_TPCC_EMR/EDMA_TPCC_QEMR) were previously clear), then an error will be signaled with the TPCC error interrupt.	R	0x0

Table 11-71. Register Call Summary for Register EDMA_TPCC_CCERR

EDMA Controller Functional Description

- Third-Party Channel Controller: [0]
- Interrupt Evaluation Operations: [1]
- Error Interrupts: [2][3]
- Queue Resource Tracking: [4][5]

EDMA Register Manual

- EDMA Register Summary: [6][7][8][9]
- EDMA Register Description: [10][11][12][13][14][15][16][17][18][19][20][21][22][23][24][25][26][27][28][29][30][31][32][33][34][35][36][37][38][39][40][41][42][43][44][45]

Table 11-72. EDMA_TPCC_CCERRCLR

Address Offset	0x0000 031C		
Physical Address	0x4330 031C 0x40D1 031C 0x4151 031C 0x01D1 031C 0x420A 031C	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE_EDMA_TPCC
Description	CC Error Clear Register		
Type	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED															TCERR	RESERVED								QTHRXCD7	QTHRXCD6	QTHRXCD5	QTHRXCD4	QTHRXCD3	QTHRXCD2	QTHRXCD1	QTHRXCD0

Bits	Field Name	Description	Type	Reset
31:17	RESERVED	Reserved	R	0x0
16	TCERR	Clear Error for EDMA_TPCC_CCERR[16] TR. Write 0x1 to clear the value of EDMA_TPCC_CCERR[16] TCERR. Write 0x0 have no affect.	W	0x0
15:8	RESERVED	Reserved	R	0x0
7	QTHRXCD7	Clear error for EDMA_TPCC_CCERR[7]QTHRXCD7 Write 0x0 have no affect. Write 0x1 to clear the values of QSTAT7.WM, QSTAT7.THRXCD, EDMA_TPCC_CCERR[7] QTHRXCD7	W	0x0

Bits	Field Name	Description	Type	Reset
6	QTHRXCD6	Clear error for EDMA_TPCC_CCERR[6] QTHRXCD6 Write 0x0 have no affect. Write 0x1 to clear the values of QSTAT6.WM, QSTAT6.THRXCD, EDMA_TPCC_CCERR[6]QTHRXCD6	W	0x0
5	QTHRXCD5	Clear error for EDMA_TPCC_CCERR[5] QTHRXCD5 Write 0x0 have no affect. Write 0x1 to clear the values of QSTAT5.WM, QSTAT5.THRXCD, EDMA_TPCC_CCERR[5]QTHRXCD5	W	0x0
4	QTHRXCD4	Clear error for EDMA_TPCC_CCERR[4] QTHRXCD4: Write 0x0 have no affect. Write 0x1 to clear the values of QSTAT4.WM, QSTAT4.THRXCD, EDMA_TPCC_CCERR[4] QTHRXCD4	W	0x0
3	QTHRXCD3	Clear error for EDMA_TPCC_CCERR[3] QTHRXCD3 Write 0x1 to clear the values of QSTAT3.WM, QSTAT3.THRXCD, EDMA_TPCC_CCERR[3] QTHRXCD3 Write 0x0 have no affect.	W	0x0
2	QTHRXCD2	Clear error for EDMA_TPCC_CCERR[2] QTHRXCD2 Write 0x0 have no affect. Write 0x1 to clear the values of QSTAT2.WM, QSTAT2.THRXCD, EDMA_TPCC_CCERR[2] QTHRXCD2	W	0x0
1	QTHRXCD1	Clear error for EDMA_TPCC_CCERR[1] QTHRXCD1 Write 0x1 to clear the values of QSTAT1.WM, QSTAT1.THRXCD, EDMA_TPCC_CCERR[1] QTHRXCD1 Write 0x0 have no affect.	W	0x0
0	QTHRXCD0	Clear error for EDMA_TPCC_CCERR[0] QTHRXCD0 Write 0x0 have no affect. Write 0x1 to clear the values of QSTAT0.WM, QSTAT0.THRXCD, EDMA_TPCC_CCERR[0] QTHRXCD0	W	0x0

Table 11-73. Register Call Summary for Register EDMA_TPCC_CCERRCLR

EDMA Register Manual

- EDMA Register Summary: [0][1][2][3]
- EDMA Register Description: [4][5][6][7][8][9][10][11][12]

Table 11-74. EDMA_TPCC_EEVAL

Address Offset	0x0000 0320		
Physical Address	0x4330 0320 0x40D1 0320 0x4151 0320 0x01D1 0320 0x420A 0320	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE_EDMA_TPCC
Description	Error Eval Register		
Type	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														SET	EVAL

Bits	Field Name	Description	Type	Reset
31:2	RESERVED	Reserved	R	0x000000
1	SET	Error Interrupt Set CPU writes 0x0 has no effect. CPU writes 0x1 to the SET bit causes the TPCC error interrupt to be pulsed regardless of state of EDMA_TPCC_EMR/EDMA_TPCC_EMRH, EDMA_TPCC_QEMR, or EDMA_TPCC_CCERR.	W	0x0
0	EVAL	Error Interrupt Evaluate CPU writes 0x0 has no effect. CPU writes 0x1 to the EVAL bit causes the TPCC error interrupt to be pulsed if any errors have not been cleared in the EDMA_TPCC_EMR/EDMA_TPCC_EMRH, EDMA_TPCC_QEMR, or EDMA_TPCC_CCERR registers. The CPU must also write 0x1 after any error interrupts are serviced in order for subsequent interrupts to be asserted.	W	0x0

Table 11-75. Register Call Summary for Register EDMA_TPCC_EEVAL

EDMA Controller Functional Description
- Interrupt Evaluation Operations: [0][1]
- Error Interrupts: [2][3]

EDMA Register Manual
- EDMA Register Summary: [4][5][6][7]

Table 11-76. EDMA_TPCC_DRAEM_k

Address Offset	0x0000 0340 + (0x8 * k)		
Physical Address	0x4330 0340 + (0x8 * k) 0x40D1 0340 + (0x8 * k) 0x4151 0340 + (0x8 * k) 0x01D1 0340 + (0x8 * k)	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC
Description	DMA Region Access enable for bit N in Region M: En = 0 : Accesses via Region M address space to Bit N in any DMA Channel Register are not allowed. Reads will return 'b0 on Bit N and writes will not modify the state of bit N. Enabled interrupt bits for bit N do not contribute to the generation of the TPCC region M interrupt. En = 1 : Accesses via Region M address space to Bit N in any DMA Channel Register are allowed. Reads will return the value from Bit N and writes will modify the state of bit N. Enabled interrupt bits for bit N do contribute to the generation of the TPCC region M interrupt.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E31	E30	E29	E28	E27	E26	E25	E24	E23	E22	E21	E20	E19	E18	E17	E16	E15	E14	E13	E12	E11	E10	E9	E8	E7	E6	E5	E4	E3	E2	E1	E0

Bits	Field Name	Description	Type	Reset
31	E31	DMA Region Access enable for Region M, bit #31	RW	0x0
30	E30	DMA Region Access enable for Region M, bit #30	RW	0x0
29	E29	DMA Region Access enable for Region M, bit #29	RW	0x0
28	E28	DMA Region Access enable for Region M, bit #28	RW	0x0
27	E27	DMA Region Access enable for Region M, bit #27	RW	0x0
26	E26	DMA Region Access enable for Region M, bit #26	RW	0x0

Bits	Field Name	Description	Type	Reset
25	E25	DMA Region Access enable for Region M, bit #25	RW	0x0
24	E24	DMA Region Access enable for Region M, bit #24	RW	0x0
23	E23	DMA Region Access enable for Region M, bit #23	RW	0x0
22	E22	DMA Region Access enable for Region M, bit #22	RW	0x0
21	E21	DMA Region Access enable for Region M, bit #21	RW	0x0
20	E20	DMA Region Access enable for Region M, bit #20	RW	0x0
19	E19	DMA Region Access enable for Region M, bit #19	RW	0x0
18	E18	DMA Region Access enable for Region M, bit #18	RW	0x0
17	E17	DMA Region Access enable for Region M, bit #17	RW	0x0
16	E16	DMA Region Access enable for Region M, bit #16	RW	0x0
15	E15	DMA Region Access enable for Region M, bit #15	RW	0x0
14	E14	DMA Region Access enable for Region M, bit #14	RW	0x0
13	E13	DMA Region Access enable for Region M, bit #13	RW	0x0
12	E12	DMA Region Access enable for Region M, bit #12	RW	0x0
11	E11	DMA Region Access enable for Region M, bit #11	RW	0x0
10	E10	DMA Region Access enable for Region M, bit #10	RW	0x0
9	E9	DMA Region Access enable for Region M, bit #9	RW	0x0
8	E8	DMA Region Access enable for Region M, bit #8	RW	0x0
7	E7	DMA Region Access enable for Region M, bit #7	RW	0x0
6	E6	DMA Region Access enable for Region M, bit #6	RW	0x0
5	E5	DMA Region Access enable for Region M, bit #5	RW	0x0
4	E4	DMA Region Access enable for Region M, bit #4	RW	0x0
3	E3	DMA Region Access enable for Region M, bit #3	RW	0x0
2	E2	DMA Region Access enable for Region M, bit #2	RW	0x0
1	E1	DMA Region Access enable for Region M, bit #1	RW	0x0
0	E0	DMA Region Access enable for Region M, bit #0	RW	0x0

Table 11-77. Register Call Summary for Register EDMA_TPCC_DRAEM_k

EDMA Controller Functional Description
- Region Overview: [0][1]
- Channel Controller Regions: [2][3][4][5][6][7]
- Transfer Completion Interrupts: [8][9][10][11][12][13][14][15][16][17][18][19][20]
- Interrupt Evaluation Operations: [21]
- Active Memory Protection: [22][23][24]

EDMA Transfer Examples
- Setting Up an EDMA Transfer: [25]

EDMA Debug Checklist and Programming Tips
- EDMA Debug Checklist: [26][27]
- EDMA Programming Tips: [28][29][30]

EDMA Register Manual
- EDMA Register Summary: [31][32][33][34]

Table 11-78. EDMA_TPCC_DRAEHM_k

Address Offset	0x0000 0344 + (0x8 * k)		
Physical Address	0x4330 0344 + (0x8 * k) 0x40D1 0344 + (0x8 * k) 0x4151 0344 + (0x8 * k) 0x01D1 0344 + (0x8 * k)	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC

Table 11-78. EDMA_TPCC_DRAEHM_k (continued)

Description	DMA Region Access enable for bit N in Region M: En = 0 : Accesses via Region M address space to Bit N in any DMA Channel Register are not allowed. Reads will return 'b0 on Bit N and writes will not modify the state of bit N. Enabled interrupt bits for bit N do not contribute to the generation of the TPCC region M interrupt. En = 1 : Accesses via Region M address space to Bit N in any DMA Channel Register are allowed. Reads will return the value from Bit N and writes will modify the state of bit N. Enabled interrupt bits for bit N do contribute to the generation of the TPCC region M interrupt. En = 0 : Accesses via Region M address space to Bit N in any DMA Channel Register are not allowed. Reads will return 'b0 on Bit N and writes will not modify the state of bit N. Enabled interrupt bits for bit N do not contribute to the generation of the TPCC region M interrupt. En = 1 : Accesses via Region M address space to Bit N in any DMA Channel Register are allowed. Reads will return the value from Bit N and writes will modify the state of bit N. Enabled interrupt bits for bit N do contribute to the generation of the TPCC region M interrupt.
Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E63	E62	E61	E60	E59	E58	E57	E56	E55	E54	E53	E52	E51	E50	E49	E48	E47	E46	E45	E44	E43	E42	E41	E40	E39	E38	E37	E36	E35	E34	E33	E32

Bits	Field Name	Description	Type	Reset
31	E63	DMA Region Access enable for Region M, bit #63	RW	0x0
30	E62	DMA Region Access enable for Region M, bit #62	RW	0x0
29	E61	DMA Region Access enable for Region M, bit #61	RW	0x0
28	E60	DMA Region Access enable for Region M, bit #60	RW	0x0
27	E59	DMA Region Access enable for Region M, bit #59	RW	0x0
26	E58	DMA Region Access enable for Region M, bit #58	RW	0x0
25	E57	DMA Region Access enable for Region M, bit #57	RW	0x0
24	E56	DMA Region Access enable for Region M, bit #56	RW	0x0
23	E55	DMA Region Access enable for Region M, bit #55	RW	0x0
22	E54	DMA Region Access enable for Region M, bit #54	RW	0x0
21	E53	DMA Region Access enable for Region M, bit #53	RW	0x0
20	E52	DMA Region Access enable for Region M, bit #52	RW	0x0
19	E51	DMA Region Access enable for Region M, bit #51	RW	0x0
18	E50	DMA Region Access enable for Region M, bit #50	RW	0x0
17	E49	DMA Region Access enable for Region M, bit #49	RW	0x0
16	E48	DMA Region Access enable for Region M, bit #48	RW	0x0
15	E47	DMA Region Access enable for Region M, bit #47	RW	0x0
14	E46	DMA Region Access enable for Region M, bit #46	RW	0x0
13	E45	DMA Region Access enable for Region M, bit #45	RW	0x0
12	E44	DMA Region Access enable for Region M, bit #44	RW	0x0
11	E43	DMA Region Access enable for Region M, bit #43	RW	0x0
10	E42	DMA Region Access enable for Region M, bit #42	RW	0x0
9	E41	DMA Region Access enable for Region M, bit #41	RW	0x0
8	E40	DMA Region Access enable for Region M, bit #40	RW	0x0
7	E39	DMA Region Access enable for Region M, bit #39	RW	0x0
6	E38	DMA Region Access enable for Region M, bit #38	RW	0x0
5	E37	DMA Region Access enable for Region M, bit #37	RW	0x0
4	E36	DMA Region Access enable for Region M, bit #36	RW	0x0
3	E35	DMA Region Access enable for Region M, bit #35	RW	0x0
2	E34	DMA Region Access enable for Region M, bit #34	RW	0x0
1	E33	DMA Region Access enable for Region M, bit #33	RW	0x0
0	E32	DMA Region Access enable for Region M, bit #32	RW	0x0

Table 11-79. Register Call Summary for Register EDMA_TPCC_DRAEHM_k

EDMA Controller Functional Description
- Region Overview: [0]
- Channel Controller Regions: [1][2][3]
- Transfer Completion Interrupts: [4][5][6][7][8][9][10][11][12]
- Interrupt Evaluation Operations: [13]

EDMA Transfer Examples
- Setting Up an EDMA Transfer: [14]

EDMA Debug Checklist and Programming Tips
- EDMA Debug Checklist: [15][16][17][18]
- EDMA Programming Tips: [19][20]

EDMA Register Manual
- EDMA Register Summary: [21][22][23][24]

Table 11-80. EDMA_TPCC_QRAEN_k

Address Offset	0x0000 0380 + (0x4 * k)		
Physical Address	0x4330 0380 + (0x4 * k) 0x40D1 0380 + (0x4 * k) 0x4151 0380 + (0x4 * k) 0x01D1 0380 + (0x4 * k) 0x420A 0380 + (0x4 * k)	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE_EDMA_TPCC
Description	QDMA Region Access enable for bit N in Region M: En = 0 : Accesses via Region M address space to Bit N in any QDMA Channel Register are not allowed. Reads will return 'b0 on Bit N and writes will not modify the state of bit N. Enabled interrupt bits for bit N do not contribute to the generation of the TPCC region M interrupt. En = 1 : Accesses via Region M address space to Bit N in any QDMA Channel Register are allowed. Reads will return the value from Bit N and writes will modify the state of bit N. Enabled interrupt bits for bit N do contribute to the generation of the TPCC region n interrupt.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								E7	E6	E5	E4	E3	E2	E1	E0

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Reserved	R	0x0
7	E7	QDMA Region Access enable for Region M, bit #7	RW	0x0
6	E6	QDMA Region Access enable for Region M, bit #6	RW	0x0
5	E5	QDMA Region Access enable for Region M, bit #5	RW	0x0
4	E4	QDMA Region Access enable for Region M, bit #4	RW	0x0
3	E3	QDMA Region Access enable for Region M, bit #3	RW	0x0
2	E2	QDMA Region Access enable for Region M, bit #2	RW	0x0
1	E1	QDMA Region Access enable for Region M, bit #1	RW	0x0
0	E0	QDMA Region Access enable for Region M, bit #0	RW	0x0

Table 11-81. Register Call Summary for Register EDMA_TPCC_QRAEN_k

EDMA Controller Functional Description
- Region Overview: [0][1]
- Channel Controller Regions: [2][3][4]

EDMA Debug Checklist and Programming Tips
- EDMA Programming Tips: [5]

EDMA Register Manual
- EDMA Register Summary: [6][7][8][9]

Table 11-82. EDMA_TPCC_Q0E_p

Address Offset	0x0000 0400 + (0x4 * l)		
Physical Address	0x4330 0400 + (0x4 * p) 0x40D1 0400 + (0x4 * p) 0x4151 0400 + (0x4 * p) 0x01D1 0400 + (0x4 * p)	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC
Description	Event Queue Entries Diagram for Queue 0 - Entry 0 through Entry 15		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								ETYPE		ENUM					

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Reserved	R	0x0
7:6	ETYPE	Event Type: Specifies the specific Event Type for the given entry in the Event Queue.	R	0x0
5:0	ENUM	Event Number: Specifies the specific Event Number for the given entry in the Event Queue. For DMA Channel events (EDMA_TPCC_ER/EDMA_TPCC_ESR/EDMA_TPCC_CER), ENUM will range between 0 and NUM_DMACH (up to 63). For QDMA Channel events (EDMA_TPCC_QER), ENUM will range between 0 and NUM_QDMACH (up to 7).	R	0x0

Table 11-83. Register Call Summary for Register EDMA_TPCC_Q0E_p

EDMA Controller Functional Description

- Event Queue(s): [0][1]

EDMA Register Manual

- EDMA Register Summary: [2][3][4][5]

Table 11-84. EDMA_TPCC_Q1E_p

Address Offset	0x0000 0440 + (0x4 * l)		
Physical Address	0x4330 0440 + (0x4 * p) 0x40D1 0440 + (0x4 * p) 0x4151 0440 + (0x4 * p) 0x01D1 0440 + (0x4 * p)	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC
Description	Event Queue Entries Diagram for Queue 1 - Entry 0 through Entry 15		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								ETYPE		ENUM					

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Reserved	R	0x0
7:6	ETYPE	Event Type: Specifies the specific Event Type for the given entry in the Event Queue.	R	0x0

Bits	Field Name	Description	Type	Reset
5:0	ENUM	Event Number: Specifies the specific Event Number for the given entry in the Event Queue. For DMA Channel events (EDMA_TPCC_ER / EDMA_TPCC_ESR / EDMA_TPCC_CER), ENUM will range between 0 and NUM_DMACH (up to 63). For QDMA Channel events (EDMA_TPCC_QER), ENUM will range between 0 and NUM_QDMACH (up to 7).	R	0x0

Table 11-85. Register Call Summary for Register EDMA_TPCC_Q1E_p

EDMA Controller Functional Description
- Event Queue(s): [0][1]

EDMA Register Manual
- EDMA Register Summary: [2][3][4][5]

Table 11-86. EDMA_TPCC_QSTATN_i

Address Offset	0x0000 0600 + (0x4 * i)		
Physical Address	0x4330 0600 + (0x4 * i) 0x40D1 0600 + (0x4 * i) 0x4151 0600 + (0x4 * i) 0x01D1 0600 + (0x4 * i) 0x420A 0600 + (0x4 * i)	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE_EDMA_TPCC
Description	QSTATn Register Set		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED							THRXCD	RESERVED			WM					RESERVED			NUMVAL					RESERVED				STRTPTR			

Bits	Field Name	Description	Type	Reset
31:25	RESERVED	Reserved	R Returns 0's	0x0
24	THRXCD	Threshold Exceeded 0x0 : Threshold specified by QWMTHR(A|B).Qn has not been exceeded. 0x1 : Threshold specified by QWMTHR(A|B).Qn has been exceeded. THRXCD is cleared via EDMA_TPCC_CCERR. WMCLRn bit.	R	0x0
23:21	RESERVED	Reserved	R Returns 0's	0x0
20:16	WM	Watermark for Maximum Queue Usage: Watermark tracks the most entries that have been in QueueN since reset or since the last time that the watermark (WM) was cleared. QSTATn. WM is cleared via EDMA_TPCC_CCERR.WMCLRn bit. Legal values: 0x0: empty 0x10: full	R	0x0
15:13	RESERVED	Reserved	Returns 0's	0x0

Bits	Field Name	Description	Type	Reset
12:8	NUMVAL	Number of Valid Entries in QueueN: Represents the total number of entries residing in the Queue Manager FIFO at a given instant. Always enabled. Legal values: = 0x0 (empty) to 0x10 (full) 0x0: empty 0x10: full	R	0x0
7:4	RESERVED	Reserved	Returns 0's	0x0
3:0	STRTPTR	Start Pointer: Represents the offset to the head entry of QueueN, in units of *entries*. Always enabled. Legal values: 0x0: 0th entry 0xF: 15th entry	R	0x0

Table 11-87. Register Call Summary for Register EDMA_TPCC_QSTATN_i

EDMA Controller Functional Description
- Queue RAM Debug Visibility: [0][1][2][3][4][5][6]
- Queue Resource Tracking: [7][8][9]

EDMA Register Manual
- EDMA Register Summary: [10][11][12][13]

Table 11-88. EDMA_TPCC_QWMTHRA

Address Offset	0x0000 0620		
Physical Address	0x4330 0620 0x40D1 0620 0x4151 0620 0x01D1 0620 0x420A 0620	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE_EDMA_TPCC
Description	Queue Threshold A, for Q[3:0]: EDMA_TPCC_CCERR.QTHRXCDn and QSTATn[24] THRXCD error bit is set when the number of Events in QueueN at an instant in time (visible via QSTATn[12:8] NUMVAL) equals or exceeds the value specified by EDMA_TPCC_QWMTHRA.Qn. Legal values = 0x0 (ever used?) to 0x10 (ever full?) A value of 0x11 disables threshold errors.		
Type	RW		

31 30 29	28 27 26 25 24	23 22 21	20 19 18 17 16	15 14 13	12 11 10 9 8	7 6 5	4 3 2 1 0
RESERVED	Q3	RESERVED	Q2	RESERVED	Q1	RESERVED	Q0

Bits	Field Name	Description	Type	Reset
31:29	RESERVED	Reserved	R	0x0
28:24	Q3	Queue Threshold for Q3 value	RW	0x10
23:21	RESERVED	Reserved	R	0x0
20:16	Q2	Queue Threshold for Q2 value	RW	0x10
15:13	RESERVED	Reserved	R	0x0
12:8	Q1	Queue Threshold for Q1 value	RW	0x10
7:5	RESERVED	Reserved	R	0x0
4:0	Q0	Queue Threshold for Q0 value	RW	0x10

Table 11-89. Register Call Summary for Register EDMA_TPCC_QWMTHRA

EDMA Controller Functional Description
- Queue Resource Tracking: [0]

EDMA Register Manual
- EDMA Register Summary: [1][2][3][4]
- EDMA Register Description: [5]

Table 11-90. EDMA_TPCC_QWMTHRB

Address Offset	0x0000 0624		
Physical Address	0x4330 0624 0x40D1 0624 0x4151 0624 0x01D1 0624 0x420A 0624	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE_EDMA_TPCC
Description	Queue Threshold B, for Q[7:4]: EDMA_TPCC_CCERR.QTHRXCDn and QSTATn[24]THRXCD error bit is set when the number of Events in QueueN at an instant in time (visible via QSTATn[12:8] NUMVAL) equals or exceeds the value specified by QWMTHRB.Qn. Legal values = 0x0 (ever used?) to 0x10 (ever full?) A value of 0x11 disables threshold errors.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED			Q7					RESERVED			Q6					RESERVED			Q5					RESERVED			Q4				

Bits	Field Name	Description	Type	Reset
31:29	RESERVED	Reserved	R	0x0
28:24	Q7	Queue Threshold for Q7 value (unused in the context of IVAHD)	RW	0x10
23:21	RESERVED	Reserved	R	0x0
20:16	Q6	Queue Threshold for Q6 value (unused in the context of IVAHD)	RW	0x10
15:13	RESERVED	Reserved	R	0x0
12:8	Q5	Queue Threshold for Q5 value (unused in the context of IVAHD)	RW	0x10
7:5	RESERVED	Reserved	R	0x0
4:0	Q4	Queue Threshold for Q4 value (unused in the context of IVAHD)	RW	0x10

Table 11-91. Register Call Summary for Register EDMA_TPCC_QWMTHRB

EDMA Register Manual
- EDMA Register Summary: [0][1][2][3]

Table 11-92. EDMA_TPCC_CCSTAT

Address Offset	0x0000 0640		
Physical Address	0x4330 0640 0x40D1 0640 0x4151 0640 0x01D1 0640 0x420A 0640	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE_EDMA_TPCC
Description	CC Status Register		

Table 11-92. EDMA_TPCC_CCSTAT (continued)

Type	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								QUEACTV7	QUEACTV6	QUEACTV5	QUEACTV4	QUEACTV3	QUEACTV2	QUEACTV1	QUEACTV0	RESERVED		COMPACTV						RESERVED			ACTV	RESERVED	TRACTV	QEVTACTV	EVTACTV

Bits	Field Name	Description	Type	Reset
31:24	RESERVED	reads return 0's	R	0x0
23	QUEACTV7	Queue 7 Active 0x0: No Evts are queued in Q7 0x1: At least one TR is queued in Q7.	R	0x0
22	QUEACTV6	Queue 6 Active 0x0: No Evts are queued in Q6. 0x1: At least one TR is queued in Q6.	R	0x0
21	QUEACTV5	Queue 5 Active 0x0: No Evts are queued in Q5 0x1: At least one TR is queued in Q5.	R	0x0
20	QUEACTV4	Queue 4 Active 0x0: No Evts are queued in Q4. 0x1: At least one TR is queued in Q4.	R	0x0
19	QUEACTV3	Queue 3 Active 0x0: No Evts are queued in Q3. 0x1: At least one TR is queued in Q3.	R	0x0
18	QUEACTV2	Queue 2 Active QUEACTV2 = 0 : No Evts are queued in Q2. QUEACTV2 = 1 : At least one TR is queued in Q2. 0x0: 0x1:	R	0x0
17	QUEACTV1	Queue 1 Active 0x0: No Evts are queued in Q1. 0x1: At least one TR is queued in Q1.	R	0x0
16	QUEACTV0	Queue 0 Active 0x0: No Evts are queued in Q0. 0x1: At least one TR is queued in Q0.	R	0x0
15:14	RESERVED	Reserved	R reads return 0's	0x0
13:8	COMPACTV	Completion Request Active: Counter that tracks the total number of completion requests submitted to the TC. The counter increments when a TR is submitted with TCINTEN or TCCHEN set to '1'. The counter decrements for every valid completion code received from any of the external TCs. The CC will not service new TRs if COMPACTV count is already at the limit. 0x0: No completion requests outstanding. 0x1: Total of '1' completion request outstanding. ... 0x3F: Total of 63 completion requests are outstanding. No additional TRs will be submitted until count is less than 63.	R	0x0

Bits	Field Name	Description	Type	Reset
7:5	RESERVED	reads return 0's	R	0x0
4	ACTV	Channel Controller Active Channel Controller Active is a logical-OR of each of the *ACTV signals. The ACTV bit must remain high through the life of a: 0x0: Channel is idle. 0x1: Channel is busy.	R	0x0
3	RESERVED	reads return 0's	R	0x0
2	TRACTV	Transfer Request Active TRACTV = 0 : Transfer Request processing/submission logic is inactive. TRACTV = 1 : Transfer Request processing/submission logic is active. 0x0: 0x1:	R	0x0
1	QEVTACTV	QDMA Event Active 0x0: No enabled QDMA Events are active within the CC. 0x1: At least one enabled DMA Event (EDMA_TPCC_ER, EDMA_TPCC_EER, EDMA_TPCC_ESR, EDMA_TPCC_CER) is active within the CC.	R	0x0
0	EVTACTV	DMA Event Active 0x0: No enabled DMA Events are active within the CC. 0x1: At least one enabled DMA Event (EDMA_TPCC_ER, EDMA_TPCC_EER, EDMA_TPCC_ESR, EDMA_TPCC_CER) is active within the CC.	R	0x0

Table 11-93. Register Call Summary for Register EDMA_TPCC_CCSTAT

EDMA Register Manual
- EDMA Register Summary: [0][1][2][3]

Table 11-94. EDMA_TPCC_AETCTL

Address Offset	0x0000 0700		
Physical Address	0x4330 0700 0x40D1 0700 0x4151 0700 0x01D1 0700 0x420A 0700	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE_EDMA_TPCC
Description	Advanced Event Trigger Control		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EN	RESERVED																	ENDINT						RESERVED	TYPE	STRTEVT					

Bits	Field Name	Description	Type	Reset
31	EN	AET Enable 0x0: AET event generation is disabled. 0x1: AET event generation is enabled.	RW	0x0
30:14	RESERVED	Reserved	R	0x0

Bits	Field Name	Description	Type	Reset
13:8	ENDINT	AET End Interrupt: Dictates the completion interrupt number that will force the tpcc_aet signal to be deasserted (low)	RW	0x0
7	RESERVED	Reserved	R	0x0
6	TYPE	AET Event Type 0x0: Event specified by STARTEVT applies to DMA Events (set by EDMA_TPCC_ER, EDMA_TPCC_ESR, or EDMA_TPCC_CER) 0x1: Event specified by STARTEVT applies to QDMA Events	RW	0x0
5:0	STRTEVT	AET Start Event: Dictates the Event Number that will force the tpcc_aet signal to be asserted (high)	RW	0x0

Table 11-95. Register Call Summary for Register EDMA_TPCC_AETCTL

EDMA Register Manual
- EDMA Register Summary: [0][1][2][3]

Table 11-96. EDMA_TPCC_AETSTAT

Address Offset	0x0000 0704		
Physical Address	0x4330 0704 0x40D1 0704 0x4151 0704 0x01D1 0704 0x420A 0704	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE_EDMA_TPCC
Description	Advanced Event Trigger Stat		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															STAT

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved	R Return 0's	0x0
0	STAT	AET Status 0x0: tpcc_aet is currently low. 0x1: tpcc_aet is currently high.	R	0x0

Table 11-97. Register Call Summary for Register EDMA_TPCC_AETSTAT

EDMA Register Manual
- EDMA Register Summary: [0][1][2][3]
- EDMA Register Description: [4]

Table 11-98. EDMA_TPCC_AETCMD

Address Offset	0x0000 0708		
Physical Address	0x4330 0708 0x40D1 0708 0x4151 0708 0x01D1 0708 0x420A 0708	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE_EDMA_TPCC
Description	AET Command		

Table 11-98. EDMA_TPCC_AETCMD (continued)

Type	W

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															CLR

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved	R	0x0
0	CLR	AET Clear command CPU writes 0x0 has no effect. CPU writes 0x1 to the CLR bit causes the tpcc_aet output signal and EDMA_TPCC_AETSTAT[0]STAT register to be cleared.	W	0x0

Table 11-99. Register Call Summary for Register EDMA_TPCC_AETCMD

EDMA Register Manual
- EDMA Register Summary: [0][1][2][3]

Table 11-100. EDMA_TPCC_MPFAR

Address Offset	0x0000 0800		
Physical Address	0x4330 0800 0x40D1 0800 0x4151 0800 0x01D1 0800 0x420A 0800	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE_EDMA_TPCC
Description	MMemory Protection Fault Address		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FADDR																															

Bits	Field Name	Description	Type	Reset
31:0	FADDR	Fault Address: 32-bit read-only status register containing the faulting address when a mMemory protection violation is detected. This register can only be cleared via the EDMA_TPCC_MPFCR.	R	0x0

Table 11-101. Register Call Summary for Register EDMA_TPCC_MPFAR

EDMA Register Manual
- EDMA Register Summary: [0][1][2][3]
- EDMA Register Description: [4]

Table 11-102. EDMA_TPCC_MPFSR

Address Offset	0x0000 0804		
Physical Address	0x4330 0804 0x40D1 0804 0x4151 0804 0x01D1 0804 0x420A 0804	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE_EDMA_TPCC
Description	Memory Protection Fault Status Register		

Table 11-102. EDMA_TPCC_MPFSR (continued)

Type	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																			FID					RESERVED		SRE	SWE	SXE	URE	UWE	UXE

Bits	Field Name	Description	Type	Reset
31:13	RESERVED	Reserved	R Returns 0	0x0
12:9	FID	Faulted ID: FID register contains valid info if any of the MP error bits (UXE, UWE, URE, SXE, SWE, SRE) are non-zero (i.e., if an error has been detected.) The FID field contains the VBus PrivID for the specific request/requestor that resulted in a MP Error.	R	0x0
8:6	RESERVED	Reserved	R Returns 0	0x0
5	SRE	Supervisor Read Error 0x0: No error detected. 0x1: Supervisor level task attempted to Read from a MP Page without SR permissions.	R	0x0
4	SWE	Supervisor Write Error 0x0: No error detected. 0x1: Supervisor level task attempted to Write to a MP Page without SW permissions.	R	0x0
3	SXE	Supervisor Execute Error 0x0: No error detected. 0x1: Supervisor level task attempted to Execute from a MP Page without SX permissions.	R	0x0
2	URE	User Read Error 0x0: No error detected. 0x1: User level task attempted to Read from a MP Page without UR permissions.	R	0x0
1	UWE	User Write Error 0x0: No error detected. 0x1: User level task attempted to Write to a MP Page without UW permissions.	R	0x0
0	UXE	User Execute Error 0x0: No error detected 0x1: User level task attempted to Execute from a MP Page without UX permissions.	R	0x0

Table 11-103. Register Call Summary for Register EDMA_TPCC_MPFSR

EDMA Register Manual

- EDMA Register Summary: [0][1][2][3]
- EDMA Register Description: [4]

Table 11-104. EDMA_TPCC_MPFCR

Address Offset	0x0000 0808		
Physical Address	0x4330 0808 0x40D1 0808 0x4151 0808 0x01D1 0808 0x420A 0808	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE_EDMA_TPCC
Description	Memory Protection Fault Command Register		
Type	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															MPFCLR

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved	R	0x0
0	MPFCLR	Fault Clear register CPU writes 0x0: has no effect CPU writes 0x1: to the MPFCLR bit causes any error conditions stored in EDMA_TPCC_MPFAR and EDMA_TPCC_MPFSR registers to be cleared.	W	0x0

Table 11-105. Register Call Summary for Register EDMA_TPCC_MPFCR

EDMA Register Manual

- EDMA Register Summary: [0][1][2][3]
- EDMA Register Description: [4]

Table 11-106. EDMA_TPCC_MPPAG

Address Offset	0x0000 080C		
Physical Address	0x4330 080C 0x40D1 080C 0x4151 080C 0x01D1 080C 0x420A 080C	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE_EDMA_TPCC
Description	Memory Protection Page Attribute for Global registers		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																AID5	AID4	AID3	AID2	AID1	AID0	EXT	RESERVED			SR	SW	SX	UR	UW	UX

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Reserved	R	0x0
15	AID5	Allowed ID 5 0x0: VBus requests with PrivID == '5' are not allowed regardless of permission settings (UW, UR, SW, SR).0 0x1: VBus requests with PrivID == '5' are permitted if access type is allowed as defined by permission settings (UW, UR, SW, SR).	RW	0x1

Bits	Field Name	Description	Type	Reset
14	AID4	Allowed ID 4 0x0: VBus requests with PrivID == '4' are not allowed regardless of permission settings (UW, UR, SW, SR). 0x1: VBus requests with PrivID == '4' are permitted if access type is allowed as defined by permission settings (UW, UR, SW, SR).	RW	0x1
13	AID3	Allowed ID 3 0x0: VBus requests with PrivID == '3' are not allowed regardless of permission settings (UW, UR, SW, SR). 0x1: VBus requests with PrivID == '3' are permitted if access type is allowed as defined by permission settings (UW, UR, SW, SR).	RW	0x1
12	AID2	Allowed ID 2 0x0: VBus requests with PrivID == '2' are not allowed regardless of permission settings (UW, UR, SW, SR). 0x1: VBus requests with PrivID == '2' are permitted if access type is allowed as defined by permission settings (UW, UR, SW, SR).	RW	0x1
11	AID1	Allowed ID 1 0x0: VBus requests with PrivID == '1' are not allowed regardless of permission settings (UW, UR, SW, SR). 0x1: VBus requests with PrivID == '1' are permitted if access type is allowed as defined by permission settings (UW, UR, SW, SR).	RW	0x1
10	AID0	Allowed ID 0 0x0: VBus requests with PrivID == '0' are not allowed regardless of permission settings (UW, UR, SW, SR). 0x1: VBus requests with PrivID == '0' are permitted if access type is allowed as defined by permission settings (UW, UR, SW, SR).	RW	0x1
9	EXT	External Allowed ID 0x0: VBus requests with PrivID = '6' are not allowed regardless of permission settings (UW, UR, SW, SR). 0x1: VBus requests with PrivID = '6' are permitted if access type is allowed as defined by permission settings (UW, UR, SW, SR).	RW	0x1
8:6	RESERVED	Reserved	R	0x1
5	SR	Supervisor Read permission 0x0: Supervisor read accesses are not allowed 0x1: Supervisor write accesses are allowed	RW	0x1
4	SW	Supervisor Write permission 0x0: Supervisor write accesses are not allowed 0x1: Supervisor write accesses are allowed	RW	0x1
3	SX	Supervisor Execute permission 0x0: Supervisor execute accesses are not allowed 0x1: Supervisor execute accesses are allowed	RW	0x0
2	UR	User Read permission 0x0: User read accesses are not allowed 0x1: User write accesses are allowed	RW	0x1
1	UW	User Write permission 0x0: User write accesses are not allowed 0x1: User write accesses are allowed	RW	0x1
0	UX	User Execute permission 0x0: User execute accesses are not allowed 0x1: User execute accesses are allowed	RW	0x0

Table 11-107. Register Call Summary for Register EDMA_TPCC_MPPAG

EDMA Controller Functional Description
- Channel Controller Regions: [0]
- Active Memory Protection: [1][2][3][4]

EDMA Register Manual
- EDMA Register Summary: [5][6][7][8]

Table 11-108. EDMA_TPCC_MPPAN_k

Address Offset	0x0000 0810 + (0x4 * k)		
Physical Address	0x4330 0810 + (0x4 * k) 0x40D1 0810 + (0x4 * k) 0x4151 0810 + (0x4 * k) 0x01D1 0810 + (0x4 * k) 0x420A 0810 + (0x4 * k)	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE_EDMA_TPCC
Description	P Permission Attribute for DMA Region n		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																AID5	AID4	AID3	AID2	AID1	AID0	EXT	RESERVED			SR	SW	SX	UR	UW	UX

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Reserved	R	0x0
15	AID5	Allowed ID 5 0x0: VBus requests with PrivID == '5' are not allowed regardless of permission settings (UW, UR, SW, SR). 0x1: VBus requests with PrivID == '5' are permitted if access type is allowed as defined by permission settings (UW, UR, SW, SR).	RW	0x1
14	AID4	Allowed ID 4 0x0: VBus requests with PrivID == '4' are not allowed regardless of permission settings (UW, UR, SW, SR). 0x1: VBus requests with PrivID == '4' are permitted if access type is allowed as defined by permission settings (UW, UR, SW, SR).	RW	0x1
13	AID3	Allowed ID 3 0x0: VBus requests with PrivID == '3' are not allowed regardless of permission settings (UW, UR, SW, SR). 0x1: VBus requests with PrivID == '3' are permitted if access type is allowed as defined by permission settings (UW, UR, SW, SR).	RW	0x1
12	AID2	Allowed ID 2 0x0: VBus requests with PrivID == '2' are not allowed regardless of permission settings (UW, UR, SW, SR). 0x1: VBus requests with PrivID == '2' are permitted if access type is allowed as defined by permission settings (UW, UR, SW, SR).	RW	0x1
11	AID1	Allowed ID 1 0x0: VBus requests with PrivID == '1' are not allowed regardless of permission settings (UW, UR, SW, SR). 0x1: VBus requests with PrivID == '1' are permitted if access type is allowed as defined by permission settings (UW, UR, SW, SR).	RW	0x1

Bits	Field Name	Description	Type	Reset
10	AID0	Allowed ID 0: AID0 = 0 : VBus requests with PrivID == '0' are not allowed regardless of permission settings (UW, UR, SW, SR). AID0 = 1 : VBus requests with PrivID == '0' are permitted if access type is allowed as defined by permission settings (UW, UR, SW, SR). 0x0: 0x1:	RW	0x1
9	EXT	External Allowed ID 0x0: VBus requests with PrivID = '6' are not allowed regardless of permission settings (UW, UR, SW, SR). 0x1: VBus requests with PrivID = '6' are permitted if access type is allowed as defined by permission settings (UW, UR, SW, SR).	RW	0x1
8:6	RESERVED	Reserved	R	0x0
5	SR	Supervisor Read permission 0x0: Supervisor read accesses are not allowed 0x1: Supervisor write accesses are allowed	RW	0x1
4	SW	Supervisor Write permission 0x0: Supervisor write accesses are not allowed 0x1: Supervisor write accesses are allowed	RW	0x1
3	SX	Supervisor Execute permission 0x0: Supervisor execute accesses are not allowed 0x1: Supervisor execute accesses are allowed	RW	0x0
2	UR	User Read permission 0x0: User read accesses are not allowed 0x1: User write accesses are allowed	RW	0x1
1	UW	User Write permission 0x0: User write accesses are not allowed 0x1: User write accesses are allowed	RW	0x1
0	UX	User Execute permission 0x0: User execute accesses are not allowed 0x0: User execute accesses are allowed	RW	0x0

Table 11-109. Register Call Summary for Register EDMA_TPCC_MPPAN_k

EDMA Controller Functional Description

- Channel Controller Regions: [0]
- Active Memory Protection: [1][2][3][4][5][6][7][8][9][10][11][12][13][14][15][16][17][18][19][20][21][22][23][24][25][26][27][28][29][30][31][32][33][34][35]
- Proxy Memory Protection: [36][37][38][39]

EDMA Register Manual

- EDMA Register Summary: [40][41][42][43]

Table 11-110. EDMA_TPCC_ER

Address Offset	0x0000 1000		
Physical Address	0x4330 1000 0x40D1 1000 0x4151 1000 0x01D1 1000 0x420A 1000	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE_EDMA_TPCC

Table 11-110. EDMA_TPCC_ER (continued)

Description	Event Register: If EDMA_TPCC_ER.En bit is set and the EDMA_TPCC_EER.En bit is also set, then the corresponding DMA channel is prioritized vs. other pending DMA events for submission to the TC. EDMA_TPCC_ER.En bit is set when the input event #n transitions from inactive (low) to active (high), regardless of the state of EDMA_TPCC_EER.En bit. EDMA_TPCC_ER.En bit is cleared when the corresponding event is prioritized and serviced. If the EDMA_TPCC_ER.En bit is already set and a new inactive to active transition is detected on the input event #n input AND the corresponding bit in the EDMA_TPCC_EER register is set, then the corresponding bit in the Event Missed Register is set. Event N can be cleared via sw by writing a '1' to the EDMA_TPCC_ECR pseudo-register.
Type	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E31	E30	E29	E28	E27	E26	E25	E24	E23	E22	E21	E20	E19	E18	E17	E16	E15	E14	E13	E12	E11	E10	E9	E8	E7	E6	E5	E4	E3	E2	E1	E0

Bits	Field Name	Description	Type	Reset
31	E31	Event #31	R	0x0
30	E30	Event #30	R	0x0
29	E29	Event #29	R	0x0
28	E28	Event #28	R	0x0
27	E27	Event #27	R	0x0
26	E26	Event #26	R	0x0
25	E25	Event #25	R	0x0
24	E24	Event #24	R	0x0
23	E23	Event #23	R	0x0
22	E22	Event #22	R	0x0
21	E21	Event #21	R	0x0
20	E20	Event #20	R	0x0
19	E19	Event #19	R	0x0
18	E18	Event #18	R	0x0
17	E17	Event #17	R	0x0
16	E16	Event #16	R	0x0
15	E15	Event #15	R	0x0
14	E14	Event #14	R	0x0
13	E13	Event #13	R	0x0
12	E12	Event #12	R	0x0
11	E11	Event #11	R	0x0
10	E10	Event #10	R	0x0
9	E9	Event #9	R	0x0
8	E8	Event #8	R	0x0
7	E7	Event #7	R	0x0
6	E6	Event #6	R	0x0
5	E5	Event #5	R	0x0
4	E4	Event #4	R	0x0
3	E3	Event #3	R	0x0
2	E2	Event #2	R	0x0
1	E1	Event #1	R	0x0
0	E0	Event #0	R	0x0

Table 11-111. Register Call Summary for Register EDMA_TPCC_ER

EDMA Controller Functional Description
- DMA Channels: [0][1][2][3][4][5][6][7][8][9]
- Region Overview: [10]
- Event Dataflow: [11][12]
- Channel Priority: [13]
- Trigger Source Priority: [14][15][16][17]

EDMA Transfer Examples
- Setting Up an EDMA Transfer: [18]

EDMA Debug Checklist and Programming Tips
- EDMA Debug Checklist: [19]
- EDMA Programming Tips: [20]

EDMA Register Manual
- EDMA Register Summary: [21][22][23][24]
- EDMA Register Description: [25][26][27][28][29][30][31][32][33][34][35][36][37][38][39][40][41][42][43][44][45][46][47][48][49][50][51][52][53][54][55][56][57][58][59][60][61][62][63]

Table 11-112. EDMA_TPCC_ERH

Address Offset	0x0000 1004		
Physical Address	0x4330 1004 0x40D1 1004 0x4151 1004 0x01D1 1004 0x420A 1004	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE_EDMA_TPCC
Description	Event Register (High Part): If EDMA_TPCC_ERH.En bit is set and the EDMA_TPCC_EERH.En bit is also set, then the corresponding DMA channel is prioritized vs. other pending DMA events for submission to the TC. EDMA_TPCC_ERH.En bit is set when the input event #n transitions from inactive (low) to active (high), regardless of the state of EDMA_TPCC_EERH.En bit. EDMA_TPCC_ER.En bit is cleared when the corresponding event is prioritized and serviced. If the EDMA_TPCC_ERH.En bit is already set and a new inactive to active transition is detected on the input event #n input AND the corresponding bit in the EDMA_TPCC_EERH register is set, then the corresponding bit in the Event Missed Register is set. Event N can be cleared via sw by writing a '1' to the EDMA_TPCC_ECRH pseudo-register.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E63	E62	E61	E60	E59	E58	E57	E56	E55	E54	E53	E52	E51	E50	E49	E48	E47	E46	E45	E44	E43	E42	E41	E40	E39	E38	E37	E36	E35	E34	E33	E32

Bits	Field Name	Description	Type	Reset
31	E63	Event #63	R	0x0
30	E62	Event #62	R	0x0
29	E61	Event #61	R	0x0
28	E60	Event #60	R	0x0
27	E59	Event #59	R	0x0
26	E58	Event #58	R	0x0
25	E57	Event #57	R	0x0
24	E56	Event #56	R	0x0
23	E55	Event #55	R	0x0
22	E54	Event #54	R	0x0
21	E53	Event #53	R	0x0
20	E52	Event #52	R	0x0
19	E51	Event #51	R	0x0

Bits	Field Name	Description	Type	Reset
18	E50	Event #50	R	0x0
17	E49	Event #49	R	0x0
16	E48	Event #48	R	0x0
15	E47	Event #47	R	0x0
14	E46	Event #46	R	0x0
13	E45	Event #45	R	0x0
12	E44	Event #44	R	0x0
11	E43	Event #43	R	0x0
10	E42	Event #42	R	0x0
9	E41	Event #41	R	0x0
8	E40	Event #40	R	0x0
7	E39	Event #39	R	0x0
6	E38	Event #38	R	0x0
5	E37	Event #37	R	0x0
4	E36	Event #36	R	0x0
3	E35	Event #35	R	0x0
2	E34	Event #34	R	0x0
1	E33	Event #33	R	0x0
0	E32	Event #32	R	0x0

Table 11-113. Register Call Summary for Register EDMA_TPCC_ERH

EDMA Controller Functional Description
- Region Overview: [0]
- Event Dataflow: [1][2]
- Channel Priority: [3]

EDMA Debug Checklist and Programming Tips
- EDMA Debug Checklist: [4]
- EDMA Programming Tips: [5]

EDMA Register Manual
- EDMA Register Summary: [6][7][8][9]
- EDMA Register Description: [10][11][12][13][14][15][16][17][18][19][20][21][22][23][24][25][26][27][28][29]

Table 11-114. EDMA_TPCC_ECR

Address Offset	0x0000 1008		
Physical Address	0x4330 1008 0x40D1 1008 0x4151 1008 0x01D1 1008 0x420A 1008	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE_EDMA_TPCC
Description	Event Clear Register: CPU write of '1' to the EDMA_TPCC_ECR.En bit causes the EDMA_TPCC_ER.En bit to be cleared. CPU write of '0' has no effect.		
Type	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E31	E30	E29	E28	E27	E26	E25	E24	E23	E22	E21	E20	E19	E18	E17	E16	E15	E14	E13	E12	E11	E10	E9	E8	E7	E6	E5	E4	E3	E2	E1	E0

Bits	Field Name	Description	Type	Reset
31	E31	Event #31	W	0x0
30	E30	Event #30	W	0x0
29	E29	Event #29	W	0x0
28	E28	Event #28	W	0x0
27	E27	Event #27	W	0x0
26	E26	Event #26	W	0x0
25	E25	Event #25	W	0x0
24	E24	Event #24	W	0x0
23	E23	Event #23	W	0x0
22	E22	Event #22	W	0x0
21	E21	Event #21	W	0x0
20	E20	Event #20	W	0x0
19	E19	Event #19	W	0x0
18	E18	Event #18	W	0x0
17	E17	Event #17	W	0x0
16	E16	Event #16	W	0x0
15	E15	Event #15	W	0x0
14	E14	Event #14	W	0x0
13	E13	Event #13	W	0x0
12	E12	Event #12	W	0x0
11	E11	Event #11	W	0x0
10	E10	Event #10	W	0x0
9	E9	Event #9	W	0x0
8	E8	Event #8	W	0x0
7	E7	Event #7	W	0x0
6	E6	Event #6	W	0x0
5	E5	Event #5	W	0x0
4	E4	Event #4	W	0x0
3	E3	Event #3	W	0x0
2	E2	Event #2	W	0x0
1	E1	Event #1	W	0x0
0	E0	Event #0	W	0x0

Table 11-115. Register Call Summary for Register EDMA_TPCC_ECR

EDMA Controller Functional Description • Region Overview: [0]
EDMA Debug Checklist and Programming Tips • EDMA Programming Tips: [1]
EDMA Register Manual • EDMA Register Summary: [2][3][4][5] • EDMA Register Description: [6][7][8][9]

Table 11-116. EDMA_TPCC_ECRH

Address Offset	0x0000 100C		
Physical Address	0x4330 100C 0x40D1 100C 0x4151 100C 0x01D1 100C 0x420A 100C	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE_EDMA_TPCC

Table 11-116. EDMA_TPCC_ECRH (continued)

Description	Event Clear Register (High Part): CPU write of '1' to the EDMA_TPCC_ECRH.En bit causes the EDMA_TPCC_ERH.En bit to be cleared. CPU write of '0' has no effect.
Type	W

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E63	E62	E61	E60	E59	E58	E57	E56	E55	E54	E53	E52	E51	E50	E49	E48	E47	E46	E45	E44	E43	E42	E41	E40	E39	E38	E37	E36	E35	E34	E33	E32

Bits	Field Name	Description	Type	Reset
31	E63	Event #63	W	0x0
30	E62	Event #62	W	0x0
29	E61	Event #61	W	0x0
28	E60	Event #60	W	0x0
27	E59	Event #59	W	0x0
26	E58	Event #58	W	0x0
25	E57	Event #57	W	0x0
24	E56	Event #56	W	0x0
23	E55	Event #55	W	0x0
22	E54	Event #54	W	0x0
21	E53	Event #53	W	0x0
20	E52	Event #52	W	0x0
19	E51	Event #51	W	0x0
18	E50	Event #50	W	0x0
17	E49	Event #49	W	0x0
16	E48	Event #48	W	0x0
15	E47	Event #47	W	0x0
14	E46	Event #46	W	0x0
13	E45	Event #45	W	0x0
12	E44	Event #44	W	0x0
11	E43	Event #43	W	0x0
10	E42	Event #42	W	0x0
9	E41	Event #41	W	0x0
8	E40	Event #40	W	0x0
7	E39	Event #39	W	0x0
6	E38	Event #38	W	0x0
5	E37	Event #37	W	0x0
4	E36	Event #36	W	0x0
3	E35	Event #35	W	0x0
2	E34	Event #34	W	0x0
1	E33	Event #33	W	0x0
0	E32	Event #32	W	0x0

Table 11-117. Register Call Summary for Register EDMA_TPCC_ECRH

EDMA Controller Functional Description • Region Overview: [0]
EDMA Debug Checklist and Programming Tips • EDMA Programming Tips: [1]

Table 11-117. Register Call Summary for Register EDMA_TPCC_ECRH (continued)

EDMA Register Manual

- EDMA Register Summary: [2][3][4][5]
- EDMA Register Description: [6][7][8][9]

Table 11-118. EDMA_TPCC_ESR

Address Offset	0x0000 1010		
Physical Address	0x4330 1010 0x40D1 1010 0x4151 1010 0x01D1 1010 0x420A 1010	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE_EDMA_TPCC
Description	Event Set Register: CPU write of '1' to the EDMA_TPCC_ESR.En bit causes the EDMA_TPCC_ER.En bit to be set. CPU write of '0' has no effect.		
Type	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E31	E30	E29	E28	E27	E26	E25	E24	E23	E22	E21	E20	E19	E18	E17	E16	E15	E14	E13	E12	E11	E10	E9	E8	E7	E6	E5	E4	E3	E2	E1	E0

Bits	Field Name	Description	Type	Reset
31	E31	Event #31	W	0x0
30	E30	Event #30	W	0x0
29	E29	Event #29	W	0x0
28	E28	Event #28	W	0x0
27	E27	Event #27	W	0x0
26	E26	Event #26	W	0x0
25	E25	Event #25	W	0x0
24	E24	Event #24	W	0x0
23	E23	Event #23	W	0x0
22	E22	Event #22	W	0x0
21	E21	Event #21	W	0x0
20	E20	Event #20	W	0x0
19	E19	Event #19	W	0x0
18	E18	Event #18	W	0x0
17	E17	Event #17	W	0x0
16	E16	Event #16	W	0x0
15	E15	Event #15	W	0x0
14	E14	Event #14	W	0x0
13	E13	Event #13	W	0x0
12	E12	Event #12	W	0x0
11	E11	Event #11	W	0x0
10	E10	Event #10	W	0x0
9	E9	Event #9	W	0x0
8	E8	Event #8	W	0x0
7	E7	Event #7	W	0x0
6	E6	Event #6	W	0x0
5	E5	Event #5	W	0x0
4	E4	Event #4	W	0x0
3	E3	Event #3	W	0x0

Bits	Field Name	Description	Type	Reset
2	E2	Event #2	W	0x0
1	E1	Event #1	W	0x0
0	E0	Event #0	W	0x0

Table 11-119. Register Call Summary for Register EDMA_TPCC_ESR

EDMA Module Overview
- EDMA Features: [0]

EDMA Controller Functional Description
- Third-Party Channel Controller: [1]
- Initiating a DMA Transfer: [2]
- DMA Channels: [3][4][5][6][7]
- Comparison Between DMA and QDMA Channels: [8]
- Region Overview: [9]
- Event Dataflow: [10][11]
- Trigger Source Priority: [12][13]

EDMA Transfer Examples
- Transfer Chaining Examples: [14]
- Setting Up an EDMA Transfer: [15]

EDMA Register Manual
- EDMA Register Summary: [16][17][18][19]
- EDMA Register Description: [20][21][22][23][24][25][26][27][28]

Table 11-120. EDMA_TPCC_ESRH

Address Offset	0x0000 1014		
Physical Address	0x4330 1014 0x40D1 1014 0x4151 1014 0x01D1 1014 0x420A 1014	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE_EDMA_TPCC
Description	Event Set Register (High Part) CPU write of '1' to the EDMA_TPCC_ESRH.En bit causes the EDMA_TPCC_ERH.En bit to be set. CPU write of '0' has no effect.		
Type	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E63	E62	E61	E60	E59	E58	E57	E56	E55	E54	E53	E52	E51	E50	E49	E48	E47	E46	E45	E44	E43	E42	E41	E40	E39	E38	E37	E36	E35	E34	E33	E32

Bits	Field Name	Description	Type	Reset
31	E63	Event #63	W	0x0
30	E62	Event #62	W	0x0
29	E61	Event #61	W	0x0
28	E60	Event #60	W	0x0
27	E59	Event #59	W	0x0
26	E58	Event #58	W	0x0
25	E57	Event #57	W	0x0
24	E56	Event #56	W	0x0
23	E55	Event #55	W	0x0
22	E54	Event #54	W	0x0
21	E53	Event #53	W	0x0
20	E52	Event #52	W	0x0

Bits	Field Name	Description	Type	Reset
19	E51	Event #51	W	0x0
18	E50	Event #50	W	0x0
17	E49	Event #49	W	0x0
16	E48	Event #48	W	0x0
15	E47	Event #47	W	0x0
14	E46	Event #46	W	0x0
13	E45	Event #45	W	0x0
12	E44	Event #44	W	0x0
11	E43	Event #43	W	0x0
10	E42	Event #42	W	0x0
9	E41	Event #41	W	0x0
8	E40	Event #40	W	0x0
7	E39	Event #39	W	0x0
6	E38	Event #38	W	0x0
5	E37	Event #37	W	0x0
4	E36	Event #36	W	0x0
3	E35	Event #35	W	0x0
2	E34	Event #34	W	0x0
1	E33	Event #33	W	0x0
0	E32	Event #32	W	0x0

Table 11-121. Register Call Summary for Register EDMA_TPCC_ESRH

EDMA Module Overview
- EDMA Features: [0]

EDMA Controller Functional Description
- Third-Party Channel Controller: [1]
- Initiating a DMA Transfer: [2]
- Region Overview: [3]
- Event Dataflow: [4][5]

EDMA Transfer Examples
- Setting Up an EDMA Transfer: [6]

EDMA Register Manual
- EDMA Register Summary: [7][8][9][10]
- EDMA Register Description: [11][12][13][14]

Table 11-122. EDMA_TPCC_CER

Address Offset	0x0000 1018		
Physical Address	0x4330 1018 0x40D1 1018 0x4151 1018 0x01D1 1018 0x420A 1018	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE_EDMA_TPCC
Description	Chained Event Register: If EDMA_TPCC_CER.En bit is set (regardless of state of EDMA_TPCC_EER.En), then the corresponding DMA channel is prioritized vs. other pending DMA events for submission to the TC. EDMA_TPCC_CER.En bit is set when a chaining completion code is returned from one of the 3PTCs via the completion interface, or is generated internally via Early Completion path. EDMA_TPCC_CER.En bit is cleared when the corresponding event is prioritized and serviced. If the EDMA_TPCC_CER.En bit is already set and the corresponding chaining completion code is returned from the TC, then the corresponding bit in the Event Missed Register is set. EDMA_TPCC_CER.En cannot be set or cleared via software.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E31	E30	E29	E28	E27	E26	E25	E24	E23	E22	E21	E20	E19	E18	E17	E16	E15	E14	E13	E12	E11	E10	E9	E8	E7	E6	E5	E4	E3	E2	E1	E0

Bits	Field Name	Description	Type	Reset
31	E31	Event #31	R	0x0
30	E30	Event #30	R	0x0
29	E29	Event #29	R	0x0
28	E28	Event #28	R	0x0
27	E27	Event #27	R	0x0
26	E26	Event #26	R	0x0
25	E25	Event #25	R	0x0
24	E24	Event #24	R	0x0
23	E23	Event #23	R	0x0
22	E22	Event #22	R	0x0
21	E21	Event #21	R	0x0
20	E20	Event #20	R	0x0
19	E19	Event #19	R	0x0
18	E18	Event #18	R	0x0
17	E17	Event #17	R	0x0
16	E16	Event #16	R	0x0
15	E15	Event #15	R	0x0
14	E14	Event #14	R	0x0
13	E13	Event #13	R	0x0
12	E12	Event #12	R	0x0
11	E11	Event #11	R	0x0
10	E10	Event #10	R	0x0
9	E9	Event #9	R	0x0
8	E8	Event #8	R	0x0
7	E7	Event #7	R	0x0
6	E6	Event #6	R	0x0
5	E5	Event #5	R	0x0
4	E4	Event #4	R	0x0
3	E3	Event #3	R	0x0
2	E2	Event #2	R	0x0
1	E1	Event #1	R	0x0
0	E0	Event #0	R	0x0

Table 11-123. Register Call Summary for Register EDMA_TPCC_CER

EDMA Controller Functional Description
- Third-Party Channel Controller: [0]
- Dummy Versus Null Transfer Comparison: [1]
- DMA Channels: [2][3][4][5][6][7]
- Completion of a DMA Transfer: [8]
- Dummy or Null Completion: [9]
- Region Overview: [10]
- Event Dataflow: [11][12]
- Trigger Source Priority: [13][14][15][16]

EDMA Transfer Examples
- Transfer Chaining Examples: [17][18]

Table 11-123. Register Call Summary for Register EDMA_TPCC_CER (continued)

EDMA Register Manual

- EDMA Register Summary: [19][20][21][22]
- EDMA Register Description: [23][24][25][26][27][28][29][30][31][32][33][34][35][36][37][38][39][40][41][42]

Table 11-124. EDMA_TPCC_CERH

Address Offset	0x0000 101C		
Physical Address	0x4330 101C 0x40D1 101C 0x4151 101C 0x01D1 101C 0x420A 101C	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE_EDMA_TPCC
Description	Chained Event Register (High Part): If EDMA_TPCC_CERH.En bit is set (regardless of state of EDMA_TPCC_EERH.En), then the corresponding DMA channel is prioritized vs. other pending DMA events for submission to the TC. EDMA_TPCC_CERH.En bit is set when a chaining completion code is returned from one of the 3PTCs via the completion interface, or is generated internally via Early Completion path. EDMA_TPCC_CERH.En bit is cleared when the corresponding event is prioritized and serviced. If the EDMA_TPCC_CERH.En bit is already set and the corresponding chaining completion code is returned from the TC, then the corresponding bit in the Event Missed Register is set. EDMA_TPCC_CERH.En cannot be set or cleared via software.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E63	E62	E61	E60	E59	E58	E57	E56	E55	E54	E53	E52	E51	E50	E49	E48	E47	E46	E45	E44	E43	E42	E41	E40	E39	E38	E37	E36	E35	E34	E33	E32

Bits	Field Name	Description	Type	Reset
31	E63	Event #63	R	0x0
30	E62	Event #62	R	0x0
29	E61	Event #61	R	0x0
28	E60	Event #60	R	0x0
27	E59	Event #59	R	0x0
26	E58	Event #58	R	0x0
25	E57	Event #57	R	0x0
24	E56	Event #56	R	0x0
23	E55	Event #55	R	0x0
22	E54	Event #54	R	0x0
21	E53	Event #53	R	0x0
20	E52	Event #52	R	0x0
19	E51	Event #51	R	0x0
18	E50	Event #50	R	0x0
17	E49	Event #49	R	0x0
16	E48	Event #48	R	0x0
15	E47	Event #47	R	0x0
14	E46	Event #46	R	0x0
13	E45	Event #45	R	0x0
12	E44	Event #44	R	0x0
11	E43	Event #43	R	0x0
10	E42	Event #42	R	0x0
9	E41	Event #41	R	0x0
8	E40	Event #40	R	0x0
7	E39	Event #39	R	0x0

Bits	Field Name	Description	Type	Reset
6	E38	Event #38	R	0x0
5	E37	Event #37	R	0x0
4	E36	Event #36	R	0x0
3	E35	Event #35	R	0x0
2	E34	Event #34	R	0x0
1	E33	Event #33	R	0x0
0	E32	Event #32	R	0x0

Table 11-125. Register Call Summary for Register EDMA_TPCC_CERH

EDMA Controller Functional Description
- Dummy Versus Null Transfer Comparison: [0]
- Dummy or Null Completion: [1]
- Region Overview: [2]
- Event Dataflow: [3][4]

EDMA Register Manual
- EDMA Register Summary: [5][6][7][8]
- EDMA Register Description: [9][10][11][12][13][14][15][16][17][18][19][20]

Table 11-126. EDMA_TPCC_EER

Address Offset	0x0000 1020		
Physical Address	0x4330 1020 0x40D1 1020 0x4151 1020 0x01D1 1020 0x420A 1020	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE_EDMA_TPCC
Description	Event Enable Register: Enables DMA transfers for EDMA_TPCC_ER.En pending events. EDMA_TPCC_ER.En is set based on externally asserted events (via tpcc_eventN_pi). This register has no effect on Chained Event Register (EDMA_TPCC_CER) or Event Set Register (EDMA_TPCC_ESR). Note that if a bit is set in EDMA_TPCC_ER.En while EDMA_TPCC_EER.En is disabled, no action is taken. If EDMA_TPCC_EER.En is enabled at a later point (and EDMA_TPCC_ER.En has not been cleared via SW) then the event will be recognized as a valid 'TR Sync' EDMA_TPCC_EER.En is not directly writeable. Events can be enabled via writes to EDMA_TPCC_EESR and can be disabled via writes to EDMA_TPCC_EECR register. EDMA_TPCC_EER.En = 0: EDMA_TPCC_ER.En is not enabled to trigger DMA transfers. EDMA_TPCC_EER.En = 1: EDMA_TPCC_ER.En is enabled to trigger DMA transfers.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E31	E30	E29	E28	E27	E26	E25	E24	E23	E22	E21	E20	E19	E18	E17	E16	E15	E14	E13	E12	E11	E10	E9	E8	E7	E6	E5	E4	E3	E2	E1	E0

Bits	Field Name	Description	Type	Reset
31	E31	Event #31	R	0x0
30	E30	Event #30	R	0x0
29	E29	Event #29	R	0x0
28	E28	Event #28	R	0x0
27	E27	Event #27	R	0x0
26	E26	Event #26	R	0x0
25	E25	Event #25	R	0x0
24	E24	Event #24	R	0x0
23	E23	Event #23	R	0x0
22	E22	Event #22	R	0x0
21	E21	Event #21	R	0x0

Bits	Field Name	Description	Type	Reset
20	E20	Event #20	R	0x0
19	E19	Event #19	R	0x0
18	E18	Event #18	R	0x0
17	E17	Event #17	R	0x0
16	E16	Event #16	R	0x0
15	E15	Event #15	R	0x0
14	E14	Event #14	R	0x0
13	E13	Event #13	R	0x0
12	E12	Event #12	R	0x0
11	E11	Event #11	R	0x0
10	E10	Event #10	R	0x0
9	E9	Event #9	R	0x0
8	E8	Event #8	R	0x0
7	E7	Event #7	R	0x0
6	E6	Event #6	R	0x0
5	E5	Event #5	R	0x0
4	E4	Event #4	R	0x0
3	E3	Event #3	R	0x0
2	E2	Event #2	R	0x0
1	E1	Event #1	R	0x0
0	E0	Event #0	R	0x0

Table 11-127. Register Call Summary for Register EDMA_TPCC_EER

EDMA Controller Functional Description
- DMA Channels: [0][1][2][3][4][5]
- Region Overview: [6][7]
- Active Memory Protection: [8][9][10][11][12][13][14][15][16][17][18][19][20][21]

EDMA Transfer Examples
- Peripheral Servicing Example: [22]
- Setting Up an EDMA Transfer: [23]

EDMA Debug Checklist and Programming Tips
- EDMA Debug Checklist: [24]
- EDMA Programming Tips: [25]

EDMA Register Manual
- EDMA Register Summary: [26][27][28][29]
- EDMA Register Description: [30][31][32][33][34][35][36][37][38][39][40][41][42][43][44][45][46][47][48][49][50][51][52][53]

Table 11-128. EDMA_TPCC_EERH

Address Offset	0x0000 1024		
Physical Address	0x4330 1024	**Instance**	SYS_EDMA_TPCC
	0x40D1 1024		DSP1_EDMA_TPCC
	0x4151 1024		DSP2_EDMA_TPCC
	0x01D1 1024		DSP_EDMA_TPCC
	0x420A 1024		EVE_EDMA_TPCC

Table 11-128. EDMA_TPCC_EERH (continued)

Description	Event Enable Register (High Part): Enables DMA transfers for EDMA_TPCC_ERH.En pending events. EDMA_TPCC_ERH.En is set based on externally asserted events (via tpcc_eventN_pi). This register has no effect on Chained Event Register (EDMA_TPCC_CERH) or Event Set Register (EDMA_TPCC_ESRH). Note that if a bit is set in EDMA_TPCC_ERH.En while EDMA_TPCC_EERH.En is disabled, no action is taken. If EDMA_TPCC_EERH.En is enabled at a later point (and EDMA_TPCC_ERH.En has not been cleared via SW) then the event will be recognized as a valid 'TR Sync' EDMA_TPCC_EERH.En is not directly writeable. Events can be enabled via writes to EDMA_TPCC_EESRH and can be disabled via writes to EDMA_TPCC_EECRH register. EDMA_TPCC_EERH.En = 0: EDMA_TPCC_ER.En is not enabled to trigger DMA transfers. EDMA_TPCC_EERH.En = 1: EDMA_TPCC_ER.En is enabled to trigger DMA transfers.
Type	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E63	E62	E61	E60	E59	E58	E57	E56	E55	E54	E53	E52	E51	E50	E49	E48	E47	E46	E45	E44	E43	E42	E41	E40	E39	E38	E37	E36	E35	E34	E33	E32

Bits	Field Name	Description	Type	Reset
31	E63	Event #63	R	0x0
30	E62	Event #62	R	0x0
29	E61	Event #61	R	0x0
28	E60	Event #60	R	0x0
27	E59	Event #59	R	0x0
26	E58	Event #58	R	0x0
25	E57	Event #57	R	0x0
24	E56	Event #56	R	0x0
23	E55	Event #55	R	0x0
22	E54	Event #54	R	0x0
21	E53	Event #53	R	0x0
20	E52	Event #52	R	0x0
19	E51	Event #51	R	0x0
18	E50	Event #50	R	0x0
17	E49	Event #49	R	0x0
16	E48	Event #48	R	0x0
15	E47	Event #47	R	0x0
14	E46	Event #46	R	0x0
13	E45	Event #45	R	0x0
12	E44	Event #44	R	0x0
11	E43	Event #43	R	0x0
10	E42	Event #42	R	0x0
9	E41	Event #41	R	0x0
8	E40	Event #40	R	0x0
7	E39	Event #39	R	0x0
6	E38	Event #38	R	0x0
5	E37	Event #37	R	0x0
4	E36	Event #36	R	0x0
3	E35	Event #35	R	0x0
2	E34	Event #34	R	0x0
1	E33	Event #33	R	0x0
0	E32	Event #32	R	0x0

Table 11-129. Register Call Summary for Register EDMA_TPCC_EERH

EDMA Controller Functional Description • Region Overview: [0]
EDMA Transfer Examples • Setting Up an EDMA Transfer: [1]
EDMA Debug Checklist and Programming Tips • EDMA Debug Checklist: [2] • EDMA Programming Tips: [3]
EDMA Register Manual • EDMA Register Summary: [4][5][6][7] • EDMA Register Description: [8][9][10][11][12][13][14][15][16][17][18][19][20][21][22][23][24][25][26][27]

Table 11-130. EDMA_TPCC_EECR

Address Offset	0x0000 1028		
Physical Address	0x4330 1028 0x40D1 1028 0x4151 1028 0x01D1 1028 0x420A 1028	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE_EDMA_TPCC
Description	Event Enable Clear Register CPU writes of '1' to the EDMA_TPCC_EECR.En bit causes the EDMA_TPCC_EER.En bit to be cleared. CPU writes of '0' has no effect..		
Type	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E31	E30	E29	E28	E27	E26	E25	E24	E23	E22	E21	E20	E19	E18	E17	E16	E15	E14	E13	E12	E11	E10	E9	E8	E7	E6	E5	E4	E3	E2	E1	E0

Bits	Field Name	Description	Type	Reset
31	E31	Event #31	W	0x0
30	E30	Event #30	W	0x0
29	E29	Event #29	W	0x0
28	E28	Event #28	W	0x0
27	E27	Event #27	W	0x0
26	E26	Event #26	W	0x0
25	E25	Event #25	W	0x0
24	E24	Event #24	W	0x0
23	E23	Event #23	W	0x0
22	E22	Event #22	W	0x0
21	E21	Event #21	W	0x0
20	E20	Event #20	W	0x0
19	E19	Event #19	W	0x0
18	E18	Event #18	W	0x0
17	E17	Event #17	W	0x0
16	E16	Event #16	W	0x0
15	E15	Event #15	W	0x0
14	E14	Event #14	W	0x0
13	E13	Event #13	W	0x0
12	E12	Event #12	W	0x0
11	E11	Event #11	W	0x0
10	E10	Event #10	W	0x0

Bits	Field Name	Description	Type	Reset
9	E9	Event #9	W	0x0
8	E8	Event #8	W	0x0
7	E7	Event #7	W	0x0
6	E6	Event #6	W	0x0
5	E5	Event #5	W	0x0
4	E4	Event #4	W	0x0
3	E3	Event #3	W	0x0
2	E2	Event #2	W	0x0
1	E1	Event #1	W	0x0
0	E0	Event #0	W	0x0

Table 11-131. Register Call Summary for Register EDMA_TPCC_EECR

EDMA Controller Functional Description
- Region Overview: [0]

EDMA Debug Checklist and Programming Tips
- EDMA Programming Tips: [1]

EDMA Register Manual
- EDMA Register Summary: [2][3][4][5]
- EDMA Register Description: [6][7][8][9]

Table 11-132. EDMA_TPCC_EECRH

Address Offset	0x0000 102C		
Physical Address	0x4330 102C 0x40D1 102C 0x4151 102C 0x01D1 102C 0x420A 102C	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE_EDMA_TPCC
Description	Event Enable Clear Register (High Part) CPU writes of '1' to the EDMA_TPCC_EECRH.En bit causes the EERH.En bit to be cleared. CPU writes of '0' has no effect..		
Type	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E63	E62	E61	E60	E59	E58	E57	E56	E55	E54	E53	E52	E51	E50	E49	E48	E47	E46	E45	E44	E43	E42	E41	E40	E39	E38	E37	E36	E35	E34	E33	E32

Bits	Field Name	Description	Type	Reset
31	E63	Event #63	W	0x0
30	E62	Event #62	W	0x0
29	E61	Event #61	W	0x0
28	E60	Event #60	W	0x0
27	E59	Event #59	W	0x0
26	E58	Event #58	W	0x0
25	E57	Event #57	W	0x0
24	E56	Event #56	W	0x0
23	E55	Event #55	W	0x0
22	E54	Event #54	W	0x0
21	E53	Event #53	W	0x0
20	E52	Event #52	W	0x0
19	E51	Event #51	W	0x0

Bits	Field Name	Description	Type	Reset
18	E50	Event #50	W	0x0
17	E49	Event #49	W	0x0
16	E48	Event #48	W	0x0
15	E47	Event #47	W	0x0
14	E46	Event #46	W	0x0
13	E45	Event #45	W	0x0
12	E44	Event #44	W	0x0
11	E43	Event #43	W	0x0
10	E42	Event #42	W	0x0
9	E41	Event #41	W	0x0
8	E40	Event #40	W	0x0
7	E39	Event #39	W	0x0
6	E38	Event #38	W	0x0
5	E37	Event #37	W	0x0
4	E36	Event #36	W	0x0
3	E35	Event #35	W	0x0
2	E34	Event #34	W	0x0
1	E33	Event #33	W	0x0
0	E32	Event #32	W	0x0

Table 11-133. Register Call Summary for Register EDMA_TPCC_EECRH

EDMA Controller Functional Description
- Region Overview: [0]

EDMA Debug Checklist and Programming Tips
- EDMA Programming Tips: [1]

EDMA Register Manual
- EDMA Register Summary: [2][3][4][5]
- EDMA Register Description: [6][7][8][9]

Table 11-134. EDMA_TPCC_EESR

Address Offset	0x0000 1030		
Physical Address	0x4330 1030 0x40D1 1030 0x4151 1030 0x01D1 1030 0x420A 1030	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE_EDMA_TPCC
Description	Event Enable Set Register CPU write of '1' to the EDMA_TPCC_EESR.En bit causes the EDMA_TPCC_EER.En bit to be set. CPU writes of '0' has no effect..		
Type	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E31	E30	E29	E28	E27	E26	E25	E24	E23	E22	E21	E20	E19	E18	E17	E16	E15	E14	E13	E12	E11	E10	E9	E8	E7	E6	E5	E4	E3	E2	E1	E0

Bits	Field Name	Description	Type	Reset
31	E31	Event #31	W	0x0
30	E30	Event #30	W	0x0
29	E29	Event #29	W	0x0
28	E28	Event #28	W	0x0

Bits	Field Name	Description	Type	Reset
27	E27	Event #27	W	0x0
26	E26	Event #26	W	0x0
25	E25	Event #25	W	0x0
24	E24	Event #24	W	0x0
23	E23	Event #23	W	0x0
22	E22	Event #22	W	0x0
21	E21	Event #21	W	0x0
20	E20	Event #20	W	0x0
19	E19	Event #19	W	0x0
18	E18	Event #18	W	0x0
17	E17	Event #17	W	0x0
16	E16	Event #16	W	0x0
15	E15	Event #15	W	0x0
14	E14	Event #14	W	0x0
13	E13	Event #13	W	0x0
12	E12	Event #12	W	0x0
11	E11	Event #11	W	0x0
10	E10	Event #10	W	0x0
9	E9	Event #9	W	0x0
8	E8	Event #8	W	0x0
7	E7	Event #7	W	0x0
6	E6	Event #6	W	0x0
5	E5	Event #5	W	0x0
4	E4	Event #4	W	0x0
3	E3	Event #3	W	0x0
2	E2	Event #2	W	0x0
1	E1	Event #1	W	0x0
0	E0	Event #0	W	0x0

Table 11-135. Register Call Summary for Register EDMA_TPCC_EESR

EDMA Controller Functional Description
- Region Overview: [0]
- Active Memory Protection: [1][2][3][4][5][6][7][8][9][10][11][12]

EDMA Transfer Examples
- Setting Up an EDMA Transfer: [13]

EDMA Debug Checklist and Programming Tips
- EDMA Programming Tips: [14]

EDMA Register Manual
- EDMA Register Summary: [15][16][17][18]
- EDMA Register Description: [19][20][21][22]

Table 11-136. EDMA_TPCC_EESRH

Address Offset	0x0000 1034		
Physical Address	0x4330 1034 0x40D1 1034 0x4151 1034 0x01D1 1034 0x420A 1034	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE_EDMA_TPCC

Table 11-136. EDMA_TPCC_EESRH (continued)

Description	Event Enable Set Register (High Part) CPU writes of '1' to the EDMA_TPCC_EESRH.En bit causes the EDMA_TPCC_EERH.En bit to be set. CPU writes of '0' has no effect..
Type	W

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E63	E62	E61	E60	E59	E58	E57	E56	E55	E54	E53	E52	E51	E50	E49	E48	E47	E46	E45	E44	E43	E42	E41	E40	E39	E38	E37	E36	E35	E34	E33	E32

Bits	Field Name	Description	Type	Reset
31	E63	Event #63	W	0x0
30	E62	Event #62	W	0x0
29	E61	Event #61	W	0x0
28	E60	Event #60	W	0x0
27	E59	Event #59	W	0x0
26	E58	Event #58	W	0x0
25	E57	Event #57	W	0x0
24	E56	Event #56	W	0x0
23	E55	Event #55	W	0x0
22	E54	Event #54	W	0x0
21	E53	Event #53	W	0x0
20	E52	Event #52	W	0x0
19	E51	Event #51	W	0x0
18	E50	Event #50	W	0x0
17	E49	Event #49	W	0x0
16	E48	Event #48	W	0x0
15	E47	Event #47	W	0x0
14	E46	Event #46	W	0x0
13	E45	Event #45	W	0x0
12	E44	Event #44	W	0x0
11	E43	Event #43	W	0x0
10	E42	Event #42	W	0x0
9	E41	Event #41	W	0x0
8	E40	Event #40	W	0x0
7	E39	Event #39	W	0x0
6	E38	Event #38	W	0x0
5	E37	Event #37	W	0x0
4	E36	Event #36	W	0x0
3	E35	Event #35	W	0x0
2	E34	Event #34	W	0x0
1	E33	Event #33	W	0x0
0	E32	Event #32	W	0x0

Table 11-137. Register Call Summary for Register EDMA_TPCC_EESRH

EDMA Controller Functional Description • Region Overview: [0]
EDMA Transfer Examples • Setting Up an EDMA Transfer: [1]

Table 11-137. Register Call Summary for Register EDMA_TPCC_EESRH (continued)

EDMA Debug Checklist and Programming Tips • EDMA Programming Tips: [2]
EDMA Register Manual • EDMA Register Summary: [3][4][5][6] • EDMA Register Description: [7][8][9][10]

Table 11-138. EDMA_TPCC_SER

Address Offset	0x0000 1038		
Physical Address	0x4330 1038 0x40D1 1038 0x4151 1038 0x01D1 1038 0x420A 1038	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE_EDMA_TPCC
Description	Secondary Event Register The secondary event register is used along with the Event Register (EDMA_TPCC_ER) to provide information on the state of an Event. En = 0 : Event is not currently in the Event Queue. En = 1 : Event is currently stored in Event Queue. Event arbiter will not prioritize additional events.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E31	E30	E29	E28	E27	E26	E25	E24	E23	E22	E21	E20	E19	E18	E17	E16	E15	E14	E13	E12	E11	E10	E9	E8	E7	E6	E5	E4	E3	E2	E1	E0

Bits	Field Name	Description	Type	Reset
31	E31	Event #31	R	0x0
30	E30	Event #30	R	0x0
29	E29	Event #29	R	0x0
28	E28	Event #28	R	0x0
27	E27	Event #27	R	0x0
26	E26	Event #26	R	0x0
25	E25	Event #25	R	0x0
24	E24	Event #24	R	0x0
23	E23	Event #23	R	0x0
22	E22	Event #22	R	0x0
21	E21	Event #21	R	0x0
20	E20	Event #20	R	0x0
19	E19	Event #19	R	0x0
18	E18	Event #18	R	0x0
17	E17	Event #17	R	0x0
16	E16	Event #16	R	0x0
15	E15	Event #15	R	0x0
14	E14	Event #14	R	0x0
13	E13	Event #13	R	0x0
12	E12	Event #12	R	0x0
11	E11	Event #11	R	0x0
10	E10	Event #10	R	0x0
9	E9	Event #9	R	0x0
8	E8	Event #8	R	0x0
7	E7	Event #7	R	0x0

Bits	Field Name	Description	Type	Reset
6	E6	Event #6	R	0x0
5	E5	Event #5	R	0x0
4	E4	Event #4	R	0x0
3	E3	Event #3	R	0x0
2	E2	Event #2	R	0x0
1	E1	Event #1	R	0x0
0	E0	Event #0	R	0x0

Table 11-139. Register Call Summary for Register EDMA_TPCC_SER

EDMA Controller Functional Description
- Null PaRAM Set: [0][1]
- Dummy PaRAM Set: [2]
- Dummy Versus Null Transfer Comparison: [3][4]
- Region Overview: [5]
- Event Dataflow: [6][7][8]

EDMA Debug Checklist and Programming Tips
- EDMA Debug Checklist: [9][10][11]

EDMA Register Manual
- EDMA Register Summary: [12][13][14][15]
- EDMA Register Description: [16][17][18][19][20][21]

Table 11-140. EDMA_TPCC_SERH

Address Offset	0x0000 103C		
Physical Address	0x4330 103C 0x40D1 103C 0x4151 103C 0x01D1 103C 0x420A 103C	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE_EDMA_TPCC
Description	Secondary Event Register (High Part) The secondary event register is used along with the Event Register (EDMA_TPCC_ERH) to provide information on the state of an Event. En = 0 : Event is not currently in the Event Queue. En = 1 : Event is currently stored in Event Queue. Event arbiter will not prioritize additional events.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E63	E62	E61	E60	E59	E58	E57	E56	E55	E54	E53	E52	E51	E50	E49	E48	E47	E46	E45	E44	E43	E42	E41	E40	E39	E38	E37	E36	E35	E34	E33	E32

Bits	Field Name	Description	Type	Reset
31	E63	Event #63	R	0x0
30	E62	Event #62	R	0x0
29	E61	Event #61	R	0x0
28	E60	Event #60	R	0x0
27	E59	Event #59	R	0x0
26	E58	Event #58	R	0x0
25	E57	Event #57	R	0x0
24	E56	Event #56	R	0x0
23	E55	Event #55	R	0x0
22	E54	Event #54	R	0x0
21	E53	Event #53	R	0x0

Bits	Field Name	Description	Type	Reset
20	E52	Event #52	R	0x0
19	E51	Event #51	R	0x0
18	E50	Event #50	R	0x0
17	E49	Event #49	R	0x0
16	E48	Event #48	R	0x0
15	E47	Event #47	R	0x0
14	E46	Event #46	R	0x0
13	E45	Event #45	R	0x0
12	E44	Event #44	R	0x0
11	E43	Event #43	R	0x0
10	E42	Event #42	R	0x0
9	E41	Event #41	R	0x0
8	E40	Event #40	R	0x0
7	E39	Event #39	R	0x0
6	E38	Event #38	R	0x0
5	E37	Event #37	R	0x0
4	E36	Event #36	R	0x0
3	E35	Event #35	R	0x0
2	E34	Event #34	R	0x0
1	E33	Event #33	R	0x0
0	E32	Event #32	R	0x0

Table 11-141. Register Call Summary for Register EDMA_TPCC_SERH

EDMA Controller Functional Description
- Null PaRAM Set: [0][1]
- Dummy PaRAM Set: [2]
- Dummy Versus Null Transfer Comparison: [3]
- Region Overview: [4]
- Event Dataflow: [5][6][7]

EDMA Debug Checklist and Programming Tips
- EDMA Debug Checklist: [8][9]

EDMA Register Manual
- EDMA Register Summary: [10][11][12][13]
- EDMA Register Description: [14][15][16][17]

Table 11-142. EDMA_TPCC_SECR

Address Offset	0x0000 1040		
Physical Address	0x4330 1040 0x40D1 1040 0x4151 1040 0x01D1 1040 0x420A 1040	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE_EDMA_TPCC
Description	Secondary Event Clear Register The secondary event clear register is used to clear the status of the EDMA_TPCC_SER registers. CPU write of '1' to the EDMA_TPCC_SECR.En bit clears the EDMA_TPCC_SER register. CPU write of '0' has no effect.		
Type	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E31	E30	E29	E28	E27	E26	E25	E24	E23	E22	E21	E20	E19	E18	E17	E16	E15	E14	E13	E12	E11	E10	E9	E8	E7	E6	E5	E4	E3	E2	E1	E0

Bits	Field Name	Description	Type	Reset
31	E31	Event #31	W	0x0
30	E30	Event #30	W	0x0
29	E29	Event #29	W	0x0
28	E28	Event #28	W	0x0
27	E27	Event #27	W	0x0
26	E26	Event #26	W	0x0
25	E25	Event #25	W	0x0
24	E24	Event #24	W	0x0
23	E23	Event #23	W	0x0
22	E22	Event #22	W	0x0
21	E21	Event #21	W	0x0
20	E20	Event #20	W	0x0
19	E19	Event #19	W	0x0
18	E18	Event #18	W	0x0
17	E17	Event #17	W	0x0
16	E16	Event #16	W	0x0
15	E15	Event #15	W	0x0
14	E14	Event #14	W	0x0
13	E13	Event #13	W	0x0
12	E12	Event #12	W	0x0
11	E11	Event #11	W	0x0
10	E10	Event #10	W	0x0
9	E9	Event #9	W	0x0
8	E8	Event #8	W	0x0
7	E7	Event #7	W	0x0
6	E6	Event #6	W	0x0
5	E5	Event #5	W	0x0
4	E4	Event #4	W	0x0
3	E3	Event #3	W	0x0
2	E2	Event #2	W	0x0
1	E1	Event #1	W	0x0
0	E0	Event #0	W	0x0

Table 11-143. Register Call Summary for Register EDMA_TPCC_SECR

EDMA Controller Functional Description
- Region Overview: [0]

EDMA Register Manual
- EDMA Register Summary: [1][2][3][4]
- EDMA Register Description: [5][6]

Table 11-144. EDMA_TPCC_SECRH

Address Offset	0x0000 1044		
Physical Address	0x4330 1044	**Instance**	SYS_EDMA_TPCC
	0x40D1 1044		DSP1_EDMA_TPCC
	0x4151 1044		DSP2_EDMA_TPCC
	0x01D1 1044		DSP_EDMA_TPCC
	0x420A 1044		EVE_EDMA_TPCC

Table 11-144. EDMA_TPCC_SECRH (continued)

Description	Secondary Event Clear Register (High Part) The secondary event clear register is used to clear the status of the EDMA_TPCC_SERH registers. CPU write of '1' to the EDMA_TPCC_SECRH.En bit clears the EDMA_TPCC_SERH register. CPU write of '0' has no effect.
Type	W

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E63	E62	E61	E60	E59	E58	E57	E56	E55	E54	E53	E52	E51	E50	E49	E48	E47	E46	E45	E44	E43	E42	E41	E40	E39	E38	E37	E36	E35	E34	E33	E32

Bits	Field Name	Description	Type	Reset
31	E63	Event #63	W	0x0
30	E62	Event #62	W	0x0
29	E61	Event #61	W	0x0
28	E60	Event #60	W	0x0
27	E59	Event #59	W	0x0
26	E58	Event #58	W	0x0
25	E57	Event #57	W	0x0
24	E56	Event #56	W	0x0
23	E55	Event #55	W	0x0
22	E54	Event #54	W	0x0
21	E53	Event #53	W	0x0
20	E52	Event #52	W	0x0
19	E51	Event #51	W	0x0
18	E50	Event #50	W	0x0
17	E49	Event #49	W	0x0
16	E48	Event #48	W	0x0
15	E47	Event #47	W	0x0
14	E46	Event #46	W	0x0
13	E45	Event #45	W	0x0
12	E44	Event #44	W	0x0
11	E43	Event #43	W	0x0
10	E42	Event #42	W	0x0
9	E41	Event #41	W	0x0
8	E40	Event #40	W	0x0
7	E39	Event #39	W	0x0
6	E38	Event #38	W	0x0
5	E37	Event #37	W	0x0
4	E36	Event #36	W	0x0
3	E35	Event #35	W	0x0
2	E34	Event #34	W	0x0
1	E33	Event #33	W	0x0
0	E32	Event #32	W	0x0

Table 11-145. Register Call Summary for Register EDMA_TPCC_SECRH

EDMA Controller Functional Description

- Region Overview: [0]

EDMA Register Manual

- EDMA Register Summary: [1][2][3][4]
- EDMA Register Description: [5][6]

Table 11-146. EDMA_TPCC_IER

Address Offset	0x0000 1050		
Physical Address	0x4330 1050 0x40D1 1050 0x4151 1050 0x01D1 1050 0x420A 1050	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE_EDMA_TPCC
Description	Int Enable Register EDMA_TPCC_IER.In is not directly writeable. Interrupts can be enabled via writes to EDMA_TPCC_IESR and can be disabled via writes to EDMA_TPCC_IECR register. EDMA_TPCC_IER. In = 0: EDMA_TPCC_IPR.In is NOT enabled for interrupts. EDMA_TPCC_IER. In = 1: EDMA_TPCC_IPR.In IS enabled for interrupts.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I31	I30	I29	I28	I27	I26	I25	I24	I23	I22	I21	I20	I19	I18	I17	I16	I15	I14	I13	I12	I11	I10	I9	I8	I7	I6	I5	I4	I3	I2	I1	I0

Bits	Field Name	Description	Type	Reset
31	I31	Interrupt associated with TCC #31	R	0x0
30	I30	Interrupt associated with TCC #30	R	0x0
29	I29	Interrupt associated with TCC #29	R	0x0
28	I28	Interrupt associated with TCC #28	R	0x0
27	I27	Interrupt associated with TCC #27	R	0x0
26	I26	Interrupt associated with TCC #26	R	0x0
25	I25	Interrupt associated with TCC #25	R	0x0
24	I24	Interrupt associated with TCC #24	R	0x0
23	I23	Interrupt associated with TCC #23	R	0x0
22	I22	Interrupt associated with TCC #22	R	0x0
21	I21	Interrupt associated with TCC #21	R	0x0
20	I20	Interrupt associated with TCC #20	R	0x0
19	I19	Interrupt associated with TCC #19	R	0x0
18	I18	Interrupt associated with TCC #18	R	0x0
17	I17	Interrupt associated with TCC #17	R	0x0
16	I16	Interrupt associated with TCC #16	R	0x0
15	I15	Interrupt associated with TCC #15	R	0x0
14	I14	Interrupt associated with TCC #14	R	0x0
13	I13	Interrupt associated with TCC #13	R	0x0
12	I12	Interrupt associated with TCC #12	R	0x0
11	I11	Interrupt associated with TCC #11	R	0x0
10	I10	Interrupt associated with TCC #10	R	0x0
9	I9	Interrupt associated with TCC #9	R	0x0
8	I8	Interrupt associated with TCC #8	R	0x0
7	I7	Interrupt associated with TCC #7	R	0x0

Bits	Field Name	Description	Type	Reset
6	I6	Interrupt associated with TCC #6	R	0x0
5	I5	Interrupt associated with TCC #5	R	0x0
4	I4	Interrupt associated with TCC #4	R	0x0
3	I3	Interrupt associated with TCC #3	R	0x0
2	I2	Interrupt associated with TCC #2	R	0x0
1	I1	Interrupt associated with TCC #1	R	0x0
0	I0	Interrupt associated with TCC #0	R	0x0

Table 11-147. Register Call Summary for Register EDMA_TPCC_IER

EDMA Controller Functional Description
- Region Overview: [0]
- Region Interrupts: [1]
- Transfer Completion Interrupts: [2][3][4][5][6][7][8][9][10][11][12][13][14][15][16]
- Interrupt Evaluation Operations: [17]

EDMA Transfer Examples
- Transfer Chaining Examples: [18]
- Setting Up an EDMA Transfer: [19]

EDMA Debug Checklist and Programming Tips
- EDMA Debug Checklist: [20][21]

EDMA Register Manual
- EDMA Register Summary: [22][23][24][25]
- EDMA Register Description: [26][27][28][29][30][31][32][33][34][35][36][37]

Table 11-148. EDMA_TPCC_IERH

Address Offset	0x0000 1054		
Physical Address	0x4330 1054 0x40D1 1054 0x4151 1054 0x01D1 1054 0x420A 1054	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE_EDMA_TPCC
Description	Int Enable Register (High Part) EDMA_TPCC_IERH.In is not directly writeable. Interrupts can be enabled via writes to EDMA_TPCC_IESRH and can be disabled via writes to EDMA_TPCC_IECRH register. EDMA_TPCC_IERH. In = 0: EDMA_TPCC_IPRH.In is NOT enabled for interrupts. EDMA_TPCC_IERH. In = 1: EDMA_TPCC_IPRH.In IS enabled for interrupts.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I63	I62	I61	I60	I59	I58	I57	I56	I55	I54	I53	I52	I51	I50	I49	I48	I47	I46	I45	I44	I43	I42	I41	I40	I39	I38	I37	I36	I35	I34	I33	I32

Bits	Field Name	Description	Type	Reset
31	I63	Interrupt associated with TCC #63	R	0x0
30	I62	Interrupt associated with TCC #62	R	0x0
29	I61	Interrupt associated with TCC #61	R	0x0
28	I60	Interrupt associated with TCC #60	R	0x0
27	I59	Interrupt associated with TCC #59	R	0x0
26	I58	Interrupt associated with TCC #58	R	0x0
25	I57	Interrupt associated with TCC #57	R	0x0
24	I56	Interrupt associated with TCC #56	R	0x0
23	I55	Interrupt associated with TCC #55	R	0x0

Bits	Field Name	Description	Type	Reset
22	I54	Interrupt associated with TCC #54	R	0x0
21	I53	Interrupt associated with TCC #53	R	0x0
20	I52	Interrupt associated with TCC #52	R	0x0
19	I51	Interrupt associated with TCC #51	R	0x0
18	I50	Interrupt associated with TCC #50	R	0x0
17	I49	Interrupt associated with TCC #49	R	0x0
16	I48	Interrupt associated with TCC #48	R	0x0
15	I47	Interrupt associated with TCC #47	R	0x0
14	I46	Interrupt associated with TCC #46	R	0x0
13	I45	Interrupt associated with TCC #45	R	0x0
12	I44	Interrupt associated with TCC #44	R	0x0
11	I43	Interrupt associated with TCC #43	R	0x0
10	I42	Interrupt associated with TCC #42	R	0x0
9	I41	Interrupt associated with TCC #41	R	0x0
8	I40	Interrupt associated with TCC #40	R	0x0
7	I39	Interrupt associated with TCC #39	R	0x0
6	I38	Interrupt associated with TCC #38	R	0x0
5	I37	Interrupt associated with TCC #37	R	0x0
4	I36	Interrupt associated with TCC #36	R	0x0
3	I35	Interrupt associated with TCC #35	R	0x0
2	I34	Interrupt associated with TCC #34	R	0x0
1	I33	Interrupt associated with TCC #33	R	0x0
0	I32	Interrupt associated with TCC #32	R	0x0

Table 11-149. Register Call Summary for Register EDMA_TPCC_IERH

EDMA Controller Functional Description
- Region Overview: [0]
- Transfer Completion Interrupts: [1][2][3][4][5][6][7][8][9]
- Interrupt Evaluation Operations: [10]

EDMA Transfer Examples
- Setting Up an EDMA Transfer: [11]

EDMA Debug Checklist and Programming Tips
- EDMA Debug Checklist: [12][13]

EDMA Register Manual
- EDMA Register Summary: [14][15][16][17]
- EDMA Register Description: [18][19][20][21][22][23][24][25]

Table 11-150. EDMA_TPCC_IECR

Address Offset	0x0000 1058		
Physical Address	0x4330 1058 0x40D1 1058 0x4151 1058 0x01D1 1058 0x420A 1058	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE_EDMA_TPCC
Description	Int Enable Clear Register CPU writes of '1' to the EDMA_TPCC_IECR.In bit causes the EDMA_TPCC_IER.In bit to be cleared. CPU writes of '0' has no effect..		
Type	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I31	I30	I29	I28	I27	I26	I25	I24	I23	I22	I21	I20	I19	I18	I17	I16	I15	I14	I13	I12	I11	I10	I9	I8	I7	I6	I5	I4	I3	I2	I1	I0

Bits	Field Name	Description	Type	Reset
31	I31	Interrupt associated with TCC #31	W	0x0
30	I30	Interrupt associated with TCC #30	W	0x0
29	I29	Interrupt associated with TCC #29	W	0x0
28	I28	Interrupt associated with TCC #28	W	0x0
27	I27	Interrupt associated with TCC #27	W	0x0
26	I26	Interrupt associated with TCC #26	W	0x0
25	I25	Interrupt associated with TCC #25	W	0x0
24	I24	Interrupt associated with TCC #24	W	0x0
23	I23	Interrupt associated with TCC #23	W	0x0
22	I22	Interrupt associated with TCC #22	W	0x0
21	I21	Interrupt associated with TCC #21	W	0x0
20	I20	Interrupt associated with TCC #20	W	0x0
19	I19	Interrupt associated with TCC #19	W	0x0
18	I18	Interrupt associated with TCC #18	W	0x0
17	I17	Interrupt associated with TCC #17	W	0x0
16	I16	Interrupt associated with TCC #16	W	0x0
15	I15	Interrupt associated with TCC #15	W	0x0
14	I14	Interrupt associated with TCC #14	W	0x0
13	I13	Interrupt associated with TCC #13	W	0x0
12	I12	Interrupt associated with TCC #12	W	0x0
11	I11	Interrupt associated with TCC #11	W	0x0
10	I10	Interrupt associated with TCC #10	W	0x0
9	I9	Interrupt associated with TCC #9	W	0x0
8	I8	Interrupt associated with TCC #8	W	0x0
7	I7	Interrupt associated with TCC #7	W	0x0
6	I6	Interrupt associated with TCC #6	W	0x0
5	I5	Interrupt associated with TCC #5	W	0x0
4	I4	Interrupt associated with TCC #4	W	0x0
3	I3	Interrupt associated with TCC #3	W	0x0
2	I2	Interrupt associated with TCC #2	W	0x0
1	I1	Interrupt associated with TCC #1	W	0x0
0	I0	Interrupt associated with TCC #0	W	0x0

Table 11-151. Register Call Summary for Register EDMA_TPCC_IECR

EDMA Controller Functional Description
- Region Overview: [0]
- Transfer Completion Interrupts: [1]

EDMA Register Manual
- EDMA Register Summary: [2][3][4][5]
- EDMA Register Description: [6][7][8][9]

Table 11-152. EDMA_TPCC_IECRH

Address Offset	0x0000 105C		
Physical Address	0x4330 105C 0x40D1 105C 0x4151 105C 0x01D1 105C 0x420A 105C	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE_EDMA_TPCC
Description	Int Enable Clear Register (High Part) CPU write of '1' to the EDMA_TPCC_IECRH.In bit causes the EDMA_TPCC_IERH.In bit to be cleared. CPU write of '0' has no effect..		
Type	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I63	I62	I61	I60	I59	I58	I57	I56	I55	I54	I53	I52	I51	I50	I49	I48	I47	I46	I45	I44	I43	I42	I41	I40	I39	I38	I37	I36	I35	I34	I33	I32

Bits	Field Name	Description	Type	Reset
31	I63	Interrupt associated with TCC #63	W	0x0
30	I62	Interrupt associated with TCC #62	W	0x0
29	I61	Interrupt associated with TCC #61	W	0x0
28	I60	Interrupt associated with TCC #60	W	0x0
27	I59	Interrupt associated with TCC #59	W	0x0
26	I58	Interrupt associated with TCC #58	W	0x0
25	I57	Interrupt associated with TCC #57	W	0x0
24	I56	Interrupt associated with TCC #56	W	0x0
23	I55	Interrupt associated with TCC #55	W	0x0
22	I54	Interrupt associated with TCC #54	W	0x0
21	I53	Interrupt associated with TCC #53	W	0x0
20	I52	Interrupt associated with TCC #52	W	0x0
19	I51	Interrupt associated with TCC #51	W	0x0
18	I50	Interrupt associated with TCC #50	W	0x0
17	I49	Interrupt associated with TCC #49	W	0x0
16	I48	Interrupt associated with TCC #48	W	0x0
15	I47	Interrupt associated with TCC #47	W	0x0
14	I46	Interrupt associated with TCC #46	W	0x0
13	I45	Interrupt associated with TCC #45	W	0x0
12	I44	Interrupt associated with TCC #44	W	0x0
11	I43	Interrupt associated with TCC #43	W	0x0
10	I42	Interrupt associated with TCC #42	W	0x0
9	I41	Interrupt associated with TCC #41	W	0x0
8	I40	Interrupt associated with TCC #40	W	0x0
7	I39	Interrupt associated with TCC #39	W	0x0
6	I38	Interrupt associated with TCC #38	W	0x0
5	I37	Interrupt associated with TCC #37	W	0x0
4	I36	Interrupt associated with TCC #36	W	0x0
3	I35	Interrupt associated with TCC #35	W	0x0
2	I34	Interrupt associated with TCC #34	W	0x0
1	I33	Interrupt associated with TCC #33	W	0x0
0	I32	Interrupt associated with TCC #32	W	0x0

Table 11-153. Register Call Summary for Register EDMA_TPCC_IECRH

EDMA Controller Functional Description

- Region Overview: [0]

EDMA Register Manual

- EDMA Register Summary: [1][2][3][4]
- EDMA Register Description: [5][6][7][8]

Table 11-154. EDMA_TPCC_IESR

Address Offset	0x0000 1060		
Physical Address	0x4330 1060 0x40D1 1060 0x4151 1060 0x01D1 1060 0x420A 1060	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE_EDMA_TPCC
Description	Int Enable Set Register CPU write of '1' to the EDMA_TPCC_IESR.In bit causes the EDMA_TPCC_IESR.In bit to be set. CPU write of '0' has no effect..		
Type	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I31	I30	I29	I28	I27	I26	I25	I24	I23	I22	I21	I20	I19	I18	I17	I16	I15	I14	I13	I12	I11	I10	I9	I8	I7	I6	I5	I4	I3	I2	I1	I0

Bits	Field Name	Description	Type	Reset
31	I31	Interrupt associated with TCC #31	W	0x0
30	I30	Interrupt associated with TCC #30	W	0x0
29	I29	Interrupt associated with TCC #29	W	0x0
28	I28	Interrupt associated with TCC #28	W	0x0
27	I27	Interrupt associated with TCC #27	W	0x0
26	I26	Interrupt associated with TCC #26	W	0x0
25	I25	Interrupt associated with TCC #25	W	0x0
24	I24	Interrupt associated with TCC #24	W	0x0
23	I23	Interrupt associated with TCC #23	W	0x0
22	I22	Interrupt associated with TCC #22	W	0x0
21	I21	Interrupt associated with TCC #21	W	0x0
20	I20	Interrupt associated with TCC #20	W	0x0
19	I19	Interrupt associated with TCC #19	W	0x0
18	I18	Interrupt associated with TCC #18	W	0x0
17	I17	Interrupt associated with TCC #17	W	0x0
16	I16	Interrupt associated with TCC #16	W	0x0
15	I15	Interrupt associated with TCC #15	W	0x0
14	I14	Interrupt associated with TCC #14	W	0x0
13	I13	Interrupt associated with TCC #13	W	0x0
12	I12	Interrupt associated with TCC #12	W	0x0
11	I11	Interrupt associated with TCC #11	W	0x0
10	I10	Interrupt associated with TCC #10	W	0x0
9	I9	Interrupt associated with TCC #9	W	0x0
8	I8	Interrupt associated with TCC #8	W	0x0
7	I7	Interrupt associated with TCC #7	W	0x0
6	I6	Interrupt associated with TCC #6	W	0x0
5	I5	Interrupt associated with TCC #5	W	0x0

Bits	Field Name	Description	Type	Reset
4	I4	Interrupt associated with TCC #4	W	0x0
3	I3	Interrupt associated with TCC #3	W	0x0
2	I2	Interrupt associated with TCC #2	W	0x0
1	I1	Interrupt associated with TCC #1	W	0x0
0	I0	Interrupt associated with TCC #0	W	0x0

Table 11-155. Register Call Summary for Register EDMA_TPCC_IESR

EDMA Controller Functional Description
- Region Overview: [0]
- Transfer Completion Interrupts: [1]

EDMA Transfer Examples
- Setting Up an EDMA Transfer: [2]

EDMA Register Manual
- EDMA Register Summary: [3][4][5][6]
- EDMA Register Description: [7][8][9][10][11][12]

Table 11-156. EDMA_TPCC_IESRH

Address Offset	0x0000 1064		
Physical Address	0x4330 1064 0x40D1 1064 0x4151 1064 0x01D1 1064 0x420A 1064	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE_EDMA_TPCC
Description	Int Enable Set Register (High Part) CPU write of '1' to the EDMA_TPCC_IESRH.In bit causes the EDMA_TPCC_IESRH.In bit to be set. CPU write of '0' has no effect..		
Type	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I63	I62	I61	I60	I59	I58	I57	I56	I55	I54	I53	I52	I51	I50	I49	I48	I47	I46	I45	I44	I43	I42	I41	I40	I39	I38	I37	I36	I35	I34	I33	I32

Bits	Field Name	Description	Type	Reset
31	I63	Interrupt associated with TCC #63	W	0x0
30	I62	Interrupt associated with TCC #62	W	0x0
29	I61	Interrupt associated with TCC #61	W	0x0
28	I60	Interrupt associated with TCC #60	W	0x0
27	I59	Interrupt associated with TCC #59	W	0x0
26	I58	Interrupt associated with TCC #58	W	0x0
25	I57	Interrupt associated with TCC #57	W	0x0
24	I56	Interrupt associated with TCC #56	W	0x0
23	I55	Interrupt associated with TCC #55	W	0x0
22	I54	Interrupt associated with TCC #54	W	0x0
21	I53	Interrupt associated with TCC #53	W	0x0
20	I52	Interrupt associated with TCC #52	W	0x0
19	I51	Interrupt associated with TCC #51	W	0x0
18	I50	Interrupt associated with TCC #50	W	0x0
17	I49	Interrupt associated with TCC #49	W	0x0
16	I48	Interrupt associated with TCC #48	W	0x0
15	I47	Interrupt associated with TCC #47	W	0x0

Bits	Field Name	Description	Type	Reset
14	I46	Interrupt associated with TCC #46	W	0x0
13	I45	Interrupt associated with TCC #45	W	0x0
12	I44	Interrupt associated with TCC #44	W	0x0
11	I43	Interrupt associated with TCC #43	W	0x0
10	I42	Interrupt associated with TCC #42	W	0x0
9	I41	Interrupt associated with TCC #41	W	0x0
8	I40	Interrupt associated with TCC #40	W	0x0
7	I39	Interrupt associated with TCC #39	W	0x0
6	I38	Interrupt associated with TCC #38	W	0x0
5	I37	Interrupt associated with TCC #37	W	0x0
4	I36	Interrupt associated with TCC #36	W	0x0
3	I35	Interrupt associated with TCC #35	W	0x0
2	I34	Interrupt associated with TCC #34	W	0x0
1	I33	Interrupt associated with TCC #33	W	0x0
0	I32	Interrupt associated with TCC #32	W	0x0

Table 11-157. Register Call Summary for Register EDMA_TPCC_IESRH

EDMA Controller Functional Description • Region Overview: [0]
EDMA Transfer Examples • Setting Up an EDMA Transfer: [1]
EDMA Register Manual • EDMA Register Summary: [2][3][4][5] • EDMA Register Description: [6][7][8][9][10][11]

Table 11-158. EDMA_TPCC_IPR

Address Offset	0x0000 1068		
Physical Address	0x4330 1068 0x40D1 1068 0x4151 1068 0x01D1 1068 0x420A 1068	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE_EDMA_TPCC
Description	Interrupt Pending Register EDMA_TPCC_IPR.In bit is set when a interrupt completion code with TCC of N is detected. EDMA_TPCC_IPR.In bit is cleared via software by writing a '1' to EDMA_TPCC_ICR.In bit.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I31	I30	I29	I28	I27	I26	I25	I24	I23	I22	I21	I20	I19	I18	I17	I16	I15	I14	I13	I12	I11	I10	I9	I8	I7	I6	I5	I4	I3	I2	I1	I0

Bits	Field Name	Description	Type	Reset
31	I31	Interrupt associated with TCC #31	R	0x0
30	I30	Interrupt associated with TCC #30	R	0x0
29	I29	Interrupt associated with TCC #29	R	0x0
28	I28	Interrupt associated with TCC #28	R	0x0
27	I27	Interrupt associated with TCC #27	R	0x0
26	I26	Interrupt associated with TCC #26	R	0x0
25	I25	Interrupt associated with TCC #25	R	0x0
24	I24	Interrupt associated with TCC #24	R	0x0
23	I23	Interrupt associated with TCC #23	R	0x0

Bits	Field Name	Description	Type	Reset
22	I22	Interrupt associated with TCC #22	R	0x0
21	I21	Interrupt associated with TCC #21	R	0x0
20	I20	Interrupt associated with TCC #20	R	0x0
19	I19	Interrupt associated with TCC #19	R	0x0
18	I18	Interrupt associated with TCC #18	R	0x0
17	I17	Interrupt associated with TCC #17	R	0x0
16	I16	Interrupt associated with TCC #16	R	0x0
15	I15	Interrupt associated with TCC #15	R	0x0
14	I14	Interrupt associated with TCC #14	R	0x0
13	I13	Interrupt associated with TCC #13	R	0x0
12	I12	Interrupt associated with TCC #12	R	0x0
11	I11	Interrupt associated with TCC #11	R	0x0
10	I10	Interrupt associated with TCC #10	R	0x0
9	I9	Interrupt associated with TCC #9	R	0x0
8	I8	Interrupt associated with TCC #8	R	0x0
7	I7	Interrupt associated with TCC #7	R	0x0
6	I6	Interrupt associated with TCC #6	R	0x0
5	I5	Interrupt associated with TCC #5	R	0x0
4	I4	Interrupt associated with TCC #4	R	0x0
3	I3	Interrupt associated with TCC #3	R	0x0
2	I2	Interrupt associated with TCC #2	R	0x0
1	I1	Interrupt associated with TCC #1	R	0x0
0	I0	Interrupt associated with TCC #0	R	0x0

Table 11-159. Register Call Summary for Register EDMA_TPCC_IPR

EDMA Controller Functional Description
- Dummy Versus Null Transfer Comparison: [0]
- Completion of a DMA Transfer: [1]
- Dummy or Null Completion: [2]
- Region Overview: [3]
- Transfer Completion Interrupts: [4][5][6][7][8][9][10][11][12][13][14][15][16][17][18][19][20][21][22][23][24][25][26][27][28][29][30][31][32][33][34][35]
- EDMA Interrupt Servicing: [36][37][38][39][40]
- Interrupt Servicing: [41][42][43][44][45][46]
- Interrupt Servicing: [47][48][49][50][51]
- Interrupt Evaluation Operations: [52]
- Event Dataflow: [53][54]

EDMA Transfer Examples
- Ping-Pong Buffering: [55][56][57]
- Transfer Chaining Examples: [58]
- Setting Up an EDMA Transfer: [59][60][61][62]

EDMA Debug Checklist and Programming Tips
- EDMA Debug Checklist: [63][64]

EDMA Register Manual
- EDMA Register Summary: [65][66][67][68]
- EDMA Register Description: [69][70][71][72][73][74][75][76][77][78][79][80][81][82][83]

Table 11-160. EDMA_TPCC_IPRH

Address Offset	0x0000 106C		
Physical Address	0x4330 106C 0x40D1 106C 0x4151 106C 0x01D1 106C 0x420A 106C	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE_EDMA_TPCC
Description	Interrupt Pending Register (High Part) EDMA_TPCC_IPRH.In bit is set when a interrupt completion code with TCC of N is detected. EDMA_TPCC_IPRH. In bit is cleared via software by writing a '1' to EDMA_TPCC_ICRH.In bit.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I63	I62	I61	I60	I59	I58	I57	I56	I55	I54	I53	I52	I51	I50	I49	I48	I47	I46	I45	I44	I43	I42	I41	I40	I39	I38	I37	I36	I35	I34	I33	I32

Bits	Field Name	Description	Type	Reset
31	I63	Interrupt associated with TCC #63	R	0x0
30	I62	Interrupt associated with TCC #62	R	0x0
29	I61	Interrupt associated with TCC #61	R	0x0
28	I60	Interrupt associated with TCC #60	R	0x0
27	I59	Interrupt associated with TCC #59	R	0x0
26	I58	Interrupt associated with TCC #58	R	0x0
25	I57	Interrupt associated with TCC #57	R	0x0
24	I56	Interrupt associated with TCC #56	R	0x0
23	I55	Interrupt associated with TCC #55	R	0x0
22	I54	Interrupt associated with TCC #54	R	0x0
21	I53	Interrupt associated with TCC #53	R	0x0
20	I52	Interrupt associated with TCC #52	R	0x0
19	I51	Interrupt associated with TCC #51	R	0x0
18	I50	Interrupt associated with TCC #50	R	0x0
17	I49	Interrupt associated with TCC #49	R	0x0
16	I48	Interrupt associated with TCC #48	R	0x0
15	I47	Interrupt associated with TCC #47	R	0x0
14	I46	Interrupt associated with TCC #46	R	0x0
13	I45	Interrupt associated with TCC #45	R	0x0
12	I44	Interrupt associated with TCC #44	R	0x0
11	I43	Interrupt associated with TCC #43	R	0x0
10	I42	Interrupt associated with TCC #42	R	0x0
9	I41	Interrupt associated with TCC #41	R	0x0
8	I40	Interrupt associated with TCC #40	R	0x0
7	I39	Interrupt associated with TCC #39	R	0x0
6	I38	Interrupt associated with TCC #38	R	0x0
5	I37	Interrupt associated with TCC #37	R	0x0
4	I36	Interrupt associated with TCC #36	R	0x0
3	I35	Interrupt associated with TCC #35	R	0x0
2	I34	Interrupt associated with TCC #34	R	0x0
1	I33	Interrupt associated with TCC #33	R	0x0
0	I32	Interrupt associated with TCC #32	R	0x0

Table 11-161. Register Call Summary for Register EDMA_TPCC_IPRH

EDMA Controller Functional Description

- Dummy Versus Null Transfer Comparison: [0]
- Dummy or Null Completion: [1]
- Region Overview: [2]
- Transfer Completion Interrupts: [3][4][5][6][7][8][9][10][11][12][13][14][15][16][17][18][19][20][21][22][23][24]
- EDMA Interrupt Servicing: [25][26][27][28][29]
- Interrupt Servicing: [30][31][32][33][34][35]
- Interrupt Servicing: [36][37][38][39][40]
- Interrupt Evaluation Operations: [41]
- Event Dataflow: [42][43]

EDMA Transfer Examples

- Setting Up an EDMA Transfer: [44][45][46][47]

EDMA Debug Checklist and Programming Tips

- EDMA Debug Checklist: [48][49]

EDMA Register Manual

- EDMA Register Summary: [50][51][52][53]
- EDMA Register Description: [54][55][56][57][58][59][60][61][62][63][64][65]

Table 11-162. EDMA_TPCC_ICR

Address Offset	0x0000 1070		
Physical Address	0x4330 1070 0x40D1 1070 0x4151 1070 0x01D1 1070 0x420A 1070	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE_EDMA_TPCC
Description	Interrupt Clear Register CPU write of '1' to the EDMA_TPCC_ICR.In bit causes the EDMA_TPCC_IPR.In bit to be cleared. CPU write of '0' has no effect. All EDMA_TPCC_IPR.In bits must be cleared before additional interrupts will be asserted by CC.		
Type	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I31	I30	I29	I28	I27	I26	I25	I24	I23	I22	I21	I20	I19	I18	I17	I16	I15	I14	I13	I12	I11	I10	I9	I8	I7	I6	I5	I4	I3	I2	I1	I0

Bits	Field Name	Description	Type	Reset
31	I31	Interrupt associated with TCC #31	W	0x0
30	I30	Interrupt associated with TCC #30	W	0x0
29	I29	Interrupt associated with TCC #29	W	0x0
28	I28	Interrupt associated with TCC #28	W	0x0
27	I27	Interrupt associated with TCC #27	W	0x0
26	I26	Interrupt associated with TCC #26	W	0x0
25	I25	Interrupt associated with TCC #25	W	0x0
24	I24	Interrupt associated with TCC #24	W	0x0
23	I23	Interrupt associated with TCC #23	W	0x0
22	I22	Interrupt associated with TCC #22	W	0x0
21	I21	Interrupt associated with TCC #21	W	0x0
20	I20	Interrupt associated with TCC #20	W	0x0
19	I19	Interrupt associated with TCC #19	W	0x0
18	I18	Interrupt associated with TCC #18	W	0x0
17	I17	Interrupt associated with TCC #17	W	0x0
16	I16	Interrupt associated with TCC #16	W	0x0

Bits	Field Name	Description	Type	Reset
15	I15	Interrupt associated with TCC #15	W	0x0
14	I14	Interrupt associated with TCC #14	W	0x0
13	I13	Interrupt associated with TCC #13	W	0x0
12	I12	Interrupt associated with TCC #12	W	0x0
11	I11	Interrupt associated with TCC #11	W	0x0
10	I10	Interrupt associated with TCC #10	W	0x0
9	I9	Interrupt associated with TCC #9	W	0x0
8	I8	Interrupt associated with TCC #8	W	0x0
7	I7	Interrupt associated with TCC #7	W	0x0
6	I6	Interrupt associated with TCC #6	W	0x0
5	I5	Interrupt associated with TCC #5	W	0x0
4	I4	Interrupt associated with TCC #4	W	0x0
3	I3	Interrupt associated with TCC #3	W	0x0
2	I2	Interrupt associated with TCC #2	W	0x0
1	I1	Interrupt associated with TCC #1	W	0x0
0	I0	Interrupt associated with TCC #0	W	0x0

Table 11-163. Register Call Summary for Register EDMA_TPCC_ICR

EDMA Controller Functional Description
- Region Overview: [0]
- Transfer Completion Interrupts: [1][2]
- Interrupt Servicing: [3]

EDMA Transfer Examples
- Setting Up an EDMA Transfer: [4][5]

EDMA Register Manual
- EDMA Register Summary: [6][7][8][9]
- EDMA Register Description: [10][11][12][13]

Table 11-164. EDMA_TPCC_ICRH

Address Offset	0x0000 1074		
Physical Address	0x4330 1074 0x40D1 1074 0x4151 1074 0x01D1 1074 0x420A 1074	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE_EDMA_TPCC
Description	Interrupt Clear Register (High Part) CPU write of '1' to the EDMA_TPCC_ICRH.In bit causes the EDMA_TPCC_IPRH.In bit to be cleared. CPU write of '0' has no effect. All EDMA_TPCC_IPRH.In bits must be cleared before additional interrupts will be asserted by CC.		
Type	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I63	I62	I61	I60	I59	I58	I57	I56	I55	I54	I53	I52	I51	I50	I49	I48	I47	I46	I45	I44	I43	I42	I41	I40	I39	I38	I37	I36	I35	I34	I33	I32

Bits	Field Name	Description	Type	Reset
31	I63	Interrupt associated with TCC #63	W	0x0
30	I62	Interrupt associated with TCC #62	W	0x0
29	I61	Interrupt associated with TCC #61	W	0x0
28	I60	Interrupt associated with TCC #60	W	0x0
27	I59	Interrupt associated with TCC #59	W	0x0

Bits	Field Name	Description	Type	Reset
26	I58	Interrupt associated with TCC #58	W	0x0
25	I57	Interrupt associated with TCC #57	W	0x0
24	I56	Interrupt associated with TCC #56	W	0x0
23	I55	Interrupt associated with TCC #55	W	0x0
22	I54	Interrupt associated with TCC #54	W	0x0
21	I53	Interrupt associated with TCC #53	W	0x0
20	I52	Interrupt associated with TCC #52	W	0x0
19	I51	Interrupt associated with TCC #51	W	0x0
18	I50	Interrupt associated with TCC #50	W	0x0
17	I49	Interrupt associated with TCC #49	W	0x0
16	I48	Interrupt associated with TCC #48	W	0x0
15	I47	Interrupt associated with TCC #47	W	0x0
14	I46	Interrupt associated with TCC #46	W	0x0
13	I45	Interrupt associated with TCC #45	W	0x0
12	I44	Interrupt associated with TCC #44	W	0x0
11	I43	Interrupt associated with TCC #43	W	0x0
10	I42	Interrupt associated with TCC #42	W	0x0
9	I41	Interrupt associated with TCC #41	W	0x0
8	I40	Interrupt associated with TCC #40	W	0x0
7	I39	Interrupt associated with TCC #39	W	0x0
6	I38	Interrupt associated with TCC #38	W	0x0
5	I37	Interrupt associated with TCC #37	W	0x0
4	I36	Interrupt associated with TCC #36	W	0x0
3	I35	Interrupt associated with TCC #35	W	0x0
2	I34	Interrupt associated with TCC #34	W	0x0
1	I33	Interrupt associated with TCC #33	W	0x0
0	I32	Interrupt associated with TCC #32	W	0x0

Table 11-165. Register Call Summary for Register EDMA_TPCC_ICRH

EDMA Controller Functional Description
- Region Overview: [0]
- Transfer Completion Interrupts: [1]
- Interrupt Servicing: [2]

EDMA Transfer Examples
- Setting Up an EDMA Transfer: [3][4]

EDMA Register Manual
- EDMA Register Summary: [5][6][7][8]
- EDMA Register Description: [9][10][11][12]

Table 11-166. EDMA_TPCC_IEVAL

Address Offset	0x0000 1078		
Physical Address	0x4330 1078 0x40D1 1078 0x4151 1078 0x01D1 1078 0x420A 1078	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE_EDMA_TPCC
Description	Interrupt Eval Register		
Type	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														SET	EVAL

Bits	Field Name	Description	Type	Reset
31:2	RESERVED	Reserved	R	0x0
1	SET	Interrupt Set: CPU write of '1' to the SETn bit causes the tpcc_intN output signal to be pulsed egardless of state of interrupts enable (IERn) and status (EDMA_TPCC_IPRn). CPU write of '0' has no effect.	W	0x0
0	EVAL	Interrupt Evaluate: CPU write of '1' to the EVALn bit causes the tpcc_intN output signal to be pulsed if any enabled interrupts (IERn) are still pending (EDMA_TPCC_IPRn). CPU write of '0' has no effect.	W	0x0

Table 11-167. Register Call Summary for Register EDMA_TPCC_IEVAL

EDMA Controller Functional Description

- Region Overview: [0][1]
- Interrupt Servicing: [2][3]
- Interrupt Evaluation Operations: [4][5][6][7][8]
- Error Interrupts: [9][10]

EDMA Register Manual

- EDMA Register Summary: [11][12][13][14]

Table 11-168. EDMA_TPCC_QER

Address Offset	0x0000 1080		
Physical Address	0x4330 1080 0x40D1 1080 0x4151 1080 0x01D1 1080 0x420A 1080	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE_EDMA_TPCC
Description	QDMA Event Register: If EDMA_TPCC_QER.En bit is set, then the corresponding QDMA channel is prioritized vs. other qdma events for submission to the TC. EDMA_TPCC_QER.En bit is set when a vbus write byte matches the address defined in the QCHMAPn register. EDMA_TPCC_QER.En bit is cleared when the corresponding event is prioritized and serviced. EDMA_TPCC_QER.En is also cleared when user writes a '1' to the EDMA_TPCC_QSECR.En bit. If the EDMA_TPCC_QER.En bit is already set and a new QDMA event is detected due to user write to QDMA trigger location and EDMA_TPCC_QEER register is set, then the corresponding bit in the QDMA Event Missed Register is set.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								E7	E6	E5	E4	E3	E2	E1	E0

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Reserved	R	0x0
7	E7	Event #7	R	0x0
6	E6	Event #6	R	0x0
5	E5	Event #5	R	0x0
4	E4	Event #4	R	0x0
3	E3	Event #3	R	0x0
2	E2	Event #2	R	0x0

Bits	Field Name	Description	Type	Reset
1	E1	Event #1	R	0x0
0	E0	Event #0	R	0x0

Table 11-169. Register Call Summary for Register EDMA_TPCC_QER

EDMA Controller Functional Description
- Dummy Versus Null Transfer Comparison: [0]
- Linking Transfers: [1]
- QDMA Channels: [2][3][4][5]
- Region Overview: [6]
- Event Dataflow: [7][8]
- Channel Priority: [9]

EDMA Register Manual
- EDMA Register Summary: [10][11][12][13]
- EDMA Register Description: [14][15][16][17][18][19][20][21][22][23][24][25][26][27][28][29][30][31][32][33][34][35]

Table 11-170. EDMA_TPCC_QEER

Address Offset	0x0000 1084		
Physical Address	0x4330 1084 0x40D1 1084 0x4151 1084 0x01D1 1084 0x420A 1084	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE_EDMA_TPCC
Description	QDMA Event Enable Register Enabled/disabled QDMA address comparator for QDMA Channel N. EDMA_TPCC_QEER.En is not directly writeable. QDMA channels can be enabled via writes to EDMA_TPCC_QEESR and can be disabled via writes to EDMA_TPCC_QEECR register. EDMA_TPCC_QEER.En = 1, The corresponding QDMA channel comparator is enabled and Events will be recognized and latched in EDMA_TPCC_QER.En. EDMA_TPCC_QEER.En = 0, The corresponding QDMA channel comparator is disabled. Events will not be recognized/latched in EDMA_TPCC_QER.En.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								E7	E6	E5	E4	E3	E2	E1	E0

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Reserved	R Return 0's	0x0
7	E7	Event #7	R	0x0
6	E6	Event #6	R	0x0
5	E5	Event #5	R	0x0
4	E4	Event #4	R	0x0
3	E3	Event #3	R	0x0
2	E2	Event #2	R	0x0
1	E1	Event #1	R	0x0
0	E0	Event #0	R	0x0

Table 11-171. Register Call Summary for Register EDMA_TPCC_QEER

EDMA Controller Functional Description
- QDMA Channels: [0][1]
- Region Overview: [2]
- Channel Controller Regions: [3]

Table 11-171. Register Call Summary for Register EDMA_TPCC_QEER (continued)

EDMA Transfer Examples • Setting Up an EDMA Transfer: [4]
EDMA Debug Checklist and Programming Tips • EDMA Debug Checklist: [5]
EDMA Register Manual • EDMA Register Summary: [6][7][8][9] • EDMA Register Description: [10][11][12][13][14][15][16][17][18][19]

Table 11-172. EDMA_TPCC_QEECR

Address Offset	0x0000 1088		
Physical Address	0x4330 1088 0x40D1 1088 0x4151 1088 0x01D1 1088 0x420A 1088	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE_EDMA_TPCC
Description	QDMA Event Enable Clear Register CPU write of '1' to the EDMA_TPCC_QEECR.En bit causes the EDMA_TPCC_QEER.En bit to be cleared. CPU write of '0' has no effect..		
Type	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								E7	E6	E5	E4	E3	E2	E1	E0

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Reserved	R	0x0
7	E7	Event #7	W	0x0
6	E6	Event #6	W	0x0
5	E5	Event #5	W	0x0
4	E4	Event #4	W	0x0
3	E3	Event #3	W	0x0
2	E2	Event #2	W	0x0
1	E1	Event #1	W	0x0
0	E0	Event #0	W	0x0

Table 11-173. Register Call Summary for Register EDMA_TPCC_QEECR

EDMA Controller Functional Description • Region Overview: [0]
EDMA Register Manual • EDMA Register Summary: [1][2][3][4] • EDMA Register Description: [5][6][7][8]

Table 11-174. EDMA_TPCC_QEESR

Address Offset	0x0000 108C		
Physical Address	0x4330 108C 0x40D1 108C 0x4151 108C 0x01D1 108C 0x420A 108C	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE_EDMA_TPCC

Table 11-174. EDMA_TPCC_QEESR (continued)

Description	QDMA Event Enable Set Register CPU write of '1' to the EDMA_TPCC_QEESR.En bit causes the EDMA_TPCC_QEESR.En bit to be set. CPU write of '0' has no effect..
Type	W

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								E7	E6	E5	E4	E3	E2	E1	E0

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Reserved	R	0x0
7	E7	Event #7	W	0x0
6	E6	Event #6	W	0x0
5	E5	Event #5	W	0x0
4	E4	Event #4	W	0x0
3	E3	Event #3	W	0x0
2	E2	Event #2	W	0x0
1	E1	Event #1	W	0x0
0	E0	Event #0	W	0x0

Table 11-175. Register Call Summary for Register EDMA_TPCC_QEESR

EDMA Controller Functional Description
- Region Overview: [0]
- Channel Controller Regions: [1]

EDMA Transfer Examples
- Setting Up an EDMA Transfer: [2]

EDMA Register Manual
- EDMA Register Summary: [3][4][5][6]
- EDMA Register Description: [7][8][9][10][11][12]

Table 11-176. EDMA_TPCC_QSER

Address Offset	0x0000 1090		
Physical Address	0x4330 1090 0x40D1 1090 0x4151 1090 0x01D1 1090 0x420A 1090	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE_EDMA_TPCC
Description	QDMA Secondary Event Register The QDMA secondary event register is used along with the QDMA Event Register (EDMA_TPCC_QER) to provide information on the state of a QDMA Event. En = 0 : Event is not currently in the Event Queue. En = 1 : Event is currently stored in Event Queue. Event arbiter will not prioritize additional events.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								E7	E6	E5	E4	E3	E2	E1	E0

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Reserved	R Return 0's	0x0
7	E7	Event #7	R	0x0
6	E6	Event #6	R	0x0
5	E5	Event #5	R	0x0
4	E4	Event #4	R	0x0
3	E3	Event #3	R	0x0
2	E2	Event #2	R	0x0
1	E1	Event #1	R	0x0
0	E0	Event #0	R	0x0

Table 11-177. Register Call Summary for Register EDMA_TPCC_QSER

EDMA Controller Functional Description
- Null PaRAM Set: [0][1]
- Dummy PaRAM Set: [2]
- Dummy Versus Null Transfer Comparison: [3]
- Event Dataflow: [4]

EDMA Debug Checklist and Programming Tips
- EDMA Debug Checklist: [5][6][7]

EDMA Register Manual
- EDMA Register Summary: [8][9][10][11]
- EDMA Register Description: [12][13][14][15]

Table 11-178. EDMA_TPCC_QSECR

Address Offset	0x0000 1094		
Physical Address	0x4330 1094 0x40D1 1094 0x4151 1094 0x01D1 1094 0x420A 1094	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE_EDMA_TPCC
Description	QDMA Secondary Event Clear Register The secondary event clear register is used to clear the status of the EDMA_TPCC_QSER and EDMA_TPCC_QER register (note that this is slightly different than the EDMA_TPCC_SER operation, which does not clear the EDMA_TPCC_ER.En register). CPU write of '1' to the EDMA_TPCC_QSECR.En bit clears the EDMA_TPCC_QSER.En and EDMA_TPCC_QER.En register fields. CPU write of '0' has no effect..		
Type	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								E7	E6	E5	E4	E3	E2	E1	E0

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Reserved	R	0x0
7	E7	Event #7	W	0x0
6	E6	Event #6	W	0x0
5	E5	Event #5	W	0x0
4	E4	Event #4	W	0x0
3	E3	Event #3	W	0x0
2	E2	Event #2	W	0x0
1	E1	Event #1	W	0x0

Bits	Field Name	Description	Type	Reset
0	E0	Event #0	W	0x0

Table 11-179. Register Call Summary for Register EDMA_TPCC_QSECR

EDMA Register Manual

- EDMA Register Summary: [0][1][2][3]
- EDMA Register Description: [4][5][6][7]

Table 11-180. EDMA_TPCC_ER_RN_k

Address Offset	0x0000 2000 + (0x200 * k)		
Physical Address	0x4330 2000 + (0x200 * k) 0x40D1 2000 + (0x200 * k) 0x4151 2000 + (0x200 * k) 0x01D1 2000 + (0x200 * k) 0x420A 2000 + (0x200 * k)	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE_EDMA_TPCC
Description	Event Register If EDMA_TPCC_ER.En bit is set and the EDMA_TPCC_EER.En bit is also set, then the corresponding DMA channel is prioritized vs. other pending DMA events for submission to the TC. EDMA_TPCC_ER.En bit is set when the input event #n transitions from inactive (low) to active (high), regardless of the state of EDMA_TPCC_EER.En bit. EDMA_TPCC_ER.En bit is cleared when the corresponding event is prioritized and serviced. If the EDMA_TPCC_ER.En bit is already set and a new inactive to active transition is detected on the input event #n input AND the corresponding bit in the EDMA_TPCC_EER register is set, then the corresponding bit in the Event Missed Register is set. Event N can be cleared via sw by writing a '1' to the EDMA_TPCC_ECR pseudo-register.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E31	E30	E29	E28	E27	E26	E25	E24	E23	E22	E21	E20	E19	E18	E17	E16	E15	E14	E13	E12	E11	E10	E9	E8	E7	E6	E5	E4	E3	E2	E1	E0

Bits	Field Name	Description	Type	Reset
31	E31	Event #31	R	0x0
30	E30	Event #30	R	0x0
29	E29	Event #29	R	0x0
28	E28	Event #28	R	0x0
27	E27	Event #27	R	0x0
26	E26	Event #26	R	0x0
25	E25	Event #25	R	0x0
24	E24	Event #24	R	0x0
23	E23	Event #23	R	0x0
22	E22	Event #22	R	0x0
21	E21	Event #21	R	0x0
20	E20	Event #20	R	0x0
19	E19	Event #19	R	0x0
18	E18	Event #18	R	0x0
17	E17	Event #17	R	0x0
16	E16	Event #16	R	0x0
15	E15	Event #15	R	0x0
14	E14	Event #14	R	0x0
13	E13	Event #13	R	0x0
12	E12	Event #12	R	0x0
11	E11	Event #11	R	0x0

Bits	Field Name	Description	Type	Reset
10	E10	Event #10	R	0x0
9	E9	Event #9	R	0x0
8	E8	Event #8	R	0x0
7	E7	Event #7	R	0x0
6	E6	Event #6	R	0x0
5	E5	Event #5	R	0x0
4	E4	Event #4	R	0x0
3	E3	Event #3	R	0x0
2	E2	Event #2	R	0x0
1	E1	Event #1	R	0x0
0	E0	Event #0	R	0x0

Table 11-181. Register Call Summary for Register EDMA_TPCC_ER_RN_k

EDMA Register Manual
- EDMA Register Summary: [0][1][2][3]

Table 11-182. EDMA_TPCC_ERH_RN_k

Address Offset	0x0000 2004 + (0x200 * k)		
Physical Address	0x4330 2004 + (0x200 * k) 0x40D1 2004 + (0x200 * k) 0x4151 2004 + (0x200 * k) 0x01D1 2004 + (0x200 * k) 0x420A 2004 + (0x200 * k)	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE_EDMA_TPCC
Description	Event Register (High Part) If EDMA_TPCC_ERH.En bit is set and the EDMA_TPCC_EERH.En bit is also set, then the corresponding DMA channel is prioritized vs. other pending DMA events for submission to the TC. EDMA_TPCC_ERH.En bit is set when the input event #n transitions from inactive (low) to active (high), regardless of the state of EERH.En bit. EDMA_TPCC_ER.En bit is cleared when the corresponding event is prioritized and serviced. If the EDMA_TPCC_ERH.En bit is already set and a new inactive to active transition is detected on the input event #n input AND the corresponding bit in the EDMA_TPCC_EERH register is set, then the corresponding bit in the Event Missed Register is set. Event N can be cleared via sw by writing a '1' to the EDMA_TPCC_ECRH pseudo-register.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E63	E62	E61	E60	E59	E58	E57	E56	E55	E54	E53	E52	E51	E50	E49	E48	E47	E46	E45	E44	E43	E42	E41	E40	E39	E38	E37	E36	E35	E34	E33	E32

Bits	Field Name	Description	Type	Reset
31	E63	Event #63	R	0x0
30	E62	Event #62	R	0x0
29	E61	Event #61	R	0x0
28	E60	Event #60	R	0x0
27	E59	Event #59	R	0x0
26	E58	Event #58	R	0x0
25	E57	Event #57	R	0x0
24	E56	Event #56	R	0x0
23	E55	Event #55	R	0x0
22	E54	Event #54	R	0x0
21	E53	Event #53	R	0x0
20	E52	Event #52	R	0x0

Bits	Field Name	Description	Type	Reset
19	E51	Event #51	R	0x0
18	E50	Event #50	R	0x0
17	E49	Event #49	R	0x0
16	E48	Event #48	R	0x0
15	E47	Event #47	R	0x0
14	E46	Event #46	R	0x0
13	E45	Event #45	R	0x0
12	E44	Event #44	R	0x0
11	E43	Event #43	R	0x0
10	E42	Event #42	R	0x0
9	E41	Event #41	R	0x0
8	E40	Event #40	R	0x0
7	E39	Event #39	R	0x0
6	E38	Event #38	R	0x0
5	E37	Event #37	R	0x0
4	E36	Event #36	R	0x0
3	E35	Event #35	R	0x0
2	E34	Event #34	R	0x0
1	E33	Event #33	R	0x0
0	E32	Event #32	R	0x0

Table 11-183. Register Call Summary for Register EDMA_TPCC_ERH_RN_k

EDMA Register Manual
- EDMA Register Summary: [0][1][2][3]

Table 11-184. EDMA_TPCC_ECR_RN_k

Address Offset	0x0000 2008 + (0x200 * k)		
Physical Address	0x4330 2008 + (0x200 * k) 0x40D1 2008 + (0x200 * k) 0x4151 2008 + (0x200 * k) 0x01D1 2008 + (0x200 * k) 0x420A 2008 + (0x200 * k)	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE_EDMA_TPCC
Description	Event Clear Register CPU write of '1' to the EDMA_TPCC_ECR.En bit causes the EDMA_TPCC_ER.En bit to be cleared. CPU write of '0' has no effect.		
Type	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E31	E30	E29	E28	E27	E26	E25	E24	E23	E22	E21	E20	E19	E18	E17	E16	E15	E14	E13	E12	E11	E10	E9	E8	E7	E6	E5	E4	E3	E2	E1	E0

Bits	Field Name	Description	Type	Reset
31	E31	Event #31	W	0x0
30	E30	Event #30	W	0x0
29	E29	Event #29	W	0x0
28	E28	Event #28	W	0x0
27	E27	Event #27	W	0x0
26	E26	Event #26	W	0x0
25	E25	Event #25	W	0x0

Bits	Field Name	Description	Type	Reset
24	E24	Event #24	W	0x0
23	E23	Event #23	W	0x0
22	E22	Event #22	W	0x0
21	E21	Event #21	W	0x0
20	E20	Event #20	W	0x0
19	E19	Event #19	W	0x0
18	E18	Event #18	W	0x0
17	E17	Event #17	W	0x0
16	E16	Event #16	W	0x0
15	E15	Event #15	W	0x0
14	E14	Event #14	W	0x0
13	E13	Event #13	W	0x0
12	E12	Event #12	W	0x0
11	E11	Event #11	W	0x0
10	E10	Event #10	W	0x0
9	E9	Event #9	W	0x0
8	E8	Event #8	W	0x0
7	E7	Event #7	W	0x0
6	E6	Event #6	W	0x0
5	E5	Event #5	W	0x0
4	E4	Event #4	W	0x0
3	E3	Event #3	W	0x0
2	E2	Event #2	W	0x0
1	E1	Event #1	W	0x0
0	E0	Event #0	W	0x0

Table 11-185. Register Call Summary for Register EDMA_TPCC_ECR_RN_k

EDMA Register Manual

- EDMA Register Summary: [0][1][2][3]

Table 11-186. EDMA_TPCC_ECRH_RN_k

Address Offset	0x0000 200C + (0x200 * k)		
Physical Address	0x4330 200C + (0x200 * k) 0x40D1 200C + (0x200 * k) 0x4151 200C + (0x200 * k) 0x01D1 200C + (0x200 * k) 0x420A 200C + (0x200 * k)	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE_EDMA_TPCC
Description	Event Clear Register (High Part) CPU write of '1' to the EDMA_TPCC_ECRH.En bit causes the EDMA_TPCC_ERH.En bit to be cleared. CPU write of '0' has no effect.		
Type	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E63	E62	E61	E60	E59	E58	E57	E56	E55	E54	E53	E52	E51	E50	E49	E48	E47	E46	E45	E44	E43	E42	E41	E40	E39	E38	E37	E36	E35	E34	E33	E32

Bits	Field Name	Description	Type	Reset
31	E63	Event #63	W	0x0
30	E62	Event #62	W	0x0
29	E61	Event #61	W	0x0
28	E60	Event #60	W	0x0
27	E59	Event #59	W	0x0
26	E58	Event #58	W	0x0
25	E57	Event #57	W	0x0
24	E56	Event #56	W	0x0
23	E55	Event #55	W	0x0
22	E54	Event #54	W	0x0
21	E53	Event #53	W	0x0
20	E52	Event #52	W	0x0
19	E51	Event #51	W	0x0
18	E50	Event #50	W	0x0
17	E49	Event #49	W	0x0
16	E48	Event #48	W	0x0
15	E47	Event #47	W	0x0
14	E46	Event #46	W	0x0
13	E45	Event #45	W	0x0
12	E44	Event #44	W	0x0
11	E43	Event #43	W	0x0
10	E42	Event #42	W	0x0
9	E41	Event #41	W	0x0
8	E40	Event #40	W	0x0
7	E39	Event #39	W	0x0
6	E38	Event #38	W	0x0
5	E37	Event #37	W	0x0
4	E36	Event #36	W	0x0
3	E35	Event #35	W	0x0
2	E34	Event #34	W	0x0
1	E33	Event #33	W	0x0
0	E32	Event #32	W	0x0

Table 11-187. Register Call Summary for Register EDMA_TPCC_ECRH_RN_k

EDMA Register Manual
- EDMA Register Summary: [0][1][2][3]

Table 11-188. EDMA_TPCC_ESR_RN_k

Address Offset	0x0000 2010 + (0x200 * k)		
Physical Address	0x4330 2010 + (0x200 * k) 0x40D1 2010 + (0x200 * k) 0x4151 2010 + (0x200 * k) 0x01D1 2010 + (0x200 * k) 0x420A 2010 + (0x200 * k)	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE_EDMA_TPCC
Description	Event Set Register CPU write of '1' to the EDMA_TPCC_ESR.En bit causes the EDMA_TPCC_ER.En bit to be set. CPU write of '0' has no effect.		
Type	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E31	E30	E29	E28	E27	E26	E25	E24	E23	E22	E21	E20	E19	E18	E17	E16	E15	E14	E13	E12	E11	E10	E9	E8	E7	E6	E5	E4	E3	E2	E1	E0

Bits	Field Name	Description	Type	Reset
31	E31	Event #31	W	0x0
30	E30	Event #30	W	0x0
29	E29	Event #29	W	0x0
28	E28	Event #28	W	0x0
27	E27	Event #27	W	0x0
26	E26	Event #26	W	0x0
25	E25	Event #25	W	0x0
24	E24	Event #24	W	0x0
23	E23	Event #23	W	0x0
22	E22	Event #22	W	0x0
21	E21	Event #21	W	0x0
20	E20	Event #20	W	0x0
19	E19	Event #19	W	0x0
18	E18	Event #18	W	0x0
17	E17	Event #17	W	0x0
16	E16	Event #16	W	0x0
15	E15	Event #15	W	0x0
14	E14	Event #14	W	0x0
13	E13	Event #13	W	0x0
12	E12	Event #12	W	0x0
11	E11	Event #11	W	0x0
10	E10	Event #10	W	0x0
9	E9	Event #9	W	0x0
8	E8	Event #8	W	0x0
7	E7	Event #7	W	0x0
6	E6	Event #6	W	0x0
5	E5	Event #5	W	0x0
4	E4	Event #4	W	0x0
3	E3	Event #3	W	0x0
2	E2	Event #2	W	0x0
1	E1	Event #1	W	0x0
0	E0	Event #0	W	0x0

Table 11-189. Register Call Summary for Register EDMA_TPCC_ESR_RN_k

EDMA Register Manual

- EDMA Register Summary: [0][1][2][3]

Table 11-190. EDMA_TPCC_ESRH_RN_k

Address Offset	0x0000 2014 + (0x200 * k)		
Physical Address	0x4330 2014 + (0x200 * k) 0x40D1 2014 + (0x200 * k) 0x4151 2014 + (0x200 * k) 0x01D1 2014 + (0x200 * k) 0x420A 2014 + (0x200 * k)	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE_EDMA_TPCC

Table 11-190. EDMA_TPCC_ESRH_RN_k (continued)

Description	Event Set Register (High Part) CPU write of '1' to the EDMA_TPCC_ESRH.En bit causes the EDMA_TPCC_ERH.En bit to be set. CPU write of '0' has no effect.
Type	W

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E63	E62	E61	E60	E59	E58	E57	E56	E55	E54	E53	E52	E51	E50	E49	E48	E47	E46	E45	E44	E43	E42	E41	E40	E39	E38	E37	E36	E35	E34	E33	E32

Bits	Field Name	Description	Type	Reset
31	E63	Event #63	W	0x0
30	E62	Event #62	W	0x0
29	E61	Event #61	W	0x0
28	E60	Event #60	W	0x0
27	E59	Event #59	W	0x0
26	E58	Event #58	W	0x0
25	E57	Event #57	W	0x0
24	E56	Event #56	W	0x0
23	E55	Event #55	W	0x0
22	E54	Event #54	W	0x0
21	E53	Event #53	W	0x0
20	E52	Event #52	W	0x0
19	E51	Event #51	W	0x0
18	E50	Event #50	W	0x0
17	E49	Event #49	W	0x0
16	E48	Event #48	W	0x0
15	E47	Event #47	W	0x0
14	E46	Event #46	W	0x0
13	E45	Event #45	W	0x0
12	E44	Event #44	W	0x0
11	E43	Event #43	W	0x0
10	E42	Event #42	W	0x0
9	E41	Event #41	W	0x0
8	E40	Event #40	W	0x0
7	E39	Event #39	W	0x0
6	E38	Event #38	W	0x0
5	E37	Event #37	W	0x0
4	E36	Event #36	W	0x0
3	E35	Event #35	W	0x0
2	E34	Event #34	W	0x0
1	E33	Event #33	W	0x0
0	E32	Event #32	W	0x0

Table 11-191. Register Call Summary for Register EDMA_TPCC_ESRH_RN_k

EDMA Register Manual

- EDMA Register Summary: [0][1][2][3]

Table 11-192. EDMA_TPCC_CER_RN_k

Address Offset	0x0000 2018 + (0x200 * k)		
Physical Address	0x4330 2018 + (0x200 * k) 0x40D1 2018 + (0x200 * k) 0x4151 2018 + (0x200 * k) 0x01D1 2018 + (0x200 * k) 0x420A 2018 + (0x200 * k)	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE_EDMA_TPCC
Description	Chained Event Register If EDMA_TPCC_CER.En bit is set (regardless of state of EDMA_TPCC_EER.En), then the corresponding DMA channel is prioritized vs. other pending DMA events for submission to the TC. EDMA_TPCC_CER.En bit is set when a chaining completion code is returned from one of the 3PTCs via the completion interface, or is generated internally via Early Completion path. EDMA_TPCC_CER.En bit is cleared when the corresponding event is prioritized and serviced. If the EDMA_TPCC_CER.En bit is already set and the corresponding chaining completion code is returned from the TC, then the corresponding bit in the Event Missed Register is set. EDMA_TPCC_CER.En cannot be set or cleared via software.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E31	E30	E29	E28	E27	E26	E25	E24	E23	E22	E21	E20	E19	E18	E17	E16	E15	E14	E13	E12	E11	E10	E9	E8	E7	E6	E5	E4	E3	E2	E1	E0

Bits	Field Name	Description	Type	Reset
31	E31	Event #31	R	0x0
30	E30	Event #30	R	0x0
29	E29	Event #29	R	0x0
28	E28	Event #28	R	0x0
27	E27	Event #27	R	0x0
26	E26	Event #26	R	0x0
25	E25	Event #25	R	0x0
24	E24	Event #24	R	0x0
23	E23	Event #23	R	0x0
22	E22	Event #22	R	0x0
21	E21	Event #21	R	0x0
20	E20	Event #20	R	0x0
19	E19	Event #19	R	0x0
18	E18	Event #18	R	0x0
17	E17	Event #17	R	0x0
16	E16	Event #16	R	0x0
15	E15	Event #15	R	0x0
14	E14	Event #14	R	0x0
13	E13	Event #13	R	0x0
12	E12	Event #12	R	0x0
11	E11	Event #11	R	0x0
10	E10	Event #10	R	0x0
9	E9	Event #9	R	0x0
8	E8	Event #8	R	0x0
7	E7	Event #7	R	0x0
6	E6	Event #6	R	0x0
5	E5	Event #5	R	0x0
4	E4	Event #4	R	0x0
3	E3	Event #3	R	0x0
2	E2	Event #2	R	0x0

Bits	Field Name	Description	Type	Reset
1	E1	Event #1	R	0x0
0	E0	Event #0	R	0x0

Table 11-193. Register Call Summary for Register EDMA_TPCC_CER_RN_k

EDMA Register Manual
- EDMA Register Summary: [0][1][2][3]

Table 11-194. EDMA_TPCC_CERH_RN_k

Address Offset	0x0000 201C + (0x200 * k)		
Physical Address	0x4330 201C + (0x200 * k) 0x40D1 201C + (0x200 * k) 0x4151 201C + (0x200 * k) 0x01D1 201C + (0x200 * k) 0x420A 201C + (0x200 * k)	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE_EDMA_TPCC
Description	Chained Event Register (High Part) If EDMA_TPCC_CERH.En bit is set (regardless of state of EDMA_TPCC_EERH.En), then the corresponding DMA channel is prioritized vs. other pending DMA events for submission to the TC. EDMA_TPCC_CERH.En bit is set when a chaining completion code is returned from one of the 3PTCs via the completion interface, or is generated internally via Early Completion path. EDMA_TPCC_CERH.En bit is cleared when the corresponding event is prioritized and serviced. If the EDMA_TPCC_CERH.En bit is already set and the corresponding chaining completion code is returned from the TC, then the corresponding bit in the Event Missed Register is set. EDMA_TPCC_CERH.En cannot be set or cleared via software.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E63	E62	E61	E60	E59	E58	E57	E56	E55	E54	E53	E52	E51	E50	E49	E48	E47	E46	E45	E44	E43	E42	E41	E40	E39	E38	E37	E36	E35	E34	E33	E32

Bits	Field Name	Description	Type	Reset
31	E63	Event #63	R	0x0
30	E62	Event #62	R	0x0
29	E61	Event #61	R	0x0
28	E60	Event #60	R	0x0
27	E59	Event #59	R	0x0
26	E58	Event #58	R	0x0
25	E57	Event #57	R	0x0
24	E56	Event #56	R	0x0
23	E55	Event #55	R	0x0
22	E54	Event #54	R	0x0
21	E53	Event #53	R	0x0
20	E52	Event #52	R	0x0
19	E51	Event #51	R	0x0
18	E50	Event #50	R	0x0
17	E49	Event #49	R	0x0
16	E48	Event #48	R	0x0
15	E47	Event #47	R	0x0
14	E46	Event #46	R	0x0
13	E45	Event #45	R	0x0
12	E44	Event #44	R	0x0
11	E43	Event #43	R	0x0
10	E42	Event #42	R	0x0

Bits	Field Name	Description	Type	Reset
9	E41	Event #41	R	0x0
8	E40	Event #40	R	0x0
7	E39	Event #39	R	0x0
6	E38	Event #38	R	0x0
5	E37	Event #37	R	0x0
4	E36	Event #36	R	0x0
3	E35	Event #35	R	0x0
2	E34	Event #34	R	0x0
1	E33	Event #33	R	0x0
0	E32	Event #32	R	0x0

Table 11-195. Register Call Summary for Register EDMA_TPCC_CERH_RN_k

EDMA Register Manual

- EDMA Register Summary: [0][1][2][3]

Table 11-196. EDMA_TPCC_EER_RN_k

Address Offset	0x0000 2020 + (0x200 * k)		
Physical Address	0x4330 2020 + (0x200 * k) 0x40D1 2020 + (0x200 * k) 0x4151 2020 + (0x200 * k) 0x01D1 2020 + (0x200 * k) 0x420A 2020 + (0x200 * k)	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE_EDMA_TPCC
Description	Event Enable Register Enables DMA transfers for EDMA_TPCC_ER.En pending events. EDMA_TPCC_ER.En is set based on externally asserted events (via tpcc_eventN_pi). This register has no effect on Chained Event Register (EDMA_TPCC_CER) or Event Set Register (EDMA_TPCC_ESR). **NOTE: If a bit is set in EDMA_TPCC_ER.En while EDMA_TPCC_EER.En is disabled, no action is taken.** If EDMA_TPCC_EER.En is enabled at a later point (and EDMA_TPCC_ER.En has not been cleared via SW) then the event will be recognized as a valid 'TR Sync' EDMA_TPCC_EER.En is not directly writeable. Events can be enabled via writes to EDMA_TPCC_EESR and can be disabled via writes to EDMA_TPCC_EECR register. EDMA_TPCC_EER.En = 0: EDMA_TPCC_ER.En is not enabled to trigger DMA transfers. EDMA_TPCC_EER.En = 1: EDMA_TPCC_ER.En is enabled to trigger DMA transfers.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E31	E30	E29	E28	E27	E26	E25	E24	E23	E22	E21	E20	E19	E18	E17	E16	E15	E14	E13	E12	E11	E10	E9	E8	E7	E6	E5	E4	E3	E2	E1	E0

Bits	Field Name	Description	Type	Reset
31	E31	Event #31	R	0x0
30	E30	Event #30	R	0x0
29	E29	Event #29	R	0x0
28	E28	Event #28	R	0x0
27	E27	Event #27	R	0x0
26	E26	Event #26	R	0x0
25	E25	Event #25	R	0x0
24	E24	Event #24	R	0x0
23	E23	Event #23	R	0x0
22	E22	Event #22	R	0x0
21	E21	Event #21	R	0x0
20	E20	Event #20	R	0x0

Bits	Field Name	Description	Type	Reset
19	E19	Event #19	R	0x0
18	E18	Event #18	R	0x0
17	E17	Event #17	R	0x0
16	E16	Event #16	R	0x0
15	E15	Event #15	R	0x0
14	E14	Event #14	R	0x0
13	E13	Event #13	R	0x0
12	E12	Event #12	R	0x0
11	E11	Event #11	R	0x0
10	E10	Event #10	R	0x0
9	E9	Event #9	R	0x0
8	E8	Event #8	R	0x0
7	E7	Event #7	R	0x0
6	E6	Event #6	R	0x0
5	E5	Event #5	R	0x0
4	E4	Event #4	R	0x0
3	E3	Event #3	R	0x0
2	E2	Event #2	R	0x0
1	E1	Event #1	R	0x0
0	E0	Event #0	R	0x0

Table 11-197. Register Call Summary for Register EDMA_TPCC_EER_RN_k

EDMA Register Manual

- EDMA Register Summary: [0][1][2][3]

Table 11-198. EDMA_TPCC_EERH_RN_k

Address Offset	0x0000 2024 + (0x200 * k)		
Physical Address	0x4330 2024 + (0x200 * k) 0x40D1 2024 + (0x200 * k) 0x4151 2024 + (0x200 * k) 0x01D1 2024 + (0x200 * k) 0x420A 2024 + (0x200 * k)	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE_EDMA_TPCC
Description	Event Enable Register (High Part) Enables DMA transfers for EDMA_TPCC_ERH.En pending events. EDMA_TPCC_ERH.En is set based on externally asserted events (via tpcc_eventN_pi). This register has no effect on Chained Event Register (EDMA_TPCC_CERH) or Event Set Register (EDMA_TPCC_ESRH). **NOTE: If a bit is set in EDMA_TPCC_ERH.En while EDMA_TPCC_EERH.En is disabled, no action is taken.** If EDMA_TPCC_EERH.En is enabled at a later point (and EDMA_TPCC_ERH.En has not been cleared via SW) then the event will be recognized as a valid 'TR Sync' EDMA_TPCC_EERH.En is not directly writeable. Events can be enabled via writes to EDMA_TPCC_EESRH and can be disabled via writes to EDMA_TPCC_EECRH register. EDMA_TPCC_EERH.En = 0: EDMA_TPCC_ER.En is not enabled to trigger DMA transfers. EDMA_TPCC_EERH.En = 1: EDMA_TPCC_ER.En is enabled to trigger DMA transfers.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E63	E62	E61	E60	E59	E58	E57	E56	E55	E54	E53	E52	E51	E50	E49	E48	E47	E46	E45	E44	E43	E42	E41	E40	E39	E38	E37	E36	E35	E34	E33	E32

Bits	Field Name	Description	Type	Reset
31	E63	Event #63	R	0x0
30	E62	Event #62	R	0x0
29	E61	Event #61	R	0x0
28	E60	Event #60	R	0x0
27	E59	Event #59	R	0x0
26	E58	Event #58	R	0x0
25	E57	Event #57	R	0x0
24	E56	Event #56	R	0x0
23	E55	Event #55	R	0x0
22	E54	Event #54	R	0x0
21	E53	Event #53	R	0x0
20	E52	Event #52	R	0x0
19	E51	Event #51	R	0x0
18	E50	Event #50	R	0x0
17	E49	Event #49	R	0x0
16	E48	Event #48	R	0x0
15	E47	Event #47	R	0x0
14	E46	Event #46	R	0x0
13	E45	Event #45	R	0x0
12	E44	Event #44	R	0x0
11	E43	Event #43	R	0x0
10	E42	Event #42	R	0x0
9	E41	Event #41	R	0x0
8	E40	Event #40	R	0x0
7	E39	Event #39	R	0x0
6	E38	Event #38	R	0x0
5	E37	Event #37	R	0x0
4	E36	Event #36	R	0x0
3	E35	Event #35	R	0x0
2	E34	Event #34	R	0x0
1	E33	Event #33	R	0x0
0	E32	Event #32	R	0x0

Table 11-199. Register Call Summary for Register EDMA_TPCC_EERH_RN_k

EDMA Register Manual
- EDMA Register Summary: [0][1][2][3]

Table 11-200. EDMA_TPCC_EECR_RN_k

Address Offset	0x0000 2028 + (0x200 * k)		
Physical Address	0x4330 2028 + (0x200 * k) 0x40D1 2028 + (0x200 * k) 0x4151 2028 + (0x200 * k) 0x01D1 2028 + (0x200 * k) 0x420A 2028 + (0x200 * k)	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE_EDMA_TPCC
Description	Event Enable Clear Register CPU write of '0' has no effect. CPU write of '1' to the EDMA_TPCC_EECR.En bit causes the EDMA_TPCC_EER.En bit to be cleared.		
Type	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E31	E30	E29	E28	E27	E26	E25	E24	E23	E22	E21	E20	E19	E18	E17	E16	E15	E14	E13	E12	E11	E10	E9	E8	E7	E6	E5	E4	E3	E2	E1	E0

Bits	Field Name	Description	Type	Reset
31	E31	Event #31	W	0x0
30	E30	Event #30	W	0x0
29	E29	Event #29	W	0x0
28	E28	Event #28	W	0x0
27	E27	Event #27	W	0x0
26	E26	Event #26	W	0x0
25	E25	Event #25	W	0x0
24	E24	Event #24	W	0x0
23	E23	Event #23	W	0x0
22	E22	Event #22	W	0x0
21	E21	Event #21	W	0x0
20	E20	Event #20	W	0x0
19	E19	Event #19	W	0x0
18	E18	Event #18	W	0x0
17	E17	Event #17	W	0x0
16	E16	Event #16	W	0x0
15	E15	Event #15	W	0x0
14	E14	Event #14	W	0x0
13	E13	Event #13	W	0x0
12	E12	Event #12	W	0x0
11	E11	Event #11	W	0x0
10	E10	Event #10	W	0x0
9	E9	Event #9	W	0x0
8	E8	Event #8	W	0x0
7	E7	Event #7	W	0x0
6	E6	Event #6	W	0x0
5	E5	Event #5	W	0x0
4	E4	Event #4	W	0x0
3	E3	Event #3	W	0x0
2	E2	Event #2	W	0x0
1	E1	Event #1	W	0x0
0	E0	Event #0	W	0x0

Table 11-201. Register Call Summary for Register EDMA_TPCC_EECR_RN_k

EDMA Register Manual

- EDMA Register Summary: [0][1][2][3]

Table 11-202. EDMA_TPCC_EECRH_RN_k

Address Offset	0x0000 202C + (0x200 * k)		
Physical Address	0x4330 202C + (0x200 * k) 0x40D1 202C + (0x200 * k) 0x4151 202C + (0x200 * k) 0x01D1 202C + (0x200 * k) 0x420A 202C + (0x200 * k)	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE_EDMA_TPCC

Table 11-202. EDMA_TPCC_EECRH_RN_k (continued)

Description	Event Enable Clear Register (High Part) CPU write of '0' has no effect. CPU write of '1' to the EDMA_TPCC_EECRH.En bit causes the EDMA_TPCC_EERH.En bit to be cleared.
Type	W

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E63	E62	E61	E60	E59	E58	E57	E56	E55	E54	E53	E52	E51	E50	E49	E48	E47	E46	E45	E44	E43	E42	E41	E40	E39	E38	E37	E36	E35	E34	E33	E32

Bits	Field Name	Description	Type	Reset
31	E63	Event #63	W	0x0
30	E62	Event #62	W	0x0
29	E61	Event #61	W	0x0
28	E60	Event #60	W	0x0
27	E59	Event #59	W	0x0
26	E58	Event #58	W	0x0
25	E57	Event #57	W	0x0
24	E56	Event #56	W	0x0
23	E55	Event #55	W	0x0
22	E54	Event #54	W	0x0
21	E53	Event #53	W	0x0
20	E52	Event #52	W	0x0
19	E51	Event #51	W	0x0
18	E50	Event #50	W	0x0
17	E49	Event #49	W	0x0
16	E48	Event #48	W	0x0
15	E47	Event #47	W	0x0
14	E46	Event #46	W	0x0
13	E45	Event #45	W	0x0
12	E44	Event #44	W	0x0
11	E43	Event #43	W	0x0
10	E42	Event #42	W	0x0
9	E41	Event #41	W	0x0
8	E40	Event #40	W	0x0
7	E39	Event #39	W	0x0
6	E38	Event #38	W	0x0
5	E37	Event #37	W	0x0
4	E36	Event #36	W	0x0
3	E35	Event #35	W	0x0
2	E34	Event #34	W	0x0
1	E33	Event #33	W	0x0
0	E32	Event #32	W	0x0

Table 11-203. Register Call Summary for Register EDMA_TPCC_EECRH_RN_k

EDMA Register Manual

- EDMA Register Summary: [0][1][2][3]

Table 11-204. EDMA_TPCC_EESR_RN_k

Address Offset	0x0000 2030 + (0x200 * k)		
Physical Address	0x4330 2030 + (0x200 * k) 0x40D1 2030 + (0x200 * k) 0x4151 2030 + (0x200 * k) 0x01D1 2030 + (0x200 * k) 0x420A 2030 + (0x200 * k)	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE_EDMA_TPCC
Description	Event Enable Set Register CPU write of '0' has no effect. CPU write of '1' to the EDMA_TPCC_EESR.En bit causes the EDMA_TPCC_EER.En bit to be set.		
Type	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E31	E30	E29	E28	E27	E26	E25	E24	E23	E22	E21	E20	E19	E18	E17	E16	E15	E14	E13	E12	E11	E10	E9	E8	E7	E6	E5	E4	E3	E2	E1	E0

Bits	Field Name	Description	Type	Reset
31	E31	Event #31	W	0x0
30	E30	Event #30	W	0x0
29	E29	Event #29	W	0x0
28	E28	Event #28	W	0x0
27	E27	Event #27	W	0x0
26	E26	Event #26	W	0x0
25	E25	Event #25	W	0x0
24	E24	Event #24	W	0x0
23	E23	Event #23	W	0x0
22	E22	Event #22	W	0x0
21	E21	Event #21	W	0x0
20	E20	Event #20	W	0x0
19	E19	Event #19	W	0x0
18	E18	Event #18	W	0x0
17	E17	Event #17	W	0x0
16	E16	Event #16	W	0x0
15	E15	Event #15	W	0x0
14	E14	Event #14	W	0x0
13	E13	Event #13	W	0x0
12	E12	Event #12	W	0x0
11	E11	Event #11	W	0x0
10	E10	Event #10	W	0x0
9	E9	Event #9	W	0x0
8	E8	Event #8	W	0x0
7	E7	Event #7	W	0x0
6	E6	Event #6	W	0x0
5	E5	Event #5	W	0x0
4	E4	Event #4	W	0x0
3	E3	Event #3	W	0x0
2	E2	Event #2	W	0x0
1	E1	Event #1	W	0x0
0	E0	Event #0	W	0x0

Table 11-205. Register Call Summary for Register EDMA_TPCC_EESR_RN_k

EDMA Register Manual

- EDMA Register Summary: [0][1][2][3]

Table 11-206. EDMA_TPCC_EESRH_RN_k

Address Offset	0x0000 2034 + (0x200 * k)		
Physical Address	0x4330 2034 + (0x200 * k) 0x40D1 2034 + (0x200 * k) 0x4151 2034 + (0x200 * k) 0x01D1 2034 + (0x200 * k) 0x420A 2034 + (0x200 * k)	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE_EDMA_TPCC
Description	Event Enable Set Register (High Part) CPU write of '0' has no effect. CPU write of '1' to the EDMA_TPCC_EESRH.En bit causes the EDMA_TPCC_EERH.En bit to be set.		
Type	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E63	E62	E61	E60	E59	E58	E57	E56	E55	E54	E53	E52	E51	E50	E49	E48	E47	E46	E45	E44	E43	E42	E41	E40	E39	E38	E37	E36	E35	E34	E33	E32

Bits	Field Name	Description	Type	Reset
31	E63	Event #63	W	0x0
30	E62	Event #62	W	0x0
29	E61	Event #61	W	0x0
28	E60	Event #60	W	0x0
27	E59	Event #59	W	0x0
26	E58	Event #58	W	0x0
25	E57	Event #57	W	0x0
24	E56	Event #56	W	0x0
23	E55	Event #55	W	0x0
22	E54	Event #54	W	0x0
21	E53	Event #53	W	0x0
20	E52	Event #52	W	0x0
19	E51	Event #51	W	0x0
18	E50	Event #50	W	0x0
17	E49	Event #49	W	0x0
16	E48	Event #48	W	0x0
15	E47	Event #47	W	0x0
14	E46	Event #46	W	0x0
13	E45	Event #45	W	0x0
12	E44	Event #44	W	0x0
11	E43	Event #43	W	0x0
10	E42	Event #42	W	0x0
9	E41	Event #41	W	0x0
8	E40	Event #40	W	0x0
7	E39	Event #39	W	0x0
6	E38	Event #38	W	0x0
5	E37	Event #37	W	0x0
4	E36	Event #36	W	0x0
3	E35	Event #35	W	0x0

Bits	Field Name	Description	Type	Reset
2	E34	Event #34	W	0x0
1	E33	Event #33	W	0x0
0	E32	Event #32	W	0x0

Table 11-207. Register Call Summary for Register EDMA_TPCC_EESRH_RN_k

EDMA Register Manual
- EDMA Register Summary: [0][1][2][3]

Table 11-208. EDMA_TPCC_SER_RN_k

Address Offset	0x0000 2038 + (0x200 * k)		
Physical Address	0x4330 2038 + (0x200 * k) 0x40D1 2038 + (0x200 * k) 0x4151 2038 + (0x200 * k) 0x01D1 2038 + (0x200 * k) 0x420A 2038 + (0x200 * k)	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE_EDMA_TPCC
Description	Secondary Event Register The secondary event register is used along with the Event Register (EDMA_TPCC_ER) to provide information on the state of an Event. En = 0 : Event is not currently in the Event Queue. En = 1 : Event is currently stored in Event Queue. Event arbiter will not prioritize additional events.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E31	E30	E29	E28	E27	E26	E25	E24	E23	E22	E21	E20	E19	E18	E17	E16	E15	E14	E13	E12	E11	E10	E9	E8	E7	E6	E5	E4	E3	E2	E1	E0

Bits	Field Name	Description	Type	Reset
31	E31	Event #31	R	0x0
30	E30	Event #30	R	0x0
29	E29	Event #29	R	0x0
28	E28	Event #28	R	0x0
27	E27	Event #27	R	0x0
26	E26	Event #26	R	0x0
25	E25	Event #25	R	0x0
24	E24	Event #24	R	0x0
23	E23	Event #23	R	0x0
22	E22	Event #22	R	0x0
21	E21	Event #21	R	0x0
20	E20	Event #20	R	0x0
19	E19	Event #19	R	0x0
18	E18	Event #18	R	0x0
17	E17	Event #17	R	0x0
16	E16	Event #16	R	0x0
15	E15	Event #15	R	0x0
14	E14	Event #14	R	0x0
13	E13	Event #13	R	0x0
12	E12	Event #12	R	0x0
11	E11	Event #11	R	0x0
10	E10	Event #10	R	0x0
9	E9	Event #9	R	0x0

Bits	Field Name	Description	Type	Reset
8	E8	Event #8	R	0x0
7	E7	Event #7	R	0x0
6	E6	Event #6	R	0x0
5	E5	Event #5	R	0x0
4	E4	Event #4	R	0x0
3	E3	Event #3	R	0x0
2	E2	Event #2	R	0x0
1	E1	Event #1	R	0x0
0	E0	Event #0	R	0x0

Table 11-209. Register Call Summary for Register EDMA_TPCC_SER_RN_k

EDMA Register Manual
- EDMA Register Summary: [0][1][2][3]

Table 11-210. EDMA_TPCC_SERH_RN_k

Address Offset	0x0000 203C + (0x200 * k)		
Physical Address	0x4330 203C + (0x200 * k) 0x40D1 203C + (0x200 * k) 0x4151 203C + (0x200 * k) 0x01D1 203C + (0x200 * k) 0x420A 203C + (0x200 * k)	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE_EDMA_TPCC
Description	Secondary Event Register (High Part) The secondary event register is used along with the Event Register (EDMA_TPCC_ERH) to provide information on the state of an Event. En = 0 : Event is not currently in the Event Queue. En = 1 : Event is currently stored in Event Queue. Event arbiter will not prioritize additional events.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E63	E62	E61	E60	E59	E58	E57	E56	E55	E54	E53	E52	E51	E50	E49	E48	E47	E46	E45	E44	E43	E42	E41	E40	E39	E38	E37	E36	E35	E34	E33	E32

Bits	Field Name	Description	Type	Reset
31	E63	Event #63	R	0x0
30	E62	Event #62	R	0x0
29	E61	Event #61	R	0x0
28	E60	Event #60	R	0x0
27	E59	Event #59	R	0x0
26	E58	Event #58	R	0x0
25	E57	Event #57	R	0x0
24	E56	Event #56	R	0x0
23	E55	Event #55	R	0x0
22	E54	Event #54	R	0x0
21	E53	Event #53	R	0x0
20	E52	Event #52	R	0x0
19	E51	Event #51	R	0x0
18	E50	Event #50	R	0x0
17	E49	Event #49	R	0x0
16	E48	Event #48	R	0x0
15	E47	Event #47	R	0x0

Bits	Field Name	Description	Type	Reset
14	E46	Event #46	R	0x0
13	E45	Event #45	R	0x0
12	E44	Event #44	R	0x0
11	E43	Event #43	R	0x0
10	E42	Event #42	R	0x0
9	E41	Event #41	R	0x0
8	E40	Event #40	R	0x0
7	E39	Event #39	R	0x0
6	E38	Event #38	R	0x0
5	E37	Event #37	R	0x0
4	E36	Event #36	R	0x0
3	E35	Event #35	R	0x0
2	E34	Event #34	R	0x0
1	E33	Event #33	R	0x0
0	E32	Event #32	R	0x0

Table 11-211. Register Call Summary for Register EDMA_TPCC_SERH_RN_k

EDMA Register Manual

- EDMA Register Summary: [0][1][2][3]

Table 11-212. EDMA_TPCC_SECR_RN_k

Address Offset	0x0000 2040 + (0x200 * k)		
Physical Address	0x4330 2040 + (0x200 * k) 0x40D1 2040 + (0x200 * k) 0x4151 2040 + (0x200 * k) 0x01D1 2040 + (0x200 * k) 0x420A 2040 + (0x200 * k)	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE_EDMA_TPCC
Description	Secondary Event Clear Register The secondary event clear register is used to clear the status of the EDMA_TPCC_SER registers. CPU write of '0' has no effect. CPU write of '1' to the EDMA_TPCC_SECR.En bit clears the EDMA_TPCC_SER register.		
Type	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E31	E30	E29	E28	E27	E26	E25	E24	E23	E22	E21	E20	E19	E18	E17	E16	E15	E14	E13	E12	E11	E10	E9	E8	E7	E6	E5	E4	E3	E2	E1	E0

Bits	Field Name	Description	Type	Reset
31	E31	Event #31	W	0x0
30	E30	Event #30	W	0x0
29	E29	Event #29	W	0x0
28	E28	Event #28	W	0x0
27	E27	Event #27	W	0x0
26	E26	Event #26	W	0x0
25	E25	Event #25	W	0x0
24	E24	Event #24	W	0x0
23	E23	Event #23	W	0x0
22	E22	Event #22	W	0x0
21	E21	Event #21	W	0x0
20	E20	Event #20	W	0x0

Bits	Field Name	Description	Type	Reset
19	E19	Event #19	W	0x0
18	E18	Event #18	W	0x0
17	E17	Event #17	W	0x0
16	E16	Event #16	W	0x0
15	E15	Event #15	W	0x0
14	E14	Event #14	W	0x0
13	E13	Event #13	W	0x0
12	E12	Event #12	W	0x0
11	E11	Event #11	W	0x0
10	E10	Event #10	W	0x0
9	E9	Event #9	W	0x0
8	E8	Event #8	W	0x0
7	E7	Event #7	W	0x0
6	E6	Event #6	W	0x0
5	E5	Event #5	W	0x0
4	E4	Event #4	W	0x0
3	E3	Event #3	W	0x0
2	E2	Event #2	W	0x0
1	E1	Event #1	W	0x0
0	E0	Event #0	W	0x0

Table 11-213. Register Call Summary for Register EDMA_TPCC_SECR_RN_k

EDMA Register Manual

- EDMA Register Summary: [0][1][2][3]

Table 11-214. EDMA_TPCC_SECRH_RN_k

Address Offset	0x0000 2044 + (0x200 * k)		
Physical Address	0x4330 2044 + (0x200 * k) 0x40D1 2044 + (0x200 * k) 0x4151 2044 + (0x200 * k) 0x01D1 2044 + (0x200 * k) 0x420A 2044 + (0x200 * k)	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE_EDMA_TPCC
Description	Secondary Event Clear Register (High Part) The secondary event clear register is used to clear the status of the EDMA_TPCC_SERH registers. CPU write of '0' has no effect. CPU write of '1' to the EDMA_TPCC_SECRH.En bit clears the EDMA_TPCC_SERH register.		
Type	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
E63	E62	E61	E60	E59	E58	E57	E56	E55	E54	E53	E52	E51	E50	E49	E48	E47	E46	E45	E44	E43	E42	E41	E40	E39	E38	E37	E36	E35	E34	E33	E32

Bits	Field Name	Description	Type	Reset
31	E63	Event #63	W	0x0
30	E62	Event #62	W	0x0
29	E61	Event #61	W	0x0
28	E60	Event #60	W	0x0
27	E59	Event #59	W	0x0
26	E58	Event #58	W	0x0
25	E57	Event #57	W	0x0

Bits	Field Name	Description	Type	Reset
24	E56	Event #56	W	0x0
23	E55	Event #55	W	0x0
22	E54	Event #54	W	0x0
21	E53	Event #53	W	0x0
20	E52	Event #52	W	0x0
19	E51	Event #51	W	0x0
18	E50	Event #50	W	0x0
17	E49	Event #49	W	0x0
16	E48	Event #48	W	0x0
15	E47	Event #47	W	0x0
14	E46	Event #46	W	0x0
13	E45	Event #45	W	0x0
12	E44	Event #44	W	0x0
11	E43	Event #43	W	0x0
10	E42	Event #42	W	0x0
9	E41	Event #41	W	0x0
8	E40	Event #40	W	0x0
7	E39	Event #39	W	0x0
6	E38	Event #38	W	0x0
5	E37	Event #37	W	0x0
4	E36	Event #36	W	0x0
3	E35	Event #35	W	0x0
2	E34	Event #34	W	0x0
1	E33	Event #33	W	0x0
0	E32	Event #32	W	0x0

Table 11-215. Register Call Summary for Register EDMA_TPCC_SECRH_RN_k

EDMA Register Manual

- EDMA Register Summary: [0][1][2][3]

Table 11-216. EDMA_TPCC_IER_RN_k

Address Offset	0x0000 2050 + (0x200 * k)		
Physical Address	0x4330 2050 + (0x200 * k) 0x40D1 2050 + (0x200 * k) 0x4151 2050 + (0x200 * k) 0x01D1 2050 + (0x200 * k) 0x420A 2050 + (0x200 * k)	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE_EDMA_TPCC
Description	Int Enable Register EDMA_TPCC_IER.In is not directly writeable. Interrupts can be enabled via writes to EDMA_TPCC_IESR and can be disabled via writes to EDMA_TPCC_IECR register. EDMA_TPCC_IER.In = 0: EDMA_TPCC_IPR.In is NOT enabled for interrupts. EDMA_TPCC_IER.In = 1: EDMA_TPCC_IPR.In IS enabled for interrupts.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I31	I30	I29	I28	I27	I26	I25	I24	I23	I22	I21	I20	I19	I18	I17	I16	I15	I14	I13	I12	I11	I10	I9	I8	I7	I6	I5	I4	I3	I2	I1	I0

Bits	Field Name	Description	Type	Reset
31	I31	Interrupt associated with TCC #31	R	0x0
30	I30	Interrupt associated with TCC #30	R	0x0
29	I29	Interrupt associated with TCC #29	R	0x0
28	I28	Interrupt associated with TCC #28	R	0x0
27	I27	Interrupt associated with TCC #27	R	0x0
26	I26	Interrupt associated with TCC #26	R	0x0
25	I25	Interrupt associated with TCC #25	R	0x0
24	I24	Interrupt associated with TCC #24	R	0x0
23	I23	Interrupt associated with TCC #23	R	0x0
22	I22	Interrupt associated with TCC #22	R	0x0
21	I21	Interrupt associated with TCC #21	R	0x0
20	I20	Interrupt associated with TCC #20	R	0x0
19	I19	Interrupt associated with TCC #19	R	0x0
18	I18	Interrupt associated with TCC #18	R	0x0
17	I17	Interrupt associated with TCC #17	R	0x0
16	I16	Interrupt associated with TCC #16	R	0x0
15	I15	Interrupt associated with TCC #15	R	0x0
14	I14	Interrupt associated with TCC #14	R	0x0
13	I13	Interrupt associated with TCC #13	R	0x0
12	I12	Interrupt associated with TCC #12	R	0x0
11	I11	Interrupt associated with TCC #11	R	0x0
10	I10	Interrupt associated with TCC #10	R	0x0
9	I9	Interrupt associated with TCC #9	R	0x0
8	I8	Interrupt associated with TCC #8	R	0x0
7	I7	Interrupt associated with TCC #7	R	0x0
6	I6	Interrupt associated with TCC #6	R	0x0
5	I5	Interrupt associated with TCC #5	R	0x0
4	I4	Interrupt associated with TCC #4	R	0x0
3	I3	Interrupt associated with TCC #3	R	0x0
2	I2	Interrupt associated with TCC #2	R	0x0
1	I1	Interrupt associated with TCC #1	R	0x0
0	I0	Interrupt associated with TCC #0	R	0x0

Table 11-217. Register Call Summary for Register EDMA_TPCC_IER_RN_k

EDMA Register Manual
- EDMA Register Summary: [0][1][2][3]

Table 11-218. EDMA_TPCC_IERH_RN_k

Address Offset	0x0000 2054 + (0x200 * k)		
Physical Address	0x4330 2054 + (0x200 * k) 0x40D1 2054 + (0x200 * k) 0x4151 2054 + (0x200 * k) 0x01D1 2054 + (0x200 * k) 0x420A 2054 + (0x200 * k)	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE_EDMA_TPCC
Description	Int Enable Register (High Part) EDMA_TPCC_IERH.In is not directly writeable. Interrupts can be enabled via writes to EDMA_TPCC_IESRH and can be disabled via writes to EDMA_TPCC_IECRH register. EDMA_TPCC_IERH.In = 0: EDMA_TPCC_IPRH.In is NOT enabled for interrupts. EDMA_TPCC_IERH.In = 1: EDMA_TPCC_IPRH.In IS enabled for interrupts.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I63	I62	I61	I60	I59	I58	I57	I56	I55	I54	I53	I52	I51	I50	I49	I48	I47	I46	I45	I44	I43	I42	I41	I40	I39	I38	I37	I36	I35	I34	I33	I32

Bits	Field Name	Description	Type	Reset
31	I63	Interrupt associated with TCC #63	R	0x0
30	I62	Interrupt associated with TCC #62	R	0x0
29	I61	Interrupt associated with TCC #61	R	0x0
28	I60	Interrupt associated with TCC #60	R	0x0
27	I59	Interrupt associated with TCC #59	R	0x0
26	I58	Interrupt associated with TCC #58	R	0x0
25	I57	Interrupt associated with TCC #57	R	0x0
24	I56	Interrupt associated with TCC #56	R	0x0
23	I55	Interrupt associated with TCC #55	R	0x0
22	I54	Interrupt associated with TCC #54	R	0x0
21	I53	Interrupt associated with TCC #53	R	0x0
20	I52	Interrupt associated with TCC #52	R	0x0
19	I51	Interrupt associated with TCC #51	R	0x0
18	I50	Interrupt associated with TCC #50	R	0x0
17	I49	Interrupt associated with TCC #49	R	0x0
16	I48	Interrupt associated with TCC #48	R	0x0
15	I47	Interrupt associated with TCC #47	R	0x0
14	I46	Interrupt associated with TCC #46	R	0x0
13	I45	Interrupt associated with TCC #45	R	0x0
12	I44	Interrupt associated with TCC #44	R	0x0
11	I43	Interrupt associated with TCC #43	R	0x0
10	I42	Interrupt associated with TCC #42	R	0x0
9	I41	Interrupt associated with TCC #41	R	0x0
8	I40	Interrupt associated with TCC #40	R	0x0
7	I39	Interrupt associated with TCC #39	R	0x0
6	I38	Interrupt associated with TCC #38	R	0x0
5	I37	Interrupt associated with TCC #37	R	0x0
4	I36	Interrupt associated with TCC #36	R	0x0
3	I35	Interrupt associated with TCC #35	R	0x0
2	I34	Interrupt associated with TCC #34	R	0x0
1	I33	Interrupt associated with TCC #33	R	0x0
0	I32	Interrupt associated with TCC #32	R	0x0

Table 11-219. Register Call Summary for Register EDMA_TPCC_IERH_RN_k

EDMA Register Manual

- EDMA Register Summary: [0][1][2][3]

Table 11-220. EDMA_TPCC_IECR_RN_k

Address Offset	0x0000 2058 + (0x200 * k)		
Physical Address	0x4330 2058 + (0x200 * k) 0x40D1 2058 + (0x200 * k) 0x4151 2058 + (0x200 * k) 0x01D1 2058 + (0x200 * k) 0x420A 2058 + (0x200 * k)	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE_EDMA_TPCC

Table 11-220. EDMA_TPCC_IECR_RN_k (continued)

Description	Int Enable Clear Register CPU write of '0' has no effect. CPU write of '1' to the EDMA_TPCC_IECR.In bit causes the EDMA_TPCC_IER.In bit to be cleared.
Type	W

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I31	I30	I29	I28	I27	I26	I25	I24	I23	I22	I21	I20	I19	I18	I17	I16	I15	I14	I13	I12	I11	I10	I9	I8	I7	I6	I5	I4	I3	I2	I1	I0

Bits	Field Name	Description	Type	Reset
31	I31	Interrupt associated with TCC #31	W	0x0
30	I30	Interrupt associated with TCC #30	W	0x0
29	I29	Interrupt associated with TCC #29	W	0x0
28	I28	Interrupt associated with TCC #28	W	0x0
27	I27	Interrupt associated with TCC #27	W	0x0
26	I26	Interrupt associated with TCC #26	W	0x0
25	I25	Interrupt associated with TCC #25	W	0x0
24	I24	Interrupt associated with TCC #24	W	0x0
23	I23	Interrupt associated with TCC #23	W	0x0
22	I22	Interrupt associated with TCC #22	W	0x0
21	I21	Interrupt associated with TCC #21	W	0x0
20	I20	Interrupt associated with TCC #20	W	0x0
19	I19	Interrupt associated with TCC #19	W	0x0
18	I18	Interrupt associated with TCC #18	W	0x0
17	I17	Interrupt associated with TCC #17	W	0x0
16	I16	Interrupt associated with TCC #16	W	0x0
15	I15	Interrupt associated with TCC #15	W	0x0
14	I14	Interrupt associated with TCC #14	W	0x0
13	I13	Interrupt associated with TCC #13	W	0x0
12	I12	Interrupt associated with TCC #12	W	0x0
11	I11	Interrupt associated with TCC #11	W	0x0
10	I10	Interrupt associated with TCC #10	W	0x0
9	I9	Interrupt associated with TCC #9	W	0x0
8	I8	Interrupt associated with TCC #8	W	0x0
7	I7	Interrupt associated with TCC #7	W	0x0
6	I6	Interrupt associated with TCC #6	W	0x0
5	I5	Interrupt associated with TCC #5	W	0x0
4	I4	Interrupt associated with TCC #4	W	0x0
3	I3	Interrupt associated with TCC #3	W	0x0
2	I2	Interrupt associated with TCC #2	W	0x0
1	I1	Interrupt associated with TCC #1	W	0x0
0	I0	Interrupt associated with TCC #0	W	0x0

Table 11-221. Register Call Summary for Register EDMA_TPCC_IECR_RN_k

EDMA Register Manual

- EDMA Register Summary: [0][1][2][3]

Table 11-222. EDMA_TPCC_IECRH_RN_k

Address Offset	0x0000 205C + (0x200 * k)		
Physical Address	0x4330 205C + (0x200 * k) 0x40D1 205C + (0x200 * k) 0x4151 205C + (0x200 * k) 0x01D1 205C + (0x200 * k) 0x420A 205C + (0x200 * k)	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE_EDMA_TPCC
Description	Int Enable Clear Register (High Part) CPU write of '0' has no effect. CPU write of '1' to the EDMA_TPCC_IECRH.In bit causes the EDMA_TPCC_IERH.In bit to be cleared.		
Type	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I63	I62	I61	I60	I59	I58	I57	I56	I55	I54	I53	I52	I51	I50	I49	I48	I47	I46	I45	I44	I43	I42	I41	I40	I39	I38	I37	I36	I35	I34	I33	I32

Bits	Field Name	Description	Type	Reset
31	I63	Interrupt associated with TCC #63	W	0x0
30	I62	Interrupt associated with TCC #62	W	0x0
29	I61	Interrupt associated with TCC #61	W	0x0
28	I60	Interrupt associated with TCC #60	W	0x0
27	I59	Interrupt associated with TCC #59	W	0x0
26	I58	Interrupt associated with TCC #58	W	0x0
25	I57	Interrupt associated with TCC #57	W	0x0
24	I56	Interrupt associated with TCC #56	W	0x0
23	I55	Interrupt associated with TCC #55	W	0x0
22	I54	Interrupt associated with TCC #54	W	0x0
21	I53	Interrupt associated with TCC #53	W	0x0
20	I52	Interrupt associated with TCC #52	W	0x0
19	I51	Interrupt associated with TCC #51	W	0x0
18	I50	Interrupt associated with TCC #50	W	0x0
17	I49	Interrupt associated with TCC #49	W	0x0
16	I48	Interrupt associated with TCC #48	W	0x0
15	I47	Interrupt associated with TCC #47	W	0x0
14	I46	Interrupt associated with TCC #46	W	0x0
13	I45	Interrupt associated with TCC #45	W	0x0
12	I44	Interrupt associated with TCC #44	W	0x0
11	I43	Interrupt associated with TCC #43	W	0x0
10	I42	Interrupt associated with TCC #42	W	0x0
9	I41	Interrupt associated with TCC #41	W	0x0
8	I40	Interrupt associated with TCC #40	W	0x0
7	I39	Interrupt associated with TCC #39	W	0x0
6	I38	Interrupt associated with TCC #38	W	0x0
5	I37	Interrupt associated with TCC #37	W	0x0
4	I36	Interrupt associated with TCC #36	W	0x0
3	I35	Interrupt associated with TCC #35	W	0x0
2	I34	Interrupt associated with TCC #34	W	0x0
1	I33	Interrupt associated with TCC #33	W	0x0
0	I32	Interrupt associated with TCC #32	W	0x0

Table 11-223. Register Call Summary for Register EDMA_TPCC_IECRH_RN_k

EDMA Register Manual

- EDMA Register Summary: [0][1][2][3]

Table 11-224. EDMA_TPCC_IESR_RN_k

Address Offset	0x0000 2060 + (0x200 * k)		
Physical Address	0x4330 2060 + (0x200 * k) 0x40D1 2060 + (0x200 * k) 0x4151 2060 + (0x200 * k) 0x01D1 2060 + (0x200 * k) 0x420A 2060 + (0x200 * k)	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE_EDMA_TPCC
Description	Int Enable Set Register CPU write of '0' has no effect. CPU write of '1' to the EDMA_TPCC_IESR.In bit causes the EDMA_TPCC_IESR.In bit to be set.		
Type	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I31	I30	I29	I28	I27	I26	I25	I24	I23	I22	I21	I20	I19	I18	I17	I16	I15	I14	I13	I12	I11	I10	I9	I8	I7	I6	I5	I4	I3	I2	I1	I0

Bits	Field Name	Description	Type	Reset
31	I31	Interrupt associated with TCC #31	W	0x0
30	I30	Interrupt associated with TCC #30	W	0x0
29	I29	Interrupt associated with TCC #29	W	0x0
28	I28	Interrupt associated with TCC #28	W	0x0
27	I27	Interrupt associated with TCC #27	W	0x0
26	I26	Interrupt associated with TCC #26	W	0x0
25	I25	Interrupt associated with TCC #25	W	0x0
24	I24	Interrupt associated with TCC #24	W	0x0
23	I23	Interrupt associated with TCC #23	W	0x0
22	I22	Interrupt associated with TCC #22	W	0x0
21	I21	Interrupt associated with TCC #21	W	0x0
20	I20	Interrupt associated with TCC #20	W	0x0
19	I19	Interrupt associated with TCC #19	W	0x0
18	I18	Interrupt associated with TCC #18	W	0x0
17	I17	Interrupt associated with TCC #17	W	0x0
16	I16	Interrupt associated with TCC #16	W	0x0
15	I15	Interrupt associated with TCC #15	W	0x0
14	I14	Interrupt associated with TCC #14	W	0x0
13	I13	Interrupt associated with TCC #13	W	0x0
12	I12	Interrupt associated with TCC #12	W	0x0
11	I11	Interrupt associated with TCC #11	W	0x0
10	I10	Interrupt associated with TCC #10	W	0x0
9	I9	Interrupt associated with TCC #9	W	0x0
8	I8	Interrupt associated with TCC #8	W	0x0
7	I7	Interrupt associated with TCC #7	W	0x0
6	I6	Interrupt associated with TCC #6	W	0x0
5	I5	Interrupt associated with TCC #5	W	0x0
4	I4	Interrupt associated with TCC #4	W	0x0
3	I3	Interrupt associated with TCC #3	W	0x0
2	I2	Interrupt associated with TCC #2	W	0x0

Bits	Field Name	Description	Type	Reset
1	I1	Interrupt associated with TCC #1	W	0x0
0	I0	Interrupt associated with TCC #0	W	0x0

Table 11-225. Register Call Summary for Register EDMA_TPCC_IESR_RN_k

EDMA Register Manual
- EDMA Register Summary: [0][1][2][3]

Table 11-226. EDMA_TPCC_IESRH_RN_k

Address Offset	0x0000 2064 + (0x200 * k)		
Physical Address	0x4330 2064 + (0x200 * k) 0x40D1 2064 + (0x200 * k) 0x4151 2064 + (0x200 * k) 0x01D1 2064 + (0x200 * k) 0x420A 2064 + (0x200 * k)	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE_EDMA_TPCC
Description	Int Enable Set Register (High Part) CPU write of '0' has no effect. CPU write of '1' to the EDMA_TPCC_IESRH.In bit causes the EDMA_TPCC_IESRH.In bit to be set.		
Type	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I63	I62	I61	I60	I59	I58	I57	I56	I55	I54	I53	I52	I51	I50	I49	I48	I47	I46	I45	I44	I43	I42	I41	I40	I39	I38	I37	I36	I35	I34	I33	I32

Bits	Field Name	Description	Type	Reset
31	I63	Interrupt associated with TCC #63	W	0x0
30	I62	Interrupt associated with TCC #62	W	0x0
29	I61	Interrupt associated with TCC #61	W	0x0
28	I60	Interrupt associated with TCC #60	W	0x0
27	I59	Interrupt associated with TCC #59	W	0x0
26	I58	Interrupt associated with TCC #58	W	0x0
25	I57	Interrupt associated with TCC #57	W	0x0
24	I56	Interrupt associated with TCC #56	W	0x0
23	I55	Interrupt associated with TCC #55	W	0x0
22	I54	Interrupt associated with TCC #54	W	0x0
21	I53	Interrupt associated with TCC #53	W	0x0
20	I52	Interrupt associated with TCC #52	W	0x0
19	I51	Interrupt associated with TCC #51	W	0x0
18	I50	Interrupt associated with TCC #50	W	0x0
17	I49	Interrupt associated with TCC #49	W	0x0
16	I48	Interrupt associated with TCC #48	W	0x0
15	I47	Interrupt associated with TCC #47	W	0x0
14	I46	Interrupt associated with TCC #46	W	0x0
13	I45	Interrupt associated with TCC #45	W	0x0
12	I44	Interrupt associated with TCC #44	W	0x0
11	I43	Interrupt associated with TCC #43	W	0x0
10	I42	Interrupt associated with TCC #42	W	0x0
9	I41	Interrupt associated with TCC #41	W	0x0
8	I40	Interrupt associated with TCC #40	W	0x0
7	I39	Interrupt associated with TCC #39	W	0x0

Bits	Field Name	Description	Type	Reset
6	I38	Interrupt associated with TCC #38	W	0x0
5	I37	Interrupt associated with TCC #37	W	0x0
4	I36	Interrupt associated with TCC #36	W	0x0
3	I35	Interrupt associated with TCC #35	W	0x0
2	I34	Interrupt associated with TCC #34	W	0x0
1	I33	Interrupt associated with TCC #33	W	0x0
0	I32	Interrupt associated with TCC #32	W	0x0

Table 11-227. Register Call Summary for Register EDMA_TPCC_IESRH_RN_k

EDMA Register Manual
- EDMA Register Summary: [0][1][2][3]

Table 11-228. EDMA_TPCC_IPR_RN_k

Address Offset	0x0000 2068 + (0x200 * k)		
Physical Address	0x4330 2068 + (0x200 * k) 0x40D1 2068 + (0x200 * k) 0x4151 2068 + (0x200 * k) 0x01D1 2068 + (0x200 * k) 0x420A 2068 + (0x200 * k)	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE_EDMA_TPCC
Description	Interrupt Pending Register EDMA_TPCC_IPR.In bit is set when a interrupt completion code with TCC of N is detected. EDMA_TPCC_IPR.In bit is cleared via software by writing a '1' to EDMA_TPCC_ICR.In bit.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I31	I30	I29	I28	I27	I26	I25	I24	I23	I22	I21	I20	I19	I18	I17	I16	I15	I14	I13	I12	I11	I10	I9	I8	I7	I6	I5	I4	I3	I2	I1	I0

Bits	Field Name	Description	Type	Reset
31	I31	Interrupt associated with TCC #31	R	0x0
30	I30	Interrupt associated with TCC #30	R	0x0
29	I29	Interrupt associated with TCC #29	R	0x0
28	I28	Interrupt associated with TCC #28	R	0x0
27	I27	Interrupt associated with TCC #27	R	0x0
26	I26	Interrupt associated with TCC #26	R	0x0
25	I25	Interrupt associated with TCC #25	R	0x0
24	I24	Interrupt associated with TCC #24	R	0x0
23	I23	Interrupt associated with TCC #23	R	0x0
22	I22	Interrupt associated with TCC #22	R	0x0
21	I21	Interrupt associated with TCC #21	R	0x0
20	I20	Interrupt associated with TCC #20	R	0x0
19	I19	Interrupt associated with TCC #19	R	0x0
18	I18	Interrupt associated with TCC #18	R	0x0
17	I17	Interrupt associated with TCC #17	R	0x0
16	I16	Interrupt associated with TCC #16	R	0x0
15	I15	Interrupt associated with TCC #15	R	0x0
14	I14	Interrupt associated with TCC #14	R	0x0
13	I13	Interrupt associated with TCC #13	R	0x0
12	I12	Interrupt associated with TCC #12	R	0x0
11	I11	Interrupt associated with TCC #11	R	0x0

Bits	Field Name	Description	Type	Reset
10	I10	Interrupt associated with TCC #10	R	0x0
9	I9	Interrupt associated with TCC #9	R	0x0
8	I8	Interrupt associated with TCC #8	R	0x0
7	I7	Interrupt associated with TCC #7	R	0x0
6	I6	Interrupt associated with TCC #6	R	0x0
5	I5	Interrupt associated with TCC #5	R	0x0
4	I4	Interrupt associated with TCC #4	R	0x0
3	I3	Interrupt associated with TCC #3	R	0x0
2	I2	Interrupt associated with TCC #2	R	0x0
1	I1	Interrupt associated with TCC #1	R	0x0
0	I0	Interrupt associated with TCC #0	R	0x0

Table 11-229. Register Call Summary for Register EDMA_TPCC_IPR_RN_k

EDMA Register Manual
- EDMA Register Summary: [0][1][2][3]

Table 11-230. EDMA_TPCC_IPRH_RN_k

Address Offset	0x0000 206C + (0x200 * k)		
Physical Address	0x4330 206C + (0x200 * k) 0x40D1 206C + (0x200 * k) 0x4151 206C + (0x200 * k) 0x01D1 206C + (0x200 * k) 0x420A 206C + (0x200 * k)	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE_EDMA_TPCC
Description	Interrupt Pending Register (High Part) EDMA_TPCC_IPRH.In bit is set when a interrupt completion code with TCC of N is detected. EDMA_TPCC_IPRH.In bit is cleared via software by writing a '1' to EDMA_TPCC_ICRH.In bit.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I63	I62	I61	I60	I59	I58	I57	I56	I55	I54	I53	I52	I51	I50	I49	I48	I47	I46	I45	I44	I43	I42	I41	I40	I39	I38	I37	I36	I35	I34	I33	I32

Bits	Field Name	Description	Type	Reset
31	I63	Interrupt associated with TCC #63	R	0x0
30	I62	Interrupt associated with TCC #62	R	0x0
29	I61	Interrupt associated with TCC #61	R	0x0
28	I60	Interrupt associated with TCC #60	R	0x0
27	I59	Interrupt associated with TCC #59	R	0x0
26	I58	Interrupt associated with TCC #58	R	0x0
25	I57	Interrupt associated with TCC #57	R	0x0
24	I56	Interrupt associated with TCC #56	R	0x0
23	I55	Interrupt associated with TCC #55	R	0x0
22	I54	Interrupt associated with TCC #54	R	0x0
21	I53	Interrupt associated with TCC #53	R	0x0
20	I52	Interrupt associated with TCC #52	R	0x0
19	I51	Interrupt associated with TCC #51	R	0x0
18	I50	Interrupt associated with TCC #50	R	0x0
17	I49	Interrupt associated with TCC #49	R	0x0
16	I48	Interrupt associated with TCC #48	R	0x0
15	I47	Interrupt associated with TCC #47	R	0x0

Bits	Field Name	Description	Type	Reset
14	I46	Interrupt associated with TCC #46	R	0x0
13	I45	Interrupt associated with TCC #45	R	0x0
12	I44	Interrupt associated with TCC #44	R	0x0
11	I43	Interrupt associated with TCC #43	R	0x0
10	I42	Interrupt associated with TCC #42	R	0x0
9	I41	Interrupt associated with TCC #41	R	0x0
8	I40	Interrupt associated with TCC #40	R	0x0
7	I39	Interrupt associated with TCC #39	R	0x0
6	I38	Interrupt associated with TCC #38	R	0x0
5	I37	Interrupt associated with TCC #37	R	0x0
4	I36	Interrupt associated with TCC #36	R	0x0
3	I35	Interrupt associated with TCC #35	R	0x0
2	I34	Interrupt associated with TCC #34	R	0x0
1	I33	Interrupt associated with TCC #33	R	0x0
0	I32	Interrupt associated with TCC #32	R	0x0

Table 11-231. Register Call Summary for Register EDMA_TPCC_IPRH_RN_k

EDMA Register Manual
- EDMA Register Summary: [0][1][2][3]

Table 11-232. EDMA_TPCC_ICR_RN_k

Address Offset	0x0000 2070 + (0x200 * k)		
Physical Address	0x4330 2070 + (0x200 * k) 0x40D1 2070 + (0x200 * k) 0x4151 2070 + (0x200 * k) 0x01D1 2070 + (0x200 * k) 0x420A 2070 + (0x200 * k)	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE_EDMA_TPCC
Description	Interrupt Clear Register CPU writes of '0' has no effect. CPU write of '1' to the EDMA_TPCC_ICR.In bit causes the EDMA_TPCC_IPR.In bit to be cleared. All EDMA_TPCC_IPR.In bits must be cleared before additional interrupts will be asserted by CC.		
Type	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I31	I30	I29	I28	I27	I26	I25	I24	I23	I22	I21	I20	I19	I18	I17	I16	I15	I14	I13	I12	I11	I10	I9	I8	I7	I6	I5	I4	I3	I2	I1	I0

Bits	Field Name	Description	Type	Reset
31	I31	Interrupt associated with TCC #31	W	0x0
30	I30	Interrupt associated with TCC #30	W	0x0
29	I29	Interrupt associated with TCC #29	W	0x0
28	I28	Interrupt associated with TCC #28	W	0x0
27	I27	Interrupt associated with TCC #27	W	0x0
26	I26	Interrupt associated with TCC #26	W	0x0
25	I25	Interrupt associated with TCC #25	W	0x0
24	I24	Interrupt associated with TCC #24	W	0x0
23	I23	Interrupt associated with TCC #23	W	0x0
22	I22	Interrupt associated with TCC #22	W	0x0
21	I21	Interrupt associated with TCC #21	W	0x0
20	I20	Interrupt associated with TCC #20	W	0x0

Bits	Field Name	Description	Type	Reset
19	I19	Interrupt associated with TCC #19	W	0x0
18	I18	Interrupt associated with TCC #18	W	0x0
17	I17	Interrupt associated with TCC #17	W	0x0
16	I16	Interrupt associated with TCC #16	W	0x0
15	I15	Interrupt associated with TCC #15	W	0x0
14	I14	Interrupt associated with TCC #14	W	0x0
13	I13	Interrupt associated with TCC #13	W	0x0
12	I12	Interrupt associated with TCC #12	W	0x0
11	I11	Interrupt associated with TCC #11	W	0x0
10	I10	Interrupt associated with TCC #10	W	0x0
9	I9	Interrupt associated with TCC #9	W	0x0
8	I8	Interrupt associated with TCC #8	W	0x0
7	I7	Interrupt associated with TCC #7	W	0x0
6	I6	Interrupt associated with TCC #6	W	0x0
5	I5	Interrupt associated with TCC #5	W	0x0
4	I4	Interrupt associated with TCC #4	W	0x0
3	I3	Interrupt associated with TCC #3	W	0x0
2	I2	Interrupt associated with TCC #2	W	0x0
1	I1	Interrupt associated with TCC #1	W	0x0
0	I0	Interrupt associated with TCC #0	W	0x0

Table 11-233. Register Call Summary for Register EDMA_TPCC_ICR_RN_k

EDMA Register Manual
- EDMA Register Summary: [0][1][2][3]

Table 11-234. EDMA_TPCC_ICRH_RN_k

Address Offset	0x0000 2074 + (0x200 * k)		
Physical Address	0x4330 2074 + (0x200 * k) 0x40D1 2074 + (0x200 * k) 0x4151 2074 + (0x200 * k) 0x01D1 2074 + (0x200 * k) 0x420A 2074 + (0x200 * k)	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE_EDMA_TPCC
Description	Interrupt Clear Register (High Part) CPU write of '0' has no effect. CPU write of '1' to the EDMA_TPCC_ICRH.In bit causes the EDMA_TPCC_IPRH.In bit to be cleared. All EDMA_TPCC_IPRH.In bits must be cleared before additional interrupts will be asserted by CC.		
Type	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I63	I62	I61	I60	I59	I58	I57	I56	I55	I54	I53	I52	I51	I50	I49	I48	I47	I46	I45	I44	I43	I42	I41	I40	I39	I38	I37	I36	I35	I34	I33	I32

Bits	Field Name	Description	Type	Reset
31	I63	Interrupt associated with TCC #63	W	0x0
30	I62	Interrupt associated with TCC #62	W	0x0
29	I61	Interrupt associated with TCC #61	W	0x0
28	I60	Interrupt associated with TCC #60	W	0x0
27	I59	Interrupt associated with TCC #59	W	0x0
26	I58	Interrupt associated with TCC #58	W	0x0

Bits	Field Name	Description	Type	Reset
25	I57	Interrupt associated with TCC #57	W	0x0
24	I56	Interrupt associated with TCC #56	W	0x0
23	I55	Interrupt associated with TCC #55	W	0x0
22	I54	Interrupt associated with TCC #54	W	0x0
21	I53	Interrupt associated with TCC #53	W	0x0
20	I52	Interrupt associated with TCC #52	W	0x0
19	I51	Interrupt associated with TCC #51	W	0x0
18	I50	Interrupt associated with TCC #50	W	0x0
17	I49	Interrupt associated with TCC #49	W	0x0
16	I48	Interrupt associated with TCC #48	W	0x0
15	I47	Interrupt associated with TCC #47	W	0x0
14	I46	Interrupt associated with TCC #46	W	0x0
13	I45	Interrupt associated with TCC #45	W	0x0
12	I44	Interrupt associated with TCC #44	W	0x0
11	I43	Interrupt associated with TCC #43	W	0x0
10	I42	Interrupt associated with TCC #42	W	0x0
9	I41	Interrupt associated with TCC #41	W	0x0
8	I40	Interrupt associated with TCC #40	W	0x0
7	I39	Interrupt associated with TCC #39	W	0x0
6	I38	Interrupt associated with TCC #38	W	0x0
5	I37	Interrupt associated with TCC #37	W	0x0
4	I36	Interrupt associated with TCC #36	W	0x0
3	I35	Interrupt associated with TCC #35	W	0x0
2	I34	Interrupt associated with TCC #34	W	0x0
1	I33	Interrupt associated with TCC #33	W	0x0
0	I32	Interrupt associated with TCC #32	W	0x0

Table 11-235. Register Call Summary for Register EDMA_TPCC_ICRH_RN_k

EDMA Register Manual
- EDMA Register Summary: [0][1][2][3]

Table 11-236. EDMA_TPCC_IEVAL_RN_k

Address Offset	0x0000 2078 + (0x200 * k)		
Physical Address	0x4330 2078 + (0x200 * k) 0x40D1 2078 + (0x200 * k) 0x4151 2078 + (0x200 * k) 0x01D1 2078 + (0x200 * k) 0x420A 2078 + (0x200 * k)	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE_EDMA_TPCC
Description	Interrupt Eval Register		
Type	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														SET	EVAL

Bits	Field Name	Description	Type	Reset
31:2	RESERVED	Reserved	R	0x0
1	SET	Interrupt Set CPU writes 0x0 has no effect. CPU writes 0x1 to the SETn bit causes the tpcc_intN output signal to be pulsed egardless of state of interrupts enable (IERn) and status (EDMA_TPCC_IPRn).	W	0x0
0	EVAL	Interrupt Evaluate CPU writes 0x0 has no effect. CPU writes 0x1 to the EVALn bit causes the tpcc_intN output signal to be pulsed if any enabled interrupts (IERn) are still pending (EDMA_TPCC_IPRn).	W	0x0

Table 11-237. Register Call Summary for Register EDMA_TPCC_IEVAL_RN_k

EDMA Controller Functional Description
- Region Overview: [0][1]

EDMA Register Manual
- EDMA Register Summary: [2][3][4][5]

Table 11-238. EDMA_TPCC_QER_RN_k

Address Offset	0x0000 2080 + (0x200 * k)		
Physical Address	0x4330 2080 + (0x200 * k) 0x40D1 2080 + (0x200 * k) 0x4151 2080 + (0x200 * k) 0x01D1 2080 + (0x200 * k) 0x420A 2080 + (0x200 * k)	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE_EDMA_TPCC
Description	QDMA Event Register If EDMA_TPCC_QER.En bit is set, then the corresponding QDMA channel is prioritized vs. other qdma events for submission to the TC. EDMA_TPCC_QER.En bit is set when a vbus write byte matches the address defined in the QCHMAPn register. EDMA_TPCC_QER.En bit is cleared when the corresponding event is prioritized and serviced. EDMA_TPCC_QER.En is also cleared when user writes a '1' to the EDMA_TPCC_QSECR.En bit. If the EDMA_TPCC_QER.En bit is already set and a new QDMA event is detected due to user write to QDMA trigger location and EDMA_TPCC_QEER register is set, then the corresponding bit in the QDMA Event Missed Register is set.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								E7	E6	E5	E4	E3	E2	E1	E0

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Reserved	R Return 0's	0x0
7	E7	Event #7	R	0x0
6	E6	Event #6	R	0x0
5	E5	Event #5	R	0x0
4	E4	Event #4	R	0x0
3	E3	Event #3	R	0x0
2	E2	Event #2	R	0x0
1	E1	Event #1	R	0x0
0	E0	Event #0	R	0x0

Table 11-239. Register Call Summary for Register EDMA_TPCC_QER_RN_k

EDMA Register Manual
- EDMA Register Summary: [0][1][2][3]

Table 11-240. EDMA_TPCC_QEER_RN_k

Address Offset	0x0000 2084 + (0x200 * k)		
Physical Address	0x4330 2084 + (0x200 * k) 0x40D1 2084 + (0x200 * k) 0x4151 2084 + (0x200 * k) 0x01D1 2084 + (0x200 * k) 0x420A 2084 + (0x200 * k)	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE_EDMA_TPCC
Description	QDMA Event Enable Register Enabled/disabled QDMA address comparator for QDMA Channel N. EDMA_TPCC_QEER.En is not directly writeable. The corresponding QDMA channel comparator is enabled and Events will be recognized and latched in EDMA_TPCC_QER.En. EDMA_TPCC_QEER.En = 0, The corresponding QDMA channel comparator is disabled. Events will not be recognized/latched in EDMA_TPCC_QER.En. QDMA channels can be enabled via writes to EDMA_TPCC_QEESR and can be disabled via writes to EDMA_TPCC_QEECR register. EDMA_TPCC_QEER.En = 1,		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								E7	E6	E5	E4	E3	E2	E1	E0

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Reserved	R Return 0's	0x0
7	E7	Event #7	R	0x0
6	E6	Event #6	R	0x0
5	E5	Event #5	R	0x0
4	E4	Event #4	R	0x0
3	E3	Event #3	R	0x0
2	E2	Event #2	R	0x0
1	E1	Event #1	R	0x0
0	E0	Event #0	R	0x0

Table 11-241. Register Call Summary for Register EDMA_TPCC_QEER_RN_k

EDMA Register Manual
- EDMA Register Summary: [0][1][2][3]

Table 11-242. EDMA_TPCC_QEECR_RN_k

Address Offset	0x0000 2088 + (0x200 * k)		
Physical Address	0x4330 2088 + (0x200 * k) 0x40D1 2088 + (0x200 * k) 0x4151 2088 + (0x200 * k) 0x01D1 2088 + (0x200 * k) 0x420A 2088 + (0x200 * k)	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE_EDMA_TPCC
Description	QDMA Event Enable Clear Register CPU write of '0' has no effect. CPU write of '1' to the EDMA_TPCC_QEECR.En bit causes the EDMA_TPCC_QEER.En bit to be cleared.		
Type	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								E7	E6	E5	E4	E3	E2	E1	E0

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Reserved	R	0x0
7	E7	Event #7	W	0x0
6	E6	Event #6	W	0x0
5	E5	Event #5	W	0x0
4	E4	Event #4	W	0x0
3	E3	Event #3	W	0x0
2	E2	Event #2	W	0x0
1	E1	Event #1	W	0x0
0	E0	Event #0	W	0x0

Table 11-243. Register Call Summary for Register EDMA_TPCC_QEECR_RN_k

EDMA Register Manual
- EDMA Register Summary: [0][1][2][3]

Table 11-244. EDMA_TPCC_QEESR_RN_k

Address Offset	0x0000 208C + (0x200 * k)		
Physical Address	0x4330 208C + (0x200 * k) 0x40D1 208C + (0x200 * k) 0x4151 208C + (0x200 * k) 0x01D1 208C + (0x200 * k) 0x420A 208C + (0x200 * k)	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE_EDMA_TPCC
Description	QDMA Event Enable Set Register CPU write of '0' has no effect. CPU write of '1' to the EDMA_TPCC_QEESR.En bit causes the EDMA_TPCC_QEESR.En bit to be set.		
Type	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								E7	E6	E5	E4	E3	E2	E1	E0

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Reserved	R	0x0
7	E7	Event #7	W	0x0
6	E6	Event #6	W	0x0
5	E5	Event #5	W	0x0
4	E4	Event #4	W	0x0
3	E3	Event #3	W	0x0
2	E2	Event #2	W	0x0
1	E1	Event #1	W	0x0
0	E0	Event #0	W	0x0

Table 11-245. Register Call Summary for Register EDMA_TPCC_QEESR_RN_k

EDMA Register Manual
- EDMA Register Summary: [0][1][2][3]

Table 11-246. EDMA_TPCC_QSER_RN_k

Address Offset	0x0000 2090 + (0x200 * k)		
Physical Address	0x4330 2090 + (0x200 * k) 0x40D1 2090 + (0x200 * k) 0x4151 2090 + (0x200 * k) 0x01D1 2090 + (0x200 * k) 0x420A 2090 + (0x200 * k)	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE_EDMA_TPCC
Description	QDMA Secondary Event Register The QDMA secondary event register is used along with the QDMA Event Register (EDMA_TPCC_QER) to provide information on the state of a QDMA Event. En = 0 : Event is not currently in the Event Queue. En = 1 : Event is currently stored in Event Queue. Event arbiter will not prioritize additional events.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								E7	E6	E5	E4	E3	E2	E1	E0

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Reserved	R Return 0's	0x0
7	E7	Event #7	R	0x0
6	E6	Event #6	R	0x0
5	E5	Event #5	R	0x0
4	E4	Event #4	R	0x0
3	E3	Event #3	R	0x0
2	E2	Event #2	R	0x0
1	E1	Event #1	R	0x0
0	E0	Event #0	R	0x0

Table 11-247. Register Call Summary for Register EDMA_TPCC_QSER_RN_k

EDMA Register Manual

- EDMA Register Summary: [0][1][2][3]

Table 11-248. EDMA_TPCC_QSECR_RN_k

Address Offset	0x0000 2094 + (0x200 * k)		
Physical Address	0x4330 2094 + (0x200 * k) 0x40D1 2094 + (0x200 * k) 0x4151 2094 + (0x200 * k) 0x01D1 2094 + (0x200 * k) 0x420A 2094 + (0x200 * k)	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE_EDMA_TPCC
Description	QDMA Secondary Event Clear Register CPU write of '0' has no effect. The secondary event clear register is used to clear the status of the EDMA_TPCC_QSER and EDMA_TPCC_QER register (note that this is slightly different than the EDMA_TPCC_SER operation, which does not clear the EDMA_TPCC_ER.En register). CPU write of '1' to the EDMA_TPCC_QSECR.En bit clears the EDMA_TPCC_QSER.En and EDMA_TPCC_QER.En register fields.		
Type	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								E7	E6	E5	E4	E3	E2	E1	E0

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	write 0's for future compatibility	W	0x0
7	E7	Event #7	W	0x0
6	E6	Event #6	W	0x0
5	E5	Event #5	W	0x0
4	E4	Event #4	W	0x0
3	E3	Event #3	W	0x0
2	E2	Event #2	W	0x0
1	E1	Event #1	W	0x0
0	E0	Event #0	W	0x0

Table 11-249. Register Call Summary for Register EDMA_TPCC_QSECR_RN_k

EDMA Register Manual
- EDMA Register Summary: [0][1][2][3]

Table 11-250. EDMA_TPCC_OPT_n

Address Offset	0x0000 4000 + (0x20 * n)		
Physical Address	0x4330 4000 + (0x20 * n) 0x40D1 4000 + (0x20 * n) 0x4151 4000 + (0x20 * n) 0x01D1 4000 + (0x20 * n) 0x420A 4000 + (0x20 * n)	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE_EDMA_TPCC
Description	Options Parameter		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PRIV	RESERVED				PRIVID			ITCCHEN	TCCHEN	ITCINTEN	TCINTEN	WIMODE	RESERVED	TCC						TCCMODE	FWID			RESERVED				STATIC	SYNCDIM	DAM	SAM

Bits	Field Name	Description	Type	Reset
31	PRIV	Privilege level privilege level (supervisor vs. user) for the host/cpu/dma that programmed this PaRAM Entry. Value is set with the vbus priv value when any part of the PaRAM Entry is written. Not writeable via vbus wdata bus. Is readable via VBus rdata bus. 0x0: User level privilege 0x1: Supervisor level privilege	R	0x0
30:28	RESERVED	Reserved	R	0x0
27:24	PRIVID	Privilege ID Privilege ID for the external host/cpu/dma that programmed this PaRAM Entry. This value is set with the vbus privid value when any part of the PaRAM Entry is written. Not writeable via vbus wdata bus. Is readable via VBus rdata bus.	R	0x0
23	ITCCHEN	Intermediate transfer completion chaining enable 0x0: Intermediate transfer complete chaining is disabled. 0x1: Intermediate transfer complete chaining is enabled.	RW	0x0
22	TCCHEN	Transfer complete chaining enable 0x0: Transfer complete chaining is disabled. 0x1: Transfer complete chaining is enabled.	RW	0x0

Bits	Field Name	Description	Type	Reset
21	ITCINTEN	Intermediate transfer completion interrupt enable 0x0: Intermediate transfer complete interrupt is disabled. 0x1: Intermediate transfer complete interrupt is enabled (corresponding EDMA_TPCC_IER[TCC] bit must be set to 1 to generate interrupt)	RW	0x0
20	TCINTEN	Transfer complete interrupt enable 0x0: Transfer complete interrupt is disabled. 0x1: Transfer complete interrupt is enabled (corresponding EDMA_TPCC_IER[TCC] bit must be set to 1 to generate interrupt)	RW	0x0
19	WIMODE	Backward compatibility mode 0x0: Normal operation 0x1: WI Backwards Compatibility mode, forces BCNT to be adjusted by '1' upon TR submission (0 means 1, 1 means 2, ...) and forces ACNT to be treated as a word-count (left shifted by 2 by hardware to create byte cnt for TR submission)	RW	0x0
18	RESERVED	Reserved	R	0x0
17:12	TCC	Transfer Complete Code The 6-bit code is used to set the relevant bit in EDMA_TPCC_CER (bit EDMA_TPCC_CER[TCC]) for chaining or in EDMA_TPCC_IER (bit EDMA_TPCC_IER[TCC]) for interrupts.	RW	0x0
11	TCCMODE	Transfer complete code mode: Indicates the point at which a transfer is considered completed. Applies to both chaining and interrupt. 0x0: Normal Completion. A transfer is considered completed after the transfer parameters are returned to the CC from the TC (which was returned from the peripheral) 0x1: Early Completion, A transfer is considered completed after the CC submits a TR to the TC. CC generates completion code internally.	RW	0x0
10:8	FWID	FIFO width: Applies if either SAM or DAM is set to FIFO mode. Pass-thru to TC.	RW	0x0
7:4	RESERVED	Reserved	R	0x0
3	STATIC	Static Entry 0x0: Entry is updated as normal 0x1: Entry is static, Count and Address updates are not updated after TRP is submitted. Linking is not performed.	RW	0x0
2	SYNCDIM	Transfer Synchronization Dimension: 0x0: A-Sync, Each event triggers the transfer of ACNT elements. 0x1: AB-Sync, Each event triggers the transfer of BCNT arrays of ACNT elements.	RW	0x0
1	DAM	Destination Address Mode: Destination Address Mode within an array. Pass-thru to TC. 0x0: INCR, Dst addressing within an array increments. Dst is not a FIFO. 0x1: FIFO, Dst addressing within an array wraps around upon reaching FIFO width.	RW	0x0
0	SAM	Source Address Mode: Source Address Mode within an array. Pass-thru to TC. 0x0: INCR, Src addressing within an array increments. Source is not a FIFO. 0x1: FIFO, Src addressing within an array wraps around upon reaching FIFO width.	RW	0x0

Table 11-251. Register Call Summary for Register EDMA_TPCC_OPT_n

EDMA Controller Functional Description

- Types of EDMA controller Transfers: [0]
- PaRAM: [1]
- EDMA Channel PaRAM Set Entry Fields: [2][3][4][5]
- Dummy Versus Null Transfer Comparison: [6]
- Parameter Set Updates: [7][8][9][10][11][12]
- Linking Transfers: [13][14][15]
- Constant Addressing Mode Transfers/Alignment Issues: [16][17][18][19][20][21]
- DMA Channels: [22]
- Comparison Between DMA and QDMA Channels: [23]
- Completion of a DMA Transfer: [24][25][26][27][28][29][30]
- Normal Completion: [31]
- Early Completion: [32]
- Channel Controller Regions: [33]
- Chaining EDMA Channels: [34][35][36][37][38][39][40][41][42][43][44][45]
- Transfer Completion Interrupts: [46][47][48][49][50][51][52][53][54][55][56][57][58][59][60][61][62][63][64][65][66][67][68][69][70][71]
- Error Interrupts: [72][73]
- Active Memory Protection: [74][75][76][77]
- Proxy Memory Protection: [78][79][80][81][82][83][84][85][86][87]

EDMA Transfer Examples

- Block Move Example: [88][89][90]
- Subframe Extraction Example: [91][92][93][94]
- Data Sorting Example: [95][96][97][98]
- Non-bursting Peripherals: [99]
- Bursting Peripherals: [100][101]
- Continuous Operation: [102][103][104][105]
- Ping-Pong Buffering: [106][107][108]
- Transfer Chaining Examples: [109][110][111][112][113][114][115][116][117]

EDMA Debug Checklist and Programming Tips

- EDMA Debug Checklist: [118][119][120][121][122][123]

EDMA Register Manual

- EDMA Register Summary: [124][125][126][127]
- EDMA Register Description: [128][129][130][131][132][133][134][135][136]

Table 11-252. EDMA_TPCC_SRC_n

Address Offset	0x0000 4004 + (0x20 * n)		
Physical Address	0x4330 4004 + (0x20 * n) 0x40D1 4004 + (0x20 * n) 0x4151 4004 + (0x20 * n) 0x01D1 4004 + (0x20 * n) 0x420A 4004 + (0x20 * n)	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE_EDMA_TPCC
Description	Source Address		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SRC																															

Bits	Field Name	Description	Type	Reset
31:0	SRC	Source Address The 32-bit source address parameters specify the starting byte address of the source. If SAM is set to FIFO mode then the user should program the Source address to be aligned to the value specified by the EDMA_TPCC_OPT_n[10:8] FWID field. No errors are recognized here but TC will assert error if this is not true.	RW	0x0

Table 11-253. Register Call Summary for Register EDMA_TPCC_SRC_n

EDMA Controller Functional Description
- PaRAM: [0]
- Parameter Set Updates: [1][2]

EDMA Register Manual
- EDMA Register Summary: [3][4][5][6]

Table 11-254. EDMA_TPCC_ABCNT_n

Address Offset	0x0000 4008 + (0x20 * n)		
Physical Address	0x4330 4008 + (0x20 * n) 0x40D1 4008 + (0x20 * n) 0x4151 4008 + (0x20 * n) 0x01D1 4008 + (0x20 * n) 0x420A 4008 + (0x20 * n)	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE_EDMA_TPCC
Description	A and B byte count		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BCNT																ACNT															

Bits	Field Name	Description	Type	Reset
31:16	BCNT	BCNT: Count for 2nd Dimension: BCNT is a 16-bit unsigned value that specifies the number of arrays of length ACNT. For normal operation, valid values for BCNT can be anywhere between 1 and 65535. Therefore, the maximum number of arrays in a frame is 65535 (64K-1 arrays). BCNT=1 means 1 array in the frame, and BCNT=0 means 0 arrays in the frame. In normal mode, a BCNT of '0' is considered as either a Null or Dummy transfer. A Dummy or Null transfer will generate a Completion code depending on the settings of the completion bit fields of the OPT field. If the EDMA_TPCC_OPT_n.WIMODE bit is set, then the programmed BCNT value will be incremented by '1' before submission to TC. I.e., 0 means 1, 1 means 2, 2 means 3, ..., 0xFFFE means 0xFFFF. A value of 0xFFFF is an illegal value that will be treated as a Null TR.	RW	0x0

Bits	Field Name	Description	Type	Reset
15:0	ACNT	ACNT: number of bytes in 1st dimension: ACNT represents the number of bytes within the first dimension of a transfer. ACNT is a 16-bit unsigned value with valid values between 0 and 65535. Therefore, the maximum number of bytes in an array is 65535 bytes (64K-1 bytes). ACNT must be greater than or equal to '1' for a TR to be submitted to TC. An ACNT of '0' is considered as either a null or dummy transfer. A Dummy or Null transfer will generate a Completion code depending on the settings of the completion bit fields of the OPT field. If the EDMA_TPCC_OPT_n.WIMODE bit is set then the ACNT field represents a word count. The CC must internally multiply by 4 to translate the word count to a byte count prior to submission to the TC. The 2 MSBs of the 16-bit ACNT are reserved and should always be written as 'b00 by the user. If user writes a value other than 0, it will still be treated as 0 since the multiply-by-4 operation (to translate between a word count and a byte count) will drop the 2 msbits. For dummy and null transfer definition, the ACNT definition will disregard the 2 msbits. I.e., a programmed ACNT value of 0x8000 in WI-mode will be treated as 0 byte transfer, resulting in null or dummy operation dependent on the state of BCNT and CCNT.	RW	0x0

Table 11-255. Register Call Summary for Register EDMA_TPCC_ABCNT_n

EDMA Controller Functional Description

- Types of EDMA controller Transfers: [0][1]
- A-Synchronized Transfers: [2]
- PaRAM: [3][4]
- EDMA Channel PaRAM Set Entry Fields: [5][6][7]
- Null PaRAM Set: [8][9]
- Dummy PaRAM Set: [10][11]
- Parameter Set Updates: [12][13][14][15][16][17][18][19][20][21][22][23][24]
- Element Size: [25][26][27][28][29][30]
- Comparison Between DMA and QDMA Channels: [31]
- Completion of a DMA Transfer: [32][33]
- Event, Channel, and PaRAM Mapping: [34][35]
- Chaining EDMA Channels: [36][37]
- Transfer Completion Interrupts: [38]
- Architecture Details: [39][40][41][42][43][44][45]

EDMA Transfer Examples

- Block Move Example: [46]
- Bursting Peripherals: [47][48]

EDMA Register Manual

- EDMA Register Summary: [49][50][51][52]

Table 11-256. EDMA_TPCC_DST_n

Address Offset	0x0000 400C + (0x20 * n)		
Physical Address	0x4330 400C + (0x20 * n) 0x40D1 400C + (0x20 * n) 0x4151 400C + (0x20 * n) 0x01D1 400C + (0x20 * n) 0x420A 400C + (0x20 * n)	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE_EDMA_TPCC
Description	Destination Address		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DST																															

Bits	Field Name	Description	Type	Reset
31:0	DST	Destination Address: The 32-bit destination address parameters specify the starting byte address of the destination. If DAM is set to FIFO mode then the user should program the Destination address to be aligned to the value specified by the EDMA_TPCC_OPT_n.FWID field. No errors are recognized here but TC will assert error if this is not true.	RW	0x0

Table 11-257. Register Call Summary for Register EDMA_TPCC_DST_n

EDMA Controller Functional Description
- PaRAM: [0]
- Parameter Set Updates: [1][2]

EDMA Register Manual
- EDMA Register Summary: [3][4][5][6]

Table 11-258. EDMA_TPCC_BIDX_n

Address Offset	0x0000 4010 + (0x20 * n)		
Physical Address	0x4330 4010 + (0x20 * n) 0x40D1 4010 + (0x20 * n) 0x4151 4010 + (0x20 * n) 0x01D1 4010 + (0x20 * n) 0x420A 4010 + (0x20 * n)	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE_EDMA_TPCC
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DBIDX																SBIDX															

Bits	Field Name	Description	Type	Reset
31:16	DBIDX	Destination 2nd Dimension Index: DBIDX is a 16-bit signed value (2's complement) used for destination address modification in between each array in the 2nd dimension. It is a signed value between -32768 and 32767. It provides a byte address offset from the beginning of the destination array to the beginning of the next destination array within the current frame. It applies to both A-Sync and AB-Sync transfers.	RW	0x0
15:0	SBIDX	Source 2nd Dimension Index: SBIDX is a 16-bit signed value (2's complement) used for source address modification in between each array in the 2nd dimension. It is a signed value between -32768 and 32767. It provides a byte address offset from the beginning of the source array to the beginning of the next source array. It applies to both A-sync and AB-sync transfers.	RW	0x0

Table 11-259. Register Call Summary for Register EDMA_TPCC_BIDX_n

EDMA Controller Functional Description

- Types of EDMA controller Transfers: [0][1]
- A-Synchronized Transfers: [2][3][4]
- AB-Synchronized Transfers: [5][6]
- PaRAM: [7][8]
- EDMA Channel PaRAM Set Entry Fields: [9][10][11][12][13][14][15]
- Parameter Set Updates: [16][17][18][19][20][21][22][23][24][25][26][27][28][29]
- Constant Addressing Mode Transfers/Alignment Issues: [30]
- Architecture Details: [31]

EDMA Transfer Examples

- Bursting Peripherals: [32]

EDMA Register Manual

- EDMA Register Summary: [33][34][35][36]

Table 11-260. EDMA_TPCC_LNK_n

Address Offset	0x0000 4014 + (0x20 * n)		
Physical Address	0x4330 4014 + (0x20 * n) 0x40D1 4014 + (0x20 * n) 0x4151 4014 + (0x20 * n) 0x01D1 4014 + (0x20 * n) 0x420A 4014 + (0x20 * n)	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE_EDMA_TPCC
Description	Link and Reload parameters		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BCNTRLD																LINK															

Bits	Field Name	Description	Type	Reset
31:16	BCNTRLD	BCNT Reload: BCNTRLD is a 16-bit unsigned value used to reload the BCNT field once the last array in the 2nd dimension is transferred. This field is only used for A-Sync'ed transfers. In this case, the CC decrements the BCNT value by one on each TR submission. When BCNT (conceptually) reaches zero, then the CC decrements CCNT and uses the BCNTRLD value to reinitialize the BCNT value. For AB-synchronized transfers, the CC submits the BCNT in the TR and therefore the TC is responsible to keep track of BCNT, not thus BCNTRLD is a don't care field.	RW	0x0

Bits	Field Name	Description	Type	Reset
15:0	LINK	Link Address: The CC provides a mechanism to reload the current PaRAM Entry upon its natural termination (i.e., after count fields are decremented to '0') with a new PaRAM Entry. This is called 'linking'. The 16-bit parameter LINK specifies the byte address offset in the PaRAM from which the CC loads/reloads the next PaRAM entry in the link. The CC should disregard the value in the upper 2 bits of the LINK field as well as the lower 5-bits of the LINK field. The upper two bits are ignored such that the user can program either the 'literal' byte address of the LINK parameter or the 'PaRAM base-relative' address of the link field. Therefore, if the user uses the literal address with a range from 0x4000 to 0x7FFF, it will be treated as a PaRAM-base-relative value of 0x0000 to 0x3FFF. The lower-5 bits are ignored and treated as 'b00000, thereby guaranteeing that all Link pointers point to a 32-byte aligned PaRAM entry. In the latter case (5-lsbs), behavior is undefined for the user (i.e., don't have to test it). In the former case (2 msbs), user should be able to take advantage of this feature (i.e., do have to test it). If a Link Update is requested to a PaRAM address that is beyond the actual range of implemented PaRAM, then the Link will be treated as a Null Linkand all 0s plus 0xFFFF will be written to the current entry location. A LINK value of 0xFFFF is referred to as a NULL link which should cause the CC to write 0x0 to all entries of the current PaRAM Entry except for the LINK field which is set to 0xFFFF. The Priv/Privid state is overwritten to 0x0 when linking. MSBs and LSBS should not be masked when comparing against the 0xFFFF value. I.e., a value of 0x3FFE is a non-NULL PaRAM link field.	RW	0x0

Table 11-261. Register Call Summary for Register EDMA_TPCC_LNK_n

EDMA Controller Functional Description
- PaRAM: [0][1]
- EDMA Channel PaRAM Set Entry Fields: [2][3][4][5][6]
- Parameter Set Updates: [7][8][9][10][11][12][13][14]
- Linking Transfers: [15][16][17][18][19][20]

EDMA Debug Checklist and Programming Tips
- EDMA Debug Checklist: [21]

EDMA Register Manual
- EDMA Register Summary: [22][23][24][25]

Table 11-262. EDMA_TPCC_CIDX_n

Address Offset	0x0000 4018 + (0x20 * n)		
Physical Address	0x4330 4018 + (0x20 * n) 0x40D1 4018 + (0x20 * n) 0x4151 4018 + (0x20 * n) 0x01D1 4018 + (0x20 * n) 0x420A 4018 + (0x20 * n)	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE_EDMA_TPCC
Description	Source and destination frame indexes		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DCIDX																SCIDX															

Bits	Field Name	Description	Type	Reset
31:16	DCIDX	Destination Frame Index: DCIDX is a 16-bit signed value (2's complement) used for destination address modification for the 3rd dimension. It is a signed value between -32768 and 32767. It provides a byte address offset from the beginning of the current array (pointed to by DST address) to the beginning of the first destination array in the next frame. It applies to both A-sync and AB-sync transfers. Note that when DCIDX is applied, the current array in an A-sync transfer is the last array in the frame, while the current array in a ABsync transfer is the first array in the frame.	RW	0x0
15:0	SCIDX	Source Frame Index: SCIDX is a 16-bit signed value (2's complement) used for source address modification for the 3rd dimension. It is a signed value between -32768 and 32767. It provides a byte address offset from the beginning of the current array (pointed to by SRC address) to the beginning of the first source array in the next frame. It applies to both A-sync and AB-sync transfers. Note that when SCIDX is applied, the current array in an A-sync transfer is the last array in the frame, while the current array in a AB-sync transfer is the first array in the frame.	RW	0x0

Table 11-263. Register Call Summary for Register EDMA_TPCC_CIDX_n

EDMA Controller Functional Description
- Types of EDMA controller Transfers: [0][1]
- A-Synchronized Transfers: [2][3]
- AB-Synchronized Transfers: [4][5]
- PaRAM: [6][7]
- EDMA Channel PaRAM Set Entry Fields: [8][9][10]
- Parameter Set Updates: [11][12][13][14]

EDMA Transfer Examples
- Bursting Peripherals: [15][16][17]

EDMA Register Manual
- EDMA Register Summary: [18][19][20][21]

Table 11-264. EDMA_TPCC_CCNT_n

Address Offset	0x0000 401C + (0x20 * n)		
Physical Address	0x4330 401C + (0x20 * n) 0x40D1 401C + (0x20 * n) 0x4151 401C + (0x20 * n) 0x01D1 401C + (0x20 * n) 0x420A 401C + (0x20 * n)	**Instance**	SYS_EDMA_TPCC DSP1_EDMA_TPCC DSP2_EDMA_TPCC DSP_EDMA_TPCC EVE_EDMA_TPCC
Description	C byte count		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																CCNT															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Reserved	RW	0x0
15:0	CCNT	CCNT: Count for 3rd Dimension: CCNT is a 16-bit unsigned value that specifies the number of frames in a block. Valid values for CCNT can be anywhere between 1 and 65535. Therefore, the maximum number of frames in a block is 65535 (64K-1 frames). CCNT of '1' means '1' frame in the block, and CCNT of '0' means '0' frames in the block. A CCNT value of '0' is considered as either a null or dummy transfer. A Dummy or Null transfer will generate a Completion code depending on the settings of the completion bit fields of the OPT field. WIMODE has no affect on CCNT operation.	RW	0x0

Table 11-265. Register Call Summary for Register EDMA_TPCC_CCNT_n

EDMA Controller Functional Description
- Types of EDMA controller Transfers: [0]
- AB-Synchronized Transfers: [1]
- PaRAM: [2]
- EDMA Channel PaRAM Set Entry Fields: [3]
- Null PaRAM Set: [4]
- Dummy PaRAM Set: [5]
- Parameter Set Updates: [6][7][8][9][10][11][12]
- Element Size: [13]
- Comparison Between DMA and QDMA Channels: [14][15]
- Completion of a DMA Transfer: [16][17][18]
- Event, Channel, and PaRAM Mapping: [19]
- Chaining EDMA Channels: [20]
- Transfer Completion Interrupts: [21]

EDMA Transfer Examples
- Bursting Peripherals: [22]

EDMA Debug Checklist and Programming Tips
- EDMA Programming Tips: [23]

EDMA Register Manual
- EDMA Register Summary: [24][25][26][27]

11.7.2.2.2 EDMA_TPTC0 and EDMA_TPTC1 Register Description

Table 11-266 through Table 11-338 describe the individual EDMA_TPTC0 and EDMA_TPTC1 module registers.

Table 11-266. EDMA_TPTCn_PID

Address Offset	0x0000 0000		
Physical Address	0x4340 0000 0x4350 0000 0x40D0 5000 0x4150 5000 0x40D0 6000 0x4150 6000 0x01D0 5000 0x01D0 6000	**Instance**	SYS_EDMA_TPTC0 SYS_EDMA_TPTC1 DSP1_EDMA_TPTC0 DSP2_EDMA_TPTC0 DSP1_EDMA_TPTC1 DSP2_EDMA_TPTC1 DSP_EDMA_TPTC0 DSP_EDMA_TPTC1
Description	Peripheral ID Register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REVISION																															

Bits	Field Name	Description	Type	Reset
31:0	REVISION	IP Revision	R	TI internal data

Table 11-267. Register Call Summary for Register EDMA_TPTCn_PID

EDMA Register Manual
- EDMA Register Summary: [0][1][2][3][4][5]

Table 11-268. EDMA_TPTCn_TCCFG

Address Offset	0x0000 0004		
Physical Address	0x4340 0004 0x4350 0004 0x40D0 5004 0x4150 5004 0x40D0 6004 0x4150 6004 0x01D0 5004 0x01D0 6004	**Instance**	SYS_EDMA_TPTC0 SYS_EDMA_TPTC1 DSP1_EDMA_TPTC0 DSP2_EDMA_TPTC0 DSP1_EDMA_TPTC1 DSP2_EDMA_TPTC1 DSP_EDMA_TPTC0 DSP_EDMA_TPTC1
Description	TC Configuration Register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						DREGDEPTH		RESERVED		BUSWIDTH		RESERVED	FIFOSIZE		

Bits	Field Name	Description	Type	Reset
31:10	RESERVED	Reads return 0's	R	0x0
9:8	DREGDEPTH	Dst Register FIFO Depth Parameterization 0x0: 1 entry 0x1: 2 entries 0x2: 4 entries	R	See Table 11-2
7:6	RESERVED	Reads return 0's	R	0x0
5:4	BUSWIDTH	Bus Width Parameterization 0x0: 32-bit 0x1: 64-bit 0x2: 128-bit	R	See Table 11-2
3	RESERVED	Reads return 0's	R	0x0
2:0	FIFOSIZE	Fifo Size Parameterization 0x0: 32 byte FIFO 0x1: 64 byte FIFO 0x2: 128 byte FIFO 0x3: 256 byte FIFO 0x4: 512 byte FIFO	R	See Table 11-2

Table 11-269. Register Call Summary for Register EDMA_TPTCn_TCCFG

EDMA Controller Functional Description
- EDMA_TPTC Configuration: [1][2][3]

EDMA Register Manual
- EDMA Register Summary: [4][5][6][7][8][9]

Table 11-270. EDMA_TPTCn_TCSTAT

Address Offset	0x0000 0100		
Physical Address	0x4340 0100	**Instance**	SYS_EDMA_TPTC0
	0x4350 0100		SYS_EDMA_TPTC1
	0x40D0 5100		DSP1_EDMA_TPTC0
	0x4150 5100		DSP2_EDMA_TPTC0
	0x40D0 6100		DSP1_EDMA_TPTC1
	0x4150 6100		DSP2_EDMA_TPTC1
	0x01D0 5100		DSP_EDMA_TPTC0
	0x01D0 6100		DSP_EDMA_TPTC1
	0x4208 6100		EVE_EDMA_TPTC0
	0x4208 7100		EVE_EDMA_TPTC1
Description	TC Status Register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																			DFSTRTPTR		RESERVED		ACTV	RESERVED	DSTACTV				RESERVED	WSACTV	SRCACTV	PROGBUSY

Bits	Field Name	Description	Type	Reset
31:13	RESERVED	Reserved	R Return 0's	0x0
12:11	DFSTRTPTR	Dst FIFO Start Pointer Represents the offset to the head entry of Dst Register FIFO, in units of *entries*. Legal values = 0x0 to 0x3	R	0x0
10:9	RESERVED	Reserved	R Return 0's	0x0
8	ACTV	Channel Active Channel Active is a logical-OR of each of the *BUSY/ACTV signals. The ACTV bit must remain high through the life of a TR. 0x0: Channel is idle 0x1: Channel is busy	R	0x1
7	RESERVED	Reserved	R Return 0's	0x0
6:4	DSTACTV	Destination Active State Specifies the number of TRs that are resident in the Dst Register FIFO at a given instant. Legal values are constrained by the DSTREGDEPTH parameter.	R	0x0
3	RESERVED	Reserved	R Return 0's	0x0
2	WSACTV	Write Status Active 0x0: Write status is not pending. Write status has been received for all previously issued write commands. 0x1: Write Status is pending. Write status has not been received for all previously issued write commands.	R	0x0
1	SRCACTV	Source Active State 0x0: Source Active set is idle. Any TR written to Prog Set will immediately transition to Source Active set as long as the Dst FIFO Set is not full (DSTFULL == 1). 0x1: Source Active set is busy either performing read transfers or waiting to perform read transfers for current Transfer Request.	R	0x0

Bits	Field Name	Description	Type	Reset
0	PROGBUSY	Program Register Set Busy 0x0: Program set idle and is available for programming. 0x1: Program set busy. User should poll for PROGBUSY equal to '0' prior to re-programming the Program Register set.	R	0x0

Table 11-271. Register Call Summary for Register EDMA_TPTCn_TCSTAT

EDMA Controller Functional Description • Debug Features: [0][1][2][3][4][5][6][7][8][9][10][11][12][13]
EDMA Register Manual • EDMA Register Summary: [14][15][16][17][18][19]

Table 11-272. EDMA_TPTCn_INTSTAT

Address Offset	0x0000 0104		
Physical Address	0x4340 0104 0x4350 0104 0x40D0 5104 0x4150 5104 0x40D0 6104 0x4150 6104 0x01D0 5104 0x01D0 6104 0x4208 6104 0x4208 7104	**Instance**	SYS_EDMA_TPTC0 SYS_EDMA_TPTC1 DSP1_EDMA_TPTC0 DSP2_EDMA_TPTC0 DSP1_EDMA_TPTC1 DSP2_EDMA_TPTC1 DSP_EDMA_TPTC0 DSP_EDMA_TPTC1 EVE_EDMA_TPTC0 EVE_EDMA_TPTC1
Description	Interrupt Status Register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														TRDONE	PROGEMPTY

Bits	Field Name	Description	Type	Reset
31:2	RESERVED	Reserved	R Return 0's	0x0
1	TRDONE	TR Done Event Status 0x0: Condition not detected. 0x1: Set when TC has completed a Transfer Request. TRDONE should be set when the write status is returned for the final write of a TR. Cleared when write '1' to INTCLR.TRDONE register bit.	R	0x0
0	PROGEMPTY	Program Set Empty Event Status 0x0: Condition not detected 0x1: Set when Program Register set transitions to empty state. Cleared when write '1' to EDMA_TPTCn_INTCLR[0] PROGEMPTY register bit.	R	0x0

Table 11-273. Register Call Summary for Register EDMA_TPTCn_INTSTAT

EDMA Register Manual

- EDMA Register Summary: [0][1][2][3][4][5]
- EDMA Register Description: [6]

Table 11-274. EDMA_TPTCn_INTEN

Address Offset	0x0000 0108		
Physical Address	0x4340 0108	**Instance**	SYS_EDMA_TPTC0
	0x4350 0108		SYS_EDMA_TPTC1
	0x40D0 5108		DSP1_EDMA_TPTC0
	0x4150 5108		DSP2_EDMA_TPTC0
	0x40D0 6108		DSP1_EDMA_TPTC1
	0x4150 6108		DSP2_EDMA_TPTC1
	0x01D0 5108		DSP_EDMA_TPTC0
	0x01D0 6108		DSP_EDMA_TPTC1
	0x4208 6108		EVE_EDMA_TPTC0
	0x4208 7108		EVE_EDMA_TPTC1
Description	Interrupt Enable Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														TRDONE	PROGEMPTY

Bits	Field Name	Description	Type	Reset
31:2	RESERVED	Reserved	R	0x0
1	TRDONE	TR Done Event Enable 0x0: TRDONE Event is disabled. 0x1: TRDONE Event is enabled, and contributes to interrupt generation	RW	0x0
0	PROGEMPTY	Program Set Empty Event Enable 0x0: PROGEMPTY Event is disabled. 0x1: PROGEMPTY Event is enabled, and contributes to interrupt generation	RW	0x0

Table 11-275. Register Call Summary for Register EDMA_TPTCn_INTEN

EDMA Register Manual

- EDMA Register Summary: [0][1][2][3][4][5]

Table 11-276. EDMA_TPTCn_INTCLR

Address Offset	0x0000 010C		
Physical Address	0x4340 010C	**Instance**	SYS_EDMA_TPTC0
	0x4350 010C		SYS_EDMA_TPTC1
	0x40D0 510C		DSP1_EDMA_TPTC0
	0x4150 510C		DSP2_EDMA_TPTC0
	0x40D0 610C		DSP1_EDMA_TPTC1
	0x4150 610C		DSP2_EDMA_TPTC1
	0x01D0 510C		DSP_EDMA_TPTC0
	0x01D0 610C		DSP_EDMA_TPTC1
	0x4208 610C		EVE_EDMA_TPTC0
	0x4208 710C		EVE_EDMA_TPTC1
Description	Interrupt Clear Register		

Table 11-276. EDMA_TPTCn_INTCLR (continued)

Type	W

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														TRDONE	PROGEMPTY

Bits	Field Name	Description	Type	Reset
31:2	RESERVED	Reserved	R	0x0
1	TRDONE	TR Done Event Clear Write 0x0: have no effect. Write 0x1: Clear	W	0x0
0	PROGEMPTY	Program Set Empty Event Clear Write 0x0: have no effect. Write 0x1: Clear	W	0x0

Table 11-277. Register Call Summary for Register EDMA_TPTCn_INTCLR

EDMA Register Manual
- EDMA Register Summary: [0][1][2][3][4][5]
- EDMA Register Description: [6]

Table 11-278. EDMA_TPTCn_INTCMD

Address Offset	0x0000 0110		
Physical Address	0x4340 0110 0x4350 0110 0x40D0 5110 0x4150 5110 0x40D0 6110 0x4150 6110 0x01D0 5110 0x01D0 6110 0x4208 6110 0x4208 7110	**Instance**	SYS_EDMA_TPTC0 SYS_EDMA_TPTC1 DSP1_EDMA_TPTC0 DSP2_EDMA_TPTC0 DSP1_EDMA_TPTC1 DSP2_EDMA_TPTC1 DSP_EDMA_TPTC0 DSP_EDMA_TPTC1 EVE_EDMA_TPTC0 EVE_EDMA_TPTC1
Description	Interrupt Command Register		
Type	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														SET	EVAL

Bits	Field Name	Description	Type	Reset
31:2	RESERVED	Reserved	R	0x0
1	SET	Set TPTC interrupt Write 0x0: have no affect. Write 0x1: SET causes TPTC interrupt to be pulsed unconditionally	W	0x0

Bits	Field Name	Description	Type	Reset
0	EVAL	Evaluate state of TPTC interrupt Write 0x0: have no affect. 0x1: causes TPTC interrupt to be pulsed if any of the EDMA_TPTCn_INTSTAT bits are set to '1'.	W	0x0

Table 11-279. Register Call Summary for Register EDMA_TPTCn_INTCMD

EDMA Register Manual
- EDMA Register Summary: [0][1][2][3][4][5]

Table 11-280. EDMA_TPTCn_ERRSTAT

Address Offset	0x0000 0120		
Physical Address	0x4340 0120	**Instance**	SYS_EDMA_TPTC0
	0x4350 0120		SYS_EDMA_TPTC1
	0x40D0 5120		DSP1_EDMA_TPTC0
	0x4150 5120		DSP2_EDMA_TPTC0
	0x40D0 6120		DSP1_EDMA_TPTC1
	0x4150 6120		DSP2_EDMA_TPTC1
	0x01D0 5120		DSP_EDMA_TPTC0
	0x01D0 6120		DSP_EDMA_TPTC1
	0x4208 6120		EVE_EDMA_TPTC0
	0x4208 7120		EVE_EDMA_TPTC1
Description	Error Status Register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												MMRAERR	TRERR	RESERVED	BUSERR

Bits	Field Name	Description	Type	Reset
31:4	RESERVED	Reserved	R Return 0's	0x0
3	MMRAERR	MR Address Error 0x0: Condition not detected 0x1: User attempted to read or write to invalid address configuration mMemory map. (Is only be set for non-emulation accesses). No additional error information is recorded.	R	0x0
2	TRERR	TR Error: TR detected that violates FIFO Mode transfer (SAM or DAM is '1') alignment rules or has ACNT or BCNT == 0. No additional error information is recorded.	R	0x0
1	RESERVED	Reserved	R Return 0's	0x0
0	BUSERR	Bus Error Event 0x0: Condition not detected. 0x1: TC has detected an error code on the write response bus or read response bus. Error information is stored in Error Details Register (EDMA_TPTCn_ERRDET).	R	0x0

Table 11-281. Register Call Summary for Register EDMA_TPTCn_ERRSTAT

EDMA Register Manual

- EDMA Register Summary: [0][1][2][3][4][5]
- EDMA Register Description: [6][7][8][9][10][11][12][13][14][15]

Table 11-282. EDMA_TPTCn_ERREN

Address Offset	0x0000 0124		
Physical Address	0x4340 0124	**Instance**	SYS_EDMA_TPTC0
	0x4350 0124		SYS_EDMA_TPTC1
	0x40D0 5124		DSP1_EDMA_TPTC0
	0x4150 5124		DSP2_EDMA_TPTC0
	0x40D0 6124		DSP1_EDMA_TPTC1
	0x4150 6124		DSP2_EDMA_TPTC1
	0x01D0 5124		DSP_EDMA_TPTC0
	0x01D0 6124		DSP_EDMA_TPTC1
	0x4208 6124		EVE_EDMA_TPTC0
	0x4208 7124		EVE_EDMA_TPTC1
Description	Error Enable Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												MMRAERR	TRERR	RESERVED	BUSERR

Bits	Field Name	Description	Type	Reset
31:4	RESERVED	Reserved	R Return 0's	0x0
3	MMRAERR	Interrupt enable for EDMA_TPTCn_ERRSTAT[3] MMRAERR 0x0: BUSERR is disabled 0x1: MMRAERR is enabled, and contributes to the TPTC error interrupt generation.	RW	0x0
2	TRERR	Interrupt enable for EDMA_TPTCn_ERRSTAT[2] TRERR 0x0: BUSERR is disabled. 0x1: TRERR is enabled, and contributes to the TPTC error interrupt generation.	RW	0x0
1	RESERVED	Reserved	R Return 0's	0x0
0	BUSERR	Interrupt enable for EDMA_TPTCn_ERRSTAT[0] BUSERR 0x0: BUSERR is disabled. 0x1: BUSERR is enabled, and contributes to the TPTC error interrupt generation.	RW	0x0

Table 11-283. Register Call Summary for Register EDMA_TPTCn_ERREN

EDMA Register Manual

- EDMA Register Summary: [0][1][2][3][4][5]

Table 11-284. EDMA_TPTCn_ERRCLR

Address Offset	0x0000 0128		
Physical Address	0x4340 0128 0x4350 0128 0x40D0 5128 0x4150 5128 0x40D0 6128 0x4150 6128 0x01D0 5128 0x01D0 6128 0x4208 6128 0x4208 7128	**Instance**	SYS_EDMA_TPTC0 SYS_EDMA_TPTC1 DSP1_EDMA_TPTC0 DSP2_EDMA_TPTC0 DSP1_EDMA_TPTC1 DSP2_EDMA_TPTC1 DSP_EDMA_TPTC0 DSP_EDMA_TPTC1 EVE_EDMA_TPTC0 EVE_EDMA_TPTC1
Description	Error Clear Register		
Type	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												MMRAERR	TRERR	RESERVED	BUSERR

Bits	Field Name	Description	Type	Reset
31:4	RESERVED	Reserved	R Return 0's	0x0
3	MMRAERR	Interrupt clear for EDMA_TPTCn_ERRSTAT[3] MMRAERR Write 0x0: have no effect Write 0x1: to clear EDMA_TPTCn_ERRSTAT[3] MMRAERR bit. Write of '1' to EDMA_TPTCn_ERRCLR[3] MMRAERR does not clear the ERRDET register.	W	0x0
2	TRERR	Interrupt clear for EDMA_TPTCn_ERRSTAT[2] TRERR Write 0x0: have no effect Write 0x1: to clear EDMA_TPTCn_ERRSTAT[2] TRERR bit. Write of '1' to EDMA_TPTCn_ERRCLR[2] TRERR does not clear the ERRDET register.	W	0x0
1	RESERVED	Reserved	R Return 0's	0x0
0	BUSERR	Interrupt clear for EDMA_TPTCn_ERRSTAT[0] BUSERR Write 0x0: have no effect Write 0x1: to clear EDMA_TPTCn_ERRSTAT[0] BUSERR bit Write of '1' to EDMA_TPTCn_ERRCLR[0] BUSERR clears the ERRDET register.	W	0x0

Table 11-285. Register Call Summary for Register EDMA_TPTCn_ERRCLR

EDMA Register Manual

- EDMA Register Summary: [0][1][2][3][4][5]
- EDMA Register Description: [6][7][8]

Table 11-286. EDMA_TPTCn_ERRDET

Address Offset	0x0000 012C		
Physical Address	0x4340 012C 0x4350 012C 0x40D0 512C 0x4150 512C 0x40D0 612C 0x4150 612C 0x01D0 512C 0x01D0 612C 0x4208 612C 0x4208 712C	**Instance**	SYS_EDMA_TPTC0 SYS_EDMA_TPTC1 DSP1_EDMA_TPTC0 DSP2_EDMA_TPTC0 DSP1_EDMA_TPTC1 DSP2_EDMA_TPTC1 DSP_EDMA_TPTC0 DSP_EDMA_TPTC1 EVE_EDMA_TPTC0 EVE_EDMA_TPTC1
Description	Error Details Register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														TCCHEN	TCINTEN	RESERVED		TCC						RESERVED				STAT			

Bits	Field Name	Description	Type	Reset
31:18	RESERVED	Reserved	R Return 0's	0x0
17	TCCHEN	Contains the EDMA_TPCC_OPT_n[17] TCCHEN value programmed by the user for the Read or Write transaction that resulted in an error.	R	0x0
16	TCINTEN	Contains the EDMA_TPCC_OPT_n[16] TCINTEN value programmed by the user for the Read or Write transaction that resulted in an error.	R	0x0
15:14	RESERVED	Reserved	R Return 0's	0x0
13:8	TCC	Transfer Complete Code: Contains the EDMA_TPCC_OPT_n[13:8] TCC value programmed by the user for the Read or Write transaction that resulted in an error.	R	0x0
7:4	RESERVED	Reserved	R Return 0's	0x0
3:0	STAT	Transaction Status: Stores the non-zero status/error code that was detected on the read status or write status bus. MS-bit effectively serves as the read vs. write error code. If read status and write status are returned on the same cycle, then the TC chooses non-zero version. If both are non-zero then write status is treated as higher priority. Encoding of errors matches the CBA spec and is summarized here: 0xF =	R	0x0

Table 11-287. Register Call Summary for Register EDMA_TPTCn_ERRDET

EDMA Register Manual

- EDMA Register Summary: [0][1][2][3][4][5]
- EDMA Register Description: [6]

Table 11-288. EDMA_TPTCn_ERRCMD

Address Offset	0x0000 0130		
Physical Address	0x4340 0130	**Instance**	SYS_EDMA_TPTC0
	0x4350 0130		SYS_EDMA_TPTC1
	0x40D0 5130		DSP1_EDMA_TPTC0
	0x4150 5130		DSP2_EDMA_TPTC0
	0x40D0 6130		DSP1_EDMA_TPTC1
	0x4150 6130		DSP2_EDMA_TPTC1
	0x01D0 5130		DSP_EDMA_TPTC0
	0x01D0 6130		DSP_EDMA_TPTC1
	0x4208 6130		EVE_EDMA_TPTC0
	0x4208 7130		EVE_EDMA_TPTC1
Description	Error Command Register		
Type	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														SET	EVAL

Bits	Field Name	Description	Type	Reset
31:2	RESERVED	Reserved	R	0x0
1	SET	Set TPTC error interrupt Write 0x0: have no affect Write 0x1: to SET causes TPTC error interrupt to be pulsed unconditionally.	W	0x0
0	EVAL	Evaluate state of TPTC error interrupt Write of '1' Write 0x0: have no affect Write 0x1: to EVAL causes TPTC error interrupt to be pulsed if any of the EDMA_TPTCn_ERRSTAT bits are set to '1'.	W	0x0

Table 11-289. Register Call Summary for Register EDMA_TPTCn_ERRCMD

EDMA Register Manual

- EDMA Register Summary: [0][1][2][3][4][5]

Table 11-290. EDMA_TPTCn_RDRATE

Address Offset	0x0000 0140		
Physical Address	0x4340 0140	**Instance**	SYS_EDMA_TPTC0
	0x4350 0140		SYS_EDMA_TPTC1
	0x40D0 5140		DSP1_EDMA_TPTC0
	0x4150 5140		DSP2_EDMA_TPTC0
	0x40D0 6140		DSP1_EDMA_TPTC1
	0x4150 6140		DSP2_EDMA_TPTC1
	0x01D0 5140		DSP_EDMA_TPTC0
	0x01D0 6140		DSP_EDMA_TPTC1
	0x4208 6140		EVE_EDMA_TPTC0
	0x4208 7140		EVE_EDMA_TPTC1
Description	Read Rate Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														RDRATE	

Bits	Field Name	Description	Type	Reset
31:3	RESERVED	Reserved	R	0x0
2:0	RDRATE	Read Rate Control: Controls the number of cycles between read commands. This is a global setting that applies to all TRs for this TC.	RW	0x0

Table 11-291. Register Call Summary for Register EDMA_TPTCn_RDRATE

EDMA Controller Functional Description
- Architecture Details: [0][1][2]

EDMA Register Manual
- EDMA Register Summary: [3][4][5][6][7][8]

Table 11-292. EDMA_TPTCn_POPT

Address Offset	0x0000 0200		
Physical Address	0x4340 0200 0x4350 0200 0x40D0 5200 0x4150 5200 0x40D0 6200 0x4150 6200 0x01D0 5200 0x01D0 6200 0x4208 6200 0x4208 7200	**Instance**	SYS_EDMA_TPTC0 SYS_EDMA_TPTC1 DSP1_EDMA_TPTC0 DSP2_EDMA_TPTC0 DSP1_EDMA_TPTC1 DSP2_EDMA_TPTC1 DSP_EDMA_TPTC0 DSP_EDMA_TPTC1 EVE_EDMA_TPTC0 EVE_EDMA_TPTC1
Description	Program Set Options		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								TCCHEN	RESERVED	TCINTEN	RESERVED			TCC		RESERVED	FWID		RESERVED	PRI		RESERVED		DAM	SAM						

Bits	Field Name	Description	Type	Reset
31:23	RESERVED	Reserved	R	0x0
22	TCCHEN	Transfer complete chaining enable 0x0: Transfer complete chaining is disabled. 0x1: Transfer complete chaining is enabled.	RW	0x0
21	RESERVED	Reserved	R	0x0
20	TCINTEN	Transfer complete interrupt enable 0x0: Transfer complete interrupt is disabled. 0x1: Transfer complete interrupt is enabled.	RW	0x0
19:18	RESERVED	Reserved	R	0x0
17:12	TCC	Transfer Complete Code: The 6-bit code is used to set the relevant bit in CER or EDMA_TPCC_IPR of the TPCC module.	RW	0x0
11	RESERVED	Reserved	R	0x0
10:8	FWID	FIFO width control: Applies if either SAM or DAM is set to FIFO mode.	RW	0x0
7	RESERVED	Reserved	R	0x0
6:4	PRI	Transfer Priority: 0x0: Priority 0 - Highest priority 0x1: Priority 1 ... 0x7: Priority 7 - Lowest priority	RW	0x0

Bits	Field Name	Description	Type	Reset
3:2	RESERVED	Reserved	R	0x0
1	DAM	Destination Address Mode within an array 0x0: INCR, Destination addressing within an array increments. 0x1: FIFO, Destination addressing within an array wraps around upon reaching FIFO width.	RW	0x0
0	SAM	Source Address Mode within an array 0x0: INCR, Source addressing within an array increments. 0x1: FIFO, Source addressing within an array wraps around upon reaching FIFO width.	RW	0x0

Table 11-293. Register Call Summary for Register EDMA_TPTCn_POPT

EDMA Register Manual
- EDMA Register Summary: [0][1][2][3][4][5]

Table 11-294. EDMA_TPTCn_PSRC

Address Offset	0x0000 0204		
Physical Address	0x4340 0204 0x4350 0204 0x40D0 5204 0x4150 5204 0x40D0 6204 0x4150 6204 0x01D0 5204 0x01D0 6204 0x4208 6204 0x4208 7204	**Instance**	SYS_EDMA_TPTC0 SYS_EDMA_TPTC1 DSP1_EDMA_TPTC0 DSP2_EDMA_TPTC0 DSP1_EDMA_TPTC1 DSP2_EDMA_TPTC1 DSP_EDMA_TPTC0 DSP_EDMA_TPTC1 EVE_EDMA_TPTC0 EVE_EDMA_TPTC1
Description	Program Set Source Address		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SADDR																															

Bits	Field Name	Description	Type	Reset
31:0	SADDR	Source address for Program Register Set	RW	0x0

Table 11-295. Register Call Summary for Register EDMA_TPTCn_PSRC

EDMA Controller Functional Description
- Architecture Details: [0][1]

EDMA Register Manual
- EDMA Register Summary: [2][3][4][5][6][7]
- EDMA Register Description: [8]

Table 11-296. EDMA_TPTCn_PCNT

Address Offset	0x0000 0208		
Physical Address	0x4340 0208 0x4350 0208 0x40D0 5208 0x4150 5208 0x40D0 6208 0x4150 6208 0x01D0 5208 0x01D0 6208 0x4208 6208 0x4208 7208	**Instance**	SYS_EDMA_TPTC0 SYS_EDMA_TPTC1 DSP1_EDMA_TPTC0 DSP2_EDMA_TPTC0 DSP1_EDMA_TPTC1 DSP2_EDMA_TPTC1 DSP_EDMA_TPTC0 DSP_EDMA_TPTC1 EVE_EDMA_TPTC0 EVE_EDMA_TPTC1
Description	Program Set Count		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BCNT																ACNT															

Bits	Field Name	Description	Type	Reset
31:16	BCNT	B-Dimension count. Number of arrays to be transferred, where each array is ACNT in length.	RW	0x0
15:0	ACNT	A-Dimension count. Number of bytes to be transferred in first dimension.	RW	0x0

Table 11-297. Register Call Summary for Register EDMA_TPTCn_PCNT

EDMA Controller Functional Description

- Architecture Details: [0][1][2][3][4][5]

EDMA Register Manual

- EDMA Register Summary: [6][7][8][9][10][11]
- EDMA Register Description: [12][13][14][15][16][17]

Table 11-298. EDMA_TPTCn_PDST

Address Offset	0x0000 020C		
Physical Address	0x4340 020C 0x4350 020C 0x40D0 520C 0x4150 520C 0x40D0 620C 0x4150 620C 0x01D0 520C 0x01D0 620C 0x4208 620C 0x4208 720C	**Instance**	SYS_EDMA_TPTC0 SYS_EDMA_TPTC1 DSP1_EDMA_TPTC0 DSP2_EDMA_TPTC0 DSP1_EDMA_TPTC1 DSP2_EDMA_TPTC1 DSP_EDMA_TPTC0 DSP_EDMA_TPTC1 EVE_EDMA_TPTC0 EVE_EDMA_TPTC1
Description	Program Set Destination Address		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DADDR																															

Bits	Field Name	Description	Type	Reset
31:0	DADDR	Destination address for Program Register Set	RW	0x0

Table 11-299. Register Call Summary for Register EDMA_TPTCn_PDST

EDMA Register Manual

- EDMA Register Summary: [0][1][2][3][4][5]
- EDMA Register Description: [6]

Table 11-300. EDMA_TPTCn_PBIDX

Address Offset	0x0000 0210		
Physical Address	0x4340 0210	**Instance**	SYS_EDMA_TPTC0
	0x4350 0210		SYS_EDMA_TPTC1
	0x40D0 5210		DSP1_EDMA_TPTC0
	0x4150 5210		DSP2_EDMA_TPTC0
	0x40D0 6210		DSP1_EDMA_TPTC1
	0x4150 6210		DSP2_EDMA_TPTC1
	0x01D0 5210		DSP_EDMA_TPTC0
	0x01D0 6210		DSP_EDMA_TPTC1
	0x4208 6210		EVE_EDMA_TPTC0
	0x4208 7210		EVE_EDMA_TPTC1
Description	Program Set B-Dim Idx		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DBIDX																SBIDX															

Bits	Field Name	Description	Type	Reset
31:16	DBIDX	Dest B-Idx for Program Register Set: B-Idx offset between Destination arrays: Represents the offset in bytes between the starting address of each destination array (recall that there are BCNT arrays of ACNT elements). DBIDX is always used, regardless of whether DAM is Increment or FIFO mode.	RW	0x0
15:0	SBIDX	Source B-Idx for Program Register Set: B-Idx offset between Source arrays: Represents the offset in bytes between the starting address of each source array (recall that there are BCNT arrays of ACNT elements). SBIDX is always used, regardless of whether SAM is Increment or FIFO mode.	RW	0x0

Table 11-301. Register Call Summary for Register EDMA_TPTCn_PBIDX

EDMA Controller Functional Description

- Architecture Details: [0][1]

EDMA Register Manual

- EDMA Register Summary: [2][3][4][5][6][7]
- EDMA Register Description: [8][9][10][11]

Table 11-302. EDMA_TPTCn_PMPPRXY

Address Offset	0x0000 0214		
Physical Address	0x4340 0214	**Instance**	SYS_EDMA_TPTC0
	0x4350 0214		SYS_EDMA_TPTC1
	0x40D0 5214		DSP1_EDMA_TPTC0
	0x4150 5214		DSP2_EDMA_TPTC0
	0x40D0 6214		DSP1_EDMA_TPTC1
	0x4150 6214		DSP2_EDMA_TPTC1
	0x01D0 5214		DSP_EDMA_TPTC0
	0x01D0 6214		DSP_EDMA_TPTC1
	0x4208 6214		EVE_EDMA_TPTC0
	0x4208 7214		EVE_EDMA_TPTC1
Description	Program Set Memory Protect Proxy		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							PRIV	RESERVED				PRIVID			

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reserved	R Return 0's	0x0
8	PRIV	Privilege Level 0x0: User level privilege 0x1: Supervisor level privilege EDMA_TPTCn_PMPPRXY.PRIV is always updated with the value from the configuration bus privilege field on any/every write to Program Set BIDX Register (trigger register). The PRIV value for the SA Set and DF Set are copied from the value in the Program set along with the remainder of the parameter values. The privilege ID is issued on the VBusM read and write command bus such that the target endpoints can perform mMemory protection checks based on the PRIV of the external host that sets up the DMA transaction.	R	0x0
7:4	RESERVED	Reserved	R Return 0's	0x0
3:0	PRIVID	Privilege ID: EDMA_TPTCn_PMPPRXY.PRIVID is always updated with the value from configuration bus privilege ID field on any/every write to Program Set BIDX Register (trigger register). The PRIVID value for the SA Set and DF Set are copied from the value in the Program set along with the remainder of the parameter values. The privilege ID is issued on the VBusM read and write command bus such that the target endpoints can perform mMemory protection checks based on the privid of the external host that sets up the DMA transaction.	R	0x0

Table 11-303. Register Call Summary for Register EDMA_TPTCn_PMPPRXY

EDMA Register Manual

- EDMA Register Summary: [0][1][2][3][4][5]
- EDMA Register Description: [6][7]

Table 11-304. EDMA_TPTCn_SAOPT

Address Offset	0x0000 0240		
Physical Address	0x4340 0240	**Instance**	SYS_EDMA_TPTC0
	0x4350 0240		SYS_EDMA_TPTC1
	0x40D0 5240		DSP1_EDMA_TPTC0
	0x4150 5240		DSP2_EDMA_TPTC0
	0x40D0 6240		DSP1_EDMA_TPTC1
	0x4150 6240		DSP2_EDMA_TPTC1
	0x01D0 5240		DSP_EDMA_TPTC0
	0x01D0 6240		DSP_EDMA_TPTC1
	0x4208 6240		EVE_EDMA_TPTC0
	0x4208 7240		EVE_EDMA_TPTC1
Description	Source Actve Set Options		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								TCCHEN	RESERVED		TCINTEN	RESERVED				TCC				RESERVED		FWID		RESERVED		PRI			RESERVED		DAM	SAM

Bits	Field Name	Description	Type	Reset
31:23	RESERVED	Reserved	R Return 0's	0x0
22	TCCHEN	Transfer complete chaining enable 0x0: Transfer complete chaining is disabled. 0x1: Transfer complete chaining is enabled.	R	0x0
21	RESERVED	Reserved	R Return 0's	0x0
20	TCINTEN	Transfer complete interrupt enable 0x0: Transfer complete interrupt is disabled. 0x1: Transfer complete interrupt is enabled.	R	0x0
19:18	RESERVED	Reserved	R Return 0's	0x0
17:12	TCC	Transfer Complete Code The 6-bit code is used to set the relevant bit in EDMA_TPCC_CER or EDMA_TPCC_IPR of the TPCC module.	R	0x0
11	RESERVED	Reserved	R Return 0's	0x0
10:8	FWID	FIFO width control Applies if either SAM or DAM is set to FIFO mode.	R	0x0
7	RESERVED	Reserved	R Return 0's	0x0
6:4	PRI	Transfer Priority 0x0: Priority 0 - Highest priority 0x1: Priority 1 ... 0x7: Priority 7 - Lowest priority	R	0x0
3:2	RESERVED	Reserved	R Return 0's	0x0

Bits	Field Name	Description	Type	Reset
1	DAM	Destination Address Mode within an array 0x0: INCR, Destination addressing within an array increments. 0x1: FIFO, Destination addressing within an array wraps around upon reaching FIFO width.	R	0x0
0	SAM	Source Address Mode within an array 0x0: INCR, Source addressing within an array increments. 0X1: FIFO, Source addressing within an array wraps around upon reaching FIFO width.	R	0x0

Table 11-305. Register Call Summary for Register EDMA_TPTCn_SAOPT

EDMA Register Manual
- EDMA Register Summary: [0][1][2][3][4][5]

Table 11-306. EDMA_TPTCn_SASRC

Address Offset	0x0000 0244		
Physical Address	0x4340 0244 0x4350 0244 0x40D0 5244 0x4150 5244 0x40D0 6244 0x4150 6244 0x01D0 5244 0x01D0 6244 0x4208 6244 0x4208 7244	**Instance**	SYS_EDMA_TPTC0 SYS_EDMA_TPTC1 DSP1_EDMA_TPTC0 DSP2_EDMA_TPTC0 DSP1_EDMA_TPTC1 DSP2_EDMA_TPTC1 DSP_EDMA_TPTC0 DSP_EDMA_TPTC1 EVE_EDMA_TPTC0 EVE_EDMA_TPTC1
Description	Source Active Set Source Address		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SADDR																															

Bits	Field Name	Description	Type	Reset
31:0	SADDR	Source address for Source Active Register Set: Initial value is copied from EDMA_TPTCn_PSRC.SADDR. TC updates value according to source addressing mode (EDMA_TPCC_OPT_n.SAM) and/or source index value (BIDX.SBIDX) after each read command is issued. When a TR is complete, the final value should be the address of the last read command issued.	R	0x0

Table 11-307. Register Call Summary for Register EDMA_TPTCn_SASRC

EDMA Register Manual
- EDMA Register Summary: [0][1][2][3][4][5]

Table 11-308. EDMA_TPTCn_SACNT

Address Offset	0x0000 0248		
Physical Address	0x4340 0248 0x4350 0248 0x40D0 5248 0x4150 5248 0x40D0 6248 0x4150 6248 0x01D0 5248 0x01D0 6248 0x4208 6248 0x4208 7248	**Instance**	SYS_EDMA_TPTC0 SYS_EDMA_TPTC1 DSP1_EDMA_TPTC0 DSP2_EDMA_TPTC0 DSP1_EDMA_TPTC1 DSP2_EDMA_TPTC1 DSP_EDMA_TPTC0 DSP_EDMA_TPTC1 EVE_EDMA_TPTC0 EVE_EDMA_TPTC1
Description	Source Active Set Count		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BCNT																ACNT															

Bits	Field Name	Description	Type	Reset
31:16	BCNT	B-Dimension count: Number of arrays to be transferred, where each array is ACNT in length. Count Remaining for Source Active Register Set. Represents the amount of data remaining to be read. Initial value is copied from EDMA_TPTCn_PCNT. TC decrements ACNT and BCNT as necessary after each read command is issued. Final value should be 0 when TR is complete.	R	0x0
15:0	ACNT	A-Dimension count: Number of bytes to be transferred in first dimension. Count Remaining for Source Active Register Set. Represents the amount of data remaining to be read. Initial value is copied from EDMA_TPTCn_PCNT. TC decrements ACNT and BCNT as necessary after each read command is issued. Final value should be 0 when TR is complete.	R	0x0

Table 11-309. Register Call Summary for Register EDMA_TPTCn_SACNT

EDMA Register Manual

- EDMA Register Summary: [0][1][2][3][4][5]

Table 11-310. EDMA_TPTCn_SADST

Address Offset	0x0000 024C		
Physical Address	0x4340 024C 0x4350 024C 0x40D0 524C 0x4150 524C 0x40D0 624C 0x4150 624C 0x01D0 524C 0x01D0 624C	**Instance**	SYS_EDMA_TPTC0 SYS_EDMA_TPTC1 DSP1_EDMA_TPTC0 DSP2_EDMA_TPTC0 DSP1_EDMA_TPTC1 DSP2_EDMA_TPTC1 DSP_EDMA_TPTC0 DSP_EDMA_TPTC1
Description	Source Active Destination Address Register Reserved, return 0x0 w/o AERROR		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DADDR																															

Bits	Field Name	Description	Type	Reset
31:0	DADDR	Destination address is not applicable for Source Active Register Set. Reads return 0x0	R	0x0

Table 11-311. Register Call Summary for Register EDMA_TPTCn_SADST

EDMA Controller Functional Description
- Architecture Details: [0][1]

EDMA Register Manual
- EDMA Register Summary: [2][3][4][5][6][7]

Table 11-312. EDMA_TPTCn_SABIDX

Address Offset	0x0000 0250		
Physical Address	0x4340 0250 0x4350 0250 0x40D0 5250 0x4150 5250 0x40D0 6250 0x4150 6250 0x01D0 5250 0x01D0 6250 0x4208 6250 0x4208 7250	**Instance**	SYS_EDMA_TPTC0 SYS_EDMA_TPTC1 DSP1_EDMA_TPTC0 DSP2_EDMA_TPTC0 DSP1_EDMA_TPTC1 DSP2_EDMA_TPTC1 DSP_EDMA_TPTC0 DSP_EDMA_TPTC1 EVE_EDMA_TPTC0 EVE_EDMA_TPTC1
Description	Source Active Set B-Dim Idx		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DBIDX																SBIDX															

Bits	Field Name	Description	Type	Reset
31:16	DBIDX	Destination B-Idx for Source Active Register Set. Value copied from EDMA_TPTCn_PBIDX: B-Idx offset between Destination arrays: Represents the offset in bytes between the starting address of each destination array (recall that there are BCNT arrays of ACNT elements). DBIDX is always used, regardless of whether DAM is Increment or FIFO mode.	R	0x0
15:0	SBIDX	Source B-Idx for Source Active Register Set. Value copied from EDMA_TPTCn_PBIDX: B-Idx offset between Source arrays: Represents the offset in bytes between the starting address of each source array (recall that there are BCNT arrays of ACNT elements). SBIDX is always used, regardless of whether SAM is Increment or FIFO mode.	R	0x0

Table 11-313. Register Call Summary for Register EDMA_TPTCn_SABIDX

EDMA Register Manual
- EDMA Register Summary: [0][1][2][3][4][5]

Table 11-314. EDMA_TPTCn_SAMPPRXY

Address Offset	0x0000 0254		
Physical Address	0x4340 0254	**Instance**	SYS_EDMA_TPTC0
	0x4350 0254		SYS_EDMA_TPTC1
	0x40D0 5254		DSP1_EDMA_TPTC0
	0x4150 5254		DSP2_EDMA_TPTC0
	0x40D0 6254		DSP1_EDMA_TPTC1
	0x4150 6254		DSP2_EDMA_TPTC1
	0x01D0 5254		DSP_EDMA_TPTC0
	0x01D0 6254		DSP_EDMA_TPTC1
	0x4208 6254		EVE_EDMA_TPTC0
	0x4208 7254		EVE_EDMA_TPTC1
Description	Source Active Set Mem Protect Proxy		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							PRIV	RESERVED				PRIVID			

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reserved	R Return 0's	0x0
8	PRIV	Privilege Level 0x0: User level privilege 0x1: Supervisor level privilege SAMPPRXY.PRIV is always updated with the value from the configuration bus privilege field on any/every write to Program Set BIDX Register (trigger register). The PRIV value for the SA Set and DF Set are copied from the value in the Program set along with the remainder of the parameter values. The privilege ID is issued on the VBusM read and write command bus such that the target endpoints can perform mMemory protection checks based on the PRIV of the external host that sets up the DMA transaction.	R	0x0
7:4	RESERVED	Reserved	R Return 0's	0x0
3:0	PRIVID	Privilege ID SAMPPRXY.PRIVID is always updated with the value from configuration bus privilege ID field on any/every write to Program Set BIDX Register (trigger register). The PRIVID value for the SA Set and DF Set are copied from the value in the Program set along with the remainder of the parameter values. The privilege ID is issued on the VBusM read and write command bus such that the target endpoints can perform mMemory protection checks based on the privid of the external host that sets up the DMA transaction.	R	0x0

Table 11-315. Register Call Summary for Register EDMA_TPTCn_SAMPPRXY

EDMA Register Manual

- EDMA Register Summary: [0][1][2][3][4][5]

Table 11-316. EDMA_TPTCn_SACNTRLD

Address Offset	0x0000 0258		
Physical Address	0x4340 0258 0x4350 0258 0x40D0 5258 0x4150 5258 0x40D0 6258 0x4150 6258 0x01D0 5258 0x01D0 6258 0x4208 6258 0x4208 7258	**Instance**	SYS_EDMA_TPTC0 SYS_EDMA_TPTC1 DSP1_EDMA_TPTC0 DSP2_EDMA_TPTC0 DSP1_EDMA_TPTC1 DSP2_EDMA_TPTC1 DSP_EDMA_TPTC0 DSP_EDMA_TPTC1 EVE_EDMA_TPTC0 EVE_EDMA_TPTC1
Description	Source Active Set Count Reload		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																ACNTRLD															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Reserved	R Return 0's	0x0
15:0	ACNTRLD	A-Cnt Reload value for Source Active Register set. Value copied from EDMA_TPTCn_PCNT[15:0] ACNT: Represents the originally programmed value of ACNT. The Reload value is used to reinitialize ACNT after each array is serviced (i.e., ACNT decrements to 0). by the Src offset in bytes between the starting address of each source array (recall that there are BCNT arrays of ACNT bytes)	R	0x0

Table 11-317. Register Call Summary for Register EDMA_TPTCn_SACNTRLD

EDMA Register Manual

- EDMA Register Summary: [0][1][2][3][4][5]

Table 11-318. EDMA_TPTCn_SASRCBREF

Address Offset	0x0000 025C		
Physical Address	0x4340 025C 0x4350 025C 0x40D0 525C 0x4150 525C 0x40D0 625C 0x4150 625C 0x01D0 525C 0x01D0 625C 0x4208 625C 0x4208 725C	**Instance**	SYS_EDMA_TPTC0 SYS_EDMA_TPTC1 DSP1_EDMA_TPTC0 DSP2_EDMA_TPTC0 DSP1_EDMA_TPTC1 DSP2_EDMA_TPTC1 DSP_EDMA_TPTC0 DSP_EDMA_TPTC1 EVE_EDMA_TPTC0 EVE_EDMA_TPTC1
Description	Source Active Set Source Address A-Reference		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SADDRBREF																															

Bits	Field Name	Description	Type	Reset
31:0	SADDRBREF	Source address reference for Source Active Register Set: Represents the starting address for the array currently being read. The next array's starting address is calculated as the 'reference address' plus the 'source b-idx' value.	R	0x0

Table 11-319. Register Call Summary for Register EDMA_TPTCn_SASRCBREF

EDMA Register Manual
- EDMA Register Summary: [0][1][2][3][4][5]

Table 11-320. EDMA_TPTCn_SADSTBREF

Address Offset	0x0000 0260		
Physical Address	0x4340 0260	**Instance**	SYS_EDMA_TPTC0
	0x4350 0260		SYS_EDMA_TPTC1
	0x40D0 5260		DSP1_EDMA_TPTC0
	0x4150 5260		DSP2_EDMA_TPTC0
	0x40D0 6260		DSP1_EDMA_TPTC1
	0x4150 6260		DSP2_EDMA_TPTC1
	0x01D0 5260		DSP_EDMA_TPTC0
	0x01D0 6260		DSP_EDMA_TPTC1
	0x4208 6260		EVE_EDMA_TPTC0
	0x4208 7260		EVE_EDMA_TPTC1
Description	Source Active Destination Address B-Reference Register Reserved, return 0x0 w/o AERROR		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DADDRBREF																															

Bits	Field Name	Description	Type	Reset
31:0	DADDRBREF	Destination address reference is not applicable for Src Active Register Set. Reads return 0x0.	R	0x0

Table 11-321. Register Call Summary for Register EDMA_TPTCn_SADSTBREF

EDMA Register Manual
- EDMA Register Summary: [0][1][2][3][4][5]

Table 11-322. EDMA_TPTCn_DFCNTRLD

Address Offset	0x0000 0280		
Physical Address	0x4340 0280	**Instance**	SYS_EDMA_TPTC0
	0x4350 0280		SYS_EDMA_TPTC1
	0x40D0 5280		DSP1_EDMA_TPTC0
	0x4150 5280		DSP2_EDMA_TPTC0
	0x40D0 6280		DSP1_EDMA_TPTC1
	0x4150 6280		DSP2_EDMA_TPTC1
	0x01D0 5280		DSP_EDMA_TPTC0
	0x01D0 6280		DSP_EDMA_TPTC1
	0x4208 6280		EVE_EDMA_TPTC0
	0x4208 7280		EVE_EDMA_TPTC1
Description	Destination FIFO Set Count Reload		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																ACNTRLD															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Reserved	R Return 0's	0x0
15:0	ACNTRLD	A-Cnt Reload value for Destination FIFO Register set. Value copied from EDMA_TPTCn_PCNT[15:0] ACNT: Represents the originally programmed value of ACNT. The Reload value is used to reinitialize ACNT after each array is serviced (i.e., ACNT decrements to 0). by the Src offset in bytes between the starting address of each source array (recall that there are BCNT arrays of ACNT bytes)	R	0x0

Table 11-323. Register Call Summary for Register EDMA_TPTCn_DFCNTRLD

EDMA Register Manual

- EDMA Register Summary: [0][1][2][3][4][5]

Table 11-324. EDMA_TPTCn_DFSRCBREF

Address Offset	0x0000 0284		
Physical Address	0x4340 0284 0x4350 0284 0x40D0 5284 0x4150 5284 0x40D0 6284 0x4150 6284 0x01D0 5284 0x01D0 6284 0x4208 6284 0x4208 7284	**Instance**	SYS_EDMA_TPTC0 SYS_EDMA_TPTC1 DSP1_EDMA_TPTC0 DSP2_EDMA_TPTC0 DSP1_EDMA_TPTC1 DSP2_EDMA_TPTC1 DSP_EDMA_TPTC0 DSP_EDMA_TPTC1 EVE_EDMA_TPTC0 EVE_EDMA_TPTC1
Description	Destination FIFO Set Destination Address B Reference Reserved, return 0x0 w/o AERROR		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SADDRBREF																															

Bits	Field Name	Description	Type	Reset
31:0	SADDRBREF	Source address reference is not applicable for Dst FIFO Register Set. Reads return 0x0.	R	0x0

Table 11-325. Register Call Summary for Register EDMA_TPTCn_DFSRCBREF

EDMA Register Manual

- EDMA Register Summary: [0][1][2][3][4][5]

Table 11-326. EDMA_TPTCn_DFDSTBREF

Address Offset	0x0000 0288		
Physical Address	0x4340 0288 0x4350 0288 0x40D0 5288 0x4150 5288 0x40D0 6288 0x4150 6288 0x01D0 5288 0x01D0 6288	**Instance**	SYS_EDMA_TPTC0 SYS_EDMA_TPTC1 DSP1_EDMA_TPTC0 DSP2_EDMA_TPTC0 DSP1_EDMA_TPTC1 DSP2_EDMA_TPTC1 DSP_EDMA_TPTC0 DSP_EDMA_TPTC1
Description	Destination FIFO Set Destination Address A-Reference		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DADDRBREF																															

Bits	Field Name	Description	Type	Reset
31:0	DADDRBREF	Destination address reference for Dst FIFO Register Set: Represents the starting address for the array currently being written. The next array's starting address is calculated as the 'reference address' plus the 'dest bidx' value.	R	0x0

Table 11-327. Register Call Summary for Register EDMA_TPTCn_DFDSTBREF

EDMA Register Manual

- EDMA Register Summary: [0][1][2][3][4][5]

Table 11-328. EDMA_TPTCn_DFOPTi

Address Offset	0x0000 0300 + (0x40 * i)		
Physical Address	0x4340 0300 + (0x40 * i) 0x4350 0300 + (0x40 * i) 0x40D0 5300 + (0x40 * i) 0x4150 5300 + (0x40 * i) 0x40D0 6300 + (0x40 * i) 0x4150 6300 + (0x40 * i) 0x01D0 5300 + (0x40 * i) 0x01D0 6300 + (0x40 * i) 0x4208 6300 + (0x40 * i) 0x4208 7300 + (0x40 * i)	**Instance**	SYS_EDMA_TPTC0 SYS_EDMA_TPTC1 DSP1_EDMA_TPTC0 DSP2_EDMA_TPTC0 DSP1_EDMA_TPTC1 DSP2_EDMA_TPTC1 DSP_EDMA_TPTC0 DSP_EDMA_TPTC1 EVE_EDMA_TPTC0 EVE_EDMA_TPTC1
Description	Destination FIFO Set Options		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								TCCHEN	RESERVED	TCINTEN	RESERVED	TCC				RESERVED	FWID			RESERVED	PRI			RESERVED		DAM	SAM				

Bits	Field Name	Description	Type	Reset
31:23	RESERVED	Reserved	R Return 0's	0x0
22	TCCHEN	Transfer complete chaining enable 0x0: Transfer complete chaining is disabled. 0x1: Transfer complete chaining is enabled.	R	0x0

Bits	Field Name	Description	Type	Reset
21	RESERVED	Reserved	R Return 0's	0x0
20	TCINTEN	Transfer complete interrupt enable 0x0: Transfer complete interrupt is disabled. 0x1: Transfer complete interrupt is enabled.	R	0x0
19:18	RESERVED	Reserved	R Return 0's	0x0
17:12	TCC	Transfer Complete Code The 6-bit code is used to set the relevant bit in CER or EDMA_TPCC_IPR of the TPCC module.	R	0x0
11	RESERVED	Reserved	R Return 0's	0x0
10:8	FWID	FIFO width control Applies if either SAM or DAM is set to FIFO mode.	R	0x0
7	RESERVED	Reserved	R Return 0's	0x0
6:4	PRI	Transfer Priority 0x0: Priority 0 - Highest priority 0x1: Priority 1 ... 0x7: Priority 7 - Lowest priority	R	0x0
3:2	RESERVED	Reserved	R Return 0's	0x0
1	DAM	Destination Address Mode within an array 0x0: INCR, Dst addressing within an array increments. 0x1: FIFO, Dst addressing within an array wraps around upon reaching FIFO width.	R	0x0
0	SAM	Source Address Mode within an array 0x0: INCR, Source addressing within an array increments. 0x1: FIFO, Source addressing within an array wraps around upon reaching FIFO width.	R	0x0

Table 11-329. Register Call Summary for Register EDMA_TPTCn_DFOPTi

EDMA Register Manual
- EDMA Register Summary: [0][1][2][3][4][5]

Table 11-330. EDMA_TPTCn_DFSRCi

Address Offset	0x0000 0304 + (0x40 * i)		
Physical Address	0x4340 0304 + (0x40 * i) 0x4350 0304 + (0x40 * i) 0x40D0 5304 + (0x40 * i) 0x4150 5304 + (0x40 * i) 0x40D0 6304 + (0x40 * i) 0x4150 6304 + (0x40 * i) 0x01D0 5304 + (0x40 * i) 0x01D0 6304 + (0x40 * i) 0x4208 6304 + (0x40 * i) 0x4208 7304 + (0x40 * i)	**Instance**	SYS_EDMA_TPTC0 SYS_EDMA_TPTC1 DSP1_EDMA_TPTC0 DSP2_EDMA_TPTC0 DSP1_EDMA_TPTC1 DSP2_EDMA_TPTC1 DSP_EDMA_TPTC0 DSP_EDMA_TPTC1 EVE_EDMA_TPTC0 EVE_EDMA_TPTC1
Description	Destination FIFO source address register Reserved, return 0x0 w/o AERROR		

Table 11-330. EDMA_TPTCn_DFSRCi (continued)

Type	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SADDR																															

Bits	Field Name	Description	Type	Reset
31:0	SADDR	Source address is not applicable for Dst FIFO Register Set: Reads return 0x0.	R	0x0

Table 11-331. Register Call Summary for Register EDMA_TPTCn_DFSRCi

EDMA Register Manual
- EDMA Register Summary: [0][1][2][3][4][5]

Table 11-332. EDMA_TPTCn_DFCNTi

Address Offset	0x0000 0308 + (0x40 * i)		
Physical Address	0x4340 0308 + (0x40 * i)	Instance	SYS_EDMA_TPTC0
	0x4350 0308 + (0x40 * i)		SYS_EDMA_TPTC1
	0x40D0 5308 + (0x40 * i)		DSP1_EDMA_TPTC0
	0x4150 5308 + (0x40 * i)		DSP2_EDMA_TPTC0
	0x40D0 6308 + (0x40 * i)		DSP1_EDMA_TPTC1
	0x4150 6308 + (0x40 * i)		DSP2_EDMA_TPTC1
	0x01D0 5308 + (0x40 * i)		DSP_EDMA_TPTC0
	0x01D0 6308 + (0x40 * i)		DSP_EDMA_TPTC1
	0x4208 6308 + (0x40 * i)		EVE_EDMA_TPTC0
	0x4208 7308 + (0x40 * i)		EVE_EDMA_TPTC1
Description	Destination FIFO count register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BCNT																ACNT															

Bits	Field Name	Description	Type	Reset
31:16	BCNT	B-Count Remaining for Dst Register Set: Number of arrays to be transferred, where each array is ACNT in length. Represents the amount of data remaining to be written. Initial value is copied from EDMA_TPTCn_PCNT. TC decrements ACNT and BCNT as necessary after each write dataphase is issued. Final value should be 0 when TR is complete.	R	0x0
15:0	ACNT	A-Count Remaining for Dst Register Set: Number of bytes to be transferred in first dimension. Represents the amount of data remaining to be written. Initial value is copied from EDMA_TPTCn_PCNT. TC decrements ACNT and BCNT as necessary after each write dataphase is issued. Final value should be 0 when TR is complete.	R	0x0

Table 11-333. Register Call Summary for Register EDMA_TPTCn_DFCNTi

EDMA Register Manual
- EDMA Register Summary: [0][1][2][3][4][5]

Table 11-334. EDMA_TPTCn_DFDSTi

Address Offset	0x0000 030C + (0x40 * i)		
Physical Address	0x4340 030C + (0x40 * i) 0x4350 030C + (0x40 * i) 0x40D0 530C + (0x40 * i) 0x4150 530C + (0x40 * i) 0x40D0 630C + (0x40 * i) 0x4150 630C + (0x40 * i) 0x01D0 530C + (0x40 * i) 0x01D0 630C + (0x40 * i) 0x4208 630C + (0x40 * i) 0x4208 730C + (0x40 * i)	**Instance**	SYS_EDMA_TPTC0 SYS_EDMA_TPTC1 DSP1_EDMA_TPTC0 DSP2_EDMA_TPTC0 DSP1_EDMA_TPTC1 DSP2_EDMA_TPTC1 DSP_EDMA_TPTC0 DSP_EDMA_TPTC1 EVE_EDMA_TPTC0 EVE_EDMA_TPTC1
Description	The destination FIFO destination address register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DADDR																															

Bits	Field Name	Description	Type	Reset
31:0	DADDR	Destination address for Dst FIFO Register Set: Initial value is copied from EDMA_TPTCn_PDST[31:0] DADDR. TC updates value according to destination addressing mode (EDMA_TPCC_OPT_n. SAM) and/or dest index value (BIDX. DBIDX) after each write command is issued. When a TR is complete, the final value should be the address of the last write command issued.	R	0x0

Table 11-335. Register Call Summary for Register EDMA_TPTCn_DFDSTi

EDMA Register Manual

- EDMA Register Summary: [0][1][2][3][4][5]

Table 11-336. EDMA_TPTCn_DFBIDXi

Address Offset	0x0000 0310 + (0x40 * i)		
Physical Address	0x4340 0310 + (0x40 * i) 0x4350 0310 + (0x40 * i) 0x40D0 5310 + (0x40 * i) 0x4150 5310 + (0x40 * i) 0x40D0 6310 + (0x40 * i) 0x4150 6310 + (0x40 * i) 0x01D0 5310 + (0x40 * i) 0x01D0 6310 + (0x40 * i) 0x4208 6310 + (0x40 * i) 0x4208 7310 + (0x40 * i)	**Instance**	SYS_EDMA_TPTC0 SYS_EDMA_TPTC1 DSP1_EDMA_TPTC0 DSP2_EDMA_TPTC0 DSP1_EDMA_TPTC1 DSP2_EDMA_TPTC1 DSP_EDMA_TPTC0 DSP_EDMA_TPTC1 EVE_EDMA_TPTC0 EVE_EDMA_TPTC1
Description	The destination FIFO B-index register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DBIDX																SBIDX															

Bits	Field Name	Description	Type	Reset
31:16	DBIDX	Dest B-Idx for Dest FIFO Register Set. Value copied from EDMA_TPTCn_PBIDX: B-Idx offset between Destination arrays: Represents the offset in bytes between the starting address of each destination array (recall that there are BCNT arrays of ACNT elements). DBIDX is always used, regardless of whether DAM is Increment or FIFO mode.	R	0x0
15:0	SBIDX	Dest B-Idx for Dest FIFO Register Set. Value copied from EDMA_TPTCn_PBIDX: B-Idx offset between Source arrays: Represents the offset in bytes between the starting address of each source array (recall that there are BCNT arrays of ACNT elements). SBIDX is always used, regardless of whether SAM is Increment or FIFO mode.	R	0x0

Table 11-337. Register Call Summary for Register EDMA_TPTCn_DFBIDXi

EDMA Register Manual
- EDMA Register Summary: [0][1][2][3][4][5]

Table 11-338. EDMA_TPTCn_DFMPPRXYi

Address Offset	0x0000 0314 + (0x40 * i)		
Physical Address	0x4340 0314 + (0x40 * i)	**Instance**	SYS_EDMA_TPTC0
	0x4350 0314 + (0x40 * i)		SYS_EDMA_TPTC1
	0x40D0 5314 + (0x40 * i)		DSP1_EDMA_TPTC0
	0x4150 5314 + (0x40 * i)		DSP2_EDMA_TPTC0
	0x40D0 6314 + (0x40 * i)		DSP1_EDMA_TPTC1
	0x4150 6314 + (0x40 * i)		DSP2_EDMA_TPTC1
	0x01D0 5314 + (0x40 * i)		DSP_EDMA_TPTC0
	0x01D0 6314 + (0x40 * i)		DSP_EDMA_TPTC1
	0x4208 6314 + (0x40 * i)		EVE_EDMA_TPTC0
	0x4208 7314 + (0x40 * i)		EVE_EDMA_TPTC1
Description	The destination FIFO memory protection proxy register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							PRIV	RESERVED				PRIVID			

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reserved	R Return 0's	0x0
8	PRIV	Privilege Level 0x0: User level privilege 0x1: Supervisor level privilege DFMPPRXY0.PRIV is always updated with the value from the configuration bus privilege field on any/every write to Program Set BIDX Register (trigger register). The PRIV value for the SA Set and DF Set are copied from the value in the Program set along with the remainder of the parameter values. The privilege ID is issued on the VBusM read and write command bus such that the target endpoints can perform mMemory protection checks based on the PRIV of the external host that sets up the DMA transaction.	R	0x0
7:4	RESERVED	Reserved	R Return 0's	0x0

Bits	Field Name	Description	Type	Reset
3:0	PRIVID	Privilege ID: DFMPPRXY0.PRIVID is always updated with the value from configuration bus privilege ID field on any/every write to Program Set BIDX Register (trigger register). The PRIVID value for the SA Set and DF Set are copied from the value in the Program set along with the remainder of the parameter values. The privilege ID is issued on the VBusM read and write command bus such that the target endpoints can perform mMemory protection checks based on the privid of the external host that sets up the DMA transaction.	R	0x0

Table 11-339. Register Call Summary for Register EDMA_TPTCn_DFMPPRXYi

EDMA Register Manual

- EDMA Register Summary: [0][1][2][3][4][5]

Interrupt Controllers

This chapter describes the interrupt controllers (INTCs) in the device.

Topic **Page**

12.1 Interrupt Controllers Overview

The device has a large number of interrupts to service the needs of its many peripherals and subsystems. The DSP (x2), IPU, and EVE subsystems are capable of servicing these interrupts via their integrated interrupt controllers. In addition, each processor's interrupt controller is preceded by an Interrupt Controller Crossbar (IRQ_CROSSBAR) that provides flexibility in mapping the device interrupts to processor interrupt inputs. For more information about IRQ crossbar, see Chapter 13, *Control Module*.

C66x DSP Subsystem Interrupt Controller (DSPx_INTC, where x = 1, 2)

There are two Digital Signal Processing (DSP) subsystems in the device - DSP1, and DSP2. Each DSP subsystem integrates an interrupt controller - DSPx_INTC, which interfaces the system events to the C66x core interrupt and exceptions inputs. It combines up to 128 interrupts into 12 prioritized interrupts presented to the C66x CPU.

For detailed information about this module, see Chapter 4, *DSP Subsystem*.

Dual Cortex-M4 IPU Subsystem Interrupt Controller (IPU_Cx_INTC, where x = 1, 2)

There is one Image Processing Unit (IPU) subsystem in the device. The IPU subsystem integrates two Arm® Cortex-M4 cores.

A Nested Vectored Interrupt Controller (NVIC) is integrated within each Cortex-M4. The interrupt mapping is the same for the two cores to facilitate parallel processing. The NVIC supports:

- 96 external interrupts (in addition to 16 Cortex-M4 internal interrupts), which are dynamically prioritized with 16 levels of priority defined for each core
- Low-latency exception and interrupt handling
- Prioritization and handling of exceptions
- Control of the local power management
- Debug accesses to the processor core

For detailed information about this module, refer to Arm *Cortex-M4 Technical Reference Manual* (available at infocenter.arm.com/help/index.jsp).

EVE Subsystem Interrupt Controller (EVE_INTC)

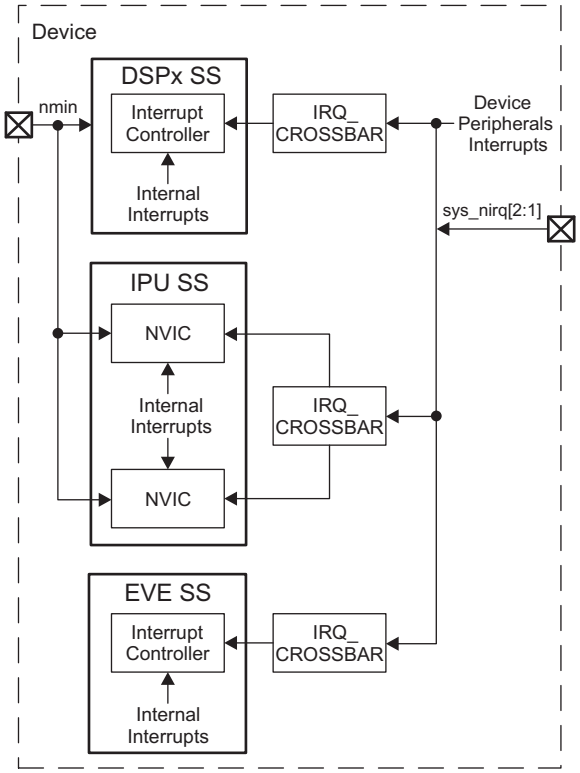
There is one Embedded Video Engine (EVE) subsystems in the device. The EVE subsystem integrates an interrupt controller - EVE_INTC, which handles incoming interrupts, merging them with internal interrupt sources to drive ARP32's interrupt inputs. It also allows ARP32 to generate outgoing interrupts or events to synchronize with other system processors and EDMA.

The EVE_INTC supports up to 32 active-high level interrupt inputs. Its architecture allows both hardware and software prioritization.

For detailed information about this module, see Section 6.1, *Embedded Vision Engine*.

Figure 12-1 shows the top-level block diagram of the interrupt controllers in this device.

Figure 12-1. Interrupt Controllers in the Device



12.2 Interrupt Controllers Environment

Figure 12-2 shows the relationship between the device INTCs and external interrupts.

Figure 12-2. Interrupts From External Devices

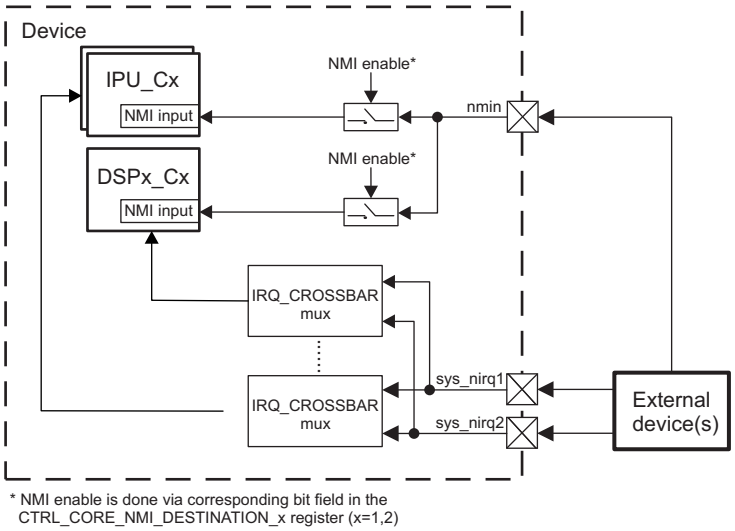


Table 12-1 describes the signals that can be used by external devices to generate interrupts to the device INTCs.

Table 12-1. Interrupts From External Devices

Device Pin	I/O[1]	Description
sys_nirq1	I	External devices can use these pins to generate a system wake-up interrupt event to any device INTC. The user must take care to configure the corresponding IRQ_CROSSBAR properly (via Control Module).
sys_nirq2	I	
nmin	I	External device can use this pin to generate a non-maskable interrupt (NMI) event to the following device processors:
		- IPU_C0, IPU_C1. Upon enable (via Control Module), the NMI is routed directly to the NMI input of Cortex M4 core. Note that NMI can be enabled separately for IPU_C0 and IPU_C1.
		- DSP1, DSP2. Upon enable (via Control Module), the NMI is routed directly to the NMI input of C66x CPU.
		The Control Module registers CTRL_CORE_NMI_DESTINATION_1 and CTRL_CORE_NMI_DESTINATION_2 are used to route the **NMI** signal to the corresponding device processor subsystems as follows: • Writing 0x1 to CTRL_CORE_NMI_DESTINATION_1[7:0] IPU_C1 enables NMI mapping to IPU_C1; • Writing 0x1 to CTRL_CORE_NMI_DESTINATION_2[31:24] IPU_C0 enables NMI mapping to IPU_C0; • Writing 0x1 to CTRL_CORE_NMI_DESTINATION_2[23:16] DSP2 enables NMI mapping to DSP2; • Writing 0x1 to CTRL_CORE_NMI_DESTINATION_2[15:8] DSP1 enables NMI mapping to DSP1; For more information about CTRL_CORE_NMI_DESTINATION_1 and CTRL_CORE_NMI_DESTINATION_2 registers, see Chapter 13, *Control Module*.

(1) I = Input, O = Output

NOTE: External devices can also use the GPIO modules to generate an interrupt to the device INTCs. For more information, see Chapter 20, *General-Purpose Interface*.

12.3 Interrupt Controllers Integration

Table 12-2 through Table 12-5 present the default interrupt mapping of the device INTCs. The mapping of device interrupts to IRQ_CROSSBAR inputs is presented in Table 12-6.

NOTE: All device interrupts (external to the DSP [x2], IPU, and EVE subsystems) are active-high, level-sensitive.

CAUTION

A single interrupt source can be physically mapped to multiple INTCs. With multiple-mapped interrupts, it is strongly recommended to unmask each interrupt source in only one INTC at a time.

12.3.1 Interrupt Requests to DSP1_INTC

Table 12-2 lists the default interrupt sources for the DSP1_INTC. In addition, interrupts DSP1_IRQ_32 through DSP1_IRQ_95 can alternatively be sourced through the DSP1's IRQ_CROSSBAR from one of the 460 multiplexed device interrupts listed in Table 12-6. The CTRL_CORE_DSP1_IRQ_y_z registers in the Control Module are used to select between the default interrupts and the multiplexed interrupts.

Table 12-2. DSP1_INTC Default Interrupt Mapping

IRQ Input Line	IRQ_ CROSSBAR Instance Number	IRQ_CROSSBAR Configuration Register	IRQ_ CROSSBAR Default Input Index	Default Interrupt Name	Default Interrupt Source Description
DSP1_IRQ_0	N/A	N/A	N/A	CGEM_IRQ_0	CGEM Internal Interrupt
DSP1_IRQ_1	N/A	N/A	N/A	CGEM_IRQ_1	CGEM Internal Interrupt
DSP1_IRQ_2	N/A	N/A	N/A	CGEM_IRQ_2	CGEM Internal Interrupt
DSP1_IRQ_3	N/A	N/A	N/A	CGEM_IRQ_3	CGEM Internal Interrupt
DSP1_IRQ_4	N/A	N/A	N/A	CGEM_IRQ_4	CGEM Internal Interrupt
DSP1_IRQ_5	N/A	N/A	N/A	CGEM_IRQ_5	CGEM Internal Interrupt
DSP1_IRQ_6	N/A	N/A	N/A	CGEM_IRQ_6	CGEM Internal Interrupt
DSP1_IRQ_7	N/A	N/A	N/A	CGEM_IRQ_7	CGEM Internal Interrupt
DSP1_IRQ_8	N/A	N/A	N/A	CGEM_IRQ_8	CGEM Internal Interrupt
DSP1_IRQ_9	N/A	N/A	N/A	CGEM_IRQ_9	CGEM Internal Interrupt
DSP1_IRQ_10	N/A	N/A	N/A	CGEM_IRQ_10	CGEM Internal Interrupt
DSP1_IRQ_11	N/A	N/A	N/A	CGEM_IRQ_11	CGEM Internal Interrupt
DSP1_IRQ_12	N/A	N/A	N/A	CGEM_IRQ_12	CGEM Internal Interrupt
DSP1_IRQ_13	N/A	N/A	N/A	CGEM_IRQ_13	CGEM Internal Interrupt
DSP1_IRQ_14	N/A	N/A	N/A	CGEM_IRQ_14	CGEM Internal Interrupt
DSP1_IRQ_15	N/A	N/A	N/A	CGEM_IRQ_15	CGEM Internal Interrupt
DSP1_IRQ_16	N/A	N/A	N/A	TPCC_INTG	EDMA CC global interrupt
DSP1_IRQ_17	N/A	N/A	N/A	TPCC_INT0	EDMA CC region0 interrupt
DSP1_IRQ_18	N/A	N/A	N/A	TPCC_INT1	EDMA CC region1 interrupt
DSP1_IRQ_19	N/A	N/A	N/A	TPCC_INT2	EDMA CC region2 interrupt
DSP1_IRQ_20	N/A	N/A	N/A	TPCC_INT3	EDMA CC region3 interrupt
DSP1_IRQ_21	N/A	N/A	N/A	FW0_FUNC_ERROR	Firewall0 func access error
DSP1_IRQ_22	N/A	N/A	N/A	FW0_DEBUG_ERROR	Firewall0 debug access error
DSP1_IRQ_23	N/A	N/A	N/A	FW1_FUNC_ERROR	Firewall1 func access error
DSP1_IRQ_24	N/A	N/A	N/A	FW1_DEBUG_ERROR	Firewall1 debug access error
DSP1_IRQ_25	N/A	N/A	N/A	MMU0_INT	DSP MMU0 Interrupt
DSP1_IRQ_26	N/A	N/A	N/A	MMU1_INT	DSP MMU1 Interrupt
DSP1_IRQ_27	N/A	N/A	N/A	TPCC_ERRINT	EDMA CC error interrupt
DSP1_IRQ_28	N/A	N/A	N/A	TPTC_ERRINT0	EDMA TC0 error interrupt
DSP1_IRQ_29	N/A	N/A	N/A	TPTC_ERRINT1	EDMA TC1 error interrupt
DSP1_IRQ_30	N/A	N/A	N/A	NOC_ERRINT	Interconnect error interrupt
DSP1_IRQ_31	NA	N/A	N/A	EDMA_WAKE_INT	EDMA wakeup interrupt

Table 12-2. DSP1_INTC Default Interrupt Mapping (continued)

IRQ Input Line	IRQ_ CROSSBAR Instance Number	IRQ_CROSSBAR Configuration Register	IRQ_ CROSSBAR Default Input Index	Default Interrupt Name	Default Interrupt Source Description
DSP1_IRQ_32	1	CTRL_CORE_DSP1_IRQ_32_33[8:0]	1	ELM_IRQ	Error location process completion interrupt
DSP1_IRQ_33	2	CTRL_CORE_DSP1_IRQ_32_33[24:16]	2	EXT_SYS_IRQ_1	External interrupt (active low) via sys_nirq1 pin
DSP1_IRQ_34	3	CTRL_CORE_DSP1_IRQ_34_35[8:0]	3	CTRL_MODULE_CORE_IRQ_SEC_EVTS	Combined firewall error interrupt. For more information, see Section 13.4.6.13.2.
DSP1_IRQ_35	4	CTRL_CORE_DSP1_IRQ_34_35[24:16]	4	L3_MAIN_IRQ_DBG_ERR	L3_MAIN debug error
DSP1_IRQ_36	5	CTRL_CORE_DSP1_IRQ_36_37[8:0]	5	L3_MAIN_IRQ_APP_ERR	L3_MAIN application or non-attributable error
DSP1_IRQ_37	6	CTRL_CORE_DSP1_IRQ_36_37[24:16]	6	Reserved	Reserved by default but can be remapped to a valid interrupt source
DSP1_IRQ_38	7	CTRL_CORE_DSP1_IRQ_38_39[8:0]	7	Reserved	Reserved by default but can be remapped to a valid interrupt source
DSP1_IRQ_39	8	CTRL_CORE_DSP1_IRQ_38_39[24:16]	8	Reserved	Reserved by default but can be remapped to a valid interrupt source
DSP1_IRQ_40	9	CTRL_CORE_DSP1_IRQ_40_41[8:0]	9	Reserved	Reserved by default but can be remapped to a valid interrupt source
DSP1_IRQ_41	10	CTRL_CORE_DSP1_IRQ_40_41[24:16]	10	Reserved	Reserved by default but can be remapped to a valid interrupt source
DSP1_IRQ_42	11	CTRL_CORE_DSP1_IRQ_42_43[8:0]	11	L3_MAIN_IRQ_STAT_ALARM	L3_MAIN statistic collector alarm interrupt
DSP1_IRQ_43	12	CTRL_CORE_DSP1_IRQ_42_43[24:16]	12	Reserved	Reserved by default but can be remapped to a valid interrupt source
DSP1_IRQ_44	13	CTRL_CORE_DSP1_IRQ_44_45[8:0]	13	Reserved	Reserved by default but can be remapped to a valid interrupt source
DSP1_IRQ_45	14	CTRL_CORE_DSP1_IRQ_44_45[24:16]	14	Reserved	Reserved by default but can be remapped to a valid interrupt source
DSP1_IRQ_46	15	CTRL_CORE_DSP1_IRQ_46_47[8:0]	15	GPMC_IRQ	GPMC interrupt
DSP1_IRQ_47	16	CTRL_CORE_DSP1_IRQ_46_47[24:16]	16	Reserved	Reserved by default but can be remapped to a valid interrupt source
DSP1_IRQ_48	17	CTRL_CORE_DSP1_IRQ_48_49[8:0]	17	Reserved	Reserved by default but can be remapped to a valid interrupt source
DSP1_IRQ_49	18	CTRL_CORE_DSP1_IRQ_48_49[24:16]	18	Reserved	Reserved by default but can be remapped to a valid interrupt source
DSP1_IRQ_50	19	CTRL_CORE_DSP1_IRQ_50_51[8:0]	19	Reserved	Reserved by default but can be remapped to a valid interrupt source
DSP1_IRQ_51	20	CTRL_CORE_DSP1_IRQ_50_51[24:16]	20	DISPC_IRQ	Display controller interrupt
DSP1_IRQ_52	21	CTRL_CORE_DSP1_IRQ_52_53[8:0]	21	MAILBOX1_IRQ_USER0	Mailbox 1 user 0 interrupt
DSP1_IRQ_53	22	CTRL_CORE_DSP1_IRQ_52_53[24:16]	22	Reserved	Reserved by default but can be remapped to a valid interrupt source

Table 12-2. DSP1_INTC Default Interrupt Mapping (continued)

IRQ Input Line	IRQ_ CROSSBAR Instance Number	IRQ_CROSSBAR Configuration Register	IRQ_ CROSSBAR Default Input Index	Default Interrupt Name	Default Interrupt Source Description
DSP1_IRQ_54	23	CTRL_CORE_DSP1_IRQ_54_55[8:0]	23	DSP1_IRQ_MMU0	DSP1 MMU0 interrupt
DSP1_IRQ_55	24	CTRL_CORE_DSP1_IRQ_54_55[24:16]	24	GPIO1_IRQ_1	GPIO1 interrupt 1
DSP1_IRQ_56	25	CTRL_CORE_DSP1_IRQ_56_57[8:0]	25	GPIO2_IRQ_1	GPIO2 interrupt 1
DSP1_IRQ_57	26	CTRL_CORE_DSP1_IRQ_56_57[24:16]	26	GPIO3_IRQ_1	GPIO3 interrupt 1
DSP1_IRQ_58	27	CTRL_CORE_DSP1_IRQ_58_59[8:0]	27	GPIO4_IRQ_1	GPIO4 interrupt 1
DSP1_IRQ_59	28	CTRL_CORE_DSP1_IRQ_58_59[24:16]	28	Reserved	Reserved by default but can be remapped to a valid interrupt source
DSP1_IRQ_60	29	CTRL_CORE_DSP1_IRQ_60_61[8:0]	29	Reserved	Reserved by default but can be remapped to a valid interrupt source
DSP1_IRQ_61	30	CTRL_CORE_DSP1_IRQ_60_61[24:16]	30	Reserved	Reserved by default but can be remapped to a valid interrupt source
DSP1_IRQ_62	31	CTRL_CORE_DSP1_IRQ_62_63[8:0]	31	Reserved	Reserved by default but can be remapped to a valid interrupt source
DSP1_IRQ_63	32	CTRL_CORE_DSP1_IRQ_62_63[24:16]	32	TIMER1_IRQ	TIMER1 interrupt
DSP1_IRQ_64	33	CTRL_CORE_DSP1_IRQ_64_65[8:0]	33	TIMER2_IRQ	TIMER2 interrupt
DSP1_IRQ_65	34	CTRL_CORE_DSP1_IRQ_64_65[24:16]	34	TIMER3_IRQ	TIMER3 interrupt
DSP1_IRQ_66	35	CTRL_CORE_DSP1_IRQ_66_67[8:0]	35	TIMER4_IRQ	TIMER4 interrupt
DSP1_IRQ_67	36	CTRL_CORE_DSP1_IRQ_66_67[24:16]	36	TIMER5_IRQ	TIMER5 interrupt
DSP1_IRQ_68	37	CTRL_CORE_DSP1_IRQ_68_69[8:0]	37	TIMER6_IRQ	TIMER6 interrupt
DSP1_IRQ_69	38	CTRL_CORE_DSP1_IRQ_68_69[24:16]	38	TIMER7_IRQ	TIMER7 interrupt
DSP1_IRQ_70	39	CTRL_CORE_DSP1_IRQ_70_71[8:0]	39	TIMER8_IRQ	TIMER8 interrupt
DSP1_IRQ_71	40	CTRL_CORE_DSP1_IRQ_70_71[24:16]	40	Reserved	Reserved by default but can be remapped to a valid interrupt source
DSP1_IRQ_72	41	CTRL_CORE_DSP1_IRQ_72_73[8:0]	41	Reserved	Reserved by default but can be remapped to a valid interrupt source
DSP1_IRQ_73	42	CTRL_CORE_DSP1_IRQ_72_73[24:16]	42	Reserved	Reserved by default but can be remapped to a valid interrupt source
DSP1_IRQ_74	43	CTRL_CORE_DSP1_IRQ_74_75[8:0]	43	MCSPI4_IRQ	McSPI4 interrupt
DSP1_IRQ_75	44	CTRL_CORE_DSP1_IRQ_74_75[24:16]	44	Reserved	Reserved by default but can be remapped to a valid interrupt source
DSP1_IRQ_76	45	CTRL_CORE_DSP1_IRQ_76_77[8:0]	45	Reserved	Reserved by default but can be remapped to a valid interrupt source
DSP1_IRQ_77	46	CTRL_CORE_DSP1_IRQ_76_77[24:16]	46	Reserved	Reserved by default but can be remapped to a valid interrupt source

Table 12-2. DSP1_INTC Default Interrupt Mapping (continued)

IRQ Input Line	IRQ_ CROSSBAR Instance Number	IRQ_CROSSBAR Configuration Register	IRQ_ CROSSBAR Default Input Index	Default Interrupt Name	Default Interrupt Source Description
DSP1_IRQ_78	47	CTRL_CORE_DSP1_IRQ_78_79[8:0]	47	Reserved	Reserved by default but can be remapped to a valid interrupt source
DSP1_IRQ_79	48	CTRL_CORE_DSP1_IRQ_78_79[24:16]	48	Reserved	Reserved by default but can be remapped to a valid interrupt source
DSP1_IRQ_80	49	CTRL_CORE_DSP1_IRQ_80_81[8:0]	49	Reserved	Reserved by default but can be remapped to a valid interrupt source
DSP1_IRQ_81	50	CTRL_CORE_DSP1_IRQ_80_81[24:16]	50	Reserved	Reserved by default but can be remapped to a valid interrupt source
DSP1_IRQ_82	51	CTRL_CORE_DSP1_IRQ_82_83[8:0]	51	I2C1_IRQ	I2C1 interrupt
DSP1_IRQ_83	52	CTRL_CORE_DSP1_IRQ_82_83[24:16]	52	I2C2_IRQ	I2C2 interrupt
DSP1_IRQ_84	53	CTRL_CORE_DSP1_IRQ_84_85[8:0]	53	Reserved	Reserved by default but can be remapped to a valid interrupt source
DSP1_IRQ_85	54	CTRL_CORE_DSP1_IRQ_84_85[24:16]	54	Reserved	Reserved by default but can be remapped to a valid interrupt source
DSP1_IRQ_86	55	CTRL_CORE_DSP1_IRQ_86_87[8:0]	55	Reserved	Reserved by default but can be remapped to a valid interrupt source
DSP1_IRQ_87	56	CTRL_CORE_DSP1_IRQ_86_87[24:16]	56	Reserved	Reserved by default but can be remapped to a valid interrupt source
DSP1_IRQ_88	57	CTRL_CORE_DSP1_IRQ_88_89[8:0]	57	Reserved	Reserved by default but can be remapped to a valid interrupt source
DSP1_IRQ_89	58	CTRL_CORE_DSP1_IRQ_88_89[24:16]	58	Reserved	Reserved by default but can be remapped to a valid interrupt source
DSP1_IRQ_90	59	CTRL_CORE_DSP1_IRQ_90_91[8:0]	59	Reserved	Reserved by default but can be remapped to a valid interrupt source
DSP1_IRQ_91	60	CTRL_CORE_DSP1_IRQ_90_91[24:16]	60	MCSPI1_IRQ	McSPI1 interrupt
DSP1_IRQ_92	61	CTRL_CORE_DSP1_IRQ_92_93[8:0]	61	MCSPI2_IRQ	McSPI2 interrupt
DSP1_IRQ_93	62	CTRL_CORE_DSP1_IRQ_92_93[24:16]	62	Reserved	Reserved by default but can be remapped to a valid interrupt source
DSP1_IRQ_94	63	CTRL_CORE_DSP1_IRQ_94_95[8:0]	63	Reserved	Reserved by default but can be remapped to a valid interrupt source
DSP1_IRQ_95	64	CTRL_CORE_DSP1_IRQ_94_95[24:16]	64	Reserved	Reserved by default but can be remapped to a valid interrupt source
DSP1_IRQ_96	N/A	N/A	N/A	CGEM_IRQ_16	CGEM Internal Interrupt
DSP1_IRQ_97	N/A	N/A	N/A	CGEM_IRQ_17	CGEM Internal Interrupt
DSP1_IRQ_98	N/A	N/A	N/A	CGEM_IRQ_18	CGEM Internal Interrupt
DSP1_IRQ_99	NA	N/A	N/A	CGEM_IRQ_19	CGEM Internal Interrupt

Table 12-2. DSP1_INTC Default Interrupt Mapping (continued)

IRQ Input Line	IRQ_ CROSSBAR Instance Number	IRQ_CROSSBAR Configuration Register	IRQ_ CROSSBAR Default Input Index	Default Interrupt Name	Default Interrupt Source Description
DSP1_IRQ_100	N/A	N/A	N/A	CGEM_IRQ_20	CGEM Internal Interrupt
DSP1_IRQ_101	N/A	N/A	N/A	CGEM_IRQ_21	CGEM Internal Interrupt
DSP1_IRQ_102	N/A	N/A	N/A	CGEM_IRQ_22	CGEM Internal Interrupt
DSP1_IRQ_103	N/A	N/A	N/A	CGEM_IRQ_23	CGEM Internal Interrupt
DSP1_IRQ_104	N/A	N/A	N/A	CGEM_IRQ_24	CGEM Internal Interrupt
DSP1_IRQ_105	NA	N/A	N/A	CGEM_IRQ_25	CGEM Internal Interrupt
DSP1_IRQ_106	N/A	N/A	N/A	CGEM_IRQ_26	CGEM Internal Interrupt
DSP1_IRQ_107	N/A	N/A	N/A	CGEM_IRQ_27	CGEM Internal Interrupt
DSP1_IRQ_108	N/A	N/A	N/A	CGEM_IRQ_28	CGEM Internal Interrupt
DSP1_IRQ_109	N/A	N/A	N/A	CGEM_IRQ_29	CGEM Internal Interrupt
DSP1_IRQ_110	N/A	N/A	N/A	CGEM_IRQ_30	CGEM Internal Interrupt
DSP1_IRQ_111	N/A	N/A	N/A	CGEM_IRQ_31	CGEM Internal Interrupt
DSP1_IRQ_112	N/A	N/A	N/A	CGEM_IRQ_32	CGEM Internal Interrupt
DSP1_IRQ_113	N/A	N/A	N/A	CGEM_IRQ_33	CGEM Internal Interrupt
DSP1_IRQ_114	N/A	N/A	N/A	CGEM_IRQ_34	CGEM Internal Interrupt
DSP1_IRQ_115	N/A	N/A	N/A	CGEM_IRQ_35	CGEM Internal Interrupt
DSP1_IRQ_116	N/A	N/A	N/A	CGEM_IRQ_36	CGEM Internal Interrupt
DSP1_IRQ_117	N/A	N/A	N/A	CGEM_IRQ_37	CGEM Internal Interrupt
DSP1_IRQ_118	N/A	N/A	N/A	CGEM_IRQ_38	CGEM Internal Interrupt
DSP1_IRQ_119	NA	N/A	N/A	CGEM_IRQ_39	CGEM Internal Interrupt
DSP1_IRQ_120	N/A	N/A	N/A	CGEM_IRQ_40	CGEM Internal Interrupt
DSP1_IRQ_121	N/A	N/A	N/A	CGEM_IRQ_41	CGEM Internal Interrupt
DSP1_IRQ_122	N/A	N/A	N/A	CGEM_IRQ_42	CGEM Internal Interrupt
DSP1_IRQ_123	N/A	N/A	N/A	CGEM_IRQ_43	CGEM Internal Interrupt
DSP1_IRQ_124	N/A	N/A	N/A	CGEM_IRQ_44	CGEM Internal Interrupt
DSP1_IRQ_125	N/A	N/A	N/A	CGEM_IRQ_45	CGEM Internal Interrupt
DSP1_IRQ_126	N/A	N/A	N/A	CGEM_IRQ_46	CGEM Internal Interrupt
DSP1_IRQ_127	N/A		N/A	CGEM_IRQ_47	CGEM Internal Interrupt

NOTE: The "**IRQ_CROSSBAR Default Input Index**" column of Table 12-2 shows which input of the corresponding IRQ_CROSSBAR instance is mapped to its output (and then routed to the corresponding DSP1_INTC input) by default. In other words, this column specifies the default (reset) values (in decimal) of the CTRL_CORE_DSP1_IRQ_y_z register bit fields that are used to control the mapping of device interrupts to DSP1_INTC inputs. For example, the DSP1_IRQ_32_33[8:0] bit field is used to configure which device interrupt would be mapped to the DSP1_IRQ_32 line. The reset value of this bit field is 0x1, meaning that ELM_IRQ would be mapped to DSP1_IRQ_32 by default because it is connected to the IRQ_CROSSBAR_1 input.

'N/A' in this column means that the corresponding interrupt is internal to the DSP1 subsystem. There is no IRQ_CROSSBAR dedicated to the associated DSP1_INTC input line and therefore, the user cannot change its default mapping.

12.3.2 Interrupt Requests to DSP2_INTC

Table 12-3 lists the default interrupt sources for the DSP2_INTC. In addition, interrupts DSP2_IRQ_32 through DSP2_IRQ_95 can alternatively be sourced through the DSP2's IRQ_CROSSBAR from one of the 460 multiplexed device interrupts listed in Table 12-6. The DSP2_IRQ_y_z registers in the Control Module are used to select between the default interrupts and the multiplexed interrupts.

Table 12-3. DSP2_INTC Default Interrupt Mapping

IRQ Input Line	IRQ_ CROSSBAR Instance Number	IRQ_CROSSBAR Configuration Register	IRQ_ CROSSBAR Default Input Index	Default Interrupt Name	Default Interrupt Source Description
DSP2_IRQ_0	N/A	N/A	N/A	CGEM_IRQ_0	CGEM Internal Interrupt
DSP2_IRQ_1	N/A	N/A	N/A	CGEM_IRQ_1	CGEM Internal Interrupt
DSP2_IRQ_2	N/A	N/A	N/A	CGEM_IRQ_2	CGEM Internal Interrupt
DSP2_IRQ_3	N/A	N/A	N/A	CGEM_IRQ_3	CGEM Internal Interrupt
DSP2_IRQ_4	N/A	N/A	N/A	CGEM_IRQ_4	CGEM Internal Interrupt
DSP2_IRQ_5	N/A	N/A	N/A	CGEM_IRQ_5	CGEM Internal Interrupt
DSP2_IRQ_6	N/A	N/A	N/A	CGEM_IRQ_6	CGEM Internal Interrupt
DSP2_IRQ_7	N/A	N/A	N/A	CGEM_IRQ_7	CGEM Internal Interrupt
DSP2_IRQ_8	N/A	N/A	N/A	CGEM_IRQ_8	CGEM Internal Interrupt
DSP2_IRQ_9	N/A	N/A	N/A	CGEM_IRQ_9	CGEM Internal Interrupt
DSP2_IRQ_10	N/A	N/A	N/A	CGEM_IRQ_10	CGEM Internal Interrupt
DSP2_IRQ_11	N/A	N/A	N/A	CGEM_IRQ_11	CGEM Internal Interrupt
DSP2_IRQ_12	N/A	N/A	N/A	CGEM_IRQ_12	CGEM Internal Interrupt
DSP2_IRQ_13	N/A	N/A	N/A	CGEM_IRQ_13	CGEM Internal Interrupt
DSP2_IRQ_14	N/A	N/A	N/A	CGEM_IRQ_14	CGEM Internal Interrupt
DSP2_IRQ_15	N/A	N/A	N/A	CGEM_IRQ_15	CGEM Internal Interrupt
DSP2_IRQ_16	N/A	N/A	N/A	TPCC_INTG	EDMA CC global interrupt

Table 12-3. DSP2_INTC Default Interrupt Mapping (continued)

IRQ Input Line	IRQ_ CROSSBAR Instance Number	IRQ_CROSSBAR Configuration Register	IRQ_ CROSSBAR Default Input Index	Default Interrupt Name	Default Interrupt Source Description
DSP2_IRQ_17	N/A	N/A	N/A	TPCC_INT0	EDMA CC region0 interrupt
DSP2_IRQ_18	N/A	N/A	N/A	TPCC_INT1	EDMA CC region1 interrupt
DSP2_IRQ_19	N/A	N/A	N/A	TPCC_INT2	EDMA CC region2 interrupt
DSP2_IRQ_20	N/A	N/A	N/A	TPCC_INT3	EDMA CC region3 interrupt
DSP2_IRQ_21	N/A	N/A	N/A	FW0_FUNC_ERROR	Firewall0 func access error
DSP2_IRQ_22	N/A	N/A	N/A	FW0_DEBUG_ERROR	Firewall0 debug access error
DSP2_IRQ_23	N/A	N/A	N/A	FW1_FUNC_ERROR	Firewall1 func access error
DSP2_IRQ_24	N/A	N/A	N/A	FW1_DEBUG_ERROR	Firewall1 debug access error
DSP2_IRQ_25	N/A	N/A	N/A	MMU0_INT	DSP MMU0 Interrupt
DSP2_IRQ_26	N/A	N/A	N/A	MMU1_INT	DSP MMU1 Interrupt
DSP2_IRQ_27	N/A	N/A	N/A	TPCC_ERRINT	EDMA CC error interrupt
DSP2_IRQ_28	N/A	N/A	N/A	TPTC_ERRINT0	EDMA TC0 error interrupt
DSP2_IRQ_29	N/A	N/A	N/A	TPTC_ERRINT1	EDMA TC1 error interrupt
DSP2_IRQ_30	N/A	N/A	N/A	NOC_ERRINT	Interconnect error interrupt
DSP2_IRQ_31	NA	N/A	N/A	EDMA_WAKE_INT	EDMA wakeup interrupt
DSP2_IRQ_32	1	CTRL_CORE_DSP2_IRQ_32_33[8:0]	1	ELM_IRQ	Error location process completion interrupt
DSP2_IRQ_33	2	CTRL_CORE_DSP2_IRQ_32_33[24:16]	2	EXT_SYS_IRQ_1	External interrupt (active low) via sys_nirq1 pin
DSP2_IRQ_34	3	CTRL_CORE_DSP2_IRQ_34_35[8:0]	3	CTRL_MODULE_CORE_IRQ_SEC_EVTS	Combined firewall error interrupt. For more information, see Section 13.4.6.13.2.
DSP2_IRQ_35	4	CTRL_CORE_DSP2_IRQ_34_35[24:16]	4	L3_MAIN_IRQ_DBG_ERR	L3_MAIN debug error
DSP2_IRQ_36	5	CTRL_CORE_DSP2_IRQ_36_37[8:0]	5	L3_MAIN_IRQ_APP_ERR	L3_MAIN application or non-attributable error
DSP2_IRQ_37	6	CTRL_CORE_DSP2_IRQ_36_37[24:16]	6	Reserved	Reserved by default but can be remapped to a valid interrupt source
DSP2_IRQ_38	7	CTRL_CORE_DSP2_IRQ_38_39[8:0]	7	Reserved	Reserved by default but can be remapped to a valid interrupt source
DSP2_IRQ_39	8	CTRL_CORE_DSP2_IRQ_38_39[24:16]	8	Reserved	Reserved by default but can be remapped to a valid interrupt source
DSP2_IRQ_40	9	CTRL_CORE_DSP2_IRQ_40_41[8:0]	9	Reserved	Reserved by default but can be remapped to a valid interrupt source
DSP2_IRQ_41	10	CTRL_CORE_DSP2_IRQ_40_41[24:16]	10	Reserved	Reserved by default but can be remapped to a valid interrupt source
DSP2_IRQ_42	11	CTRL_CORE_DSP2_IRQ_42_43[8:0]	11	L3_MAIN_IRQ_STAT_ALARM	L3_MAIN statistic collector alarm interrupt
DSP2_IRQ_43	12	CTRL_CORE_DSP2_IRQ_42_43[24:16]	12	Reserved	Reserved by default but can be remapped to a valid interrupt source

Table 12-3. DSP2_INTC Default Interrupt Mapping (continued)

IRQ Input Line	IRQ_ CROSSBAR Instance Number	IRQ_CROSSBAR Configuration Register	IRQ_ CROSSBAR Default Input Index	Default Interrupt Name	Default Interrupt Source Description
DSP2_IRQ_44	13	CTRL_CORE_DSP2_IRQ_44_45[8:0]	13	Reserved	Reserved by default but can be remapped to a valid interrupt source
DSP2_IRQ_45	14	CTRL_CORE_DSP2_IRQ_44_45[24:16]	14	Reserved	Reserved by default but can be remapped to a valid interrupt source
DSP2_IRQ_46	15	CTRL_CORE_DSP2_IRQ_46_47[8:0]	15	GPMC_IRQ	GPMC interrupt
DSP2_IRQ_47	16	CTRL_CORE_DSP2_IRQ_46_47[24:16]	16	Reserved	Reserved by default but can be remapped to a valid interrupt source
DSP2_IRQ_48	17	CTRL_CORE_DSP2_IRQ_48_49[8:0]	17	Reserved	Reserved by default but can be remapped to a valid interrupt source
DSP2_IRQ_49	18	CTRL_CORE_DSP2_IRQ_48_49[24:16]	18	Reserved	Reserved by default but can be remapped to a valid interrupt source
DSP2_IRQ_50	19	CTRL_CORE_DSP2_IRQ_50_51[8:0]	19	Reserved	Reserved by default but can be remapped to a valid interrupt source
DSP2_IRQ_51	20	CTRL_CORE_DSP2_IRQ_50_51[24:16]	20	DISPC_IRQ	Display controller interrupt
DSP2_IRQ_52	21	CTRL_CORE_DSP2_IRQ_52_53[8:0]	21	MAILBOX1_IRQ_USER0	Mailbox 1 user 0 interrupt
DSP2_IRQ_53	22	CTRL_CORE_DSP2_IRQ_52_53[24:16]	22	Reserved	Reserved by default but can be remapped to a valid interrupt source
DSP2_IRQ_54	23	CTRL_CORE_DSP2_IRQ_54_55[8:0]	23	DSP2_IRQ_MMU0	DSP2 MMU0 interrupt
DSP2_IRQ_55	24	CTRL_CORE_DSP2_IRQ_54_55[24:16]	24	GPIO1_IRQ_1	GPIO1 interrupt 1
DSP2_IRQ_56	25	CTRL_CORE_DSP2_IRQ_56_57[8:0]	25	GPIO2_IRQ_1	GPIO2 interrupt 1
DSP2_IRQ_57	26	CTRL_CORE_DSP2_IRQ_56_57[24:16]	26	GPIO3_IRQ_1	GPIO3 interrupt 1
DSP2_IRQ_58	27	CTRL_CORE_DSP2_IRQ_58_59[8:0]	27	GPIO4_IRQ_1	GPIO4 interrupt 1
DSP2_IRQ_59	28	CTRL_CORE_DSP2_IRQ_58_59[24:16]	28	Reserved	Reserved by default but can be remapped to a valid interrupt source
DSP2_IRQ_60	29	CTRL_CORE_DSP2_IRQ_60_61[8:0]	29	Reserved	Reserved by default but can be remapped to a valid interrupt source
DSP2_IRQ_61	30	CTRL_CORE_DSP2_IRQ_60_61[24:16]	30	Reserved	Reserved by default but can be remapped to a valid interrupt source
DSP2_IRQ_62	31	CTRL_CORE_DSP2_IRQ_62_63[8:0]	31	Reserved	Reserved by default but can be remapped to a valid interrupt source
DSP2_IRQ_63	32	CTRL_CORE_DSP2_IRQ_62_63[24:16]	32	TIMER1_IRQ	TIMER1 interrupt
DSP2_IRQ_64	33	CTRL_CORE_DSP2_IRQ_64_65[8:0]	33	TIMER2_IRQ	TIMER2 interrupt
DSP2_IRQ_65	34	CTRL_CORE_DSP2_IRQ_64_65[24:16]	34	TIMER3_IRQ	TIMER3 interrupt
DSP2_IRQ_66	35	CTRL_CORE_DSP2_IRQ_66_67[8:0]	35	TIMER4_IRQ	TIMER4 interrupt
DSP2_IRQ_67	36	CTRL_CORE_DSP2_IRQ_66_67[24:16]	36	TIMER5_IRQ	TIMER5 interrupt

Table 12-3. DSP2_INTC Default Interrupt Mapping (continued)

IRQ Input Line	IRQ_ CROSSBAR Instance Number	IRQ_CROSSBAR Configuration Register	IRQ_ CROSSBAR Default Input Index	Default Interrupt Name	Default Interrupt Source Description
DSP2_IRQ_68	37	CTRL_CORE_DSP2_IRQ_68_69[8:0]	37	TIMER6_IRQ	TIMER6 interrupt
DSP2_IRQ_69	38	CTRL_CORE_DSP2_IRQ_68_69[24:16]	38	TIMER7_IRQ	TIMER7 interrupt
DSP2_IRQ_70	39	CTRL_CORE_DSP2_IRQ_70_71[8:0]	39	TIMER8_IRQ	TIMER8 interrupt
DSP2_IRQ_71	40	CTRL_CORE_DSP2_IRQ_70_71[24:16]	40	Reserved	Reserved by default but can be remapped to a valid interrupt source
DSP2_IRQ_72	41	CTRL_CORE_DSP2_IRQ_72_73[8:0]	41	Reserved	Reserved by default but can be remapped to a valid interrupt source
DSP2_IRQ_73	42	CTRL_CORE_DSP2_IRQ_72_73[24:16]	42	Reserved	Reserved by default but can be remapped to a valid interrupt source
DSP2_IRQ_74	43	CTRL_CORE_DSP2_IRQ_74_75[8:0]	43	MCSPI4_IRQ	McSPI4 interrupt
DSP2_IRQ_75	44	CTRL_CORE_DSP2_IRQ_74_75[24:16]	44	Reserved	Reserved by default but can be remapped to a valid interrupt source
DSP2_IRQ_76	45	CTRL_CORE_DSP2_IRQ_76_77[8:0]	45	Reserved	Reserved by default but can be remapped to a valid interrupt source
DSP2_IRQ_77	46	CTRL_CORE_DSP2_IRQ_76_77[24:16]	46	Reserved	Reserved by default but can be remapped to a valid interrupt source
DSP2_IRQ_78	47	CTRL_CORE_DSP2_IRQ_78_79[8:0]	47	Reserved	Reserved by default but can be remapped to a valid interrupt source
DSP2_IRQ_79	48	CTRL_CORE_DSP2_IRQ_78_79[24:16]	48	Reserved	Reserved by default but can be remapped to a valid interrupt source
DSP2_IRQ_80	49	CTRL_CORE_DSP2_IRQ_80_81[8:0]	49	Reserved	Reserved by default but can be remapped to a valid interrupt source
DSP2_IRQ_81	50	CTRL_CORE_DSP2_IRQ_80_81[24:16]	50	Reserved	Reserved by default but can be remapped to a valid interrupt source
DSP2_IRQ_82	51	CTRL_CORE_DSP2_IRQ_82_83[8:0]	51	I2C1_IRQ	I2C1 interrupt
DSP2_IRQ_83	52	CTRL_CORE_DSP2_IRQ_82_83[24:16]	52	I2C2_IRQ	I2C2 interrupt
DSP2_IRQ_84	53	CTRL_CORE_DSP2_IRQ_84_85[8:0]	53	Reserved	Reserved by default but can be remapped to a valid interrupt source
DSP2_IRQ_85	54	CTRL_CORE_DSP2_IRQ_84_85[24:16]	54	Reserved	Reserved by default but can be remapped to a valid interrupt source
DSP2_IRQ_86	55	CTRL_CORE_DSP2_IRQ_86_87[8:0]	55	Reserved	Reserved by default but can be remapped to a valid interrupt source
DSP2_IRQ_87	56	CTRL_CORE_DSP2_IRQ_86_87[24:16]	56	Reserved	Reserved by default but can be remapped to a valid interrupt source
DSP2_IRQ_88	57	CTRL_CORE_DSP2_IRQ_88_89[8:0]	57	Reserved	Reserved by default but can be remapped to a valid interrupt source

Table 12-3. DSP2_INTC Default Interrupt Mapping (continued)

IRQ Input Line	IRQ_ CROSSBAR Instance Number	IRQ_CROSSBAR Configuration Register	IRQ_ CROSSBAR Default Input Index	Default Interrupt Name	Default Interrupt Source Description
DSP2_IRQ_89	58	CTRL_CORE_DSP2_IRQ_88_89[24:16]	58	Reserved	Reserved by default but can be remapped to a valid interrupt source
DSP2_IRQ_90	59	CTRL_CORE_DSP2_IRQ_90_91[8:0]	59	Reserved	Reserved by default but can be remapped to a valid interrupt source
DSP2_IRQ_91	60	CTRL_CORE_DSP2_IRQ_90_91[24:16]	60	MCSPI1_IRQ	McSPI1 interrupt
DSP2_IRQ_92	61	CTRL_CORE_DSP2_IRQ_92_93[8:0]	61	MCSPI2_IRQ	McSPI2 interrupt
DSP2_IRQ_93	62	CTRL_CORE_DSP2_IRQ_92_93[24:16]	62	Reserved	Reserved by default but can be remapped to a valid interrupt source
DSP2_IRQ_94	63	CTRL_CORE_DSP2_IRQ_94_95[8:0]	63	Reserved	Reserved by default but can be remapped to a valid interrupt source
DSP2_IRQ_95	64	CTRL_CORE_DSP2_IRQ_94_95[24:16]	64	Reserved	Reserved by default but can be remapped to a valid interrupt source
DSP2_IRQ_96	N/A	N/A	N/A	CGEM_IRQ_16	CGEM Internal Interrupt
DSP2_IRQ_97	N/A	N/A	N/A	CGEM_IRQ_17	CGEM Internal Interrupt
DSP2_IRQ_98	N/A	N/A	N/A	CGEM_IRQ_18	CGEM Internal Interrupt
DSP2_IRQ_99	NA	N/A	N/A	CGEM_IRQ_19	CGEM Internal Interrupt
DSP2_IRQ_100	N/A	N/A	N/A	CGEM_IRQ_20	CGEM Internal Interrupt
DSP2_IRQ_101	N/A	N/A	N/A	CGEM_IRQ_21	CGEM Internal Interrupt
DSP2_IRQ_102	N/A	N/A	N/A	CGEM_IRQ_22	CGEM Internal Interrupt
DSP2_IRQ_103	N/A	N/A	N/A	CGEM_IRQ_23	CGEM Internal Interrupt
DSP2_IRQ_104	N/A	N/A	N/A	CGEM_IRQ_24	CGEM Internal Interrupt
DSP2_IRQ_105	NA	N/A	N/A	CGEM_IRQ_25	CGEM Internal Interrupt
DSP2_IRQ_106	N/A	N/A	N/A	CGEM_IRQ_26	CGEM Internal Interrupt
DSP2_IRQ_107	N/A	N/A	N/A	CGEM_IRQ_27	CGEM Internal Interrupt
DSP2_IRQ_108	N/A	N/A	N/A	CGEM_IRQ_28	CGEM Internal Interrupt
DSP2_IRQ_109	N/A	N/A	N/A	CGEM_IRQ_29	CGEM Internal Interrupt
DSP2_IRQ_110	N/A	N/A	N/A	CGEM_IRQ_30	CGEM Internal Interrupt
DSP2_IRQ_111	N/A	N/A	N/A	CGEM_IRQ_31	CGEM Internal Interrupt
DSP2_IRQ_112	N/A	N/A	N/A	CGEM_IRQ_32	CGEM Internal Interrupt
DSP2_IRQ_113	N/A	N/A	N/A	CGEM_IRQ_33	CGEM Internal Interrupt
DSP2_IRQ_114	N/A	N/A	N/A	CGEM_IRQ_34	CGEM Internal Interrupt
DSP2_IRQ_115	N/A	N/A	N/A	CGEM_IRQ_35	CGEM Internal Interrupt
DSP2_IRQ_116	N/A	N/A	N/A	CGEM_IRQ_36	CGEM Internal Interrupt

Table 12-3. DSP2_INTC Default Interrupt Mapping (continued)

IRQ Input Line	IRQ_ CROSSBAR Instance Number	IRQ_CROSSBAR Configuration Register	IRQ_ CROSSBAR Default Input Index	Default Interrupt Name	Default Interrupt Source Description
DSP2_IRQ_117	N/A	N/A	N/A	CGEM_IRQ_37	CGEM Internal Interrupt
DSP2_IRQ_118	N/A	N/A	N/A	CGEM_IRQ_38	CGEM Internal Interrupt
DSP2_IRQ_119	NA	N/A	N/A	CGEM_IRQ_39	CGEM Internal Interrupt
DSP2_IRQ_120	N/A	N/A	N/A	CGEM_IRQ_40	CGEM Internal Interrupt
DSP2_IRQ_121	N/A	N/A	N/A	CGEM_IRQ_41	CGEM Internal Interrupt
DSP2_IRQ_122	N/A	N/A	N/A	CGEM_IRQ_42	CGEM Internal Interrupt
DSP2_IRQ_123	N/A	N/A	N/A	CGEM_IRQ_43	CGEM Internal Interrupt
DSP2_IRQ_124	N/A	N/A	N/A	CGEM_IRQ_44	CGEM Internal Interrupt
DSP2_IRQ_125	N/A	N/A	N/A	CGEM_IRQ_45	CGEM Internal Interrupt
DSP2_IRQ_126	N/A	N/A	N/A	CGEM_IRQ_46	CGEM Internal Interrupt
DSP2_IRQ_127	N/A		N/A	CGEM_IRQ_47	CGEM Internal Interrupt

NOTE: The "**IRQ_CROSSBAR Default Input Index**" column of Table 12-3 shows which input of the corresponding IRQ_CROSSBAR instance is mapped to its output (and then routed to the corresponding DSP2_INTC input) by default. In other words, this column specifies the default (reset) values (in decimal) of the CTRL_CORE_DSP2_IRQ_y_z register bit fields that are used to control the mapping of device interrupts to DSP2_INTC inputs. For example, the DSP2_IRQ_32_33[8:0] bit field is used to configure which device interrupt would be mapped to the DSP2_IRQ_32 line. The reset value of this bit field is 0x1, meaning that ELM_IRQ would be mapped to DSP2_IRQ_32 by default because it is connected to the IRQ_CROSSBAR_1 input.

'N/A' in this column means that the corresponding interrupt is internal to the DSP2 subsystem. There is no IRQ_CROSSBAR dedicated to the associated DSP2_INTC input line and therefore, the user cannot change its default mapping.

12.3.3 Interrupt Requests to IPU_Cx_INTC

Table 12-4 lists the default interrupt sources for the IPU_Cx_INTC. In addition, device interrupts IPU_IRQ_23 through IPU_IRQ_79 can alternatively be sourced through the IPU's IRQ_CROSSBAR from one of the 460 multiplexed device interrupts listed in Table 12-6. The CTRL_CORE_IPU_IRQ_y_z registers in the Control Module are used to select between the default interrupts and the multiplexed interrupts.

Table 12-4. IPU_Cx_INTC Default Interrupt Mapping

IRQ Input Line[(1)]	IRQ_ CROSSBAR Instance Number	IRQ_CROSSBAR Configuration Register	IRQ_ CROSSBAR Default Input Index	Default Interrupt Name	Default Interrupt Source Description
IPU_IRQ_0	N/A	N/A	N/A	Reserved	MSP initial value in exception vector table

(1) This column shows the number of the corresponding exception type.

Table 12-4. IPU_Cx_INTC Default Interrupt Mapping (continued)

IRQ Input Line[1]	IRQ_ CROSSBAR Instance Number	IRQ_CROSSBAR Configuration Register	IRQ_ CROSSBAR Default Input Index	Default Interrupt Name	Default Interrupt Source Description
IPU_IRQ_1	N/A	N/A	N/A	RESET_IRQ	Reset
IPU_IRQ_2	N/A	N/A	N/A	NMI_IRQ	External NMI input (interrupt from **nmin** device pad)
IPU_IRQ_3	N/A	N/A	N/A	HARD_FAULT_IRQ	All fault conditions, if the fault handle is not enabled
IPU_IRQ_4	N/A	N/A	N/A	MEM_MANAGE_FAULT_IRQ	Memory management fault; access to illegal locations
IPU_IRQ_5	N/A	N/A	N/A	BUS_FAULT_IRQ	Bus error (on AHB intf)
IPU_IRQ_6	N/A	N/A	N/A	USAGE_FAULT_IRQ	Program error
IPU_IRQ_7	N/A	N/A	N/A	Reserved	Reserved
IPU_IRQ_8	N/A	N/A	N/A	Reserved	Reserved
IPU_IRQ_9	NA	N/A	N/A	Reserved	Reserved
IPU_IRQ_10	N/A	N/A	N/A	Reserved	Reserved
IPU_IRQ_11	N/A	N/A	N/A	SV_CALL_IRQ	Service system Call
IPU_IRQ_12	N/A	N/A	N/A	DEBUG_MON_IRQ	BP, WP or external debug req.
IPU_IRQ_13	N/A	N/A	N/A	Reserved	Reserved
IPU_IRQ_14	N/A	N/A	N/A	PEND_SV_IRQ	Pendable request for system device
IPU_IRQ_15	N/A	N/A	N/A	SYS_TICK_TIMER_IRQ	System Tick Timer
IPU_IRQ_16	N/A	N/A	N/A	XLATE_MMU_FAULT_IRQ	xlate_mmu_fault (from L2 MMU)
IPU_IRQ_17	N/A	N/A	N/A	UNICACHE_MMU_IRQ	Unicache or MMU maintenance complete
IPU_IRQ_18	N/A	N/A	N/A	CTM_TIM_EVENT1_IRQ	CTM timer event (timer #1)
IPU_IRQ_19	N/A	N/A	N/A	HWSEM_M4_IRQ	Semaphore interrupt (1 to each core)
IPU_IRQ_20	N/A	N/A	N/A	ICE_NEMU_IRQ	ICECrusher (1 to each core)
IPU_IRQ_21	N/A	N/A	N/A	IPU_IMP_FAULT_IRQ	IPU imprecise fault (from interconnect)
IPU_IRQ_22	N/A	N/A	N/A	CTM_TIM_EVENT2_IRQ	CTM timer event (timer #2)
IPU_IRQ_23	1	CTRL_CORE_IPU_IRQ_23_24[8:0]	20	DISPC_IRQ	Display controller interrupt
IPU_IRQ_24	2	CTRL_CORE_IPU_IRQ_23_24[24:16]	48	Reserved	Reserved by default but can be remapped to a valid interrupt source
IPU_IRQ_25	3	CTRL_CORE_IPU_IRQ_25_26[8:0]	0	Reserved	Reserved by default but can be remapped to a valid interrupt source
IPU_IRQ_26	4	CTRL_CORE_IPU_IRQ_25_26[24:16]	96	Reserved	Reserved by default but can be remapped to a valid interrupt source
IPU_IRQ_27	5	CTRL_CORE_IPU_IRQ_27_28[8:0]	126	ISS_IRQ_INT0	ISS interrupt 0[2]

(2) ISS is not supported on the DRA78x family of devices.

Table 12-4. IPU_Cx_INTC Default Interrupt Mapping (continued)

IRQ Input Line[1]	IRQ_ CROSSBAR Instance Number	IRQ_CROSSBAR Configuration Register	IRQ_ CROSSBAR Default Input Index	Default Interrupt Name	Default Interrupt Source Description
IPU_IRQ_28	6	CTRL_CORE_IPU_IRQ_27_28[24:16]	127	Reserved	Reserved by default but can be remapped to a valid interrupt source
IPU_IRQ_29	7	CTRL_CORE_IPU_IRQ_29_30[8:0]	128	Reserved	Reserved by default but can be remapped to a valid interrupt source
IPU_IRQ_30	8	CTRL_CORE_IPU_IRQ_29_30[24:16]	129	Reserved	Reserved by default but can be remapped to a valid interrupt source
IPU_IRQ_31	9	CTRL_CORE_IPU_IRQ_31_32[8:0]	130	Reserved	Reserved by default but can be remapped to a valid interrupt source
IPU_IRQ_32	10	CTRL_CORE_IPU_IRQ_31_32[24:16]	19	Reserved	Reserved by default but can be remapped to a valid interrupt source
IPU_IRQ_33	11	CTRL_CORE_IPU_IRQ_33_34[8:0]	131	Reserved	Reserved by default but can be remapped to a valid interrupt source
IPU_IRQ_34	12	CTRL_CORE_IPU_IRQ_33_34[24:16]	7	Reserved	Reserved by default but can be remapped to a valid interrupt source
IPU_IRQ_35	13	CTRL_CORE_IPU_IRQ_35_36[8:0]	8	Reserved	Reserved by default but can be remapped to a valid interrupt source
IPU_IRQ_36	14	CTRL_CORE_IPU_IRQ_35_36[24:16]	9	Reserved	Reserved by default but can be remapped to a valid interrupt source
IPU_IRQ_37	15	CTRL_CORE_IPU_IRQ_37_38[8:0]	10	Reserved	Reserved by default but can be remapped to a valid interrupt source
IPU_IRQ_38	16	CTRL_CORE_IPU_IRQ_37_38[24:16]	132	Reserved	Reserved by default but can be remapped to a valid interrupt source
IPU_IRQ_39	17	CTRL_CORE_IPU_IRQ_39_40[8:0]	98	Reserved	Reserved by default but can be remapped to a valid interrupt source
IPU_IRQ_40	18	CTRL_CORE_IPU_IRQ_39_40[24:16]	99	Reserved	Reserved by default but can be remapped to a valid interrupt source
IPU_IRQ_41	19	CTRL_CORE_IPU_IRQ_41_42[8:0]	51	I2C1_IRQ	I2C1 interrupt
IPU_IRQ_42	20	CTRL_CORE_IPU_IRQ_41_42[24:16]	52	I2C2_IRQ	I2C2 interrupt
IPU_IRQ_43	21	CTRL_CORE_IPU_IRQ_43_44[8:0]	56	Reserved	Reserved by default but can be remapped to a valid interrupt source
IPU_IRQ_44	22	CTRL_CORE_IPU_IRQ_43_44[24:16]	57	Reserved	Reserved by default but can be remapped to a valid interrupt source
IPU_IRQ_45	23	CTRL_CORE_IPU_IRQ_45_46[8:0]	69	UART3_IRQ	UART3 interrupt
IPU_IRQ_46	24	CTRL_CORE_IPU_IRQ_45_46[24:16]	5	L3_MAIN_IRQ_APP_ERR	L3_MAIN application or non-attributable error
IPU_IRQ_47	25	CTRL_CORE_IPU_IRQ_47_48[8:0]	133	PRM_IRQ_IPU	PRCM interrupt to IPU
IPU_IRQ_48	26	CTRL_CORE_IPU_IRQ_47_48[24:16]	14	Reserved	Reserved by default but can be remapped to a valid interrupt source

Table 12-4. IPU_Cx_INTC Default Interrupt Mapping (continued)

IRQ Input Line[1]	IRQ_ CROSSBAR Instance Number	IRQ_CROSSBAR Configuration Register	IRQ_ CROSSBAR Default Input Index	Default Interrupt Name	Default Interrupt Source Description
IPU_IRQ_49	27	CTRL_CORE_IPU_IRQ_49_50[8:0]	66	Reserved	Reserved by default but can be remapped to a valid interrupt source
IPU_IRQ_50	28	CTRL_CORE_IPU_IRQ_49_50[24:16]	134	MAILBOX1_IRQ_USER2	Mailbox 1 user 2 interrupt
IPU_IRQ_51	29	CTRL_CORE_IPU_IRQ_51_52[8:0]	24	GPIO1_IRQ_1	GPIO1 interrupt 1
IPU_IRQ_52	30	CTRL_CORE_IPU_IRQ_51_52[24:16]	25	GPIO2_IRQ_1	GPIO2 interrupt 1
IPU_IRQ_53	31	CTRL_CORE_IPU_IRQ_53_54[8:0]	34	TIMER3_IRQ	TIMER3 interrupt
IPU_IRQ_54	32	CTRL_CORE_IPU_IRQ_53_54[24:16]	35	TIMER4_IRQ	TIMER4 interrupt
IPU_IRQ_55	33	CTRL_CORE_IPU_IRQ_55_56[8:0]	40	Reserved	Reserved by default but can be remapped to a valid interrupt source
IPU_IRQ_56	34	CTRL_CORE_IPU_IRQ_55_56[24:16]	42	Reserved	Reserved by default but can be remapped to a valid interrupt source
IPU_IRQ_57	35	CTRL_CORE_IPU_IRQ_57_58[8:0]	60	MCSPI1_IRQ	McSPI1 interrupt
IPU_IRQ_58	36	CTRL_CORE_IPU_IRQ_57_58[24:16]	61	MCSPI2_IRQ	McSPI2 interrupt
IPU_IRQ_59	37	CTRL_CORE_IPU_IRQ_59_60[8:0]	50	Reserved	Reserved by default but can be remapped to a valid interrupt source
IPU_IRQ_60	38	CTRL_CORE_IPU_IRQ_59_60[24:16]	0	Reserved	Reserved by default but can be remapped to a valid interrupt source
IPU_IRQ_61	39	CTRL_CORE_IPU_IRQ_61_62[8:0]	0	Reserved	Reserved by default but can be remapped to a valid interrupt source
IPU_IRQ_62	40	CTRL_CORE_IPU_IRQ_61_62[24:16]	22	Reserved	Reserved by default but can be remapped to a valid interrupt source
IPU_IRQ_63	41	CTRL_CORE_IPU_IRQ_63_64[8:0]	83	Reserved	Reserved by default but can be remapped to a valid interrupt source
IPU_IRQ_64	42	CTRL_CORE_IPU_IRQ_63_64[24:16]	108	Reserved	Reserved by default but can be remapped to a valid interrupt source
IPU_IRQ_65	43	CTRL_CORE_IPU_IRQ_65_66[8:0]	120	Reserved	Reserved by default but can be remapped to a valid interrupt source
IPU_IRQ_66	44	CTRL_CORE_IPU_IRQ_65_66[24:16]	78	Reserved	Reserved by default but can be remapped to a valid interrupt source
IPU_IRQ_67	45	CTRL_CORE_IPU_IRQ_67_68[8:0]	81	Reserved	Reserved by default but can be remapped to a valid interrupt source
IPU_IRQ_68	46	CTRL_CORE_IPU_IRQ_67_68[24:16]	89	Reserved	Reserved by default but can be remapped to a valid interrupt source
IPU_IRQ_69	47	CTRL_CORE_IPU_IRQ_69_70[8:0]	0	Reserved	Reserved by default but can be remapped to a valid interrupt source

Table 12-4. IPU_Cx_INTC Default Interrupt Mapping (continued)

IRQ Input Line[(1)]	IRQ_ CROSSBAR Instance Number	IRQ_CROSSBAR Configuration Register	IRQ_ CROSSBAR Default Input Index	Default Interrupt Name	Default Interrupt Source Description
IPU_IRQ_70	48	CTRL_CORE_IPU_IRQ_69_70[24:16]	0	Reserved	Reserved by default but can be remapped to a valid interrupt source
IPU_IRQ_71	49	CTRL_CORE_IPU_IRQ_71_72[8:0]	119	Reserved	Reserved by default but can be remapped to a valid interrupt source
IPU_IRQ_72	50	CTRL_CORE_IPU_IRQ_71_72[24:16]	118	Reserved	Reserved by default but can be remapped to a valid interrupt source
IPU_IRQ_73	51	CTRL_CORE_IPU_IRQ_73_74[8:0]	72	Reserved	Reserved by default but can be remapped to a valid interrupt source
IPU_IRQ_74	52	CTRL_CORE_IPU_IRQ_73_74[24:16]	73	Reserved	Reserved by default but can be remapped to a valid interrupt source
IPU_IRQ_75	53	CTRL_CORE_IPU_IRQ_75_76[8:0]	117	Reserved	Reserved by default but can be remapped to a valid interrupt source
IPU_IRQ_76	54	CTRL_CORE_IPU_IRQ_75_76[24:16]	87	Reserved	Reserved by default but can be remapped to a valid interrupt source
IPU_IRQ_77	55	CTRL_CORE_IPU_IRQ_77_78[8:0]	88	Reserved	Reserved by default but can be remapped to a valid interrupt source
IPU_IRQ_78	56	CTRL_CORE_IPU_IRQ_77_78[24:16]	62	Reserved	Reserved by default but can be remapped to a valid interrupt source
IPU_IRQ_79	57	CTRL_CORE_IPU_IRQ_79_80[8:0]	63	Reserved	Reserved by default but can be remapped to a valid interrupt source
IPU_IRQ_80	N/A	N/A	N/A	L1TAG_ECC_SEC_IRQ	SEC interrupt for L1-Tag
IPU_IRQ_81	N/A	N/A	N/A	L1TAG_ECC_DED_IRQ	DED interrupt for L1-Tag
IPU_IRQ_82	N/A	N/A	N/A	L1DATA_ECC_SEC_IRQ	SEC interrupt for L1-Data
IPU_IRQ_83	N/A	N/A	N/A	L1DATA_ECC_DED_IRQ	DED interrupt for L1-Data
IPU_IRQ_84	N/A	N/A	N/A	L2_RAM_ECC_SEC_IRQ	SEC interrupt for L2-RAM
IPU_IRQ_85	N/A	N/A	N/A	L2_RAM_ECC_DED_IRQ	DED interrupt for L2-RAM

NOTE: Exceptions/interrupts IPU_IRQ_[15:0] are all internal to the Cortex-M4 core.

Exceptions/interrupts IPU_IRQ_[79:16] are all external to the Cortex-M4 core – that is, the first Cortex-M4 external interrupt is mapped to IPU_IRQ_16 (exception #16), and the last (sixty-fourth) Cortex-M4 external interrupt is mapped to IPU_IRQ_79 (exception #79).

For more information about Cortex-M4 exception types, refer to Arm *Cortex™-M4 Devices Generic User Guide* (available at http://infocenter.arm.com/help/index.jsp).

NOTE: The "**IRQ_CROSSBAR Default Input Index**" column of Table 12-4 shows which input of the corresponding IRQ_CROSSBAR instance is mapped to its output (and then routed to the corresponding IPU1_Cx_INTC input) by default. In other words, this column specifies the default (reset) values (in decimal) of the CTRL_CORE_IPU1_IRQ_y_z register bit fields that are used to control the mapping of device interrupts to IPU1_Cx_INTC inputs. For example, the IPU1_IRQ_23_24[8:0] bit field is used to configure which device interrupt would be mapped to the IPU1_IRQ_23 line. The reset value of this bit field is 0x14, meaning that DISPC_IRQ would be mapped to IPU1_IRQ_23 by default because it is connected to the IRQ_CROSSBAR_20 input.

'N/A' in this column means that the corresponding interrupt is internal to the IPU1 subsystem. There is no IRQ_CROSSBAR dedicated to the associated IPU1_Cx_INTC input line and therefore, the user cannot change its default mapping.

The CTRL_CORE_IPU1_IRQ_y_z registers control the IRQ_CROSSBAR settings for both NVICs in the IPU1 subsystem. That is, it is not possible to map different interrupts to the same interrupt input of the NVICs in IPU1.

12.3.4 Interrupt Requests to EVE_INTC1

Table 12-5 lists the default interrupt sources for the EVE_INTC1. In addition, device interrupts EVE_IRQ_0 through EVE_IRQ_7 can alternatively be sourced through the EVE's IRQ_CROSSBAR from one of the 460 multiplexed device interrupts listed in Table 12-6. The CTRL_CORE_EVE_IRQ_y_z registers in the Control Module are used to select between the default interrupts and the multiplexed interrupts.

Table 12-5. EVE_INTC1 Default Interrupt Mapping

IRQ Input Line	IRQ_ CROSSBAR Instance Number	IRQ_CROSSBAR Configuration Register	IRQ_ CROSSBAR Default Input Index	Default Interrupt Name	Default Interrupt Source Description
EVE_IRQ_0	1	CTRL_CORE_EVE_IRQ_0_1[8:0]	1	ELM_IRQ	Error location process completion interrupt
EVE_IRQ_1	2	CTRL_CORE_EVE_IRQ_0_1[24:16]	2	EXT_SYS_IRQ_1	External interrupt (active low) via sys_nirq1 pin
EVE_IRQ_2	3	CTRL_CORE_EVE_IRQ_2_3[8:0]	3	CTRL_MODULE_CORE_IRQ_SEC_EVTS	Combined firewall error interrupt. For more information, see Section 13.4.6.13.2.
EVE_IRQ_3	4	CTRL_CORE_EVE_IRQ_2_3[24:16]	4	L3_MAIN_IRQ_DBG_ERR	L3_MAIN debug error
EVE_IRQ_4	5	CTRL_CORE_EVE_IRQ_4_5[8:0]	5	L3_MAIN_IRQ_APP_ERR	L3_MAIN application or non-attributable error
EVE_IRQ_5	6	CTRL_CORE_EVE_IRQ_4_5[24:16]	6	Reserved	Reserved by default but can be remapped to a valid interrupt source
EVE_IRQ_6	7	CTRL_CORE_EVE_IRQ_6_7[8:0]	7	Reserved	Reserved by default but can be remapped to a valid interrupt source
EVE_IRQ_7	8	CTRL_CORE_EVE_IRQ_6_7[24:16]	8	Reserved	Reserved by default but can be remapped to a valid interrupt source
EVE_IRQ_8	N/A	N/A	N/A	Reserved	Reserved
EVE_IRQ_9	N/A	N/A	N/A	Reserved	Reserved
EVE_IRQ_10	N/A	N/A	N/A	Reserved	Reserved
EVE_IRQ_11	N/A	N/A	N/A	Reserved	Reserved
EVE_IRQ_12	N/A	N/A	N/A	Reserved	Reserved
EVE_IRQ_13	N/A	N/A	N/A	Reserved	Reserved
EVE_IRQ_14	N/A	N/A	N/A	Reserved	Reserved

Table 12-5. EVE_INTC1 Default Interrupt Mapping (continued)

IRQ Input Line	IRQ_ CROSSBAR Instance Number	IRQ_CROSSBAR Configuration Register	IRQ_ CROSSBAR Default Input Index	Default Interrupt Name	Default Interrupt Source Description
EVE_IRQ_15	N/A	N/A	N/A	Reserved	Reserved
EVE_IRQ_16	N/A	N/A	N/A	Reserved	Reserved
EVE_IRQ_17	N/A	N/A	N/A	Reserved	Reserved
EVE_IRQ_18	N/A	N/A	N/A	Reserved	Reserved
EVE_IRQ_19	N/A	N/A	N/A	Reserved	Reserved
EVE_IRQ_20	N/A	N/A	N/A	Reserved	Reserved
EVE_IRQ_21	N/A	N/A	N/A	Reserved	Reserved
EVE_IRQ_22	N/A	N/A	N/A	Reserved	Reserved
EVE_IRQ_23	N/A	N/A	N/A	Reserved	Reserved
EVE_IRQ_24	N/A	N/A	N/A	Reserved	Reserved
EVE_IRQ_25	N/A	N/A	N/A	Reserved	Reserved
EVE_IRQ_26	N/A	N/A	N/A	Reserved	Reserved
EVE_IRQ_27	N/A	N/A	N/A	Reserved	Reserved
EVE_IRQ_28	N/A	N/A	N/A	EVE_MBX2_INT0	EVE Mailbox 2 Interrupt 0
EVE_IRQ_29	N/A	N/A	N/A	Reserved	Reserved
EVE_IRQ_30	N/A	N/A	N/A	Reserved	Reserved
EVE_IRQ_31	N/A	N/A	N/A	Reserved	Reserved

NOTE: The "**IRQ_CROSSBAR Default Input Index**" column of Table 12-5 shows which input of the corresponding IRQ_CROSSBAR instance is mapped to its output (and then routed to the corresponding EVE_INTC1 input) by default. In other words, this column specifies the default (reset) values (in decimal) of the CTRL_CORE_EVE1_IRQ_y_z register bit fields that are used to control the mapping of device interrupts to EVE_INTC1 inputs. For example, the EVE1_IRQ_0_1[8:0] bit field is used to configure which device interrupt would be mapped to the EVE1_IRQ_0 line. The reset value of this bit field is 0x1, meaning that ELM_IRQ would be mapped to EVE1_IRQ_0 by default because it is connected to the IRQ_CROSSBAR_1 input.

'N/A' in this column means that the corresponding interrupt is internal to the EVE1 subsystem. There is no IRQ_CROSSBAR dedicated to the associated EVE_INTC1 input line and therefore, the user cannot change its default mapping.

12.3.5 Mapping of Device Interrupts to IRQ_CROSSBAR Inputs

Table 12-6 shows the individual connection between all module IRQs and all IRQ_CROSSBAR inputs.

Table 12-6. Connection of Device IRQs to IRQ_CROSSBAR Inputs

IRQ_CROSSBAR Input	Interrupt Name	Interrupt Source	Description
IRQ_CROSSBAR_0	Reserved	Reserved	Reserved
IRQ_CROSSBAR_1	ELM_IRQ	ELM	Error location process completion interrupt
IRQ_CROSSBAR_2	EXT_SYS_IRQ_1	External system	External interrupt (active low) via sys_nirq1 pin
IRQ_CROSSBAR_3	CTRL_MODULE_CORE_IRQ _SEC_EVTS	CTRL_MODULE_CORE	Combined firewall error interrupt. For more information, see Section 13.4.6.13.2
IRQ_CROSSBAR_4	L3_MAIN_IRQ_DBG_ERR	L3_MAIN	L3_MAIN debug error
IRQ_CROSSBAR_5	L3_MAIN_IRQ_APP_ERR	L3_MAIN	L3_MAIN application or non-attributable error
IRQ_CROSSBAR_6	Reserved	Reserved	Reserved
IRQ_CROSSBAR_7	Reserved	Reserved	Reserved
IRQ_CROSSBAR_8	Reserved	Reserved	Reserved
IRQ_CROSSBAR_9	Reserved	Reserved	Reserved
IRQ_CROSSBAR_10	Reserved	Reserved	Reserved
IRQ_CROSSBAR_11	L3_MAIN_IRQ_STAT_ALARM	L3_MAIN	L3_MAIN statistic collector alarm interrupt
IRQ_CROSSBAR_12	Reserved	Reserved	Reserved
IRQ_CROSSBAR_13	Reserved	Reserved	Reserved
IRQ_CROSSBAR_14	Reserved	Reserved	Reserved
IRQ_CROSSBAR_15	GPMC_IRQ	GPMC	GPMC interrupt
IRQ_CROSSBAR_16	Reserved	Reserved	Reserved
IRQ_CROSSBAR_17	Reserved	Reserved	Reserved
IRQ_CROSSBAR_18	Reserved	Reserved	Reserved
IRQ_CROSSBAR_19	Reserved	Reserved	Reserved
IRQ_CROSSBAR_20	DISPC_IRQ	DISPC	Display controller interrupt
IRQ_CROSSBAR_21	MAILBOX1_IRQ_USER0	MAILBOX1	Mailbox 1 user 0 interrupt
IRQ_CROSSBAR_22	Reserved	Reserved	Reserved
IRQ_CROSSBAR_23	DSP1_IRQ_MMU0	DSP1	DSP1 MMU0 interrupt
IRQ_CROSSBAR_24	GPIO1_IRQ_1	GPIO1	GPIO1 interrupt 1
IRQ_CROSSBAR_25	GPIO2_IRQ_1	GPIO2	GPIO2 interrupt 1
IRQ_CROSSBAR_26	GPIO3_IRQ_1	GPIO3	GPIO3 interrupt 1
IRQ_CROSSBAR_27	GPIO4_IRQ_1	GPIO4	GPIO4 interrupt 1
IRQ_CROSSBAR_28	Reserved	Reserved	Reserved
IRQ_CROSSBAR_29	Reserved	Reserved	Reserved
IRQ_CROSSBAR_30	Reserved	Reserved	Reserved
IRQ_CROSSBAR_31	Reserved	Reserved	Reserved
IRQ_CROSSBAR_32	TIMER1_IRQ	TIMER1	TIMER1 interrupt
IRQ_CROSSBAR_33	TIMER2_IRQ	TIMER2	TIMER2 interrupt
IRQ_CROSSBAR_34	TIMER3_IRQ	TIMER3	TIMER3 interrupt
IRQ_CROSSBAR_35	TIMER4_IRQ	TIMER4	TIMER4 interrupt
IRQ_CROSSBAR_36	TIMER5_IRQ	TIMER5	TIMER5 interrupt
IRQ_CROSSBAR_37	TIMER6_IRQ	TIMER6	TIMER6 interrupt
IRQ_CROSSBAR_38	TIMER7_IRQ	TIMER7	TIMER7 interrupt
IRQ_CROSSBAR_39	TIMER8_IRQ	TIMER8	TIMER8 interrupt
IRQ_CROSSBAR_40	Reserved	Reserved	Reserved
IRQ_CROSSBAR_41	Reserved	Reserved	Reserved
IRQ_CROSSBAR_42	Reserved	Reserved	Reserved
IRQ_CROSSBAR_43	MCSPI4_IRQ	McSPI4	McSPI4 interrupt
IRQ_CROSSBAR_44	Reserved	Reserved	Reserved
IRQ_CROSSBAR_45	Reserved	Reserved	Reserved
IRQ_CROSSBAR_46	Reserved	Reserved	Reserved

Table 12-6. Connection of Device IRQs to IRQ_CROSSBAR Inputs (continued)

IRQ_CROSSBAR Input	Interrupt Name	Interrupt Source	Description
IRQ_CROSSBAR_47	Reserved	Reserved	Reserved
IRQ_CROSSBAR_48	Reserved	Reserved	Reserved
IRQ_CROSSBAR_49	Reserved	Reserved	Reserved
IRQ_CROSSBAR_50	Reserved	Reserved	Reserved
IRQ_CROSSBAR_51	I2C1_IRQ	I2C1	I2C1 interrupt
IRQ_CROSSBAR_52	I2C2_IRQ	I2C2	I2C2 interrupt
IRQ_CROSSBAR_53	Reserved	Reserved	Reserved
IRQ_CROSSBAR_54	Reserved	Reserved	Reserved
IRQ_CROSSBAR_55	Reserved	Reserved	Reserved
IRQ_CROSSBAR_56	Reserved	Reserved	Reserved
IRQ_CROSSBAR_57	Reserved	Reserved	Reserved
IRQ_CROSSBAR_58	Reserved	Reserved	Reserved
IRQ_CROSSBAR_59	Reserved	Reserved	Reserved
IRQ_CROSSBAR_60	MCSPI1_IRQ	McSPI1	McSPI1 interrupt
IRQ_CROSSBAR_61	MCSPI2_IRQ	McSPI2	McSPI2 interrupt
IRQ_CROSSBAR_62	Reserved	Reserved	Reserved
IRQ_CROSSBAR_63	Reserved	Reserved	Reserved
IRQ_CROSSBAR_64	Reserved	Reserved	Reserved
IRQ_CROSSBAR_65	Reserved	Reserved	Reserved
IRQ_CROSSBAR_66	Reserved	Reserved	Reserved
IRQ_CROSSBAR_67	UART1_IRQ	UART1	UART1 interrupt
IRQ_CROSSBAR_68	UART2_IRQ	UART2	UART2 interrupt
IRQ_CROSSBAR_69	UART3_IRQ	UART3	UART3 interrupt
IRQ_CROSSBAR_70	Reserved	Reserved	Reserved
IRQ_CROSSBAR_71	Reserved	Reserved	Reserved
IRQ_CROSSBAR_72	Reserved	Reserved	Reserved
IRQ_CROSSBAR_73	Reserved	Reserved	Reserved
IRQ_CROSSBAR_74	Reserved	Reserved	Reserved
IRQ_CROSSBAR_75	RTI2_IRQ_WWD	RTI2	RTI2 WWD interrupt
IRQ_CROSSBAR_76	Reserved	Reserved	Reserved
IRQ_CROSSBAR_77	Reserved	Reserved	Reserved
IRQ_CROSSBAR_78	Reserved	Reserved	Reserved
IRQ_CROSSBAR_79	Reserved	Reserved	Reserved
IRQ_CROSSBAR_80	Reserved	Reserved	Reserved
IRQ_CROSSBAR_81	Reserved	Reserved	Reserved
IRQ_CROSSBAR_82	Reserved	Reserved	Reserved
IRQ_CROSSBAR_83	Reserved	Reserved	Reserved
IRQ_CROSSBAR_84	Reserved	Reserved	Reserved
IRQ_CROSSBAR_85	DEBUGSS_IRQ_CT_UART	DEBUGSS	CT_UART interrupt generated when data ready on RX or when TX empty
IRQ_CROSSBAR_86	MCSPI3_IRQ	McSPI3	McSPI3 interrupt
IRQ_CROSSBAR_87	Reserved	Reserved	Reserved
IRQ_CROSSBAR_88	Reserved	Reserved	Reserved
IRQ_CROSSBAR_89	Reserved	Reserved	Reserved
IRQ_CROSSBAR_90	Reserved	Reserved	Reserved
IRQ_CROSSBAR_91	MMC_IRQ	MMC4	MMC interrupt
IRQ_CROSSBAR_92	Reserved	Reserved	Reserved
IRQ_CROSSBAR_93	Reserved	Reserved	Reserved

Table 12-6. Connection of Device IRQs to IRQ_CROSSBAR Inputs (continued)

IRQ_CROSSBAR Input	Interrupt Name	Interrupt Source	Description
IRQ_CROSSBAR_94	Reserved	Reserved	Reserved
IRQ_CROSSBAR_95	Reserved	Reserved	Reserved
IRQ_CROSSBAR_96	Reserved	Reserved	Reserved
IRQ_CROSSBAR_97	Reserved	Reserved	Reserved
IRQ_CROSSBAR_98	Reserved	Reserved	Reserved
IRQ_CROSSBAR_99	Reserved	Reserved	Reserved
IRQ_CROSSBAR_100	Reserved	Reserved	Reserved
IRQ_CROSSBAR_101	Reserved	Reserved	Reserved
IRQ_CROSSBAR_102	Reserved	Reserved	Reserved
IRQ_CROSSBAR_103	MCASP1_IRQ_AREVT	McASP1	McASP1 receive interrupt
IRQ_CROSSBAR_104	MCASP1_IRQ_AXEVT	McASP1	McASP1 transmit interrupt
IRQ_CROSSBAR_105	EMIF_IRQ	EMIF1	EMIF interrupt
IRQ_CROSSBAR_106	Reserved	Reserved	Reserved
IRQ_CROSSBAR_107	Reserved	Reserved	Reserved
IRQ_CROSSBAR_108	Reserved	Reserved	Reserved
IRQ_CROSSBAR_109	Reserved	Reserved	Reserved
IRQ_CROSSBAR_110	Reserved	Reserved	Reserved
IRQ_CROSSBAR_111	Reserved	Reserved	Reserved
IRQ_CROSSBAR_112	Reserved	Reserved	Reserved
IRQ_CROSSBAR_113	Reserved	Reserved	Reserved
IRQ_CROSSBAR_114	EXT_SYS_IRQ_2	External system	External interrupt (active low) via sys_nirq2 pin
IRQ_CROSSBAR_115	Reserved	Reserved	Reserved
IRQ_CROSSBAR_116	Reserved	Reserved	Reserved
IRQ_CROSSBAR_117	Reserved	Reserved	Reserved
IRQ_CROSSBAR_118	Reserved	Reserved	Reserved
IRQ_CROSSBAR_119	Reserved	Reserved	Reserved
IRQ_CROSSBAR_120	Reserved	Reserved	Reserved
IRQ_CROSSBAR_121	CTRL_MODULE_CORE_IRQ _THERMAL_ALERT	CTRL_MODULE	CTRL_MODULE thermal alert interrupt
IRQ_CROSSBAR_122	Reserved	Reserved	Reserved
IRQ_CROSSBAR_123	Reserved	Reserved	Reserved
IRQ_CROSSBAR_124	Reserved	Reserved	Reserved
IRQ_CROSSBAR_125	Reserved	Reserved	Reserved
IRQ_CROSSBAR_126	ISS_IRQ_INT0	ISS	ISS interrupt 0[(1)]
IRQ_CROSSBAR_127	Reserved	Reserved	Reserved
IRQ_CROSSBAR_128	Reserved	Reserved	Reserved
IRQ_CROSSBAR_129	Reserved	Reserved	Reserved
IRQ_CROSSBAR_130	Reserved	Reserved	Reserved
IRQ_CROSSBAR_131	Reserved	Reserved	Reserved
IRQ_CROSSBAR_132	Reserved	Reserved	Reserved
IRQ_CROSSBAR_133	PRM_IRQ_IPU	PRM	PRCM interrupt to IPU
IRQ_CROSSBAR_134	MAILBOX1_IRQ_USER2	MAILBOX1	Mailbox 1 user 2 interrupt
IRQ_CROSSBAR_135	MAILBOX1_IRQ_USER1	MAILBOX1	Mailbox 1 user 1 interrupt
IRQ_CROSSBAR_136	Reserved	Reserved	Reserved
IRQ_CROSSBAR_137	PRM_IRQ_DSP1	PRM	PRCM interrupt to DSP1
IRQ_CROSSBAR_138	GPIO1_IRQ_2	GPIO1	GPIO1 interrupt 2
IRQ_CROSSBAR_139	GPIO2_IRQ_2	GPIO2	GPIO2 interrupt 2

(1) DCC, CRC, TESOC, ESM and ISS are not supported on the DRA78x family of devices.

Table 12-6. Connection of Device IRQs to IRQ_CROSSBAR Inputs (continued)

IRQ_CROSSBAR Input	Interrupt Name	Interrupt Source	Description
IRQ_CROSSBAR_140	GPIO3_IRQ_2	GPIO3	GPIO3 interrupt 2
IRQ_CROSSBAR_141	GPIO4_IRQ_2	GPIO4	GPIO4 interrupt 2
IRQ_CROSSBAR_142	Reserved	Reserved	Reserved
IRQ_CROSSBAR_143	Reserved	Reserved	Reserved
IRQ_CROSSBAR_144	Reserved	Reserved	Reserved
IRQ_CROSSBAR_145	DSP1_IRQ_MMU1	DSP1	DSP1 MMU1 interrupt
IRQ_CROSSBAR_146	DSP2_IRQ_MMU0	DSP2	DSP2 MMU0 interrupt
IRQ_CROSSBAR_147	DSP2_IRQ_MMU1	DSP2	DSP2 MMU1 interrupt
IRQ_CROSSBAR_148 to IRQ_CROSSBAR_163	Reserved	Reserved	Reserved
IRQ_CROSSBAR_164	OCMC_RAM_IRQ	OCMC_RAM	OCMC_RAM interrupt
IRQ_CROSSBAR_165	Reserved	Reserved	Reserved
IRQ_CROSSBAR_166	Reserved	Reserved	Reserved
IRQ_CROSSBAR_167	CRC_IRQ_CH1_CPR_COMP	CRC	CRC interrupt(1)
IRQ_CROSSBAR_168	EVE_IRQ_OUT0	EVE	EVE output interrupt 0
IRQ_CROSSBAR_169	EVE_IRQ_OUT1	EVE	EVE output interrupt 1
IRQ_CROSSBAR_170	EVE_IRQ_OUT2	EVE	EVE output interrupt 2
IRQ_CROSSBAR_171	EVE_IRQ_OUT3	EVE	EVE output interrupt 3
IRQ_CROSSBAR_172 to IRQ_CROSSBAR_183	Reserved	Reserved	Reserved
IRQ_CROSSBAR_184	SD_DAC_IRQ_TVINT	SD_DAC	SD DAC TV interrupt
IRQ_CROSSBAR_185	SD_DAC_IRQ_TVSHORT	SD_DAC	SD DAC TV short interrupt
IRQ_CROSSBAR_186 to IRQ_CROSSBAR_193	Reserved	Reserved	Reserved
IRQ_CROSSBAR_194	ISS_IRQ_INT1	ISS	ISS interrupt 1(1)
IRQ_CROSSBAR_195	ISS_IRQ_INT2	ISS	ISS interrupt 2(1)
IRQ_CROSSBAR_196 to IRQ_CROSSBAR_203	Reserved	Reserved	Reserved
IRQ_CROSSBAR_204	PWMSS1_IRQ_ePWM0_TZINT	PWMSS1	eHRPWM0 TZ interrupt
IRQ_CROSSBAR_205	Reserved	Reserved	Reserved
IRQ_CROSSBAR_206	Reserved	Reserved	Reserved
IRQ_CROSSBAR_207	PWMSS1_IRQ_ePWM0INT	PWMSS1	eHRPWM0 event/interrupt
IRQ_CROSSBAR_208	Reserved	Reserved	Reserved
IRQ_CROSSBAR_209	Reserved	Reserved	Reserved
IRQ_CROSSBAR_210	PWMSS1_IRQ_eQEP0INT	PWMSS1	eQEP0 event/interrupt
IRQ_CROSSBAR_211	Reserved	Reserved	Reserved
IRQ_CROSSBAR_212	Reserved	Reserved	Reserved
IRQ_CROSSBAR_213	PWMSS1_IRQ_eCAP0INT	PWMSS1	eCAP0 event/interrupt
IRQ_CROSSBAR_214	Reserved	Reserved	Reserved
IRQ_CROSSBAR_215	Reserved	Reserved	Reserved
IRQ_CROSSBAR_216	ISS_IRQ_INT3	ISS	ISS interrupt 3(1)
IRQ_CROSSBAR_217	Reserved	Reserved	Reserved
IRQ_CROSSBAR_218	Reserved	Reserved	Reserved
IRQ_CROSSBAR_219	Reserved	Reserved	Reserved
IRQ_CROSSBAR_220	Reserved	Reserved	Reserved
IRQ_CROSSBAR_221	Reserved	Reserved	Reserved
IRQ_CROSSBAR_222	DCAN_IRQ_INT0	DCAN	DCAN interrupt 0
IRQ_CROSSBAR_223	DCAN_IRQ_INT1	DCAN	DCAN interrupt 1
IRQ_CROSSBAR_224	DCAN_IRQ_PARITY	DCAN	DCAN parity interrupt

Table 12-6. Connection of Device IRQs to IRQ_CROSSBAR Inputs (continued)

IRQ_CROSSBAR Input	Interrupt Name	Interrupt Source	Description
IRQ_CROSSBAR_225	MCAN_IRQ_INT0	MCAN	MCAN interrupt 0
IRQ_CROSSBAR_226	MCAN_IRQ_INT1	MCAN	MCAN interrupt 1
IRQ_CROSSBAR_227	MCAN_IRQ_ECC	MCAN	MCAN ECC interrupt
IRQ_CROSSBAR_228	Reserved	Reserved	Reserved
IRQ_CROSSBAR_229	Reserved	Reserved	Reserved
IRQ_CROSSBAR_230	Reserved	Reserved	Reserved
IRQ_CROSSBAR_231	Reserved	Reserved	Reserved
IRQ_CROSSBAR_232	Reserved	Reserved	Reserved
IRQ_CROSSBAR_233	Reserved	Reserved	Reserved
IRQ_CROSSBAR_234	DCC1_IRQ_ERROR	DCC1	DCC error interrupt(1)
IRQ_CROSSBAR_235	DCC1_IRQ_DONE	DCC1	DCC single-shot mode done interrupt(1)
IRQ_CROSSBAR_236	TSC_ADC_IRQ_GENINT	TSC_ADC	TSC ADC interrupt
IRQ_CROSSBAR_237	MAILBOX2_IRQ_USER0	MAILBOX2	Mailbox 2 user 0 interrupt
IRQ_CROSSBAR_238	MAILBOX2_IRQ_USER1	MAILBOX2	Mailbox 2 user 1 interrupt
IRQ_CROSSBAR_239	MAILBOX2_IRQ_USER2	MAILBOX2	Mailbox 2 user 2 interrupt
IRQ_CROSSBAR_240	MAILBOX2_IRQ_USER3	MAILBOX2	Mailbox 2 user 3 interrupt
IRQ_CROSSBAR_241 to IRQ_CROSSBAR_280	Reserved	Reserved	Reserved
IRQ_CROSSBAR_281	EVE_IRQ_TPCC_REGION1	EVE	EVE TPCC region 1 interrupt
IRQ_CROSSBAR_282	EVE_IRQ_TPCC_REGION2	EVE	EVE TPCC region 2 interrupt
IRQ_CROSSBAR_283	EVE_IRQ_TPCC_REGION3	EVE	EVE TPCC region 3 interrupt
IRQ_CROSSBAR_284	EVE_IRQ_MBX0_USER1	EVE	EVE mailbox 0 user 1 interrupt
IRQ_CROSSBAR_285	EVE_IRQ_MBX0_USER2	EVE	EVE mailbox 0 user 2 interrupt
IRQ_CROSSBAR_286	EVE_IRQ_MBX0_USER3	EVE	EVE mailbox 0 user 3 interrupt
IRQ_CROSSBAR_287	EVE_IRQ_MBX1_USER1	EVE	EVE mailbox 1 user 1 interrupt
IRQ_CROSSBAR_288	EVE_IRQ_MBX1_USER2	EVE	EVE mailbox 1 user 2 interrupt
IRQ_CROSSBAR_289	EVE_IRQ_MBX1_USER3	EVE	EVE mailbox 1 user 3 interrupt
IRQ_CROSSBAR_290 to IRQ_CROSSBAR_316	Reserved	Reserved	Reserved
IRQ_CROSSBAR_317	DSP1_IRQ_TPCC_ERR	DSP1	DSP1 TPCC error interrupt
IRQ_CROSSBAR_318	DSP1_IRQ_TPCC_GLOBAL	DSP1	DSP1 TPCC global interrupt
IRQ_CROSSBAR_319	DSP1_IRQ_TPCC_REGION0	DSP1	DSP1 TPCC region 0 interrupt
IRQ_CROSSBAR_320	DSP1_IRQ_TPCC_REGION1	DSP1	DSP1 TPCC region 1 interrupt
IRQ_CROSSBAR_321	DSP1_IRQ_TPCC_REGION2	DSP1	DSP1 TPCC region 2 interrupt
IRQ_CROSSBAR_322	DSP1_IRQ_TPCC_REGION3	DSP1	DSP1 TPCC region 3 interrupt
IRQ_CROSSBAR_323	DSP1_IRQ_TPCC_REGION4	DSP1	DSP1 TPCC region 4 interrupt
IRQ_CROSSBAR_324	DSP1_IRQ_TPCC_REGION5	DSP1	DSP1 TPCC region 5 interrupt
IRQ_CROSSBAR_325	DSP2_IRQ_TPCC_ERR	DSP2	DSP2 TPCC error interrupt
IRQ_CROSSBAR_326	DSP2_IRQ_TPCC_GLOBAL	DSP2	DSP2 TPCC global interrupt
IRQ_CROSSBAR_327	DSP2_IRQ_TPCC_REGION0	DSP2	DSP2 TPCC region 0 interrupt
IRQ_CROSSBAR_328	DSP2_IRQ_TPCC_REGION1	DSP2	DSP2 TPCC region 1 interrupt
IRQ_CROSSBAR_329	DSP2_IRQ_TPCC_REGION2	DSP2	DSP2 TPCC region 2 interrupt
IRQ_CROSSBAR_330	DSP2_IRQ_TPCC_REGION3	DSP2	DSP2 TPCC region 3 interrupt
IRQ_CROSSBAR_331	DSP2_IRQ_TPCC_REGION4	DSP2	DSP2 TPCC region 4 interrupt
IRQ_CROSSBAR_332	DSP2_IRQ_TPCC_REGION5	DSP2	DSP2 TPCC region 5 interrupt
IRQ_CROSSBAR_333	MMU_IRQ	MMU	Top level MMU interrupt
IRQ_CROSSBAR_334	GMAC_SW_IRQ_RX_THRESH_PULSE	GMAC_SW	GMAC_SW receive threshold interrupt

Table 12-6. Connection of Device IRQs to IRQ_CROSSBAR Inputs (continued)

IRQ_CROSSBAR Input	Interrupt Name	Interrupt Source	Description
IRQ_CROSSBAR_335	GMAC_SW_IRQ_RX_PULSE	GMAC_SW	GMAC_SW receive interrupt
IRQ_CROSSBAR_336	GMAC_SW_IRQ_TX_PULSE	GMAC_SW	GMAC_SW transmit interrupt
IRQ_CROSSBAR_337	GMAC_SW_IRQ_MISC_PULSE	GMAC_SW	GMAC_SW miscellaneous interrupt
IRQ_CROSSBAR_338	RTI1_IRQ_WWD	RTI1	RTI1 WWD interrupt
IRQ_CROSSBAR_339	Reserved	Reserved	Reserved
IRQ_CROSSBAR_340	Reserved	Reserved	Reserved
IRQ_CROSSBAR_341	Reserved	Reserved	Reserved
IRQ_CROSSBAR_342	Reserved	Reserved	Reserved
IRQ_CROSSBAR_343	QSPI_IRQ	QSPI	QSPI interrupt
IRQ_CROSSBAR_344	Reserved	Reserved	Reserved
IRQ_CROSSBAR_345	Reserved	Reserved	Reserved
IRQ_CROSSBAR_346	Reserved	Reserved	Reserved
IRQ_CROSSBAR_347	Reserved	Reserved	Reserved
IRQ_CROSSBAR_348	Reserved	Reserved	Reserved
IRQ_CROSSBAR_349	ISS_IRQ_INT4	ISS	ISS interrupt 4[(1)]
IRQ_CROSSBAR_350	ISS_IRQ_INT5	ISS	ISS interrupt 5[(1)]
IRQ_CROSSBAR_351	VIP_IRQ_1	VIP	VIP interrupt 1
IRQ_CROSSBAR_352	Reserved	Reserved	Reserved
IRQ_CROSSBAR_353	Reserved	Reserved	Reserved
IRQ_CROSSBAR_354	Reserved	Reserved	Reserved
IRQ_CROSSBAR_355	Reserved	Reserved	Reserved
IRQ_CROSSBAR_356	Reserved	Reserved	Reserved
IRQ_CROSSBAR_357	ESM_IRQ_HIGH	ESM	ESM high-level interrupt[(1)]
IRQ_CROSSBAR_358	ESM_IRQ_LOW	ESM	ESM low-level interrupt[(1)]
IRQ_CROSSBAR_359	EDMA_TPCC_IRQ_ERR	EDMA TPCC	EDMA TPCC error interrupt
IRQ_CROSSBAR_360	EDMA_TPCC_IRQ_MP	EDMA TPCC	EDMA TPCC memory protection interrupt
IRQ_CROSSBAR_361	EDMA_TPCC_IRQ_REGION0	EDMA TPCC	EDMA TPCC region 0 interrupt
IRQ_CROSSBAR_362	EDMA_TPCC_IRQ_REGION1	EDMA TPCC	EDMA TPCC region 1 interrupt
IRQ_CROSSBAR_363	EDMA_TPCC_IRQ_REGION2	EDMA TPCC	EDMA TPCC region 2 interrupt
IRQ_CROSSBAR_364	EDMA_TPCC_IRQ_REGION3	EDMA TPCC	EDMA TPCC region 3 interrupt
IRQ_CROSSBAR_365	EDMA_TPCC_IRQ_REGION4	EDMA TPCC	EDMA TPCC region 4 interrupt
IRQ_CROSSBAR_366	EDMA_TPCC_IRQ_REGION5	EDMA TPCC	EDMA TPCC region 5 interrupt
IRQ_CROSSBAR_367	EDMA_TPCC_IRQ_REGION6	EDMA TPCC	EDMA TPCC region 6 interrupt
IRQ_CROSSBAR_368	EDMA_TPCC_IRQ_REGION7	EDMA TPCC	EDMA TPCC region 7 interrupt
IRQ_CROSSBAR_369	Reserved	Reserved	Reserved
IRQ_CROSSBAR_370	EDMA_TC0_IRQ_ERR	EDMA TC0	EDMA TPTC0 error interrupt
IRQ_CROSSBAR_371	EDMA_TC1_IRQ_ERR	EDMA TC1	EDMA TPTC1 error interrupt
IRQ_CROSSBAR_372	OCMC_RAM_IRQ_CBUF	OCMC_RAM	OCMC_RAM CBUF interrupt
IRQ_CROSSBAR_373	Reserved	Reserved	Reserved
IRQ_CROSSBAR_374	Reserved	Reserved	Reserved
IRQ_CROSSBAR_375	DSP1_IRQ_TPCC_REGION6	DSP1	DSP1 TPCC region 6 interrupt
IRQ_CROSSBAR_376	DSP1_IRQ_TPCC_REGION7	DSP1	DSP1 TPCC region 7 interrupt
IRQ_CROSSBAR_377	DSP2_IRQ_TPCC_REGION6	DSP2	DSP2 TPCC region 6 interrupt
IRQ_CROSSBAR_378	DSP2_IRQ_TPCC_REGION7	DSP2	DSP2 TPCC region 7 interrupt
IRQ_CROSSBAR_379	Reserved	Reserved	Reserved
IRQ_CROSSBAR_380	Reserved	Reserved	Reserved
IRQ_CROSSBAR_381	Reserved	Reserved	Reserved

Table 12-6. Connection of Device IRQs to IRQ_CROSSBAR Inputs (continued)

IRQ_CROSSBAR Input	Interrupt Name	Interrupt Source	Description
IRQ_CROSSBAR_382	Reserved	Reserved	Reserved
IRQ_CROSSBAR_383	Reserved	Reserved	Reserved
IRQ_CROSSBAR_384	Reserved	Reserved	Reserved
IRQ_CROSSBAR_385	Reserved	Reserved	Reserved
IRQ_CROSSBAR_386	Reserved	Reserved	Reserved
IRQ_CROSSBAR_387	PRM_IRQ_DSP2	PRM	PRCM interrupt to DSP2
IRQ_CROSSBAR_388	Reserved	Reserved	Reserved
IRQ_CROSSBAR_389	Reserved	Reserved	Reserved
IRQ_CROSSBAR_390	Reserved	Reserved	Reserved
IRQ_CROSSBAR_391	Reserved	Reserved	Reserved
IRQ_CROSSBAR_392	VIP_IRQ_2	VIP	VIP interrupt 2
IRQ_CROSSBAR_393	Reserved	Reserved	Reserved
IRQ_CROSSBAR_394	Reserved	Reserved	Reserved
IRQ_CROSSBAR_395	IPU_IRQ_MMU	IPU	IPU MMU interrupt
IRQ_CROSSBAR_396	Reserved	Reserved	Reserved
IRQ_CROSSBAR_397	Reserved	Reserved	Reserved
IRQ_CROSSBAR_398	EVE_IRQ_TPCC_REGION4	EVE	EVE TPCC region 4 interrupt
IRQ_CROSSBAR_399	Reserved	Reserved	Reserved
IRQ_CROSSBAR_400	Reserved	Reserved	Reserved
IRQ_CROSSBAR_401	Reserved	Reserved	Reserved
IRQ_CROSSBAR_402	RTI1_IRQ_INT0	RTI1	RTI1 VIM request 0
IRQ_CROSSBAR_403	RTI1_IRQ_INT1	RTI1	RTI1 VIM request 1
IRQ_CROSSBAR_404	TESOC_IRQ_DONE	TESOC	TESOC done interrupt(1)
IRQ_CROSSBAR_405	RTI3_IRQ_WWD	RTI3	RTI3 WWD interrupt
IRQ_CROSSBAR_406	RTI4_IRQ_WWD	RTI4	RTI4 WWD interrupt
IRQ_CROSSBAR_407	RTI5_IRQ_WWD	RTI5	RTI5 WWD interrupt
IRQ_CROSSBAR_408	RTI1_IRQ_INT2	RTI1	RTI1 VIM request 2
IRQ_CROSSBAR_409	RTI1_IRQ_INT3	RTI1	RTI1 VIM request 3
IRQ_CROSSBAR_410	RTI1_IRQ_OVL0	RTI1	RTI1 overflow interrupt 0
IRQ_CROSSBAR_411	RTI1_IRQ_OVL1	RTI1	RTI1 overflow interrupt 1
IRQ_CROSSBAR_412	RTI2_IRQ_INT0	RTI2	RTI2 VIM request 0
IRQ_CROSSBAR_413	DCC2_IRQ_ERROR	DCC2	DCC2 error interrupt(1)
IRQ_CROSSBAR_414	DCC2_IRQ_DONE	DCC2	DCC2 single-shot mode done interrupt(1)
IRQ_CROSSBAR_415	DCC3_IRQ_ERROR	DCC3	DCC3 error interrupt(1)
IRQ_CROSSBAR_416	DCC3_IRQ_DONE	DCC3	DCC3 single-shot mode done interrupt(1)
IRQ_CROSSBAR_417	DCC4_IRQ_ERROR	DCC4	DCC4 error interrupt(1)
IRQ_CROSSBAR_418	DCC4_IRQ_DONE	DCC4	DCC4 single-shot mode done interrupt(1)
IRQ_CROSSBAR_419	DCC5_IRQ_ERROR	DCC5	DCC5 error interrupt(1)
IRQ_CROSSBAR_420	DCC5_IRQ_DONE	DCC5	DCC5 single-shot mode done interrupt(1)
IRQ_CROSSBAR_421	DCC6_IRQ_ERROR	DCC6	DCC6 error interrupt(1)
IRQ_CROSSBAR_422	DCC6_IRQ_DONE	DCC6	DCC6 single-shot mode done interrupt(1)
IRQ_CROSSBAR_423	DCC7_IRQ_ERROR	DCC7	DCC7 error interrupt(1)
IRQ_CROSSBAR_424	DCC7_IRQ_DONE	DCC7	DCC7 single-shot mode done interrupt(1)
IRQ_CROSSBAR_425	RTI2_IRQ_INT1	RTI2	RTI2 VIM request 1
IRQ_CROSSBAR_426	RTI2_IRQ_INT2	RTI2	RTI2 VIM request 2
IRQ_CROSSBAR_427	RTI2_IRQ_INT3	RTI2	RTI2 VIM request 3
IRQ_CROSSBAR_428	RTI2_IRQ_OVL0	RTI2	RTI2 overflow interrupt 0

Table 12-6. Connection of Device IRQs to IRQ_CROSSBAR Inputs (continued)

IRQ_CROSSBAR Input	Interrupt Name	Interrupt Source	Description
IRQ_CROSSBAR_429	RTI2_IRQ_OVL1	RTI2	RTI2 overflow interrupt 1
IRQ_CROSSBAR_430	RTI3_IRQ_INT0	RTI3	RTI3 VIM request 0
IRQ_CROSSBAR_431	RTI3_IRQ_INT1	RTI3	RTI3 VIM request 1
IRQ_CROSSBAR_432	RTI3_IRQ_INT2	RTI3	RTI3 VIM request 2
IRQ_CROSSBAR_433	RTI3_IRQ_INT3	RTI3	RTI3 VIM request 3
IRQ_CROSSBAR_434	RTI3_IRQ_OVL0	RTI3	RTI3 overflow interrupt 0
IRQ_CROSSBAR_435	RTI3_IRQ_OVL1	RTI3	RTI3 overflow interrupt 1
IRQ_CROSSBAR_436	RTI4_IRQ_INT0	RTI4	RTI4 VIM request 0
IRQ_CROSSBAR_437	RTI4_IRQ_INT1	RTI4	RTI4 VIM request 1
IRQ_CROSSBAR_438	RTI4_IRQ_INT2	RTI4	RTI4 VIM request 2
IRQ_CROSSBAR_439	RTI4_IRQ_INT3	RTI4	RTI4 VIM request 3
IRQ_CROSSBAR_440	RTI4_IRQ_OVL0	RTI4	RTI4 overflow interrupt 0
IRQ_CROSSBAR_441	RTI4_IRQ_OVL1	RTI4	RTI4 overflow interrupt 1
IRQ_CROSSBAR_442	RTI5_IRQ_INT0	RTI5	RTI5 VIM request 0
IRQ_CROSSBAR_443	RTI5_IRQ_INT1	RTI5	RTI5 VIM request 1
IRQ_CROSSBAR_444	RTI5_IRQ_INT2	RTI5	RTI5 VIM request 2
IRQ_CROSSBAR_445	RTI5_IRQ_INT3	RTI5	RTI5 VIM request 3
IRQ_CROSSBAR_446	RTI5_IRQ_OVL0	RTI5	RTI5 overflow interrupt 0
IRQ_CROSSBAR_447	RTI5_IRQ_OVL1	RTI5	RTI5 overflow interrupt 1
IRQ_CROSSBAR_448	RTI1_IRQ_TBINT	RTI1	RTI1 timebase interrupt
IRQ_CROSSBAR_449	RTI2_IRQ_TBINT	RTI2	RTI2 timebase interrupt
IRQ_CROSSBAR_450	RTI3_IRQ_TBINT	RTI3	RTI3 timebase interrupt
IRQ_CROSSBAR_451	RTI4_IRQ_TBINT	RTI4	RTI4 timebase interrupt
IRQ_CROSSBAR_452	RTI5_IRQ_TBINT	RTI5	RTI5 timebase interrupt
IRQ_CROSSBAR_453	MCASP2_IRQ_AREVT	McASP2	McASP2 receive interrupt. **Note: Only available on IPU IRQ_75, DSP1 IRQ_84, and DSP2 IRQ_84 (connectivity restriction).**
IRQ_CROSSBAR_454	MCASP2_IRQ_AXEVT	McASP2	McASP2 transmit interrupt. **Note: Only available on IPU IRQ_76, DSP1 IRQ_85, DSP2 IRQ_85 (connectivity restriction).**
IRQ_CROSSBAR_455	MCASP3_IRQ_AREVT	McASP3	McASP3 receive interrupt. **Note: Only available on IPU IRQ_77, DSP1 IRQ_86, DSP2 IRQ_86 (connectivity restriction).**
IRQ_CROSSBAR_456	MCASP3_IRQ_AXEVT	McASP3	McASP3 transmit interrupt **Note: Only available on IPU IRQ_78, DSP1 IRQ_87, DSP2 IRQ_87 (connectivity restriction).**
IRQ_CROSSBAR_457	MCAN_IRQ_TS	MCAN	MCAN timestamp interrupt. **Note: Only available on IPU IRQ_79 (connectivity restriction).**
IRQ_CROSSBAR_458	Reserved	Reserved	Reserved
IRQ_CROSSBAR_459	Reserved	Reserved	Reserved
IRQ_CROSSBAR_460	Reserved	Reserved	Reserved

12.4 Interrupt Controllers Functional Description

For detailed information about each device INTC (including functional description and registers descriptions), see the TRM chapters and Arm documents referenced in Section 12.1, *Interrupt Controllers Overview*.

Control Module

This chapter describes the system control module for the device.

Topic **Page**

13.1 Control Module Overview

The control module is composed of two submodules, the CTRL_MODULE_CORE submodule and the CTRL_MODULE_WKUP submodule. These two submodules represent a set of registers which are used to control the device I/O ports and also various kinds of settings related to the different modules available in this device.

The CTRL_MODULE_CORE submodule has registers for the following features:

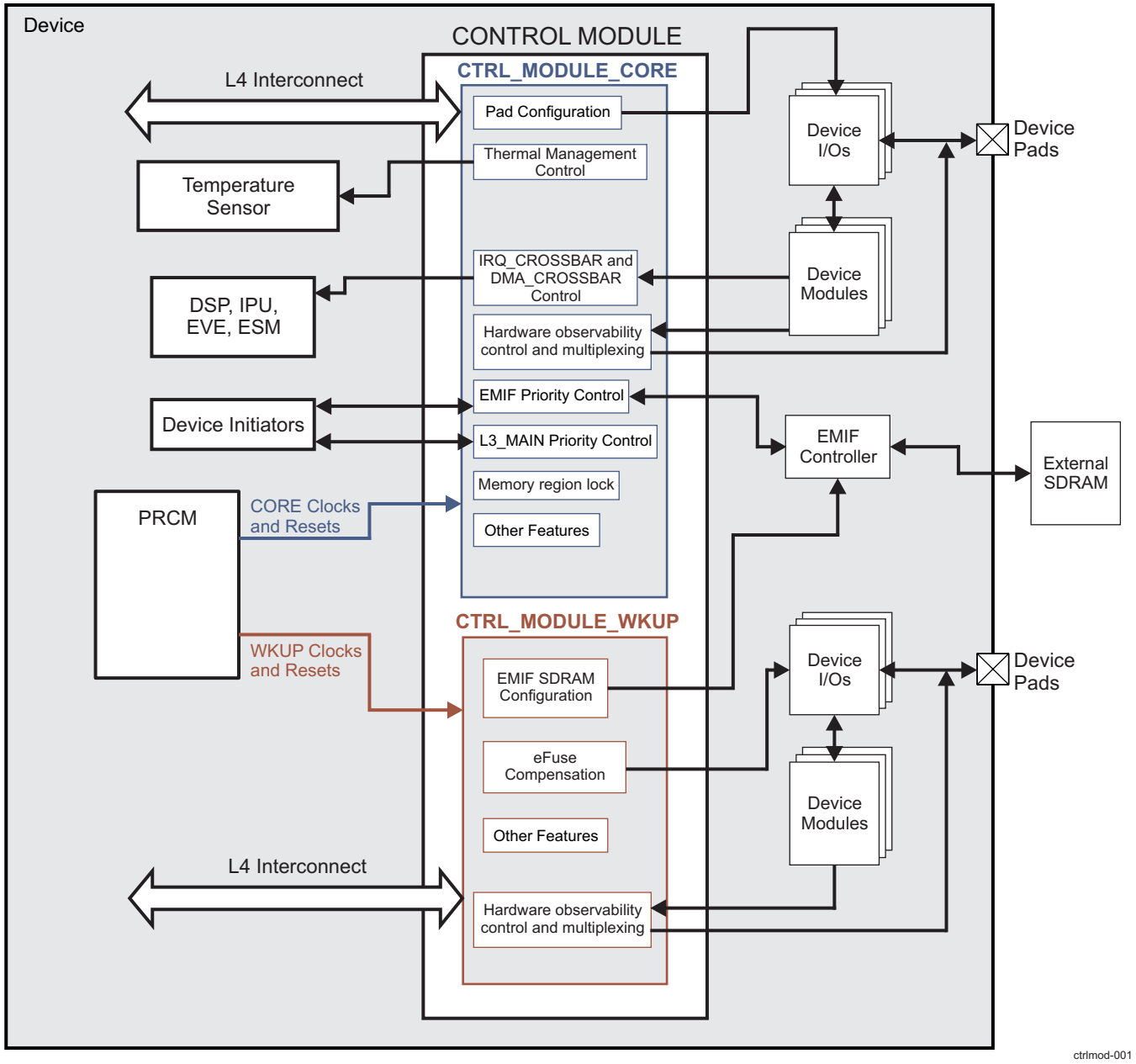
- Pad configuration with following controls:
  - Pad I/O multiplexing
  - Pad pull-up and pull-down configuration
  - Pad input buffer enabling
  - Pad slew rate control
- Device thermal management control and status registers
- IRQ_CROSSBAR and DMA_CROSSBAR control
- Control the priority of initiator accesses to the external SDRAM
- Control the priority of initiators connected to L3_MAIN interconnect
- Memory region lock registers
- Mapping of the device non-maskable interrupt (NMI) to respective cores
- Controls for the LPDDR2/DDR2/DDR3 I/O Cells
- Controls for the LPDDR2/DDR2/DDR3 associated vref-generation cells
- AVS Class 0 associated registers
- Status of the system boot settings
- Firewalls error status
- Settings related to different peripheral modules
- Hardware observability control and multiplexing
- Registers related to PRCM and PLL control
- Registers related to the boot ROM
- Other functions

The CTRL_MODULE_WKUP submodule has registers for the following features:

- Basic EMIF configuration settings
- XTAL Oscilator control
- Efuse I/O compensation
- Hardware observability control and multiplexing
- Other functions

Figure 13-1 represents an overview block diagram of the control module.

Figure 13-1. Control Module Overview Block Diagram



13.2 Control Module Environment

Figure 13-2 shows the control module environment.

Figure 13-2. Control Module Environment

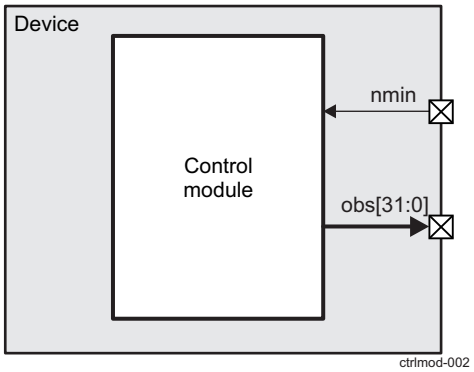


Table 13-1 shows the NMI input and observability output signals.

Table 13-1. Control Module I/O Signals Description

Signal	I/O[1]	Description
nmin	I	External non-maskable interrupt signal[2]
obs[31:0]	O	32 hardware observability signals [3]

(1) I = Input; O = Output; I/O = Bidirectional
(2) For more information, see Section 13.4.6.9, *NMI Mapping to respective cores*
(3) For more information, see Section 13.4.6.14, *Hardware Observability Related Registers*

13.3 Control Module Integration

Figure 13-3 shows the integration of the control module in the device.

Figure 13-3. Control Module Integration

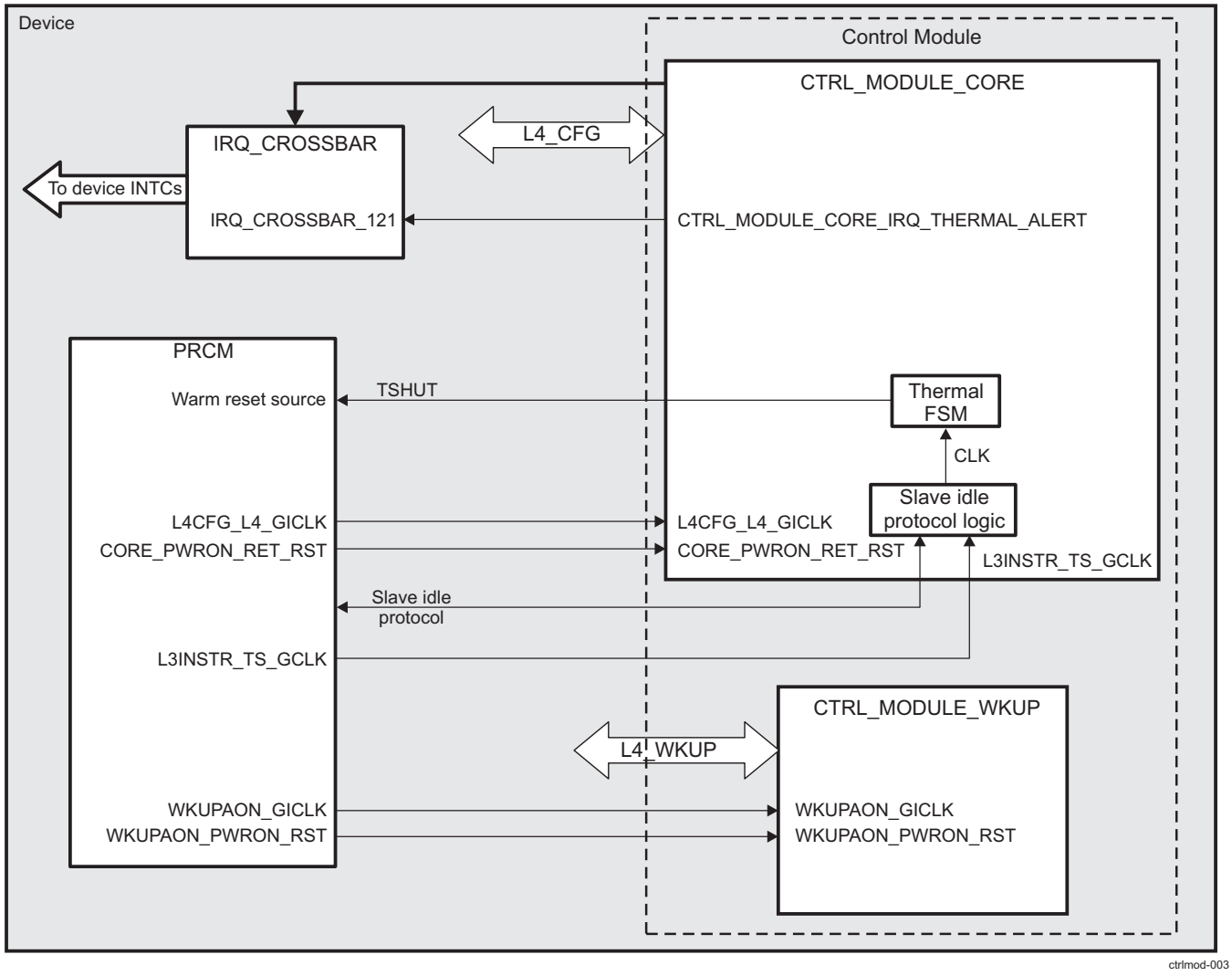


Table 13-2 through Table 13-4 summarize the integration of the Control Module in the device.

Table 13-2. Control Module Integration Attributes

Submodule	Attributes	
	Power Domain	Interconnect
CTRL_MODULE_CORE	PD_COREAON	L4_CFG
CTRL_MODULE_WKUP	PD_WKUPAON	L4_WKUP

Table 13-3. Control Module Clocks and Resets

	Clocks			
Submodule	Destination Signal Name	Source Signal Name	Source	Description

Table 13-3. Control Module Clocks and Resets (continued)

CTRL_MODULE_CORE	L4CFG_L4_GICLK	L4CFG_L4_GICLK	PRCM	Interface clock to the CTRL_MODULE_CORE submodule
	L3INSTR_TS_GCLK	L3INSTR_TS_GCLK	PRCM	Functional clock for the thermal FSM instantiated in the CTRL_MODULE_CORE submodule
CTRL_MODULE_WKUP	WKUPAON_GICLK	WKUPAON_GICLK	PRCM	Interface clock to the CTRL_MODULE_WKUP submodule
		Resets		
CTRL_MODULE_CORE	CORE_PWRON_RET_RST	CORE_PWRON_RET_RST	PRCM	Internal power-on reset (POR) affecting the CTRL_MODULE_CORE submodule
CTRL_MODULE_WKUP	WKUPAON_PWRON_RST	WKUPAON_PWRON_RST	PRCM	Internal POR affecting the CTRL_MODULE_WKUP submodule

Table 13-4. Control Module Hardware Requests

Interrupt Requests				
Submodule	**Source Signal Name**	**Destination IRQ_CROSSBAR Input**	**Default Mapping**	**Description**
CTRL_MODULE_CORE	CTRL_MODULE_CORE_IRQ_THERMAL_ALERT	IRQ_CROSSBAR_121	-	Thermal alert interrupt signal generated when the thermal sensor goes over the tepmerature threshold value. This IRQ source signal is not mapped by default to any device INTC.

NOTE: The "**Default Mapping**" column in Table 13-4 *Control Module Hardware Requests* shows the default mapping of module IRQ source signals. These IRQ source signals can also be mapped to other lines of each device Interrupt controller through the IRQ_CROSSBAR module.
For more information about the IRQ_CROSSBAR module, see Section 13.4.6.4, *IRQ_CROSSBAR Module Functional Description*.
For more information about the device interrupt controllers, see Chapter 12, *Interrupt Controllers*.

The integration of the control module is the following:

- No wake-up request generation
- No DMA request generation
- No master standby protocol with the PRCM
- Slave idle protocol, between the CTRL_MODULE_CORE submodule and the PRCM, related only to the L3INSTR_TS_GCLK
- One interrupt request to the IRQ_CROSSBAR module
- One (thermal shutdown) TSHUT signal used as PRCM warm reset source
- Two clocks and one reset signal to the CTRL_MODULE_CORE submodule
- One clock and one reset signal to the CTRL_MODULE_WKUP submodule

13.4 Control Module Functional Description

13.4.1 Control Module Clock Configuration

There is no software control over the L4CFG_L4_GICLK and WKUPAON_GICLK clocks neither in the control module nor in the PRCM module. The L4CFG_L4_GICLK clock is automatically gated when there is no access to the CTRL_MODULE_CORE registers and the WKUPAON_GICLK is automatically gated when there is no access to the CTRL_MODULE_WKUP registers. There are clock activity status bits for these two clocks in the PRCM module.

The L3INSTR_TS_GCLK is controlled by the CTRL_CORE_BANDGAP_MASK_1[31:30] SIDLEMODE bit field. For more information, see Section 13.4.6.3.5.

The L3INSTR_TS_GCLK has also the following software controls located in the PRCM module:

- Status: see Chapter 3, *Power, Reset, and Clock Management*
- Divider ratio: see Chapter 3, *Power, Reset, and Clock Management*

13.4.2 Control Module Resets

The control module is not sensitive to software reset. It does not respond to global warm reset too. The control module can be reset only by the internal POR (global cold reset).

Despite the previously stated that control module is not sensitive to global warm reset the following registers are exception of this rule and are sensitive to global warm reset:

- All CTRL_CORE_PAD_x registers
- CTRL_CORE_IPU_WAKEUP
- CTRL_CORE_TESOC_LAST_RESET_INDICATOR
- CTRL_CORE_ROM_AUXBOOT0
- CTRL_CORE_ROM_AUXBOOT1

All other control module registers are sensitive only to global cold reset.

The PRCM provides the CORE_PWRON_RET_RST POR signal to the CTRL_MODULE_CORE and the WKUPAON_PWRON_RST POR signal to the CTRL_MODULE_WKUP. For more information, see Section 3.5.5, *Reset Domains*, in Chapter 3, *Power, Reset, and Clock Management*.

13.4.3 Control Module Power Management

13.4.3.1 Power Management Protocols

The control module, which is slave on the L4 interconnect, does not support master standby or slave idle protocols for handshaking with the PRCM. Only the thermal FSM supports slave idle protocol used to control its functional clock, that is the L3INSTR_TS_GCLK clock.

13.4.4 Hardware Requests

The control module does not generate DMA and wake-up requests. The CTRL_MODULE_CORE submodule generates only one IRQ to the IRQ_CROSSBAR module. This is the CTRL_MODULE_CORE_IRQ_THERMAL_ALERT interrupt. For more information, see Section 13.4.6.4, *IRQ_CROSSBAR Module Functional Description*.

13.4.5 Control Module Initialization

The control module responds to the internal POR. During device initialization, only modules used at boot time are associated with the pads. Other module inputs are internally tied and output buffers are turned off. After POR, software must set the pad configuration registers to appropriate values according to the desired device configuration.

The CTRL_CORE_BOOTSTRAP[15:0] bit field reflects the state of the sysboot[15:0] pads captured at POR in the PRCM module.

13.4.6 Functional Description Of The Various Register Types In CTRL_MODULE_CORE Submodule

The following sections describe the purpose of the various kinds of registers and register groups which reside in the CTRL_MODULE_CORE submodule.

13.4.6.1 Pad Configuration Registers

The pad configuration registers are used to configure most of the device pads. Each pad configuration register is assotiated only with one pad. The name of each register is formed by the corresponding pad name and the prefix "CTRL_CORE_PAD_". Almost all pad configuration registers have same bits. In some of these registers certain bits cannot be present. Figure 13-4 shows the general case in which all the pad configuration register bits are present. Table 13-5 describes these bits.

After POR, software must set the pad configuration registers to appropriate values depending on the desired device configuration.

Figure 13-4. Pad Configuration Register Bits

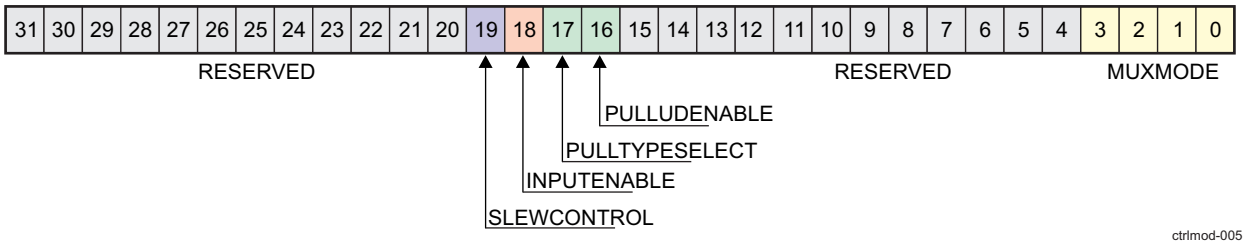


Table 13-5. Description Of The Pad Configuration Register Bits

Bit/Bit Field	Bit Meaning		Description
	0b0	0b1	
SLEWCONTROL	Fast slew is selected	Slow slew is selected	Selects the slew rate for a given pad. The slew rate should be set to the value specified in the device Data Manual for a given mode of operation.
INPUTENABLE	Receive mode is disabled. The pad is configured in output mode only.	Receive mode is enabled. The pad is configured in bidirectional mode.	Enables the input buffer of a given I/O
PULLTYPESELECT	Weak pull-down resistor is selected	Weak pull-up resistor is selected	Weak pull-up or weak pull-down resistor selection for a given pad
PULLUDENABLE	Weak pull-up/pull-ldown resistor is enabled	Weak pull-up/pull-down resistor is disabled	Enables weak pull-up/pull-down feature of a given pad
MUXMODE	This bit field selects one of maximum sixteen possible functions to be available on the corresponding pad. Mostly, only several functions are available.		

When MUXMODE is set to 0x0 the function mapped to the pad corresponds to the name of the pad. In other words, pad names are signal names available when MUXMODE = 0x0.

By default the MUXMODE bit field of most device pads is set to 0xF, which means that there are no signals routed to these pads and the pads are in Hi-Z. The only exception are the sysboot signals available on several pads when MUXMODE = 0xF.

Only MUXMODE values which correspond to defined functions should be used.

NOTE: The default SLEWCONTROL settings in each pad configuration register must be used to guarantee timings, unless specific instructions otherwise are given in the individual timing sub-sections of the device Data Manual. The only exception is when the MUXMODE is configured to select a vout*_* signal. In this case the corresponding SLEWCONTROL bit must be configured to slow slew instead of the default fast slew.

NOTE: When an external pull resistor is desired, the internal pull resistors must be disabled by software because they are enabled by default.

NOTE: For information about the signals available on each pad, see the *Multiplexing Characteristics* table in the device Data Manual.

NOTE: GPIO pins can directly snoop device pads even if those are configured in the other modes. For GPIO output user needs to set pad mux.

The following signals require CTRL_CORE_PAD_x to be programmed with INPUTENABLE=1 for retiming purposes:

- mmc_clk
- gpmc_clk
- i2cj_scl where j=1-2
- spim_sclk where m=1-4
- mcasp1_aclkx, mcasp1_ahclkx, and mcasp1_aclkr
- mcasp2_aclkx, mcasp2_ahclkx, mcasp2_aclkr, mcasp3_aclkx, mcasp3_ahclkx, and mcasp3_aclkr

Except the maximum 16 possible combinations through the MUXMODE bit fields using the bits within CTRL_CORE_SMA_SW_14 and CTRL_CORE_SMA_SW_15 registers an additional signal can be mapped to several device pads. In other words, for these pads there are up to 17 possible signal combinations although not all of them are really implemented. These registers control McASP2, McASP3 and ATL signal multiplexing. If a bit is set to 0x1, function 17 is selected for the corresponding pad. In that case the MUXMODE field of the corresponding CTRL_CORE_PAD_x register must be set to 0xF. If a bit is set to 0x0, function 17 is not selected.

13.4.6.2 Pull Selection

There is no automatic gating control to ensure that internal weak pull-up or pull-down resistors on a pad are disconnected whenever pad is configured as output. If a pad is always configured in output mode, it is recommended for user software to disable any internal pull resistor tied to it to avoid unnecessary consumption.

Table 13-6 describes the software controls available for pad internal pull-up and pull-down resistors in the control module pad configuration registers.

Table 13-6. Pull Selection

PULL		Pad Behavior
PULLTYPESELECT	PULLUDENABLE	
0	1	Pull-down selected but not activated
0	0	Pull-down selected and activated
1	1	Pull-up selected but not activated
1	0	Pull-up selected and activated

13.4.6.3 Thermal Management Related Registers

There is one temperature sensor on the device die. It is part of a VBGAPTS cell. This cell has a 10-bit ADC. The ADC converts the temperature values into digital output values proportional to the temperature measured. The VBGAPTS cell is controlled by a dedicated FSM referred to as thermal FSM. The registers associated with this FSM reside in the CTRL_MODULE_CORE submodule. The FSM is clocked by the L3INSTR_TS_GCLK clock. The PRCM module controlls this clock through the slave idle protocol.

The device thermal management related registers can be split into the following classes:

- Temperature sensor control registers
- Registers for the thermal alert comparator block
- Temperature timestamp registers
- Other registers used for:
  - controlling the FIFO
  - controlling the clock provided to the thermal FSM

Figure 13-5 shows the block diagram of the device thermal management.

Figure 13-5. Thermal Management Functional Block Diagram

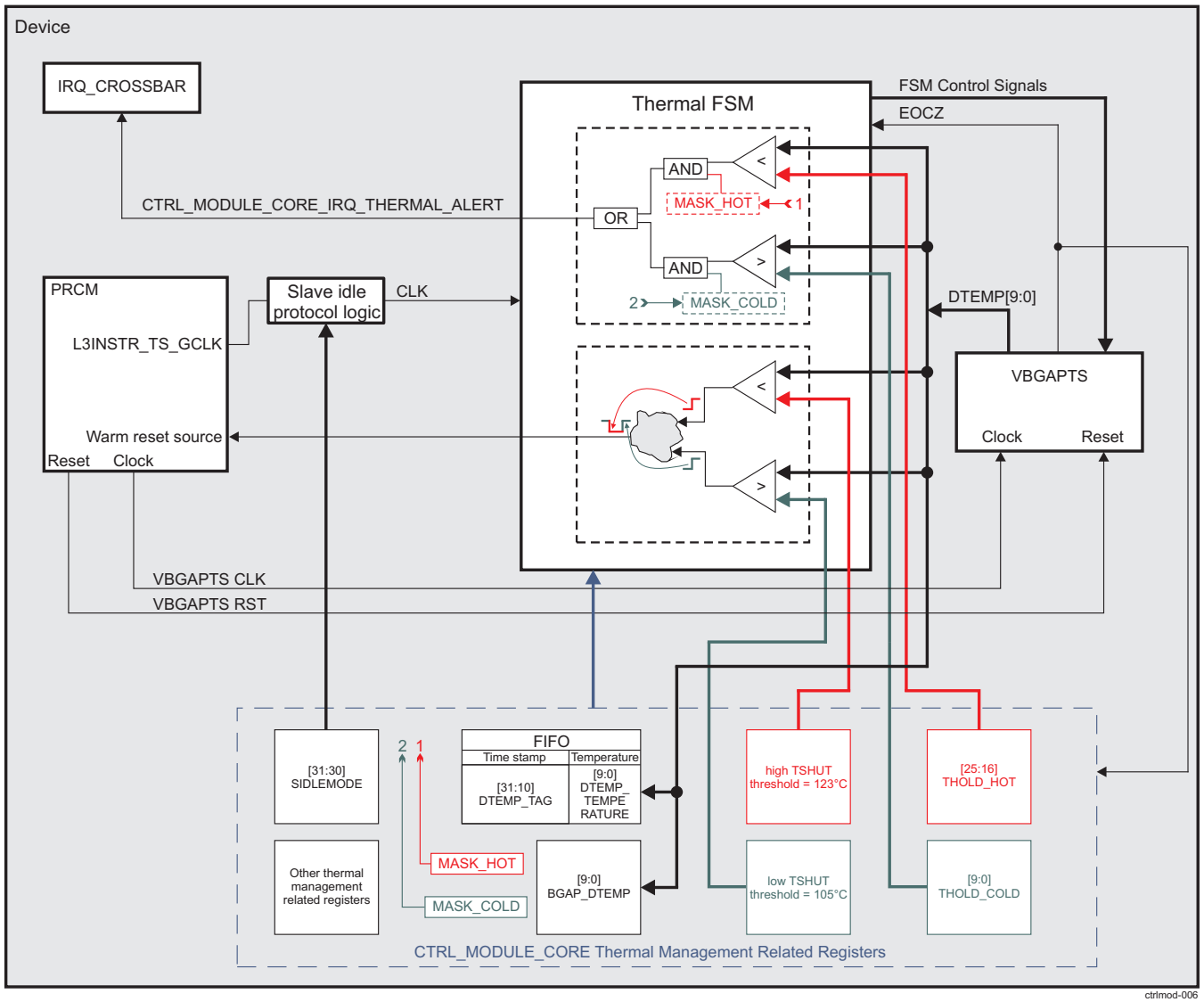


Table 13-7 describes the signals related to the device thermal management logic.

Table 13-7. Thermal Management Signals Description

Signal	I/O[1]	Description
EOCZ	O	End of conversion signal. When low, this signal indicates that the value of DTEMP[9:0] is valid.
DTEMP[9:0]	O	Temperature data from the temperature sensor. This value is valid when EOCZ is low.

(1) I = Input; O = Output

Table 13-7. Thermal Management Signals Description (continued)

Signal	I/O[1]	Description
VBGAPTS CLK	I	Functional clock from the WKUPAON power domain used by the temperature sensor during temperature conversion.
THERMAL ALERT	O	This is an output signal from the thermal FSM mapped as an interrupt request to the IRQ_CROSSBAR module. Software uses this interrupt to implement the device thermal management policy.
TSHUT	O	This is an output signal from the thermal FSM mapped to the PRCM and is used as a warm reset signal. This overheat protection signal is high during normal operation and go low during thermal shutdown event.

The ADC values which correspond to the current temperature are listed in Table 13-10.

13.4.6.3.1 Temperature Sensor Control Registers

The VBGAPTS cell works in continuous conversion mode controlled by the thermal FSM. This means that the temperature is measured at regular time intervals. The start of temperature measurement is initiated automatically by the thermal FSM after it goes out of reset state. To control the main delay between two measurements the CTRL_CORE_BANDGAP_MASK_1[29:27] COUNTER_DELAY bit field is used. After this delay expires the FSM automatically starts a temperature conversion.

The conversion is complete when the CTRL_CORE_TEMP_SENSOR_CORE[10] BGAP_EOCZ_CORE status bits is set to 0x0. After this the valid temperature is written automatically in the CTRL_CORE_TEMP_SENSOR_CORE[9:0] BGAP_DTEMP_CORE bit field, and then software is able to read it. After writing the valid temperature value the FSM waits one clock cycle and start another conversion cycle.

13.4.6.3.2 Registers For The Thermal Alert Comparator Block

There is a comparator block responsible for the thermal alert function of the CTRL_MODULE_CORE thermal management logic. This comparator block is composed of two comparators. One dedicated to low temperature threshold and the other one to high temperature threshold. Software can configure the low temperature threshold through the CTRL_CORE_BANDGAP_THRESHOLD_CORE[9:0] THOLD_COLD_CORE bit field and high temperature threshold through the CTRL_CORE_BANDGAP_THRESHOLD_CORE[25:16] THOLD_HOT_CORE bit field. The values which have to be loaded in these bit fields are the same as those listed in Table 13-10. For example, to set the high temperature threshold value to 120°C, the THOLD_HOT_CORE filed must be loaded with value of 931 (0x3A3).

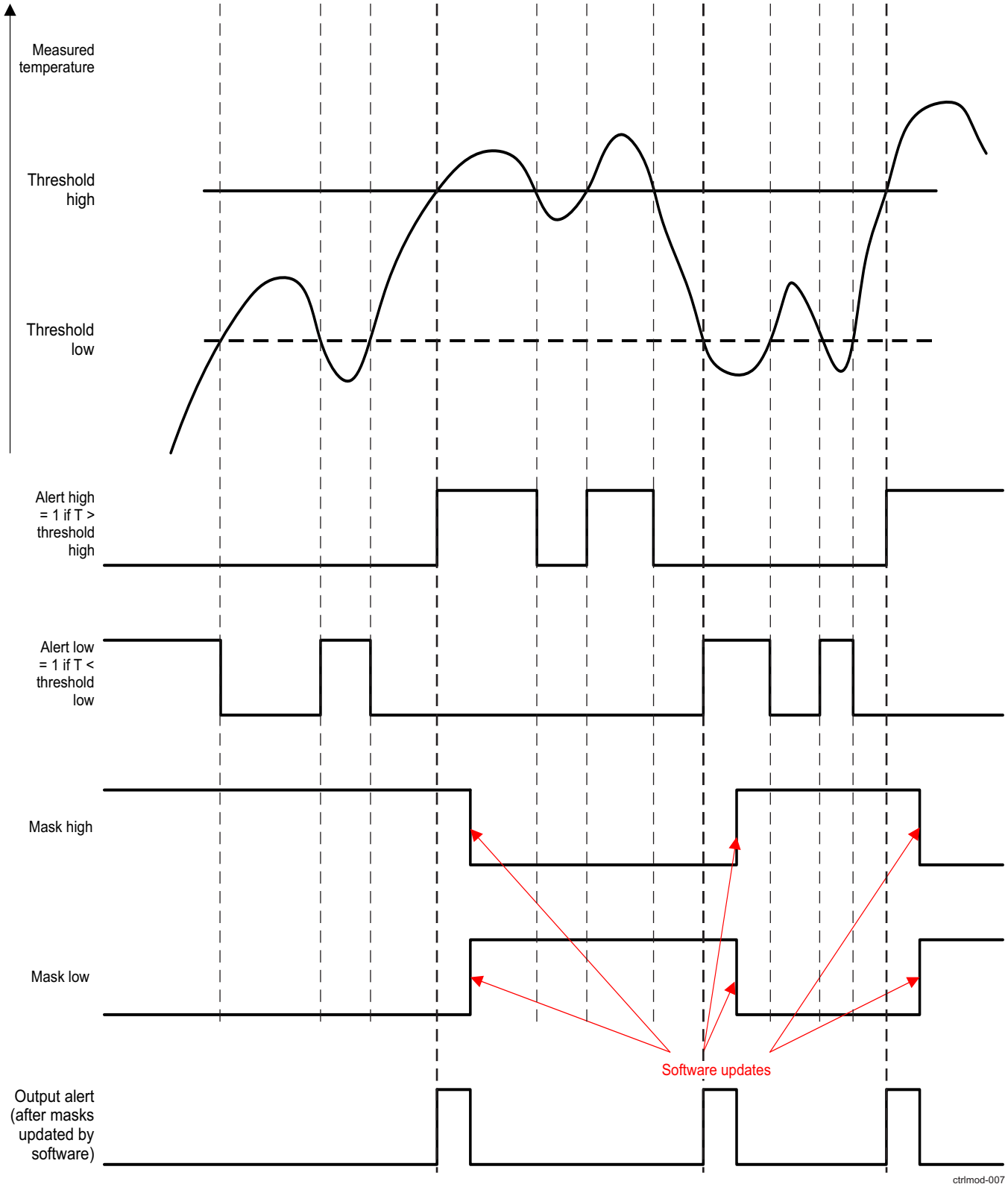
The thermal alert logic provides also a capability to mask the outputs of the comparators associated with low and high temperature thresholds. The low temperature threshold comparator output can be masked by setting to 0x0 the CTRL_CORE_BANDGAP_MASK_1[4] MASK_COLD_CORE bit. The high temperature threshold comparator output can be masked by setting to 0x0 the CTRL_CORE_BANDGAP_MASK_1[5] MASK_HOT_CORE bit.

The masked low and high temperature threshold outputs are ORed (see Figure 13-5) in the CTRL_MODULE_CORE_IRQ_THERMAL_ALERT signal, which delivers an interrupt to the IRQ_CROSSBAR_121 input line of the IRQ_CROSSBAR module. Software can use this interrupt to implement the device thermal management policy. The CTRL_MODULE_CORE_IRQ_THERMAL_ALERT signal is also routed to the CTRL_CORE_BANDGAP_STATUS_1[31] ALERT bit. Value of 0x1 indicates that the CTRL_MODULE_CORE_IRQ_THERMAL_ALERT signal is asserted.

The CTRL_CORE_BANDGAP_STATUS_1[4] COLD_CORE bit indicates the non masked (raw) comparator output for the low temperature threshold. The CTRL_CORE_BANDGAP_STATUS_1[5] HOT_CORE bit indicates the non masked (raw) comparator output for the high temperature threshold.

Figure 13-6 shows the behavior of the thermal alert logic.

Figure 13-6. Behavior Of The Thermal Alert Logic



13.4.6.3.3 Thermal Shutdown Comparator Block

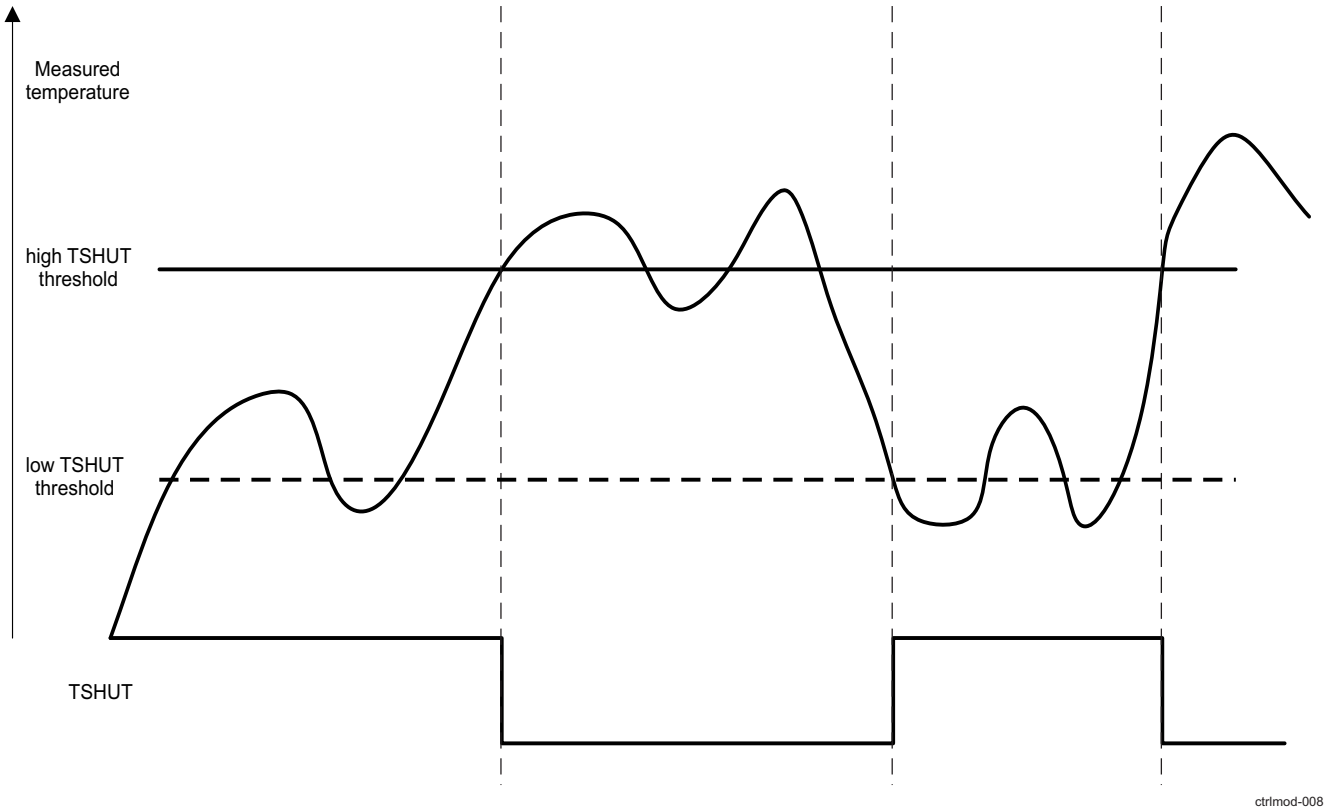
There is also a comparator block responsible for the thermal shutdown (TSHUT) function of the CTRL_MODULE_CORE thermal management logic. This comparator block is also composed of two comparators. One dedicated to low TSHUT threshold and the other one to high TSHUT threshold. The comparator outputs for the low and high TSHUT thresholds are connected to a logic which then generates a single TSHUT signal tied to the PRCM and used as warm reset source. For details, see Figure 13-5.

The values of the low and high TSHUT thresholds are fixed and can be neither overridden nor read by software. The value for the high TSHUT threshold is 123°C (assuming +/-2C temperature sensor accuracy) and for the low TSHUT threshold it is 105°C.

When the high TSHUT threshold is reached, the TSHUT output is activated. To deactivate the TSHUT output, the temperature must go below the value of the low TSHUT threshold.

The TSHUT signal is used as an overheat protection. Figure 13-7 shows the behavior of the TSHUT logic.

Figure 13-7. Behavior Of The Thermal Shutdown Logic



13.4.6.3.4 Temperature Timestamp Registers

Each time when the CTRL_CORE_TEMP_SENSOR_CORE[9:0] BGAP_DTEMP_CORE bit field is updated with new temperature value, this value is also automatically stored into a 5-level deep FIFO and a timestamp is registered too. This FIFO is used to store a brief history for the last few temperature measurements and is also dedicated to temperature timestamping. The FIFO has two fields. The first one is 10 bits wide, 5 levels deep, and is intended to store the temperature values for the last five measurements. The second field is 22 bits wide, 5 levels deep, and acts like a counter for the number of temperature measurements. The FIFO is composed of the following registers:

- CTRL_CORE_DTEMP_CORE_0
- CTRL_CORE_DTEMP_CORE_1
- CTRL_CORE_DTEMP_CORE_2
- CTRL_CORE_DTEMP_CORE_3

- CTRL_CORE_DTEMP_CORE_4

Table 13-8 describes the FIFO.

Table 13-8. FIFO Description

FIFO Levels	Second FIFO Field (22 Bits) – Timestamp Bits [31:10]	Description	First FIFO field (10 Bits) – Temperature Bits [9:0]	Description
Level 1	DTEMP_TAG_CORE_0	Indicates the number of temperature measurements	DTEMP_TEMPERATURE_CORE_0	Indicates the last measured temperature value (the most recent sample)
Level 2	DTEMP_TAG_CORE_1	Indicates the number of temperature measurements minus one (DTEMP_TEMPERATURE_CORE_0 – 1)	DTEMP_TEMPERATURE_CORE_1	Indicates the penultimate measured temperature value
Level 3	DTEMP_TAG_CORE_2	Indicates the number of temperature measurements minus two (DTEMP_TEMPERATURE_CORE_0 – 2)	DTEMP_TEMPERATURE_CORE_2	Indicates temperature value measured before DTEMP_TEMPERATURE_CORE_1
Level 4	DTEMP_TAG_CORE_3	Indicates the number of temperature measurements minus three (DTEMP_TEMPERATURE_CORE_0 – 3)	DTEMP_TEMPERATURE_CORE_3	Indicates temperature value measured before DTEMP_TEMPERATURE_CORE_2
Level 5	DTEMP_TAG_CORE_4	Indicates the number of temperature measurements minus four (DTEMP_TEMPERATURE_CORE_0 – 4)	DTEMP_TEMPERATURE_CORE_4	Indicates temperature value measured before DTEMP_TEMPERATURE_CORE_3 (the oldest sample)

NOTE: DTEMP_TAG_CORE_4 increments its value with one after each fifth temperature measurement.

DTEMP_TAG_CORE_3 increments its value with one after each fourth temperature measurement.

DTEMP_TAG_CORE_2 increments its value with one after each third temperature measurement.

DTEMP_TAG_CORE_1 increments its value with one after each second temperature measurement.

DTEMP_TAG_CORE_0 increments its value with one after each temperature measurement.

13.4.6.3.5 Other Thermal Management Related Registers

- **Controlling the FIFO:**

  Software can stop the FIFO to update with new temperature and timestamp values when setting to 0x1 the CTRL_CORE_BANDGAP_MASK_1[23] FREEZE_CORE bit. This bit is automatically cleared by hardware after the FIFO has been cleared.

  The FIFO is cleared by setting to 0x1 the CTRL_CORE_BANDGAP_MASK_1[20] CLEAR_CORE bit. This bit is automatically set by hardware to 0x0 after the FIFO clearing procedure completes.

- **Controlling the clock provided to the thermal FSM:**

  The thermal FSM complies with the PRCM slave idle protocol. Its functional clock (L3INSTR_TS_GCLK) is automatically gated by PRCM depending on the value of the CTRL_CORE_BANDGAP_MASK_1[31:30] SIDLEMODE bit field. L3INSTR_TS_GCLK clock is also enabled automatically by the PRCM module.

13.4.6.3.6 Summary Of The Thermal Management Related Registers

Table 13-9 summarizes all the thermal management related registers.

Table 13-9. Summary Of The Thermal Management Related Registers

Register Name	Description	Access
CTRL_CORE_TEMP_SENSOR_CORE	Temperature sensor control register	RW
CTRL_CORE_BANDGAP_THRESHOLD_CORE	Register for the thermal alert comparator	RW
CTRL_CORE_BANDGAP_TSHUT_CORE	Register for the thermal shutdown comparator	RW
CTRL_CORE_DTEMP_CORE_0 CTRL_CORE_DTEMP_CORE_1 CTRL_CORE_DTEMP_CORE_2 CTRL_CORE_DTEMP_CORE_3 CTRL_CORE_DTEMP_CORE_4	Temperature timestamp registers	RO
CTRL_CORE_BANDGAP_STATUS_1	Register with status bits for the non masked comparator outputs and the thermal alert signal.	RO
CTRL_CORE_BANDGAP_MASK_1	Register used to mask the comparator outputs for the low and high thermal alert thresholds. This register is also used to control the FIFO and the clock provided to the thermal FSM.	RW

13.4.6.3.7 ADC Values Versus Temperature

Table 13-10 provides all valid ADC values corresponding to the measured temperature which should be read from the CTRL_CORE_TEMP_SENSOR_CORE[9:0] BGAP_DTEMP_CORE bit field. Table 13-10 also provides the values for the low and high temperature thresholds which are configurable through the CTRL_CORE_BANDGAP_THRESHOLD_CORE[9:0] THOLD_COLD_CORE and CTRL_CORE_BANDGAP_THRESHOLD_CORE[25:16] THOLD_HOT_CORE bit fields.

Table 13-10. ADC Values Versus Temperature

ADC Code	Temperature		ADC Code	Temperature		ADC Code	Temperature		ADC Code	Temperature	
	From	To		From	To		From	To		From	To
0-539	Outside region of operation		641	0.8	1.2	743	43.4	44	845	85.2	85.6
540	-40	-40	642	1.2	1.6	744	44	44.4	846	85.6	86
541	-40	-40	643	1.6	2	745	44.4	44.8	847	86	86.4
542	-40	-40	644	2	2.4	746	44.8	45.2	848	86.4	86.8
543	-40	-40	645	2.4	2.8	747	45.2	45.6	849	86.8	87.2
544	-40	-39.6	646	2.8	3.2	748	45.6	46	850	87.2	87.6
545	-39.6	-39.2	647	3.2	3.6	749	46	46.4	851	87.6	88
546	-39.2	-38.8	648	3.6	4.2	750	46.4	46.8	852	88	88.4
547	-38.8	-38.4	649	4.2	4.8	751	46.8	47.2	853	88.4	88.8
548	-38.4	-38	650	4.8	5.2	752	47.2	47.6	854	88.8	89.2
549	-38	-37.6	651	5.2	5.6	753	47.6	48	855	89.2	89.6
550	-37.6	-37.2	652	5.6	6	754	48	48.4	856	89.6	90
551	-37.2	-36.8	653	6	6.4	755	48.4	48.8	857	90	90.4
552	-36.8	-36.4	654	6.4	6.8	756	48.8	49.2	858	90.4	90.8
553	-36.4	-36	655	6.8	7.2	757	49.2	49.6	859	90.8	91.2
554	-36	-35.6	656	7.2	7.6	758	49.6	50	860	91.2	91.6
555	-35.6	-35	657	7.6	8	759	50	50.4	861	91.6	92
556	-35	-34.4	658	8	8.4	760	50.4	50.8	862	92	92.4
557	-34.4	-34	659	8.4	8.8	761	50.8	51.2	863	92.4	92.8

Table 13-10. ADC Values Versus Temperature (continued)

ADC Code	Temperature		ADC Code	Temperature		ADC Code	Temperature		ADC Code	Temperature	
558	-34	-33.6	660	8.8	9.2	762	51.2	51.6	864	92.8	93.2
559	-33.6	-33.2	661	9.2	9.6	763	51.6	52	865	93.2	93.6
560	-33.2	-32.8	662	9.6	10	764	52	52.4	866	93.6	94
561	-32.8	-32.4	663	10	10.4	765	52.4	52.8	867	94	94.4
562	-32.4	-32	664	10.4	10.8	766	52.8	53.2	868	94.4	94.8
563	-32	-31.6	665	10.8	11.2	767	53.2	53.6	869	94.8	95.2
564	-31.6	-31.2	666	11.2	11.6	768	53.6	54	870	95.2	95.6
565	-31.2	-30.8	667	11.6	12	769	54	54.4	871	95.6	96
566	-30.8	-30.4	668	12	12.4	770	54.4	54.8	872	96	96.4
567	-30.4	-30	669	12.4	13	771	54.8	55.2	873	96.4	96.8
568	-30	-29.6	670	13	13.6	772	55.2	55.6	874	96.8	97.2
569	-29.6	-29.2	671	13.6	14	773	55.6	56.2	875	97.2	97.8
570	-29.2	-28.8	672	14	14.4	774	56.2	56.8	876	97.8	98.4
571	-28.8	-28.4	673	14.4	14.8	775	56.8	57.2	877	98.4	98.8
572	-28.4	-28	674	14.8	15.2	776	57.2	57.6	878	98.8	99.2
573	-28	-27.4	675	15.2	15.6	777	57.6	58	879	99.2	99.6
574	-27.4	-26.8	676	15.6	16	778	58	58.4	880	99.6	100
575	-26.8	-26.4	677	16	16.4	779	58.4	58.8	881	100	100.4
576	-26.4	-26	678	16.4	16.8	780	58.8	59.2	882	100.4	100.8
577	-26	-25.6	679	16.8	17.2	781	59.2	59.6	883	100.8	101.2
578	-25.6	-25.2	680	17.2	17.6	782	59.6	60	884	101.2	101.6
579	-25.2	-24.8	681	17.6	18	783	60	60.4	885	101.6	102
580	-24.8	-24.4	682	18	18.4	784	60.4	60.8	886	102	102.4
581	-24.4	-24	683	18.4	18.8	785	60.8	61.2	887	102.4	102.8
582	-24	-23.6	684	18.8	19.2	786	61.2	61.6	888	102.8	103.2
583	-23.6	-23.2	685	19.2	19.6	787	61.6	62	889	103.2	103.6
584	-23.2	-22.8	686	19.6	20	788	62	62.4	890	103.6	104
585	-22.8	-22.4	687	20	20.4	789	62.4	62.8	891	104	104.4
586	-22.4	-22	688	20.4	20.8	790	62.8	63.2	892	104.4	104.8
587	-22	-21.6	689	20.8	21.2	791	63.2	63.6	893	104.8	105.2
588	-21.6	-21.2	690	21.2	21.6	792	63.6	64	894	105.2	105.6
589	-21.2	-20.8	691	21.6	22.2	793	64	64.4	895	105.6	106
590	-20.8	-20.2	692	22.2	22.8	794	64.4	64.8	896	106	106.4
591	-20.2	-19.6	693	22.8	23.2	795	64.8	65.2	897	106.4	106.8
592	-19.6	-19.2	694	23.2	23.6	796	65.2	65.6	898	106.8	107.2
593	-19.2	-18.8	695	23.6	24	797	65.6	66	899	107.2	107.6
594	-18.8	-18.4	696	24	24.4	798	66	66.4	900	107.6	108
595	-18.4	-18	697	24.4	24.8	799	66.4	66.8	901	108	108.4
596	-18	-17.6	698	24.8	25.2	800	66.8	67.2	902	108.4	108.8
597	-17.6	-17.2	699	25.2	25.6	801	67.2	67.6	903	108.8	109.2
598	-17.2	-16.8	700	25.6	26	802	67.6	68	904	109.2	109.6
599	-16.8	-16.4	701	26	26.4	803	68	68.4	905	109.6	110
600	-16.4	-16	702	26.4	26.8	804	68.4	68.8	906	110	110.4
601	-16	-15.6	703	26.8	27.2	805	68.8	69.2	907	110.4	110.8
602	-15.6	-15.2	704	27.2	27.6	806	69.2	69.6	908	110.8	111.2
603	-15.2	-14.8	705	27.6	28	807	69.6	70	909	111.2	111.6
604	-14.8	-14.4	706	28	28.4	808	70	70.4	910	111.6	112

Table 13-10. ADC Values Versus Temperature (continued)

ADC Code	Temperature		ADC Code	Temperature		ADC Code	Temperature		ADC Code	Temperature	
605	-14.4	-14	707	28.4	28.8	809	70.4	70.8	911	112	112.4
606	-14	-13.6	708	28.8	29.2	810	70.8	71.2	912	112.4	112.8
607	-13.6	-13.2	709	29.2	29.6	811	71.2	71.8	913	112.8	113.2
608	-13.2	-12.8	710	29.6	30	812	71.8	72.4	914	113.2	113.6
609	-12.8	-12.2	711	30	30.4	813	72.4	72.8	915	113.6	114
610	-12.2	-11.6	712	30.4	30.8	814	72.8	73.2	916	114	114.4
611	-11.6	-11.2	713	30.8	31.2	815	73.2	73.6	917	114.4	114.8
612	-11.2	-10.8	714	31.2	31.6	816	73.6	74	918	114.8	115.2
613	-10.8	-10.4	715	31.6	32.2	817	74	74.4	919	115.2	115.6
614	-10.4	-10	716	32.2	32.8	818	74.4	74.8	920	115.6	116
615	-10	-9.6	717	32.8	33.2	819	74.8	75.2	921	116	116.4
616	-9.6	-9.2	718	33.2	33.6	820	75.2	75.6	922	116.4	116.8
617	-9.2	-8.8	719	33.6	34	821	75.6	76	923	116.8	117.2
618	-8.8	-8.4	720	34	34.4	822	76	76.4	924	117.2	117.6
619	-8.4	-8	721	34.4	34.8	823	76.4	76.8	925	117.6	118
620	-8	-7.6	722	34.8	35.2	824	76.8	77.2	926	118	118.4
621	-7.6	-7.2	723	35.2	35.6	825	77.2	77.6	927	118.4	118.8
622	-7.2	-6.8	724	35.6	36	826	77.6	78	928	118.8	119.2
623	-6.8	-6.4	725	36	36.4	827	78	78.4	929	119.2	119.6
624	-6.4	-6	726	36.4	36.8	828	78.4	78.8	930	119.6	120
625	-6	-5.6	727	36.8	37.2	829	78.8	79.2	931	120	120.4
626	-5.6	-5.2	728	37.2	37.6	830	79.2	79.6	932	120.4	120.8
627	-5.2	-4.8	729	37.6	38	831	79.6	80	933	120.8	121.2
628	-4.8	-4.2	730	38	38.4	832	80	80.4	934	121.2	121.6
629	-4.2	-3.6	731	38.4	38.8	833	80.4	80.8	935	121.6	122
630	-3.6	-3.2	732	38.8	39.2	834	80.8	81.2	936	122	122.4
631	-3.2	-2.8	733	39.2	39.6	835	81.2	81.6	937	122.4	122.8
632	-2.8	-2.4	734	39.6	40	836	81.6	82	938	122.8	123.2
633	-2.4	-2	735	40	40.4	837	82	82.4	939	123.2	123.6
634	-2	-1.6	736	40.4	40.8	838	82.4	82.8	940	123.6	124
635	-1.6	-1.2	737	40.8	41.2	839	82.8	83.2	941	124	124.4
636	-1.2	-0.8	738	41.2	41.6	840	83.2	83.6	942	124.4	124.8
637	-0.8	-0.4	739	41.6	42	841	83.6	84	943	124.8	125
638	-0.4	0	740	42	42.4	842	84	84.4	945	125	125
639	0	0.4	741	42.4	42.8	843	84.4	84.8	946-1023	Outside region of operation	
640	0.4	0.8	742	42.8	43.4	844	84.8	85.2			

13.4.6.4 IRQ_CROSSBAR Module Functional Description

There is an IRQ_CROSSBAR module in the device, which is controlled by registers in the CTRL_MODULE_CORE submodule. The IRQ_CROSSBAR is able to map any of its input signals to any of its outputs. This module is associated with the device interrupt sources. The IRQs from all the device modules are connected to the IRQ_CROSSBAR inputs. Each module IRQ is connected only to one cross-bar input. Each output of the IRQ_CROSSBAR module is connected only to one interrupt line of certain interrupt controller (INTC). Thus, the device IRQs are mapped to the device INTCs through the

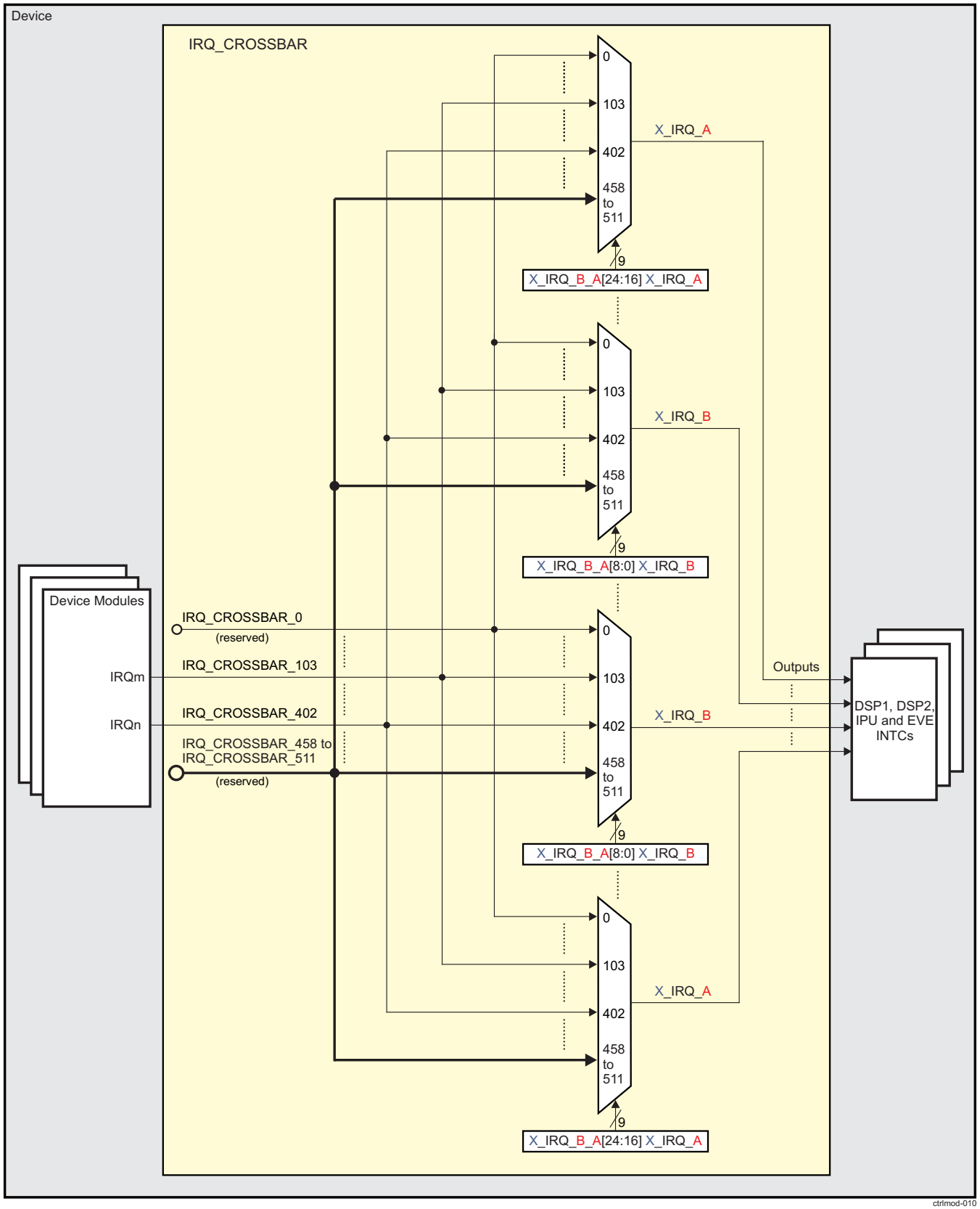
IRQ_CROSSBAR. Some of these IRQs are mapped by default to certain interrupt lines of one of the device INTCs, but there are IRQs which are not mapped by default to any interrupt line of any device INTC. All IRQs, connected to the IRQ_CROSSBAR inputs, can be remapped to other interrupt lines of the different device INTCs through the CTRL_CORE_X_IRQ_B_A registers. Each of these registers has a structure described in Table 13-11.

Table 13-11. Generic Description of the CTRL_CORE_X_IRQ_B_A IRQ_CROSSBAR Control Registers

Bits	Field Name	Description	Type	Note
31:25	RESERVED		R	
24:16	X_IRQ_A	Selects an interrupt source signal for the X_IRQ_A INTC line 0x0: Reserved 0x1: Maps IRQ_CROSSBAR input 1 to X_IRQ_A INTC line 0x2: Maps IRQ_CROSSBAR input 2 to X_IRQ_A INTC line 0x-: ..... 0x67: Maps IRQ_CROSSBAR input 103 to X_IRQ_A INTC line 0x-: ..... 0x1C5 to 0x1FF: Reserved	RW	X is summarization. It is equal to: • DSP1 • DSP2 • IPU • EVE X shows the module name to which interrupt controller all inputs of the IRQ_CROSSBAR module can be mapped to. A is also summarization. It shows the number of the line for the corresponding INTC. A is equal to 0, 1, 2, ..., and so on, depending on the count of the INTC lines controlled by the IRQ_CROSSBAR module. For more details, see Table 13-12.
15:9	RESERVED		R	
8:0	X_IRQ_B	Selects an interrupt source signal for the X_IRQ_B INTC line 0x0: Reserved 0x1: Maps IRQ_CROSSBAR input 1 to X_IRQ_B INTC line 0x2: Maps IRQ_CROSSBAR input 2 to X_IRQ_B INTC line 0x-: ..... 0x67: Maps IRQ_CROSSBAR input 103 to X_IRQ_B INTC line 0x-: ..... 0x1C5 to 0x1FF: Reserved	RW	B is summarization. It shows the number of the line for the corresponding INTC. B is equal to 0, 1, 2, ..., and so on, depending on the count of the INTC lines controlled by the IRQ_CROSSBAR module.

Figure 13-8 represents the way in which the IRQ_CROSSBAR module works. It shows the device modules and their IRQs connected to the IRQ_CROSSBAR inputs, the structure of the cross-bar and its outputs connected to the device INTCs.

Figure 13-8. IRQ_CROSSBAR Module Functional Diagram



Each IRQ_CROSSBAR control register has two 9-bit fields. Each 9-bit field is associated only with one interrupt line. Through this 9-bit field any of the IRQs connected to the IRQ_CROSSBAR inputs can be mapped to the INTC line associated with this 9-bit field. For example, the register CTRL_CORE_DSP1_IRQ_56_57 is associated with DSP1_IRQ_56 and DSP1_IRQ_57. The 9-bit field CTRL_CORE_DSP1_IRQ_56_57[24:16] DSP1_IRQ_57 is associated only with DSP1_IRQ_57 interrupt line of the DSP1 INTC. Setting this bit field to any other value different than its reset value will map another module IRQ to the DSP1_IRQ_57 interrupt line. The default (reset) value of this bit field is 0x1A which corresponds to IRQ_CROSSBAR_26 input. The GPIO3_IRQ_1 is connected to this cross-bar input. Setting another register to 0x1A will cause the GPIO3_IRQ_1 to be mapped to another INTC line. For example, if the CTRL_CORE_DSP1_IRQ_40_41[8:0] DSP1_IRQ_40 is set to 0x1A the GPIO3_IRQ_1 will be mapped to the DSP1_IRQ_40 interrupt line. The same logic also applies to the other interrupt lines of the device INTCs.

In addition, not all of the interrupt lines of a given INTC are controlled by the IRQ_CROSSBAR registers. This means, that there are interrupt lines which are not connected to any IRQ_CROSSBAR output and thus only one IRQ is connected to these interrupt lines. In the most common case these lines are used by interrupt sources internal or specific for the corresponding module, which has an INTC.

Table 13-12 shows which lines of each INTC are associated with the IRQ_CROSSBAR control registers. The rest of the INTC lines (not listed in the table) cannot be controlled by the cross-bar registers.

Table 13-12. Interrupt Lines Associated With The IRQ_CROSSBAR Control Registers

DSP INTC Lines	IPU INTC Lines	EVE INTC Lines
DSP1_IRQ_32 to DSP1_IRQ_95, DSP2_IRQ_32 to DSP2_IRQ_95	IPU_IRQ_23 to IPU_IRQ_80	EVE_IRQ_0 to EVE_IRQ_7

The individual connection between all module IRQs and all IRQ_CROSSBAR inputs is shown in Section 12.3.5, *Mapping of Device Interrupts to IRQ_CROSSBAR Inputs* of Chapter 12, *Interrupt Controllers*.

In addition, there are three IRQ_CROSSBAR registers associated with three channels of the device ESM. The registers are the following:

- CTRL_CORE_ESM_GROUP1_0
- CTRL_CORE_ESM_GROUP1_1
- CTRL_CORE_ESM_GROUP1_2

These three registers follow the same logic as previously described. For example, using the CTRL_CORE_ESM_GROUP1_0[8:0] ESM_GROUP1_0_IRQ_0 bit field any of the device interrupts connected to an IRQ_CROSSBAR input can be mapped to channel ESM_GROUP1_0.

NOTE: ESM is not supported on the DRA78x family of devices.

13.4.6.5 DMA_CROSSBAR Module Functional Description

There is a DMA_CROSSBAR module in the device, which is controlled by registers in the CTRL_MODULE_CORE submodule. The DMA_CROSSBAR is able to map any of its input signals to any of its outputs. This module is associated with the device DMA request (DREQ) source signals. The DREQs from all the device modules are connected to the DMA_CROSSBAR inputs. Each DREQ signal is connected only to one DMA cross-bar input. Each output of the DMA_CROSSBAR module is connected only to one input line of the device DMA modules. The DMAs associated with the DMA_CROSSBAR are the following:

- DMA_EDMA
- DMA_DSP1_EDMA
- DMA_DSP2_EDMA

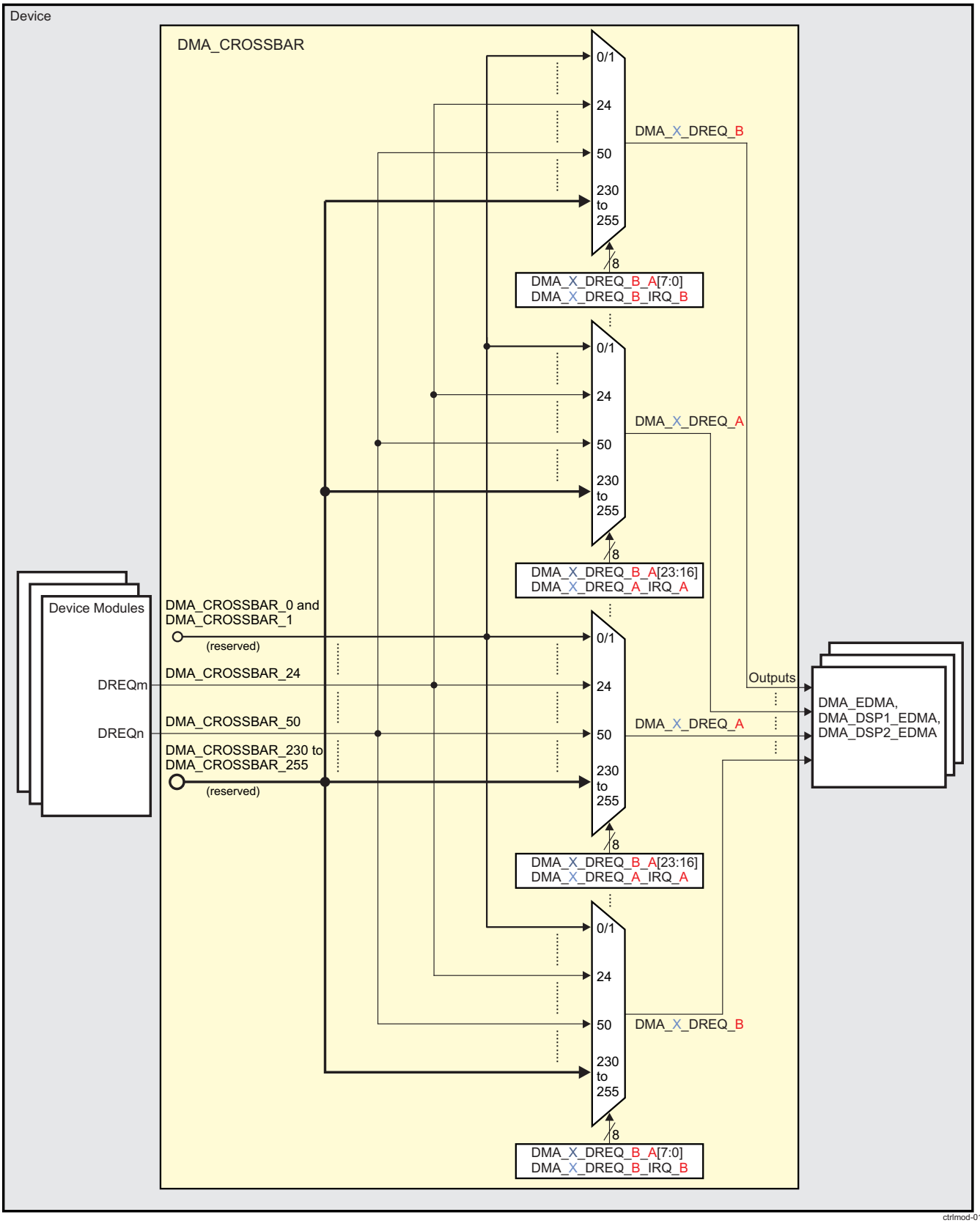
All DREQs, connected to the DMA_CROSSBAR inputs, can be remapped to other lines of the these DMA modules through the CTRL_CORE_DMA_X_DREQ_B_A registers. Each of these registers has a structure described in Table 13-13.

Table 13-13. Generic Description of the CTRL_CORE_DMA_X_DREQ_B_A DMA_CROSSBAR Control Registers

Bits	Field Name	Description	Type	Note
31:24	RESERVED		R	
23:16	DMA_X_DREQ_A_IRQ_A	Selects a DMA request source signal for the DMA_X_DREQ_A_IRQ_A DMA line 0x0: Reserved 0x1: Reserved 0x2: Maps DMA_CROSSBAR input 2 to DMA_X_DREQ_A_IRQ_A DMA line 0x3: Maps DMA_CROSSBAR input 3 to DMA_X_DREQ_A_IRQ_A DMA line 0x-: ..... 0x32: Maps DMA_CROSSBAR input 50 to DMA_X_DREQ_A_IRQ_A DMA line 0x-: ..... 0xE6 to 0xFF: Reserved	RW	X is summarization. It is equal to: • EDMA • DSP1_EDMA • DSP2_EDMA X shows to which DMA all inputs of the DMA_CROSSBAR module can be mapped to. A is also summarization. It shows the number of the DREQ line for the corresponding DMA module. A is equal to 0, 1, 2, ..., and so on, depending on the count of the DMA lines controlled by the DMA_CROSSBAR module. For more details, see Table 13-14.
15:8	RESERVED		R	
7:0	DMA_X_DREQ_B_IRQ_B	Selects a DMA request source signal for the DMA_X_DREQ_B_IRQ_B DMA line 0x0: Reserved 0x1: Reserved 0x2: Maps DMA_CROSSBAR input 2 to DMA_X_DREQ_B_IRQ_B DMA line 0x3: Maps DMA_CROSSBAR input 3 to DMA_X_DREQ_B_IRQ_B DMA line 0x-: ..... 0x32: Maps DMA_CROSSBAR input 50 to DMA_X_DREQ_B_IRQ_B DMA line 0x-: ..... 0xE6 to 0xFF: Reserved	RW	B is also summarization. It shows the number of the DREQ line for the corresponding DMA module. B is equal to 0, 1, 2, ..., and so on, depending on the count of the DMA lines controlled by the DMA_CROSSBAR module. For more details, see Table 13-14.

Figure 13-9 represents the way in which the DMA_CROSSBAR module works. It shows the device modules and their DREQs connected to the DMA_CROSSBAR inputs, the structure of the cross-bar and its outputs connected to the device DMA modules.

Figure 13-9. DMA_CROSSBAR Module Functional Diagram



Each DMA_CROSSBAR control register has two 8-bit fields. Each 8-bit field is associated only with one line of certain DMA module. Through this 8-bit field any of the DREQs connected to the DMA_CROSSBAR inputs can be mapped to the DMA line associated with this 8-bit field. For example, the register CTRL_CORE_DMA_EDMA_DREQ_48_49 is associated with DMA_EDMA_DREQ_48 and DMA_EDMA_DREQ_49 lines. The 8-bit field CTRL_CORE_DMA_EDMA_DREQ_48_49[7:0] DMA_EDMA_DREQ_48_IRQ_48 is associated only with DMA_EDMA_DREQ_48 line of the DMA_EDMA module. Setting this bit field to any other value different than its reset value will map another DREQ from the device modules to the DMA_EDMA_DREQ_48 line. The default (reset) value of this bit field is 0x31 which corresponds to the DMA_CROSSBAR_49 input. The UART1_DREQ_TX is connected to this cross-bar input. Setting another register to 0x31 will cause the UART1_DREQ_TX to be mapped to another DMA line. For example, if the CTRL_CORE_DMA_EDMA_DREQ_40_41[23:16] DMA_EDMA_DREQ_41_IRQ_41 is set to 0x31 the UART1_DREQ_TX will be mapped to the DMA_EDMA_DREQ_41 line. The same logic also applies to the other lines of the device DMAs.

Table 13-14 shows which lines of each DMA are associated with the DMA_CROSSBAR control registers. The rest of the lines (not listed in the table) cannot be controlled by the cross-bar registers.

Table 13-14. DREQ Lines Associated With The DMA_CROSSBAR Control Registers

DMA_EDMA DREQ Lines	DMA_DSP1_EDMA DREQ Lines	DMA_DSP2_EDMA DREQ Lines
DMA_EDMA_DREQ_0 to DMA_EDMA_DREQ_63	DMA_DSP1_DREQ_0 to DMA_DSP1_DREQ_19	DMA_DSP2_DREQ_0 to DMA_DSP2_DREQ_19

The individual connection between all module DREQs and all DMA_CROSSBAR inputs is shown in , *Mapping of DMA Requests to DMA_CROSSBAR Inputs* of Chapter 11, *Enhanced DMA Controllers*.

In addition, the CTRL_CORE_ESM_GROUP1_3 register can be used to map any of the device DREQs connected to a DMA_CROSSBAR input to channel ESM_GROUP1_3 of the device ESM. The same logic as previously described applies when configuring this register.

NOTE: ESM is not supported on the DRA78x family of devices.

13.4.6.6 SDRAM Initiator Priority Registers

The CTRL_CORE_EMIF_INITIATOR_PRIORITY_1 to CTRL_CORE_EMIF_INITIATOR_PRIORITY_8 registers are intended to control the priority of each initiator accessing the EMIF controller. Each 3-bit field in these registers is associated only with one initiator. Setting this bit field to 0x0 means that the corresponding initiator has a highest priority over the others and setting it to 0x7 is for lowest priority. This is useful in case of concurrent access to the external SDRAM from several initiators.

13.4.6.7 L3_MAIN Initiator Priority Registers

The CTRL_CORE_L3_INITIATOR_PRESSURE_1 register is used for controlling the priority of certain initiators on the L3_MAIN. Each 2-bit field in this register is associated only with one initiator. Setting this bit field to 0x3 means that the traffic of this initiator has highest proiroty over the other traffics. A value of 0x0 is for lowest priority. This register provides a dynamic priority escalation for the following L3_MAIN initiators:

- DSP1
- DSP2

13.4.6.8 Memory Region Lock Registers

There are five registers used to lock different memory regions of CTRL_MODULE_CORE memory mapped space. A memory region is locked, means that all wite accesses to this region are ignored. Writing a value unique for each register will lock certain memory region and writing another unique value results in unlocking of the same region. These five registers can lock the entire memory space of the CTRL_MODULE_CORE submodule. Table 13-15 gives more details.

Table 13-15. Memory Region Lock Registers

Register	Memory Space to Lock	Groups of Registers Associated With This Memory Region	Lock/Unlock (register reset value)
CTRL_CORE_MMR_LOCK_1	Region from 0x0000 0100 to 0x0000 079F and from 0x0000 1A00 to 0x0000 1FFF	Thermal management related registers, EMIF initiator priority, L3_MAIN initiator pressure (priority) and other registers	locked
CTRL_CORE_MMR_LOCK_2	Region from 0x0000 07A0 to 0x0000 0D9F	IRQ_CROSSBAR and DMA_CROSSBAR registers	locked
CTRL_CORE_MMR_LOCK_3	Region from 0x0000 0DA0 to 0x0000 0FFF	A few registers associated with device I/Os	locked
CTRL_CORE_MMR_LOCK_4	Region from 0x0000 1000 to 0x0000 13FF	Hardware observability related registers	locked
CTRL_CORE_MMR_LOCK_5	Region from 0x0000 1400 to 0x0000 19FF	Pad configuration registers	locked

NOTE: By default the entire CTRL_MODULE_CORE memory space is locked but the ROM code unlocks it by writing corresponding unlock values to all CTRL_CORE_MMR_LOCK_x registers.

13.4.6.9 NMI Mapping To Respective Cores

Two registers CTRL_CORE_NMI_DESTINATION_1 and CTRL_CORE_NMI_DESTINATION_2 are intended to map the external non-maskable interrupt (NMI) to certain of the device host processors. Writing 0x1 into a bit field of these registers enables the NMI to be mapped to the corresponding processor associated with this bit field. Writing 0x0 disables the NMI mapping to this processor.

13.4.6.10 Software Controls for the LPDDR2/DDR2/DDR3 I/O Cells

NOTE: LPDDR2 is not supported on the DRA78x family of devices.

There are two types of I/O cells associated with the LPDDR2/DDR2/DDR3 interface. These are single-ended and differential I/O cells.

These I/O cells have the following software controls which reside in registers of the CTRL_MODULE_CORE:

- Output impedance controls - I[2:0]
- Slew rate controls - SR[2:0]
- Weak driver controls - WD[1:0]

The I and SR controls apply when the I/Os operate as outputs. The WD controls apply when the I/Os operate as both inputs or outputs.

The bits I[2:0] are used for programming the desired impedance value of the output buffer. Table 13-16 describes the I[2:0] controls which are valid for pull-up and pull-down outputs.

Table 13-16. Output Impedance Controls - I[2:0]

I[2]	I[1]	I[0]	Drive Setting Name	Output Impedance
0	0	0	Imp80	80 Ohms
0	0	1	Imp60	60 Ohms
0	1	0	Imp48	48 Ohms
0	1	1	Imp40	40 Ohms
1	0	0	Imp34	34 Ohms
1	0	1	Reserved	Reserved

Table 13-16. Output Impedance Controls - I[2:0] (continued)

I[2]	I[1]	I[0]	Drive Setting Name	Output Impedance
1	1	0	Reserved	Reserved
1	1	1	Reserved	Reserved

To achieve optimal noise/speed trade-off the slew rate of the output signal can also be programmed using the slew rate control bits SR[2:0] shown in Table 13-17. These SR settings do not affect the DC drive strength of the output buffer. They only control its turn-on time.

Table 13-17. Slew Rate Controls - SR[2:0]

SR[2]	SR[1]	SR[0]	Turn-On Time Level	Note
0	0	0	fastest	All 8 values are valid.
...	...	...		
...	...	...		
1	1	1	slowest	

Weak pull-up, pull-down or keeper option for device LPDDR2/DDR2/DDR3 pads is enabled through the WD[1:0] bits. The weak pull-up or pull-down option is used to define the pad state (high or low) when no signal is driving the pad. The weak keeper option is used to maintain the previous output value when nothing is driving the pad. Table 13-18 describes the WD controls. They are used for avoiding floating pads on the LPDDR2/DDR2/DDR3 interface.

Table 13-18. Weak Driver Controls - WD[1:0]

WD[1]	WD[0]	Single-Ended Operation	Differential Pair Operation	
			padp	padn
0	0	Pull logic disabled	Pull logic disabled	Pull logic disabled
0	1	Weak pullup enabled	Weak pullup enabled	Weak pulldown enabled
1	0	Weak pulldown enabled	Weak pulldown enabled	Weak pullup enabled
1	1	Weak keeper enabled	Weak keeper enabled	Weak keeper enabled

NOTE: To avoid unnecessary power consumption, software must overwrite the Weak Driver (WD) reset values by setting them to 0x0.

It must be taken into account that the I, SR and WD software controls apply for several pads combined in groups, and not for a single pad. For example, writing 0x1 to the CTRL_CORE_CONTROL_DDRCACH1_0[17:16] DDR3CH1_PART5A_WD bit field enables the weak pull-up resistors for all pads in the PART5A group. These are the ddr1_a[15:0] pads. Table 13-19 shows the I, SR and WD controls for the different LPDDR2/DDR2/DDR3 pad groups.

Table 13-19. Software Group Controls for the LPDDR2/DDR2/DDR3 Pads

LPDDR2/DDR2/DDR3 Interface I/O Group Controls	Group Name	EMIF Pads in a Group
CTRL_CORE_CONTROL_DDRCACH1_0[31:29] DDR3CH1_PART0_I CTRL_CORE_CONTROL_DDRCACH1_0[28:26] DDR3CH1_PART0_SR CTRL_CORE_CONTROL_DDRCACH1_0[25:24] DDR3CH1_PART0_WD	PART0	ddr1_casn, ddr1_rasn, ddr1_rst, ddr1_wen, ddr1_csn[0], ddr1_cke, ddr1_odt[0]

Table 13-19. Software Group Controls for the LPDDR2/DDR2/DDR3 Pads (continued)

LPDDR2/DDR2/DDR3 Interface I/O Group Controls	Group Name	EMIF Pads in a Group
CTRL_CORE_CONTROL_DDRCACH1_0[23:21] DDR3CH1_PART5A_I CTRL_CORE_CONTROL_DDRCACH1_0[20:18] DDR3CH1_PART5A_SR CTRL_CORE_CONTROL_DDRCACH1_0[17:16] DDR3CH1_PART5A_WD	PART5A	ddr1_a[15:0]
CTRL_CORE_CONTROL_DDRCACH1_0[15:13] DDR3CH1_PART5B_I CTRL_CORE_CONTROL_DDRCACH1_0[12:10] DDR3CH1_PART5B_SR CTRL_CORE_CONTROL_DDRCACH1_0[9:8] DDR3CH1_PART5B_WD	PART5B	ddr1_ba[0], ddr1_ba[1], ddr1_ba[2]
CTRL_CORE_CONTROL_DDRCACH1_0[7:5] DDR3CH1_PART6_I CTRL_CORE_CONTROL_DDRCACH1_0[4:2] DDR3CH1_PART6_SR CTRL_CORE_CONTROL_DDRCACH1_0[1:0] DDR3CH1_PART6_WD	PART6	ddr1_ck, ddr1_nck
CTRL_CORE_CONTROL_DDRCH1_0[31:29] DDRCH1_PART1A_I CTRL_CORE_CONTROL_DDRCH1_0[28:26] DDRCH1_PART1A_SR CTRL_CORE_CONTROL_DDRCH1_0[25:24] DDRCH1_PART1A_WD	PART1A	ddr1_d[7:0], ddr1_dqm[0]
CTRL_CORE_CONTROL_DDRCH1_0[23:21] DDRCH1_PART1B_I CTRL_CORE_CONTROL_DDRCH1_0[20:18] DDRCH1_PART1B_SR CTRL_CORE_CONTROL_DDRCH1_0[17:16] DDRCH1_PART1B_WD	PART1B	ddr1_dqs[0], ddr1_dqsn[0]
CTRL_CORE_CONTROL_DDRCH1_0[15:13] DDRCH1_PART2A_I CTRL_CORE_CONTROL_DDRCH1_0[12:10] DDRCH1_PART2A_SR CTRL_CORE_CONTROL_DDRCH1_0[9:8] DDRCH1_PART2A_WD	PART2A	ddr1_d[15:8], ddr1_dqm[1]
CTRL_CORE_CONTROL_DDRCH1_0[7:5] DDRCH1_PART2B_I CTRL_CORE_CONTROL_DDRCH1_0[4:2] DDRCH1_PART2B_SR CTRL_CORE_CONTROL_DDRCH1_0[1:0] DDRCH1_PART2B_WD	PART2B	ddr1_dqs[1], ddr1_dqsn[1]
CTRL_CORE_CONTROL_DDRCH1_1[31:29] DDRCH1_PART3A_I CTRL_CORE_CONTROL_DDRCH1_1[28:26] DDRCH1_PART3A_SR CTRL_CORE_CONTROL_DDRCH1_1[25:24] DDRCH1_PART3A_WD	PART3A	ddr1_d[23:16], ddr1_dqm[2]
CTRL_CORE_CONTROL_DDRCH1_1[23:21] DDRCH1_PART3B_I CTRL_CORE_CONTROL_DDRCH1_1[20:18] DDRCH1_PART3B_SR CTRL_CORE_CONTROL_DDRCH1_1[17:16] DDRCH1_PART3B_WD	PART3B	ddr1_dqs[2], ddr1_dqsn[2]

Table 13-19. Software Group Controls for the LPDDR2/DDR2/DDR3 Pads (continued)

LPDDR2/DDR2/DDR3 Interface I/O Group Controls	Group Name	EMIF Pads in a Group
CTRL_CORE_CONTROL_DDRCH1_1[15:13] DDRCH1_PART4A_I CTRL_CORE_CONTROL_DDRCH1_1[12:10] DDRCH1_PART4A_SR CTRL_CORE_CONTROL_DDRCH1_1[9:8] DDRCH1_PART4A_WD	PART4A	ddr1_d[31:24], ddr1_dqm[3]
CTRL_CORE_CONTROL_DDRCH1_1[7:5] DDRCH1_PART4B_I CTRL_CORE_CONTROL_DDRCH1_1[4:2] DDRCH1_PART4B_SR CTRL_CORE_CONTROL_DDRCH1_1[1:0] DDRCH1_PART4B_WD	PART4B	ddr1_dqs[3], ddr1_dqsn[3]
CTRL_CORE_CONTROL_DDRCH1_2[23:21] DDRCH1_PART7A_I CTRL_CORE_CONTROL_DDRCH1_2[20:18] DDRCH1_PART7A_SR CTRL_CORE_CONTROL_DDRCH1_2[17:16] DDRCH1_PART7A_WD	PART7A	ddr1_ecc_d[7:0], ddr1_dqm_ecc
CTRL_CORE_CONTROL_DDRCH1_2[15:13] DDRCH1_PART7B_I CTRL_CORE_CONTROL_DDRCH1_2[12:10] DDRCH1_PART7B_SR CTRL_CORE_CONTROL_DDRCH1_2[9:8] DDRCH1_PART7B_WD	PART7B	ddr1_dqs_ecc, ddr1_dqsn_ecc

13.4.6.11 Reference Voltage for the Device LPDDR2/DDR2/DDR3 Receivers

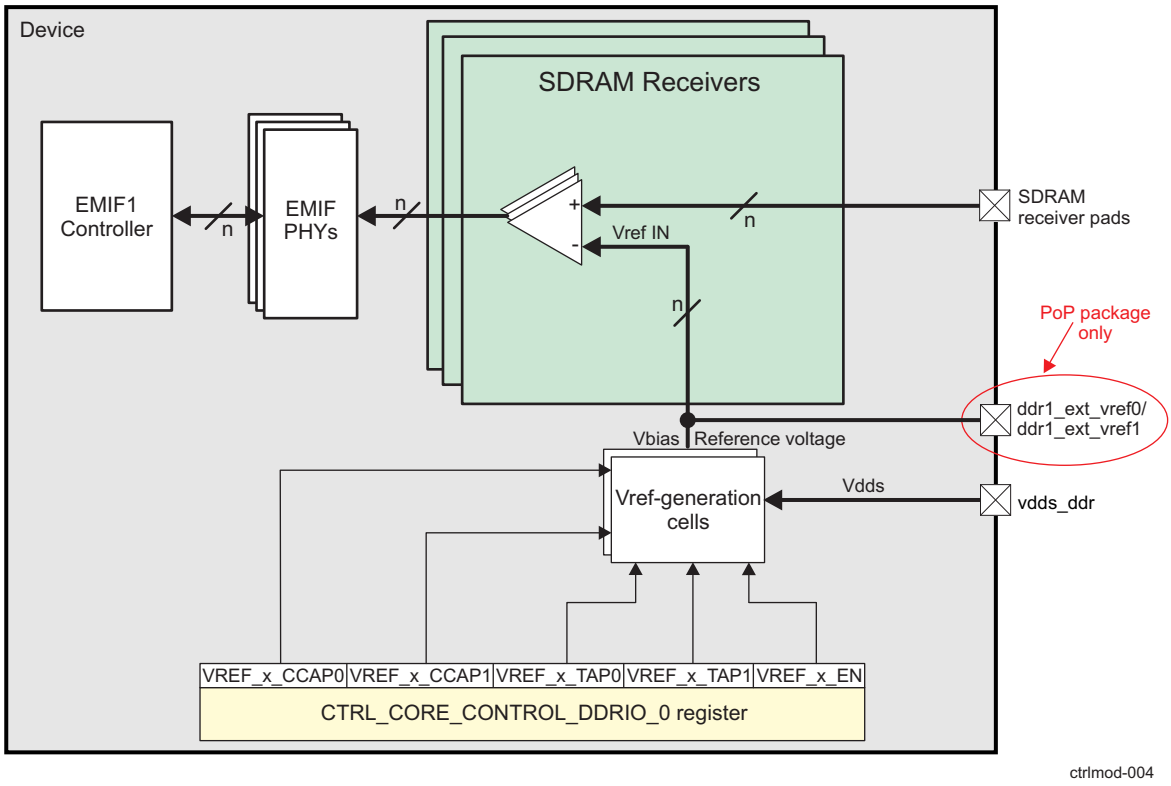
NOTE: LPDDR2 is not supported on the DRA78x family of devices.

The device LPDDR2/DDR2/DDR3 input buffers work in so-called Vref-based receiver mode. In this mode, the buffers act like differential comparators with positive terminal connected to a device pad which receives signals from LPDDR2/DDR2/DDR3 memory and negative terminal connected to a source of reference voltage.

To work properly, a reference voltage must be provided to the device LPDDR2/DDR2/DDR3 input buffers. There are two Vref-generation cells in the device intended to supply this internal reference voltage.

Figure 13-10 shows the Vref-generation cells in the device.

Figure 13-10. Vref-Generation Cells and Their Controls



The control bits for the Vref-generation cells reside in the CTRL_CORE_CONTROL_DDRIO_0 register. There are VREF_x_TAP[1:0] control bits which set the output drive capability of the Vref cells. Table 13-20 lists the possible options for selection of load current sourced from the output of each Vref-generation cell.

Table 13-20. Vref Cell Load Current Selection

VREF_x_TAP1	VREF_x_TAP0	Description
0	0	2-µA load current
0	1	4-µA load current
1	0	8-µA load current
1	1	32-µA load current

According to the noise environment, the user can choose to filter the supplied reference voltage. Two coupling capacitors internal to each Vref-generation cell are available and configurable through the VREF_x_CCAP[1:0] control bits in the CTRL_CORE_CONTROL_DDRIO_0 register as specified in Table 13-21.

Table 13-21. Vref Cell Coupling Capacitor Selection

VREF_x_CCAP1	VREF_x_CCAP0	Capacitor
0	0	No capacitor connected.
0	1	One capacitor connected between Vbias and ground.(1)
1	0	One capacitor connected between Vbias and Vdds.(2)
1	1	One capacitor connected between Vbias and ground and one capacitor connected between Vbias and Vdds.

(1) Vbias is the output of the Vref-generation cell which provides the reference voltage.
(2) Vdds is the power supply voltage of the Vref-generation cell.

The Vref-gereration cells can be enabled by setting to 0x1 the VREF_x_EN bits in the CTRL_CORE_CONTROL_DDRIO_0 register. These cells can be disabled (for leakage improvement and when not in use) by clearing the same VREF_x_EN bits.

Table 13-22 shows the Vref-generation cells control bits and the LPDDR2/DDR2/DDR3 pads used as receivers to which the corresponding Vref cell supplies reference voltage.

Table 13-22. Controls for the Vref-Generation Cells Versus LPDDR2/DDR2/DDR3 Receiver Pads

Vref-generation Cell Control Bits	LPDDR2/DDR2/DDR3 Vref Cell Associated Pads Used as Receivers
CTRL_CORE_CONTROL_DDRIO_0[19] DDRCH1_VREF_DQ0_INT_CCAP0	ddr1_d[7:0], ddr1_d[15:8]
CTRL_CORE_CONTROL_DDRIO_0[18] DDRCH1_VREF_DQ0_INT_CCAP1	
CTRL_CORE_CONTROL_DDRIO_0[17] DDRCH1_VREF_DQ0_INT_TAP0	
CTRL_CORE_CONTROL_DDRIO_0[16] DDRCH1_VREF_DQ0_INT_TAP1	
CTRL_CORE_CONTROL_DDRIO_0[15] DDRCH1_VREF_DQ0_INT_EN	
CTRL_CORE_CONTROL_DDRIO_0[14] DDRCH1_VREF_DQ1_INT_CCAP0	ddr1_d[23:16], ddr1_d[31:24], ddr1_ecc_d[7:0]
CTRL_CORE_CONTROL_DDRIO_0[13] DDRCH1_VREF_DQ1_INT_CCAP1	
CTRL_CORE_CONTROL_DDRIO_0[12] DDRCH1_VREF_DQ1_INT_TAP0	
CTRL_CORE_CONTROL_DDRIO_0[11] DDRCH1_VREF_DQ1_INT_TAP1	
CTRL_CORE_CONTROL_DDRIO_0[10] DDRCH1_VREF_DQ1_INT_EN	

13.4.6.12 AVS Class 0 Associated Registers

AVS Class 0 attempts to normalize the power consumption across all devices by lowering the operating voltage of certain voltage rails. This procedure of lowering the voltage should be performed in the boot loader after ROM code. The new voltage to be set for each AVS Class 0 supported voltage rail should be read from the eFuse using dedicated registers. Then the power supply output voltage is adjusted to this new voltage value.

The following voltage rails support AVS Class 0:

- vdd_dspeve
- vdd

NOTE: For descriptions of the voltage rails previously listed see the *"Power Supply Signal Descriptions"* table in the device Data Manual.

Table 13-23 shows all registers associated with AVS Class 0. The corresponding AVS Class 0 voltage value can be read from the 12 LSbits of each of the listed registers. They contain the voltage value in hex format. To find the actual value in mV a conversion from hex to decimal value is needed. For example, if the value read from CTRL_CORE_STD_FUSE_OPP_VMIN_CORE_2[11:0] STD_FUSE_OPP_VMIN_CORE_2 is 0x041F, then this corresponds to 1055 mV.

Table 13-23. Registers Associated With AVS Class 0 Voltage

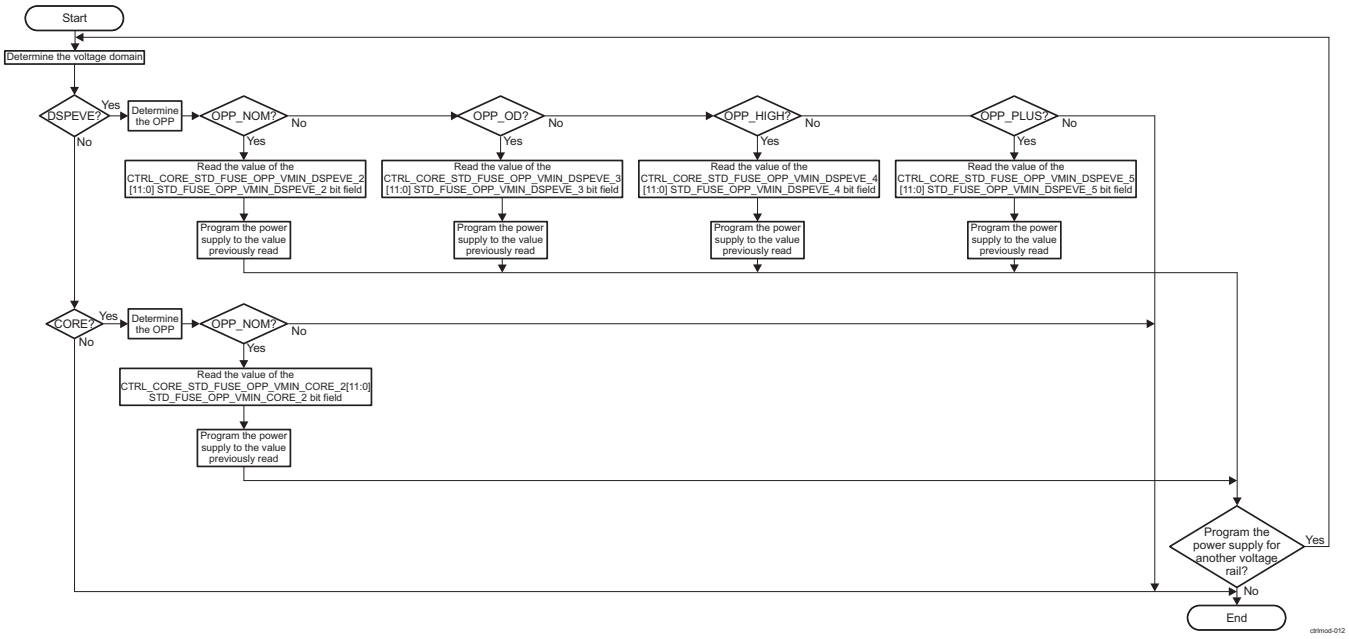
Physical Address	Bit Field Containing the AVS Class 0 Voltage Value	Voltage Rail	Supported OPP
0x4A00 25E0	CTRL_CORE_STD_FUSE_OPP_VMIN_DSPEVE_2[11:0] STD_FUSE_OPP_VMIN_DSPEVE_2	vdd_dspeve	OPP_NOM
0x4A00 25E4	CTRL_CORE_STD_FUSE_OPP_VMIN_DSPEVE_3[11:0] STD_FUSE_OPP_VMIN_DSPEVE_3		OPP_OD
0x4A00 25E8	CTRL_CORE_STD_FUSE_OPP_VMIN_DSPEVE_4[11:0] STD_FUSE_OPP_VMIN_DSPEVE_4		OPP_HIGH
0x4A00 25D8	CTRL_CORE_STD_FUSE_OPP_VMIN_DSPEVE_5[11:0] STD_FUSE_OPP_VMIN_DSPEVE_5		OPP_PLUS
0x4A00 25F4	CTRL_CORE_STD_FUSE_OPP_VMIN_CORE_2[11:0] STD_FUSE_OPP_VMIN_CORE_2	vdd	OPP_NOM

NOTE: Some of the OPPs listed in Table 13-23 may not be supported for some devices. In these cases the voltage values in the corresponding AVS Class 0 registers can be disregarded. For more information about the supported OPPs, see the "*Operating Performance Points*" section of the device Data Manual.

In some cases, the AVS Class 0 voltage that is read from the CTRL_CORE_STD_FUSE_OPP_VMIN_xxx_y registers has a value between two incremental voltage steps of the power supply. If such a case occurs, the higher voltage value should be selected.

Figure 13-11 shows a general example of how AVS Class 0 should be performed.

Figure 13-11. AVS Class 0 Procedure



13.4.6.13 Registers For Other Miscellaneous Functions

13.4.6.13.1 System Boot Status Settings

The CTRL_CORE_BOOTSTRAP register is a status register which indicates the state of the sysboot0 through sysboot15 input signals. Their purpose is to select the boot interface, the device source clock configuration and also other boot related settings.

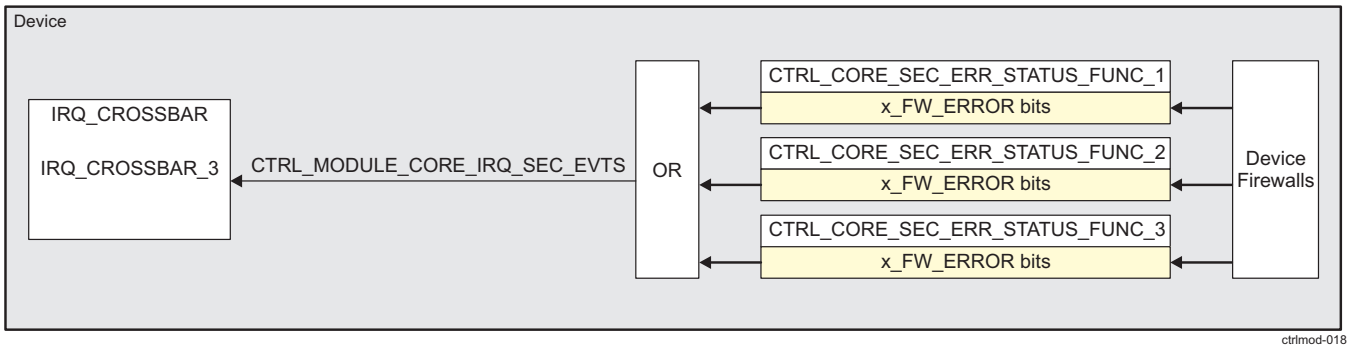
NOTE: For proper device operation, sysboot15 must be tied to vdd and sysboot14 must be tied to vss.

13.4.6.13.2 Firewall Error Status Registers

There are several status registers which show when there is a firewall error. The CTRL_CORE_SEC_ERR_STATUS_FUNC_1, CTRL_CORE_SEC_ERR_STATUS_FUNC_2 and CTRL_CORE_SEC_ERR_STATUS_FUNC_3 registers are used in device normal operation mode. The x_FW_ERROR bits from these three registers are combined into a single interrupt signal sent to the IRQ_CROSSBAR module as shown in Figure 13-12. The CTRL_CORE_SEC_ERR_STATUS_DEBUG_1, CTRL_CORE_SEC_ERR_STATUS_DEBUG_2 and CTRL_CORE_SEC_ERR_STATUS_DEBUG_3 registers are used when the device is in debug mode. These registers have same bits as in the CTRL_CORE_SEC_ERR_STATUS_FUNC_1, CTRL_CORE_SEC_ERR_STATUS_FUNC_2 and CTRL_CORE_SEC_ERR_STATUS_FUNC_3 registers.

All bits in these registers are cleared when the ERROR_LOG_k and L4_IA_ERROR_LOG_L registers are cleared.

Figure 13-12. Combined Firewall Error Interrupt



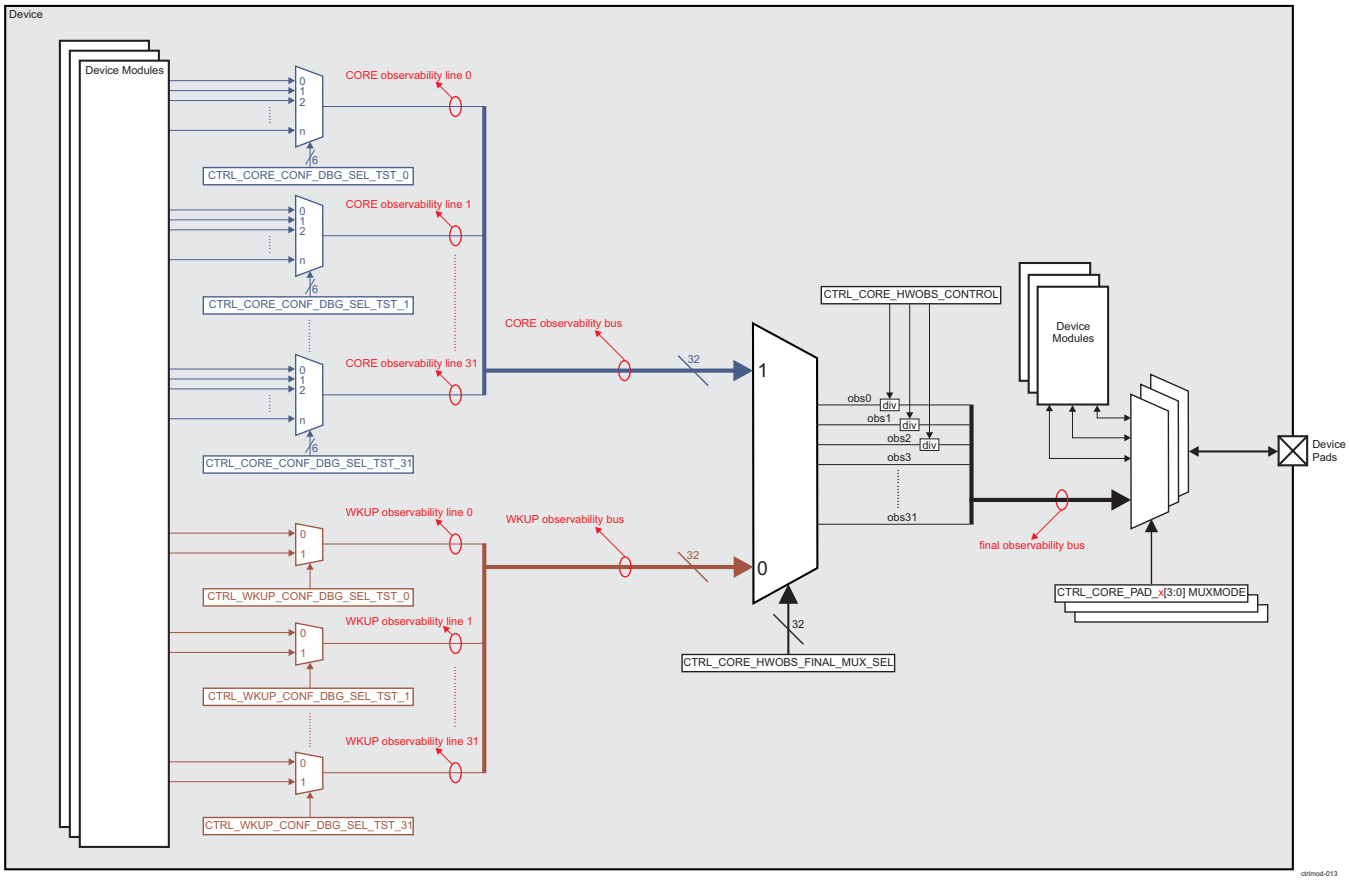
13.4.6.13.3 Settings Related To Different Peripheral Modules

The CTRL_CORE_CONTROL_IO_1 and CTRL_CORE_CONTROL_IO_2 registers have controls for specific settings of several device peripheral modules. For more information, see the bit field descriptions.

13.4.6.14 Hardware Observability Related Registers

There is a set of registers which are associated with the device observability logic. Using these registers a desired module signal can be mapped to dedicated-for-observation pad and then this signal can be observed. Figure 13-13 shows the device observability logic.

Figure 13-13. Hardware Observability Logic



As can be seen on Figure 13-13 there are two 32-bit wide observability buses. One associated with signals in CORE and one with signals in WKUP power domain. The multiplexing of the CORE observable signals is controlled by CTRL_CORE_CONF_DEBUG_SEL_TST_0 through CTRL_CORE_CONF_DEBUG_SEL_TST_31 registers. The multiplexing of the WKUP observable signals is controlled by CTRL_WKUP_CONF_DEBUG_SEL_TST_0 through CTRL_WKUP_CONF_DEBUG_SEL_TST_31 registers. Each of these registers is associated with one observability line. For example, the CTRL_CORE_CONF_DEBUG_SEL_TST_0 register selects one of N observability signals to be mapped to CORE observability line 0.

The CTRL_CORE_HWOBS_FINAL_MUX_SEL register is used to control the final multiplexer which selects CORE or WKUP observable signals to be mapped to the final observability bus. This bus is composed of obs0 through obs31 lines. Each bit in the CTRL_CORE_HWOBS_FINAL_MUX_SEL register is associated with one of these lines. A value of 0x0 selects a WKUP observable signal and 0x1 select CORE observable signal.

Optionally, the frequency of obs0, obs1 and obs2 lines can be divided through the bit fields HWOBS_CLKDIV_SEL, HWOBS_CLKDIV_SEL_1 and HWOBS_CLKDIV_SEL_2 from register CTRL_CORE_HWOBS_CONTROL. This register provides also an option to force all 32 signals (obs0 through obs31) of the observability bus to 0 or to 1 setting to 0x1 bit HWOBS_ALL_ZERO_MODE or bit HWOBS_ALL_ONE_MODE, respectively.

13.4.7 Functional Description Of The Various Register Types In CTRL_MODULE_WKUP Submodule

The following sections describe the purpose of the registers which reside in the CTRL_MODULE_WKUP submodule.

13.4.7.1 Registers For Basic EMIF configuration

The CTRL_WKUP_SECURE_EMIF1_SDRAM_CONFIG register has bits which determine the basic settings of the EMIF controller. These are, for example, settings like CAS write latency, SDRAM drive strength, SDRAM termination resistor values, SDRAM type and others. The CTRL_WKUP_SECURE_EMIF1_SDRAM_CONFIG bit field values are exported upon POR to the EMIF.EMIF_SDRAM_CONFIG register.

The CTRL_WKUP_EMIF1_SDRAM_CONFIG_EXT register is associated with DDR PHY controls, ODT values for device DDR I/Os, some leveling related parameters and others.

13.5 Control Module Register Manual

13.5.1 Control Module Instance Summary

Table 13-24. CONTROL MODULE Instance Summary

Module Name	Module Base Address	Size
CTRL_MODULE_CORE	0x4A00 2000	8 KiB
CTRL_MODULE_WKUP	0x4AE0 C000	4 KiB

13.5.2 CTRL_MODULE_CORE Registers

13.5.2.1 CTRL_MODULE_CORE Register Summary

Table 13-25. CTRL_MODULE_CORE Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	CTRL_MODULE_CORE Base Address
CTRL_CORE_STATUS	R	32	0x0000 0134	0x4A00 2134
RESERVED	R	32	0x0000 0138	0x4A00 2138
RESERVED	R	32	0x0000 013C	0x4A00 213C
RESERVED	R	32	0x0000 0140	0x4A00 2140
RESERVED	R	32	0x0000 0144	0x4A00 2144
CTRL_CORE_SEC_ERR_STATUS_FUNC_1	RW	32	0x0000 0148	0x4A00 2148
RESERVED	R	32	0x0000 014C	0x4A00 214C
CTRL_CORE_SEC_ERR_STATUS_DEBUG_1	RW	32	0x0000 0150	0x4A00 2150
RESERVED	R	32	0x0000 0154	0x4A00 2154
RESERVED	R	32	0x0000 01C8	0x4A00 21C8
CTRL_CORE_STD_FUSE_OPP_VDD_CORE_0	R	32	0x0000 01CC	0x4A00 21CC
CTRL_CORE_STD_FUSE_OPP_VDD_CORE_1	R	32	0x0000 01D0	0x4A00 21D0
CTRL_CORE_STD_FUSE_OPP_VDD_CORE_2	R	32	0x0000 01D4	0x4A00 21D4
CTRL_CORE_STD_FUSE_OPP_VDD_CORE_3	R	32	0x0000 01D8	0x4A00 21D8
CTRL_CORE_STD_FUSE_OPP_VDD_CORE_4	R	32	0x0000 01DC	0x4A00 21DC
RESERVED	R	32	0x0000 01E0	0x4A00 21E0
RESERVED	R	32	0x0000 01E4	0x4A00 21E4
CTRL_CORE_STD_FUSE_OPP_BGAP_CORE	R	32	0x0000 01E8	0x4A00 21E8
RESERVED_b (b = 0 to 12)	R	32	0x0000 01EC + (b*4)	0x4A00 21EC + (b*4)
CTRL_CORE_STD_FUSE_MPK_0	R	32	0x0000 0220	0x4A00 2220
CTRL_CORE_STD_FUSE_MPK_1	R	32	0x0000 0224	0x4A00 2224
CTRL_CORE_STD_FUSE_MPK_2	R	32	0x0000 0228	0x4A00 2228
CTRL_CORE_STD_FUSE_MPK_3	R	32	0x0000 022C	0x4A00 222C
CTRL_CORE_STD_FUSE_MPK_4	R	32	0x0000 0230	0x4A00 2230
CTRL_CORE_STD_FUSE_MPK_5	R	32	0x0000 0234	0x4A00 2234

Table 13-25. CTRL_MODULE_CORE Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	CTRL_MODULE_CORE Base Address
CTRL_CORE_STD_FUSE_MPK_6	R	32	0x0000 0238	0x4A00 2238
CTRL_CORE_STD_FUSE_MPK_7	R	32	0x0000 023C	0x4A00 223C
RESERVED_c (c = 0 to 30)	R	32	0x0000 0240 + (c*4)	0x4A00 2240 + (c*4)
CTRL_CORE_CUST_FUSE_SWRV_0	R	32	0x0000 02BC	0x4A00 22BC
CTRL_CORE_CUST_FUSE_SWRV_1	R	32	0x0000 02C0	0x4A00 22C0
CTRL_CORE_CUST_FUSE_SWRV_2	R	32	0x0000 02C4	0x4A00 22C4
CTRL_CORE_CUST_FUSE_SWRV_3	R	32	0x0000 02C8	0x4A00 22C8
CTRL_CORE_CUST_FUSE_SWRV_4	R	32	0x0000 02CC	0x4A00 22CC
CTRL_CORE_CUST_FUSE_SWRV_5	R	32	0x0000 02D0	0x4A00 22D0
CTRL_CORE_CUST_FUSE_SWRV_6	R	32	0x0000 02D4	0x4A00 22D4
RESERVED	R	32	0x0000 02D8	0x4A00 22D8
RESERVED	R	32	0x0000 02DC	0x4A00 22DC
RESERVED	R	32	0x0000 02E0	0x4A00 22E0
RESERVED	R	32	0x0000 02E4	0x4A00 22E4
RESERVED	R	32	0x0000 02E8	0x4A00 22E8
RESERVED	R	32	0x0000 02EC	0x4A00 22EC
RESERVED	R	32	0x0000 0300	0x4A00 2300
RESERVED	R	32	0x0000 0304	0x4A00 2304
RESERVED	R	32	0x0000 032C	0x4A00 232C
RESERVED	R	32	0x0000 0330	0x4A00 2330
CTRL_CORE_TEMP_SENSOR_CORE	R	32	0x0000 0334	0x4A00 2334
RESERVED	R	32	0x0000 033C	0x4A00 233C
RESERVED	R	32	0x0000 0340	0x4A00 2340
RESERVED	R	32	0x0000 0344	0x4A00 2344
CTRL_CORE_CORTEX_M4_MMUADDRTRANSLTR	RW	32	0x0000 0358	0x4A00 2358
CTRL_CORE_CORTEX_M4_MMUADDRLOGICTR	RW	32	0x0000 035C	0x4A00 235C
CTRL_CORE_HWOBS_CONTROL	RW	32	0x0000 0360	0x4A00 2360
RESERVED	R	32	0x0000 0364	0x4A00 2364
RESERVED	R	32	0x0000 0368	0x4A00 2368
RESERVED	R	32	0x0000 036C	0x4A00 236C
RESERVED	R	32	0x0000 0370	0x4A00 2370
RESERVED	R	32	0x0000 0374	0x4A00 2374
CTRL_CORE_BANDGAP_MASK_1	RW	32	0x0000 0380	0x4A00 2380
RESERVED	R	32	0x0000 0384	0x4A00 2384
RESERVED	R	32	0x0000 0388	0x4A00 2388
CTRL_CORE_BANDGAP_THRESHOLD_CORE	RW	32	0x0000 038C	0x4A00 238C
RESERVED	R	32	0x0000 0390	0x4A00 2390
RESERVED	R	32	0x0000 0394	0x4A00 2394
CTRL_CORE_BANDGAP_TSHUT_CORE	RW	32	0x0000 0398	0x4A00 2398
RESERVED	R	32	0x0000 039C	0x4A00 239C
RESERVED	R	32	0x0000 03A0	0x4A00 23A0
RESERVED	R	32	0x0000 03A4	0x4A00 23A4
CTRL_CORE_BANDGAP_STATUS_1	R	32	0x0000 03A8	0x4A00 23A8
RESERVED	R	32	0x0000 03AC	0x4A00 23AC
RESERVED	R	32	0x0000 03E4	0x4A00 23E4
CTRL_CORE_DTEMP_CORE_0	R	32	0x0000 03E8	0x4A00 23E8
CTRL_CORE_DTEMP_CORE_1	R	32	0x0000 03EC	0x4A00 23EC
CTRL_CORE_DTEMP_CORE_2	R	32	0x0000 03F0	0x4A00 23F0
CTRL_CORE_DTEMP_CORE_3	R	32	0x0000 03F4	0x4A00 23F4

Table 13-25. CTRL_MODULE_CORE Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	CTRL_MODULE_CORE Base Address
CTRL_CORE_DTEMP_CORE_4	R	32	0x0000 03F8	0x4A00 23F8
CTRL_CORE_SMA_SW_0	RW	32	0x0000 03FC	0x4A00 23FC
RESERVED	R	32	0x0000 0400	0x4A00 2400
RESERVED	R	32	0x0000 0404	0x4A00 2404
RESERVED	R	32	0x0000 0408	0x4A00 2408
RESERVED	R	32	0x0000 040C	0x4A00 240C
CTRL_CORE_SEC_ERR_STATUS_FUNC_2	RW	32	0x0000 0414	0x4A00 2414
RESERVED	R	32	0x0000 0418	0x4A00 2418
CTRL_CORE_SEC_ERR_STATUS_DEBUG_2	RW	32	0x0000 041C	0x4A00 241C
CTRL_CORE_EMIF_INITIATOR_PRIORITY_1	RW	32	0x0000 0420	0x4A00 2420
CTRL_CORE_EMIF_INITIATOR_PRIORITY_2	RW	32	0x0000 0424	0x4A00 2424
CTRL_CORE_EMIF_INITIATOR_PRIORITY_3	RW	32	0x0000 0428	0x4A00 2428
CTRL_CORE_EMIF_INITIATOR_PRIORITY_4	RW	32	0x0000 042C	0x4A00 242C
CTRL_CORE_EMIF_INITIATOR_PRIORITY_5	RW	32	0x0000 0430	0x4A00 2430
CTRL_CORE_EMIF_INITIATOR_PRIORITY_6	RW	32	0x0000 0434	0x4A00 2434
RESERVED	R	32	0x0000 0438	0x4A00 2438
CTRL_CORE_L3_INITIATOR_PRESSURE_1	RW	32	0x0000 043C	0x4A00 243C
RESERVED_d (d = 0 to 41)	R	32	0x0000 0440 + (c*4)	0x4A00 2440 + (c*4)
CTRL_CORE_CUST_FUSE_UID_0	R	32	0x0000 04E8	0x4A00 24E8
CTRL_CORE_CUST_FUSE_UID_1	R	32	0x0000 04EC	0x4A00 24EC
CTRL_CORE_CUST_FUSE_UID_2	R	32	0x0000 04F0	0x4A00 24F0
CTRL_CORE_CUST_FUSE_UID_3	R	32	0x0000 04F4	0x4A00 24F4
CTRL_CORE_CUST_FUSE_UID_4	R	32	0x0000 04F8	0x4A00 24F8
CTRL_CORE_CUST_FUSE_UID_5	R	32	0x0000 04FC	0x4A00 24FC
CTRL_CORE_CUST_FUSE_UID_6	R	32	0x0000 0500	0x4A00 2500
RESERVED	R	32	0x0000 0504	0x4A00 2504
RESERVED	R	32	0x0000 0508	0x4A00 2508
RESERVED	R	32	0x0000 050C	0x4A00 250C
RESERVED	R	32	0x0000 0510	0x4A00 2510
CTRL_CORE_MAC_ID_SW_0	R	32	0x0000 0514	0x4A00 2514
CTRL_CORE_MAC_ID_SW_1	R	32	0x0000 0518	0x4A00 2518
CTRL_CORE_MAC_ID_SW_2	R	32	0x0000 051C	0x4A00 251C
CTRL_CORE_MAC_ID_SW_3	R	32	0x0000 0520	0x4A00 2520
RESERVED_d (d = 0 to 3)	R	32	0x0000 0524 + (d*4)	0x4A00 2524 + (d*4)
CTRL_CORE_SMA_SW_1	RW	32	0x0000 0534	0x4A00 2534
RESERVED	R	32	0x0000 0538	0x4A00 2538
CTRL_CORE_EMIF_INITIATOR_PRIORITY_8	RW	32	0x0000 053C	0x4A00 253C
CTRL_CORE_MMR_LOCK_1	RW	32	0x0000 0540	0x4A00 2540
CTRL_CORE_MMR_LOCK_2	RW	32	0x0000 0544	0x4A00 2544
CTRL_CORE_MMR_LOCK_3	RW	32	0x0000 0548	0x4A00 2548
CTRL_CORE_MMR_LOCK_4	RW	32	0x0000 054C	0x4A00 254C
CTRL_CORE_MMR_LOCK_5	RW	32	0x0000 0550	0x4A00 2550
CTRL_CORE_CONTROL_IO_1	RW	32	0x0000 0554	0x4A00 2554
CTRL_CORE_CONTROL_IO_2	RW	32	0x0000 0558	0x4A00 2558
CTRL_CORE_CONTROL_DSP1_RST_VECT	RW	32	0x0000 055C	0x4A00 255C
CTRL_CORE_CONTROL_DSP2_RST_VECT	RW	32	0x0000 0560	0x4A00 2560
CTRL_CORE_STD_FUSE_OPP_BGAP_DSPEVE	R	32	0x0000 0564	0x4A00 2564
RESERVED	R	32	0x0000 0568	0x4A00 2568
CTRL_CORE_LDOSRAM_DSPEVE_VOLTAGE_CTRL	RW	32	0x0000 056C	0x4A00 256C

Table 13-25. CTRL_MODULE_CORE Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	CTRL_MODULE_CORE Base Address
RESERVED	R	32	0x0000 0570	0x4A00 2570
RESERVED	R	32	0x0000 05D4	0x4A00 25D4
CTRL_CORE_STD_FUSE_OPP_VMIN_DSPEVE_5	R	32	0x0000 05D8	0x4A00 25D8
RESERVED	R	32	0x0000 05DC	0x4A00 25DC
CTRL_CORE_STD_FUSE_OPP_VMIN_DSPEVE_2	R	32	0x0000 05E0	0x4A00 25E0
CTRL_CORE_STD_FUSE_OPP_VMIN_DSPEVE_3	R	32	0x0000 05E4	0x4A00 25E4
CTRL_CORE_STD_FUSE_OPP_VMIN_DSPEVE_4	R	32	0x0000 05E8	0x4A00 25E8
CTRL_CORE_RC_OSC_FREQUENCY	R	32	0x0000 05EC	0x4A00 25EC
RESERVED	R	32	0x0000 05F0	0x4A00 25F0
CTRL_CORE_STD_FUSE_OPP_VMIN_CORE_2	R	32	0x0000 05F4	0x4A00 25F4
RESERVED_e (e = 0 to 33)	R	32	0x0000 05F8 + (e*4)	0x4A00 25F8 + (e*4)
CTRL_CORE_LDOSRAM_CORE_2_VOLTAGE_CTRL	RW	32	0x0000 0680	0x4A00 2680
CTRL_CORE_LDOSRAM_CORE_3_VOLTAGE_CTRL	RW	32	0x0000 0684	0x4A00 2684
RESERVED	R	32	0x0000 0688	0x4A00 2688
CTRL_CORE_NMI_DESTINATION_1	RW	32	0x0000 068C	0x4A00 268C
CTRL_CORE_NMI_DESTINATION_2	RW	32	0x0000 0690	0x4A00 2690
RESERVED	R	32	0x0000 0694	0x4A00 2694
RESERVED	R	32	0x0000 0698	0x4A00 2698
RESERVED	R	32	0x0000 069C	0x4A00 269C
CTRL_CORE_STD_FUSE_OPP_VDD_DSPEVE_0	R	32	0x0000 06A0	0x4A00 26A0
CTRL_CORE_STD_FUSE_OPP_VDD_DSPEVE_1	R	32	0x0000 06A4	0x4A00 26A4
CTRL_CORE_STD_FUSE_OPP_VDD_DSPEVE_2	R	32	0x0000 06A8	0x4A00 26A8
CTRL_CORE_STD_FUSE_OPP_VDD_DSPEVE_3	R	32	0x0000 06AC	0x4A00 26AC
CTRL_CORE_STD_FUSE_OPP_VDD_DSPEVE_4	R	32	0x0000 06B0	0x4A00 26B0
CTRL_CORE_CUST_FUSE_SWRV_7	R	32	0x0000 06B4	0x4A00 26B4
CTRL_CORE_STD_FUSE_CALIBRATION_OVERRIDE_VALUE_0	R	32	0x0000 06B8	0x4A00 26B8
CTRL_CORE_STD_FUSE_CALIBRATION_OVERRIDE_VALUE_1	R	32	0x0000 06BC	0x4A00 26BC
RESERVED	R	32	0x0000 06C0	0x4A00 26C0
CTRL_CORE_BOOTSTRAP	R	32	0x0000 06C4	0x4A00 26C4
RESERVED_f (f = 0 to 50)	R	32	0x0000 06C8 + (f*4)	0x4A00 26C8 + (f*4)
CTRL_CORE_SEC_ERR_STATUS_FUNC_3	RW	32	0x0000 0794	0x4A00 2794
RESERVED	R	32	0x0000 0798	0x4A00 2798
CTRL_CORE_SEC_ERR_STATUS_DEBUG_3	RW	32	0x0000 079C	0x4A00 279C
CTRL_CORE_EVE_IRQ_0_1	RW	32	0x0000 07A0	0x4A00 27A0
CTRL_CORE_EVE_IRQ_2_3	RW	32	0x0000 07A4	0x4A00 27A4
CTRL_CORE_EVE_IRQ_4_5	RW	32	0x0000 07A8	0x4A00 27A8
CTRL_CORE_EVE_IRQ_6_7	RW	32	0x0000 07AC	0x4A00 27AC
RESERVED_g (g = 0 to 11)	R	32	0x0000 07B0 + (g*4)	0x4A00 27B0 + (g*4)
CTRL_CORE_IPU_IRQ_23_24	RW	32	0x0000 07E0	0x4A00 27E0
CTRL_CORE_IPU_IRQ_25_26	RW	32	0x0000 07E4	0x4A00 27E4
CTRL_CORE_IPU_IRQ_27_28	RW	32	0x0000 07E8	0x4A00 27E8
CTRL_CORE_IPU_IRQ_29_30	RW	32	0x0000 07EC	0x4A00 27EC
CTRL_CORE_IPU_IRQ_31_32	RW	32	0x0000 07F0	0x4A00 27F0
CTRL_CORE_IPU_IRQ_33_34	RW	32	0x0000 07F4	0x4A00 27F4
CTRL_CORE_IPU_IRQ_35_36	RW	32	0x0000 07F8	0x4A00 27F8
CTRL_CORE_IPU_IRQ_37_38	RW	32	0x0000 07FC	0x4A00 27FC
CTRL_CORE_IPU_IRQ_39_40	RW	32	0x0000 0800	0x4A00 2800
CTRL_CORE_IPU_IRQ_41_42	RW	32	0x0000 0804	0x4A00 2804

Table 13-25. CTRL_MODULE_CORE Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	CTRL_MODULE_CORE Base Address
CTRL_CORE_IPU_IRQ_43_44	RW	32	0x0000 0808	0x4A00 2808
CTRL_CORE_IPU_IRQ_45_46	RW	32	0x0000 080C	0x4A00 280C
CTRL_CORE_IPU_IRQ_47_48	RW	32	0x0000 0810	0x4A00 2810
CTRL_CORE_IPU_IRQ_49_50	RW	32	0x0000 0814	0x4A00 2814
CTRL_CORE_IPU_IRQ_51_52	RW	32	0x0000 0818	0x4A00 2818
CTRL_CORE_IPU_IRQ_53_54	RW	32	0x0000 081C	0x4A00 281C
CTRL_CORE_IPU_IRQ_55_56	RW	32	0x0000 0820	0x4A00 2820
CTRL_CORE_IPU_IRQ_57_58	RW	32	0x0000 0824	0x4A00 2824
CTRL_CORE_IPU_IRQ_59_60	RW	32	0x0000 0828	0x4A00 2828
CTRL_CORE_IPU_IRQ_61_62	RW	32	0x0000 082C	0x4A00 282C
CTRL_CORE_IPU_IRQ_63_64	RW	32	0x0000 0830	0x4A00 2830
CTRL_CORE_IPU_IRQ_65_66	RW	32	0x0000 0834	0x4A00 2834
CTRL_CORE_IPU_IRQ_67_68	RW	32	0x0000 0838	0x4A00 2838
CTRL_CORE_IPU_IRQ_69_70	RW	32	0x0000 083C	0x4A00 283C
CTRL_CORE_IPU_IRQ_71_72	RW	32	0x0000 0840	0x4A00 2840
CTRL_CORE_IPU_IRQ_73_74	RW	32	0x0000 0844	0x4A00 2844
CTRL_CORE_IPU_IRQ_75_76	RW	32	0x0000 0848	0x4A00 2848
CTRL_CORE_IPU_IRQ_77_78	RW	32	0x0000 084C	0x4A00 284C
CTRL_CORE_IPU_IRQ_79_80	RW	32	0x0000 0850	0x4A00 2850
RESERVED_h (h = 0 to 60)	R	32	0x0000 0854 + (h*4)	0x4A00 2854 + (h*4)
CTRL_CORE_DSP1_IRQ_32_33	RW	32	0x0000 0948	0x4A00 2948
CTRL_CORE_DSP1_IRQ_34_35	RW	32	0x0000 094C	0x4A00 294C
CTRL_CORE_DSP1_IRQ_36_37	RW	32	0x0000 0950	0x4A00 2950
CTRL_CORE_DSP1_IRQ_38_39	RW	32	0x0000 0954	0x4A00 2954
CTRL_CORE_DSP1_IRQ_40_41	RW	32	0x0000 0958	0x4A00 2958
CTRL_CORE_DSP1_IRQ_42_43	RW	32	0x0000 095C	0x4A00 295C
CTRL_CORE_DSP1_IRQ_44_45	RW	32	0x0000 0960	0x4A00 2960
CTRL_CORE_DSP1_IRQ_46_47	RW	32	0x0000 0964	0x4A00 2964
CTRL_CORE_DSP1_IRQ_48_49	RW	32	0x0000 0968	0x4A00 2968
CTRL_CORE_DSP1_IRQ_50_51	RW	32	0x0000 096C	0x4A00 296C
CTRL_CORE_DSP1_IRQ_52_53	RW	32	0x0000 0970	0x4A00 2970
CTRL_CORE_DSP1_IRQ_54_55	RW	32	0x0000 0974	0x4A00 2974
CTRL_CORE_DSP1_IRQ_56_57	RW	32	0x0000 0978	0x4A00 2978
CTRL_CORE_DSP1_IRQ_58_59	RW	32	0x0000 097C	0x4A00 297C
CTRL_CORE_DSP1_IRQ_60_61	RW	32	0x0000 0980	0x4A00 2980
CTRL_CORE_DSP1_IRQ_62_63	RW	32	0x0000 0984	0x4A00 2984
CTRL_CORE_DSP1_IRQ_64_65	RW	32	0x0000 0988	0x4A00 2988
CTRL_CORE_DSP1_IRQ_66_67	RW	32	0x0000 098C	0x4A00 298C
CTRL_CORE_DSP1_IRQ_68_69	RW	32	0x0000 0990	0x4A00 2990
CTRL_CORE_DSP1_IRQ_70_71	RW	32	0x0000 0994	0x4A00 2994
CTRL_CORE_DSP1_IRQ_72_73	RW	32	0x0000 0998	0x4A00 2998
CTRL_CORE_DSP1_IRQ_74_75	RW	32	0x0000 099C	0x4A00 299C
CTRL_CORE_DSP1_IRQ_76_77	RW	32	0x0000 09A0	0x4A00 29A0
CTRL_CORE_DSP1_IRQ_78_79	RW	32	0x0000 09A4	0x4A00 29A4
CTRL_CORE_DSP1_IRQ_80_81	RW	32	0x0000 09A8	0x4A00 29A8
CTRL_CORE_DSP1_IRQ_82_83	RW	32	0x0000 09AC	0x4A00 29AC
CTRL_CORE_DSP1_IRQ_84_85	RW	32	0x0000 09B0	0x4A00 29B0
CTRL_CORE_DSP1_IRQ_86_87	RW	32	0x0000 09B4	0x4A00 29B4
CTRL_CORE_DSP1_IRQ_88_89	RW	32	0x0000 09B8	0x4A00 29B8

Table 13-25. CTRL_MODULE_CORE Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	CTRL_MODULE_CORE Base Address
CTRL_CORE_DSP1_IRQ_90_91	RW	32	0x0000 09BC	0x4A00 29BC
CTRL_CORE_DSP1_IRQ_92_93	RW	32	0x0000 09C0	0x4A00 29C0
CTRL_CORE_DSP1_IRQ_94_95	RW	32	0x0000 09C4	0x4A00 29C4
CTRL_CORE_DSP2_IRQ_32_33	RW	32	0x0000 09C8	0x4A00 29C8
CTRL_CORE_DSP2_IRQ_34_35	RW	32	0x0000 09CC	0x4A00 29CC
CTRL_CORE_DSP2_IRQ_36_37	RW	32	0x0000 09D0	0x4A00 29D0
CTRL_CORE_DSP2_IRQ_38_39	RW	32	0x0000 09D4	0x4A00 29D4
CTRL_CORE_DSP2_IRQ_40_41	RW	32	0x0000 09D8	0x4A00 29D8
CTRL_CORE_DSP2_IRQ_42_43	RW	32	0x0000 09DC	0x4A00 29DC
CTRL_CORE_DSP2_IRQ_44_45	RW	32	0x0000 09E0	0x4A00 29E0
CTRL_CORE_DSP2_IRQ_46_47	RW	32	0x0000 09E4	0x4A00 29E4
CTRL_CORE_DSP2_IRQ_48_49	RW	32	0x0000 09E8	0x4A00 29E8
CTRL_CORE_DSP2_IRQ_50_51	RW	32	0x0000 09EC	0x4A00 29EC
CTRL_CORE_DSP2_IRQ_52_53	RW	32	0x0000 09F0	0x4A00 29F0
CTRL_CORE_DSP2_IRQ_54_55	RW	32	0x0000 09F4	0x4A00 29F4
CTRL_CORE_DSP2_IRQ_56_57	RW	32	0x0000 09F8	0x4A00 29F8
CTRL_CORE_DSP2_IRQ_58_59	RW	32	0x0000 09FC	0x4A00 29FC
CTRL_CORE_DSP2_IRQ_60_61	RW	32	0x0000 0A00	0x4A00 2A00
CTRL_CORE_DSP2_IRQ_62_63	RW	32	0x0000 0A04	0x4A00 2A04
CTRL_CORE_DSP2_IRQ_64_65	RW	32	0x0000 0A08	0x4A00 2A08
CTRL_CORE_DSP2_IRQ_66_67	RW	32	0x0000 0A0C	0x4A00 2A0C
CTRL_CORE_DSP2_IRQ_68_69	RW	32	0x0000 0A10	0x4A00 2A10
CTRL_CORE_DSP2_IRQ_70_71	RW	32	0x0000 0A14	0x4A00 2A14
CTRL_CORE_DSP2_IRQ_72_73	RW	32	0x0000 0A18	0x4A00 2A18
CTRL_CORE_DSP2_IRQ_74_75	RW	32	0x0000 0A1C	0x4A00 2A1C
CTRL_CORE_DSP2_IRQ_76_77	RW	32	0x0000 0A20	0x4A00 2A20
CTRL_CORE_DSP2_IRQ_78_79	RW	32	0x0000 0A24	0x4A00 2A24
CTRL_CORE_DSP2_IRQ_80_81	RW	32	0x0000 0A28	0x4A00 2A28
CTRL_CORE_DSP2_IRQ_82_83	RW	32	0x0000 0A2C	0x4A00 2A2C
CTRL_CORE_DSP2_IRQ_84_85	RW	32	0x0000 0A30	0x4A00 2A30
CTRL_CORE_DSP2_IRQ_86_87	RW	32	0x0000 0A34	0x4A00 2A34
CTRL_CORE_DSP2_IRQ_88_89	RW	32	0x0000 0A38	0x4A00 2A38
CTRL_CORE_DSP2_IRQ_90_91	RW	32	0x0000 0A3C	0x4A00 2A3C
CTRL_CORE_DSP2_IRQ_92_93	RW	32	0x0000 0A40	0x4A00 2A40
CTRL_CORE_DSP2_IRQ_94_95	RW	32	0x0000 0A44	0x4A00 2A44
RESERVED_i (i = 0 to 139)	RW	32	0x0000 0A48 + (i*4)	0x4A00 2A48 + (i*4)
CTRL_CORE_DMA_EDMA_DREQ_0_1	RW	32	0x0000 0C78	0x4A00 2C78
CTRL_CORE_DMA_EDMA_DREQ_2_3	RW	32	0x0000 0C7C	0x4A00 2C7C
CTRL_CORE_DMA_EDMA_DREQ_4_5	RW	32	0x0000 0C80	0x4A00 2C80
CTRL_CORE_DMA_EDMA_DREQ_6_7	RW	32	0x0000 0C84	0x4A00 2C84
CTRL_CORE_DMA_EDMA_DREQ_8_9	RW	32	0x0000 0C88	0x4A00 2C88
CTRL_CORE_DMA_EDMA_DREQ_10_11	RW	32	0x0000 0C8C	0x4A00 2C8C
CTRL_CORE_DMA_EDMA_DREQ_12_13	RW	32	0x0000 0C90	0x4A00 2C90
CTRL_CORE_DMA_EDMA_DREQ_14_15	RW	32	0x0000 0C94	0x4A00 2C94
CTRL_CORE_DMA_EDMA_DREQ_16_17	RW	32	0x0000 0C98	0x4A00 2C98
CTRL_CORE_DMA_EDMA_DREQ_18_19	RW	32	0x0000 0C9C	0x4A00 2C9C
CTRL_CORE_DMA_EDMA_DREQ_20_21	RW	32	0x0000 0CA0	0x4A00 2CA0
CTRL_CORE_DMA_EDMA_DREQ_22_23	RW	32	0x0000 0CA4	0x4A00 2CA4
CTRL_CORE_DMA_EDMA_DREQ_24_25	RW	32	0x0000 0CA8	0x4A00 2CA8

Table 13-25. CTRL_MODULE_CORE Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	CTRL_MODULE_CORE Base Address
CTRL_CORE_DMA_EDMA_DREQ_26_27	RW	32	0x0000 0CAC	0x4A00 2CAC
CTRL_CORE_DMA_EDMA_DREQ_28_29	RW	32	0x0000 0CB0	0x4A00 2CB0
CTRL_CORE_DMA_EDMA_DREQ_30_31	RW	32	0x0000 0CB4	0x4A00 2CB4
CTRL_CORE_DMA_EDMA_DREQ_32_33	RW	32	0x0000 0CB8	0x4A00 2CB8
CTRL_CORE_DMA_EDMA_DREQ_34_35	RW	32	0x0000 0CBC	0x4A00 2CBC
CTRL_CORE_DMA_EDMA_DREQ_36_37	RW	32	0x0000 0CC0	0x4A00 2CC0
CTRL_CORE_DMA_EDMA_DREQ_38_39	RW	32	0x0000 0CC4	0x4A00 2CC4
CTRL_CORE_DMA_EDMA_DREQ_40_41	RW	32	0x0000 0CC8	0x4A00 2CC8
CTRL_CORE_DMA_EDMA_DREQ_42_43	RW	32	0x0000 0CCC	0x4A00 2CCC
CTRL_CORE_DMA_EDMA_DREQ_44_45	RW	32	0x0000 0CD0	0x4A00 2CD0
CTRL_CORE_DMA_EDMA_DREQ_46_47	RW	32	0x0000 0CD4	0x4A00 2CD4
CTRL_CORE_DMA_EDMA_DREQ_48_49	RW	32	0x0000 0CD8	0x4A00 2CD8
CTRL_CORE_DMA_EDMA_DREQ_50_51	RW	32	0x0000 0CDC	0x4A00 2CDC
CTRL_CORE_DMA_EDMA_DREQ_52_53	RW	32	0x0000 0CE0	0x4A00 2CE0
CTRL_CORE_DMA_EDMA_DREQ_54_55	RW	32	0x0000 0CE4	0x4A00 2CE4
CTRL_CORE_DMA_EDMA_DREQ_56_57	RW	32	0x0000 0CE8	0x4A00 2CE8
CTRL_CORE_DMA_EDMA_DREQ_58_59	RW	32	0x0000 0CEC	0x4A00 2CEC
CTRL_CORE_DMA_EDMA_DREQ_60_61	RW	32	0x0000 0CF0	0x4A00 2CF0
CTRL_CORE_DMA_EDMA_DREQ_62_63	RW	32	0x0000 0CF4	0x4A00 2CF4
CTRL_CORE_DMA_DSP1_DREQ_0_1	RW	32	0x0000 0CF8	0x4A00 2CF8
CTRL_CORE_DMA_DSP1_DREQ_2_3	RW	32	0x0000 0CFC	0x4A00 2CFC
CTRL_CORE_DMA_DSP1_DREQ_4_5	RW	32	0x0000 0D00	0x4A00 2D00
CTRL_CORE_DMA_DSP1_DREQ_6_7	RW	32	0x0000 0D04	0x4A00 2D04
CTRL_CORE_DMA_DSP1_DREQ_8_9	RW	32	0x0000 0D08	0x4A00 2D08
CTRL_CORE_DMA_DSP1_DREQ_10_11	RW	32	0x0000 0D0C	0x4A00 2D0C
CTRL_CORE_DMA_DSP1_DREQ_12_13	RW	32	0x0000 0D10	0x4A00 2D10
CTRL_CORE_DMA_DSP1_DREQ_14_15	RW	32	0x0000 0D14	0x4A00 2D14
CTRL_CORE_DMA_DSP1_DREQ_16_17	RW	32	0x0000 0D18	0x4A00 2D18
CTRL_CORE_DMA_DSP1_DREQ_18_19	RW	32	0x0000 0D1C	0x4A00 2D1C
CTRL_CORE_DMA_DSP2_DREQ_0_1	RW	32	0x0000 0D20	0x4A00 2D20
CTRL_CORE_DMA_DSP2_DREQ_2_3	RW	32	0x0000 0D24	0x4A00 2D24
CTRL_CORE_DMA_DSP2_DREQ_4_5	RW	32	0x0000 0D28	0x4A00 2D28
CTRL_CORE_DMA_DSP2_DREQ_6_7	RW	32	0x0000 0D2C	0x4A00 2D2C
CTRL_CORE_DMA_DSP2_DREQ_8_9	RW	32	0x0000 0D30	0x4A00 2D30
CTRL_CORE_DMA_DSP2_DREQ_10_11	RW	32	0x0000 0D34	0x4A00 2D34
CTRL_CORE_DMA_DSP2_DREQ_12_13	RW	32	0x0000 0D38	0x4A00 2D38
CTRL_CORE_DMA_DSP2_DREQ_14_15	RW	32	0x0000 0D3C	0x4A00 2D3C
CTRL_CORE_DMA_DSP2_DREQ_16_17	RW	32	0x0000 0D40	0x4A00 2D40
CTRL_CORE_DMA_DSP2_DREQ_18_19	RW	32	0x0000 0D44	0x4A00 2D44
RESERVED	R	32	0x0000 0D48	0x4A00 2D48
RESERVED	R	32	0x0000 0D4C	0x4A00 2D4C
RESERVED	R	32	0x0000 0D50	0x4A00 2D50
CTRL_CORE_ESM_GROUP1_0	RW	32	0x0000 0D54	0x4A00 2D54
CTRL_CORE_ESM_GROUP1_1	RW	32	0x0000 0D58	0x4A00 2D58
CTRL_CORE_ESM_GROUP1_2	RW	32	0x0000 0D5C	0x4A00 2D5C
CTRL_CORE_ESM_GROUP1_3	RW	32	0x0000 0D60	0x4A00 2D60
RESERVED_j (j = 0 to 39)	R	32	0x0000 0D64 + (j*4)	0x4A00 2D64 + (j*4)
RESERVED	R	32	0x0000 0E04	0x4A00 2E04
CTRL_CORE_CAMERRX_CONTROL	RW	32	0x0000 0E08	0x4A00 2E08

Table 13-25. CTRL_MODULE_CORE Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	CTRL_MODULE_CORE Base Address
RESERVED	R	32	0x0000 0E0C	0x4A00 2E0C
RESERVED	R	32	0x0000 0E14	0x4A00 2E14
RESERVED	R	32	0x0000 0E18	0x4A00 2E18
RESERVED	R	32	0x0000 0E1C	0x4A00 2E1C
RESERVED	R	32	0x0000 0E20	0x4A00 2E20
RESERVED	R	32	0x0000 0E24	0x4A00 2E24
CTRL_CORE_CONTROL_DDRCACH1_0	RW	32	0x0000 0E30	0x4A00 2E30
RESERVED	R	32	0x0000 0E34	0x4A00 2E34
CTRL_CORE_CONTROL_DDRCH1_0	RW	32	0x0000 0E38	0x4A00 2E38
CTRL_CORE_CONTROL_DDRCH1_1	RW	32	0x0000 0E3C	0x4A00 2E3C
RESERVED	R	32	0x0000 0E40	0x4A00 2E40
RESERVED	R	32	0x0000 0E44	0x4A00 2E44
CTRL_CORE_CONTROL_DDRCH1_2	RW	32	0x0000 0E48	0x4A00 2E48
RESERVED	R	32	0x0000 0E4C	0x4A00 2E4C
CTRL_CORE_CONTROL_DDRIO_0	RW	32	0x0000 0E50	0x4A00 2E50
RESERVED_k (k = 0 to 106)	R	32	0x0000 0E54 + (k*4)	0x4A00 2E54 + (k*4)
CTRL_CORE_HWOBS_FINAL_MUX_SEL	RW	32	0x0000 1000	0x4A00 3000
RESERVED	R	32	0x0000 1004	0x4A00 3004
RESERVED	R	32	0x0000 1008	0x4A00 3008
CTRL_CORE_CONF_DEBUG_SEL_TST_0	RW	32	0x0000 100C	0x4A00 300C
CTRL_CORE_CONF_DEBUG_SEL_TST_1	RW	32	0x0000 1010	0x4A00 3010
CTRL_CORE_CONF_DEBUG_SEL_TST_2	RW	32	0x0000 1014	0x4A00 3014
CTRL_CORE_CONF_DEBUG_SEL_TST_3	RW	32	0x0000 1018	0x4A00 3018
CTRL_CORE_CONF_DEBUG_SEL_TST_4	RW	32	0x0000 101C	0x4A00 301C
CTRL_CORE_CONF_DEBUG_SEL_TST_5	RW	32	0x0000 1020	0x4A00 3020
CTRL_CORE_CONF_DEBUG_SEL_TST_6	RW	32	0x0000 1024	0x4A00 3024
CTRL_CORE_CONF_DEBUG_SEL_TST_7	RW	32	0x0000 1028	0x4A00 3028
CTRL_CORE_CONF_DEBUG_SEL_TST_8	RW	32	0x0000 102C	0x4A00 302C
CTRL_CORE_CONF_DEBUG_SEL_TST_9	RW	32	0x0000 1030	0x4A00 3030
CTRL_CORE_CONF_DEBUG_SEL_TST_10	RW	32	0x0000 1034	0x4A00 3034
CTRL_CORE_CONF_DEBUG_SEL_TST_11	RW	32	0x0000 1038	0x4A00 3038
CTRL_CORE_CONF_DEBUG_SEL_TST_12	RW	32	0x0000 103C	0x4A00 303C
CTRL_CORE_CONF_DEBUG_SEL_TST_13	RW	32	0x0000 1040	0x4A00 3040
CTRL_CORE_CONF_DEBUG_SEL_TST_14	RW	32	0x0000 1044	0x4A00 3044
CTRL_CORE_CONF_DEBUG_SEL_TST_15	RW	32	0x0000 1048	0x4A00 3048
CTRL_CORE_CONF_DEBUG_SEL_TST_16	RW	32	0x0000 104C	0x4A00 304C
CTRL_CORE_CONF_DEBUG_SEL_TST_17	RW	32	0x0000 1050	0x4A00 3050
CTRL_CORE_CONF_DEBUG_SEL_TST_18	RW	32	0x0000 1054	0x4A00 3054
CTRL_CORE_CONF_DEBUG_SEL_TST_19	RW	32	0x0000 1058	0x4A00 3058
CTRL_CORE_CONF_DEBUG_SEL_TST_20	RW	32	0x0000 105C	0x4A00 305C
CTRL_CORE_CONF_DEBUG_SEL_TST_21	RW	32	0x0000 1060	0x4A00 3060
CTRL_CORE_CONF_DEBUG_SEL_TST_22	RW	32	0x0000 1064	0x4A00 3064
CTRL_CORE_CONF_DEBUG_SEL_TST_23	RW	32	0x0000 1068	0x4A00 3068
CTRL_CORE_CONF_DEBUG_SEL_TST_24	RW	32	0x0000 106C	0x4A00 306C
CTRL_CORE_CONF_DEBUG_SEL_TST_25	RW	32	0x0000 1070	0x4A00 3070
CTRL_CORE_CONF_DEBUG_SEL_TST_26	RW	32	0x0000 1074	0x4A00 3074
CTRL_CORE_CONF_DEBUG_SEL_TST_27	RW	32	0x0000 1078	0x4A00 3078
CTRL_CORE_CONF_DEBUG_SEL_TST_28	RW	32	0x0000 107C	0x4A00 307C
CTRL_CORE_CONF_DEBUG_SEL_TST_29	RW	32	0x0000 1080	0x4A00 3080

Table 13-25. CTRL_MODULE_CORE Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	CTRL_MODULE_CORE Base Address
CTRL_CORE_CONF_DEBUG_SEL_TST_30	RW	32	0x0000 1084	0x4A00 3084
CTRL_CORE_CONF_DEBUG_SEL_TST_31	RW	32	0x0000 1088	0x4A00 3088
RESERVED_I (I = 0 to 220)	R	32	0x0000 108C + (I*4)	0x4A00 308C + (I*4)
CTRL_CORE_PAD_GPMC_CLK	RW	32	0x0000 1400	0x4A00 3400
CTRL_CORE_PAD_GPMC_BEN0	RW	32	0x0000 1404	0x4A00 3404
CTRL_CORE_PAD_GPMC_BEN1	RW	32	0x0000 1408	0x4A00 3408
CTRL_CORE_PAD_GPMC_ADVN_ALE	RW	32	0x0000 140C	0x4A00 340C
CTRL_CORE_PAD_GPMC_OEN_REN	RW	32	0x0000 1410	0x4A00 3410
CTRL_CORE_PAD_GPMC_WEN	RW	32	0x0000 1414	0x4A00 3414
CTRL_CORE_PAD_GPMC_CS0	RW	32	0x0000 1418	0x4A00 3418
CTRL_CORE_PAD_GPMC_CS1	RW	32	0x0000 141C	0x4A00 341C
CTRL_CORE_PAD_GPMC_CS2	RW	32	0x0000 1420	0x4A00 3420
CTRL_CORE_PAD_GPMC_CS3	RW	32	0x0000 1424	0x4A00 3424
CTRL_CORE_PAD_GPMC_CS4	RW	32	0x0000 1428	0x4A00 3428
CTRL_CORE_PAD_GPMC_CS5	RW	32	0x0000 142C	0x4A00 342C
CTRL_CORE_PAD_GPMC_CS6	RW	32	0x0000 1430	0x4A00 3430
CTRL_CORE_PAD_GPMC_WAIT0	RW	32	0x0000 1434	0x4A00 3434
CTRL_CORE_PAD_GPMC_AD0	RW	32	0x0000 1438	0x4A00 3438
CTRL_CORE_PAD_GPMC_AD1	RW	32	0x0000 143C	0x4A00 343C
CTRL_CORE_PAD_GPMC_AD2	RW	32	0x0000 1440	0x4A00 3440
CTRL_CORE_PAD_GPMC_AD3	RW	32	0x0000 1444	0x4A00 3444
CTRL_CORE_PAD_GPMC_AD4	RW	32	0x0000 1448	0x4A00 3448
CTRL_CORE_PAD_GPMC_AD5	RW	32	0x0000 144C	0x4A00 344C
CTRL_CORE_PAD_GPMC_AD6	RW	32	0x0000 1450	0x4A00 3450
CTRL_CORE_PAD_GPMC_AD7	RW	32	0x0000 1454	0x4A00 3454
CTRL_CORE_PAD_GPMC_AD8	RW	32	0x0000 1458	0x4A00 3458
CTRL_CORE_PAD_GPMC_AD9	RW	32	0x0000 145C	0x4A00 345C
CTRL_CORE_PAD_GPMC_AD10	RW	32	0x0000 1460	0x4A00 3460
CTRL_CORE_PAD_GPMC_AD11	RW	32	0x0000 1464	0x4A00 3464
CTRL_CORE_PAD_GPMC_AD12	RW	32	0x0000 1468	0x4A00 3468
CTRL_CORE_PAD_GPMC_AD13	RW	32	0x0000 146C	0x4A00 346C
CTRL_CORE_PAD_GPMC_AD14	RW	32	0x0000 1470	0x4A00 3470
CTRL_CORE_PAD_GPMC_AD15	RW	32	0x0000 1474	0x4A00 3474
CTRL_CORE_PAD_VIN1A_CLK0	RW	32	0x0000 1478	0x4A00 3478
CTRL_CORE_PAD_VIN1A_DE0	RW	32	0x0000 147C	0x4A00 347C
CTRL_CORE_PAD_VIN1A_FLD0	RW	32	0x0000 1480	0x4A00 3480
CTRL_CORE_PAD_VIN1A_HSYNC0	RW	32	0x0000 1484	0x4A00 3484
CTRL_CORE_PAD_VIN1A_VSYNC0	RW	32	0x0000 1488	0x4A00 3488
CTRL_CORE_PAD_VIN1A_D0	RW	32	0x0000 148C	0x4A00 348C
CTRL_CORE_PAD_VIN1A_D1	RW	32	0x0000 1490	0x4A00 3490
CTRL_CORE_PAD_VIN1A_D2	RW	32	0x0000 1494	0x4A00 3494
CTRL_CORE_PAD_VIN1A_D3	RW	32	0x0000 1498	0x4A00 3498
CTRL_CORE_PAD_VIN1A_D4	RW	32	0x0000 149C	0x4A00 349C
CTRL_CORE_PAD_VIN1A_D5	RW	32	0x0000 14A0	0x4A00 34A0
CTRL_CORE_PAD_VIN1A_D6	RW	32	0x0000 14A4	0x4A00 34A4
CTRL_CORE_PAD_VIN1A_D7	RW	32	0x0000 14A8	0x4A00 34A8
CTRL_CORE_PAD_VIN1A_D8	RW	32	0x0000 14AC	0x4A00 34AC
CTRL_CORE_PAD_VIN1A_D9	RW	32	0x0000 14B0	0x4A00 34B0
CTRL_CORE_PAD_VIN1A_D10	RW	32	0x0000 14B4	0x4A00 34B4

Table 13-25. CTRL_MODULE_CORE Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	CTRL_MODULE_CORE Base Address
CTRL_CORE_PAD_VIN1A_D11	RW	32	0x0000 14B8	0x4A00 34B8
CTRL_CORE_PAD_VIN1A_D12	RW	32	0x0000 14BC	0x4A00 34BC
CTRL_CORE_PAD_VIN1A_D13	RW	32	0x0000 14C0	0x4A00 34C0
CTRL_CORE_PAD_VIN1A_D14	RW	32	0x0000 14C4	0x4A00 34C4
CTRL_CORE_PAD_VIN1A_D15	RW	32	0x0000 14C8	0x4A00 34C8
CTRL_CORE_PAD_VIN2A_CLK0	RW	32	0x0000 14CC	0x4A00 34CC
CTRL_CORE_PAD_VIN2A_DE0	RW	32	0x0000 14D0	0x4A00 34D0
CTRL_CORE_PAD_VIN2A_FLD0	RW	32	0x0000 14D4	0x4A00 34D4
CTRL_CORE_PAD_VOUT1_CLK	RW	32	0x0000 14D8	0x4A00 34D8
CTRL_CORE_PAD_VOUT1_DE	RW	32	0x0000 14DC	0x4A00 34DC
CTRL_CORE_PAD_VOUT1_FLD	RW	32	0x0000 14E0	0x4A00 34E0
CTRL_CORE_PAD_VOUT1_HSYNC	RW	32	0x0000 14E4	0x4A00 34E4
CTRL_CORE_PAD_VOUT1_VSYNC	RW	32	0x0000 14E8	0x4A00 34E8
CTRL_CORE_PAD_VOUT1_D0	RW	32	0x0000 14EC	0x4A00 34EC
CTRL_CORE_PAD_VOUT1_D1	RW	32	0x0000 14F0	0x4A00 34F0
CTRL_CORE_PAD_VOUT1_D2	RW	32	0x0000 14F4	0x4A00 34F4
CTRL_CORE_PAD_VOUT1_D3	RW	32	0x0000 14F8	0x4A00 34F8
CTRL_CORE_PAD_VOUT1_D4	RW	32	0x0000 14FC	0x4A00 34FC
CTRL_CORE_PAD_VOUT1_D5	RW	32	0x0000 1500	0x4A00 3500
CTRL_CORE_PAD_VOUT1_D6	RW	32	0x0000 1504	0x4A00 3504
CTRL_CORE_PAD_VOUT1_D7	RW	32	0x0000 1508	0x4A00 3508
CTRL_CORE_PAD_VOUT1_D8	RW	32	0x0000 150C	0x4A00 350C
CTRL_CORE_PAD_VOUT1_D9	RW	32	0x0000 1510	0x4A00 3510
CTRL_CORE_PAD_VOUT1_D10	RW	32	0x0000 1514	0x4A00 3514
CTRL_CORE_PAD_VOUT1_D11	RW	32	0x0000 1518	0x4A00 3518
CTRL_CORE_PAD_VOUT1_D12	RW	32	0x0000 151C	0x4A00 351C
CTRL_CORE_PAD_VOUT1_D13	RW	32	0x0000 1520	0x4A00 3520
CTRL_CORE_PAD_VOUT1_D14	RW	32	0x0000 1524	0x4A00 3524
CTRL_CORE_PAD_VOUT1_D15	RW	32	0x0000 1528	0x4A00 3528
CTRL_CORE_PAD_VOUT1_D16	RW	32	0x0000 152C	0x4A00 352C
CTRL_CORE_PAD_VOUT1_D17	RW	32	0x0000 1530	0x4A00 3530
CTRL_CORE_PAD_VOUT1_D18	RW	32	0x0000 1534	0x4A00 3534
CTRL_CORE_PAD_VOUT1_D19	RW	32	0x0000 1538	0x4A00 3538
CTRL_CORE_PAD_VOUT1_D20	RW	32	0x0000 153C	0x4A00 353C
CTRL_CORE_PAD_VOUT1_D21	RW	32	0x0000 1540	0x4A00 3540
CTRL_CORE_PAD_VOUT1_D22	RW	32	0x0000 1544	0x4A00 3544
CTRL_CORE_PAD_VOUT1_D23	RW	32	0x0000 1548	0x4A00 3548
CTRL_CORE_PAD_MCAN_TX	RW	32	0x0000 154C	0x4A00 354C
CTRL_CORE_PAD_MCAN_RX	RW	32	0x0000 1550	0x4A00 3550
CTRL_CORE_PAD_MDIO_MCLK	RW	32	0x0000 1554	0x4A00 3554
CTRL_CORE_PAD_MDIO_D	RW	32	0x0000 1558	0x4A00 3558
CTRL_CORE_PAD_RGMII0_TXC	RW	32	0x0000 155C	0x4A00 355C
CTRL_CORE_PAD_RGMII0_TXCTL	RW	32	0x0000 1560	0x4A00 3560
CTRL_CORE_PAD_RGMII0_TXD3	RW	32	0x0000 1564	0x4A00 3564
CTRL_CORE_PAD_RGMII0_TXD2	RW	32	0x0000 1568	0x4A00 3568
CTRL_CORE_PAD_RGMII0_TXD1	RW	32	0x0000 156C	0x4A00 356C
CTRL_CORE_PAD_RGMII0_TXD0	RW	32	0x0000 1570	0x4A00 3570
CTRL_CORE_PAD_RGMII0_RXC	RW	32	0x0000 1574	0x4A00 3574
CTRL_CORE_PAD_RGMII0_RXCTL	RW	32	0x0000 1578	0x4A00 3578
CTRL_CORE_PAD_RGMII0_RXD3	RW	32	0x0000 157C	0x4A00 357C

Table 13-25. CTRL_MODULE_CORE Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	CTRL_MODULE_CORE Base Address
CTRL_CORE_PAD_RGMII0_RXD2	RW	32	0x0000 1580	0x4A00 3580
CTRL_CORE_PAD_RGMII0_RXD1	RW	32	0x0000 1584	0x4A00 3584
CTRL_CORE_PAD_RGMII0_RXD0	RW	32	0x0000 1588	0x4A00 3588
CTRL_CORE_PAD_XREF_CLK0	RW	32	0x0000 158C	0x4A00 358C
CTRL_CORE_PAD_SPI1_SCLK	RW	32	0x0000 1590	0x4A00 3590
CTRL_CORE_PAD_SPI1_D1	RW	32	0x0000 1594	0x4A00 3594
CTRL_CORE_PAD_SPI1_D0	RW	32	0x0000 1598	0x4A00 3598
CTRL_CORE_PAD_SPI1_CS0	RW	32	0x0000 159C	0x4A00 359C
CTRL_CORE_PAD_SPI1_CS1	RW	32	0x0000 15A0	0x4A00 35A0
CTRL_CORE_PAD_SPI2_SCLK	RW	32	0x0000 15A4	0x4A00 35A4
CTRL_CORE_PAD_SPI2_D1	RW	32	0x0000 15A8	0x4A00 35A8
CTRL_CORE_PAD_SPI2_D0	RW	32	0x0000 15AC	0x4A00 35AC
CTRL_CORE_PAD_SPI2_CS0	RW	32	0x0000 15B0	0x4A00 35B0
CTRL_CORE_PAD_DCAN_TX	RW	32	0x0000 15B4	0x4A00 35B4
CTRL_CORE_PAD_DCAN_RX	RW	32	0x0000 15B8	0x4A00 35B8
CTRL_CORE_PAD_UART1_RXD	RW	32	0x0000 15BC	0x4A00 35BC
CTRL_CORE_PAD_UART1_TXD	RW	32	0x0000 15C0	0x4A00 35C0
CTRL_CORE_PAD_UART1_CTSN	RW	32	0x0000 15C4	0x4A00 35C4
CTRL_CORE_PAD_UART1_RTSN	RW	32	0x0000 15C8	0x4A00 35C8
CTRL_CORE_PAD_UART2_RXD	RW	32	0x0000 15CC	0x4A00 35CC
CTRL_CORE_PAD_UART2_TXD	RW	32	0x0000 15D0	0x4A00 35D0
CTRL_CORE_PAD_UART2_CTSN	RW	32	0x0000 15D4	0x4A00 35D4
CTRL_CORE_PAD_UART2_RTSN	RW	32	0x0000 15D8	0x4A00 35D8
CTRL_CORE_PAD_I2C1_SDA	RW	32	0x0000 15DC	0x4A00 35DC
CTRL_CORE_PAD_I2C1_SCL	RW	32	0x0000 15E0	0x4A00 35E0
CTRL_CORE_PAD_I2C2_SDA	RW	32	0x0000 15E4	0x4A00 35E4
CTRL_CORE_PAD_I2C2_SCL	RW	32	0x0000 15E8	0x4A00 35E8
CTRL_CORE_PAD_TMS	RW	32	0x0000 15EC	0x4A00 35EC
CTRL_CORE_PAD_TDI	RW	32	0x0000 15F0	0x4A00 35F0
CTRL_CORE_PAD_TDO	RW	32	0x0000 15F4	0x4A00 35F4
CTRL_CORE_PAD_TCLK	RW	32	0x0000 15F8	0x4A00 35F8
CTRL_CORE_PAD_TRSTN	RW	32	0x0000 15FC	0x4A00 35FC
CTRL_CORE_PAD_RTCK	RW	32	0x0000 1600	0x4A00 3600
CTRL_CORE_PAD_EMU0	RW	32	0x0000 1604	0x4A00 3604
CTRL_CORE_PAD_EMU1	RW	32	0x0000 1608	0x4A00 3608
CTRL_CORE_PAD_RESETN	RW	32	0x0000 160C	0x4A00 360C
CTRL_CORE_PAD_NMIN	RW	32	0x0000 1610	0x4A00 3610
CTRL_CORE_PAD_RSTOUTN	RW	32	0x0000 1614	0x4A00 3614
RESERVED_m (m = 0 to 277)	R	32	0x0000 1618 + (m*4)	0x4A00 3618 + (m*4)
CTRL_CORE_STD_FUSE_OPP_VDD_DSPEVE_LVT_0	R	32	0x0000 1B38	0x4A00 3B38
CTRL_CORE_STD_FUSE_OPP_VDD_DSPEVE_LVT_1	R	32	0x0000 1B3C	0x4A00 3B3C
CTRL_CORE_STD_FUSE_OPP_VDD_DSPEVE_LVT_2	R	32	0x0000 1B40	0x4A00 3B40
CTRL_CORE_STD_FUSE_OPP_VDD_DSPEVE_LVT_3	R	32	0x0000 1B44	0x4A00 3B44
CTRL_CORE_STD_FUSE_OPP_VDD_DSPEVE_LVT_4	R	32	0x0000 1B48	0x4A00 3B48
RESERVED	R	32	0x0000 1B4C	0x4A00 3B4C
RESERVED	R	32	0x0000 1B50	0x4A00 3B50
RESERVED	R	32	0x0000 1B54	0x4A00 3B54
RESERVED	R	32	0x0000 1B58	0x4A00 3B58
RESERVED	R	32	0x0000 1B5C	0x4A00 3B5C

Table 13-25. CTRL_MODULE_CORE Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	CTRL_MODULE_CORE Base Address
CTRL_CORE_STD_FUSE_OPP_VDD_CORE_LVT_0	R	32	0x0000 1B60	0x4A00 3B60
CTRL_CORE_STD_FUSE_OPP_VDD_CORE_LVT_1	R	32	0x0000 1B64	0x4A00 3B64
CTRL_CORE_STD_FUSE_OPP_VDD_CORE_LVT_2	R	32	0x0000 1B68	0x4A00 3B68
CTRL_CORE_STD_FUSE_OPP_VDD_CORE_LVT_3	R	32	0x0000 1B6C	0x4A00 3B6C
CTRL_CORE_STD_FUSE_OPP_VDD_CORE_LVT_4	R	32	0x0000 1B70	0x4A00 3B70
CTRL_CORE_LDOSRAM_CORE_4_VOLTAGE_CTRL	RW	32	0x0000 1B74	0x4A00 3B74
CTRL_CORE_LDOSRAM_CORE_5_VOLTAGE_CTRL	RW	32	0x0000 1B78	0x4A00 3B78
CTRL_CORE_LDOSRAM_DSPEVE_2_VOLTAGE_CTRL	RW	32	0x0000 1B7C	0x4A00 3B7C
RESERVED_n (n = 0 to 32)	R	32	0x0000 1B80 + (n*4)	0x4A00 3B80 + (n*4)
CTRL_CORE_SMA_SW_2	RW	32	0x0000 1C04	0x4A00 3C04
CTRL_CORE_SMA_SW_3	RW	32	0x0000 1C08	0x4A00 3C08
RESERVED	R	32	0x0000 1C0C	0x4A00 3C0C
RESERVED	R	32	0x0000 1C10	0x4A00 3C10
CTRL_CORE_SMA_SW_6	RW	32	0x0000 1C14	0x4A00 3C14
CTRL_CORE_SMA_SW_7	RW	32	0x0000 1C18	0x4A00 3C18
RESERVED_o (o = 0 to 10)	R	32	0x0000 1C1C + (o*4)	0x4A00 3C1C + (o*4)
CTRL_CORE_FIREWALL_CONNID_CONTROL_0	RW	32	0x0000 1C48	0x4A00 3C48
CTRL_CORE_FIREWALL_CONNID_CONTROL_1	RW	32	0x0000 1C4C	0x4A00 3C4C
CTRL_CORE_FIREWALL_CONNID_CONTROL_2	RW	32	0x0000 1C50	0x4A00 3C50
CTRL_CORE_FIREWALL_CONNID_CONTROL_3	RW	32	0x0000 1C54	0x4A00 3C54
CTRL_CORE_EMIF_MPU_ROUTING	RW	32	0x0000 1C58	0x4A00 3C58
CTRL_CORE_PRCM_CLKSEL_CONTROL	RW	32	0x0000 1C5C	0x4A00 3C5C
CTRL_CORE_PRCM_CLKDIV_CONTROL1	RW	32	0x0000 1C60	0x4A00 3C60
CTRL_CORE_PRCM_CLKDIV_CONTROL2	RW	32	0x0000 1C64	0x4A00 3C64
CTRL_CORE_SMA_SW_10	RW	32	0x0000 1C68	0x4A00 3C68
CTRL_CORE_SMA_SW_11	RW	32	0x0000 1C6C	0x4A00 3C6C
CTRL_CORE_SMA_SW_12	RW	32	0x0000 1C70	0x4A00 3C70
CTRL_CORE_SMA_SW_13	RW	32	0x0000 1C74	0x4A00 3C74
CTRL_CORE_TESOC_LAST_RESET_INDICATOR	RW	32	0x0000 1C78	0x4A00 3C78
CTRL_CORE_SD_DAC_CONTROL	RW	32	0x0000 1C7C	0x4A00 3C7C
CTRL_CORE_SD_DAC_TRIM_VALUE	RW	32	0x0000 1C80	0x4A00 3C80
CTRL_CORE_ADC_ERROR_OFFSET	RW	32	0x0000 1C84	0x4A00 3C84
CTRL_CORE_IPU_WAKEUP	RW	32	0x0000 1C88	0x4A00 3C88
CTRL_CORE_ISS_EFUSE	RW	32	0x0000 1C8C	0x4A00 3C8C
CTRL_CORE_SMA_SW_14	RW	32	0x0000 1C90	0x4A00 3C90
CTRL_CORE_SMA_SW_15	RW	32	0x0000 1C94	0x4A00 3C94
CTRL_CORE_SMA_SW_16	RW	32	0x0000 1C98	0x4A00 3C98
CTRL_CORE_SMA_SW_17	RW	32	0x0000 1C9C	0x4A00 3C9C
CTRL_CORE_ROM_CPU0_BRANCH	RW	32	0x0000 1CA0	0x4A00 3CA0
CTRL_CORE_ROM_CPU1_BRANCH	RW	32	0x0000 1CA4	0x4A00 3CA4
CTRL_CORE_ROM_AUXBOOT0	RW	32	0x0000 1CA8	0x4A00 3CA8
CTRL_CORE_ROM_AUXBOOT1	RW	32	0x0000 1CAC	0x4A00 3CAC
CTRL_CORE_SMA_SW_18	R	32	0x0000 1CB0	0x4A00 3CB0
CTRL_CORE_SMA_SW_19	R	32	0x0000 1CB4	0x4A00 3CB4
CTRL_CORE_SMA_SW_20	R	32	0x0000 1CB8	0x4A00 3CB8
CTRL_CORE_SMA_SW_21	R	32	0x0000 1CBC	0x4A00 3CBC
CTRL_CORE_SMA_SW_22	RW	32	0x0000 1CC0	0x4A00 3CC0
CTRL_CORE_SMA_SW_23	RW	32	0x0000 1CC4	0x4A00 3CC4
CTRL_CORE_SMA_SW_24	RW	32	0x0000 1CC8	0x4A00 3CC8

Table 13-25. CTRL_MODULE_CORE Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	CTRL_MODULE_CORE Base Address
CTRL_CORE_SMA_SW_25	RW	32	0x0000 1CCC	0x4A00 3CCC
CTRL_CORE_SMA_SW_26	RW	32	0x0000 1CD0	0x4A00 3CD0
CTRL_CORE_SMA_SW_27	RW	32	0x0000 1CD4	0x4A00 3CD4

13.5.2.2 CTRL_MODULE_CORE Register Description

Table 13-26. CTRL_CORE_STATUS

Address Offset	0x0000 0134		
Physical Address	0x4A00 2134	**Instance**	CTRL_MODULE_CORE
Description	Control Module Status Register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								DEVICE_TYPE			RESERVED				

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8:6	DEVICE_TYPE	Device type captured at reset time. Device type value sampled at power-on reset. Read 0x3 = General Purpose (GP)	R	0x3
5:0	RESERVED	Reserved	R	0x0

Table 13-27. Register Call Summary for Register CTRL_CORE_STATUS

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-28. CTRL_CORE_SEC_ERR_STATUS_FUNC_1

Address Offset	0x0000 0148		
Physical Address	0x4A00 2148	**Instance**	CTRL_MODULE_CORE
Description	Firewall Error Status functional Register 1		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED			EVE1_FW_ERROR		RESERVED			L4_WAKEUP_FW_ERROR		RESERVED			DEBUGSS_FW_ERROR	L4_CONFIG_FW_ERROR	L4_PERIPH1_FW_ERROR	RESERVED	DSS_FW_ERROR		RESERVED					IPU1_FW_ERROR	RESERVED		EMIF_FW_ERROR	GPMC_FW_ERROR	L3RAM1_FW_ERROR	RESERVED	

Bits	Field Name	Description	Type	Reset
31:29	RESERVED		R	0x0
28	EVE1_FW_ERROR	EVE1 firewall 0x0 = No error from firewall 0x1 = Error from firewall	RW W1toClr	0x0
27:23	RESERVED		R	0x0
22	L4_WAKEUP_FW_ERROR	L4 wakeup firewall 0x0 = No error from firewall 0x1 = Error from firewall	RW W1toClr	0x0
21:19	RESERVED		R	0x0
18	DEBUGSS_FW_ERROR	DebugSS firewall 0x0 = No error from firewall 0x1 = Error from firewall	RW W1toClr	0x0
17	L4_CONFIG_FW_ERROR	L4 config firewall 0x0 = No error from firewall 0x1 = Error from firewall	RW W1toClr	0x0
16	L4_PERIPH1_FW_ERROR	L4 periph1 firewall 0x0 = No error from firewall 0x1 = Error from firewall	RW W1toClr	0x0
15	RESERVED		R	0x0
14	DSS_FW_ERROR	DSS firewall 0x0 = No error from firewall 0x1 = Error from firewall	RW W1toClr	0x0
13:6	RESERVED		R	0x0
5	IPU1_FW_ERROR	IPU1 firewall 0x0 = No error from firewall 0x1 = Error from firewall	RW W1toClr	0x0
4	RESERVED		R	0x0
3	EMIF_FW_ERROR	EMIF firewall 0x0 = No error from firewall 0x1 = Error from firewall	RW W1toClr	0x0
2	GPMC_FW_ERROR	GPMC firewall 0x0 = No error from firewall 0x1 = Error from firewall	RW W1toClr	0x0
1	L3RAM1_FW_ERROR	L3RAM1 firewall 0x0 = No error from firewall 0x1 = Error from firewall	RW W1toClr	0x0
0	RESERVED		R	0x0

Table 13-29. Register Call Summary for Register CTRL_CORE_SEC_ERR_STATUS_FUNC_1

Control Module Functional Description • Firewall Error Status Registers: [0][1]
Control Module Register Manual • CTRL_MODULE_CORE Register Summary: [2]

Table 13-30. CTRL_CORE_SEC_ERR_STATUS_DEBUG_1

Address Offset	0x0000 0150		
Physical Address	0x4A00 2150	**Instance**	CTRL_MODULE_CORE
Description	Firewall Error Status Debug Register 1		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED			EVE1_DBGFW_ERROR	RESERVED				L4_WAKEUP_DBGFW_ERROR	RESERVED				DEBUGSS_DBGFW_ERROR	L4_CONFIG_DBGFW_ERROR	L4_PERIPH1_DBGFW_ERROR	RESERVED	DSS_DBGFW_ERROR	RESERVED						IPU1_DBGFW_ERROR	RESERVED	EMIF_DBGFW_ERROR	GPMC_DBGFW_ERROR	L3RAM1_DBGFW_ERROR	RESERVED		

Bits	Field Name	Description	Type	Reset
31:29	RESERVED		R	0x0
28	EVE1_DBGFW_ERROR	EVE1 firewall 0x0 = No error from firewall 0x1 = Error from firewall	RW W1toClr	0x0
27:23	RESERVED		R	0x0
22	L4_WAKEUP_DBGFW_ERROR	L4 wakeup firewall 0x0 = No error from firewall 0x1 = Error from firewall	RW W1toClr	0x0
21:19	RESERVED		R	0x0
18	DEBUGSS_DBGFW_ERROR	DebugSS firewall 0x0 = No error from firewall 0x1 = Error from firewall	RW W1toClr	0x0
17	L4_CONFIG_DBGFW_ERROR	L4 config firewall 0x0 = No error from firewall 0x1 = Error from firewall	RW W1toClr	0x0
16	L4_PERIPH1_DBGFW_ERROR	L4 periph1 firewall 0x0 = No error from firewall 0x1 = Error from firewall	RW W1toClr	0x0
15	RESERVED		R	0x0
14	DSS_DBGFW_ERROR	DSS debug firewall 0x0 = No error from firewall 0x1 = Error from firewall	RW W1toClr	0x0
13:6	RESERVED		R	0x0
5	IPU1_DBGFW_ERROR	IPU1 debug firewall 0x0 = No error from firewall 0x1 = Error from firewall	RW W1toClr	0x0
4	RESERVED		R	0x0
3	EMIF_DBGFW_ERROR	EMIF debug firewall 0x0 = No error from firewall 0x1 = Error from firewall	RW W1toClr	0x0
2	GPMC_DBGFW_ERROR	GPMC debug firewall 0x0 = No error from firewall 0x1 = Error from firewall	RW W1toClr	0x0

Bits	Field Name	Description	Type	Reset
1	L3RAM1_DBGFW_ERROR	L3RAM1 debug firewall 0x0 = No error from firewall 0x1 = Error from firewall	RW W1toClr	0x0
0	RESERVED		R	0x0

Table 13-31. Register Call Summary for Register CTRL_CORE_SEC_ERR_STATUS_DEBUG_1

Control Module Functional Description • Firewall Error Status Registers: [0]
Control Module Register Manual • CTRL_MODULE_CORE Register Summary: [1]

Table 13-32. CTRL_CORE_STD_FUSE_OPP_VDD_CORE_0

Address Offset	0x0000 01CC		
Physical Address	0x4A00 21CC	**Instance**	CTRL_MODULE_CORE
Description	Standard Fuse OPP VDD_CORE [31:0]. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STD_FUSE_OPP_VDD_CORE_0																															

Bits	Field Name	Description	Type	Reset
31:0	STD_FUSE_OPP_VDD_CORE_0		R	0x0

Table 13-33. Register Call Summary for Register CTRL_CORE_STD_FUSE_OPP_VDD_CORE_0

Control Module Register Manual • CTRL_MODULE_CORE Register Summary: [0]

Table 13-34. CTRL_CORE_STD_FUSE_OPP_VDD_CORE_1

Address Offset	0x0000 01D0		
Physical Address	0x4A00 21D0	**Instance**	CTRL_MODULE_CORE
Description	Standard Fuse OPP VDD_CORE [63:32]. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STD_FUSE_OPP_VDD_CORE_1																															

Bits	Field Name	Description	Type	Reset
31:0	STD_FUSE_OPP_VDD_CORE_1		R	0x0

Table 13-35. Register Call Summary for Register CTRL_CORE_STD_FUSE_OPP_VDD_CORE_1

Control Module Register Manual • CTRL_MODULE_CORE Register Summary: [0]

Table 13-36. CTRL_CORE_STD_FUSE_OPP_VDD_CORE_2

Address Offset	0x0000 01D4
Physical Address	0x4A00 21D4 **Instance** CTRL_MODULE_CORE
Description	Standard Fuse OPP VDD_CORE [95:64]. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.
Type	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STD_FUSE_OPP_VDD_CORE_2																															

Bits	Field Name	Description	Type	Reset
31:0	STD_FUSE_OPP_VDD_CORE_2		R	0x0

Table 13-37. Register Call Summary for Register CTRL_CORE_STD_FUSE_OPP_VDD_CORE_2

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-38. CTRL_CORE_STD_FUSE_OPP_VDD_CORE_3

Address Offset	0x0000 01D8
Physical Address	0x4A00 21D8 **Instance** CTRL_MODULE_CORE
Description	Standard Fuse OPP VDD_CORE [127:96]. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.
Type	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STD_FUSE_OPP_VDD_CORE_3																															

Bits	Field Name	Description	Type	Reset
31:0	STD_FUSE_OPP_VDD_CORE_3		R	0x0

Table 13-39. Register Call Summary for Register CTRL_CORE_STD_FUSE_OPP_VDD_CORE_3

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-40. CTRL_CORE_STD_FUSE_OPP_VDD_CORE_4

Address Offset	0x0000 01DC
Physical Address	0x4A00 21DC **Instance** CTRL_MODULE_CORE
Description	Standard Fuse OPP VDD_CORE [159:128]. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.
Type	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STD_FUSE_OPP_VDD_CORE_4																															

Bits	Field Name	Description	Type	Reset
31:0	STD_FUSE_OPP_VDD_CORE_4		R	0x0

Table 13-41. Register Call Summary for Register CTRL_CORE_STD_FUSE_OPP_VDD_CORE_4

Control Module Register Manual

- CTRL_MODULE_CORE Register Summary: [0]

Table 13-42. CTRL_CORE_STD_FUSE_OPP_BGAP_CORE

Address Offset	0x0000 01E8		
Physical Address	0x4A00 21E8	**Instance**	CTRL_MODULE_CORE
Description	Trim values for CORE associated temperature sensor and bandgap. Contains TI Internal information, not intended for application use.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STD_FUSE_OPP_BGAP_CORE_0								STD_FUSE_OPP_BGAP_CORE_1								STD_FUSE_OPP_BGAP_CORE_2								STD_FUSE_OPP_BGAP_CORE_3							

Bits	Field Name	Description	Type	Reset
31:24	STD_FUSE_OPP_BGAP_CORE_0	Trim values for CORE associated temperature sensor and bandgap. Contains TI Internal information, not intended for application use.	R	0x-
23:16	STD_FUSE_OPP_BGAP_CORE_1	Trim values for CORE associated temperature sensor and bandgap. Contains TI Internal information, not intended for application use.	R	0x-
15:8	STD_FUSE_OPP_BGAP_CORE_2	Trim values for CORE associated temperature sensor and bandgap. Contains TI Internal information, not intended for application use.	R	0x-
7:0	STD_FUSE_OPP_BGAP_CORE_3	Trim values for CORE associated temperature sensor and bandgap. Contains TI Internal information, not intended for application use.	R	0x-

Table 13-43. Register Call Summary for Register CTRL_CORE_STD_FUSE_OPP_BGAP_CORE

Control Module Register Manual

- CTRL_MODULE_CORE Register Summary: [0]

Table 13-44. CTRL_CORE_STD_FUSE_MPK_0

Address Offset	0x0000 0220		
Physical Address	0x4A00 2220	**Instance**	CTRL_MODULE_CORE
Description	Standard Fuse keys. Root_public_key_hash [31:0]. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STD_FUSE_MPK_0																															

Bits	Field Name	Description	Type	Reset
31:0	STD_FUSE_MPK_0		R	0x0

Table 13-45. Register Call Summary for Register CTRL_CORE_STD_FUSE_MPK_0

Control Module Register Manual

- CTRL_MODULE_CORE Register Summary: [0]

Table 13-46. CTRL_CORE_STD_FUSE_MPK_1

Address Offset	0x0000 0224
Physical Address	0x4A00 2224 **Instance** CTRL_MODULE_CORE
Description	Standard Fuse keys. Root_public_key_hash [63:32]. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.
Type	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STD_FUSE_MPK_1																															

Bits	Field Name	Description	Type	Reset
31:0	STD_FUSE_MPK_1		R	0x0

Table 13-47. Register Call Summary for Register CTRL_CORE_STD_FUSE_MPK_1

Control Module Register Manual

- CTRL_MODULE_CORE Register Summary: [0]

Table 13-48. CTRL_CORE_STD_FUSE_MPK_2

Address Offset	0x0000 0228
Physical Address	0x4A00 2228 **Instance** CTRL_MODULE_CORE
Description	Standard Fuse keys. Root_public_key_hash [95:64]. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.
Type	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STD_FUSE_MPK_2																															

Bits	Field Name	Description	Type	Reset
31:0	STD_FUSE_MPK_2		R	0x0

Table 13-49. Register Call Summary for Register CTRL_CORE_STD_FUSE_MPK_2

Control Module Register Manual

- CTRL_MODULE_CORE Register Summary: [0]

Table 13-50. CTRL_CORE_STD_FUSE_MPK_3

Address Offset	0x0000 022C
Physical Address	0x4A00 222C **Instance** CTRL_MODULE_CORE
Description	Standard Fuse keys. Root_public_key_hash [127:96]. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.
Type	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STD_FUSE_MPK_3																															

Bits	Field Name	Description	Type	Reset
31:0	STD_FUSE_MPK_3		R	0x0

Table 13-51. Register Call Summary for Register CTRL_CORE_STD_FUSE_MPK_3

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-52. CTRL_CORE_STD_FUSE_MPK_4

Address Offset	0x0000 0230
Physical Address	0x4A00 2230 **Instance** CTRL_MODULE_CORE
Description	Standard Fuse keys. Root_public_key_hash [159:128]. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.
Type	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STD_FUSE_MPK_4																															

Bits	Field Name	Description	Type	Reset
31:0	STD_FUSE_MPK_4		R	0x0

Table 13-53. Register Call Summary for Register CTRL_CORE_STD_FUSE_MPK_4

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-54. CTRL_CORE_STD_FUSE_MPK_5

Address Offset	0x0000 0234
Physical Address	0x4A00 2234 **Instance** CTRL_MODULE_CORE
Description	Standard Fuse keys. Root_public_key_hash [191:160]. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.
Type	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STD_FUSE_MPK_5																															

Bits	Field Name	Description	Type	Reset
31:0	STD_FUSE_MPK_5		R	0x0

Table 13-55. Register Call Summary for Register CTRL_CORE_STD_FUSE_MPK_5

Control Module Register Manual

- CTRL_MODULE_CORE Register Summary: [0]

Table 13-56. CTRL_CORE_STD_FUSE_MPK_6

Address Offset	0x0000 0238
Physical Address	0x4A00 2238 **Instance** CTRL_MODULE_CORE
Description	Standard Fuse keys. Root_public_key_hash [223:192]. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.
Type	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STD_FUSE_MPK_6																															

Bits	Field Name	Description	Type	Reset
31:0	STD_FUSE_MPK_6		R	0x0

Table 13-57. Register Call Summary for Register CTRL_CORE_STD_FUSE_MPK_6

Control Module Register Manual

- CTRL_MODULE_CORE Register Summary: [0]

Table 13-58. CTRL_CORE_STD_FUSE_MPK_7

Address Offset	0x0000 023C
Physical Address	0x4A00 223C **Instance** CTRL_MODULE_CORE
Description	Standard Fuse keys. Root_public_key_hash [255:224]. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.
Type	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STD_FUSE_MPK_7																															

Bits	Field Name	Description	Type	Reset
31:0	STD_FUSE_MPK_7		R	0x0

Table 13-59. Register Call Summary for Register CTRL_CORE_STD_FUSE_MPK_7

Control Module Register Manual

- CTRL_MODULE_CORE Register Summary: [0]

Table 13-60. CTRL_CORE_CUST_FUSE_SWRV_0

Address Offset	0x0000 02BC
Physical Address	0x4A00 22BC **Instance** CTRL_MODULE_CORE
Description	Customer Fuse keys. Software Version Control [031:000] (16 bits upper Redundant field) [FIELD OVERFLOW]. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.
Type	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CUST_FUSE_SWRV_0																															

Bits	Field Name	Description	Type	Reset
31:0	CUST_FUSE_SWRV_0		R	0x0

Table 13-61. Register Call Summary for Register CTRL_CORE_CUST_FUSE_SWRV_0

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-62. CTRL_CORE_CUST_FUSE_SWRV_1

Address Offset	0x0000 02C0
Physical Address	0x4A00 22C0 **Instance** CTRL_MODULE_CORE
Description	Customer Fuse keys. Software Version Control [063:032] (16 bits upper Redundant field) [FIELD F]. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.
Type	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CUST_FUSE_SWRV_1																															

Bits	Field Name	Description	Type	Reset
31:0	CUST_FUSE_SWRV_1		R	0x0

Table 13-63. Register Call Summary for Register CTRL_CORE_CUST_FUSE_SWRV_1

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-64. CTRL_CORE_CUST_FUSE_SWRV_2

Address Offset	0x0000 02C4
Physical Address	0x4A00 22C4 **Instance** CTRL_MODULE_CORE
Description	Customer Fuse keys. Software Version Control [095:064] (16 bits upper Redundant field) [FIELD E]. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.
Type	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CUST_FUSE_SWRV_2																															

Bits	Field Name	Description	Type	Reset
31:0	CUST_FUSE_SWRV_2		R	0x0

Table 13-65. Register Call Summary for Register CTRL_CORE_CUST_FUSE_SWRV_2

Control Module Register Manual

- CTRL_MODULE_CORE Register Summary: [0]

Table 13-66. CTRL_CORE_CUST_FUSE_SWRV_3

Address Offset	0x0000 02C8
Physical Address	0x4A00 22C8 **Instance** CTRL_MODULE_CORE
Description	Customer Fuse keys. Software Version Control [127:096] (16 bits upper Redundant field) [FIELD D]. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.
Type	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CUST_FUSE_SWRV_3																															

Bits	Field Name	Description	Type	Reset
31:0	CUST_FUSE_SWRV_3		R	0x0

Table 13-67. Register Call Summary for Register CTRL_CORE_CUST_FUSE_SWRV_3

Control Module Register Manual

- CTRL_MODULE_CORE Register Summary: [0]

Table 13-68. CTRL_CORE_CUST_FUSE_SWRV_4

Address Offset	0x0000 02CC
Physical Address	0x4A00 22CC **Instance** CTRL_MODULE_CORE
Description	Customer Fuse keys. Software Version Control [159:127] (16 bits upper Redundant field) [FIELD C]. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.
Type	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CUST_FUSE_SWRV_4																															

Bits	Field Name	Description	Type	Reset
31:0	CUST_FUSE_SWRV_4		R	0x0

Table 13-69. Register Call Summary for Register CTRL_CORE_CUST_FUSE_SWRV_4

Control Module Register Manual

- CTRL_MODULE_CORE Register Summary: [0]

Table 13-70. CTRL_CORE_CUST_FUSE_SWRV_5

Address Offset	0x0000 02D0
Physical Address	0x4A00 22D0 **Instance** CTRL_MODULE_CORE
Description	Customer Fuse keys. Software Version Control [191:160] (16 bits upper Redundant field) [FIELD B]. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.
Type	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CUST_FUSE_SWRV_5																															

Bits	Field Name	Description	Type	Reset
31:0	CUST_FUSE_SWRV_5		R	0x0

Table 13-71. Register Call Summary for Register CTRL_CORE_CUST_FUSE_SWRV_5

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-72. CTRL_CORE_CUST_FUSE_SWRV_6

Address Offset	0x0000 02D4
Physical Address	0x4A00 22D4 **Instance** CTRL_MODULE_CORE
Description	Customer Fuse keys. Software Version Control [223:192] (16 bits upper Redundant field) [FIELD A]. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.
Type	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CUST_FUSE_SWRV_6																															

Bits	Field Name	Description	Type	Reset
31:0	CUST_FUSE_SWRV_6		R	0x0

Table 13-73. Register Call Summary for Register CTRL_CORE_CUST_FUSE_SWRV_6

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-74. CTRL_CORE_TEMP_SENSOR_CORE

Address Offset	0x0000 0334		
Physical Address	0x4A00 2334	**Instance**	CTRL_MODULE_CORE
Description	Control VBGAPTS temperature sensor and thermal comparator shutdown register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																				BGAP_TMPSOFF_CORE	BGAP_EOCZ_CORE	BGAP_DTEMP_CORE									

Bits	Field Name	Description	Type	Reset
31:12	RESERVED		R	0x0
11	BGAP_TMPSOFF_CORE	This bit indicates the temperature sensor state. 0x0: temperature sensor is ON 0x1: temperature sensor is OFF **NOTE:** Software doesn't take care of this bit to get the temperature data. Only the BGAP_EOCZ_CORE bit is needed.	R	0x1
10	BGAP_EOCZ_CORE	ADC End of Conversion. Active low, when BGAP_DTEMP_CORE is valid.	R	0x0
9:0	BGAP_DTEMP_CORE	Temperature data from the ADC. Valid if EOCZ is low.	R	0x0

Table 13-75. Register Call Summary for Register CTRL_CORE_TEMP_SENSOR_CORE

Control Module Functional Description

- Temperature Sensor Control Registers: [0][1]
- Temperature Timestamp Registers: [2]
- Summary Of The Thermal Management Related Registers: [3]
- ADC Values Versus Temperature: [4]

Control Module Register Manual

- CTRL_MODULE_CORE Register Summary: [5]

Table 13-76. CTRL_CORE_CORTEX_M4_MMUADDRTRANSLTR

Address Offset	0x0000 0358		
Physical Address	0x4A00 2358	**Instance**	CTRL_MODULE_CORE
Description	Cortex M4 register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								CORTEX_M4_MMUADDRTRANSLTR																									

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19:0	CORTEX_M4_MMUADDRTRANSLTR	Used to save the IPU AMMU translated/boot address	RW	0x0

Table 13-77. Register Call Summary for Register CTRL_CORE_CORTEX_M4_MMUADDRTRANSLTR

Control Module Register Manual

- CTRL_MODULE_CORE Register Summary: [0]

Table 13-78. CTRL_CORE_CORTEX_M4_MMUADDRLOGICTR

Address Offset	0x0000 035C		
Physical Address	0x4A00 235C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								CORTEX_M4_MMUADDRLOGICTR																									

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19:0	CORTEX_M4_MMUADDRLOGICTR	Used to save the IPU AMMU logical source address	RW	0x0

Table 13-79. Register Call Summary for Register CTRL_CORE_CORTEX_M4_MMUADDRLOGICTR

Control Module Register Manual

- CTRL_MODULE_CORE Register Summary: [0]

Table 13-80. CTRL_CORE_HWOBS_CONTROL

Address Offset	0x0000 0360		
Physical Address	0x4A00 2360	**Instance**	CTRL_MODULE_CORE
Description	HW observability control. This register enables or disables HW observability outputs (to save power primarily)		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED													HWOBS_CLKDIV_SEL_2					HWOBS_CLKDIV_SEL_1				RESERVED		HWOBS_CLKDIV_SEL					HWOBS_ALL_ZERO_MODE	HWOBS_ALL_ONE_MODE	HWOBS_MACRO_ENABLE

Bits	Field Name	Description	Type	Reset
31:19	RESERVED		R	0x0
18:14	HWOBS_CLKDIV_SEL_2	Clock divider selection for obs2 line. 0x1 = output is not divided 0x2 = output is divided by 2 0x4 = output is divided by 4 0x8 = output is divided by 8 0x10 = output is divided by 16	RW	0x0

Bits	Field Name	Description	Type	Reset
13:9	HWOBS_CLKDIV_SEL_1	Clock divider selection for obs1 line. 0x1 = output is not divided 0x2 = output is divided by 2 0x4 = output is divided by 4 0x8 = output is divided by 8 0x10 = output is divided by 16	RW	0x0
8	RESERVED		R	0x0
7:3	HWOBS_CLKDIV_SEL	Clock divider selection for obs0 line. 0x1 = output is not divided 0x2 = output is divided by 2 0x4 = output is divided by 4 0x8 = output is divided by 8 0x10 = output is divided by 16	RW	0x0
2	HWOBS_ALL_ZERO_MODE	Used to gate observable signals. When set all outputs are set to zero (can be used to check the path from HW observability to external pads). 0x0 = hw observability ports are not gated 0x1 = hw observability ports are all set to 0	RW	0x0
1	HWOBS_ALL_ONE_MODE	Used to gate observable signals. When set all outputs are set to one (can be used to check the path from HW observability to external pads). 0x0 = hw observability ports are not gated 0x1 = hw observability ports are all set to 1	RW	0x0
0	HWOBS_MACRO_ENABLE	Used to gate observable signals coming from macros using the 32-bit HWOBS bus definition. When deasserted all outputs of the HWOBS busdef are set to zero. 0x0 = hw observability ports from macros are gated and set to zero 0x1 = hw observability ports from macros are not gated	RW	0x0

Table 13-81. Register Call Summary for Register CTRL_CORE_HWOBS_CONTROL

Control Module Functional Description
- Hardware Observability Related Registers: [0]

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [1]

Table 13-82. CTRL_CORE_BANDGAP_MASK_1

Address Offset	0x0000 0380		
Physical Address	0x4A00 2380	**Instance**	CTRL_MODULE_CORE
Description	bgap_mask		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SIDLEMODE		COUNTER_DELAY			RESERVED			FREEZE_CORE	RESERVED		CLEAR_CORE	RESERVED														MASK_HOT_CORE	MASK_COLD_CORE	RESERVED				

Bits	Field Name	Description	Type	Reset
31:30	SIDLEMODE	sidlemode for bandgap 0x0 = No Idle 0x1 = Force Idle 0x2 = Smart Idle 0x3 = Reserved	RW	0x0
29:27	COUNTER_DELAY	Counter delay 0x0 = Imediat 0x1 = Delay of 1ms 0x2 = Delay of 10ms 0x3 = Delay of 100ms 0x4 = Delay of 250ms 0x5 = Delay of 500ms	RW	0x0
26:24	RESERVED		R	0x0
23	FREEZE_CORE	Freeze the FIFO CORE 0x0 = No operation 0x1 = Freeze the FIFO	RW	0x0
22:21	RESERVED		R	0x0
20	CLEAR_CORE	Reset the FIFO CORE 0x0 = No operation 0x1 = Reset the FIFO	RW	0x0
19:6	RESERVED		R	0x0
5	MASK_HOT_CORE	Mask for hot event CORE 0x0 = hot event is masked 0x1 = hot event is not masked	RW	0x0
4	MASK_COLD_CORE	Mask for cold event CORE 0x0 = cold event is masked 0x1 = cold event is not masked	RW	0x0
3:0	RESERVED		R	0x0

Table 13-83. Register Call Summary for Register CTRL_CORE_BANDGAP_MASK_1

Control Module Functional Description

- Control Module Clock Configuration: [0]
- Temperature Sensor Control Registers: [1]
- Registers For The Thermal Alert Comparator Block: [2][3]
- Other Thermal Management Related Registers: [4][5][6]
- Summary Of The Thermal Management Related Registers: [7]

Control Module Register Manual

- CTRL_MODULE_CORE Register Summary: [8]

Table 13-84. CTRL_CORE_BANDGAP_THRESHOLD_CORE

Address Offset	0x0000 038C		
Physical Address	0x4A00 238C	**Instance**	CTRL_MODULE_CORE
Description	BGAP THRESHOLD CORE		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								THOLD_HOT_CORE								RESERVED								THOLD_COLD_CORE							

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25:16	THOLD_HOT_CORE	Value for the high temperature threshold. The values for loading this bit field are listed in Table 13-10.	RW	0x0
15:10	RESERVED		R	0x0
9:0	THOLD_COLD_CORE	Value for the low temperature threshold. The values for loading this bit field are listed in Table 13-10.	RW	0x0

Table 13-85. Register Call Summary for Register CTRL_CORE_BANDGAP_THRESHOLD_CORE

Control Module Functional Description

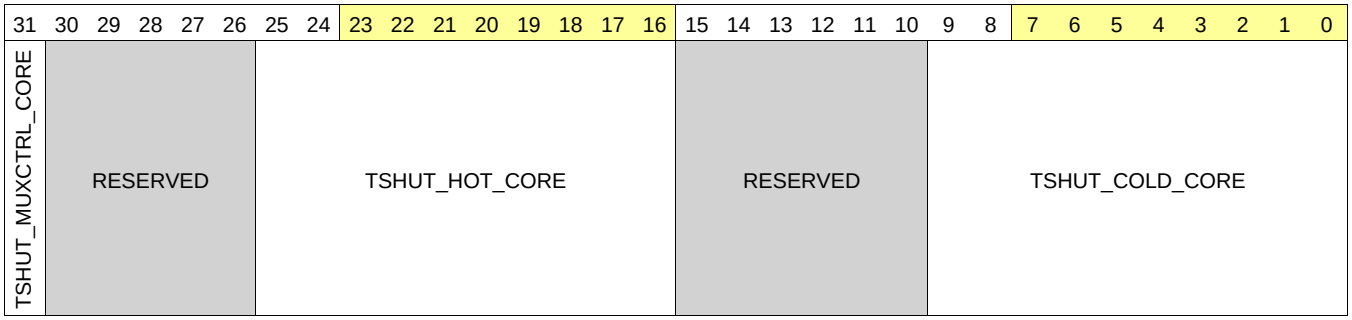
- Registers For The Thermal Alert Comparator Block: [0][1]
- Summary Of The Thermal Management Related Registers: [2]
- ADC Values Versus Temperature: [3][4]

Control Module Register Manual

- CTRL_MODULE_CORE Register Summary: [5]

Table 13-86. CTRL_CORE_BANDGAP_TSHUT_CORE

Address Offset	0x0000 0398		
Physical Address	0x4A00 2398	**Instance**	CTRL_MODULE_CORE
Description	BGAP TSHUT THRESHOLD CORE		
Type	RW		



Bits	Field Name	Description	Type	Reset
31	TSHUT_MUXCTRL_CORE	Writing a '1' to this field allows SW to override the TSHUT_HOT and TSHUT_COLD values that are set by default in efuse.	RW	0x0
30:26	RESERVED		R	0x0
25:16	TSHUT_HOT_CORE	Controls the TSHUT_HOT reset threshold, which protects the device from thermal runaway and potential device damage. The register defaults to 123°C. **Software override of this register value is not recommended, and should only be done with extreme caution as damage to the device can occur above the default setting.**	RW	0x0
15:10	RESERVED		R	0x0
9:0	TSHUT_COLD_CORE	Controls the TSHUT_COLD reset threshold, which is the limit where the TSHUT comparator releases the device from reset after cooling from a TSHUT condition. The register defaults to 105°C. **Software override of this register value is not recommended, and should only be done with extreme caution.**	RW	0x0

Table 13-87. Register Call Summary for Register CTRL_CORE_BANDGAP_TSHUT_CORE

Control Module Functional Description • Summary Of The Thermal Management Related Registers: [0]
Control Module Register Manual • CTRL_MODULE_CORE Register Summary: [1]

Table 13-88. CTRL_CORE_BANDGAP_STATUS_1

Address Offset	0x0000 03A8		
Physical Address	0x4A00 23A8	**Instance**	CTRL_MODULE_CORE
Description	BGAP STATUS		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ALERT	RESERVED																									HOT_CORE	COLD_CORE	RESERVED			

Bits	Field Name	Description	Type	Reset
31	ALERT	Alert temperature when '1'	R	0x0
30:6	RESERVED		R	0x0
5	HOT_CORE	Event for hot temperature mpu bandgap when '1' 0x0 = event not detected 0x1 = event detected	R	0x0
4	COLD_CORE	Event for cold temperature mpu bandgap when '1' 0x0 = event not detected 0x1 = event detected	R	0x0
3:0	RESERVED		R	0x0

Table 13-89. Register Call Summary for Register CTRL_CORE_BANDGAP_STATUS_1

Control Module Functional Description • Registers For The Thermal Alert Comparator Block: [0][1][2] • Summary Of The Thermal Management Related Registers: [3]
Control Module Register Manual • CTRL_MODULE_CORE Register Summary: [4]

Table 13-90. CTRL_CORE_DTEMP_CORE_0

Address Offset	0x0000 03E8		
Physical Address	0x4A00 23E8	**Instance**	CTRL_MODULE_CORE
Description	TAGGED TEMPERATURE CORE DOMAIN. Most recent sample.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DTEMP_TAG_CORE_0																						DTEMP_TEMPERATURE_CORE_0									

Bits	Field Name	Description	Type	Reset
31:10	DTEMP_TAG_CORE_0	tag. Indicate number of times in the bgap state machine.	R	0x0
9:0	DTEMP_TEMPERATURE_CORE_0	temperature	R	0x0

Table 13-91. Register Call Summary for Register CTRL_CORE_DTEMP_CORE_0

Control Module Functional Description
- Temperature Timestamp Registers: [0]
- Summary Of The Thermal Management Related Registers: [1]

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [2]

Table 13-92. CTRL_CORE_DTEMP_CORE_1

Address Offset	0x0000 03EC		
Physical Address	0x4A00 23EC	**Instance**	CTRL_MODULE_CORE
Description	TAGGED TEMPERATURE CORE DOMAIN		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DTEMP_TAG_CORE_1																DTEMP_TEMPERATURE_CORE_1															

Bits	Field Name	Description	Type	Reset
31:10	DTEMP_TAG_CORE_1	tag. Indicate number of times in the bgap state machine.	R	0x0
9:0	DTEMP_TEMPERATURE_CORE_1	temperature	R	0x0

Table 13-93. Register Call Summary for Register CTRL_CORE_DTEMP_CORE_1

Control Module Functional Description
- Temperature Timestamp Registers: [0]
- Summary Of The Thermal Management Related Registers: [1]

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [2]

Table 13-94. CTRL_CORE_DTEMP_CORE_2

Address Offset	0x0000 03F0		
Physical Address	0x4A00 23F0	**Instance**	CTRL_MODULE_CORE
Description	TAGGED TEMPERATURE CORE DOMAIN		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DTEMP_TAG_CORE_2																DTEMP_TEMPERATURE_CORE_2															

Bits	Field Name	Description	Type	Reset
31:10	DTEMP_TAG_CORE_2	tag. Indicate number of times in the bgap state machine.	R	0x0
9:0	DTEMP_TEMPERATURE_CORE_2	temperature	R	0x0

Table 13-95. Register Call Summary for Register CTRL_CORE_DTEMP_CORE_2

Control Module Functional Description
- Temperature Timestamp Registers: [0]
- Summary Of The Thermal Management Related Registers: [1]

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [2]

Table 13-96. CTRL_CORE_DTEMP_CORE_3

Address Offset	0x0000 03F4		
Physical Address	0x4A00 23F4	**Instance**	CTRL_MODULE_CORE
Description	TAGGED TEMPERATURE CORE DOMAIN		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DTEMP_TAG_CORE_3																						DTEMP_TEMPERATURE_CORE_3									

Bits	Field Name	Description	Type	Reset
31:10	DTEMP_TAG_CORE_3	tag. Indicate number of times in the bgap state machine.	R	0x0
9:0	DTEMP_TEMPERATURE_CORE_3	temperature	R	0x0

Table 13-97. Register Call Summary for Register CTRL_CORE_DTEMP_CORE_3

Control Module Functional Description
- Temperature Timestamp Registers: [0]
- Summary Of The Thermal Management Related Registers: [1]

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [2]

Table 13-98. CTRL_CORE_DTEMP_CORE_4

Address Offset	0x0000 03F8		
Physical Address	0x4A00 23F8	**Instance**	CTRL_MODULE_CORE
Description	TAGGED TEMPERATURE CORE DOMAIN. Oldest sample.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DTEMP_TAG_CORE_4																						DTEMP_TEMPERATURE_CORE_4									

Bits	Field Name	Description	Type	Reset
31:10	DTEMP_TAG_CORE_4	tag. Indicate number of times in the bgap state machine.	R	0x0
9:0	DTEMP_TEMPERATURE_CORE_4	temperature	R	0x0

Table 13-99. Register Call Summary for Register CTRL_CORE_DTEMP_CORE_4

Control Module Functional Description
- Temperature Timestamp Registers: [0]
- Summary Of The Thermal Management Related Registers: [1]

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [2]

Table 13-100. CTRL_CORE_SMA_SW_0

Address Offset	0x0000 03FC		
Physical Address	0x4A00 23FC	**Instance**	CTRL_MODULE_CORE
Description	Inversion control for SD_DAC input data – DIN[9:0].		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																													INPUTINV	RESERVED	CKE_ASSERTION

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0
2	INPUTINV	Inversion control for SD_DAC input data – DIN[9:0]. Allows polarity alignment with VENC data. 0x0: This value should be used for inverted video input data 0x1: This value should be used for non-inverted video input data	RW	0x0
1	RESERVED		R	0x0
0	CKE_ASSERTION	Forces the EMIF CKE pad to tri-state. 0x0: The CKE pad is not in tri-state and can be controlled by EMIF 0x1: The CKE pad is in tri-state	RW	0x0

Table 13-101. Register Call Summary for Register CTRL_CORE_SMA_SW_0

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-102. CTRL_CORE_SEC_ERR_STATUS_FUNC_2

Address Offset	0x0000 0414		
Physical Address	0x4A00 2414	**Instance**	CTRL_MODULE_CORE
Description	Firewall Error Status functional Register 2		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED					TC1_EDMA_FW_ERROR	RESERVED		QSPI_FW_ERROR	RESERVED					TPCC_EDMA_FW_ERROR	TC0_EDMA_FW_ERROR	RESERVED				MCASP1_FW_ERROR	RESERVED			L4_PERIPH3_FW_ERROR	L4_PERIPH2_FW_ERROR	RESERVED		DSP2_FW_ERROR	DSP1_FW_ERROR

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26	TC1_EDMA_FW_ERROR	EDMA TC1 firewall 0x0 = No error from firewall 0x1 = Error from firewall	RW W1toClr	0x0
25:23	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
22	QSPI_FW_ERROR	QSPI firewall 0x0 = No error from firewall 0x1 = Error from firewall	RW W1toClr	0x0
21:18	RESERVED		R	0x0
17	TPCC_EDMA_FW_ERROR	EDMA TPCC firewall 0x0 = No error from firewall 0x1 = Error from firewall	RW W1toClr	0x0
16	TC0_EDMA_FW_ERROR	EDMA TC0 firewall 0x0 = No error from firewall 0x1 = Error from firewall	RW W1toClr	0x0
15:12	RESERVED		R	0x0
11	MCASP1_FW_ERROR	McASP1 firewall 0x0 = No error from firewall 0x1 = Error from firewall	RW W1toClr	0x0
10:6	RESERVED		R	0x0
5	L4_PERIPH3_FW_ERROR	L4 periph3 init firewall 0x0 = No error from firewall 0x1 = Error from firewall	RW W1toClr	0x0
4	L4_PERIPH2_FW_ERROR	L4 periph2 init firewall 0x0 = No error from firewall 0x1 = Error from firewall	RW W1toClr	0x0
3:2	RESERVED		R	0x0
1	DSP2_FW_ERROR	DSP2 firewall 0x0 = No error from firewall 0x1 = Error from firewall	RW W1toClr	0x0
0	DSP1_FW_ERROR	DSP1 firewall 0x0 = No error from firewall 0x1 = Error from firewall	RW W1toClr	0x0

Table 13-103. Register Call Summary for Register CTRL_CORE_SEC_ERR_STATUS_FUNC_2

Control Module Functional Description • Firewall Error Status Registers: [0][1]
Control Module Register Manual • CTRL_MODULE_CORE Register Summary: [2]

Table 13-104. CTRL_CORE_SEC_ERR_STATUS_DEBUG_2

Address Offset	0x0000 041C		
Physical Address	0x4A00 241C	**Instance**	CTRL_MODULE_CORE
Description	Firewall Error Status debug Register 2		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED					TC1_EDMA_DBGFW_ERROR	RESERVED			QSPI_DBGFW_ERROR	RESERVED				TPCC_EDMA_DBGFW_ERROR	TC0_EDMA_DBGFW_ERROR	RESERVED				MCASP1_DBGFW_ERROR	RESERVED					L4_PERIPH3_DBGFW_ERROR	L4_PERIPH2_DBGFW_ERROR	RESERVED		DSP2_DBGFW_ERROR	DSP1_DBGFW_ERROR

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26	TC1_EDMA_DBGFW_ERROR	EDMA TC1 debug firewall 0x0 = No error from firewall 0x1 = Error from firewall	RW W1toClr	0x0
25:23	RESERVED		R	0x0
22	QSPI_DBGFW_ERROR	QSPI debug firewall 0x0 = No error from firewall 0x1 = Error from firewall	RW W1toClr	0x0
21:18	RESERVED		R	0x0
17	TPCC_EDMA_DBGFW_ERROR	EDMA TPCC debug firewall 0x0 = No error from firewall 0x1 = Error from firewall	RW W1toClr	0x0
16	TC0_EDMA_DBGFW_ERROR	EDMA TC0 debug firewall 0x0 = No error from firewall 0x1 = Error from firewall	RW W1toClr	0x0
15:12	RESERVED		R	0x0
11	MCASP1_DBGFW_ERROR	McASP1 debug firewall 0x0 = No error from firewall 0x1 = Error from firewall	RW W1toClr	0x0
10:6	RESERVED		R	0x0
5	L4_PERIPH3_DBGFW_ERROR	L4 periph3 init firewall 0x0 = No error from firewall 0x1 = Error from firewall	RW W1toClr	0x0
4	L4_PERIPH2_DBGFW_ERROR	L4 periph2 init firewall 0x0 = No error from firewall 0x1 = Error from firewall	RW W1toClr	0x0
3:2	RESERVED		R	0x0
1	DSP2_DBGFW_ERROR	DSP2 firewall 0x0 = No error from firewall 0x1 = Error from firewall	RW W1toClr	0x0

Bits	Field Name	Description	Type	Reset
0	DSP1_DBGFW_ERROR	DSP1 firewall 0x0 = No error from firewall 0x1 = Error from firewall	RW W1toClr	0x0

Table 13-105. Register Call Summary for Register CTRL_CORE_SEC_ERR_STATUS_DEBUG_2

Control Module Functional Description • Firewall Error Status Registers: [0]
Control Module Register Manual • CTRL_MODULE_CORE Register Summary: [1]

Table 13-106. CTRL_CORE_EMIF_INITIATOR_PRIORITY_1

Address Offset	0x0000 0420		
Physical Address	0x4A00 2420	**Instance**	CTRL_MODULE_CORE
Description	Register for priority settings for EMIF arbitration		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																DSP1_CFG_EMIF_PRIORITY				RESERVED				DSP1_EDMA_EMIF_PRIORITY				RESERVED				DSP2_EDMA_EMIF_PRIORITY				RESERVED				DSP2_CFG_EMIF_PRIORITY			

Bits	Field Name	Description	Type	Reset
31:15	RESERVED		R	0x8888
14:12	DSP1_CFG_EMIF_PRIORITY	DSP1 CFG priority setting 0x0 = highest priority 0x7 = lowest priority	RW	0x4
11	RESERVED		R	0x0
10:8	DSP1_EDMA_EMIF_PRIORITY	DSP1 EDMA priority setting 0x0 = highest priority 0x7 = lowest priority	RW	0x4
7	RESERVED		R	0x0
6:4	DSP2_EDMA_EMIF_PRIORITY	DSP2 EDMA priority setting 0x0 = highest priority 0x7 = lowest priority	RW	0x4
3	RESERVED		R	0x0
2:0	DSP2_CFG_EMIF_PRIORITY	DSP2 CFG priority setting 0x0 = highest priority 0x7 = lowest priority	RW	0x4

Table 13-107. Register Call Summary for Register CTRL_CORE_EMIF_INITIATOR_PRIORITY_1

Control Module Functional Description • SDRAM Initiator Priority Registers: [0]
Control Module Register Manual • CTRL_MODULE_CORE Register Summary: [1]

Table 13-108. CTRL_CORE_EMIF_INITIATOR_PRIORITY_2

Address Offset	0x0000 0424		
Physical Address	0x4A00 2424	**Instance**	CTRL_MODULE_CORE
Description	Register for priority settings for EMIF arbitration		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED													EVE1_TC0_EMIF_PRIORITY			RESERVED															

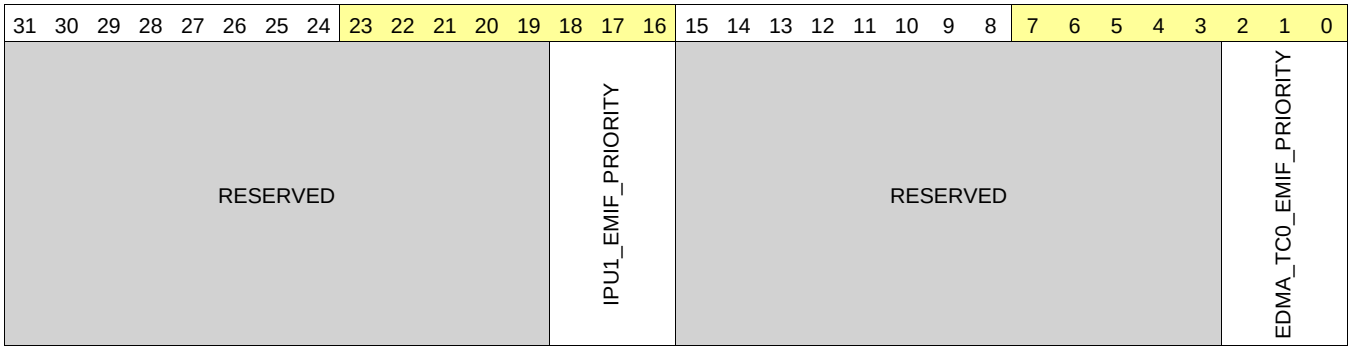
Bits	Field Name	Description	Type	Reset
31:19	RESERVED		R	0x880
18:16	EVE1_TC0_EMIF_PRIORITY	EVE1 TC0 priority setting 0x0 = highest priority 0x7 = lowest priority	RW	0x4
15:0	RESERVED		R	0x4444

Table 13-109. Register Call Summary for Register CTRL_CORE_EMIF_INITIATOR_PRIORITY_2

Control Module Register Manual • CTRL_MODULE_CORE Register Summary: [0]

Table 13-110. CTRL_CORE_EMIF_INITIATOR_PRIORITY_3

Address Offset	0x0000 0428		
Physical Address	0x4A00 2428	**Instance**	CTRL_MODULE_CORE
Description	Register for priority settings for EMIF arbitration		
Type	RW		



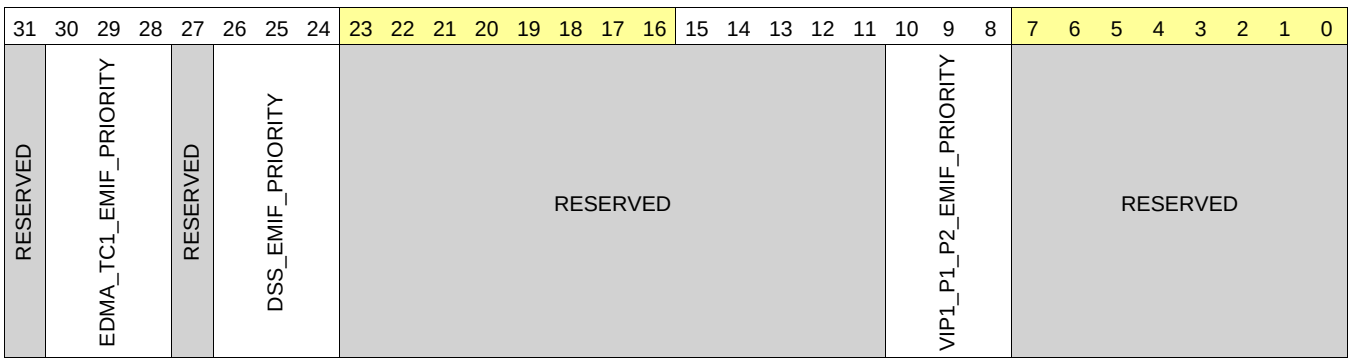
Bits	Field Name	Description	Type	Reset
31:19	RESERVED		R	0x888
18:16	IPU1_EMIF_PRIORITY	IPU1 priority setting 0x0 = highest prioroty 0x7 = lowest priority	RW	0x4
15:3	RESERVED		R	0x888
2:0	EDMA_TC0_EMIF_PRIORITY	EDMA TC0 priority setting 0x0 = highest priority 0x7 = lowest priority	RW	0x4

Table 13-111. Register Call Summary for Register CTRL_CORE_EMIF_INITIATOR_PRIORITY_3

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-112. CTRL_CORE_EMIF_INITIATOR_PRIORITY_4

Address Offset	0x0000 042C		
Physical Address	0x4A00 242C	**Instance**	CTRL_MODULE_CORE
Description	Register for priority settings for EMIF arbitration		
Type	RW		



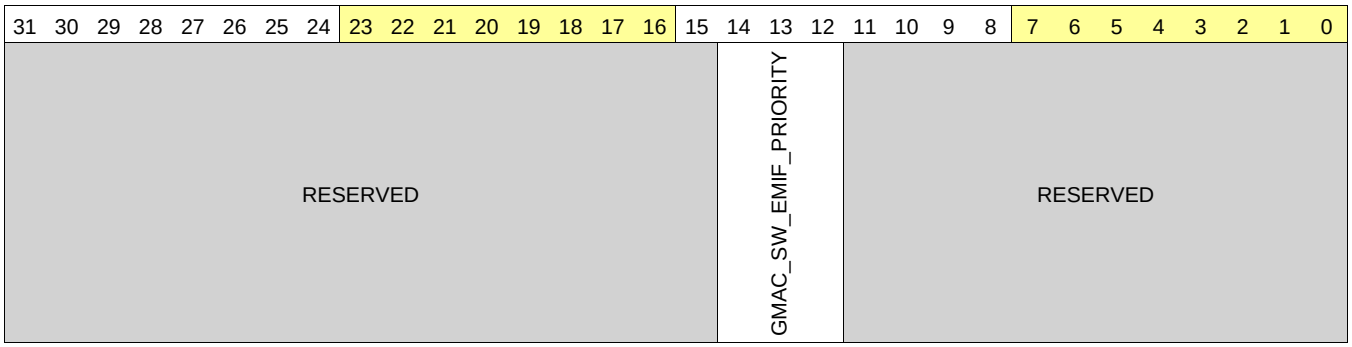
Bits	Field Name	Description	Type	Reset
31	RESERVED		R	0x0
30:28	EDMA_TC1_EMIF_PRIORITY	EDMA TC1 priority setting 0x0 = highest priority 0x7 = lowest priority	RW	0x4
27	RESERVED		R	0x0
26:24	DSS_EMIF_PRIORITY	DSS priority setting 0x0 = highest priority 0x7 = lowest priority	RW	0x4
23:11	RESERVED		R	0x888
10:8	VIP1_P1_P2_EMIF_PRIORITY	VIP1 priority setting 0x0 = highest priority 0x7 = lowest priority	RW	0x4
7:0	RESERVED		R	0x44

Table 13-113. Register Call Summary for Register CTRL_CORE_EMIF_INITIATOR_PRIORITY_4

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-114. CTRL_CORE_EMIF_INITIATOR_PRIORITY_5

Address Offset	0x0000 0430		
Physical Address	0x4A00 2430	**Instance**	CTRL_MODULE_CORE
Description	Register for priority settings for EMIF arbitration		
Type	RW		



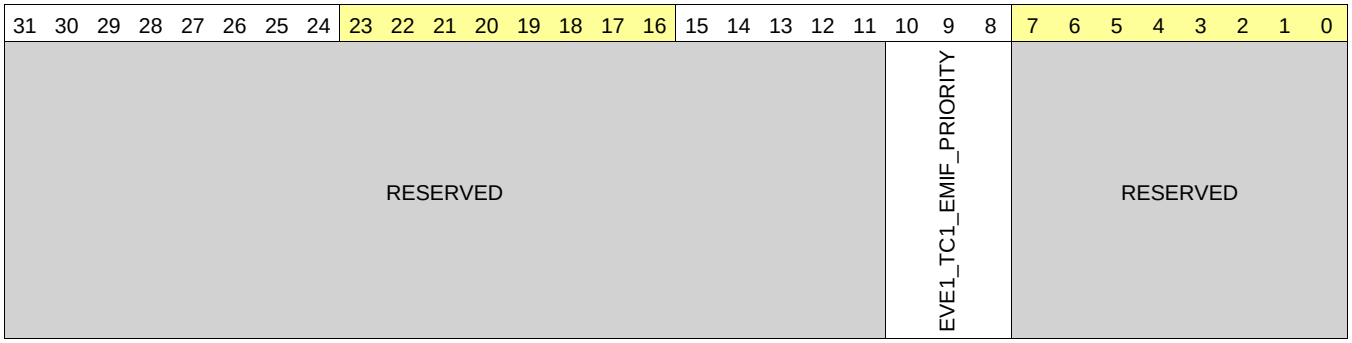
Bits	Field Name	Description	Type	Reset
31:15	RESERVED		R	0x8888
14:12	GMAC_SW_EMIF_PRIORITY	GMAC_SW priority setting 0x0 = highest priority 0x7 = lowest priority	RW	0x4
11:0	RESERVED		R	0x444

Table 13-115. Register Call Summary for Register CTRL_CORE_EMIF_INITIATOR_PRIORITY_5

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-116. CTRL_CORE_EMIF_INITIATOR_PRIORITY_6

Address Offset	0x0000 0434		
Physical Address	0x4A00 2434	**Instance**	CTRL_MODULE_CORE
Description	Register for priority settings for EMIF arbitration		
Type	RW		



Bits	Field Name	Description	Type	Reset
31:11	RESERVED		R	0x88888
10:8	EVE1_TC1_EMIF_PRIORITY	EVE1 TC1 priority setting 0x0 = highest priority 0x7 = lowest priority	RW	0x4
7:0	RESERVED		R	0x44

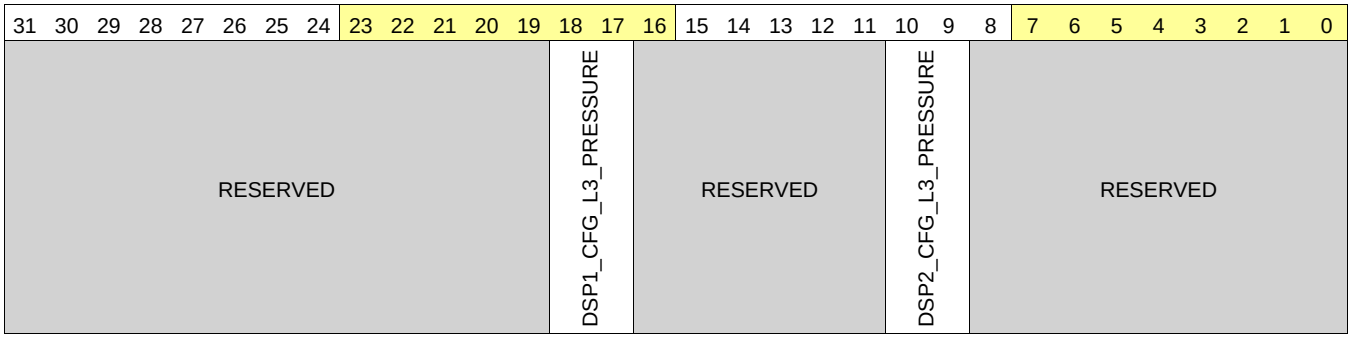
Table 13-117. Register Call Summary for Register CTRL_CORE_EMIF_INITIATOR_PRIORITY_6

Control Module Register Manual

- CTRL_MODULE_CORE Register Summary: [0]

Table 13-118. CTRL_CORE_L3_INITIATOR_PRESSURE_1

Address Offset	0x0000 043C		
Physical Address	0x4A00 243C	**Instance**	CTRL_MODULE_CORE
Description	Register for pressure settings for L3 arbitration		
Type	RW		



Bits	Field Name	Description	Type	Reset
31:19	RESERVED		R	0x0
18:17	DSP1_CFG_L3_PRESSURE	DSP1 CFG pressure setting 0x0 = lowest 0x3 = highest	RW	0x0

Bits	Field Name	Description	Type	Reset
16:11	RESERVED		R	0x0
10:9	DSP2_CFG_L3_PRESSURE	DSP2 CFG pressure setting 0x0 = lowest 0x3 = highest	RW	0x0
8:0	RESERVED		R	0x0

Table 13-119. Register Call Summary for Register CTRL_CORE_L3_INITIATOR_PRESSURE_1

Control Module Functional Description
- L3_MAIN Initiator Priority Registers: [0]

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [1]

Table 13-120. CTRL_CORE_CUST_FUSE_UID_0

Address Offset	0x0000 04E8		
Physical Address	0x4A00 24E8	**Instance**	CTRL_MODULE_CORE
Description	Customer Fuse keys. UID [031:000] (16 bits upper Redundant field) [FIELD OVERFLOW]. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CUST_FUSE_UID_0																															

Bits	Field Name	Description	Type	Reset
31:0	CUST_FUSE_UID_0		R	0x0

Table 13-121. Register Call Summary for Register CTRL_CORE_CUST_FUSE_UID_0

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-122. CTRL_CORE_CUST_FUSE_UID_1

Address Offset	0x0000 04EC		
Physical Address	0x4A00 24EC	**Instance**	CTRL_MODULE_CORE
Description	Customer Fuse keys. UID [063:032] (16 bits upper Redundant field) [FIELD F]. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CUST_FUSE_UID_1																															

Bits	Field Name	Description	Type	Reset
31:0	CUST_FUSE_UID_1		R	0x0

Table 13-123. Register Call Summary for Register CTRL_CORE_CUST_FUSE_UID_1

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-124. CTRL_CORE_CUST_FUSE_UID_2

Address Offset	0x0000 04F0		
Physical Address	0x4A00 24F0	**Instance**	CTRL_MODULE_CORE
Description	Customer Fuse keys. UID [095:064] (16 bits upper Redundant field) [FIELD E]. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CUST_FUSE_UID_2																															

Bits	Field Name	Description	Type	Reset
31:0	CUST_FUSE_UID_2		R	0x0

Table 13-125. Register Call Summary for Register CTRL_CORE_CUST_FUSE_UID_2

Control Module Register Manual

- CTRL_MODULE_CORE Register Summary: [0]

Table 13-126. CTRL_CORE_CUST_FUSE_UID_3

Address Offset	0x0000 04F4		
Physical Address	0x4A00 24F4	**Instance**	CTRL_MODULE_CORE
Description	Customer Fuse keys. UID [127:096] (16 bits upper Redundant field) [FIELD D]. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CUST_FUSE_UID_3																															

Bits	Field Name	Description	Type	Reset
31:0	CUST_FUSE_UID_3		R	0x0

Table 13-127. Register Call Summary for Register CTRL_CORE_CUST_FUSE_UID_3

Control Module Register Manual

- CTRL_MODULE_CORE Register Summary: [0]

Table 13-128. CTRL_CORE_CUST_FUSE_UID_4

Address Offset	0x0000 04F8		
Physical Address	0x4A00 24F8	**Instance**	CTRL_MODULE_CORE
Description	Customer Fuse keys. UID [159:127] (16 bits upper Redundant field) [FIELD C]. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CUST_FUSE_UID_4																															

Bits	Field Name	Description	Type	Reset
31:0	CUST_FUSE_UID_4		R	0x0

Table 13-129. Register Call Summary for Register CTRL_CORE_CUST_FUSE_UID_4

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-130. CTRL_CORE_CUST_FUSE_UID_5

Address Offset	0x0000 04FC		
Physical Address	0x4A00 24FC	**Instance**	CTRL_MODULE_CORE
Description	Customer Fuse keys. UID [191:160] (16 bits upper Redundant field) [FIELD B]. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CUST_FUSE_UID_5																															

Bits	Field Name	Description	Type	Reset
31:0	CUST_FUSE_UID_5		R	0x0

Table 13-131. Register Call Summary for Register CTRL_CORE_CUST_FUSE_UID_5

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-132. CTRL_CORE_CUST_FUSE_UID_6

Address Offset	0x0000 0500		
Physical Address	0x4A00 2500	**Instance**	CTRL_MODULE_CORE
Description	Customer Fuse keys. UID [223:192] (16 bits upper Redundant field) [FIELD A]. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CUST_FUSE_UID_6																															

Bits	Field Name	Description	Type	Reset
31:0	CUST_FUSE_UID_6		R	0x0

Table 13-133. Register Call Summary for Register CTRL_CORE_CUST_FUSE_UID_6

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-134. CTRL_CORE_MAC_ID_SW_0

Address Offset	0x0000 0514		
Physical Address	0x4A00 2514	**Instance**	CTRL_MODULE_CORE
Description	Standard Fuse keys, MAC ID_1 [63:32].		

Table 13-134. CTRL_CORE_MAC_ID_SW_0 (continued)

Type	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								STD_FUSE_MAC_ID_SW_0																									

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:0	STD_FUSE_MAC_ID_SW_0	This bit field contains the last three octets (NIC specific) of the MAC address of the GMAC_SW port 0. Bits [23:16] contain the fourth octet of the MAC address. Bits [15:8] contain the fifth octet of the MAC address. Bits [7:0] contain the last (sixth) octet of the MAC address.	R	0x0

Table 13-135. Register Call Summary for Register CTRL_CORE_MAC_ID_SW_0

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-136. CTRL_CORE_MAC_ID_SW_1

Address Offset	0x0000 0518		
Physical Address	0x4A00 2518	**Instance**	CTRL_MODULE_CORE
Description	Standard Fuse keys, MAC ID_1 [31:0].		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								STD_FUSE_MAC_ID_SW_1																									

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:0	STD_FUSE_MAC_ID_SW_1	This bit field contains the last three octets (the OUI) of the MAC address of the GMAC_SW port 0. Bits [23:16] contain the first octet of the MAC address. Bits [15:8] contain the second octet of the MAC address. Bits [7:0] contain the third octet of the MAC address.	R	0x0

Table 13-137. Register Call Summary for Register CTRL_CORE_MAC_ID_SW_1

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-138. CTRL_CORE_MAC_ID_SW_2

Address Offset	0x0000 051C		
Physical Address	0x4A00 251C	**Instance**	CTRL_MODULE_CORE
Description	Standard Fuse keys, MAC ID_2 [63:32].		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								STD_FUSE_MAC_ID_SW_2																									

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:0	STD_FUSE_MAC_ID_SW_2	This bit field contains the last three octets (NIC specific) of the MAC address of the GMAC_SW port 1. Bits [23:16] contain the fourth octet of the MAC address. Bits [15:8] contain the fifth octet of the MAC address. Bits [7:0] contain the last (sixth) octet of the MAC address.	R	0x0

Table 13-139. Register Call Summary for Register CTRL_CORE_MAC_ID_SW_2

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-140. CTRL_CORE_MAC_ID_SW_3

Address Offset	0x0000 0520		
Physical Address	0x4A00 2520	**Instance**	CTRL_MODULE_CORE
Description	Standard Fuse keys, MAC ID_2 [31:0].		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								STD_FUSE_MAC_ID_SW_3																									

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:0	STD_FUSE_MAC_ID_SW_3	This bit field contains the last three octets (the OUI) of the MAC address of the GMAC_SW port 1. Bits [23:16] contain the first octet of the MAC address. Bits [15:8] contain the second octet of the MAC address. Bits [7:0] contain the third octet of the MAC address.	R	0x0

Table 13-141. Register Call Summary for Register CTRL_CORE_MAC_ID_SW_3

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-142. CTRL_CORE_SMA_SW_1

Address Offset	0x0000 0534		
Physical Address	0x4A00 2534	**Instance**	CTRL_MODULE_CORE
Description	OCP Spare Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED					RGMII2_ID_MODE_N	RGMII1_ID_MODE_N	RESERVED	DSS_CH0_ON_OFF	RESERVED		DSS_CH0_IPC	RESERVED		DSS_CH0_RF	RESERVED													VIP1_CLK_INV_PORT_2B	VIP1_CLK_INV_PORT_1B	VIP1_CLK_INV_PORT_2A	VIP1_CLK_INV_PORT_1A

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26	RGMII2_ID_MODE_N	Ethernet RGMII port 2 internal delay on transmit 0x0: Internal delay enabled 0x1: Internal delay disabled	RW	0x0
25	RGMII1_ID_MODE_N	Ethernet RGMII port 1 internal delay on transmit 0x0: Internal delay enabled 0x1: Internal delay disabled	RW	0x0
24:23	RESERVED		R	0x0
22	DSS_CH0_ON_OFF	DSS Channel 0 Pixel clock control On/Off 0x0: HSYNC and VSYNC are driven on opposite edges of pixel clock than pixel data 0x1: HSYNC and VSYNC are driven according to bit DSS_CH0_RF	RW	0x0
21:20	RESERVED		R	0x0
19	DSS_CH0_IPC	DSS Channel 0 IPC control 0x0: Data is driven on the LCD data lines on the rising edge of the pixel clock 0x1: Data is driven on the LCD data lines on the falling edge of the pixel clock	RW	0x0
18:17	RESERVED		R	0x0
16	DSS_CH0_RF	DSS Channel 0 Rise/Fall control 0x0: HSYNC and VSYNC are driven on falling edge of pixel clock (if bit DSS_CH0_ON_OFF set to 1) 0x1: HSYNC and VSYNC are driven on rising edge of pixel clock (if bit DSS_CH0_ON_OFF set to 1)	RW	0x0
15:4	RESERVED		R	0x0
3	VIP1_CLK_INV_PORT_2B	VIP1 Slice 1 Clock inversion for Port B enable 0x0: Clock inversion is disabled 0x1: Clock inversion is enabled	RW	0x0
2	VIP1_CLK_INV_PORT_1B	VIP1 Slice 0 Clock inversion for Port B enable 0x0: Clock inversion is disabled 0x1: Clock inversion is enabled	RW	0x0
1	VIP1_CLK_INV_PORT_2A	VIP1 Slice 1 Clock inversion for Port A enable 0x0: Clock inversion is disabled 0x1: Clock inversion is enabled	RW	0x0
0	VIP1_CLK_INV_PORT_1A	VIP1 Slice 0 Clock inversion for Port A enable 0x0: Clock inversion is disabled 0x1: Clock inversion is enabled	RW	0x0

Table 13-143. Register Call Summary for Register CTRL_CORE_SMA_SW_1

Control Module Register Manual

- CTRL_MODULE_CORE Register Summary: [0]

Table 13-144. CTRL_CORE_EMIF_INITIATOR_PRIORITY_8

Address Offset	0x0000 053C		
Physical Address	0x4A00 253C	**Instance**	CTRL_MODULE_CORE
Description	Register for priority settings for EMIF arbitration		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																ISS_NRT_EMIF_PRIORITY				RESERVED		ISS_RT_EMIF_PRIORITY									
RESERVED																								RESERVED							

Bits	Field Name	Description	Type	Reset
31:15	RESERVED		R	0x0
14:12	ISS_NRT_EMIF_PRIORITY	ISS NRT ports priority setting. Note: ISS is not supported on the DRA78x family of devices. 0x0: highest priority 0x7: lowest priority	RW	0x4
11	RESERVED		R	0x0
10:8	ISS_RT_EMIF_PRIORITY	ISS RT port priority setting. 0x0: highest priority 0x7: lowest priority	RW	0x4
7:0	RESERVED		R	0x44

Table 13-145. Register Call Summary for Register CTRL_CORE_EMIF_INITIATOR_PRIORITY_8

Control Module Functional Description
- SDRAM Initiator Priority Registers: [0]

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [1]

Table 13-146. CTRL_CORE_MMR_LOCK_1

Address Offset	0x0000 0540		
Physical Address	0x4A00 2540	**Instance**	CTRL_MODULE_CORE
Description	Register to lock memory regions from 0x0000 0100 to 0x0000 079F and from 0x0000 1A00 to 0x0000 1FFF		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MMR_LOCK_1																															

Bits	Field Name	Description	Type	Reset
31:0	MMR_LOCK_1	Lock value for regions 0x0000 0100 to 0x0000 079F and 0x0000 1A00 to 0x0000 1FFF 0x1A1C8144 = lock value 0x2FF1AC2B = unlock value	RW	0x1A1C8144

Table 13-147. Register Call Summary for Register CTRL_CORE_MMR_LOCK_1

Control Module Functional Description
- Memory Region Lock Registers: [0]

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [1]

Table 13-148. CTRL_CORE_MMR_LOCK_2

Address Offset	0x0000 0544		
Physical Address	0x4A00 2544	**Instance**	CTRL_MODULE_CORE
Description	Register to lock memory region starting at address offset 0x0000 07A0 and ending at address offset 0x0000 0D9F		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MMR_LOCK_2																															

Bits	Field Name	Description	Type	Reset
31:0	MMR_LOCK_2	Lock value for region 0x0000 07A0 to 0x0000 0D9F 0xFDF45530 = lock value 0xF757FDC0 = unlock value	RW	0xFDF45530

Table 13-149. Register Call Summary for Register CTRL_CORE_MMR_LOCK_2

Control Module Functional Description
- Memory Region Lock Registers: [0]

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [1]

Table 13-150. CTRL_CORE_MMR_LOCK_3

Address Offset	0x0000 0548		
Physical Address	0x4A00 2548	**Instance**	CTRL_MODULE_CORE
Description	Register to lock memory region starting at address offset 0x0000 0DA0 and ending at address offset 0x0000 0FFF		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MMR_LOCK_3																															

Bits	Field Name	Description	Type	Reset
31:0	MMR_LOCK_3	Lock value for region 0x0000 0DA0 to 0x0000 0FFF 0x1AE6E320 = lock value 0xE2BC3A6D = unlock value	RW	0x1AE6E320

Table 13-151. Register Call Summary for Register CTRL_CORE_MMR_LOCK_3

Control Module Functional Description
- Memory Region Lock Registers: [0]

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [1]

Table 13-152. CTRL_CORE_MMR_LOCK_4

Address Offset	0x0000 054C		
Physical Address	0x4A00 254C	**Instance**	CTRL_MODULE_CORE
Description	Register to lock memory region starting at address offset 0x0000 1000 and ending at address offset 0x0000 13FF		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MMR_LOCK_4																															

Bits	Field Name	Description	Type	Reset
31:0	MMR_LOCK_4	Lock value for region 0x0000 1000 to 0x0000 13FF 0x2FFA927C = lock value 0x1EBF131D = unlock value	RW	0x2FFA927C

Table 13-153. Register Call Summary for Register CTRL_CORE_MMR_LOCK_4

Control Module Functional Description
- Memory Region Lock Registers: [0]

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [1]

Table 13-154. CTRL_CORE_MMR_LOCK_5

Address Offset	0x0000 0550		
Physical Address	0x4A00 2550	**Instance**	CTRL_MODULE_CORE
Description	Register to lock memory region starting at address offset 0x0000 1400 and ending at address offset 0x0000 19FF		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MMR_LOCK_5																															

Bits	Field Name	Description	Type	Reset
31:0	MMR_LOCK_5	Lock value for region 0x0000 1400 to 0x0000 19FF 0x143F832C = lock value 0x6F361E05 = unlock value	RW	0x143F832C

Table 13-155. Register Call Summary for Register CTRL_CORE_MMR_LOCK_5

Control Module Functional Description
- Memory Region Lock Registers: [0]

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [1]

Table 13-156. CTRL_CORE_CONTROL_IO_1

Address Offset	0x0000 0554		
Physical Address	0x4A00 2554	**Instance**	CTRL_MODULE_CORE
Description	Register to configure some IP level signals		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED															MMU1_DISABLE	RESERVED		TC1_DEFAULT_BURST_SIZE		RESERVED		TC0_DEFAULT_BURST_SIZE		RESERVED		GMII2_SEL		RESERVED		GMII1_SEL	

Bits	Field Name	Description	Type	Reset
31:17	RESERVED		R	0x0
16	MMU1_DISABLE	MMU1 DISABLE setting	RW	0x0
15:14	RESERVED		R	0x0
13:12	TC1_DEFAULT_BURST_SIZE	EDMA TC1 Default Burst Size (DBS) setting 0x0: 16 byte burst 0x1: 32 byte burst 0x2: 64 byte burst 0x3: 128 byte burst	RW	0x3
11:10	RESERVED		R	0x0
9:8	TC0_DEFAULT_BURST_SIZE	EDMA TC0 Default Burst Size (DBS) setting 0x0: 16 byte burst 0x1: 32 byte burst 0x2: 64 byte burst 0x3: 128 byte burst	RW	0x3
7:6	RESERVED		R	0x0
5:4	GMII2_SEL	GMII2 selection setting 0x0: Reserved 0x1: Reserved 0x2: RGMII 0x3: Reserved	RW	0x0
3:2	RESERVED		R	0x0
1:0	GMII1_SEL	GMII1 selection setting 0x0: Reserved 0x1: Reserved 0x2: RGMII 0x3: Reserved	RW	0x0

Table 13-157. Register Call Summary for Register CTRL_CORE_CONTROL_IO_1

Control Module Functional Description
- Settings Related To Different Peripheral Modules: [0]

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [1]

Table 13-158. CTRL_CORE_CONTROL_IO_2

Address Offset	0x0000 0558		
Physical Address	0x4A00 2558	**Instance**	CTRL_MODULE_CORE
Description	Register to configure some IP level signals		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED					RTI_RESET_SELECTION_TO_PRCM			GMAC_RESET_ISOLATION_ENABLE	RESERVED		PWMSS1_TBCLKEN	RESERVED				VIP1_VIN2_INPUT_SELECTION	VIP1_VIN1_INPUT_SELECTION	RESERVED			QSPI_MEMMAPPED_CS		RESERVED		DCAN2_RAMINIT_START	DSS_DESHDCP_DISABLE	DCAN_RAMINIT_START	DCAN2_RAMINIT_DONE	DCAN_RAMINIT_DONE	DSS_DESHDCP_CLKEN

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26:24	RTI_RESET_SELECTION_TO_PRCM	RTI reset mux select value. 0x0: RTI1 0x1: RTI2 0x2: RTI3 0x3: RTI4 0x4: RTI5	RW	0x0
23	GMAC_RESET_ISOLATION_ENABLE	Reset isolation enable setting 0x0 = Reset is not isolated 0x1 = Reset is isolated	RW	0x0
22:21	RESERVED		R	0x0
20	PWMSS1_TBCLKEN	PWMSS1 CLOCK ENABLE setting	RW	0x0
19:16	RESERVED		R	0x0
15	VIP1_VIN2_INPUT_SELECTION	VIP1 vin2 input selection setting. 0x0: Input from pins selected. 0x1: Input from LVDSRX port2 selected.	RW	0x0
14	VIP1_VIN1_INPUT_SELECTION	VIP1 vin1 input selection setting. 0x0: Input from pins selected. 0x1: Input from LVDSRX port1 selected.	RW	0x0
13:11	RESERVED		R	0x0
10:8	QSPI_MEMMAPPED_CS	QSPI CS MAPPING setting. 0x0: The QSPI configuration registers are accessed 0x1: An external device connected to CS0 is accessed 0x2: An external device connected to CS1 is accessed 0x3: An external device connected to CS2 is accessed 0x4-0x7: An external device connected to CS3 is accessed	RW	0x0
7:6	RESERVED		R	0x0
5	DCAN2_RAMINIT_START	Reserved on this device.	RW	0x0
4	DSS_DESHDCP_DISABLE	DSS DESHDCP DISABLE setting	RW	0x0

Bits	Field Name	Description	Type	Reset
3	DCAN_RAMINIT_START	DCAN RAM INIT START setting To initialize DCAN RAM, the bit should be set to 0x1. It is not auto cleared by hardware. **Note**: If DCAN RAMINIT sequence needs to be redone, this bit should be first cleared and then set again.	RW	0x0
2	DCAN2_RAMINIT_DONE	Reserved on this device.	RW	0x0
1	DCAN_RAMINIT_DONE	DCAN RAM INIT DONE status	RW	0x0
0	DSS_DESHDCP_CLKEN	DSS DESHDCP CLOCK ENABLE setting	RW	0x0

Table 13-159. Register Call Summary for Register CTRL_CORE_CONTROL_IO_2

Control Module Functional Description
- Settings Related To Different Peripheral Modules: [0]

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [1]

Table 13-160. CTRL_CORE_CONTROL_DSP1_RST_VECT

Address Offset	0x0000 055C		
Physical Address	0x4A00 255C	**Instance**	CTRL_MODULE_CORE
Description	Register for storing DSP1 reset vector		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED					DSP1_NUM_MM			RESERVED		DSP1_RST_VECT																					

Bits	Field Name	Description	Type	Reset
31:27	RESERVED	Reserved	RW	0x0
26:24	DSP1_NUM_MM	Number of DSP instances in the SoC 0x1 = 1 0x2 = 2	RW	0x0
23:22	RESERVED		R	0x0
21:0	DSP1_RST_VECT	DSP1 reset vector address	RW	0x0

Table 13-161. Register Call Summary for Register CTRL_CORE_CONTROL_DSP1_RST_VECT

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-162. CTRL_CORE_CONTROL_DSP2_RST_VECT

Address Offset	0x0000 0560		
Physical Address	0x4A00 2560	**Instance**	CTRL_MODULE_CORE
Description	Register for storing DSP2 reset vector		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED					DSP2_NUM_MM			RESERVED								DSP2_RST_VECT																

Bits	Field Name	Description	Type	Reset
31:27	RESERVED	Reserved	RW	0x0
26:24	DSP2_NUM_MM	Number of DSP instances in the SoC 0x1 = 1 0x2 = 2	RW	0x0
23:22	RESERVED		R	0x0
21:0	DSP2_RST_VECT	DSP2 reset vector address	RW	0x0

Table 13-163. Register Call Summary for Register CTRL_CORE_CONTROL_DSP2_RST_VECT

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-164. CTRL_CORE_STD_FUSE_OPP_BGAP_DSPEVE

Address Offset	0x0000 0564		
Physical Address	0x4A00 2564	**Instance**	CTRL_MODULE_CORE
Description	Trim values for DSPEVE associated bandgap. Contains TI Internal information, not intended for application use.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																STD_FUSE_OPP_BGAP_DSPEVE_0								STD_FUSE_OPP_BGAP_DSPEVE_1							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x-
15:8	STD_FUSE_OPP_BGAP_DSPEVE_0	Trim values for DSPEVE associated bandgap. Contains TI Internal information, not intended for application use.	R	0x-

Bits	Field Name	Description	Type	Reset
7:0	STD_FUSE_OPP_BGAP_DSPEVE_1	Trim values for DSPEVE associated bandgap. Contains TI Internal information, not intended for application use.	R	0x-

Table 13-165. Register Call Summary for Register CTRL_CORE_STD_FUSE_OPP_BGAP_DSPEVE

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-166. CTRL_CORE_LDOSRAM_DSPEVE_VOLTAGE_CTRL

Address Offset	0x0000 056C		
Physical Address	0x4A00 256C	**Instance**	CTRL_MODULE_CORE
Description	DSPEVE SRAM LDO Control register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED					LDOSRAMDSPEVE_RETMODE_MUX_CTRL	LDOSRAMDSPEVE_RETMODE_VSET_IN					LDOSRAMDSPEVE_RETMODE_VSET_OUT					RESERVED					LDOSRAMDSPEVE_ACTMODE_MUX_CTRL	LDOSRAMDSPEVE_ACTMODE_VSET_IN					LDOSRAMDSPEVE_ACTMODE_VSET_OUT					

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26	LDOSRAMDSPEVE_RETMODE_MUX_CTRL	Override control of EFUSE Retention Mode Voltage value 0x0: eFuse value is used 0x1: Override value is used	RW	0x0
25:21	LDOSRAMDSPEVE_RETMODE_VSET_IN	EFUSE Retention Mode Voltage value (vset[9:5])	R	0x0
20:16	LDOSRAMDSPEVE_RETMODE_VSET_OUT	Override value for Retention Mode Voltage	RW	0x0
15:11	RESERVED		R	0x0
10	LDOSRAMDSPEVE_ACTMODE_MUX_CTRL	Override control of EFUSE Active Mode Voltage value 0x0: eFuse value is used 0x1: Override value is used	RW	0x0
9:5	LDOSRAMDSPEVE_ACTMODE_VSET_IN	EFUSE Active Mode Voltage value (vset[4:0])	R	0x0
4:0	LDOSRAMDSPEVE_ACTMODE_VSET_OUT	Override value for Active Mode Voltage value	RW	0x0

Table 13-167. Register Call Summary for Register CTRL_CORE_LDOSRAM_DSPEVE_VOLTAGE_CTRL

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-168. CTRL_CORE_STD_FUSE_OPP_VMIN_DSPEVE_5

Address Offset	0x0000 05D8		
Physical Address	0x4A00 25D8	**Instance**	CTRL_MODULE_CORE
Description	This register contains the AVS Class 0 voltage value for the vdd_dspeve voltage rail when running at OPP_PLUS. This register also stores information about ABB configuration for that OPP.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																				STD_FUSE_OPP_VMIN_DSPEVE_5															

Bits	Field Name	Description	Type	Reset
31:12	RESERVED	Reserved	R	0x-
11:0	STD_FUSE_OPP_VMIN_DSPEVE_5	AVS Class 0 voltage value for the vdd_dspeve voltage rail when running at OPP_PLUS. To get the actual value in mV, the value read from this bit field must be converted to decimal value.	R	0x-

Table 13-169. Register Call Summary for Register CTRL_CORE_STD_FUSE_OPP_VMIN_DSPEVE_5

Control Module Functional Description
- AVS Class 0 Associated Registers: [0]

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [1]

Table 13-170. CTRL_CORE_STD_FUSE_OPP_VMIN_DSPEVE_2

Address Offset	0x0000 05E0		
Physical Address	0x4A00 25E0	**Instance**	CTRL_MODULE_CORE
Description	This register contains the AVS Class 0 voltage value for the vdd_dspeve voltage rail when running at OPP_NOM.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																				STD_FUSE_OPP_VMIN_DSPEVE_2															

Bits	Field Name	Description	Type	Reset
31:12	RESERVED		R	0x-
11:0	STD_FUSE_OPP_VMIN_DSPEVE_2	AVS Class 0 voltage value for the vdd_dspeve voltage rail when running at OPP_NOM. To get the actual value in mV, the value read from this bit field must be converted to decimal value.	R	0x-

Table 13-171. Register Call Summary for Register CTRL_CORE_STD_FUSE_OPP_VMIN_DSPEVE_2

Control Module Functional Description
- AVS Class 0 Associated Registers: [0]

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [1]

Table 13-172. CTRL_CORE_STD_FUSE_OPP_VMIN_DSPEVE_3

Address offset	0x0000 05E4		
Physical Address	0x4A00 25E4	**Instance**	CTRL_MODULE_CORE
Description	This register contains the AVS Class 0 voltage value for the vdd_dspeve voltage rail when running at OPP_OD.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																				STD_FUSE_OPP_VMIN_DSPEVE_3											

Bits	Field Name	Description	Type	Reset
31:12	RESERVED	Reserved	R	0x-
11:0	STD_FUSE_OPP_VMIN_DSPEVE_3	AVS Class 0 voltage value for the vdd_dspeve voltage rail when running at OPP_OD. To get the actual value in mV, the value read from this bit field must be converted to decimal value.	R	0x-

Table 13-173. Register Call Summary for Register CTRL_CORE_STD_FUSE_OPP_VMIN_DSPEVE_3

Control Module Functional Description
- AVS Class 0 Associated Registers: [0]

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [1]

Table 13-174. CTRL_CORE_STD_FUSE_OPP_VMIN_DSPEVE_4

Address offset	0x0000 05E8		
Physical Address	0x4A00 25E8	**Instance**	CTRL_MODULE_CORE
Description	This register contains the AVS Class 0 voltage value for the vdd_dspeve voltage rail when running at OPP_HIGH.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																				STD_FUSE_OPP_VMIN_DSPEVE_4											

Bits	Field Name	Description	Type	Reset
31:12	RESERVED	Reserved	R	0x-
11:0	STD_FUSE_OPP_VMIN_DSPEVE_4	AVS Class 0 voltage value for the vdd_dspeve voltage rail when running at OPP_HIGH. To get the actual value in mV, the value read from this bit field must be converted to decimal value.	R	0x-

Table 13-175. Register Call Summary for Register CTRL_CORE_STD_FUSE_OPP_VMIN_DSPEVE_4

Control Module Functional Description
- AVS Class 0 Associated Registers: [0]

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [1]

Table 13-176. CTRL_CORE_RC_OSC_FREQUENCY

Address offset	0x0000 05EC		
Physical Address	0x4A00 25EC	**Instance**	CTRL_MODULE_CORE
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								RC_OSC_CALIBRATION_VALUE																									

Bits	Field Name	Description	Type	Reset
31:26	RESERVED	Reserved	R	0x-
25:0	RC_OSC_CALIBRATION_VALUE		R	0x-

Table 13-177. Register Call Summary for Register CTRL_CORE_RC_OSC_FREQUENCY

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-178. CTRL_CORE_STD_FUSE_OPP_VMIN_CORE_2

Address Offset	0x0000 05F4		
Physical Address	0x4A00 25F4	**Instance**	CTRL_MODULE_CORE
Description	This register contains the AVS Class 0 voltage value for the vdd voltage rail when running at OPP_NOM.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																				STD_FUSE_OPP_VMIN_CORE_2											

Bits	Field Name	Description	Type	Reset
31:12	RESERVED		R	0x-
11:0	STD_FUSE_OPP_VMIN_CORE_2	AVS Class 0 voltage value for the vdd voltage rail when running at OPP_NOM. To get the actual value in mV, the value read from this bit field must be converted to decimal value.	R	0x-

Table 13-179. Register Call Summary for Register CTRL_CORE_STD_FUSE_OPP_VMIN_CORE_2

Control Module Functional Description
- AVS Class 0 Associated Registers: [0][1]

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [2]

Table 13-180. CTRL_CORE_LDOSRAM_CORE_2_VOLTAGE_CTRL

Address Offset	0x0000 0680		
Physical Address	0x4A00 2680	**Instance**	CTRL_MODULE_CORE
Description	CORE 2nd SRAM LDO Control register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED					LDOSRAMCORE_2_RETMODE_MUX_CTRL	LDOSRAMCORE_2_RETMODE_VSET_IN					LDOSRAMCORE_2_RETMODE_VSET_OUT					RESERVED					LDOSRAMCORE_2_ACTMODE_MUX_CTRL	LDOSRAMCORE_2_ACTMODE_VSET_IN					LDOSRAMCORE_2_ACTMODE_VSET_OUT			

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26	LDOSRAMCORE_2_RETMODE_MUX_CTRL	Override control of EFUSE Retention Mode Voltage value 0x0: eFuse value is used 0x1: Override value is used	RW	0x0
25:21	LDOSRAMCORE_2_RETMODE_VSET_IN	EFUSE Retention Mode Voltage value (vset[9:5])	R	0x0
20:16	LDOSRAMCORE_2_RETMODE_VSET_OUT	Override value for Retention Mode Voltage	RW	0x0
15:11	RESERVED		R	0x0
10	LDOSRAMCORE_2_ACTMODE_MUX_CTRL	Override control of EFUSE Active Mode Voltage value 0x0: eFuse value is used 0x1: Override value is used	RW	0x0
9:5	LDOSRAMCORE_2_ACTMODE_VSET_IN	EFUSE Active Mode Voltage value (vset[4:0])	R	0x0
4:0	LDOSRAMCORE_2_ACTMODE_VSET_OUT	Override value for Active Mode Voltage value	RW	0x0

Table 13-181. Register Call Summary for Register CTRL_CORE_LDOSRAM_CORE_2_VOLTAGE_CTRL

Control Module Register Manual

- CTRL_MODULE_CORE Register Summary: [0]

Table 13-182. CTRL_CORE_LDOSRAM_CORE_3_VOLTAGE_CTRL

Address Offset	0x0000 0684		
Physical Address	0x4A00 2684	**Instance**	CTRL_MODULE_CORE
Description	CORE 3rd SRAM LDO Control register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED					LDOSRAMCORE_3_RETMODE_MUX_CTRL	LDOSRAMCORE_3_RETMODE_VSET_IN					LDOSRAMCORE_3_RETMODE_VSET_OUT					RESERVED					LDOSRAMCORE_3_ACTMODE_MUX_CTRL	LDOSRAMCORE_3_ACTMODE_VSET_IN					LDOSRAMCORE_3_ACTMODE_VSET_OUT			

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26	LDOSRAMCORE_3_RETMODE_MUX_CTRL	Override control of EFUSE Retention Mode Voltage value 0x0: eFuse value is used 0x1: Override value is used	RW	0x0
25:21	LDOSRAMCORE_3_RETMODE_VSET_IN	EFUSE Retention Mode Voltage value (vset[9:5])	R	0x0
20:16	LDOSRAMCORE_3_RETMODE_VSET_OUT	Override value for Retention Mode Voltage	RW	0x0
15:11	RESERVED		R	0x0
10	LDOSRAMCORE_3_ACTMODE_MUX_CTRL	Override control of EFUSE Active Mode Voltage value 0x0: eFuse value is used 0x1: Override value is used	RW	0x0
9:5	LDOSRAMCORE_3_ACTMODE_VSET_IN	EFUSE Active Mode Voltage value (vset[4:0])	R	0x0
4:0	LDOSRAMCORE_3_ACTMODE_VSET_OUT	Override value for Active Mode Voltage value	RW	0x0

Table 13-183. Register Call Summary for Register CTRL_CORE_LDOSRAM_CORE_3_VOLTAGE_CTRL

Control Module Register Manual

- CTRL_MODULE_CORE Register Summary: [0]

Table 13-184. CTRL_CORE_NMI_DESTINATION_1

Address Offset	0x0000 068C		
Physical Address	0x4A00 268C	**Instance**	CTRL_MODULE_CORE
Description	Register for routing NMI interrupt to respective cores		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								IPU1_C1							

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Reserved	RW	0x0
7:0	IPU1_C1	Enable IPU1 CORE1 to receive the NMI interrupt 0x0 = NMI disabled 0x1 = NMI enabled	RW	0x0

Table 13-185. Register Call Summary for Register CTRL_CORE_NMI_DESTINATION_1

Control Module Functional Description
• NMI Mapping To Respective Cores: [0]
Control Module Register Manual
• CTRL_MODULE_CORE Register Summary: [1]

Table 13-186. CTRL_CORE_NMI_DESTINATION_2

Address Offset	0x0000 0690		
Physical Address	0x4A00 2690	**Instance**	CTRL_MODULE_CORE
Description	Register for routing NMI interrupt to respective cores		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IPU1_C0								DSP2								DSP1								RESERVED							

Bits	Field Name	Description	Type	Reset
31:24	IPU1_C0	Enable IPU1 CORE0 to receive the NMI interrupt 0x0 = NMI disabled 0x1 = NMI enabled	RW	0x0
23:16	DSP2	Enable DSP2 to receive the NMI interrupt 0x0 = NMI disabled 0x1 = NMI enabled	RW	0x0
15:8	DSP1	Enable DSP1 to receive the NMI interrupt 0x0 = NMI disabled 0x1 = NMI enabled	RW	0x0
7:0	RESERVED		R	0x0

Table 13-187. Register Call Summary for Register CTRL_CORE_NMI_DESTINATION_2

Control Module Functional Description
• NMI Mapping To Respective Cores: [0]
Control Module Register Manual
• CTRL_MODULE_CORE Register Summary: [1]

Table 13-188. CTRL_CORE_STD_FUSE_OPP_VDD_DSPEVE_0

Address Offset	0x0000 06A0		
Physical Address	0x4A00 26A0	**Instance**	CTRL_MODULE_CORE
Description	Standard Fuse OPP VDD_DSPEVE [31:0]. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STD_FUSE_OPP_VDD_DSPEVE_0																															

Bits	Field Name	Description	Type	Reset
31:0	STD_FUSE_OPP_VDD_DSPEVE_0		R	0x0

Table 13-189. Register Call Summary for Register CTRL_CORE_STD_FUSE_OPP_VDD_DSPEVE_0

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-190. CTRL_CORE_STD_FUSE_OPP_VDD_DSPEVE_1

Address Offset	0x0000 06A4		
Physical Address	0x4A00 26A4	**Instance**	CTRL_MODULE_CORE
Description	Standard Fuse OPP VDD_DSPEVE [63:32]. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STD_FUSE_OPP_VDD_DSPEVE_1																															

Bits	Field Name	Description	Type	Reset
31:0	STD_FUSE_OPP_VDD_DSPEVE_1		R	0x0

Table 13-191. Register Call Summary for Register CTRL_CORE_STD_FUSE_OPP_VDD_DSPEVE_1

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-192. CTRL_CORE_STD_FUSE_OPP_VDD_DSPEVE_2

Address Offset	0x0000 06A8		
Physical Address	0x4A00 26A8	**Instance**	CTRL_MODULE_CORE
Description	Standard Fuse OPP VDD_DSPEVE [95:64]. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STD_FUSE_OPP_VDD_DSPEVE_2																															

Bits	Field Name	Description	Type	Reset
31:0	STD_FUSE_OPP_VDD_DSPEVE_2		R	0x0

Table 13-193. Register Call Summary for Register CTRL_CORE_STD_FUSE_OPP_VDD_DSPEVE_2

Control Module Register Manual

- CTRL_MODULE_CORE Register Summary: [0]

Table 13-194. CTRL_CORE_STD_FUSE_OPP_VDD_DSPEVE_3

Address Offset	0x0000 06AC		
Physical Address	0x4A00 26AC	**Instance**	CTRL_MODULE_CORE
Description	Standard Fuse OPP VDD_DSPEVE [127:96]. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STD_FUSE_OPP_VDD_DSPEVE_3																															

Bits	Field Name	Description	Type	Reset
31:0	STD_FUSE_OPP_VDD_DSPEVE_3		R	0x0

Table 13-195. Register Call Summary for Register CTRL_CORE_STD_FUSE_OPP_VDD_DSPEVE_3

Control Module Register Manual

- CTRL_MODULE_CORE Register Summary: [0]

Table 13-196. CTRL_CORE_STD_FUSE_OPP_VDD_DSPEVE_4

Address Offset	0x0000 06B0		
Physical Address	0x4A00 26B0	**Instance**	CTRL_MODULE_CORE
Description	Standard Fuse OPP VDD_DSPEVE [159:128]. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STD_FUSE_OPP_VDD_DSPEVE_4																															

Bits	Field Name	Description	Type	Reset
31:0	STD_FUSE_OPP_VDD_DSPEVE_4		R	0x0

Table 13-197. Register Call Summary for Register CTRL_CORE_STD_FUSE_OPP_VDD_DSPEVE_4

Control Module Register Manual

- CTRL_MODULE_CORE Register Summary: [0]

Table 13-198. CTRL_CORE_CUST_FUSE_SWRV_7

Address Offset	0x0000 06B4
Physical Address	0x4A00 26B4 **Instance** CTRL_MODULE_CORE
Description	Customer Fuse keys. SWRV [31:0] (16 bits upper Redundant field) [FIELD A]. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.
Type	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CUST_FUSE_SWRV_7																															

Bits	Field Name	Description	Type	Reset
31:0	CUST_FUSE_SWRV_7		R	0x0

Table 13-199. Register Call Summary for Register CTRL_CORE_CUST_FUSE_SWRV_7

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-200. CTRL_CORE_STD_FUSE_CALIBRATION_OVERRIDE_VALUE_0

Address Offset	0x0000 06B8
Physical Address	0x4A00 26B8 **Instance** CTRL_MODULE_CORE
Description	Standard Fuse Calibration override value [31:0]. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.
Type	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STD_FUSE_CALIBRATION_OVERRIDE_VALUE_0																															

Bits	Field Name	Description	Type	Reset
31:0	STD_FUSE_CALIBRATIO N_OVERRIDE_VALUE_0		R	0x0

Table 13-201. Register Call Summary for Register CTRL_CORE_STD_FUSE_CALIBRATION_OVERRIDE_VALUE_0

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-202. CTRL_CORE_STD_FUSE_CALIBRATION_OVERRIDE_VALUE_1

Address Offset	0x0000 06BC
Physical Address	0x4A00 26BC **Instance** CTRL_MODULE_CORE
Description	Standard Fuse Calibration override value [63:32]. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.
Type	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STD_FUSE_CALIBRATION_OVERRIDE_VALUE_1																															

Bits	Field Name	Description	Type	Reset
31:0	STD_FUSE_CALIBRATION_OVERRIDE_VALUE_1		R	0x0

Table 13-203. Register Call Summary for Register CTRL_CORE_STD_FUSE_CALIBRATION_OVERRIDE_VALUE_1

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-204. CTRL_CORE_SEC_ERR_STATUS_FUNC_3

Address Offset	0x0000 0794		
Physical Address	0x4A00 2794	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															CRC_FW_ERROR

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	CRC_FW_ERROR	CRC firewall Note: CRC is not supported on the DRA78x family of devices. 0x0: No error from firewall 0x1: Error from firewall	RW W1toClr	0x0

Table 13-205. Register Call Summary for Register CTRL_CORE_SEC_ERR_STATUS_FUNC_3

Control Module Functional Description
- Firewall Error Status Registers: [0][1]

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [2]

Table 13-206. CTRL_CORE_SEC_ERR_STATUS_DEBUG_3

Address Offset	0x0000 079C		
Physical Address	0x4A00 279C	**Instance**	CTRL_MODULE_CORE
Description	Security Error Status functional Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															CRC_DBGFW_ERROR

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	CRC_DBGFW_ERROR	CRC debug firewall Note: CRC is not supported on the DRA78x family of devices. 0x0: No error from firewall 0x1: Error from firewall	RW W1toClr	0x0

Table 13-207. Register Call Summary for Register CTRL_CORE_SEC_ERR_STATUS_DEBUG_3

Control Module Functional Description • Firewall Error Status Registers: [0]
Control Module Register Manual • CTRL_MODULE_CORE Register Summary: [1]

Table 13-208. CTRL_CORE_BOOTSTRAP

Address Offset	0x0000 06C4		
Physical Address	0x4A00 26C4	**Instance**	CTRL_MODULE_CORE
Description	Register to view all the sysboot settings		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																DSP_CLOCK_DIVIDER	RESERVED	BOOTDEVICESIZE	MUXCS0DEVICE	BOOTWAITEN	SPEEDSELECT			RESERVED	HWOBS_IO_SELECTION	ADC_CLOCK_DIVIDER	BOOTMODE				

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0
15	DSP_CLOCK_DIVIDER	Divide factor for DSP clock 0x0: DSP_CLK2 is selected. Not supported on this SoC. 0x1: DSP_CLK3 is selected	R	0x0
14	RESERVED		R	0x0
13	BOOTDEVICESIZE	Select the size of the flash device on CS0. 0x0: 8-bit 0x1: 16-bit	R	0x0

Bits	Field Name	Description	Type	Reset
12:11	MUXCS0DEVICE	Select IC boot sequence to be executed from a multiplexed address and data device attached to CS0. 0x0: Non-muxed device attached 0x1: Addr-Data Mux device attached 0x2: Reserved 0x3: Reserved	R	0x0
10	BOOTWAITEN	Enable the monitoring on CS0 of the wait pin at IC reset release time for read accesses. 0x0: Wait pin is not monitored for read accesses 0x1: Wait pin is monitored for read accesses	R	0x0
9:8	SPEEDSELECT	Indicates the SYS_CLK1 frequency (from osc0). Note that the internal FUNC_32K_CLK is equal to SYS_CLK1/610, which is nominally 32.7869 kHz with 20 MHz clock. 0x0: Reserved 0x1: 20 MHz 0x2: 27 MHz 0x3: 19.2 MHz	R	0x0
7	RESERVED		R	0x0
6	HWOBS_IO_SELECTION		R	0x0
5	ADC_CLOCK_DIVIDER		R	0x0
4:0	BOOTMODE	SYSBOOT mode	R	0x0

Table 13-209. Register Call Summary for Register CTRL_CORE_BOOTSTRAP

Control Module Functional Description
- Control Module Initialization: [0]
- System Boot Status Settings: [1]

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [2]

Table 13-210. CTRL_CORE_EVE_IRQ_0_1

Address Offset	0x0000 07A0		
Physical Address	0x4A00 27A0	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								EVE1_IRQ_1								RESERVED								EVE1_IRQ_0							

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	EVE1_IRQ_1		RW	0x2
15:9	RESERVED		R	0x0
8:0	EVE1_IRQ_0		RW	0x1

Table 13-211. Register Call Summary for Register CTRL_CORE_EVE_IRQ_0_1

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-212. CTRL_CORE_EVE_IRQ_2_3

Address Offset	0x0000 07A4		
Physical Address	0x4A00 27A4	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								EVE1_IRQ_3								RESERVED								EVE1_IRQ_2							

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	EVE1_IRQ_3		RW	0x4
15:9	RESERVED		R	0x0
8:0	EVE1_IRQ_2		RW	0x3

Table 13-213. Register Call Summary for Register CTRL_CORE_EVE_IRQ_2_3

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-214. CTRL_CORE_EVE_IRQ_4_5

Address Offset	0x0000 07A8		
Physical Address	0x4A00 27A8	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								EVE1_IRQ_5								RESERVED								EVE1_IRQ_4							

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	EVE1_IRQ_5		RW	0x6
15:9	RESERVED		R	0x0
8:0	EVE1_IRQ_4		RW	0x5

Table 13-215. Register Call Summary for Register CTRL_CORE_EVE_IRQ_4_5

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-216. CTRL_CORE_EVE_IRQ_6_7

Address Offset	0x0000 07AC		
Physical Address	0x4A00 27AC	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								EVE1_IRQ_7								RESERVED								EVE1_IRQ_6							

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	EVE1_IRQ_7		RW	0x8
15:9	RESERVED		R	0x0
8:0	EVE1_IRQ_6		RW	0x7

Table 13-217. Register Call Summary for Register CTRL_CORE_EVE_IRQ_6_7

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-218. CTRL_CORE_IPU_IRQ_23_24

Address Offset	0x0000 07E0		
Physical Address	0x4A00 27E0	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								IPU1_IRQ_24								RESERVED								IPU1_IRQ_23							

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	IPU1_IRQ_24		RW	0x30
15:9	RESERVED		R	0x0
8:0	IPU1_IRQ_23		RW	0x14

Table 13-219. Register Call Summary for Register CTRL_CORE_IPU_IRQ_23_24

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-220. CTRL_CORE_IPU_IRQ_25_26

Address Offset	0x0000 07E4		
Physical Address	0x4A00 27E4	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								IPU1_IRQ_26								RESERVED								IPU1_IRQ_25							

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	IPU1_IRQ_26		RW	0x60
15:9	RESERVED		R	0x0
8:0	IPU1_IRQ_25		RW	0x0

Table 13-221. Register Call Summary for Register CTRL_CORE_IPU_IRQ_25_26

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-222. CTRL_CORE_IPU_IRQ_27_28

Address Offset	0x0000 07E8		
Physical Address	0x4A00 27E8	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								IPU1_IRQ_28								RESERVED								IPU1_IRQ_27							

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	IPU1_IRQ_28		RW	0x7F
15:9	RESERVED		R	0x0
8:0	IPU1_IRQ_27		RW	0x7E

Table 13-223. Register Call Summary for Register CTRL_CORE_IPU_IRQ_27_28

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-224. CTRL_CORE_IPU_IRQ_29_30

Address Offset	0x0000 07EC		
Physical Address	0x4A00 27EC	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								IPU1_IRQ_30								RESERVED								IPU1_IRQ_29							

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	IPU1_IRQ_30		RW	0x81
15:9	RESERVED		R	0x0
8:0	IPU1_IRQ_29		RW	0x80

Table 13-225. Register Call Summary for Register CTRL_CORE_IPU_IRQ_29_30

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-226. CTRL_CORE_IPU_IRQ_31_32

Address Offset	0x0000 07F0		
Physical Address	0x4A00 27F0	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								IPU1_IRQ_32								RESERVED								IPU1_IRQ_31							

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	IPU1_IRQ_32		RW	0x13
15:9	RESERVED		R	0x0
8:0	IPU1_IRQ_31		RW	0x82

Table 13-227. Register Call Summary for Register CTRL_CORE_IPU_IRQ_31_32

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-228. CTRL_CORE_IPU_IRQ_33_34

Address Offset	0x0000 07F4		
Physical Address	0x4A00 27F4	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								IPU1_IRQ_34								RESERVED								IPU1_IRQ_33							

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	IPU1_IRQ_34		RW	0x7
15:9	RESERVED		R	0x0
8:0	IPU1_IRQ_33		RW	0x83

Table 13-229. Register Call Summary for Register CTRL_CORE_IPU_IRQ_33_34

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-230. CTRL_CORE_IPU_IRQ_35_36

Address Offset	0x0000 07F8		
Physical Address	0x4A00 27F8	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								IPU1_IRQ_36								RESERVED								IPU1_IRQ_35							

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	IPU1_IRQ_36		RW	0x9
15:9	RESERVED		R	0x0
8:0	IPU1_IRQ_35		RW	0x8

Table 13-231. Register Call Summary for Register CTRL_CORE_IPU_IRQ_35_36

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-232. CTRL_CORE_IPU_IRQ_37_38

Address Offset	0x0000 07FC		
Physical Address	0x4A00 27FC	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								IPU1_IRQ_38								RESERVED								IPU1_IRQ_37							

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	IPU1_IRQ_38		RW	0x84
15:9	RESERVED		R	0x0
8:0	IPU1_IRQ_37		RW	0xA

Table 13-233. Register Call Summary for Register CTRL_CORE_IPU_IRQ_37_38

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-234. CTRL_CORE_IPU_IRQ_39_40

Address Offset	0x0000 0800		
Physical Address	0x4A00 2800	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								IPU1_IRQ_40								RESERVED								IPU1_IRQ_39							

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	IPU1_IRQ_40		RW	0x63
15:9	RESERVED		R	0x0
8:0	IPU1_IRQ_39		RW	0x62

Table 13-235. Register Call Summary for Register CTRL_CORE_IPU_IRQ_39_40

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-236. CTRL_CORE_IPU_IRQ_41_42

Address Offset	0x0000 0804		
Physical Address	0x4A00 2804	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								IPU1_IRQ_42								RESERVED								IPU1_IRQ_41							

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	IPU1_IRQ_42		RW	0x34
15:9	RESERVED		R	0x0
8:0	IPU1_IRQ_41		RW	0x33

Table 13-237. Register Call Summary for Register CTRL_CORE_IPU_IRQ_41_42

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-238. CTRL_CORE_IPU_IRQ_43_44

Address Offset	0x0000 0808		
Physical Address	0x4A00 2808	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								IPU1_IRQ_44								RESERVED								IPU1_IRQ_43							

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	IPU1_IRQ_44		RW	0x39
15:9	RESERVED		R	0x0
8:0	IPU1_IRQ_43		RW	0x38

Table 13-239. Register Call Summary for Register CTRL_CORE_IPU_IRQ_43_44

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-240. CTRL_CORE_IPU_IRQ_45_46

Address Offset	0x0000 080C		
Physical Address	0x4A00 280C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								IPU1_IRQ_46								RESERVED								IPU1_IRQ_45							

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	IPU1_IRQ_46		RW	0x5
15:9	RESERVED		R	0x0
8:0	IPU1_IRQ_45		RW	0x45

Table 13-241. Register Call Summary for Register CTRL_CORE_IPU_IRQ_45_46

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-242. CTRL_CORE_IPU_IRQ_47_48

Address Offset	0x0000 0810		
Physical Address	0x4A00 2810	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								IPU1_IRQ_48								RESERVED								IPU1_IRQ_47							

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	IPU1_IRQ_48		RW	0xE
15:9	RESERVED		R	0x0
8:0	IPU1_IRQ_47		RW	0x85

Table 13-243. Register Call Summary for Register CTRL_CORE_IPU_IRQ_47_48

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-244. CTRL_CORE_IPU_IRQ_49_50

Address Offset	0x0000 0814		
Physical Address	0x4A00 2814	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								IPU1_IRQ_50								RESERVED								IPU1_IRQ_49							

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	IPU1_IRQ_50		RW	0x86
15:9	RESERVED		R	0x0
8:0	IPU1_IRQ_49		RW	0x42

Table 13-245. Register Call Summary for Register CTRL_CORE_IPU_IRQ_49_50

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-246. CTRL_CORE_IPU_IRQ_51_52

Address Offset	0x0000 0818		
Physical Address	0x4A00 2818	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								IPU1_IRQ_52								RESERVED								IPU1_IRQ_51							

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	IPU1_IRQ_52		RW	0x19
15:9	RESERVED		R	0x0
8:0	IPU1_IRQ_51		RW	0x18

Table 13-247. Register Call Summary for Register CTRL_CORE_IPU_IRQ_51_52

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-248. CTRL_CORE_IPU_IRQ_53_54

Address Offset	0x0000 081C		
Physical Address	0x4A00 281C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								IPU1_IRQ_54								RESERVED								IPU1_IRQ_53							

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	IPU1_IRQ_54		RW	0x23
15:9	RESERVED		R	0x0
8:0	IPU1_IRQ_53		RW	0x22

Table 13-249. Register Call Summary for Register CTRL_CORE_IPU_IRQ_53_54

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-250. CTRL_CORE_IPU_IRQ_55_56

Address Offset	0x0000 0820		
Physical Address	0x4A00 2820	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								IPU1_IRQ_56								RESERVED								IPU1_IRQ_55							

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	IPU1_IRQ_56		RW	0x2A
15:9	RESERVED		R	0x0
8:0	IPU1_IRQ_55		RW	0x28

Table 13-251. Register Call Summary for Register CTRL_CORE_IPU_IRQ_55_56

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-252. CTRL_CORE_IPU_IRQ_57_58

Address Offset	0x0000 0824		
Physical Address	0x4A00 2824	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								IPU1_IRQ_58								RESERVED								IPU1_IRQ_57							

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	IPU1_IRQ_58		RW	0x3D
15:9	RESERVED		R	0x0
8:0	IPU1_IRQ_57		RW	0x3C

Table 13-253. Register Call Summary for Register CTRL_CORE_IPU_IRQ_57_58

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-254. CTRL_CORE_IPU_IRQ_59_60

Address Offset	0x0000 0828		
Physical Address	0x4A00 2828	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								IPU1_IRQ_60								RESERVED								IPU1_IRQ_59							

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	IPU1_IRQ_60		RW	0x0
15:9	RESERVED		R	0x0
8:0	IPU1_IRQ_59		RW	0x32

Table 13-255. Register Call Summary for Register CTRL_CORE_IPU_IRQ_59_60

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-256. CTRL_CORE_IPU_IRQ_61_62

Address Offset	0x0000 082C		
Physical Address	0x4A00 282C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								IPU1_IRQ_62								RESERVED								IPU1_IRQ_61							

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	IPU1_IRQ_62		RW	0x16
15:9	RESERVED		R	0x0
8:0	IPU1_IRQ_61		RW	0x0

Table 13-257. Register Call Summary for Register CTRL_CORE_IPU_IRQ_61_62

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-258. CTRL_CORE_IPU_IRQ_63_64

Address Offset	0x0000 0830		
Physical Address	0x4A00 2830	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								IPU1_IRQ_64								RESERVED								IPU1_IRQ_63							

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	IPU1_IRQ_64		RW	0x6C
15:9	RESERVED		R	0x0
8:0	IPU1_IRQ_63		RW	0x53

Table 13-259. Register Call Summary for Register CTRL_CORE_IPU_IRQ_63_64

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-260. CTRL_CORE_IPU_IRQ_65_66

Address Offset	0x0000 0834		
Physical Address	0x4A00 2834	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								IPU1_IRQ_66								RESERVED								IPU1_IRQ_65							

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	IPU1_IRQ_66		RW	0x4E
15:9	RESERVED		R	0x0
8:0	IPU1_IRQ_65		RW	0x78

Table 13-261. Register Call Summary for Register CTRL_CORE_IPU_IRQ_65_66

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-262. CTRL_CORE_IPU_IRQ_67_68

Address Offset	0x0000 0838		
Physical Address	0x4A00 2838	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								IPU1_IRQ_68								RESERVED								IPU1_IRQ_67							

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	IPU1_IRQ_68		RW	0x59
15:9	RESERVED		R	0x0
8:0	IPU1_IRQ_67		RW	0x51

Table 13-263. Register Call Summary for Register CTRL_CORE_IPU_IRQ_67_68

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-264. CTRL_CORE_IPU_IRQ_69_70

Address Offset	0x0000 083C		
Physical Address	0x4A00 283C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								IPU1_IRQ_70								RESERVED								IPU1_IRQ_69							

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	IPU1_IRQ_70		RW	0x0
15:9	RESERVED		R	0x0
8:0	IPU1_IRQ_69		RW	0x0

Table 13-265. Register Call Summary for Register CTRL_CORE_IPU_IRQ_69_70

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-266. CTRL_CORE_IPU_IRQ_71_72

Address Offset	0x0000 0840		
Physical Address	0x4A00 2840	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								IPU1_IRQ_72								RESERVED								IPU1_IRQ_71							

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	IPU1_IRQ_72		RW	0x76
15:9	RESERVED		R	0x0
8:0	IPU1_IRQ_71		RW	0x77

Table 13-267. Register Call Summary for Register CTRL_CORE_IPU_IRQ_71_72

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-268. CTRL_CORE_IPU_IRQ_73_74

Address Offset	0x0000 0844		
Physical Address	0x4A00 2844	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								IPU1_IRQ_74								RESERVED								IPU1_IRQ_73							

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	IPU1_IRQ_74		RW	0x49
15:9	RESERVED		R	0x0
8:0	IPU1_IRQ_73		RW	0x48

Table 13-269. Register Call Summary for Register CTRL_CORE_IPU_IRQ_73_74

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-270. CTRL_CORE_IPU_IRQ_75_76

Address Offset	0x0000 0848		
Physical Address	0x4A00 2848	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								IPU1_IRQ_76								RESERVED								IPU1_IRQ_75							

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	IPU1_IRQ_76		RW	0x57
15:9	RESERVED		R	0x0
8:0	IPU1_IRQ_75		RW	0x75

Table 13-271. Register Call Summary for Register CTRL_CORE_IPU_IRQ_75_76

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-272. CTRL_CORE_IPU_IRQ_77_78

Address Offset	0x0000 084C		
Physical Address	0x4A00 284C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								IPU1_IRQ_78								RESERVED								IPU1_IRQ_77							

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	IPU1_IRQ_78		RW	0x3E
15:9	RESERVED		R	0x0
8:0	IPU1_IRQ_77		RW	0x58

Table 13-273. Register Call Summary for Register CTRL_CORE_IPU_IRQ_77_78

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-274. CTRL_CORE_IPU_IRQ_79_80

Address Offset	0x0000 0850		
Physical Address	0x4A00 2850	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								IPU1_IRQ_79							

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8:0	IPU1_IRQ_79		RW	0x3F

Table 13-275. Register Call Summary for Register CTRL_CORE_IPU_IRQ_79_80

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-276. CTRL_CORE_DSP1_IRQ_32_33

Address Offset	0x0000 0948		
Physical Address	0x4A00 2948	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DSP1_IRQ_33								RESERVED								DSP1_IRQ_32							

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	DSP1_IRQ_33		RW	0x2
15:9	RESERVED		R	0x0
8:0	DSP1_IRQ_32		RW	0x1

Table 13-277. Register Call Summary for Register CTRL_CORE_DSP1_IRQ_32_33

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-278. CTRL_CORE_DSP1_IRQ_34_35

Address Offset 0x0000 094C
Physical Address 0x4A00 294C **Instance** CTRL_MODULE_CORE
Description
Type RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DSP1_IRQ_35								RESERVED								DSP1_IRQ_34							

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	DSP1_IRQ_35		RW	0x4
15:9	RESERVED		R	0x0
8:0	DSP1_IRQ_34		RW	0x3

Table 13-279. Register Call Summary for Register CTRL_CORE_DSP1_IRQ_34_35

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-280. CTRL_CORE_DSP1_IRQ_36_37

Address Offset 0x0000 0950
Physical Address 0x4A00 2950 **Instance** CTRL_MODULE_CORE
Description
Type RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DSP1_IRQ_37								RESERVED								DSP1_IRQ_36							

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	DSP1_IRQ_37		RW	0x6
15:9	RESERVED		R	0x0
8:0	DSP1_IRQ_36		RW	0x5

Table 13-281. Register Call Summary for Register CTRL_CORE_DSP1_IRQ_36_37

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-282. CTRL_CORE_DSP1_IRQ_38_39

Address Offset	0x0000 0954		
Physical Address	0x4A00 2954	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DSP1_IRQ_39								RESERVED								DSP1_IRQ_38							

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	DSP1_IRQ_39		RW	0x8
15:9	RESERVED		R	0x0
8:0	DSP1_IRQ_38		RW	0x7

Table 13-283. Register Call Summary for Register CTRL_CORE_DSP1_IRQ_38_39

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-284. CTRL_CORE_DSP1_IRQ_40_41

Address Offset	0x0000 0958		
Physical Address	0x4A00 2958	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DSP1_IRQ_41								RESERVED								DSP1_IRQ_40							

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	DSP1_IRQ_41		RW	0xA
15:9	RESERVED		R	0x0
8:0	DSP1_IRQ_40		RW	0x9

Table 13-285. Register Call Summary for Register CTRL_CORE_DSP1_IRQ_40_41

Control Module Functional Description
- IRQ_CROSSBAR Module Functional Description: [0]

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [1]

Table 13-286. CTRL_CORE_DSP1_IRQ_42_43

Address Offset	0x0000 095C		
Physical Address	0x4A00 295C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DSP1_IRQ_43								RESERVED								DSP1_IRQ_42							

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	DSP1_IRQ_43		RW	0xC
15:9	RESERVED		R	0x0
8:0	DSP1_IRQ_42		RW	0xB

Table 13-287. Register Call Summary for Register CTRL_CORE_DSP1_IRQ_42_43

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-288. CTRL_CORE_DSP1_IRQ_44_45

Address Offset	0x0000 0960		
Physical Address	0x4A00 2960	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DSP1_IRQ_45								RESERVED								DSP1_IRQ_44							

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	DSP1_IRQ_45		RW	0xE
15:9	RESERVED		R	0x0
8:0	DSP1_IRQ_44		RW	0xD

Table 13-289. Register Call Summary for Register CTRL_CORE_DSP1_IRQ_44_45

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-290. CTRL_CORE_DSP1_IRQ_46_47

Address Offset	0x0000 0964		
Physical Address	0x4A00 2964	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DSP1_IRQ_47								RESERVED								DSP1_IRQ_46							

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	DSP1_IRQ_47		RW	0x10
15:9	RESERVED		R	0x0
8:0	DSP1_IRQ_46		RW	0xF

Table 13-291. Register Call Summary for Register CTRL_CORE_DSP1_IRQ_46_47

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-292. CTRL_CORE_DSP1_IRQ_48_49

Address Offset	0x0000 0968		
Physical Address	0x4A00 2968	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DSP1_IRQ_49								RESERVED								DSP1_IRQ_48							

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	DSP1_IRQ_49		RW	0x12
15:9	RESERVED		R	0x0
8:0	DSP1_IRQ_48		RW	0x11

Table 13-293. Register Call Summary for Register CTRL_CORE_DSP1_IRQ_48_49

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-294. CTRL_CORE_DSP1_IRQ_50_51

Address Offset	0x0000 096C		
Physical Address	0x4A00 296C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DSP1_IRQ_51								RESERVED								DSP1_IRQ_50							

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	DSP1_IRQ_51		RW	0x14
15:9	RESERVED		R	0x0
8:0	DSP1_IRQ_50		RW	0x13

Table 13-295. Register Call Summary for Register CTRL_CORE_DSP1_IRQ_50_51

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-296. CTRL_CORE_DSP1_IRQ_52_53

Address Offset	0x0000 0970		
Physical Address	0x4A00 2970	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DSP1_IRQ_53								RESERVED								DSP1_IRQ_52							

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	DSP1_IRQ_53		RW	0x16
15:9	RESERVED		R	0x0
8:0	DSP1_IRQ_52		RW	0x15

Table 13-297. Register Call Summary for Register CTRL_CORE_DSP1_IRQ_52_53

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-298. CTRL_CORE_DSP1_IRQ_54_55

Address Offset	0x0000 0974		
Physical Address	0x4A00 2974	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DSP1_IRQ_55								RESERVED								DSP1_IRQ_54							

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	DSP1_IRQ_55		RW	0x18
15:9	RESERVED		R	0x0
8:0	DSP1_IRQ_54		RW	0x17

Table 13-299. Register Call Summary for Register CTRL_CORE_DSP1_IRQ_54_55

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-300. CTRL_CORE_DSP1_IRQ_56_57

Address Offset	0x0000 0978		
Physical Address	0x4A00 2978	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DSP1_IRQ_57								RESERVED								DSP1_IRQ_56							

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	DSP1_IRQ_57		RW	0x1A
15:9	RESERVED		R	0x0
8:0	DSP1_IRQ_56		RW	0x19

Table 13-301. Register Call Summary for Register CTRL_CORE_DSP1_IRQ_56_57

Control Module Functional Description
- IRQ_CROSSBAR Module Functional Description: [0][1]

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [2]

Table 13-302. CTRL_CORE_DSP1_IRQ_58_59

Address Offset	0x0000 097C		
Physical Address	0x4A00 297C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DSP1_IRQ_59								RESERVED								DSP1_IRQ_58							

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	DSP1_IRQ_59		RW	0x1C
15:9	RESERVED		R	0x0
8:0	DSP1_IRQ_58		RW	0x1B

Table 13-303. Register Call Summary for Register CTRL_CORE_DSP1_IRQ_58_59

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-304. CTRL_CORE_DSP1_IRQ_60_61

Address Offset	0x0000 0980		
Physical Address	0x4A00 2980	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DSP1_IRQ_61								RESERVED								DSP1_IRQ_60							

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	DSP1_IRQ_61		RW	0x1E
15:9	RESERVED		R	0x0
8:0	DSP1_IRQ_60		RW	0x1D

Table 13-305. Register Call Summary for Register CTRL_CORE_DSP1_IRQ_60_61

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-306. CTRL_CORE_DSP1_IRQ_62_63

Address Offset	0x0000 0984		
Physical Address	0x4A00 2984	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DSP1_IRQ_63								RESERVED								DSP1_IRQ_62							

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	DSP1_IRQ_63		RW	0x20
15:9	RESERVED		R	0x0
8:0	DSP1_IRQ_62		RW	0x1F

Table 13-307. Register Call Summary for Register CTRL_CORE_DSP1_IRQ_62_63

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-308. CTRL_CORE_DSP1_IRQ_64_65

Address Offset	0x0000 0988		
Physical Address	0x4A00 2988	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DSP1_IRQ_65								RESERVED								DSP1_IRQ_64							

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	DSP1_IRQ_65		RW	0x22
15:9	RESERVED		R	0x0
8:0	DSP1_IRQ_64		RW	0x21

Table 13-309. Register Call Summary for Register CTRL_CORE_DSP1_IRQ_64_65

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-310. CTRL_CORE_DSP1_IRQ_66_67

Address Offset	0x0000 098C		
Physical Address	0x4A00 298C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DSP1_IRQ_67								RESERVED								DSP1_IRQ_66							

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	DSP1_IRQ_67		RW	0x24
15:9	RESERVED		R	0x0
8:0	DSP1_IRQ_66		RW	0x23

Table 13-311. Register Call Summary for Register CTRL_CORE_DSP1_IRQ_66_67

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-312. CTRL_CORE_DSP1_IRQ_68_69

Address Offset	0x0000 0990		
Physical Address	0x4A00 2990	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DSP1_IRQ_69								RESERVED								DSP1_IRQ_68							

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	DSP1_IRQ_69		RW	0x26
15:9	RESERVED		R	0x0
8:0	DSP1_IRQ_68		RW	0x25

Table 13-313. Register Call Summary for Register CTRL_CORE_DSP1_IRQ_68_69

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-314. CTRL_CORE_DSP1_IRQ_70_71

Address Offset	0x0000 0994		
Physical Address	0x4A00 2994	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DSP1_IRQ_71								RESERVED								DSP1_IRQ_70							

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	DSP1_IRQ_71		RW	0x28
15:9	RESERVED		R	0x0
8:0	DSP1_IRQ_70		RW	0x27

Table 13-315. Register Call Summary for Register CTRL_CORE_DSP1_IRQ_70_71

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-316. CTRL_CORE_DSP1_IRQ_72_73

Address Offset	0x0000 0998		
Physical Address	0x4A00 2998	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DSP1_IRQ_73								RESERVED								DSP1_IRQ_72							

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	DSP1_IRQ_73		RW	0x2A
15:9	RESERVED		R	0x0
8:0	DSP1_IRQ_72		RW	0x29

Table 13-317. Register Call Summary for Register CTRL_CORE_DSP1_IRQ_72_73

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-318. CTRL_CORE_DSP1_IRQ_74_75

Address Offset	0x0000 099C		
Physical Address	0x4A00 299C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DSP1_IRQ_75								RESERVED								DSP1_IRQ_74							

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	DSP1_IRQ_75		RW	0x2C
15:9	RESERVED		R	0x0
8:0	DSP1_IRQ_74		RW	0x2B

Table 13-319. Register Call Summary for Register CTRL_CORE_DSP1_IRQ_74_75

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-320. CTRL_CORE_DSP1_IRQ_76_77

Address Offset	0x0000 09A0		
Physical Address	0x4A00 29A0	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DSP1_IRQ_77								RESERVED								DSP1_IRQ_76							

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	DSP1_IRQ_77		RW	0x2E
15:9	RESERVED		R	0x0
8:0	DSP1_IRQ_76		RW	0x2D

Table 13-321. Register Call Summary for Register CTRL_CORE_DSP1_IRQ_76_77

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-322. CTRL_CORE_DSP1_IRQ_78_79

Address Offset	0x0000 09A4		
Physical Address	0x4A00 29A4	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DSP1_IRQ_79								RESERVED								DSP1_IRQ_78							

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	DSP1_IRQ_79		RW	0x30
15:9	RESERVED		R	0x0
8:0	DSP1_IRQ_78		RW	0x2F

Table 13-323. Register Call Summary for Register CTRL_CORE_DSP1_IRQ_78_79

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-324. CTRL_CORE_DSP1_IRQ_80_81

Address Offset	0x0000 09A8		
Physical Address	0x4A00 29A8	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DSP1_IRQ_81								RESERVED								DSP1_IRQ_80							

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	DSP1_IRQ_81		RW	0x32
15:9	RESERVED		R	0x0
8:0	DSP1_IRQ_80		RW	0x31

Table 13-325. Register Call Summary for Register CTRL_CORE_DSP1_IRQ_80_81

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-326. CTRL_CORE_DSP1_IRQ_82_83

Address Offset	0x0000 09AC		
Physical Address	0x4A00 29AC	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DSP1_IRQ_83								RESERVED								DSP1_IRQ_82							

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	DSP1_IRQ_83		RW	0x34
15:9	RESERVED		R	0x0
8:0	DSP1_IRQ_82		RW	0x33

Table 13-327. Register Call Summary for Register CTRL_CORE_DSP1_IRQ_82_83

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-328. CTRL_CORE_DSP1_IRQ_84_85

Address Offset	0x0000 09B0		
Physical Address	0x4A00 29B0	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DSP1_IRQ_85								RESERVED								DSP1_IRQ_84							

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	DSP1_IRQ_85		RW	0x36
15:9	RESERVED		R	0x0
8:0	DSP1_IRQ_84		RW	0x35

Table 13-329. Register Call Summary for Register CTRL_CORE_DSP1_IRQ_84_85

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-330. CTRL_CORE_DSP1_IRQ_86_87

Address Offset	0x0000 09B4		
Physical Address	0x4A00 29B4	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DSP1_IRQ_87								RESERVED								DSP1_IRQ_86							

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	DSP1_IRQ_87		RW	0x38
15:9	RESERVED		R	0x0
8:0	DSP1_IRQ_86		RW	0x37

Table 13-331. Register Call Summary for Register CTRL_CORE_DSP1_IRQ_86_87

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-332. CTRL_CORE_DSP1_IRQ_88_89

Address Offset	0x0000 09B8		
Physical Address	0x4A00 29B8	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DSP1_IRQ_89								RESERVED								DSP1_IRQ_88							

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	DSP1_IRQ_89		RW	0x3A
15:9	RESERVED		R	0x0
8:0	DSP1_IRQ_88		RW	0x39

Table 13-333. Register Call Summary for Register CTRL_CORE_DSP1_IRQ_88_89

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-334. CTRL_CORE_DSP1_IRQ_90_91

Address Offset	0x0000 09BC		
Physical Address	0x4A00 29BC	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DSP1_IRQ_91								RESERVED								DSP1_IRQ_90							

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	DSP1_IRQ_91		RW	0x3C
15:9	RESERVED		R	0x0
8:0	DSP1_IRQ_90		RW	0x3B

Table 13-335. Register Call Summary for Register CTRL_CORE_DSP1_IRQ_90_91

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-336. CTRL_CORE_DSP1_IRQ_92_93

Address Offset	0x0000 09C0		
Physical Address	0x4A00 29C0	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DSP1_IRQ_93								RESERVED								DSP1_IRQ_92							

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	DSP1_IRQ_93		RW	0x3E
15:9	RESERVED		R	0x0
8:0	DSP1_IRQ_92		RW	0x3D

Table 13-337. Register Call Summary for Register CTRL_CORE_DSP1_IRQ_92_93

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-338. CTRL_CORE_DSP1_IRQ_94_95

Address Offset	0x0000 09C4		
Physical Address	0x4A00 29C4	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DSP1_IRQ_95								RESERVED								DSP1_IRQ_94							

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	DSP1_IRQ_95		RW	0x40
15:9	RESERVED		R	0x0
8:0	DSP1_IRQ_94		RW	0x3F

Table 13-339. Register Call Summary for Register CTRL_CORE_DSP1_IRQ_94_95

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-340. CTRL_CORE_DSP2_IRQ_32_33

Address Offset	0x0000 09C8		
Physical Address	0x4A00 29C8	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DSP2_IRQ_33								RESERVED								DSP2_IRQ_32							

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	DSP2_IRQ_33		RW	0x2
15:9	RESERVED		R	0x0
8:0	DSP2_IRQ_32		RW	0x1

Table 13-341. Register Call Summary for Register CTRL_CORE_DSP2_IRQ_32_33

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-342. CTRL_CORE_DSP2_IRQ_34_35

Address Offset	0x0000 09CC		
Physical Address	0x4A00 29CC	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DSP2_IRQ_35								RESERVED								DSP2_IRQ_34							

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	DSP2_IRQ_35		RW	0x4
15:9	RESERVED		R	0x0
8:0	DSP2_IRQ_34		RW	0x3

Table 13-343. Register Call Summary for Register CTRL_CORE_DSP2_IRQ_34_35

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-344. CTRL_CORE_DSP2_IRQ_36_37

Address Offset	0x0000 09D0		
Physical Address	0x4A00 29D0	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DSP2_IRQ_37								RESERVED								DSP2_IRQ_36							

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	DSP2_IRQ_37		RW	0x6
15:9	RESERVED		R	0x0
8:0	DSP2_IRQ_36		RW	0x5

Table 13-345. Register Call Summary for Register CTRL_CORE_DSP2_IRQ_36_37

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-346. CTRL_CORE_DSP2_IRQ_38_39

Address Offset	0x0000 09D4		
Physical Address	0x4A00 29D4	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DSP2_IRQ_39								RESERVED								DSP2_IRQ_38							

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	DSP2_IRQ_39		RW	0x8
15:9	RESERVED		R	0x0
8:0	DSP2_IRQ_38		RW	0x7

Table 13-347. Register Call Summary for Register CTRL_CORE_DSP2_IRQ_38_39

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-348. CTRL_CORE_DSP2_IRQ_40_41

Address Offset	0x0000 09D8		
Physical Address	0x4A00 29D8	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DSP2_IRQ_41								RESERVED								DSP2_IRQ_40							

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	DSP2_IRQ_41		RW	0xA
15:9	RESERVED		R	0x0
8:0	DSP2_IRQ_40		RW	0x9

Table 13-349. Register Call Summary for Register CTRL_CORE_DSP2_IRQ_40_41

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-350. CTRL_CORE_DSP2_IRQ_42_43

Address Offset	0x0000 09DC		
Physical Address	0x4A00 29DC	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DSP2_IRQ_43								RESERVED								DSP2_IRQ_42							

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	DSP2_IRQ_43		RW	0xC
15:9	RESERVED		R	0x0
8:0	DSP2_IRQ_42		RW	0xB

Table 13-351. Register Call Summary for Register CTRL_CORE_DSP2_IRQ_42_43

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-352. CTRL_CORE_DSP2_IRQ_44_45

Address Offset	0x0000 09E0		
Physical Address	0x4A00 29E0	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DSP2_IRQ_45								RESERVED								DSP2_IRQ_44							

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	DSP2_IRQ_45		RW	0xE
15:9	RESERVED		R	0x0
8:0	DSP2_IRQ_44		RW	0xD

Table 13-353. Register Call Summary for Register CTRL_CORE_DSP2_IRQ_44_45

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-354. CTRL_CORE_DSP2_IRQ_46_47

Address Offset	0x0000 09E4		
Physical Address	0x4A00 29E4	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DSP2_IRQ_47								RESERVED								DSP2_IRQ_46							

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	DSP2_IRQ_47		RW	0x10
15:9	RESERVED		R	0x0
8:0	DSP2_IRQ_46		RW	0xF

Table 13-355. Register Call Summary for Register CTRL_CORE_DSP2_IRQ_46_47

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-356. CTRL_CORE_DSP2_IRQ_48_49

Address Offset	0x0000 09E8		
Physical Address	0x4A00 29E8	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DSP2_IRQ_49								RESERVED								DSP2_IRQ_48							

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	DSP2_IRQ_49		RW	0x12
15:9	RESERVED		R	0x0
8:0	DSP2_IRQ_48		RW	0x11

Table 13-357. Register Call Summary for Register CTRL_CORE_DSP2_IRQ_48_49

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-358. CTRL_CORE_DSP2_IRQ_50_51

Address Offset	0x0000 09EC		
Physical Address	0x4A00 29EC	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DSP2_IRQ_51								RESERVED								DSP2_IRQ_50							

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	DSP2_IRQ_51		RW	0x14
15:9	RESERVED		R	0x0
8:0	DSP2_IRQ_50		RW	0x13

Table 13-359. Register Call Summary for Register CTRL_CORE_DSP2_IRQ_50_51

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-360. CTRL_CORE_DSP2_IRQ_52_53

Address Offset	0x0000 09F0		
Physical Address	0x4A00 29F0	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DSP2_IRQ_53								RESERVED								DSP2_IRQ_52							

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	DSP2_IRQ_53		RW	0x16
15:9	RESERVED		R	0x0
8:0	DSP2_IRQ_52		RW	0x15

Table 13-361. Register Call Summary for Register CTRL_CORE_DSP2_IRQ_52_53

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-362. CTRL_CORE_DSP2_IRQ_54_55

Address Offset	0x0000 09F4		
Physical Address	0x4A00 29F4	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DSP2_IRQ_55								RESERVED								DSP2_IRQ_54							

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	DSP2_IRQ_55		RW	0x18
15:9	RESERVED		R	0x0
8:0	DSP2_IRQ_54		RW	0x17

Table 13-363. Register Call Summary for Register CTRL_CORE_DSP2_IRQ_54_55

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-364. CTRL_CORE_DSP2_IRQ_56_57

Address Offset	0x0000 09F8		
Physical Address	0x4A00 29F8	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DSP2_IRQ_57								RESERVED								DSP2_IRQ_56							

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	DSP2_IRQ_57		RW	0x1A
15:9	RESERVED		R	0x0
8:0	DSP2_IRQ_56		RW	0x19

Table 13-365. Register Call Summary for Register CTRL_CORE_DSP2_IRQ_56_57

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-366. CTRL_CORE_DSP2_IRQ_58_59

Address Offset	0x0000 09FC		
Physical Address	0x4A00 29FC	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DSP2_IRQ_59								RESERVED								DSP2_IRQ_58							

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	DSP2_IRQ_59		RW	0x1C
15:9	RESERVED		R	0x0
8:0	DSP2_IRQ_58		RW	0x1B

Table 13-367. Register Call Summary for Register CTRL_CORE_DSP2_IRQ_58_59

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-368. CTRL_CORE_DSP2_IRQ_60_61

Address Offset	0x0000 0A00		
Physical Address	0x4A00 2A00	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DSP2_IRQ_61								RESERVED								DSP2_IRQ_60							

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	DSP2_IRQ_61		RW	0x1E
15:9	RESERVED		R	0x0
8:0	DSP2_IRQ_60		RW	0x1D

Table 13-369. Register Call Summary for Register CTRL_CORE_DSP2_IRQ_60_61

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-370. CTRL_CORE_DSP2_IRQ_62_63

Address Offset	0x0000 0A04		
Physical Address	0x4A00 2A04	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DSP2_IRQ_63								RESERVED								DSP2_IRQ_62							

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	DSP2_IRQ_63		RW	0x20
15:9	RESERVED		R	0x0
8:0	DSP2_IRQ_62		RW	0x1F

Table 13-371. Register Call Summary for Register CTRL_CORE_DSP2_IRQ_62_63

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-372. CTRL_CORE_DSP2_IRQ_64_65

Address Offset	0x0000 0A08		
Physical Address	0x4A00 2A08	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DSP2_IRQ_65								RESERVED								DSP2_IRQ_64							

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	DSP2_IRQ_65		RW	0x22
15:9	RESERVED		R	0x0
8:0	DSP2_IRQ_64		RW	0x21

Table 13-373. Register Call Summary for Register CTRL_CORE_DSP2_IRQ_64_65

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-374. CTRL_CORE_DSP2_IRQ_66_67

Address Offset	0x0000 0A0C		
Physical Address	0x4A00 2A0C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DSP2_IRQ_67								RESERVED								DSP2_IRQ_66							

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	DSP2_IRQ_67		RW	0x24
15:9	RESERVED		R	0x0
8:0	DSP2_IRQ_66		RW	0x23

Table 13-375. Register Call Summary for Register CTRL_CORE_DSP2_IRQ_66_67

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-376. CTRL_CORE_DSP2_IRQ_68_69

Address Offset	0x0000 0A10		
Physical Address	0x4A00 2A10	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DSP2_IRQ_69								RESERVED								DSP2_IRQ_68							

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	DSP2_IRQ_69		RW	0x26
15:9	RESERVED		R	0x0
8:0	DSP2_IRQ_68		RW	0x25

Table 13-377. Register Call Summary for Register CTRL_CORE_DSP2_IRQ_68_69

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-378. CTRL_CORE_DSP2_IRQ_70_71

Address Offset	0x0000 0A14		
Physical Address	0x4A00 2A14	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DSP2_IRQ_71								RESERVED								DSP2_IRQ_70							

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	DSP2_IRQ_71		RW	0x28
15:9	RESERVED		R	0x0
8:0	DSP2_IRQ_70		RW	0x27

Table 13-379. Register Call Summary for Register CTRL_CORE_DSP2_IRQ_70_71

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-380. CTRL_CORE_DSP2_IRQ_72_73

Address Offset	0x0000 0A18		
Physical Address	0x4A00 2A18	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DSP2_IRQ_73								RESERVED								DSP2_IRQ_72							

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	DSP2_IRQ_73		RW	0x2A
15:9	RESERVED		R	0x0
8:0	DSP2_IRQ_72		RW	0x29

Table 13-381. Register Call Summary for Register CTRL_CORE_DSP2_IRQ_72_73

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-382. CTRL_CORE_DSP2_IRQ_74_75

Address Offset	0x0000 0A1C		
Physical Address	0x4A00 2A1C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DSP2_IRQ_75								RESERVED								DSP2_IRQ_74							

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	DSP2_IRQ_75		RW	0x2C
15:9	RESERVED		R	0x0
8:0	DSP2_IRQ_74		RW	0x2B

Table 13-383. Register Call Summary for Register CTRL_CORE_DSP2_IRQ_74_75

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-384. CTRL_CORE_DSP2_IRQ_76_77

Address Offset	0x0000 0A20		
Physical Address	0x4A00 2A20	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DSP2_IRQ_77								RESERVED								DSP2_IRQ_76							

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	DSP2_IRQ_77		RW	0x2E
15:9	RESERVED		R	0x0
8:0	DSP2_IRQ_76		RW	0x2D

Table 13-385. Register Call Summary for Register CTRL_CORE_DSP2_IRQ_76_77

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-386. CTRL_CORE_DSP2_IRQ_78_79

Address Offset	0x0000 0A24		
Physical Address	0x4A00 2A24	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DSP2_IRQ_79								RESERVED								DSP2_IRQ_78							

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	DSP2_IRQ_79		RW	0x30
15:9	RESERVED		R	0x0
8:0	DSP2_IRQ_78		RW	0x2F

Table 13-387. Register Call Summary for Register CTRL_CORE_DSP2_IRQ_78_79

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-388. CTRL_CORE_DSP2_IRQ_80_81

Address Offset	0x0000 0A28		
Physical Address	0x4A00 2A28	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DSP2_IRQ_81								RESERVED								DSP2_IRQ_80							

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	DSP2_IRQ_81		RW	0x32
15:9	RESERVED		R	0x0
8:0	DSP2_IRQ_80		RW	0x31

Table 13-389. Register Call Summary for Register CTRL_CORE_DSP2_IRQ_80_81

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-390. CTRL_CORE_DSP2_IRQ_82_83

Address Offset	0x0000 0A2C		
Physical Address	0x4A00 2A2C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DSP2_IRQ_83								RESERVED								DSP2_IRQ_82							

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	DSP2_IRQ_83		RW	0x34
15:9	RESERVED		R	0x0
8:0	DSP2_IRQ_82		RW	0x33

Table 13-391. Register Call Summary for Register CTRL_CORE_DSP2_IRQ_82_83

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-392. CTRL_CORE_DSP2_IRQ_84_85

Address Offset	0x0000 0A30		
Physical Address	0x4A00 2A30	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DSP2_IRQ_85								RESERVED								DSP2_IRQ_84							

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	DSP2_IRQ_85		RW	0x36
15:9	RESERVED		R	0x0
8:0	DSP2_IRQ_84		RW	0x35

Table 13-393. Register Call Summary for Register CTRL_CORE_DSP2_IRQ_84_85

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-394. CTRL_CORE_DSP2_IRQ_86_87

Address Offset	0x0000 0A34		
Physical Address	0x4A00 2A34	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DSP2_IRQ_87								RESERVED								DSP2_IRQ_86							

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	DSP2_IRQ_87		RW	0x38
15:9	RESERVED		R	0x0
8:0	DSP2_IRQ_86		RW	0x37

Table 13-395. Register Call Summary for Register CTRL_CORE_DSP2_IRQ_86_87

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-396. CTRL_CORE_DSP2_IRQ_88_89

Address Offset	0x0000 0A38		
Physical Address	0x4A00 2A38	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DSP2_IRQ_89								RESERVED								DSP2_IRQ_88							

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	DSP2_IRQ_89		RW	0x3A
15:9	RESERVED		R	0x0
8:0	DSP2_IRQ_88		RW	0x39

Table 13-397. Register Call Summary for Register CTRL_CORE_DSP2_IRQ_88_89

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-398. CTRL_CORE_DSP2_IRQ_90_91

Address Offset	0x0000 0A3C		
Physical Address	0x4A00 2A3C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DSP2_IRQ_91								RESERVED								DSP2_IRQ_90							

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	DSP2_IRQ_91		RW	0x3C
15:9	RESERVED		R	0x0
8:0	DSP2_IRQ_90		RW	0x3B

Table 13-399. Register Call Summary for Register CTRL_CORE_DSP2_IRQ_90_91

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-400. CTRL_CORE_DSP2_IRQ_92_93

Address Offset	0x0000 0A40		
Physical Address	0x4A00 2A40	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DSP2_IRQ_93								RESERVED								DSP2_IRQ_92							

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	DSP2_IRQ_93		RW	0x3E
15:9	RESERVED		R	0x0
8:0	DSP2_IRQ_92		RW	0x3D

Table 13-401. Register Call Summary for Register CTRL_CORE_DSP2_IRQ_92_93

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-402. CTRL_CORE_DSP2_IRQ_94_95

Address Offset	0x0000 0A44		
Physical Address	0x4A00 2A44	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DSP2_IRQ_95								RESERVED								DSP2_IRQ_94							

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	DSP2_IRQ_95		RW	0x40
15:9	RESERVED		R	0x0
8:0	DSP2_IRQ_94		RW	0x3F

Table 13-403. Register Call Summary for Register CTRL_CORE_DSP2_IRQ_94_95

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-404. CTRL_CORE_DMA_EDMA_DREQ_0_1

Address Offset	0x0000 0C78		
Physical Address	0x4A00 2C78	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DMA_EDMA_DREQ_1_IRQ_1								RESERVED								DMA_EDMA_DREQ_0_IRQ_0							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_EDMA_DREQ_1_IRQ_1		RW	0x2
15:8	RESERVED		R	0x0
7:0	DMA_EDMA_DREQ_0_IRQ_0		RW	0x1

Table 13-405. Register Call Summary for Register CTRL_CORE_DMA_EDMA_DREQ_0_1

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-406. CTRL_CORE_DMA_EDMA_DREQ_2_3

Address Offset	0x0000 0C7C		
Physical Address	0x4A00 2C7C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DMA_EDMA_DREQ_3_IRQ_3								RESERVED								DMA_EDMA_DREQ_2_IRQ_2							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_EDMA_DREQ_3_IRQ_3		RW	0x4
15:8	RESERVED		R	0x0
7:0	DMA_EDMA_DREQ_2_IRQ_2		RW	0x3

Table 13-407. Register Call Summary for Register CTRL_CORE_DMA_EDMA_DREQ_2_3

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-408. CTRL_CORE_DMA_EDMA_DREQ_4_5

Address Offset	0x0000 0C80		
Physical Address	0x4A00 2C80	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DMA_EDMA_DREQ_5_IRQ_5								RESERVED								DMA_EDMA_DREQ_4_IRQ_4							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_EDMA_DREQ_5_IRQ_5		RW	0x6
15:8	RESERVED		R	0x0
7:0	DMA_EDMA_DREQ_4_IRQ_4		RW	0x5

Table 13-409. Register Call Summary for Register CTRL_CORE_DMA_EDMA_DREQ_4_5

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-410. CTRL_CORE_DMA_EDMA_DREQ_6_7

Address Offset	0x0000 0C84		
Physical Address	0x4A00 2C84	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DMA_EDMA_DREQ_7_IRQ_7								RESERVED								DMA_EDMA_DREQ_6_IRQ_6							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_EDMA_DREQ_7_IRQ_7		RW	0x8
15:8	RESERVED		R	0x0
7:0	DMA_EDMA_DREQ_6_IRQ_6		RW	0x7

Table 13-411. Register Call Summary for Register CTRL_CORE_DMA_EDMA_DREQ_6_7

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-412. CTRL_CORE_DMA_EDMA_DREQ_8_9

Address Offset	0x0000 0C88		
Physical Address	0x4A00 2C88	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DMA_EDMA_DREQ_9_IRQ_9								RESERVED								DMA_EDMA_DREQ_8_IRQ_8							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_EDMA_DREQ_9_IRQ_9		RW	0xA
15:8	RESERVED		R	0x0
7:0	DMA_EDMA_DREQ_8_IRQ_8		RW	0x9

Table 13-413. Register Call Summary for Register CTRL_CORE_DMA_EDMA_DREQ_8_9

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-414. CTRL_CORE_DMA_EDMA_DREQ_10_11

Address Offset	0x0000 0C8C		
Physical Address	0x4A00 2C8C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DMA_EDMA_DREQ_11_IRQ_11								RESERVED								DMA_EDMA_DREQ_10_IRQ_10							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_EDMA_DREQ_11_IRQ_11		RW	0xC
15:8	RESERVED		R	0x0
7:0	DMA_EDMA_DREQ_10_IRQ_10		RW	0xB

Table 13-415. Register Call Summary for Register CTRL_CORE_DMA_EDMA_DREQ_10_11

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-416. CTRL_CORE_DMA_EDMA_DREQ_12_13

Address Offset	0x0000 0C90		
Physical Address	0x4A00 2C90	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DMA_EDMA_DREQ_13_IRQ_13								RESERVED								DMA_EDMA_DREQ_12_IRQ_12							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_EDMA_DREQ_13_IRQ_13		RW	0xE
15:8	RESERVED		R	0x0
7:0	DMA_EDMA_DREQ_12_IRQ_12		RW	0xD

Table 13-417. Register Call Summary for Register CTRL_CORE_DMA_EDMA_DREQ_12_13

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-418. CTRL_CORE_DMA_EDMA_DREQ_14_15

Address Offset	0x0000 0C94		
Physical Address	0x4A00 2C94	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DMA_EDMA_DREQ_15_IRQ_15								RESERVED								DMA_EDMA_DREQ_14_IRQ_14							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_EDMA_DREQ_15_IRQ_15		RW	0x10
15:8	RESERVED		R	0x0
7:0	DMA_EDMA_DREQ_14_IRQ_14		RW	0xF

Table 13-419. Register Call Summary for Register CTRL_CORE_DMA_EDMA_DREQ_14_15

Control Module Register Manual

- CTRL_MODULE_CORE Register Summary: [0]

Table 13-420. CTRL_CORE_DMA_EDMA_DREQ_16_17

Address Offset	0x0000 0C98		
Physical Address	0x4A00 2C98	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DMA_EDMA_DREQ_17_IRQ_17								RESERVED								DMA_EDMA_DREQ_16_IRQ_16							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_EDMA_DREQ_17_IRQ_17		RW	0x12
15:8	RESERVED		R	0x0
7:0	DMA_EDMA_DREQ_16_IRQ_16		RW	0x11

Table 13-421. Register Call Summary for Register CTRL_CORE_DMA_EDMA_DREQ_16_17

Control Module Register Manual

- CTRL_MODULE_CORE Register Summary: [0]

Table 13-422. CTRL_CORE_DMA_EDMA_DREQ_18_19

Address Offset	0x0000 0C9C		
Physical Address	0x4A00 2C9C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DMA_EDMA_DREQ_19_IRQ_19								RESERVED								DMA_EDMA_DREQ_18_IRQ_18							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_EDMA_DREQ_19_IRQ_19		RW	0x14
15:8	RESERVED		R	0x0
7:0	DMA_EDMA_DREQ_18_IRQ_18		RW	0x13

Table 13-423. Register Call Summary for Register CTRL_CORE_DMA_EDMA_DREQ_18_19

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-424. CTRL_CORE_DMA_EDMA_DREQ_20_21

Address Offset	0x0000 0CA0		
Physical Address	0x4A00 2CA0	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DMA_EDMA_DREQ_21_IRQ_21								RESERVED								DMA_EDMA_DREQ_20_IRQ_20							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_EDMA_DREQ_21_IRQ_21		RW	0x16
15:8	RESERVED		R	0x0
7:0	DMA_EDMA_DREQ_20_IRQ_20		RW	0x15

Table 13-425. Register Call Summary for Register CTRL_CORE_DMA_EDMA_DREQ_20_21

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-426. CTRL_CORE_DMA_EDMA_DREQ_22_23

Address Offset	0x0000 0CA4		
Physical Address	0x4A00 2CA4	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DMA_EDMA_DREQ_23_IRQ_23								RESERVED								DMA_EDMA_DREQ_22_IRQ_22							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_EDMA_DREQ_23_IRQ_23		RW	0x18
15:8	RESERVED		R	0x0
7:0	DMA_EDMA_DREQ_22_IRQ_22		RW	0x17

Table 13-427. Register Call Summary for Register CTRL_CORE_DMA_EDMA_DREQ_22_23

Control Module Register Manual

- CTRL_MODULE_CORE Register Summary: [0]

Table 13-428. CTRL_CORE_DMA_EDMA_DREQ_24_25

Address Offset	0x0000 0CA8		
Physical Address	0x4A00 2CA8	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DMA_EDMA_DREQ_25_IRQ_25								RESERVED								DMA_EDMA_DREQ_24_IRQ_24							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_EDMA_DREQ_25_IRQ_25		RW	0x1A
15:8	RESERVED		R	0x0
7:0	DMA_EDMA_DREQ_24_IRQ_24		RW	0x19

Table 13-429. Register Call Summary for Register CTRL_CORE_DMA_EDMA_DREQ_24_25

Control Module Register Manual

- CTRL_MODULE_CORE Register Summary: [0]

Table 13-430. CTRL_CORE_DMA_EDMA_DREQ_26_27

Address Offset	0x0000 0CAC		
Physical Address	0x4A00 2CAC	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DMA_EDMA_DREQ_27_IRQ_27								RESERVED								DMA_EDMA_DREQ_26_IRQ_26							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_EDMA_DREQ_27_IRQ_27		RW	0x1C
15:8	RESERVED		R	0x0
7:0	DMA_EDMA_DREQ_26_IRQ_26		RW	0x1B

Table 13-431. Register Call Summary for Register CTRL_CORE_DMA_EDMA_DREQ_26_27

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-432. CTRL_CORE_DMA_EDMA_DREQ_28_29

Address Offset	0x0000 0CB0		
Physical Address	0x4A00 2CB0	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DMA_EDMA_DREQ_29_IRQ_29								RESERVED								DMA_EDMA_DREQ_28_IRQ_28							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_EDMA_DREQ_29_IRQ_29		RW	0x1E
15:8	RESERVED		R	0x0
7:0	DMA_EDMA_DREQ_28_IRQ_28		RW	0x1D

Table 13-433. Register Call Summary for Register CTRL_CORE_DMA_EDMA_DREQ_28_29

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-434. CTRL_CORE_DMA_EDMA_DREQ_30_31

Address Offset	0x0000 0CB4		
Physical Address	0x4A00 2CB4	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DMA_EDMA_DREQ_31_IRQ_31								RESERVED								DMA_EDMA_DREQ_30_IRQ_30							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_EDMA_DREQ_31_IRQ_31		RW	0x20
15:8	RESERVED		R	0x0
7:0	DMA_EDMA_DREQ_30_IRQ_30		RW	0x1F

Table 13-435. Register Call Summary for Register CTRL_CORE_DMA_EDMA_DREQ_30_31

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-436. CTRL_CORE_DMA_EDMA_DREQ_32_33

Address Offset	0x0000 0CB8		
Physical Address	0x4A00 2CB8	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DMA_EDMA_DREQ_33_IRQ_33								RESERVED								DMA_EDMA_DREQ_32_IRQ_32							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_EDMA_DREQ_33_IRQ_33		RW	0x22
15:8	RESERVED		R	0x0
7:0	DMA_EDMA_DREQ_32_IRQ_32		RW	0x21

Table 13-437. Register Call Summary for Register CTRL_CORE_DMA_EDMA_DREQ_32_33

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-438. CTRL_CORE_DMA_EDMA_DREQ_34_35

Address Offset	0x0000 0CBC		
Physical Address	0x4A00 2CBC	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DMA_EDMA_DREQ_35_IRQ_35								RESERVED								DMA_EDMA_DREQ_34_IRQ_34							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_EDMA_DREQ_35_IRQ_35		RW	0x24
15:8	RESERVED		R	0x0
7:0	DMA_EDMA_DREQ_34_IRQ_34		RW	0x23

Table 13-439. Register Call Summary for Register CTRL_CORE_DMA_EDMA_DREQ_34_35

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-440. CTRL_CORE_DMA_EDMA_DREQ_36_37

Address Offset	0x0000 0CC0		
Physical Address	0x4A00 2CC0	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DMA_EDMA_DREQ_37_IRQ_37								RESERVED								DMA_EDMA_DREQ_36_IRQ_36							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_EDMA_DREQ_37_IRQ_37		RW	0x26
15:8	RESERVED		R	0x0
7:0	DMA_EDMA_DREQ_36_IRQ_36		RW	0x25

Table 13-441. Register Call Summary for Register CTRL_CORE_DMA_EDMA_DREQ_36_37

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-442. CTRL_CORE_DMA_EDMA_DREQ_38_39

Address Offset	0x0000 0CC4		
Physical Address	0x4A00 2CC4	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DMA_EDMA_DREQ_39_IRQ_39								RESERVED								DMA_EDMA_DREQ_38_IRQ_38							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_EDMA_DREQ_39_IRQ_39		RW	0x28
15:8	RESERVED		R	0x0
7:0	DMA_EDMA_DREQ_38_IRQ_38		RW	0x27

Table 13-443. Register Call Summary for Register CTRL_CORE_DMA_EDMA_DREQ_38_39

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-444. CTRL_CORE_DMA_EDMA_DREQ_40_41

Address Offset	0x0000 0CC8		
Physical Address	0x4A00 2CC8	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DMA_EDMA_DREQ_41_IRQ_41								RESERVED								DMA_EDMA_DREQ_40_IRQ_40							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_EDMA_DREQ_41_IRQ_41		RW	0x2A
15:8	RESERVED		R	0x0
7:0	DMA_EDMA_DREQ_40_IRQ_40		RW	0x29

Table 13-445. Register Call Summary for Register CTRL_CORE_DMA_EDMA_DREQ_40_41

Control Module Functional Description
- DMA_CROSSBAR Module Functional Description: [0]

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [1]

Table 13-446. CTRL_CORE_DMA_EDMA_DREQ_42_43

Address Offset	0x0000 0CCC		
Physical Address	0x4A00 2CCC	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DMA_EDMA_DREQ_43_IRQ_43								RESERVED								DMA_EDMA_DREQ_42_IRQ_42							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_EDMA_DREQ_43_IRQ_43		RW	0x2C
15:8	RESERVED		R	0x0
7:0	DMA_EDMA_DREQ_42_IRQ_42		RW	0x2B

Table 13-447. Register Call Summary for Register CTRL_CORE_DMA_EDMA_DREQ_42_43

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-448. CTRL_CORE_DMA_EDMA_DREQ_44_45

Address Offset	0x0000 0CD0		
Physical Address	0x4A00 2CD0	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DMA_EDMA_DREQ_45_IRQ_45								RESERVED								DMA_EDMA_DREQ_44_IRQ_44							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_EDMA_DREQ_45_IRQ_45		RW	0x2E
15:8	RESERVED		R	0x0
7:0	DMA_EDMA_DREQ_44_IRQ_44		RW	0x2D

Table 13-449. Register Call Summary for Register CTRL_CORE_DMA_EDMA_DREQ_44_45

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-450. CTRL_CORE_DMA_EDMA_DREQ_46_47

Address Offset	0x0000 0CD4		
Physical Address	0x4A00 2CD4	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DMA_EDMA_DREQ_47_IRQ_47								RESERVED								DMA_EDMA_DREQ_46_IRQ_46							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_EDMA_DREQ_47_IRQ_47		RW	0x30
15:8	RESERVED		R	0x0
7:0	DMA_EDMA_DREQ_46_IRQ_46		RW	0x2F

Table 13-451. Register Call Summary for Register CTRL_CORE_DMA_EDMA_DREQ_46_47

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-452. CTRL_CORE_DMA_EDMA_DREQ_48_49

Address Offset	0x0000 0CD8		
Physical Address	0x4A00 2CD8	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DMA_EDMA_DREQ_49_IRQ_49								RESERVED								DMA_EDMA_DREQ_48_IRQ_48							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_EDMA_DREQ_49_IRQ_49		RW	0x32
15:8	RESERVED		R	0x0
7:0	DMA_EDMA_DREQ_48_IRQ_48		RW	0x31

Table 13-453. Register Call Summary for Register CTRL_CORE_DMA_EDMA_DREQ_48_49

Control Module Functional Description
- DMA_CROSSBAR Module Functional Description: [0][1]

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [2]

Table 13-454. CTRL_CORE_DMA_EDMA_DREQ_50_51

Address Offset	0x0000 0CDC		
Physical Address	0x4A00 2CDC	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DMA_EDMA_DREQ_51_IRQ_51								RESERVED								DMA_EDMA_DREQ_50_IRQ_50							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_EDMA_DREQ_51_IRQ_51		RW	0x34
15:8	RESERVED		R	0x0
7:0	DMA_EDMA_DREQ_50_IRQ_50		RW	0x33

Table 13-455. Register Call Summary for Register CTRL_CORE_DMA_EDMA_DREQ_50_51

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-456. CTRL_CORE_DMA_EDMA_DREQ_52_53

Address Offset	0x0000 0CE0		
Physical Address	0x4A00 2CE0	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DMA_EDMA_DREQ_53_IRQ_53								RESERVED								DMA_EDMA_DREQ_52_IRQ_52							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_EDMA_DREQ_53_IRQ_53		RW	0x36
15:8	RESERVED		R	0x0
7:0	DMA_EDMA_DREQ_52_IRQ_52		RW	0x35

Table 13-457. Register Call Summary for Register CTRL_CORE_DMA_EDMA_DREQ_52_53

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-458. CTRL_CORE_DMA_EDMA_DREQ_54_55

Address Offset	0x0000 0CE4		
Physical Address	0x4A00 2CE4	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DMA_EDMA_DREQ_55_IRQ_55								RESERVED								DMA_EDMA_DREQ_54_IRQ_54							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_EDMA_DREQ_55_IRQ_55		RW	0x38
15:8	RESERVED		R	0x0
7:0	DMA_EDMA_DREQ_54_IRQ_54		RW	0x37

Table 13-459. Register Call Summary for Register CTRL_CORE_DMA_EDMA_DREQ_54_55

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-460. CTRL_CORE_DMA_EDMA_DREQ_56_57

Address Offset	0x0000 0CE8		
Physical Address	0x4A00 2CE8	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DMA_EDMA_DREQ_57_IRQ_57								RESERVED								DMA_EDMA_DREQ_56_IRQ_56							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_EDMA_DREQ_57_IRQ_57		RW	0x3A
15:8	RESERVED		R	0x0
7:0	DMA_EDMA_DREQ_56_IRQ_56		RW	0x39

Table 13-461. Register Call Summary for Register CTRL_CORE_DMA_EDMA_DREQ_56_57

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-462. CTRL_CORE_DMA_EDMA_DREQ_58_59

Address Offset	0x0000 0CEC		
Physical Address	0x4A00 2CEC	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DMA_EDMA_DREQ_59_IRQ_59								RESERVED								DMA_EDMA_DREQ_58_IRQ_58							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_EDMA_DREQ_59_IRQ_59		RW	0x3C
15:8	RESERVED		R	0x0
7:0	DMA_EDMA_DREQ_58_IRQ_58		RW	0x3B

Table 13-463. Register Call Summary for Register CTRL_CORE_DMA_EDMA_DREQ_58_59

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-464. CTRL_CORE_DMA_EDMA_DREQ_60_61

Address Offset	0x0000 0CF0		
Physical Address	0x4A00 2CF0	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DMA_EDMA_DREQ_61_IRQ_61								RESERVED								DMA_EDMA_DREQ_60_IRQ_60							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_EDMA_DREQ_61_IRQ_61		RW	0x3E
15:8	RESERVED		R	0x0
7:0	DMA_EDMA_DREQ_60_IRQ_60		RW	0x3D

Table 13-465. Register Call Summary for Register CTRL_CORE_DMA_EDMA_DREQ_60_61

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-466. CTRL_CORE_DMA_EDMA_DREQ_62_63

Address Offset	0x0000 0CF4		
Physical Address	0x4A00 2CF4	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DMA_EDMA_DREQ_63_IRQ_63								RESERVED								DMA_EDMA_DREQ_62_IRQ_62							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_EDMA_DREQ_63_IRQ_63		RW	0x40
15:8	RESERVED		R	0x0
7:0	DMA_EDMA_DREQ_62_IRQ_62		RW	0x3F

Table 13-467. Register Call Summary for Register CTRL_CORE_DMA_EDMA_DREQ_62_63

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-468. CTRL_CORE_DMA_DSP1_DREQ_0_1

Address Offset	0x0000 0CF8		
Physical Address	0x4A00 2CF8	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DMA_DSP1_DREQ_1_IRQ_1								RESERVED								DMA_DSP1_DREQ_0_IRQ_0							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_DSP1_DREQ_1_IRQ_1		RW	0x81
15:8	RESERVED		R	0x0
7:0	DMA_DSP1_DREQ_0_IRQ_0		RW	0x80

Table 13-469. Register Call Summary for Register CTRL_CORE_DMA_DSP1_DREQ_0_1

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-470. CTRL_CORE_DMA_DSP1_DREQ_2_3

Address Offset	0x0000 0CFC		
Physical Address	0x4A00 2CFC	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DMA_DSP1_DREQ_3_IRQ_3								RESERVED								DMA_DSP1_DREQ_2_IRQ_2							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_DSP1_DREQ_3_IRQ_3		RW	0x83
15:8	RESERVED		R	0x0
7:0	DMA_DSP1_DREQ_2_IRQ_2		RW	0x82

Table 13-471. Register Call Summary for Register CTRL_CORE_DMA_DSP1_DREQ_2_3

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-472. CTRL_CORE_DMA_DSP1_DREQ_4_5

Address Offset	0x0000 0D00		
Physical Address	0x4A00 2D00	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DMA_DSP1_DREQ_5_IRQ_5								RESERVED								DMA_DSP1_DREQ_4_IRQ_4							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_DSP1_DREQ_5_IRQ_5		RW	0x85
15:8	RESERVED		R	0x0
7:0	DMA_DSP1_DREQ_4_IRQ_4		RW	0x84

Table 13-473. Register Call Summary for Register CTRL_CORE_DMA_DSP1_DREQ_4_5

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-474. CTRL_CORE_DMA_DSP1_DREQ_6_7

Address Offset	0x0000 0D04		
Physical Address	0x4A00 2D04	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DMA_DSP1_DREQ_7_IRQ_7								RESERVED								DMA_DSP1_DREQ_6_IRQ_6							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_DSP1_DREQ_7_IRQ_7		RW	0x87
15:8	RESERVED		R	0x0
7:0	DMA_DSP1_DREQ_6_IRQ_6		RW	0x86

Table 13-475. Register Call Summary for Register CTRL_CORE_DMA_DSP1_DREQ_6_7

Control Module Register Manual

- CTRL_MODULE_CORE Register Summary: [0]

Table 13-476. CTRL_CORE_DMA_DSP1_DREQ_8_9

Address Offset	0x0000 0D08		
Physical Address	0x4A00 2D08	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DMA_DSP1_DREQ_9_IRQ_9								RESERVED								DMA_DSP1_DREQ_8_IRQ_8							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_DSP1_DREQ_9_IRQ_9		RW	0x89
15:8	RESERVED		R	0x0
7:0	DMA_DSP1_DREQ_8_IRQ_8		RW	0x88

Table 13-477. Register Call Summary for Register CTRL_CORE_DMA_DSP1_DREQ_8_9

Control Module Register Manual

- CTRL_MODULE_CORE Register Summary: [0]

Table 13-478. CTRL_CORE_DMA_DSP1_DREQ_10_11

Address Offset	0x0000 0D0C		
Physical Address	0x4A00 2D0C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DMA_DSP1_DREQ_11_IRQ_11								RESERVED								DMA_DSP1_DREQ_10_IRQ_10							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_DSP1_DREQ_11_IRQ_11		RW	0x8B
15:8	RESERVED		R	0x0
7:0	DMA_DSP1_DREQ_10_IRQ_10		RW	0x8A

Table 13-479. Register Call Summary for Register CTRL_CORE_DMA_DSP1_DREQ_10_11

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-480. CTRL_CORE_DMA_DSP1_DREQ_12_13

Address Offset	0x0000 0D10		
Physical Address	0x4A00 2D10	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DMA_DSP1_DREQ_13_IRQ_13								RESERVED								DMA_DSP1_DREQ_12_IRQ_12							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_DSP1_DREQ_13_IRQ_13		RW	0x8D
15:8	RESERVED		R	0x0
7:0	DMA_DSP1_DREQ_12_IRQ_12		RW	0x8C

Table 13-481. Register Call Summary for Register CTRL_CORE_DMA_DSP1_DREQ_12_13

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-482. CTRL_CORE_DMA_DSP1_DREQ_14_15

Address Offset	0x0000 0D14		
Physical Address	0x4A00 2D14	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DMA_DSP1_DREQ_15_IRQ_15								RESERVED								DMA_DSP1_DREQ_14_IRQ_14							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_DSP1_DREQ_15_IRQ_15		RW	0x8F
15:8	RESERVED		R	0x0
7:0	DMA_DSP1_DREQ_14_IRQ_14		RW	0x8E

Table 13-483. Register Call Summary for Register CTRL_CORE_DMA_DSP1_DREQ_14_15

Control Module Register Manual

- CTRL_MODULE_CORE Register Summary: [0]

Table 13-484. CTRL_CORE_DMA_DSP1_DREQ_16_17

Address Offset	0x0000 0D18		
Physical Address	0x4A00 2D18	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DMA_DSP1_DREQ_17_IRQ_17								RESERVED								DMA_DSP1_DREQ_16_IRQ_16							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_DSP1_DREQ_17_IRQ_17		RW	0x9B
15:8	RESERVED		R	0x0
7:0	DMA_DSP1_DREQ_16_IRQ_16		RW	0x9A

Table 13-485. Register Call Summary for Register CTRL_CORE_DMA_DSP1_DREQ_16_17

Control Module Register Manual

- CTRL_MODULE_CORE Register Summary: [0]

Table 13-486. CTRL_CORE_DMA_DSP1_DREQ_18_19

Address Offset	0x0000 0D1C		
Physical Address	0x4A00 2D1C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DMA_DSP1_DREQ_19_IRQ_19								RESERVED								DMA_DSP1_DREQ_18_IRQ_18							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_DSP1_DREQ_19_IRQ_19		RW	0x9D
15:8	RESERVED		R	0x0
7:0	DMA_DSP1_DREQ_18_IRQ_18		RW	0x9C

Table 13-487. Register Call Summary for Register CTRL_CORE_DMA_DSP1_DREQ_18_19

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-488. CTRL_CORE_DMA_DSP2_DREQ_0_1

Address Offset	0x0000 0D20		
Physical Address	0x4A00 2D20	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DMA_DSP2_DREQ_1_IRQ_1								RESERVED								DMA_DSP2_DREQ_0_IRQ_0							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_DSP2_DREQ_1_IRQ_1		RW	0x81
15:8	RESERVED		R	0x0
7:0	DMA_DSP2_DREQ_0_IRQ_0		RW	0x80

Table 13-489. Register Call Summary for Register CTRL_CORE_DMA_DSP2_DREQ_0_1

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-490. CTRL_CORE_DMA_DSP2_DREQ_2_3

Address Offset	0x0000 0D24		
Physical Address	0x4A00 2D24	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DMA_DSP2_DREQ_3_IRQ_3								RESERVED								DMA_DSP2_DREQ_2_IRQ_2							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_DSP2_DREQ_3_IRQ_3		RW	0x83
15:8	RESERVED		R	0x0
7:0	DMA_DSP2_DREQ_2_IRQ_2		RW	0x82

Table 13-491. Register Call Summary for Register CTRL_CORE_DMA_DSP2_DREQ_2_3

Control Module Register Manual

- CTRL_MODULE_CORE Register Summary: [0]

Table 13-492. CTRL_CORE_DMA_DSP2_DREQ_4_5

Address Offset	0x0000 0D28		
Physical Address	0x4A00 2D28	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DMA_DSP2_DREQ_5_IRQ_5								RESERVED								DMA_DSP2_DREQ_4_IRQ_4							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_DSP2_DREQ_5_IRQ_5		RW	0x85
15:8	RESERVED		R	0x0
7:0	DMA_DSP2_DREQ_4_IRQ_4		RW	0x84

Table 13-493. Register Call Summary for Register CTRL_CORE_DMA_DSP2_DREQ_4_5

Control Module Register Manual

- CTRL_MODULE_CORE Register Summary: [0]

Table 13-494. CTRL_CORE_DMA_DSP2_DREQ_6_7

Address Offset	0x0000 0D2C		
Physical Address	0x4A00 2D2C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DMA_DSP2_DREQ_7_IRQ_7								RESERVED								DMA_DSP2_DREQ_6_IRQ_6							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_DSP2_DREQ_7_IRQ_7		RW	0x87
15:8	RESERVED		R	0x0
7:0	DMA_DSP2_DREQ_6_IRQ_6		RW	0x86

Table 13-495. Register Call Summary for Register CTRL_CORE_DMA_DSP2_DREQ_6_7

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-496. CTRL_CORE_DMA_DSP2_DREQ_8_9

Address Offset	0x0000 0D30		
Physical Address	0x4A00 2D30	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DMA_DSP2_DREQ_9_IRQ_9								RESERVED								DMA_DSP2_DREQ_8_IRQ_8							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_DSP2_DREQ_9_IRQ_9		RW	0x89
15:8	RESERVED		R	0x0
7:0	DMA_DSP2_DREQ_8_IRQ_8		RW	0x88

Table 13-497. Register Call Summary for Register CTRL_CORE_DMA_DSP2_DREQ_8_9

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-498. CTRL_CORE_DMA_DSP2_DREQ_10_11

Address Offset	0x0000 0D34		
Physical Address	0x4A00 2D34	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DMA_DSP2_DREQ_11_IRQ_11								RESERVED								DMA_DSP2_DREQ_10_IRQ_10							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_DSP2_DREQ_11_IRQ_11		RW	0x8B
15:8	RESERVED		R	0x0
7:0	DMA_DSP2_DREQ_10_IRQ_10		RW	0x8A

Table 13-499. Register Call Summary for Register CTRL_CORE_DMA_DSP2_DREQ_10_11

Control Module Register Manual

- CTRL_MODULE_CORE Register Summary: [0]

Table 13-500. CTRL_CORE_DMA_DSP2_DREQ_12_13

Address Offset	0x0000 0D38		
Physical Address	0x4A00 2D38	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DMA_DSP2_DREQ_13_IRQ_13								RESERVED								DMA_DSP2_DREQ_12_IRQ_12							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_DSP2_DREQ_13_IRQ_13		RW	0x8D
15:8	RESERVED		R	0x0
7:0	DMA_DSP2_DREQ_12_IRQ_12		RW	0x8C

Table 13-501. Register Call Summary for Register CTRL_CORE_DMA_DSP2_DREQ_12_13

Control Module Register Manual

- CTRL_MODULE_CORE Register Summary: [0]

Table 13-502. CTRL_CORE_DMA_DSP2_DREQ_14_15

Address Offset	0x0000 0D3C		
Physical Address	0x4A00 2D3C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DMA_DSP2_DREQ_15_IRQ_15								RESERVED								DMA_DSP2_DREQ_14_IRQ_14							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_DSP2_DREQ_15_IRQ_15		RW	0x8F
15:8	RESERVED		R	0x0
7:0	DMA_DSP2_DREQ_14_IRQ_14		RW	0x8E

Table 13-503. Register Call Summary for Register CTRL_CORE_DMA_DSP2_DREQ_14_15

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-504. CTRL_CORE_DMA_DSP2_DREQ_16_17

Address Offset	0x0000 0D40		
Physical Address	0x4A00 2D40	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DMA_DSP2_DREQ_17_IRQ_17								RESERVED								DMA_DSP2_DREQ_16_IRQ_16							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_DSP2_DREQ_17_IRQ_17		RW	0x9B
15:8	RESERVED		R	0x0
7:0	DMA_DSP2_DREQ_16_IRQ_16		RW	0x9A

Table 13-505. Register Call Summary for Register CTRL_CORE_DMA_DSP2_DREQ_16_17

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-506. CTRL_CORE_DMA_DSP2_DREQ_18_19

Address Offset	0x0000 0D44		
Physical Address	0x4A00 2D44	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DMA_DSP2_DREQ_19_IRQ_19								RESERVED								DMA_DSP2_DREQ_18_IRQ_18							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:16	DMA_DSP2_DREQ_19_IRQ_19		RW	0x9D
15:8	RESERVED		R	0x0
7:0	DMA_DSP2_DREQ_18_IRQ_18		RW	0x9C

Table 13-507. Register Call Summary for Register CTRL_CORE_DMA_DSP2_DREQ_18_19

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-508. CTRL_CORE_ESM_GROUP1_0

Address Offset	0x0000 0D54		
Physical Address	0x4A00 2D54	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								ESM_GROUP1_0_IRQ_0							

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8:0	ESM_GROUP1_0_IRQ_0	Note: ESM is not supported on the DRA78x family of devices.	RW	0xea

Table 13-509. Register Call Summary for Register CTRL_CORE_ESM_GROUP1_0

Control Module Functional Description
- IRQ_CROSSBAR Module Functional Description: [0][1]

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [2]

Table 13-510. CTRL_CORE_ESM_GROUP1_1

Address Offset	0x0000 0D58		
Physical Address	0x4A00 2D58	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								ESM_GROUP1_1_IRQ_1							

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8:0	ESM_GROUP1_1_IRQ_1	Note: ESM is not supported on the DRA78x family of devices.	RW	0x19d

Table 13-511. Register Call Summary for Register CTRL_CORE_ESM_GROUP1_1

Control Module Functional Description • IRQ_CROSSBAR Module Functional Description: [0]
Control Module Register Manual • CTRL_MODULE_CORE Register Summary: [1]

Table 13-512. CTRL_CORE_ESM_GROUP1_2

Address Offset	0x0000 0D5C		
Physical Address	0x4A00 2D5C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								ESM_GROUP1_2_IRQ_2							

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8:0	ESM_GROUP1_2_IRQ_2	Note: ESM is not supported on the DRA78x family of devices.	RW	0x19f

Table 13-513. Register Call Summary for Register CTRL_CORE_ESM_GROUP1_2

Control Module Functional Description • IRQ_CROSSBAR Module Functional Description: [0]
Control Module Register Manual • CTRL_MODULE_CORE Register Summary: [1]

Table 13-514. CTRL_CORE_ESM_GROUP1_3

Address Offset	0x0000 0D60		
Physical Address	0x4A00 2D60	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								ESM_GROUP1_3_DMA_3							

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x0
7:0	ESM_GROUP1_3_DMA_3	Note: ESM is not supported on the DRA78x family of devices.	RW	0x4

Table 13-515. Register Call Summary for Register CTRL_CORE_ESM_GROUP1_3

Control Module Functional Description • DMA_CROSSBAR Module Functional Description: [0]
Control Module Register Manual • CTRL_MODULE_CORE Register Summary: [1]

Table 13-516. CTRL_CORE_CAMERRX_CONTROL

Address Offset	0x0000 0E08		
Physical Address	0x4A00 2E08	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							CSIPORTA_MODE	CSIPORTA_LANEENABLE					CSIPORTA_CAMMODE		CSIPORTA_CTRLCLKEN

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	CSIPORTA_MODE	porta camera mode	RW	0x0
7:3	CSIPORTA_LANEENABLE	porta camera lane enable 0x0: Lane module disabled 0x1: Lane module enabled	RW	0x0
2:1	CSIPORTA_CAMMODE	porta camera mode control 0x0: DPHY mode 0x1: Data/Strobe Transmission format 0x2: Data/Clock Transmission format 0x3: GPI mode	RW	0x3
0	CSIPORTA_CTRLCLKEN	porta camera clock enable control 0x0: Disable for CTRLCLK 0x1: Active high enable for CTRLCLK	RW	0x0

Table 13-517. Register Call Summary for Register CTRL_CORE_CAMERRX_CONTROL

Control Module Register Manual • CTRL_MODULE_CORE Register Summary: [0]

Table 13-518. CTRL_CORE_CONTROL_DDRCACH1_0

Address Offset	0x0000 0E30		
Physical Address	0x4A00 2E30	**Instance**	CTRL_MODULE_CORE
Description	ddrcaCH1 control		
Type	RW		

31 30 29	28 27 26	25 24	23 22 21	20 19 18	17 16	15 14 13	12 11 10	9 8	7 6	5 4	3 2	1 0
DDRCH1_PART0_I	DDRCH1_PART0_SR	DDRCH1_PART0_WD	DDRCH1_PART5A_I	DDRCH1_PART5A_SR	DDRCH1_PART5A_WD	DDRCH1_PART5B_I	DDRCH1_PART5B_SR	DDRCH1_PART5B_WD	DDRCH1_PART6_I	DDRCH1_PART6_SR	DDRCH1_PART6_WD	

Bits	Field Name	Description	Type	Reset
31:29	DDRCH1_PART0_I	PART0 Impedence control I[2:0] 0x0: Imp80 0x1: Imp60 0x2: Imp48 0x3: Imp40 0x4: Imp34 0x5: Reserved 0x6: Reserved 0x7: Reserved	RW	0x2
28:26	DDRCH1_PART0_SR	PART0 Slew Rate control SR[2:0]. All 8 values are valid. 0x0: Fastest .... 0x7: Slowest	RW	0x2
25:24	DDRCH1_PART0_WD	PART0 Weak driver control WD[1:0] -For single-ended operation: 0x0: Pull logic is disabled 0x1: Pull-up selected 0x2: Pull-down selected 0x3: Maintain the previous output value -For differential pair operation: 0x0: Pull logic is disabled 0x1: Pull-up selected for padp, pull-down selected for padn 0x2: Pull-down selected for padp, pull-up selected for padn 0x3: Maintain the previous output value	RW	0x2
23:21	DDRCH1_PART5A_I	PART5A Impedence control I[2:0] 0x0: Imp80 0x1: Imp60 0x2: Imp48 0x3: Imp40 0x4: Imp34 0x5: Reserved 0x6: Reserved 0x7: Reserved	RW	0x2
20:18	DDRCH1_PART5A_SR	PART5A Slew Rate control SR[2:0]. All 8 values are valid. 0x0: Fastest .... 0x7: Slowest	RW	0x2

Bits	Field Name	Description	Type	Reset
17:16	DDRCH1_PART5A_WD	PART5A Weak driver control WD[1:0] -For single-ended operation: 0x0: Pull logic is disabled 0x1: Pull-up selected 0x2: Pull-down selected 0x3: Maintain the previous output value -For differential pair operation: 0x0: Pull logic is disabled 0x1: Pull-up selected for padp, pull-down selected for padn 0x2: Pull-down selected for padp, pull-up selected for padn 0x3: Maintain the previous output value	RW	0x2
15:13	DDRCH1_PART5B_I	PART5B Impedence control I[2:0] 0x0: Imp80 0x1: Imp60 0x2: Imp48 0x3: Imp40 0x4: Imp34 0x5: Reserved 0x6: Reserved 0x7: Reserved	RW	0x2
12:10	DDRCH1_PART5B_SR	PART5B Slew Rate control SR[2:0]. All 8 values are valid. 0x0: Fastest .... 0x7: Slowest	RW	0x2
9:8	DDRCH1_PART5B_WD	PART5B Weak driver control WD[1:0] -For single-ended operation: 0x0: Pull logic is disabled 0x1: Pull-up selected 0x2: Pull-down selected 0x3: Maintain the previous output value -For differential pair operation: 0x0: Pull logic is disabled 0x1: Pull-up selected for padp, pull-down selected for padn 0x2: Pull-down selected for padp, pull-up selected for padn 0x3: Maintain the previous output value	RW	0x2
7:5	DDRCH1_PART6_I	PART6 Impedence control I[2:0] 0x0: Imp80 0x1: Imp60 0x2: Imp48 0x3: Imp40 0x4: Imp34 0x5: Reserved 0x6: Reserved 0x7: Reserved	RW	0x2
4:2	DDRCH1_PART6_SR	PART6 Slew Rate control SR[2:0]. All 8 values are valid. 0x0: Fastest .... 0x7: Slowest	RW	0x2

Bits	Field Name	Description	Type	Reset
1:0	DDRCH1_PART6_WD	PART6 Weak driver control WD[1:0] -For single-ended operation: 0x0: Pull logic is disabled 0x1: Pull-up selected 0x2: Pull-down selected 0x3: Maintain the previous output value -For differential pair operation: 0x0: Pull logic is disabled 0x1: Pull-up selected for padp, pull-down selected for padn 0x2: Pull-down selected for padp, pull-up selected for padn 0x3: Maintain the previous output value	RW	0x2

Table 13-519. Register Call Summary for Register CTRL_CORE_CONTROL_DDRCACH1_0

Control Module Functional Description
- Software Controls for the LPDDR2/DDR2/DDR3 I/O Cells: [0][1][2][3][4][5][6][7][8][9][10][11][12]

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [13]

Table 13-520. CTRL_CORE_CONTROL_DDRCH1_0

Address Offset	0x0000 0E38		
Physical Address	0x4A00 2E38	**Instance**	CTRL_MODULE_CORE
Description	DDRCH1 control 0		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DDRCH1_PART1A_I			DDRCH1_PART1A_SR			DDRCH1_PART1A_WD		DDRCH1_PART1B_I			DDRCH1_PART1B_SR			DDRCH1_PART1B_WD		DDRCH1_PART2A_I			DDRCH1_PART2A_SR			DDRCH1_PART2A_WD		DDRCH1_PART2B_I			DDRCH1_PART2B_SR			DDRCH1_PART2B_WD	

Bits	Field Name	Description	Type	Reset
31:29	DDRCH1_PART1A_I	PART1A Impedence control I[2:0] 0x0: Imp80 0x1: Imp60 0x2: Imp48 0x3: Imp40 0x4: Imp34 0x5: Reserved 0x6: Reserved 0x7: Reserved	RW	0x2
28:26	DDRCH1_PART1A_SR	PART1A Slew Rate control SR[2:0]. All 8 values are valid. 0x0: Fastest .... 0x7: Slowest	RW	0x2

Bits	Field Name	Description	Type	Reset
25:24	DDRCH1_PART1A_WD	PART1A Weak driver control WD[1:0] -For single-ended operation: 0x0: Pull logic is disabled 0x1: Pull-up selected 0x2: Pull-down selected 0x3: Maintain the previous output value -For differential pair operation: 0x0: Pull logic is disabled 0x1: Pull-up selected for padp, pull-down selected for padn 0x2: Pull-down selected for padp, pull-up selected for padn 0x3: Maintain the previous output value	RW	0x2
23:21	DDRCH1_PART1B_I	PART1B Impedence control I[2:0] 0x0: Imp80 0x1: Imp60 0x2: Imp48 0x3: Imp40 0x4: Imp34 0x5: Reserved 0x6: Reserved 0x7: Reserved	RW	0x2
20:18	DDRCH1_PART1B_SR	PART1B Slew Rate control SR[2:0]. All 8 values are valid. 0x0: Fastest .... 0x7: Slowest	RW	0x2
17:16	DDRCH1_PART1B_WD	PART1B Weak driver control WD[1:0] -For single-ended operation: 0x0: Pull logic is disabled 0x1: Pull-up selected 0x2: Pull-down selected 0x3: Maintain the previous output value -For differential pair operation: 0x0: Pull logic is disabled 0x1: Pull-up selected for padp, pull-down selected for padn 0x2: Pull-down selected for padp, pull-up selected for padn 0x3: Maintain the previous output value	RW	0x2
15:13	DDRCH1_PART2A_I	PART2A Impedence control I[2:0] 0x0: Imp80 0x1: Imp60 0x2: Imp48 0x3: Imp40 0x4: Imp34 0x5: Reserved 0x6: Reserved 0x7: Reserved	RW	0x2
12:10	DDRCH1_PART2A_SR	PART2A Slew Rate control SR[2:0]. All 8 values are valid. 0x0: Fastest .... 0x7: Slowest	RW	0x2

Bits	Field Name	Description	Type	Reset
9:8	DDRCH1_PART2A_WD	PART2A Weak driver control WD[1:0] -For single-ended operation: 0x0: Pull logic is disabled 0x1: Pull-up selected 0x2: Pull-down selected 0x3: Maintain the previous output value -For differential pair operation: 0x0: Pull logic is disabled 0x1: Pull-up selected for padp, pull-down selected for padn 0x2: Pull-down selected for padp, pull-up selected for padn 0x3: Maintain the previous output value	RW	0x2
7:5	DDRCH1_PART2B_I	PART2B Impedence control I[2:0] 0x0: Imp80 0x1: Imp60 0x2: Imp48 0x3: Imp40 0x4: Imp34 0x5: Reserved 0x6: Reserved 0x7: Reserved	RW	0x2
4:2	DDRCH1_PART2B_SR	PART2B Slew Rate control SR[2:0]. All 8 values are valid. 0x0: Fastest .... 0x7: Slowest	RW	0x2
1:0	DDRCH1_PART2B_WD	PART2B Weak driver control WD[1:0] -For single-ended operation: 0x0: Pull logic is disabled 0x1: Pull-up selected 0x2: Pull-down selected 0x3: Maintain the previous output value -For differential pair operation: 0x0: Pull logic is disabled 0x1: Pull-up selected for padp, pull-down selected for padn 0x2: Pull-down selected for padp, pull-up selected for padn 0x3: Maintain the previous output value	RW	0x2

Table 13-521. Register Call Summary for Register CTRL_CORE_CONTROL_DDRCH1_0

Control Module Functional Description • Software Controls for the LPDDR2/DDR2/DDR3 I/O Cells: [0][1][2][3][4][5][6][7][8][9][10][11]
Control Module Register Manual • CTRL_MODULE_CORE Register Summary: [12]

Table 13-522. CTRL_CORE_CONTROL_DDRCH1_1

Address Offset	0x0000 0E3C		
Physical Address	0x4A00 2E3C	**Instance**	CTRL_MODULE_CORE
Description	DDRCH1 control 1		
Type	RW		

31 30 29	28 27 26	25 24	23 22 21	20 19 18	17 16	15 14 13	12 11 10	9 8	7 6 5	4 3 2	1 0
DDRCH1_PART3A_I	DDRCH1_PART3A_SR	DDRCH1_PART3A_WD	DDRCH1_PART3B_I	DDRCH1_PART3B_SR	DDRCH1_PART3B_WD	DDRCH1_PART4A_I	DDRCH1_PART4A_SR	DDRCH1_PART4A_WD	DDRCH1_PART4B_I	DDRCH1_PART4B_SR	DDRCH1_PART4B_WD

Bits	Field Name	Description	Type	Reset
31:29	DDRCH1_PART3A_I	PART3A Impedence control I[2:0] 0x0: Imp80 0x1: Imp60 0x2: Imp48 0x3: Imp40 0x4: Imp34 0x5: Reserved 0x6: Reserved 0x7: Reserved	RW	0x2
28:26	DDRCH1_PART3A_SR	PART3A Slew Rate control SR[2:0]. All 8 values are valid. 0x0: Fastest .... 0x7: Slowest	RW	0x2
25:24	DDRCH1_PART3A_WD	PART3A Weak driver control WD[1:0] -For single-ended operation: 0x0: Pull logic is disabled 0x1: Pull-up selected 0x2: Pull-down selected 0x3: Maintain the previous output value -For differential pair operation: 0x0: Pull logic is disabled 0x1: Pull-up selected for padp, pull-down selected for padn 0x2: Pull-down selected for padp, pull-up selected for padn 0x3: Maintain the previous output value	RW	0x2
23:21	DDRCH1_PART3B_I	PART3B Impedence control I[2:0] 0x0: Imp80 0x1: Imp60 0x2: Imp48 0x3: Imp40 0x4: Imp34 0x5: Reserved 0x6: Reserved 0x7: Reserved	RW	0x2
20:18	DDRCH1_PART3B_SR	PART3B Slew Rate control SR[2:0]. All 8 values are valid. 0x0: Fastest .... 0x7: Slowest	RW	0x2

Bits	Field Name	Description	Type	Reset
17:16	DDRCH1_PART3B_WD	PART3B Weak driver control WD[1:0] -For single-ended operation: 0x0: Pull logic is disabled 0x1: Pull-up selected 0x2: Pull-down selected 0x3: Maintain the previous output value -For differential pair operation: 0x0: Pull logic is disabled 0x1: Pull-up selected for padp, pull-down selected for padn 0x2: Pull-down selected for padp, pull-up selected for padn 0x3: Maintain the previous output value	RW	0x2
15:13	DDRCH1_PART4A_I	PART4A Impedence control I[2:0] 0x0: Imp80 0x1: Imp60 0x2: Imp48 0x3: Imp40 0x4: Imp34 0x5: Reserved 0x6: Reserved 0x7: Reserved	RW	0x2
12:10	DDRCH1_PART4A_SR	PART4A Slew Rate control SR[2:0]. All 8 values are valid. 0x0: Fastest .... 0x7: Slowest	RW	0x2
9:8	DDRCH1_PART4A_WD	PART4A Weak driver control WD[1:0] -For single-ended operation: 0x0: Pull logic is disabled 0x1: Pull-up selected 0x2: Pull-down selected 0x3: Maintain the previous output value -For differential pair operation: 0x0: Pull logic is disabled 0x1: Pull-up selected for padp, pull-down selected for padn 0x2: Pull-down selected for padp, pull-up selected for padn 0x3: Maintain the previous output value	RW	0x2
7:5	DDRCH1_PART4B_I	PART4B Impedence control I[2:0] 0x0: Imp80 0x1: Imp60 0x2: Imp48 0x3: Imp40 0x4: Imp34 0x5: Reserved 0x6: Reserved 0x7: Reserved	RW	0x2
4:2	DDRCH1_PART4B_SR	PART4B Slew Rate control SR[2:0]. All 8 values are valid. 0x0: Fastest .... 0x7: Slowest	RW	0x2

Bits	Field Name	Description	Type	Reset
1:0	DDRCH1_PART4B_WD	PART4B Weak driver control WD[1:0] -For single-ended operation: 0x0: Pull logic is disabled 0x1: Pull-up selected 0x2: Pull-down selected 0x3: Maintain the previous output value -For differential pair operation: 0x0: Pull logic is disabled 0x1: Pull-up selected for padp, pull-down selected for padn 0x2: Pull-down selected for padp, pull-up selected for padn 0x3: Maintain the previous output value	RW	0x2

Table 13-523. Register Call Summary for Register CTRL_CORE_CONTROL_DDRCH1_1

Control Module Functional Description
- Software Controls for the LPDDR2/DDR2/DDR3 I/O Cells: [0][1][2][3][4][5][6][7][8][9][10][11]

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [12]

Table 13-524. CTRL_CORE_CONTROL_DDRCH1_2

Address Offset	0x0000 0E48		
Physical Address	0x4A00 2E48	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31 30 29 28 27 26 25 24	23 22 21	20 19 18	17 16	15 14 13	12 11 10	9 8	7 6 5 4 3 2 1 0
RESERVED	DDRCH1_PART7A_I	DDRCH1_PART7A_SR	DDRCH1_PART7A_WD	DDRCH1_PART7B_I	DDRCH1_PART7B_SR	DDRCH1_PART7B_WD	RESERVED

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:21	DDRCH1_PART7A_I	PART7A Impedence control I[2:0] 0x0: Imp80 0x1: Imp60 0x2: Imp48 0x3: Imp40 0x4: Imp34 0x5: Reserved 0x6: Reserved 0x7: Reserved	RW	0x2
20:18	DDRCH1_PART7A_SR	PART7A Slew Rate control SR[2:0]. All 8 values are valid. 0x0: Fastest .... 0x7: Slowest	RW	0x2

Bits	Field Name	Description	Type	Reset
17:16	DDRCH1_PART7A_WD	PART7A Weak driver control WD[1:0] -For single-ended operation: 0x0: Pull logic is disabled 0x1: Pull-up selected 0x2: Pull-down selected 0x3: Maintain the previous output value -For differential pair operation: 0x0: Pull logic is disabled 0x1: Pull-up selected for padp, pull-down selected for padn 0x2: Pull-down selected for padp, pull-up selected for padn 0x3: Maintain the previous output value	RW	0x2
15:13	DDRCH1_PART7B_I	PART7B Impedence control I[2:0] 0x0: Imp80 0x1: Imp60 0x2: Imp48 0x3: Imp40 0x4: Imp34 0x5: Reserved 0x6: Reserved 0x7: Reserved	RW	0x2
12:10	DDRCH1_PART7B_SR	PART7B Slew Rate control SR[2:0]. All 8 values are valid. 0x0: Fastest .... 0x7: Slowest	RW	0x2
9:8	DDRCH1_PART7B_WD	PART7B Weak driver control WD[1:0] -For single-ended operation: 0x0: Pull logic is disabled 0x1: Pull-up selected 0x2: Pull-down selected 0x3: Maintain the previous output value -For differential pair operation: 0x0: Pull logic is disabled 0x1: Pull-up selected for padp, pull-down selected for padn 0x2: Pull-down selected for padp, pull-up selected for padn 0x3: Maintain the previous output value	RW	0x2
7:0	RESERVED		R	0x0

Table 13-525. Register Call Summary for Register CTRL_CORE_CONTROL_DDRCH1_2

Control Module Functional Description • Software Controls for the LPDDR2/DDR2/DDR3 I/O Cells: [0][1][2][3][4][5]
Control Module Register Manual • CTRL_MODULE_CORE Register Summary: [6]

Table 13-526. CTRL_CORE_CONTROL_DDRIO_0

Address Offset	0x0000 0E50		
Physical Address	0x4A00 2E50	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												DDRCH1_VREF_DQ0_INT_CCAP0	DDRCH1_VREF_DQ0_INT_CCAP1	DDRCH1_VREF_DQ0_INT_TAP0	DDRCH1_VREF_DQ0_INT_TAP1	DDRCH1_VREF_DQ0_INT_EN	DDRCH1_VREF_DQ1_INT_CCAP0	DDRCH1_VREF_DQ1_INT_CCAP1	DDRCH1_VREF_DQ1_INT_TAP0	DDRCH1_VREF_DQ1_INT_TAP1	DDRCH1_VREF_DQ1_INT_EN	RESERVED								

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	DDRCH1_VREF_DQ0_INT_CCAP0	Selection for coupling cap connection 0x0: Disabled 0x1: Enabled	RW	0x1
18	DDRCH1_VREF_DQ0_INT_CCAP1	Selection for coupling cap connection 0x0: Disabled 0x1: Enabled	RW	0x0
17	DDRCH1_VREF_DQ0_INT_TAP0	Selection for internal reference voltage drive 0x0: Disabled 0x1: Enabled	RW	0x0
16	DDRCH1_VREF_DQ0_INT_TAP1	Selection for internal reference voltage drive 0x0: Disabled 0x1: Enabled	RW	0x1
15	DDRCH1_VREF_DQ0_INT_EN	Enable 0x0: Disabled 0x1: Enabled	RW	0x1
14	DDRCH1_VREF_DQ1_INT_CCAP0	Selection for coupling cap connection 0x0: Disabled 0x1: Enabled	RW	0x1
13	DDRCH1_VREF_DQ1_INT_CCAP1	Selection for coupling cap connection 0x0: Disabled 0x1: Enabled	RW	0x0
12	DDRCH1_VREF_DQ1_INT_TAP0	Selection for internal reference voltage drive 0x0: Disabled 0x1: Enabled	RW	0x0
11	DDRCH1_VREF_DQ1_INT_TAP1	Selection for internal reference voltage drive 0x0: Disabled 0x1: Enabled	RW	0x1
10	DDRCH1_VREF_DQ1_INT_EN	Enable 0x0: Disabled 0x1: Enabled	RW	0x1
9:0	RESERVED		R	0x260

Table 13-527. Register Call Summary for Register CTRL_CORE_CONTROL_DDRIO_0

Control Module Functional Description

- Reference Voltage for the Device LPDDR2/DDR2/DDR3 Receivers: [0][1][2][3][4][5][6][7][8][9][10][11][12]

Control Module Register Manual

- CTRL_MODULE_CORE Register Summary: [13]

Table 13-528. CTRL_CORE_HWOBS_FINAL_MUX_SEL

Address Offset	0x0000 1000		
Physical Address	0x4A00 3000	**Instance**	CTRL_MODULE_CORE
Description	This register controls the final multiplexor which selects whether CORE or WKUP observable signals are mapped to the final observability bus, that is obs[31:0] bus.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SELECT																															

Bits	Field Name	Description	Type	Reset
31:0	SELECT	Each bit is associated with one observability signal line. Bit 31 selects between CORE and WKUP signal for obs31 line and bit 0 does the same for obs0 line. The same logic applies for all other bits. 0x0: WKUP signal selected. 0x1: CORE signal selected.	RW	0x0

Table 13-529. Register Call Summary for Register CTRL_CORE_HWOBS_FINAL_MUX_SEL

Control Module Functional Description

- Hardware Observability Related Registers: [0][1]

Control Module Register Manual

- CTRL_MODULE_CORE Register Summary: [2]
- CTRL_MODULE_CORE Register Description: [3][4][5][6][7][8][9][10][11][12][13][14][15][16][17][18][19][20][21][22][23][24][25][26][27][28][29][30][31][32][33][34]
- CTRL_MODULE_WKUP Register Description: [35][36][37][38][39][40][41][42][43][44][45][46][47][48][49][50][51][52][53][54][55][56][57][58][59][60][61][62][63][64][65][66]

Table 13-530. CTRL_CORE_CONF_DEBUG_SEL_TST_0

Address Offset	0x0000 100C		
Physical Address	0x4A00 300C	**Instance**	CTRL_MODULE_CORE
Description	This register is used to select an observable signal for CORE observability line 0.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												MODE			

Bits	Field Name	Description	Type	Reset
31:6	RESERVED		R	0x0
5:0	MODE	Selects one of the following signals to be available on CORE observability line 0. This signal can then be mapped to obs0 line through CTRL_CORE_HWOBS_FINAL_MUX_SEL[0] bit. Values not shown are reserved. 0x7: hwobs_debug_dss(0) 0xB: hwobs_dsshdcpducatidisable 0xE: hwobs_emif1_prcm_deviceoff_wkup_core_rstactst 0x12: hwobs_debug_debugss(0) 0x15: hwobs_emif1_phy_reg_status_phy_ctrl_dll_lock(0) 0x16: hwobs_emif1_phy_reg_rdlvl_dqs_ratio_byte0_rank0(0) 0x17: hwobs_emif1_phy_reg_rdlvl_dqs_ratio_byte0_rank1(0) 0x18: hwobs_emif1_phy_reg_status_phy_ctrl_dll_lock(1) 0x1E: hwobs_eve1_eve_dbgout[23] 0x22: hwobs_dsp1_Turing_hw_dbgout [31] 0x23: hwobs_dsp2_Turing_hw_dbgout [31] 0x26: hwobs_vip1_outmux_testport[0] 0x2A: hwobs_debug_prm(0) 0x2B: hwobs_debug_cm_core_aon(0) 0x2C: hwobs_int_iss(0) 0x2D: hwobs_int_iss(32) 0x2E: func_spare_out[0] 0x2F: dft_test_pad_i[0] 0x30: dft_test_pad_o[0]	RW	0x0

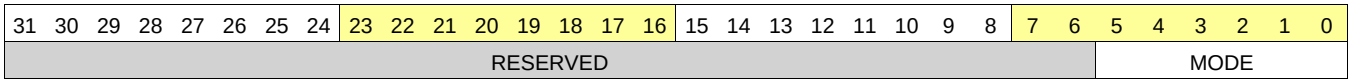
Table 13-531. Register Call Summary for Register CTRL_CORE_CONF_DEBUG_SEL_TST_0

Control Module Functional Description
- Hardware Observability Related Registers: [0][1]

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [2]

Table 13-532. CTRL_CORE_CONF_DEBUG_SEL_TST_1

Address Offset	0x0000 1010		
Physical Address	0x4A00 3010	**Instance**	CTRL_MODULE_CORE
Description	This register is used to select an observable signal for CORE observability line 1.		
Type	RW		



Bits	Field Name	Description	Type	Reset
31:6	RESERVED		R	0x0
5:0	MODE	Selects one of the following signals to be available on CORE observability line 1. This signal can then be mapped to obs1 line through CTRL_CORE_HWOBS_FINAL_MUX_SEL[1] bit. Values not shown are reserved. 0x7: hwobs_debug_dss(1) 0x9: hwobs_dspdpll_freqlock 0xB: hwobs_dsshdcpsecurity_violation 0xC: hwobs_ddrdpll_freqlock 0xE: hwobs_emif1_pwr_sidlereq 0x12: hwobs_debug_debugss(1) 0x15: hwobs_emif1_phy_reg_status_phy_ctrl_dll_lock(1) 0x16: hwobs_emif1_phy_reg_rdlvl_dqs_ratio_byte0_rank0(1) 0x17: hwobs_emif1_phy_reg_rdlvl_dqs_ratio_byte0_rank1(1) 0x18: hwobs_emif1_phy_reg_status_phy_ctrl_dll_lock(0) 0x1E: hwobs_eve1_eve_dbgout[22] 0x22: hwobs_dsp1_Turing_hw_dbgout [30] 0x23: hwobs_dsp2_Turing_hw_dbgout [30] 0x26: hwobs_vip1_outmux_testport[1] 0x2A: hwobs_debug_prm(1) 0x2B: hwobs_debug_cm_core_aon(1) 0x2C: hwobs_int_iss(1) 0x2D: hwobs_int_iss(33) 0x2E: func_spare_out[1] 0x2F: dft_test_pad_i[1] 0x30: dft_test_pad_o[1]	RW	0x0

Table 13-533. Register Call Summary for Register CTRL_CORE_CONF_DEBUG_SEL_TST_1

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-534. CTRL_CORE_CONF_DEBUG_SEL_TST_2

Address Offset	0x0000 1014		
Physical Address	0x4A00 3014	**Instance**	CTRL_MODULE_CORE
Description	This register is used to select an observable signal for CORE observability line 2.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								MODE							

Bits	Field Name	Description	Type	Reset
31:6	RESERVED		R	0x0
5:0	MODE	Selects one of the following signals to be available on CORE observability line 2. This signal can then be mapped to obs2 line through CTRL_CORE_HWOBS_FINAL_MUX_SEL[2] bit. Values not shown are reserved. 0x7: hwobs_debug_dss(2) 0x9: hwobs_dspdpll_tinitz 0xB: hwobs_dsshdmiphypwrcmdtxon 0xC: hwobs_ddrdpll_tinitz 0xE: hwobs_emif1_pwr_sidleack(1) 0x12: hwobs_debug_debugss(2) 0x15: hwobs_emif1_phy_reg_status_dll_lock_byte0 0x16: hwobs_emif1_phy_reg_rdlvl_dqs_ratio_byte0_rank0(2) 0x17: hwobs_emif1_phy_reg_rdlvl_dqs_ratio_byte0_rank1(2) 0x18: hwobs_emif1_phy_reg_status_phy_ctrl_dll_slave_value(0) 0x1E: hwobs_eve1_eve_dbgout[21] 0x22: hwobs_dsp1_Turing_hw_dbgout [29] 0x23: hwobs_dsp2_Turing_hw_dbgout [29] 0x26: hwobs_vip1_outmux_testport[2] 0x2A: hwobs_debug_prm(2) 0x2B: hwobs_debug_cm_core_aon(2) 0x2C: hwobs_int_iss(2) 0x2D: hwobs_int_iss(34) 0x2E: func_spare_out[2] 0x2F: dft_test_pad_i[2] 0x30: dft_test_pad_o[2]	RW	0x0

Table 13-535. Register Call Summary for Register CTRL_CORE_CONF_DEBUG_SEL_TST_2

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-536. CTRL_CORE_CONF_DEBUG_SEL_TST_3

Address Offset	0x0000 1018		
Physical Address	0x4A00 3018	**Instance**	CTRL_MODULE_CORE
Description	This register is used to select an observable signal for CORE observability line 3.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								MODE							

Bits	Field Name	Description	Type	Reset
31:6	RESERVED		R	0x0
5:0	MODE	Selects one of the following signals to be available on CORE observability line 3. This signal can then be mapped to obs3 line through CTRL_CORE_HWOBS_FINAL_MUX_SEL[3] bit. Values not shown are reserved. 0x1: hwobs_coredpll_freqlock 0x3: hwobs_dsi1adpll_freqlock 0x7: hwobs_debug_dss(3) 0x9: hwobs_dspdpll_phaselock 0xB: hwobs_dsshdmiphypwrcmdoff 0xC: hwobs_ddrdpll_phaselock 0xE: hwobs_emif1_pwr_sidleack(0) 0x12: hwobs_debug_debugss(3) 0x15: hwobs_emif1_phy_reg_rdfifo_rdptr_byte0(0) 0x16: hwobs_emif1_phy_reg_rdlvl_dqs_ratio_byte0_rank0(3) 0x17: hwobs_emif1_phy_reg_rdlvl_dqs_ratio_byte0_rank1(3) 0x18: hwobs_emif1_phy_reg_status_phy_ctrl_dll_slave_value(1) 0x1E: hwobs_eve1_eve_dbgout[20] 0x22: hwobs_dsp1_Turing_hw_dbgout [28] 0x23: hwobs_dsp2_Turing_hw_dbgout [28] 0x26: hwobs_vip1_outmux_testport[3] 0x2A: hwobs_debug_prm(3) 0x2B: hwobs_debug_cm_core_aon(3) 0x2C: hwobs_int_iss(3) 0x2D: hwobs_int_iss(35) 0x2E: func_spare_out[3] 0x2F: dft_test_pad_i[3] 0x30: dft_test_pad_o[3]	RW	0x0

Table 13-537. Register Call Summary for Register CTRL_CORE_CONF_DEBUG_SEL_TST_3

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-538. CTRL_CORE_CONF_DEBUG_SEL_TST_4

Address Offset	0x0000 101C		
Physical Address	0x4A00 301C	**Instance**	CTRL_MODULE_CORE
Description	This register is used to select an observable signal for CORE observability line 4.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								MODE							

Bits	Field Name	Description	Type	Reset
31:6	RESERVED		R	0x0
5:0	MODE	Selects one of the following signals to be available on CORE observability line 4. This signal can then be mapped to obs4 line through CTRL_CORE_HWOBS_FINAL_MUX_SEL[4] bit. Values not shown are reserved. 0x1: hwobs_coredpll_tinitz 0x3: hwobs_dsi1adpll_tinitz 0x7: hwobs_debug_dss(4) 0x9: hwobs_dspdpll_tenable 0xB: hwobs_dsshdmiphypwrack 0xC: hwobs_ddrdpll_tenable 0xE: hwobs_emif1_pwr_fclken 0x12: hwobs_debug_debugss(4) 0x15: hwobs_emif1_phy_reg_rdfifo_rdptr_byte0(1) 0x16: hwobs_emif1_phy_reg_rdlvl_dqs_ratio_byte0_rank0(4) 0x17: hwobs_emif1_phy_reg_rdlvl_dqs_ratio_byte0_rank1(4) 0x18: hwobs_emif1_phy_reg_status_phy_ctrl_dll_slave_value(2) 0x1E: hwobs_eve1_eve_dbgout[19] 0x22: hwobs_dsp1_Turing_hw_dbgout [27] 0x23: hwobs_dsp2_Turing_hw_dbgout [27] 0x26: hwobs_vip1_outmux_testport[4] 0x2A: hwobs_debug_prm(4) 0x2B: hwobs_debug_cm_core_aon(4) 0x2C: hwobs_int_iss(4) 0x2D: hwobs_int_iss(36) 0x2E: func_spare_out[4] 0x2F: dft_test_pad_i[4] 0x30: dft_test_pad_o[4]	RW	0x0

Table 13-539. Register Call Summary for Register CTRL_CORE_CONF_DEBUG_SEL_TST_4

Control Module Register Manual

- CTRL_MODULE_CORE Register Summary: [0]

Table 13-540. CTRL_CORE_CONF_DEBUG_SEL_TST_5

Address Offset	0x0000 1020		
Physical Address	0x4A00 3020	**Instance**	CTRL_MODULE_CORE
Description	This register is used to select an observable signal for CORE observability line 5.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								MODE							

Bits	Field Name	Description	Type	Reset
31:6	RESERVED		R	0x0
5:0	MODE	Selects one of the following signals to be available on CORE observability line 5. This signal can then be mapped to obs5 line through CTRL_CORE_HWOBS_FINAL_MUX_SEL[5] bit. Values not shown are reserved. 0x1: hwobs_coredpll_phaselock 0x3: hwobs_dsi1adpll_phaselock 0x7: hwobs_debug_dss(5) 0x9: hwobs_dspdpll_tenablediv 0xB: hwobs_dsshdmihpd 0xC: hwobs_ddrdpll_tenablediv 0xE: hwobs_emif1_sys_err_intr_req 0x12: hwobs_debug_debugss(5) 0x15: hwobs_emif1_phy_reg_rdfifo_rdptr_byte0(2) 0x16: hwobs_emif1_phy_reg_rdlvl_dqs_ratio_byte0_rank0(5) 0x17: hwobs_emif1_phy_reg_rdlvl_dqs_ratio_byte0_rank1(5) 0x18: hwobs_emif1_phy_reg_status_phy_ctrl_dll_slave_value(3) 0x1E: hwobs_eve1_eve_dbgout[18] 0x22: hwobs_dsp1_Turing_hw_dbgout [26] 0x23: hwobs_dsp2_Turing_hw_dbgout [26] 0x26: hwobs_vip1_outmux_testport[5] 0x2A: hwobs_debug_prm(5) 0x2B: hwobs_debug_cm_core_aon(5) 0x2C: hwobs_int_iss(5) 0x2D: hwobs_int_iss(37) 0x2E: func_spare_out[5] 0x2F: dft_test_pad_i[5] 0x30: dft_test_pad_o[5]	RW	0x0

Table 13-541. Register Call Summary for Register CTRL_CORE_CONF_DEBUG_SEL_TST_5

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-542. CTRL_CORE_CONF_DEBUG_SEL_TST_6

Address Offset	0x0000 1024		
Physical Address	0x4A00 3024	**Instance**	CTRL_MODULE_CORE
Description	This register is used to select an observable signal for CORE observability line 6.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								MODE							

Bits	Field Name	Description	Type	Reset
31:6	RESERVED		R	0x0
5:0	MODE	Selects one of the following signals to be available on CORE observability line 6. This signal can then be mapped to obs6 line through CTRL_CORE_HWOBS_FINAL_MUX_SEL[6] bit. Values not shown are reserved. 0x1: hwobs_coredpll_tenable 0x3: hwobs_dsi1adpll_tenable 0x7: hwobs_debug_dss(6) 0x9: hwobs_dspdpll_bypassack 0xB: hwobs_dssdsi1ascpenphy 0xC: hwobs_ddrdpll_bypassack 0xE: hwobs_emif1_sys_err_intr_pend 0x12: hwobs_debug_debugss(6) 0x15: hwobs_emif1_phy_reg_rdfifo_wrptr_byte0(0) 0x16: hwobs_emif1_phy_reg_rdlvl_dqs_ratio_byte0_rank0(6) 0x17: hwobs_emif1_phy_reg_rdlvl_dqs_ratio_byte0_rank1(6) 0x18: hwobs_emif1_phy_reg_status_phy_ctrl_dll_slave_value(4) 0x1E: hwobs_eve1_eve_dbgout[17] 0x22: hwobs_dsp1_Turing_hw_dbgout [25] 0x23: hwobs_dsp2_Turing_hw_dbgout [25] 0x26: hwobs_vip1_outmux_testport[6] 0x2A: hwobs_debug_prm(6) 0x2B: hwobs_debug_cm_core_aon(6) 0x2C: hwobs_int_iss(6) 0x2D: hwobs_int_iss(38) 0x2E: func_spare_out[6] 0x2F: dft_test_pad_i[6] 0x30: dft_test_pad_o[6]	RW	0x0

Table 13-543. Register Call Summary for Register CTRL_CORE_CONF_DEBUG_SEL_TST_6

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-544. CTRL_CORE_CONF_DEBUG_SEL_TST_7

Address Offset	0x0000 1028		
Physical Address	0x4A00 3028	**Instance**	CTRL_MODULE_CORE
Description	This register is used to select an observable signal for CORE observability line 7.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								MODE							

Bits	Field Name	Description	Type	Reset
31:6	RESERVED		R	0x0
5:0	MODE	Selects one of the following signals to be available on CORE observability line 7. This signal can then be mapped to obs7 line through CTRL_CORE_HWOBS_FINAL_MUX_SEL[7] bit. Values not shown are reserved. 0x1: hwobs_coredpll_tenablediv 0x3: hwobs_dsi1adpll_tenablediv 0x7: hwobs_debug_dss(7) 0x9: hwobs_dspdpll_idle 0xB: hwobs_dssdsi1ascpaddr(3) 0xC: hwobs_ddrdpll_idle 0xE: hwobs_emif1_ll_err_intr_req 0x12: hwobs_debug_debugss(7) 0x15: hwobs_emif1_phy_reg_rdfifo_wrptr_byte0(1) 0x16: hwobs_emif1_phy_reg_rdlvl_dqs_ratio_byte0_rank0(7) 0x17: hwobs_emif1_phy_reg_rdlvl_dqs_ratio_byte0_rank1(7) 0x18: hwobs_emif1_phy_reg_status_phy_ctrl_dll_slave_value(5) 0x1E: hwobs_eve1_eve_dbgout[16] 0x22: hwobs_dsp1_Turing_hw_dbgout [24] 0x23: hwobs_dsp2_Turing_hw_dbgout [24] 0x26: hwobs_vip1_outmux_testport[7] 0x2A: hwobs_debug_prm(7) 0x2B: hwobs_debug_cm_core_aon(7) 0x2C: hwobs_int_iss(7) 0x2D: hwobs_int_iss(39) 0x2E: func_spare_out[7] 0x2F: dft_test_pad_i[7] 0x30: dft_test_pad_o[7]	RW	0x0

Table 13-545. Register Call Summary for Register CTRL_CORE_CONF_DEBUG_SEL_TST_7

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-546. CTRL_CORE_CONF_DEBUG_SEL_TST_8

Address Offset	0x0000 102C		
Physical Address	0x4A00 302C	**Instance**	CTRL_MODULE_CORE
Description	This register is used to select an observable signal for CORE observability line 8.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								MODE							

Bits	Field Name	Description	Type	Reset
31:6	RESERVED		R	0x0
5:0	MODE	Selects one of the following signals to be available on CORE observability line 8. This signal can then be mapped to obs8 line through CTRL_CORE_HWOBS_FINAL_MUX_SEL[8] bit. Values not shown are reserved. 0x1: hwobs_coredpll_bypassack 0x3: hwobs_dsi1adpll_bypassack 0x7: hwobs_debug_dss(8) 0xB: hwobs_dssdsi1ascpaddr(2) 0xC: hwobs_dsi1a_plllock 0xE: hwobs_emif1_ll_err_intr_pend 0x12: hwobs_debug_debugss(8) 0x15: hwobs_emif1_phy_reg_rdfifo_wrptr_byte0(2) 0x16: hwobs_emif1_phy_reg_rdlvl_dqs_ratio_byte0_rank0(8) 0x17: hwobs_emif1_phy_reg_rdlvl_dqs_ratio_byte0_rank1(8) 0x18: hwobs_emif1_phy_reg_status_phy_ctrl_dll_slave_value(6) 0x1E: hwobs_eve1_eve_dbgout[15] 0x22: hwobs_dsp1_Turing_hw_dbgout [23] 0x23: hwobs_dsp2_Turing_hw_dbgout [23] 0x26: hwobs_vip1_outmux_testport[8] 0x2A: hwobs_debug_prm(8) 0x2B: hwobs_debug_cm_core_aon(8) 0x2C: hwobs_int_iss(8) 0x2D: hwobs_int_iss(40) 0x2E: func_spare_out[8] 0x2F: dft_test_pad_i[8] 0x30: dft_test_pad_o[8]	RW	0x0

Table 13-547. Register Call Summary for Register CTRL_CORE_CONF_DEBUG_SEL_TST_8

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-548. CTRL_CORE_CONF_DEBUG_SEL_TST_9

Address Offset	0x0000 1030		
Physical Address	0x4A00 3030	**Instance**	CTRL_MODULE_CORE
Description	This register is used to select an observable signal for CORE observability line 9.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								MODE							

Bits	Field Name	Description	Type	Reset
31:6	RESERVED		R	0x0
5:0	MODE	Selects one of the following signals to be available on CORE observability line 9. This signal can then be mapped to obs9 line through CTRL_CORE_HWOBS_FINAL_MUX_SEL[9] bit. Values not shown are reserved. 0x1: hwobs_coredpll_idle 0x3: hwobs_dsi1adpll_idle 0x7: hwobs_debug_dss(9) 0x9: hwobs_gmacdpll_freqlock 0xB: hwobs_dssdsi1ascpaddr(1) 0xC: hwobs_dsi1a_pllrecal 0xE: hwobs_emif1_ret_powerRet 0x11: hwobs_mmc_adpidle 0x12: hwobs_debug_debugss(9) 0x15: hwobs_emif1_phy_reg_gate_level_fsm_byte0(0) 0x16: hwobs_emif1_phy_reg_rdlvl_dqs_ratio_byte0_rank0(9) 0x17: hwobs_emif1_phy_reg_rdlvl_dqs_ratio_byte0_rank1(9) 0x18: hwobs_emif1_phy_reg_status_phy_ctrl_dll_slave_value(7) 0x1E: hwobs_eve1_eve_dbgout[14] 0x22: hwobs_dsp1_Turing_hw_dbgout [22] 0x23: hwobs_dsp2_Turing_hw_dbgout [22] 0x26: hwobs_vip1_outmux_testport[9] 0x2A: hwobs_debug_prm(9) 0x2B: hwobs_debug_cm_core_aon(9) 0x2C: hwobs_int_iss(9) 0x2D: hwobs_int_iss(41) 0x2E: func_spare_out[9] 0x2F: dft_test_pad_i[9] 0x30: dft_test_pad_o[9]	RW	0x0

Table 13-549. Register Call Summary for Register CTRL_CORE_CONF_DEBUG_SEL_TST_9

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-550. CTRL_CORE_CONF_DEBUG_SEL_TST_10

Address Offset	0x0000 1034		
Physical Address	0x4A00 3034	**Instance**	CTRL_MODULE_CORE
Description	This register is used to select an observable signal for CORE observability line 10.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								MODE							

Bits	Field Name	Description	Type	Reset
31:6	RESERVED		R	0x0
5:0	MODE	Selects one of the following signals to be available on CORE observability line 10. This signal can then be mapped to obs10 line through CTRL_CORE_HWOBS_FINAL_MUX_SEL[10] bit. Values not shown are reserved. 0x1: hwobs_emif1_prcm_deviceoff_wkup_core_rstactst 0x7: hwobs_debug_dss(10) 0x9: hwobs_gmacdpll_tinitz 0xB: hwobs_dssdsi1ascpaddr(0) 0xC: hwobs_dsi1a_stopclock 0xE: hwobs_emif1_phy_sdramclkstop_cmd(1) 0x11: hwobs_mmc_adpdat1paden 0x12: hwobs_debug_debugss(10) 0x15: hwobs_emif1_phy_reg_gate_level_fsm_byte0(1) 0x16: hwobs_emif1_phy_reg_rdlvl_fifowein_ratio_byte0_rank0(0) 0x17: hwobs_emif1_phy_reg_rdlvl_fifowein_ratio_byte0_rank1(0) 0x18: hwobs_emif1_phy_reg_status_phy_ctrl_dll_slave_value(8) 0x1E: hwobs_eve1_eve_dbgout[13] 0x22: hwobs_dsp1_Turing_hw_dbgout [21] 0x23: hwobs_dsp2_Turing_hw_dbgout [21] 0x26: hwobs_vip1_outmux_testport[10] 0x2A: hwobs_debug_prm(10) 0x2B: hwobs_debug_cm_core_aon(10) 0x2C: hwobs_int_iss(10) 0x2D: hwobs_int_iss(42) 0x2E: func_spare_out[10] 0x2F: dft_test_pad_i[10] 0x30: dft_test_pad_o[10]	RW	0x0

Table 13-551. Register Call Summary for Register CTRL_CORE_CONF_DEBUG_SEL_TST_10

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-552. CTRL_CORE_CONF_DEBUG_SEL_TST_11

Address Offset	0x0000 1038		
Physical Address	0x4A00 3038	**Instance**	CTRL_MODULE_CORE
Description	This register is used to select an observable signal for CORE observability line 11.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								MODE							

Bits	Field Name	Description	Type	Reset
31:6	RESERVED		R	0x0
5:0	MODE	Selects one of the following signals to be available on CORE observability line 11. This signal can then be mapped to obs11 line through CTRL_CORE_HWOBS_FINAL_MUX_SEL[11] bit. Values not shown are reserved. 0x1: hwobs_emif1_pwr_sidlereq 0x7: hwobs_debug_dss(11) 0x9: hwobs_gmacdpll_phaselock 0xB: hwobs_dssdsi1ascpcmd(1) 0xC: hwobs_dsi1a_stopclockackz 0xE: hwobs_emif1_phy_sdramclkstop_cmd(0) 0x11: hwobs_mmc_ocpl4idlereq 0x12: hwobs_debug_debugss(11) 0x15: hwobs_emif1_phy_reg_gate_level_fsm_byte0(2) 0x16: hwobs_emif1_phy_reg_rdlvl_fifowein_ratio_byte0_rank0(1) 0x17: hwobs_emif1_phy_reg_rdlvl_fifowein_ratio_byte0_rank1(1) 0x18: hwobs_emif1_phy_reg_status_phy_ctrl_dll_slave_value(9) 0x1E: hwobs_eve1_eve_dbgout[12] 0x22: hwobs_dsp1_Turing_hw_dbgout [20] 0x23: hwobs_dsp2_Turing_hw_dbgout [20] 0x26: hwobs_vip1_outmux_testport[11] 0x2A: hwobs_debug_prm(11) 0x2B: hwobs_debug_cm_core_aon(11) 0x2C: hwobs_int_iss(11) 0x2D: hwobs_int_iss(43) 0x2E: func_spare_out[11] 0x2F: dft_test_pad_i[11] 0x30: dft_test_pad_o[11]	RW	0x0

Table 13-553. Register Call Summary for Register CTRL_CORE_CONF_DEBUG_SEL_TST_11

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-554. CTRL_CORE_CONF_DEBUG_SEL_TST_12

Address Offset	0x0000 103C		
Physical Address	0x4A00 303C	**Instance**	CTRL_MODULE_CORE
Description	This register is used to select an observable signal for CORE observability line 12.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								MODE							

Bits	Field Name	Description	Type	Reset
31:6	RESERVED		R	0x0
5:0	MODE	Selects one of the following signals to be available on CORE observability line 12. This signal can then be mapped to obs12 line through CTRL_CORE_HWOBS_FINAL_MUX_SEL[12] bit. Values not shown are reserved. 0x1: hwobs_emif1_pwr_sidleack(1) 0x2: hwobs_coredivider_clkout3 0x7: hwobs_debug_dss(12) 0x9: hwobs_gmacdpll_tenable 0xB: hwobs_dssdsi1ascpcmd(0) 0xC: hwobs_dsi1a_dispcupdatesync 0x11: hwobs_mmc_ocpl3mwait 0x12: hwobs_debug_debugss(12) 0x15: hwobs_emif1_phy_reg_rd_level_fsm_byte0(0) 0x16: hwobs_emif1_phy_reg_rdlvl_fifowein_ratio_byte0_rank0(2) 0x17: hwobs_emif1_phy_reg_rdlvl_fifowein_ratio_byte0_rank1(2) 0x18: hwobs_emif1_phy_reg_status_phy_ctrl_dll_slave_value(10) 0x1E: hwobs_eve1_eve_dbgout[11] 0x22: hwobs_dsp1_Turing_hw_dbgout [19] 0x23: hwobs_dsp2_Turing_hw_dbgout [19] 0x26: hwobs_vip1_outmux_testport[12] 0x2A: hwobs_debug_prm(12) 0x2B: hwobs_debug_cm_core_aon(12) 0x2C: hwobs_int_iss(12) 0x2D: hwobs_int_iss(44) 0x2E: func_spare_out[12] 0x2F: dft_test_pad_i[12] 0x30: dft_test_pad_o[12]	RW	0x0

Table 13-555. Register Call Summary for Register CTRL_CORE_CONF_DEBUG_SEL_TST_12

Control Module Register Manual

- CTRL_MODULE_CORE Register Summary: [0]

Table 13-556. CTRL_CORE_CONF_DEBUG_SEL_TST_13

Address Offset	0x0000 1040		
Physical Address	0x4A00 3040	**Instance**	CTRL_MODULE_CORE
Description	This register is used to select an observable signal for CORE observability line 13.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								MODE							

Bits	Field Name	Description	Type	Reset
31:6	RESERVED		R	0x0
5:0	MODE	Selects one of the following signals to be available on CORE observability line 13. This signal can then be mapped to obs13 line through CTRL_CORE_HWOBS_FINAL_MUX_SEL[13] bit. Values not shown are reserved. 0x1: hwobs_emif1_pwr_sidleack(0) 0x2: hwobs_perdivider_clkout4 0x7: hwobs_debug_dss(13) 0x9: hwobs_gmacdpll_tenablediv 0xB: hwobs_dssdsi1cscpenphy 0xC: hwobs_dsi1a_scpenpllctrl 0x11: hwobs_mmc_pirffret 0x12: hwobs_debug_debugss(13) 0x15: hwobs_emif1_phy_reg_rd_level_fsm_byte0(1) 0x16: hwobs_emif1_phy_reg_rdlvl_fifowein_ratio_byte0_rank0(3) 0x17: hwobs_emif1_phy_reg_rdlvl_fifowein_ratio_byte0_rank1(3) 0x18: hwobs_emif1_phy_reg_status_phy_ctrl_dll_slave_value(11) 0x1E: hwobs_eve1_eve_dbgout[10] 0x22: hwobs_dsp1_Turing_hw_dbgout [18] 0x23: hwobs_dsp2_Turing_hw_dbgout [18] 0x26: hwobs_vip1_outmux_testport[13] 0x2A: hwobs_debug_prm(13) 0x2B: hwobs_debug_cm_core_aon(13) 0x2C: hwobs_int_iss(13) 0x2D: hwobs_int_iss(45) 0x2E: func_spare_out[13] 0x2F: dft_test_pad_i[13] 0x30: dft_test_pad_o[13]	RW	0x0

Table 13-557. Register Call Summary for Register CTRL_CORE_CONF_DEBUG_SEL_TST_13

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-558. CTRL_CORE_CONF_DEBUG_SEL_TST_14

Address Offset	0x0000 1044		
Physical Address	0x4A00 3044	**Instance**	CTRL_MODULE_CORE
Description	This register is used to select an observable signal for CORE observability line 14.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								MODE							

Bits	Field Name	Description	Type	Reset
31:6	RESERVED		R	0x0
5:0	MODE	Selects one of the following signals to be available on CORE observability line 14. This signal can then be mapped to obs14 line through CTRL_CORE_HWOBS_FINAL_MUX_SEL[14] bit. Values not shown are reserved. 0x1: hwobs_emif1_pwr_fclken 0x7: hwobs_debug_dss(14) 0x9: hwobs_gmacdpll_bypassack 0xB: hwobs_dssdsi1cscpaddr(3) 0xC: hwobs_dsi1a_scpcmd(1) 0x11: hwobs_mmc_ocpl4sidleacko(1) 0x12: hwobs_debug_debugss(14) 0x15: hwobs_emif1_phy_reg_rd_level_fsm_byte0(2) 0x16: hwobs_emif1_phy_reg_rdlvl_fifowein_ratio_byte0_rank0(4) 0x17: hwobs_emif1_phy_reg_rdlvl_fifowein_ratio_byte0_rank1(4) 0x18: hwobs_emif1_phy_reg_status_phy_ctrl_dll_slave_value(12) 0x1E: hwobs_eve1_eve_dbgout[9] 0x22: hwobs_dsp1_Turing_hw_dbgout [17] 0x23: hwobs_dsp2_Turing_hw_dbgout [17] 0x26: hwobs_vip1_outmux_testport[14] 0x2A: hwobs_debug_prm(14) 0x2B: hwobs_debug_cm_core_aon(14) 0x2C: hwobs_int_iss(14) 0x2D: hwobs_int_iss(46) 0x2E: func_spare_out[14] 0x2F: dft_test_pad_i[14] 0x30: dft_test_pad_o[14]	RW	0x0

Table 13-559. Register Call Summary for Register CTRL_CORE_CONF_DEBUG_SEL_TST_14

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-560. CTRL_CORE_CONF_DEBUG_SEL_TST_15

Address Offset	0x0000 1048		
Physical Address	0x4A00 3048	**Instance**	CTRL_MODULE_CORE
Description	This register is used to select an observable signal for CORE observability line 15.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								MODE							

Bits	Field Name	Description	Type	Reset
31:6	RESERVED		R	0x0
5:0	MODE	Selects one of the following signals to be available on CORE observability line 15. This signal can then be mapped to obs15 line through CTRL_CORE_HWOBS_FINAL_MUX_SEL[15] bit. Values not shown are reserved. 0x1: hwobs_emif1_sys_err_intr_req 0x7: hwobs_debug_dss(15) 0x9: hwobs_gmacdpll_idle 0xB: hwobs_dssdsi1cscpaddr(2) 0xC: hwobs_dsi1a_scpcmd(0) 0x11: hwobs_mmc_ocpl4sidleacko(0) 0x12: hwobs_debug_debugss(15) 0x15: hwobs_emif1_phy_reg_rd_level_fsm_byte0(3) 0x16: hwobs_emif1_phy_reg_rdlvl_fifowein_ratio_byte0_rank0(5) 0x17: hwobs_emif1_phy_reg_rdlvl_fifowein_ratio_byte0_rank1(5) 0x18: hwobs_emif1_phy_reg_status_phy_ctrl_dll_slave_value(13) 0x1E: hwobs_eve1_eve_dbgout[8] 0x22: hwobs_dsp1_Turing_hw_dbgout [16] 0x23: hwobs_dsp2_Turing_hw_dbgout [16] 0x26: hwobs_vip1_outmux_testport[15] 0x2A: hwobs_debug_prm(15) 0x2B: hwobs_debug_cm_core_aon(15) 0x2C: hwobs_int_iss(15) 0x2D: hwobs_int_iss(47) 0x2E: func_spare_out[15] 0x2F: dft_test_pad_i[15] 0x30: dft_test_pad_o[15]	RW	0x0

Table 13-561. Register Call Summary for Register CTRL_CORE_CONF_DEBUG_SEL_TST_15

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-562. CTRL_CORE_CONF_DEBUG_SEL_TST_16

Address Offset	0x0000 104C		
Physical Address	0x4A00 304C	**Instance**	CTRL_MODULE_CORE
Description	This register is used to select an observable signal for CORE observability line 16.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								MODE							

Bits	Field Name	Description	Type	Reset
31:6	RESERVED		R	0x0
5:0	MODE	Selects one of the following signals to be available on CORE observability line 16. This signal can then be mapped to obs16 line through CTRL_CORE_HWOBS_FINAL_MUX_SEL[16] bit. Values not shown are reserved. 0x1: hwobs_emif1_sys_err_intr_pend 0x7: hwobs_debug_dss(16) 0xB: hwobs_dssdsi1cscpaddr(1) 0xC: hwobs_dsi1a_scpaddr(3) 0x11: hwobs_mmc_ocpl3mstandbyo 0x12: hwobs_debug_debugss(16) 0x15: hwobs_emif1_phy_reg_wr_level_fsm_byte0(0) 0x16: hwobs_emif1_phy_reg_rdlvl_fifowein_ratio_byte0_rank0(6) 0x17: hwobs_emif1_phy_reg_rdlvl_fifowein_ratio_byte0_rank1(6) 0x18: hwobs_emif1_phy_reg_status_phy_ctrl_dll_slave_value(14) 0x1E: hwobs_eve1_eve_dbgout[7] 0x22: hwobs_dsp1_Turing_hw_dbgout [15] 0x23: hwobs_dsp2_Turing_hw_dbgout [15] 0x26: hwobs_vip1_outmux_testport[16] 0x2A: hwobs_debug_prm(16) 0x2B: hwobs_debug_cm_core_aon(16) 0x2C: hwobs_int_iss(16) 0x2D: hwobs_int_iss(48) 0x2E: func_spare_out[16] 0x2F: dft_test_pad_i[16] 0x30: dft_test_pad_o[16]	RW	0x0

Table 13-563. Register Call Summary for Register CTRL_CORE_CONF_DEBUG_SEL_TST_16

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-564. CTRL_CORE_CONF_DEBUG_SEL_TST_17

Address Offset	0x0000 1050		
Physical Address	0x4A00 3050	**Instance**	CTRL_MODULE_CORE
Description	This register is used to select an observable signal for CORE observability line 17.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								MODE							

Bits	Field Name	Description	Type	Reset
31:6	RESERVED		R	0x0
5:0	MODE	Selects one of the following signals to be available on CORE observability line 17. This signal can then be mapped to obs17 line through CTRL_CORE_HWOBS_FINAL_MUX_SEL[17] bit. Values not shown are reserved. 0x1: hwobs_emif1_ll_err_intr_req 0x7: hwobs_debug_dss(17) 0xB: hwobs_dssdsi1cscpaddr(0) 0xC: hwobs_dsi1a_scpaddr(2) 0x11: hwobs_mmc_swakeup 0x12: hwobs_debug_debugss(17) 0x15: hwobs_emif1_phy_reg_wr_level_fsm_byte0(1) 0x16: hwobs_emif1_phy_reg_rdlvl_fifowein_ratio_byte0_rank0(7) 0x17: hwobs_emif1_phy_reg_rdlvl_fifowein_ratio_byte0_rank1(7) 0x18: hwobs_emif1_phy_reg_status_phy_ctrl_dll_slave_value(15) 0x1E: hwobs_eve1_eve_dbgout[6] 0x22: hwobs_dsp1_Turing_hw_dbgout [14] 0x23: hwobs_dsp2_Turing_hw_dbgout [14] 0x26: hwobs_vip1_outmux_testport[17] 0x2A: hwobs_debug_prm(17) 0x2B: hwobs_debug_cm_core_aon(17) 0x2C: hwobs_int_iss(17) 0x2D: hwobs_int_iss(49) 0x2E: func_spare_out[17] 0x2F: dft_test_pad_i[17] 0x30: dft_test_pad_o[17]	RW	0x0

Table 13-565. Register Call Summary for Register CTRL_CORE_CONF_DEBUG_SEL_TST_17

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-566. CTRL_CORE_CONF_DEBUG_SEL_TST_18

Address Offset	0x0000 1054		
Physical Address	0x4A00 3054	**Instance**	CTRL_MODULE_CORE
Description	This register is used to select an observable signal for CORE observability line 18.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								MODE							

Bits	Field Name	Description	Type	Reset
31:6	RESERVED		R	0x0
5:0	MODE	Selects one of the following signals to be available on CORE observability line 18. This signal can then be mapped to obs18 line through CTRL_CORE_HWOBS_FINAL_MUX_SEL[18] bit. Values not shown are reserved. 0x1: hwobs_emif1_ll_err_intr_pend 0x7: hwobs_debug_dss(18) 0xB: hwobs_dssdsi1cscpcmd(1) 0xC: hwobs_dsi1a_scpaddr(1) 0x12: hwobs_debug_debugss(18) 0x15: hwobs_emif1_phy_reg_wr_level_fsm_byte0(2) 0x16: hwobs_emif1_phy_reg_rdlvl_fifowein_ratio_byte0_rank0(8) 0x17: hwobs_emif1_phy_reg_rdlvl_fifowein_ratio_byte0_rank1(8) 0x18: hwobs_emif1_phy_reg_status_phy_ctrl_dll_slave_value(16) 0x1E: hwobs_eve1_eve_dbgout[5] 0x22: hwobs_dsp1_Turing_hw_dbgout [13] 0x23: hwobs_dsp2_Turing_hw_dbgout [13] 0x26: hwobs_vip1_outmux_testport[18] 0x2A: hwobs_debug_prm(18) 0x2B: hwobs_debug_cm_core_aon(18) 0x2C: hwobs_int_iss(18) 0x2D: hwobs_int_iss(50) 0x2E: func_spare_out[18] 0x2F: dft_test_pad_i[18] 0x30: dft_test_pad_o[18]	RW	0x0

Table 13-567. Register Call Summary for Register CTRL_CORE_CONF_DEBUG_SEL_TST_18

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-568. CTRL_CORE_CONF_DEBUG_SEL_TST_19

Address Offset	0x0000 1058		
Physical Address	0x4A00 3058	**Instance**	CTRL_MODULE_CORE
Description	This register is used to select an observable signal for CORE observability line 19.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								MODE							

Bits	Field Name	Description	Type	Reset
31:6	RESERVED		R	0x0
5:0	MODE	Selects one of the following signals to be available on CORE observability line 19. This signal can then be mapped to obs19 line through CTRL_CORE_HWOBS_FINAL_MUX_SEL[19] bit. Values not shown are reserved. 0x1: hwobs_emif1_ret_powerRet 0x2: hwobs_perdpll_freqlock 0x7: hwobs_debug_dss(19) 0xB: hwobs_dssdsi1cscpcmd(0) 0xC: hwobs_dsi1a_scpaddr(0) 0x12: hwobs_debug_debugss(19) 0x15: hwobs_emif1_phy_reg_status_dll_slave_value_byte0(0) 0x16: hwobs_emif1_phy_reg_rdlvl_fifowein_ratio_byte0_rank0(9) 0x17: hwobs_emif1_phy_reg_rdlvl_fifowein_ratio_byte0_rank1(9) 0x18: hwobs_emif1_phy_reg_status_phy_ctrl_dll_slave_value(17) 0x1E: hwobs_eve1_eve_dbgout[4] 0x22: hwobs_dsp1_Turing_hw_dbgout [12] 0x23: hwobs_dsp2_Turing_hw_dbgout [12] 0x26: hwobs_vip1_outmux_testport[19] 0x2A: hwobs_debug_prm(19) 0x2B: hwobs_debug_cm_core_aon(19) 0x2C: hwobs_int_iss(19) 0x2D: hwobs_int_iss(51) 0x2E: func_spare_out[19] 0x2F: dft_test_pad_i[19] 0x30: dft_test_pad_o[19]	RW	0x0

Table 13-569. Register Call Summary for Register CTRL_CORE_CONF_DEBUG_SEL_TST_19

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-570. CTRL_CORE_CONF_DEBUG_SEL_TST_20

Address Offset	0x0000 105C		
Physical Address	0x4A00 305C	**Instance**	CTRL_MODULE_CORE
Description	This register is used to select an observable signal for CORE observability line 20.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								MODE							

Bits	Field Name	Description	Type	Reset
31:6	RESERVED		R	0x0
5:0	MODE	Selects one of the following signals to be available on CORE observability line 20. This signal can then be mapped to obs20 line through CTRL_CORE_HWOBS_FINAL_MUX_SEL[20] bit. Values not shown are reserved. 0x1: hwobs_emif1_phy_sdramclkstop_cmd(0) 0x2: hwobs_perdpll_tinitz 0x7: hwobs_debug_dss(20) 0xB: hwobs_dsshdmiscpaddr(3) 0xC: hwobs_dsi1a_scpout 0x12: hwobs_debug_debugss(20) 0x15: hwobs_emif1_phy_reg_status_dll_slave_value_byte0(1) 0x16: hwobs_emif1_phy_reg_rdlvl_fifowein_ratio_byte0_rank0(10) 0x17: hwobs_emif1_phy_reg_rdlvl_fifowein_ratio_byte0_rank1(10) 0x18: hwobs_emif1_phy_dll_calib_cmd0 0x1E: hwobs_eve1_eve_dbgout[3] 0x22: hwobs_dsp1_Turing_hw_dbgout [11] 0x23: hwobs_dsp2_Turing_hw_dbgout [11] 0x26: hwobs_vip1_outmux_testport[20] 0x2A: hwobs_debug_prm(20) 0x2B: hwobs_debug_cm_core_aon(20) 0x2C: hwobs_int_iss(20) 0x2D: hwobs_int_iss(52) 0x2E: func_spare_out[20] 0x2F: dft_test_pad_i[20] 0x30: dft_test_pad_o[20]	RW	0x0

Table 13-571. Register Call Summary for Register CTRL_CORE_CONF_DEBUG_SEL_TST_20

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-572. CTRL_CORE_CONF_DEBUG_SEL_TST_21

Address Offset	0x0000 1060		
Physical Address	0x4A00 3060	**Instance**	CTRL_MODULE_CORE
Description	This register is used to select an observable signal for CORE observability line 21.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								MODE							

Bits	Field Name	Description	Type	Reset
31:6	RESERVED		R	0x0
5:0	MODE	Selects one of the following signals to be available on CORE observability line 21. This signal can then be mapped to obs21 line through CTRL_CORE_HWOBS_FINAL_MUX_SEL[21] bit. Values not shown are reserved. 0x2: hwobs_perdpll_phaselock 0x7: hwobs_debug_dss(21) 0xB: hwobs_dsshdmiscpaddr(2) 0xC: hwobs_dsi1a_scpinpllctrl 0x12: hwobs_debug_debugss(21) 0x15: hwobs_emif1_phy_reg_status_dll_slave_value_byte0(2) 0x16: hwobs_emif1_phy_reg_gate_level_fsm_byte0(0) 0x17: hwobs_emif1_phy_reg_wr_level_fsm_byte0(2) 0x18: hwobs_emif1_phy_dll_calib_cmd1 0x1E: hwobs_eve1_eve_dbgout[2] 0x22: hwobs_dsp1_Turing_hw_dbgout [10] 0x23: hwobs_dsp2_Turing_hw_dbgout [10] 0x26: hwobs_vip1_outmux_testport[21] 0x2A: hwobs_debug_prm(21) 0x2B: hwobs_debug_cm_core_aon(21) 0x2C: hwobs_int_iss(21) 0x2D: hwobs_int_iss(53) 0x2E: func_spare_out[21] 0x2F: dft_test_pad_i[21] 0x30: dft_test_pad_o[21]	RW	0x0

Table 13-573. Register Call Summary for Register CTRL_CORE_CONF_DEBUG_SEL_TST_21

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-574. CTRL_CORE_CONF_DEBUG_SEL_TST_22

Address Offset	0x0000 1064		
Physical Address	0x4A00 3064	**Instance**	CTRL_MODULE_CORE
Description	This register is used to select an observable signal for CORE observability line 22.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								MODE							

Bits	Field Name	Description	Type	Reset
31:6	RESERVED		R	0x0
5:0	MODE	Selects one of the following signals to be available on CORE observability line 22. This signal can then be mapped to obs22 line through CTRL_CORE_HWOBS_FINAL_MUX_SEL[22] bit. Values not shown are reserved. 0x2: hwobs_perdpll_tenable 0x7: hwobs_debug_dss(22) 0xB: hwobs_dsshdmiscpaddr(1) 0xC: hwobs_dsi1a_scpbusy 0x12: hwobs_debug_debugss(22) 0x15: hwobs_emif1_phy_reg_status_dll_slave_value_byte0(3) 0x16: hwobs_emif1_phy_reg_gate_level_fsm_byte0(1) 0x17: hwobs_emif1_phy_reg_wr_level_fsm_byte0(1) 0x1E: hwobs_eve1_eve_dbgout[1] 0x22: hwobs_dsp1_Turing_hw_dbgout [9] 0x23: hwobs_dsp2_Turing_hw_dbgout [9] 0x26: hwobs_vip1_outmux_testport[22] 0x2A: hwobs_debug_prm(22) 0x2B: hwobs_debug_cm_core_aon(22) 0x2C: hwobs_int_iss(22) 0x2D: hwobs_int_iss(54) 0x2E: func_spare_out[22] 0x2F: dft_test_pad_i[22] 0x30: dft_test_pad_o[22]	RW	0x0

Table 13-575. Register Call Summary for Register CTRL_CORE_CONF_DEBUG_SEL_TST_22

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-576. CTRL_CORE_CONF_DEBUG_SEL_TST_23

Address Offset	0x0000 1068		
Physical Address	0x4A00 3068	**Instance**	CTRL_MODULE_CORE
Description	This register is used to select an observable signal for CORE observability line 23.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								MODE							

Bits	Field Name	Description	Type	Reset
31:6	RESERVED		R	0x0
5:0	MODE	Selects one of the following signals to be available on CORE observability line 23. This signal can then be mapped to obs23 line through CTRL_CORE_HWOBS_FINAL_MUX_SEL[23] bit. Values not shown are reserved. 0x2: hwobs_perdpll_tenablediv 0x7: hwobs_debug_dss(23) 0xB: hwobs_dsshdmiscpaddr(0) 0x12: hwobs_debug_debugss(23) 0x15: hwobs_emif1_phy_reg_status_dll_slave_value_byte0(4) 0x16: hwobs_emif1_phy_reg_gate_level_fsm_byte0(2) 0x17: hwobs_emif1_phy_reg_wr_level_fsm_byte0(0) 0x1E: hwobs_eve1_eve_dbgout[0] 0x22: hwobs_dsp1_Turing_hw_dbgout [8] 0x23: hwobs_dsp2_Turing_hw_dbgout [8] 0x26: hwobs_vip1_outmux_testport[23] 0x2A: hwobs_debug_prm(23) 0x2B: hwobs_debug_cm_core_aon(23) 0x2C: hwobs_int_iss(23) 0x2D: hwobs_int_iss(55) 0x2E: func_spare_out[23] 0x2F: dft_test_pad_i[23] 0x30: dft_test_pad_o[23]	RW	0x0

Table 13-577. Register Call Summary for Register CTRL_CORE_CONF_DEBUG_SEL_TST_23

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-578. CTRL_CORE_CONF_DEBUG_SEL_TST_24

Address Offset	0x0000 106C		
Physical Address	0x4A00 306C	**Instance**	CTRL_MODULE_CORE
Description	This register is used to select an observable signal for CORE observability line 24.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								MODE							

Bits	Field Name	Description	Type	Reset
31:6	RESERVED		R	0x0
5:0	MODE	Selects one of the following signals to be available on CORE observability line 24. This signal can then be mapped to obs24 line through CTRL_CORE_HWOBS_FINAL_MUX_SEL[24] bit. Values not shown are reserved. 0x2: hwobs_perdpll_bypassack 0x7: hwobs_debug_dss(24) 0xB: hwobs_dsshdmiscpcmd(1) 0x12: hwobs_debug_debugss(24) 0x15: hwobs_emif1_phy_reg_status_dll_slave_value_byte0(5) 0x16: hwobs_emif1_phy_reg_rd_level_fsm_byte0(0) 0x17: hwobs_emif1_phy_reg_rd_level_fsm_byte0(3) 0x22: hwobs_dsp1_Turing_hw_dbgout [7] 0x23: hwobs_dsp2_Turing_hw_dbgout [7] 0x26: hwobs_vip1_outmux_testport[24] 0x2A: hwobs_debug_prm(24) 0x2B: hwobs_debug_cm_core_aon(24) 0x2C: hwobs_int_iss(24) 0x2D: hwobs_int_iss(56) 0x2E: func_spare_out[24] 0x2F: dft_test_pad_i[24] 0x30: dft_test_pad_o[24]	RW	0x0

Table 13-579. Register Call Summary for Register CTRL_CORE_CONF_DEBUG_SEL_TST_24

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-580. CTRL_CORE_CONF_DEBUG_SEL_TST_25

Address Offset	0x0000 1070		
Physical Address	0x4A00 3070	**Instance**	CTRL_MODULE_CORE
Description	This register is used to select an observable signal for CORE observability line 25.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								MODE							

Bits	Field Name	Description	Type	Reset
31:6	RESERVED		R	0x0
5:0	MODE	Selects one of the following signals to be available on CORE observability line 25. This signal can then be mapped to obs25 line through CTRL_CORE_HWOBS_FINAL_MUX_SEL[25] bit. Values not shown are reserved. 0x2: hwobs_perdpll_idle 0x7: hwobs_debug_dss(25) 0xB: hwobs_dsshdmiscpcmd(0) 0x12: hwobs_debug_debugss(25) 0x15: hwobs_emif1_phy_reg_status_dll_slave_value_byte0(6) 0x16: hwobs_emif1_phy_reg_rd_level_fsm_byte0(1) 0x17: hwobs_emif1_phy_reg_rd_level_fsm_byte0(2) 0x22: hwobs_dsp1_Turing_hw_dbgout [6] 0x23: hwobs_dsp2_Turing_hw_dbgout [6] 0x26: hwobs_vip1_outmux_testport[25] 0x2A: hwobs_debug_prm(25) 0x2B: hwobs_debug_cm_core_aon(25) 0x2C: hwobs_int_iss(25) 0x2E: func_spare_out[25] 0x2F: dft_test_pad_i[25] 0x30: dft_test_pad_o[25]	RW	0x0

Table 13-581. Register Call Summary for Register CTRL_CORE_CONF_DEBUG_SEL_TST_25

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-582. CTRL_CORE_CONF_DEBUG_SEL_TST_26

Address Offset	0x0000 1074		
Physical Address	0x4A00 3074	**Instance**	CTRL_MODULE_CORE
Description	This register is used to select an observable signal for CORE observability line 26.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								MODE							

Bits	Field Name	Description	Type	Reset
31:6	RESERVED		R	0x0
5:0	MODE	Selects one of the following signals to be available on CORE observability line 26. This signal can then be mapped to obs26 line through CTRL_CORE_HWOBS_FINAL_MUX_SEL[26] bit. Values not shown are reserved. 0x7: hwobs_debug_dss(26) 0x12: hwobs_debug_debugss(26) 0x15: hwobs_emif1_phy_reg_status_dll_slave_value_byte0(7) 0x16: hwobs_emif1_phy_reg_rd_level_fsm_byte0(2) 0x17: hwobs_emif1_phy_reg_rd_level_fsm_byte0(1) 0x22: hwobs_dsp1_Turing_hw_dbgout [5] 0x23: hwobs_dsp2_Turing_hw_dbgout [5] 0x26: hwobs_vip1_outmux_testport[26] 0x2A: hwobs_debug_prm(26) 0x2B: hwobs_debug_cm_core_aon(26) 0x2C: hwobs_int_iss(26) 0x2E: func_spare_out[26] 0x2F: dft_test_pad_i[26] 0x30: dft_test_pad_o[26]	RW	0x0

Table 13-583. Register Call Summary for Register CTRL_CORE_CONF_DEBUG_SEL_TST_26

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-584. CTRL_CORE_CONF_DEBUG_SEL_TST_27

Address Offset	0x0000 1078		
Physical Address	0x4A00 3078	**Instance**	CTRL_MODULE_CORE
Description	This register is used to select an observable signal for CORE observability line 27.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								MODE							

Bits	Field Name	Description	Type	Reset
31:6	RESERVED		R	0x0
5:0	MODE	Selects one of the following signals to be available on CORE observability line 27. This signal can then be mapped to obs27 line through CTRL_CORE_HWOBS_FINAL_MUX_SEL[27] bit. Values not shown are reserved. 0x7: hwobs_debug_dss(27) 0x12: hwobs_debug_debugss(27) 0x15: hwobs_emif1_phy_reg_status_dll_slave_value_byte0(8) 0x16: hwobs_emif1_phy_reg_rd_level_fsm_byte0(3) 0x17: hwobs_emif1_phy_reg_rd_level_fsm_byte0(0) 0x22: hwobs_dsp1_Turing_hw_dbgout [4] 0x23: hwobs_dsp2_Turing_hw_dbgout [4] 0x26: hwobs_vip1_outmux_testport[27] 0x2A: hwobs_debug_prm(27) 0x2B: hwobs_debug_cm_core_aon(27) 0x2C: hwobs_int_iss(27) 0x2E: func_spare_out[27] 0x2F: dft_test_pad_i[27] 0x30: dft_test_pad_o[27]	RW	0x0

Table 13-585. Register Call Summary for Register CTRL_CORE_CONF_DEBUG_SEL_TST_27

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-586. CTRL_CORE_CONF_DEBUG_SEL_TST_28

Address Offset	0x0000 107C		
Physical Address	0x4A00 307C	**Instance**	CTRL_MODULE_CORE
Description	This register is used to select an observable signal for CORE observability line 28.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								MODE							

Bits	Field Name	Description	Type	Reset
31:6	RESERVED		R	0x0
5:0	MODE	Selects one of the following signals to be available on CORE observability line 28. This signal can then be mapped to obs28 line through CTRL_CORE_HWOBS_FINAL_MUX_SEL[28] bit. Values not shown are reserved. 0x7: hwobs_debug_dss(28) 0x12: hwobs_debug_debugss(28) 0x15: hwobs_emif1_phy_dll_calib_data0 0x16: hwobs_emif1_phy_reg_wr_level_fsm_byte0(0) 0x17: hwobs_emif1_phy_reg_gate_level_fsm_byte0(2) 0x22: hwobs_dsp1_Turing_hw_dbgout [3] 0x23: hwobs_dsp2_Turing_hw_dbgout [3] 0x26: hwobs_vip1_outmux_testport[28] 0x2A: hwobs_debug_prm(28) 0x2B: hwobs_debug_cm_core_aon(28) 0x2C: hwobs_int_iss(28) 0x2E: func_spare_out[28] 0x2F: dft_test_pad_i[28] 0x30: dft_test_pad_o[28]	RW	0x0

Table 13-587. Register Call Summary for Register CTRL_CORE_CONF_DEBUG_SEL_TST_28

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-588. CTRL_CORE_CONF_DEBUG_SEL_TST_29

Address Offset	0x0000 1080		
Physical Address	0x4A00 3080	**Instance**	CTRL_MODULE_CORE
Description	This register is used to select an observable signal for CORE observability line 29.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								MODE							

Bits	Field Name	Description	Type	Reset
31:6	RESERVED		R	0x0
5:0	MODE	Selects one of the following signals to be available on CORE observability line 29. This signal can then be mapped to obs29 line through CTRL_CORE_HWOBS_FINAL_MUX_SEL[29] bit. Values not shown are reserved. 0x7: hwobs_debug_dss(29) 0x12: hwobs_debug_debugss(29) 0x15: hwobs_emif1_pwrdn_dq0 0x16: hwobs_emif1_phy_reg_wr_level_fsm_byte0(1) 0x17: hwobs_emif1_phy_reg_gate_level_fsm_byte0(1) 0x22: hwobs_dsp1_Turing_hw_dbgout [2] 0x23: hwobs_dsp2_Turing_hw_dbgout [2] 0x26: hwobs_vip1_outmux_testport[29] 0x2A: hwobs_debug_prm(29) 0x2B: hwobs_debug_cm_core_aon(29) 0x2C: hwobs_int_iss(29) 0x2E: func_spare_out[29] 0x2F: dft_test_pad_i[29] 0x30: dft_test_pad_o[29]	RW	0x0

Table 13-589. Register Call Summary for Register CTRL_CORE_CONF_DEBUG_SEL_TST_29

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-590. CTRL_CORE_CONF_DEBUG_SEL_TST_30

Address Offset	0x0000 1084		
Physical Address	0x4A00 3084	**Instance**	CTRL_MODULE_CORE
Description	This register is used to select an observable signal for CORE observability line 30.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								MODE							

Bits	Field Name	Description	Type	Reset
31:6	RESERVED		R	0x0
5:0	MODE	Selects one of the following signals to be available on CORE observability line 30. This signal can then be mapped to obs30 line through CTRL_CORE_HWOBS_FINAL_MUX_SEL[30] bit. Values not shown are reserved. 0x2: hwobs_emif1_phy_sdramclkstop_cmd(1) 0x7: hwobs_debug_dss(30) 0x12: hwobs_debug_debugss(30) 0x16: hwobs_emif1_phy_reg_wr_level_fsm_byte0(2) 0x17: hwobs_emif1_phy_reg_gate_level_fsm_byte0(0) 0x22: hwobs_dsp1_Turing_hw_dbgout [1] 0x23: hwobs_dsp2_Turing_hw_dbgout [1] 0x26: hwobs_vip1_outmux_testport[30] 0x2A: hwobs_debug_prm(30) 0x2B: hwobs_debug_cm_core_aon(30) 0x2C: hwobs_int_iss(30) 0x2E: func_spare_out[30] 0x2F: dft_test_pad_i[30] 0x30: dft_test_pad_o[30]	RW	0x0

Table 13-591. Register Call Summary for Register CTRL_CORE_CONF_DEBUG_SEL_TST_30

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-592. CTRL_CORE_CONF_DEBUG_SEL_TST_31

Address Offset	0x0000 1088		
Physical Address	0x4A00 3088	**Instance**	CTRL_MODULE_CORE
Description	This register is used to select an observable signal for CORE observability line 31.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								MODE							

Bits	Field Name	Description	Type	Reset
31:6	RESERVED		R	0x0
5:0	MODE	Selects one of the following signals to be available on CORE observability line 31. This signal can then be mapped to obs31 line through CTRL_CORE_HWOBS_FINAL_MUX_SEL[31] bit. Values not shown are reserved. 0x7: hwobs_debug_dss(31) 0x12: hwobs_debug_debugss(31) 0x22: hwobs_dsp1_Turing_hw_dbgout [0] 0x23: hwobs_dsp2_Turing_hw_dbgout [0] 0x26: hwobs_vip1_outmux_testport[31] 0x2A: hwobs_debug_prm(31) 0x2B: hwobs_debug_cm_core_aon(31) 0x2C: hwobs_int_iss(31) 0x2E: func_spare_out[31] 0x2F: dft_test_pad_i[31] 0x30: dft_test_pad_o[31]	RW	0x0

Table 13-593. Register Call Summary for Register CTRL_CORE_CONF_DEBUG_SEL_TST_31

Control Module Functional Description • Hardware Observability Related Registers: [0]
Control Module Register Manual • CTRL_MODULE_CORE Register Summary: [1]

Table 13-594. CTRL_CORE_PAD_GPMC_CLK

Address Offset	0x0000 1400		
Physical Address	0x4A00 3400	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												GPMC_CLK_SLEWCONTROL	GPMC_CLK_INPUTENABLE	GPMC_CLK_PULLTYPESELECT	GPMC_CLK_PULLUDENABLE	RESERVED								GPMC_CLK_MUXMODE							

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	GPMC_CLK_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	GPMC_CLK_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1

Bits	Field Name	Description	Type	Reset
17	GPMC_CLK_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	GPMC_CLK_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:4	RESERVED		R	0x0
3:0	GPMC_CLK_MUXMODE	0x0: gpmc_clk 0x1: rgmii1_txc 0x5: clkout0 0x6: dma_evt1 0xE: gpio1_0 0xF: Driver off	RW	0x0

Table 13-595. Register Call Summary for Register CTRL_CORE_PAD_GPMC_CLK

Control Module Register Manual

- CTRL_MODULE_CORE Register Summary: [0]

Table 13-596. CTRL_CORE_PAD_GPMC_BEN0

Address Offset	0x0000 1404		
Physical Address	0x4A00 3404	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												GPMC_BEN0_SLEWCONTROL	GPMC_BEN0_INPUTENABLE	GPMC_BEN0_PULLTYPESELECT	GPMC_BEN0_PULLUDENABLE	RESERVED								GPMC_BEN0_MUXMODE							

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	GPMC_BEN0_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	GPMC_BEN0_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	GPMC_BEN0_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	GPMC_BEN0_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:4	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
3:0	GPMC_BEN0_MUXMODE	0x0: gpmc_ben0 0x1: rgmii1_txctl 0x4: ehrpwm1A 0x6: dma_evt2 0xE: gpio1_1 0xF: Driver off	RW	0x0

Table 13-597. Register Call Summary for Register CTRL_CORE_PAD_GPMC_BEN0

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-598. CTRL_CORE_PAD_GPMC_BEN1

Address Offset	0x0000 1408		
Physical Address	0x4A00 3408	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												GPMC_BEN1_SLEWCONTROL	GPMC_BEN1_INPUTENABLE	GPMC_BEN1_PULLTYPESELECT	GPMC_BEN1_PULLUDENABLE	RESERVED								GPMC_BEN1_MUXMODE							

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	GPMC_BEN1_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	GPMC_BEN1_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	GPMC_BEN1_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	GPMC_BEN1_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:4	RESERVED		R	0x0
3:0	GPMC_BEN1_MUXMODE	0x0: gpmc_ben1 0x1: rgmii1_txd3 0x4: ehrpwm1B 0x6: dma_evt3 0xE: gpio1_2 0xF: Driver off	RW	0xF

Table 13-599. Register Call Summary for Register CTRL_CORE_PAD_GPMC_BEN1

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-600. CTRL_CORE_PAD_GPMC_ADVN_ALE

Address Offset	0x0000 140C		
Physical Address	0x4A00 340C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												GPMC_ADVN_ALE_SLEWCONTROL	GPMC_ADVN_ALE_INPUTENABLE	GPMC_ADVN_ALE_PULLTYPESELECT	GPMC_ADVN_ALE_PULLUDENABLE	RESERVED								GPMC_ADVN_ALE_MUXMODE							

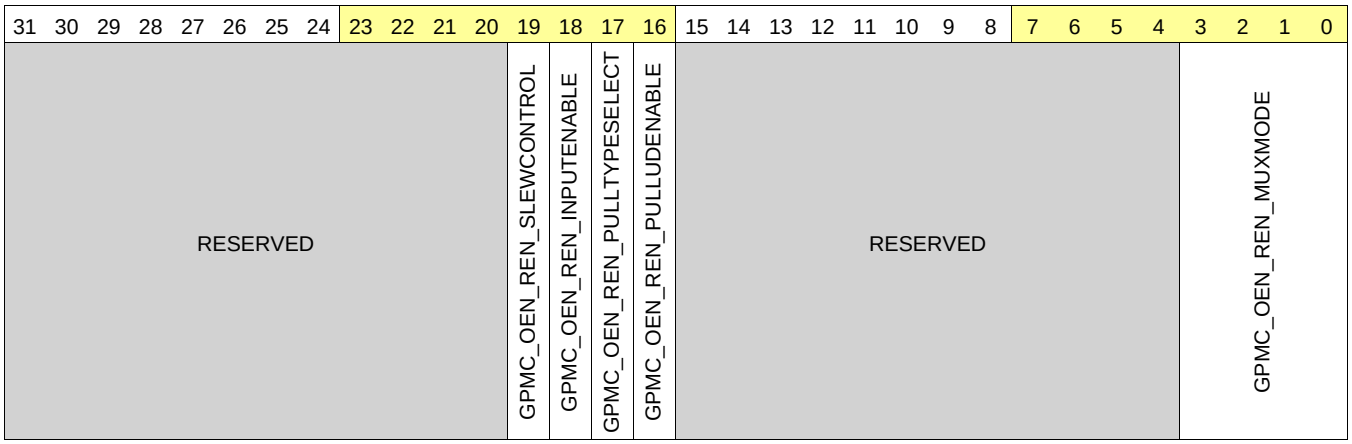
Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	GPMC_ADVN_ALE_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	GPMC_ADVN_ALE_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	GPMC_ADVN_ALE_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	GPMC_ADVN_ALE_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:4	RESERVED		R	0x0
3:0	GPMC_ADVN_ALE_MUXMODE	0x0: gpmc_advn_ale 0x1: rgmii1_txd2 0x4: ehrpwm1_tripzone_input 0x5: clkout1 0x6: dma_evt4 0xE: gpio1_3 0xF: Driver off	RW	0x0

Table 13-601. Register Call Summary for Register CTRL_CORE_PAD_GPMC_ADVN_ALE

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-602. CTRL_CORE_PAD_GPMC_OEN_REN

Address Offset	0x0000 1410		
Physical Address	0x4A00 3410	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		



Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	GPMC_OEN_REN_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	GPMC_OEN_REN_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	GPMC_OEN_REN_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	GPMC_OEN_REN_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:4	RESERVED		R	0x0
3:0	GPMC_OEN_REN_MUXMODE	0x0: gpmc_oen_ren 0x1: rgmii1_txd1 0x4: ehrpwm1_synci 0x5: clkout2 0xE: gpio1_4 0xF: Driver off	RW	0x0

Table 13-603. Register Call Summary for Register CTRL_CORE_PAD_GPMC_OEN_REN

Control Module Register Manual

- CTRL_MODULE_CORE Register Summary: [0]

Table 13-604. CTRL_CORE_PAD_GPMC_WEN

Address Offset	0x0000 1414		
Physical Address	0x4A00 3414	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												GPMC_WEN_SLEWCONTROL	GPMC_WEN_INPUTENABLE	GPMC_WEN_PULLTYPESELECT	GPMC_WEN_PULLUDENABLE	RESERVED								GPMC_WEN_MUXMODE							

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	GPMC_WEN_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	GPMC_WEN_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	GPMC_WEN_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	GPMC_WEN_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:4	RESERVED		R	0x0
3:0	GPMC_WEN_MUXMODE	0x0: gpmc_wen 0x1: rgmii1_txd0 0x4: ehrpwm1_synco 0xE: gpio1_5 0xF: Driver off	RW	0x0

Table 13-605. Register Call Summary for Register CTRL_CORE_PAD_GPMC_WEN

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-606. CTRL_CORE_PAD_GPMC_CS0

Address Offset	0x0000 1418		
Physical Address	0x4A00 3418	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												GPMC_CS0_SLEWCONTROL	GPMC_CS0_INPUTENABLE	GPMC_CS0_PULLTYPESELECT	GPMC_CS0_PULLUDENABLE	RESERVED								GPMC_CS0_MUXMODE							

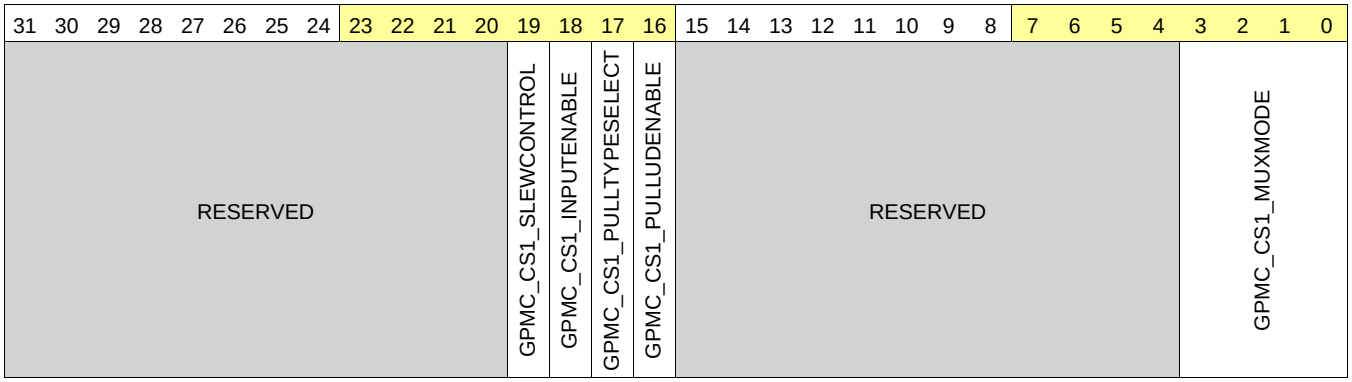
Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	GPMC_CS0_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	GPMC_CS0_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	GPMC_CS0_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x1
16	GPMC_CS0_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:4	RESERVED		R	0x0
3:0	GPMC_CS0_MUXMODE	0x0: gpmc_cs0 0x1: rgmii1_rxctl 0xE: gpio1_6 0xF: Driver off	RW	0x0

Table 13-607. Register Call Summary for Register CTRL_CORE_PAD_GPMC_CS0

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-608. CTRL_CORE_PAD_GPMC_CS1

Address Offset	0x0000 141C		
Physical Address	0x4A00 341C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		



Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	GPMC_CS1_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	GPMC_CS1_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	GPMC_CS1_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x1
16	GPMC_CS1_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:4	RESERVED		R	0x0
3:0	GPMC_CS1_MUXMODE	0x0: gpmc_cs1 0x1: qspi1_cs0 0xE: gpio1_7 0xF: Driver off	RW	0xF

Table 13-609. Register Call Summary for Register CTRL_CORE_PAD_GPMC_CS1

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-610. CTRL_CORE_PAD_GPMC_CS2

Address Offset	0x0000 1420		
Physical Address	0x4A00 3420	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												GPMC_CS2_SLEWCONTROL	GPMC_CS2_INPUTENABLE	GPMC_CS2_PULLTYPESELECT	GPMC_CS2_PULLUDENABLE	RESERVED								GPMC_CS2_MUXMODE							

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	GPMC_CS2_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	GPMC_CS2_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	GPMC_CS2_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x1
16	GPMC_CS2_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:4	RESERVED		R	0x0
3:0	GPMC_CS2_MUXMODE	0x0: gpmc_cs2 0x1: qspi1_d3 0xE: gpio1_8 0xF: Driver off	RW	0xF

Table 13-611. Register Call Summary for Register CTRL_CORE_PAD_GPMC_CS2

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-612. CTRL_CORE_PAD_GPMC_CS3

Address Offset	0x0000 1424		
Physical Address	0x4A00 3424	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												GPMC_CS3_SLEWCONTROL	GPMC_CS3_INPUTENABLE	GPMC_CS3_PULLTYPESELECT	GPMC_CS3_PULLUDENABLE	RESERVED								GPMC_CS3_MUXMODE							

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	GPMC_CS3_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	GPMC_CS3_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	GPMC_CS3_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x1
16	GPMC_CS3_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:4	RESERVED		R	0x0
3:0	GPMC_CS3_MUXMODE	0x0: gpmc_cs3 0x1: qspi1_d2 0xE: gpio1_9 0xF: Driver off	RW	0xF

Table 13-613. Register Call Summary for Register CTRL_CORE_PAD_GPMC_CS3

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-614. CTRL_CORE_PAD_GPMC_CS4

Address Offset	0x0000 1428		
Physical Address	0x4A00 3428	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												GPMC_CS4_SLEWCONTROL	GPMC_CS4_INPUTENABLE	GPMC_CS4_PULLTYPESELECT	GPMC_CS4_PULLUDENABLE	RESERVED								GPMC_CS4_MUXMODE							

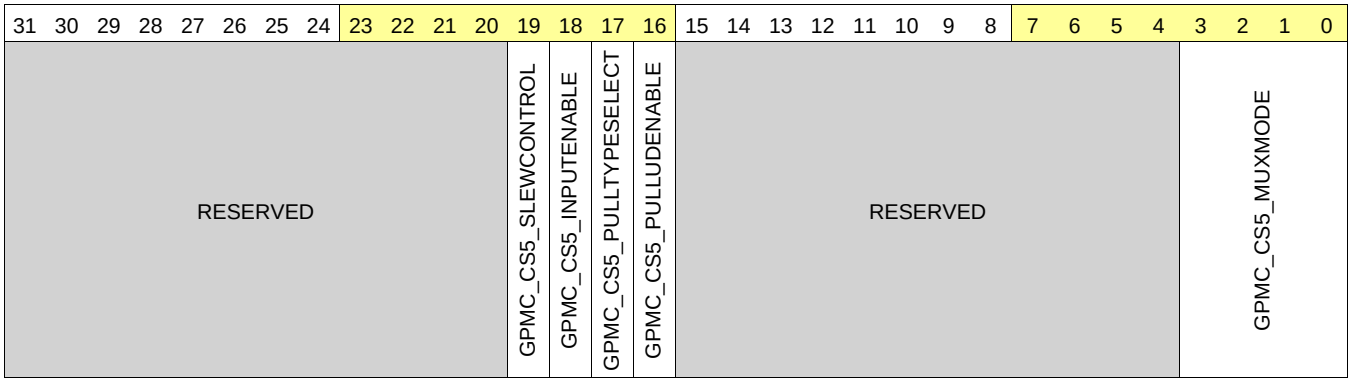
Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	GPMC_CS4_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	GPMC_CS4_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	GPMC_CS4_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x1
16	GPMC_CS4_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:4	RESERVED		R	0x0
3:0	GPMC_CS4_MUXMODE	0x0: gpmc_cs4 0x1: qspi1_d0 0xE: gpio1_10 0xF: Driver off	RW	0xF

Table 13-615. Register Call Summary for Register CTRL_CORE_PAD_GPMC_CS4

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-616. CTRL_CORE_PAD_GPMC_CS5

Address Offset	0x0000 142C		
Physical Address	0x4A00 342C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		



Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	GPMC_CS5_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	GPMC_CS5_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	GPMC_CS5_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x1
16	GPMC_CS5_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:4	RESERVED		R	0x0
3:0	GPMC_CS5_MUXMODE	0x0: gpmc_cs5 0x1: qspi1_d1 0xE: gpio1_11 0xF: Driver off	RW	0xF

Table 13-617. Register Call Summary for Register CTRL_CORE_PAD_GPMC_CS5

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-618. CTRL_CORE_PAD_GPMC_CS6

Address Offset	0x0000 1430		
Physical Address	0x4A00 3430	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												GPMC_CS6_SLEWCONTROL	GPMC_CS6_INPUTENABLE	GPMC_CS6_PULLTYPESELECT	GPMC_CS6_PULLUDENABLE	RESERVED								GPMC_CS6_MUXMODE							

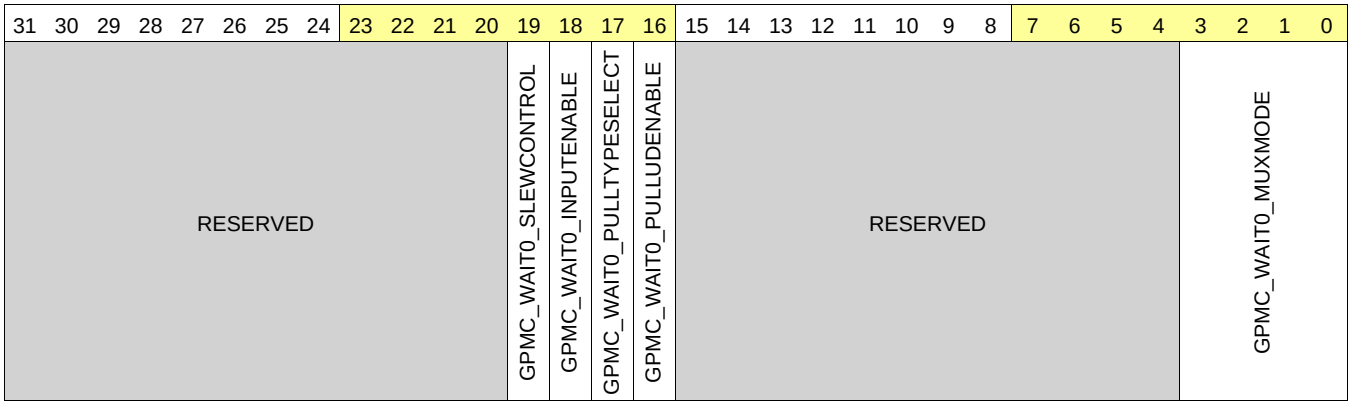
Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	GPMC_CS6_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	GPMC_CS6_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	GPMC_CS6_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x1
16	GPMC_CS6_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:4	RESERVED		R	0x0
3:0	GPMC_CS6_MUXMODE	0x0: gpmc_cs6 0x1: qspi1_sclk 0xE: gpio1_12 0xF: Driver off	RW	0xF

Table 13-619. Register Call Summary for Register CTRL_CORE_PAD_GPMC_CS6

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-620. CTRL_CORE_PAD_GPMC_WAIT0

Address Offset	0x0000 1434		
Physical Address	0x4A00 3434	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		



Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	GPMC_WAIT0_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	GPMC_WAIT0_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	GPMC_WAIT0_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x1
16	GPMC_WAIT0_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:4	RESERVED		R	0x0
3:0	GPMC_WAIT0_MUXMODE	0x0: gpmc_wait0 0x1: rgmii1_rxd3 0x2: qspi1_rtclk 0x6: dma_evt4 0xE: gpio1_13 0xF: Driver off	RW	0x0

Table 13-621. Register Call Summary for Register CTRL_CORE_PAD_GPMC_WAIT0

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-622. CTRL_CORE_PAD_GPMC_AD0

Address Offset	0x0000 1438		
Physical Address	0x4A00 3438	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												GPMC_AD0_SLEWCONTROL	GPMC_AD0_INPUTENABLE	GPMC_AD0_PULLTYPESELECT	GPMC_AD0_PULLUDENABLE	RESERVED								GPMC_AD0_MUXMODE							

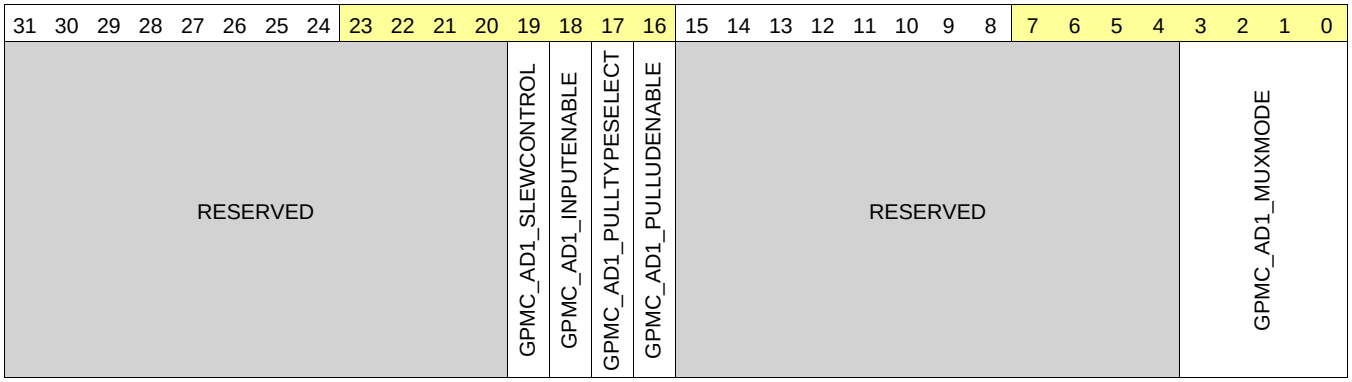
Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	GPMC_AD0_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	GPMC_AD0_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	GPMC_AD0_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	GPMC_AD0_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x1
15:4	RESERVED		R	0x0
3:0	GPMC_AD0_MUXMODE	0x0: gpmc_ad0 0x1: rgmii1_rxd2 0xE: gpio1_14 0xF: sysboot0	RW	0xF

Table 13-623. Register Call Summary for Register CTRL_CORE_PAD_GPMC_AD0

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-624. CTRL_CORE_PAD_GPMC_AD1

Address Offset	0x0000 143C		
Physical Address	0x4A00 343C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		



Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	GPMC_AD1_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	GPMC_AD1_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	GPMC_AD1_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	GPMC_AD1_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x1
15:4	RESERVED		R	0x0
3:0	GPMC_AD1_MUXMODE	0x0: gpmc_ad1 0x1: rgmii1_rxd1 0xE: gpio1_15 0xF: sysboot1	RW	0xF

Table 13-625. Register Call Summary for Register CTRL_CORE_PAD_GPMC_AD1

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-626. CTRL_CORE_PAD_GPMC_AD2

Address Offset	0x0000 1440		
Physical Address	0x4A00 3440	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												GPMC_AD2_SLEWCONTROL	GPMC_AD2_INPUTENABLE	GPMC_AD2_PULLTYPESELECT	GPMC_AD2_PULLUDENABLE	RESERVED								GPMC_AD2_MUXMODE							

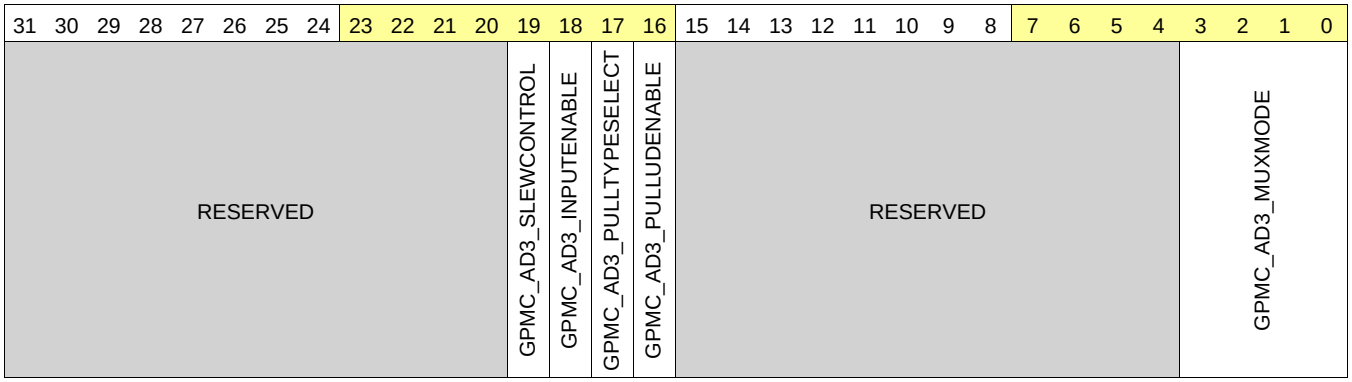
Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	GPMC_AD2_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	GPMC_AD2_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	GPMC_AD2_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	GPMC_AD2_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x1
15:4	RESERVED		R	0x0
3:0	GPMC_AD2_MUXMODE	0x0: gpmc_ad2 0x1: rgmii1_rxd0 0xE: gpio1_16 0xF: sysboot2	RW	0xF

Table 13-627. Register Call Summary for Register CTRL_CORE_PAD_GPMC_AD2

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-628. CTRL_CORE_PAD_GPMC_AD3

Address Offset	0x0000 1444		
Physical Address	0x4A00 3444	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		



Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	GPMC_AD3_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	GPMC_AD3_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	GPMC_AD3_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	GPMC_AD3_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x1
15:4	RESERVED		R	0x0
3:0	GPMC_AD3_MUXMODE	0x0: gpmc_ad3 0x1: qspi1_rtclk 0xE: gpio1_17 0xF: sysboot3	RW	0xF

Table 13-629. Register Call Summary for Register CTRL_CORE_PAD_GPMC_AD3

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-630. CTRL_CORE_PAD_GPMC_AD4

Address Offset	0x0000 1448		
Physical Address	0x4A00 3448	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												GPMC_AD4_SLEWCONTROL	GPMC_AD4_INPUTENABLE	GPMC_AD4_PULLTYPESELECT	GPMC_AD4_PULLUDENABLE	RESERVED								GPMC_AD4_MUXMODE							

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	GPMC_AD4_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	GPMC_AD4_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	GPMC_AD4_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	GPMC_AD4_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x1
15:4	RESERVED		R	0x0
3:0	GPMC_AD4_MUXMODE	0x0: gpmc_ad4 0x1: cam_strobe 0xE: gpio1_18 0xF: sysboot4	RW	0xF

Table 13-631. Register Call Summary for Register CTRL_CORE_PAD_GPMC_AD4

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-632. CTRL_CORE_PAD_GPMC_AD5

Address Offset	0x0000 144C		
Physical Address	0x4A00 344C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												GPMC_AD5_SLEWCONTROL	GPMC_AD5_INPUTENABLE	GPMC_AD5_PULLTYPESELECT	GPMC_AD5_PULLUDENABLE	RESERVED								GPMC_AD5_MUXMODE							

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	GPMC_AD5_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	GPMC_AD5_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	GPMC_AD5_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	GPMC_AD5_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x1
15:4	RESERVED		R	0x0
3:0	GPMC_AD5_MUXMODE	0x0: gpmc_ad5 0x2: uart2_txd 0x3: timer6 0x4: spi3_d1 0xE: gpio1_19 0xF: sysboot5	RW	0xF

Table 13-633. Register Call Summary for Register CTRL_CORE_PAD_GPMC_AD5

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-634. CTRL_CORE_PAD_GPMC_AD6

Address Offset	0x0000 1450		
Physical Address	0x4A00 3450	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												GPMC_AD6_SLEWCONTROL	GPMC_AD6_INPUTENABLE	GPMC_AD6_PULLTYPESELECT	GPMC_AD6_PULLUDENABLE	RESERVED								GPMC_AD6_MUXMODE							

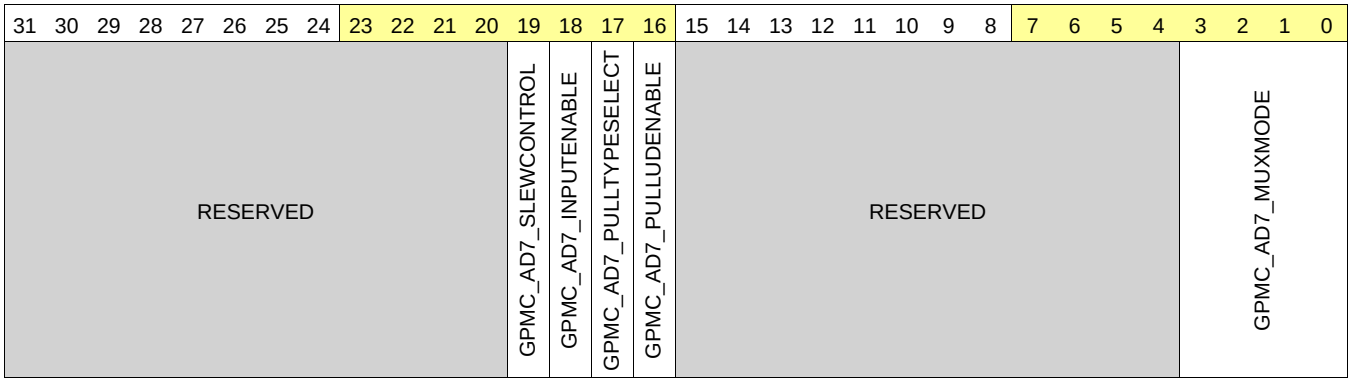
Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	GPMC_AD6_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	GPMC_AD6_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	GPMC_AD6_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	GPMC_AD6_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x1
15:4	RESERVED		R	0x0
3:0	GPMC_AD6_MUXMODE	0x0: gpmc_ad6 0x2: uart2_rxd 0x3: timer5 0x4: spi3_d0 0xE: gpio1_20 0xF: sysboot6	RW	0xF

Table 13-635. Register Call Summary for Register CTRL_CORE_PAD_GPMC_AD6

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-636. CTRL_CORE_PAD_GPMC_AD7

Address Offset	0x0000 1454		
Physical Address	0x4A00 3454	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		



Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	GPMC_AD7_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	GPMC_AD7_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	GPMC_AD7_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	GPMC_AD7_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x1
15:4	RESERVED		R	0x0
3:0	GPMC_AD7_MUXMODE	0x0: gpmc_ad7 0x1: cam_shutter 0x3: timer4 0x4: spi3_sclk 0xE: gpio1_21 0xF: Driver off	RW	0x0

Table 13-637. Register Call Summary for Register CTRL_CORE_PAD_GPMC_AD7

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-638. CTRL_CORE_PAD_GPMC_AD8

Address Offset	0x0000 1458		
Physical Address	0x4A00 3458	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												GPMC_AD8_SLEWCONTROL	GPMC_AD8_INPUTENABLE	GPMC_AD8_PULLTYPESELECT	GPMC_AD8_PULLUDENABLE	RESERVED								GPMC_AD8_MUXMODE							

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	GPMC_AD8_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	GPMC_AD8_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	GPMC_AD8_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	GPMC_AD8_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x1
15:4	RESERVED		R	0x0
3:0	GPMC_AD8_MUXMODE	0x0: gpmc_ad8 0x3: timer7 0x4: spi3_cs0 0xE: gpio1_22 0xF: sysboot8	RW	0xF

Table 13-639. Register Call Summary for Register CTRL_CORE_PAD_GPMC_AD8

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-640. CTRL_CORE_PAD_GPMC_AD9

Address Offset	0x0000 145C		
Physical Address	0x4A00 345C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												GPMC_AD9_SLEWCONTROL	GPMC_AD9_INPUTENABLE	GPMC_AD9_PULLTYPESELECT	GPMC_AD9_PULLUDENABLE	RESERVED								GPMC_AD9_MUXMODE							

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	GPMC_AD9_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	GPMC_AD9_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	GPMC_AD9_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	GPMC_AD9_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x1
15:4	RESERVED		R	0x0
3:0	GPMC_AD9_MUXMODE	0x0: gpmc_ad9 0x3: eCAP1_in_PWM1_out 0x4: spi3_cs1 0xE: gpio1_23 0xF: sysboot9	RW	0xF

Table 13-641. Register Call Summary for Register CTRL_CORE_PAD_GPMC_AD9

Control Module Register Manual

- CTRL_MODULE_CORE Register Summary: [0]

Table 13-642. CTRL_CORE_PAD_GPMC_AD10

Address Offset	0x0000 1460		
Physical Address	0x4A00 3460	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												GPMC_AD10_SLEWCONTROL	GPMC_AD10_INPUTENABLE	GPMC_AD10_PULLTYPESELECT	GPMC_AD10_PULLUDENABLE	RESERVED								GPMC_AD10_MUXMODE							

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	GPMC_AD10_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	GPMC_AD10_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	GPMC_AD10_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	GPMC_AD10_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x1
15:4	RESERVED		R	0x0
3:0	GPMC_AD10_MUXMODE	0x0: gpmc_ad10 0x3: timer2 0xE: gpio1_24 0xF: sysboot10	RW	0xF

Table 13-643. Register Call Summary for Register CTRL_CORE_PAD_GPMC_AD10

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-644. CTRL_CORE_PAD_GPMC_AD11

Address Offset	0x0000 1464		
Physical Address	0x4A00 3464	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												GPMC_AD11_SLEWCONTROL	GPMC_AD11_INPUTENABLE	GPMC_AD11_PULLTYPESELECT	GPMC_AD11_PULLUDENABLE	RESERVED								GPMC_AD11_MUXMODE							

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	GPMC_AD11_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	GPMC_AD11_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	GPMC_AD11_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	GPMC_AD11_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x1
15:4	RESERVED		R	0x0
3:0	GPMC_AD11_MUXMODE	0x0: gpmc_ad11 0x3: timer3 0xE: gpio1_25 0xF: sysboot11	RW	0xF

Table 13-645. Register Call Summary for Register CTRL_CORE_PAD_GPMC_AD11

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-646. CTRL_CORE_PAD_GPMC_AD12

Address Offset	0x0000 1468		
Physical Address	0x4A00 3468	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												GPMC_AD12_SLEWCONTROL	GPMC_AD12_INPUTENABLE	GPMC_AD12_PULLTYPESELECT	GPMC_AD12_PULLUDENABLE	RESERVED												GPMC_AD12_MUXMODE			

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	GPMC_AD12_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	GPMC_AD12_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	GPMC_AD12_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	GPMC_AD12_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x1
15:4	RESERVED		R	0x0
3:0	GPMC_AD12_MUXMODE	0x0: gpmc_ad12 0xE: gpio1_26 0xF: sysboot12	RW	0xF

Table 13-647. Register Call Summary for Register CTRL_CORE_PAD_GPMC_AD12

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-648. CTRL_CORE_PAD_GPMC_AD13

Address Offset	0x0000 146C		
Physical Address	0x4A00 346C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												GPMC_AD13_SLEWCONTROL	GPMC_AD13_INPUTENABLE	GPMC_AD13_PULLTYPESELECT	GPMC_AD13_PULLUDENABLE	RESERVED								GPMC_AD13_MUXMODE							

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	GPMC_AD13_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	GPMC_AD13_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	GPMC_AD13_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	GPMC_AD13_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x1
15:4	RESERVED		R	0x0
3:0	GPMC_AD13_MUXMODE	0x0: gpmc_ad13 0x1: rgmii1_rxc 0xE: gpio1_27 0xF: sysboot13	RW	0xF

Table 13-649. Register Call Summary for Register CTRL_CORE_PAD_GPMC_AD13

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-650. CTRL_CORE_PAD_GPMC_AD14

Address Offset	0x0000 1470		
Physical Address	0x4A00 3470	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												GPMC_AD14_SLEWCONTROL	GPMC_AD14_INPUTENABLE	GPMC_AD14_PULLTYPESELECT	GPMC_AD14_PULLUDENABLE	RESERVED								GPMC_AD14_MUXMODE							

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	GPMC_AD14_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	GPMC_AD14_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	GPMC_AD14_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	GPMC_AD14_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x1
15:4	RESERVED		R	0x0
3:0	GPMC_AD14_MUXMODE	0x0: gpmc_ad14 0x4: spi2_cs1 0xE: gpio1_28 0xF: sysboot14	RW	0xF

Table 13-651. Register Call Summary for Register CTRL_CORE_PAD_GPMC_AD14

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-652. CTRL_CORE_PAD_GPMC_AD15

Address Offset	0x0000 1474		
Physical Address	0x4A00 3474	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												GPMC_AD15_SLEWCONTROL	GPMC_AD15_INPUTENABLE	GPMC_AD15_PULLTYPESELECT	GPMC_AD15_PULLUDENABLE	RESERVED								GPMC_AD15_MUXMODE							

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	GPMC_AD15_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	GPMC_AD15_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	GPMC_AD15_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	GPMC_AD15_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x1
15:4	RESERVED		R	0x0
3:0	GPMC_AD15_MUXMODE	0x0: gpmc_ad15 0x4: spi2_cs0 0xE: gpio1_29 0xF: sysboot15	RW	0xF

Table 13-653. Register Call Summary for Register CTRL_CORE_PAD_GPMC_AD15

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-654. CTRL_CORE_PAD_VIN1A_CLK0

Address Offset	0x0000 1478		
Physical Address	0x4A00 3478	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												VIN1A_CLK0_SLEWCONTROL	VIN1A_CLK0_INPUTENABLE	VIN1A_CLK0_PULLTYPESELECT	VIN1A_CLK0_PULLUDENABLE	RESERVED												VIN1A_CLK0_MUXMODE			

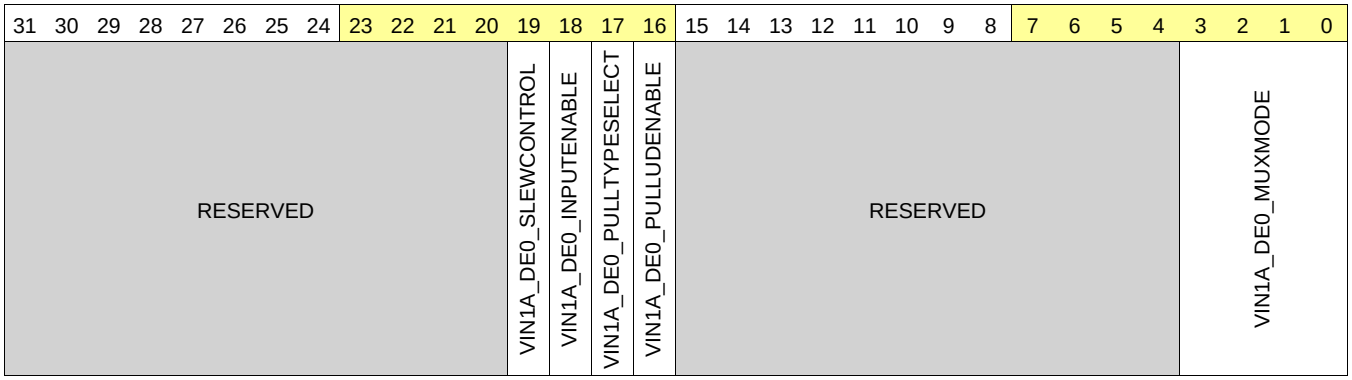
Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	VIN1A_CLK0_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	VIN1A_CLK0_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VIN1A_CLK0_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VIN1A_CLK0_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:4	RESERVED		R	0x0
3:0	VIN1A_CLK0_MUXMODE	0x0: vin1a_clk0 0x1: cpi_pclk 0x4: clkout0 0xE: gpio1_30 0xF: Driver off	RW	0xF

Table 13-655. Register Call Summary for Register CTRL_CORE_PAD_VIN1A_CLK0

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-656. CTRL_CORE_PAD_VIN1A_DE0

Address Offset	0x0000 147C		
Physical Address	0x4A00 347C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		



Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	VIN1A_DE0_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x1
18	VIN1A_DE0_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VIN1A_DE0_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VIN1A_DE0_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:4	RESERVED		R	0x0
3:0	VIN1A_DE0_MUXMODE	0x0: vin1a_de0 0x1: cpi_hsync 0x2: vin1b_clk1 0x4: clkout1 0xE: gpio1_31 0xF: Driver off	RW	0xF

Table 13-657. Register Call Summary for Register CTRL_CORE_PAD_VIN1A_DE0

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-658. CTRL_CORE_PAD_VIN1A_FLD0

Address Offset	0x0000 1480		
Physical Address	0x4A00 3480	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												VIN1A_FLD0_SLEWCONTROL	VIN1A_FLD0_INPUTENABLE	VIN1A_FLD0_PULLTYPESELECT	VIN1A_FLD0_PULLUDENABLE	RESERVED								VIN1A_FLD0_MUXMODE							

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	VIN1A_FLD0_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	VIN1A_FLD0_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VIN1A_FLD0_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VIN1A_FLD0_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:4	RESERVED		R	0x0
3:0	VIN1A_FLD0_MUXMODE	0x0: vin1a_fld0 0x1: cpi_vsync 0x2: vin2b_clk1 0x4: clkout2 0xE: gpio2_0 0xF: Driver off	RW	0xF

Table 13-659. Register Call Summary for Register CTRL_CORE_PAD_VIN1A_FLD0

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-660. CTRL_CORE_PAD_VIN1A_HSYNC0

Address Offset	0x0000 1484		
Physical Address	0x4A00 3484	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												VIN1A_HSYNC0_SLEWCONTROL	VIN1A_HSYNC0_INPUTENABLE	VIN1A_HSYNC0_PULLTYPESELECT	VIN1A_HSYNC0_PULLUDENABLE	RESERVED								VIN1A_HSYNC0_MUXMODE							

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	VIN1A_HSYNC0_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	VIN1A_HSYNC0_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VIN1A_HSYNC0_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VIN1A_HSYNC0_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:4	RESERVED		R	0x0
3:0	VIN1A_HSYNC0_MUXMODE	0x0: vin1a_hsync0 0x1: cpi_data0 0x2: vin1a_de0 0xE: gpio2_1 0xF: Driver off	RW	0xF

Table 13-661. Register Call Summary for Register CTRL_CORE_PAD_VIN1A_HSYNC0

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-662. CTRL_CORE_PAD_VIN1A_VSYNC0

Address Offset	0x0000 1488		
Physical Address	0x4A00 3488	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												VIN1A_VSYNC0_SLEWCONTROL	VIN1A_VSYNC0_INPUTENABLE	VIN1A_VSYNC0_PULLTYPESELECT	VIN1A_VSYNC0_PULLUDENABLE	RESERVED								VIN1A_VSYNC0_MUXMODE							

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	VIN1A_VSYNC0_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	VIN1A_VSYNC0_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VIN1A_VSYNC0_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VIN1A_VSYNC0_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:4	RESERVED		R	0x0
3:0	VIN1A_VSYNC0_MUXMODE	0x0: vin1a_vsync0 0x1: cpi_data1 0xE: gpio2_2 0xF: Driver off	RW	0xF

Table 13-663. Register Call Summary for Register CTRL_CORE_PAD_VIN1A_VSYNC0

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-664. CTRL_CORE_PAD_VIN1A_D0

Address Offset	0x0000 148C		
Physical Address	0x4A00 348C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												VIN1A_D0_SLEWCONTROL	VIN1A_D0_INPUTENABLE	VIN1A_D0_PULLTYPESELECT	VIN1A_D0_PULLUDENABLE	RESERVED								VIN1A_D0_MUXMODE							

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	VIN1A_D0_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	VIN1A_D0_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VIN1A_D0_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VIN1A_D0_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:4	RESERVED		R	0x0
3:0	VIN1A_D0_MUXMODE	0x0: vin1a_d0 0x1: cpi_data2 0xE: gpio2_3 0xF: Driver off	RW	0xF

Table 13-665. Register Call Summary for Register CTRL_CORE_PAD_VIN1A_D0

Control Module Register Manual

- CTRL_MODULE_CORE Register Summary: [0]

Table 13-666. CTRL_CORE_PAD_VIN1A_D1

Address Offset	0x0000 1490		
Physical Address	0x4A00 3490	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												VIN1A_D1_SLEWCONTROL	VIN1A_D1_INPUTENABLE	VIN1A_D1_PULLTYPESELECT	VIN1A_D1_PULLUDENABLE	RESERVED								VIN1A_D1_MUXMODE							

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	VIN1A_D1_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	VIN1A_D1_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VIN1A_D1_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VIN1A_D1_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:4	RESERVED		R	0x0
3:0	VIN1A_D1_MUXMODE	0x0: vin1a_d1 0x1: cpi_data3 0xE: gpio2_4 0xF: Driver off	RW	0xF

Table 13-667. Register Call Summary for Register CTRL_CORE_PAD_VIN1A_D1

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-668. CTRL_CORE_PAD_VIN1A_D2

Address Offset	0x0000 1494		
Physical Address	0x4A00 3494	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												VIN1A_D2_SLEWCONTROL	VIN1A_D2_INPUTENABLE	VIN1A_D2_PULLTYPESELECT	VIN1A_D2_PULLUDENABLE	RESERVED								VIN1A_D2_MUXMODE							

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	VIN1A_D2_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	VIN1A_D2_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VIN1A_D2_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0

Bits	Field Name	Description	Type	Reset
16	VIN1A_D2_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:4	RESERVED		R	0x0
3:0	VIN1A_D2_MUXMODE	0x0: vin1a_d2 0x1: cpi_data4 0xE: gpio2_5 0xF: Driver off	RW	0xF

Table 13-669. Register Call Summary for Register CTRL_CORE_PAD_VIN1A_D2

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-670. CTRL_CORE_PAD_VIN1A_D3

Address Offset	0x0000 1498		
Physical Address	0x4A00 3498	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												VIN1A_D3_SLEWCONTROL	VIN1A_D3_INPUTENABLE	VIN1A_D3_PULLTYPESELECT	VIN1A_D3_PULLUDENABLE	RESERVED								VIN1A_D3_MUXMODE							

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	VIN1A_D3_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	VIN1A_D3_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VIN1A_D3_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VIN1A_D3_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:4	RESERVED		R	0x0
3:0	VIN1A_D3_MUXMODE	0x0: vin1a_d3 0x1: cpi_data5 0xE: gpio2_6 0xF: Driver off	RW	0xF

Table 13-671. Register Call Summary for Register CTRL_CORE_PAD_VIN1A_D3

Control Module Register Manual
• CTRL_MODULE_CORE Register Summary: [0]

Table 13-672. CTRL_CORE_PAD_VIN1A_D4

Address Offset	0x0000 149C		
Physical Address	0x4A00 349C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												VIN1A_D4_SLEWCONTROL	VIN1A_D4_INPUTENABLE	VIN1A_D4_PULLTYPESELECT	VIN1A_D4_PULLUDENABLE	RESERVED								VIN1A_D4_MUXMODE							

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	VIN1A_D4_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	VIN1A_D4_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VIN1A_D4_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VIN1A_D4_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:4	RESERVED		R	0x0
3:0	VIN1A_D4_MUXMODE	0x0: vin1a_d4 0x1: cpi_data6 0xE: gpio2_7 0xF: Driver off	RW	0xF

Table 13-673. Register Call Summary for Register CTRL_CORE_PAD_VIN1A_D4

Control Module Register Manual
• CTRL_MODULE_CORE Register Summary: [0]

Table 13-674. CTRL_CORE_PAD_VIN1A_D5

Address Offset	0x0000 14A0		
Physical Address	0x4A00 34A0	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												VIN1A_D5_SLEWCONTROL	VIN1A_D5_INPUTENABLE	VIN1A_D5_PULLTYPESELECT	VIN1A_D5_PULLUDENABLE	RESERVED								VIN1A_D5_MUXMODE							

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	VIN1A_D5_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	VIN1A_D5_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VIN1A_D5_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VIN1A_D5_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:4	RESERVED		R	0x0
3:0	VIN1A_D5_MUXMODE	0x0: vin1a_d5 0x1: cpi_data7 0xE: gpio2_8 0xF: xref_clk2	RW	0xF

Table 13-675. Register Call Summary for Register CTRL_CORE_PAD_VIN1A_D5

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-676. CTRL_CORE_PAD_VIN1A_D6

Address Offset	0x0000 14A4		
Physical Address	0x4A00 34A4	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												VIN1A_D6_SLEWCONTROL	VIN1A_D6_INPUTENABLE	VIN1A_D6_PULLTYPESELECT	VIN1A_D6_PULLUDENABLE	RESERVED								VIN1A_D6_MUXMODE							

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	VIN1A_D6_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	VIN1A_D6_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VIN1A_D6_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VIN1A_D6_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:4	RESERVED		R	0x0
3:0	VIN1A_D6_MUXMODE	0x0: vin1a_d6 0x1: cpi_data8 0xE: gpio2_9 0xF: Driver off	RW	0xF

Table 13-677. Register Call Summary for Register CTRL_CORE_PAD_VIN1A_D6

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-678. CTRL_CORE_PAD_VIN1A_D7

Address Offset	0x0000 14A8		
Physical Address	0x4A00 34A8	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												VIN1A_D7_SLEWCONTROL	VIN1A_D7_INPUTENABLE	VIN1A_D7_PULLTYPESELECT	VIN1A_D7_PULLUDENABLE	RESERVED								VIN1A_D7_MUXMODE							

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	VIN1A_D7_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	VIN1A_D7_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VIN1A_D7_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0

Bits	Field Name	Description	Type	Reset
16	VIN1A_D7_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:4	RESERVED		R	0x0
3:0	VIN1A_D7_MUXMODE	0x0: vin1a_d7 0x1: cpi_data9 0xE: gpio2_10 0xF: Driver off	RW	0xF

Table 13-679. Register Call Summary for Register CTRL_CORE_PAD_VIN1A_D7

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-680. CTRL_CORE_PAD_VIN1A_D8

Address Offset	0x0000 14AC		
Physical Address	0x4A00 34AC	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												VIN1A_D8_SLEWCONTROL	VIN1A_D8_INPUTENABLE	VIN1A_D8_PULLTYPESELECT	VIN1A_D8_PULLUDENABLE	RESERVED								VIN1A_D8_MUXMODE							

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	VIN1A_D8_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	VIN1A_D8_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VIN1A_D8_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VIN1A_D8_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:4	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
3:0	VIN1A_D8_MUXMODE	0x0: vin1a_d8 0x1: cpi_data10 0x2: vin1b_d0 0x3: gpmc_a8 0x7: sys_nirq2 0xE: gpio2_11 0xF: Driver off	RW	0xF

Table 13-681. Register Call Summary for Register CTRL_CORE_PAD_VIN1A_D8

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-682. CTRL_CORE_PAD_VIN1A_D9

Address Offset	0x0000 14B0		
Physical Address	0x4A00 34B0	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												VIN1A_D9_SLEWCONTROL	VIN1A_D9_INPUTENABLE	VIN1A_D9_PULLTYPESELECT	VIN1A_D9_PULLUDENABLE	RESERVED								VIN1A_D9_MUXMODE							

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	VIN1A_D9_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	VIN1A_D9_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VIN1A_D9_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VIN1A_D9_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:4	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
3:0	VIN1A_D9_MUXMODE	0x0: vin1a_d9 0x1: cpi_data11 0x2: vin1b_d1 0x3: gpmc_a9 0x7: sys_nirq1 0xE: gpio2_12 0xF: Driver off	RW	0xF

Table 13-683. Register Call Summary for Register CTRL_CORE_PAD_VIN1A_D9

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-684. CTRL_CORE_PAD_VIN1A_D10

Address Offset	0x0000 14B4		
Physical Address	0x4A00 34B4	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												VIN1A_D10_SLEWCONTROL	VIN1A_D10_INPUTENABLE	VIN1A_D10_PULLTYPESELECT	VIN1A_D10_PULLUDENABLE	RESERVED								VIN1A_D10_MUXMODE							

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	VIN1A_D10_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	VIN1A_D10_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VIN1A_D10_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VIN1A_D10_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:4	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
3:0	VIN1A_D10_MUXMODE	0x0: vin1a_d10 0x1: cpi_data12 0x2: vin1b_d2 0x3: gpmc_a10 0x7: sys_nirq2 0xE: gpio2_13 0xF: Driver off	RW	0xF

Table 13-685. Register Call Summary for Register CTRL_CORE_PAD_VIN1A_D10

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-686. CTRL_CORE_PAD_VIN1A_D11

Address Offset	0x0000 14B8		
Physical Address	0x4A00 34B8	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												VIN1A_D11_SLEWCONTROL	VIN1A_D11_INPUTENABLE	VIN1A_D11_PULLTYPESELECT	VIN1A_D11_PULLUDENABLE	RESERVED								VIN1A_D11_MUXMODE							

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	VIN1A_D11_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	VIN1A_D11_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VIN1A_D11_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VIN1A_D11_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:4	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
3:0	VIN1A_D11_MUXMODE	0x0: vin1a_d11 0x1: cpi_data13 0x2: vin1b_d3 0x3: gpmc_a11 0x7: sys_nirq1 0xE: gpio2_14 0xF: Driver off	RW	0xF

Table 13-687. Register Call Summary for Register CTRL_CORE_PAD_VIN1A_D11

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-688. CTRL_CORE_PAD_VIN1A_D12

Address Offset	0x0000 14BC		
Physical Address	0x4A00 34BC	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												VIN1A_D12_SLEWCONTROL	VIN1A_D12_INPUTENABLE	VIN1A_D12_PULLTYPESELECT	VIN1A_D12_PULLUDENABLE	RESERVED								VIN1A_D12_MUXMODE							

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	VIN1A_D12_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	VIN1A_D12_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VIN1A_D12_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VIN1A_D12_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:4	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
3:0	VIN1A_D12_MUXMODE	0x0: vin1a_d12 0x1: cpi_data14 0x2: vin1b_d4 0x3: gpmc_a12 0x6: dma_evt1 0xE: gpio2_15 0xF: Driver off	RW	0xF

Table 13-689. Register Call Summary for Register CTRL_CORE_PAD_VIN1A_D12

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-690. CTRL_CORE_PAD_VIN1A_D13

Address Offset	0x0000 14C0		
Physical Address	0x4A00 34C0	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												VIN1A_D13_SLEWCONTROL	VIN1A_D13_INPUTENABLE	VIN1A_D13_PULLTYPESELECT	VIN1A_D13_PULLUDENABLE	RESERVED								VIN1A_D13_MUXMODE							

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	VIN1A_D13_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	VIN1A_D13_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VIN1A_D13_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VIN1A_D13_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:4	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
3:0	VIN1A_D13_MUXMODE	0x0: vin1a_d13 0x1: cpi_wen 0x2: vin1b_d5 0x3: gpmc_a13 0x6: dma_evt2 0xE: gpio2_16 0xF: Driver off	RW	0xF

Table 13-691. Register Call Summary for Register CTRL_CORE_PAD_VIN1A_D13

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-692. CTRL_CORE_PAD_VIN1A_D14

Address Offset	0x0000 14C4		
Physical Address	0x4A00 34C4	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												VIN1A_D14_SLEWCONTROL	VIN1A_D14_INPUTENABLE	VIN1A_D14_PULLTYPESELECT	VIN1A_D14_PULLUDENABLE	RESERVED								VIN1A_D14_MUXMODE							

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	VIN1A_D14_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	VIN1A_D14_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VIN1A_D14_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VIN1A_D14_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:4	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
3:0	VIN1A_D14_MUXMODE	0x0: vin1a_d14 0x1: cpi_fid 0x2: vin1b_d6 0x3: gpmc_a14 0xE: gpio2_17 0xF: Driver off	RW	0xF

Table 13-693. Register Call Summary for Register CTRL_CORE_PAD_VIN1A_D14

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-694. CTRL_CORE_PAD_VIN1A_D15

Address Offset	0x0000 14C8		
Physical Address	0x4A00 34C8	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												VIN1A_D15_SLEWCONTROL	VIN1A_D15_INPUTENABLE	VIN1A_D15_PULLTYPESELECT	VIN1A_D15_PULLUDENABLE	RESERVED								VIN1A_D15_MUXMODE							

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	VIN1A_D15_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	VIN1A_D15_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VIN1A_D15_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VIN1A_D15_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:4	RESERVED		R	0x0
3:0	VIN1A_D15_MUXMODE	0x0: vin1a_d15 0x1: cpi_data15 0x2: vin1b_d7 0x3: gpmc_a15 0xE: gpio2_18 0xF: Driver off	RW	0xF

Table 13-695. Register Call Summary for Register CTRL_CORE_PAD_VIN1A_D15

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-696. CTRL_CORE_PAD_VIN2A_CLK0

Address Offset	0x0000 14CC		
Physical Address	0x4A00 34CC	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												VIN2A_CLK0_SLEWCONTROL	VIN2A_CLK0_INPUTENABLE	VIN2A_CLK0_PULLTYPESELECT	VIN2A_CLK0_PULLUDENABLE	RESERVED								VIN2A_CLK0_MUXMODE							

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	VIN2A_CLK0_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	VIN2A_CLK0_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VIN2A_CLK0_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VIN2A_CLK0_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:4	RESERVED		R	0x0
3:0	VIN2A_CLK0_MUXMODE	0x0: vin2a_clk0 0xE: gpio2_19 0xF: Driver off	RW	0xF

Table 13-697. Register Call Summary for Register CTRL_CORE_PAD_VIN2A_CLK0

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-698. CTRL_CORE_PAD_VIN2A_DE0

Address Offset	0x0000 14D0		
Physical Address	0x4A00 34D0	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												VIN2A_DE0_SLEWCONTROL	VIN2A_DE0_INPUTENABLE	VIN2A_DE0_PULLTYPESELECT	VIN2A_DE0_PULLUDENABLE	RESERVED								VIN2A_DE0_MUXMODE							

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	VIN2A_DE0_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	VIN2A_DE0_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VIN2A_DE0_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VIN2A_DE0_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:4	RESERVED		R	0x0
3:0	VIN2A_DE0_MUXMODE	0x0: vin2a_de0 0x1: cam_strobe 0x2: vin2b_hsync1 0x5: vin2b_de1 0xE: gpio4_21 0xF: Driver off	RW	0xF

Table 13-699. Register Call Summary for Register CTRL_CORE_PAD_VIN2A_DE0

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-700. CTRL_CORE_PAD_VIN2A_FLD0

Address Offset	0x0000 14D4		
Physical Address	0x4A00 34D4	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												VIN2A_FLD0_SLEWCONTROL	VIN2A_FLD0_INPUTENABLE	VIN2A_FLD0_PULLTYPESELECT	VIN2A_FLD0_PULLUDENABLE	RESERVED								VIN2A_FLD0_MUXMODE							

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	VIN2A_FLD0_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	VIN2A_FLD0_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VIN2A_FLD0_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VIN2A_FLD0_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:4	RESERVED		R	0x0
3:0	VIN2A_FLD0_MUXMODE	0x0: vin2a_fld0 0x1: cam_shutter 0x2: vin2b_vsync1 0xE: gpio4_22 0xF: Driver off	RW	0xF

Table 13-701. Register Call Summary for Register CTRL_CORE_PAD_VIN2A_FLD0

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-702. CTRL_CORE_PAD_VOUT1_CLK

Address Offset	0x0000 14D8		
Physical Address	0x4A00 34D8	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												VOUT1_CLK_SLEWCONTROL	VOUT1_CLK_INPUTENABLE	VOUT1_CLK_PULLTYPESELECT	VOUT1_CLK_PULLUDENABLE	RESERVED								VOUT1_CLK_MUXMODE							

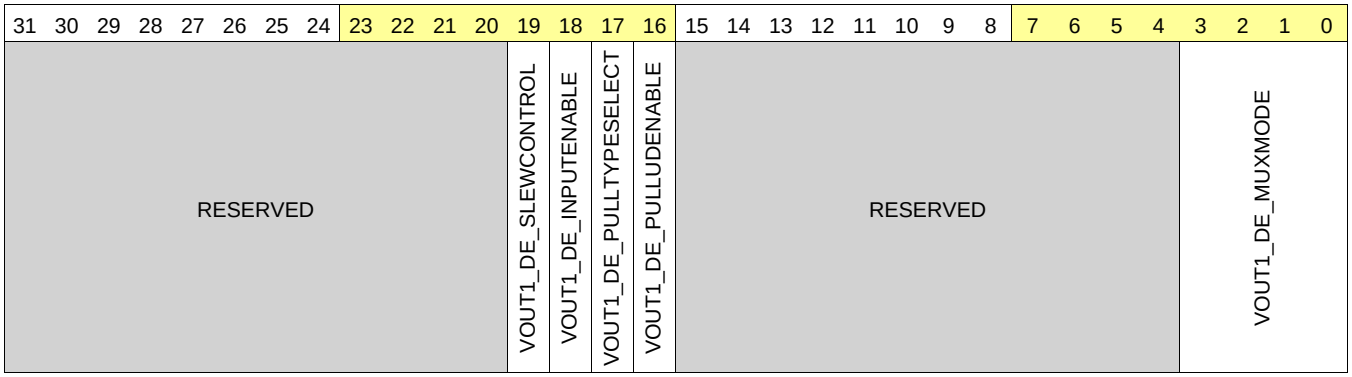
Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	VOUT1_CLK_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	VOUT1_CLK_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VOUT1_CLK_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VOUT1_CLK_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:4	RESERVED		R	0x0
3:0	VOUT1_CLK_MUXMODE	0x0: vout1_clk 0x2: vin1a_d12 0x4: clkout0 0x9: vin2a_clk0 0xE: gpio2_20 0xF: Driver off	RW	0xF

Table 13-703. Register Call Summary for Register CTRL_CORE_PAD_VOUT1_CLK

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-704. CTRL_CORE_PAD_VOUT1_DE

Address Offset	0x0000 14DC		
Physical Address	0x4A00 34DC	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		



Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	VOUT1_DE_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	VOUT1_DE_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VOUT1_DE_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VOUT1_DE_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:4	RESERVED		R	0x0
3:0	VOUT1_DE_MUXMODE	0x0: vout1_de 0x1: mcasp1_aclkx 0x2: vin1a_d13 0x4: clkout1 0xE: gpio2_21 0xF: Driver off	RW	0xF

Table 13-705. Register Call Summary for Register CTRL_CORE_PAD_VOUT1_DE

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-706. CTRL_CORE_PAD_VOUT1_FLD

Address Offset	0x0000 14E0		
Physical Address	0x4A00 34E0	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												VOUT1_FLD_SLEWCONTROL	VOUT1_FLD_INPUTENABLE	VOUT1_FLD_PULLTYPESELECT	VOUT1_FLD_PULLUDENABLE	RESERVED								VOUT1_FLD_MUXMODE							

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	VOUT1_FLD_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	VOUT1_FLD_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VOUT1_FLD_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VOUT1_FLD_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:4	RESERVED		R	0x0
3:0	VOUT1_FLD_MUXMODE	0x0: vout1_fld 0x1: mcasp1_fsx 0x2: vin1a_d14 0x4: clkout2 0xE: gpio2_22 0xF: Driver off	RW	0xF

Table 13-707. Register Call Summary for Register CTRL_CORE_PAD_VOUT1_FLD

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-708. CTRL_CORE_PAD_VOUT1_HSYNC

Address Offset	0x0000 14E4		
Physical Address	0x4A00 34E4	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												VOUT1_HSYNC_SLEWCONTROL	VOUT1_HSYNC_INPUTENABLE	VOUT1_HSYNC_PULLTYPESELECT	VOUT1_HSYNC_PULLUDENABLE	RESERVED								VOUT1_HSYNC_MUXMODE							

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	VOUT1_HSYNC_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	VOUT1_HSYNC_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VOUT1_HSYNC_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VOUT1_HSYNC_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:4	RESERVED		R	0x0
3:0	VOUT1_HSYNC_MUXMODE	0x0: vout1_hsync 0x1: mcasp1_aclkr 0x2: vin1a_d15 0x9: vin2a_de0 0xE: gpio2_23 0xF: Driver off	RW	0xF

Table 13-709. Register Call Summary for Register CTRL_CORE_PAD_VOUT1_HSYNC

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-710. CTRL_CORE_PAD_VOUT1_VSYNC

Address Offset	0x0000 14E8		
Physical Address	0x4A00 34E8	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												VOUT1_VSYNC_SLEWCONTROL	VOUT1_VSYNC_INPUTENABLE	VOUT1_VSYNC_PULLTYPESELECT	VOUT1_VSYNC_PULLUDENABLE	RESERVED												VOUT1_VSYNC_MUXMODE			

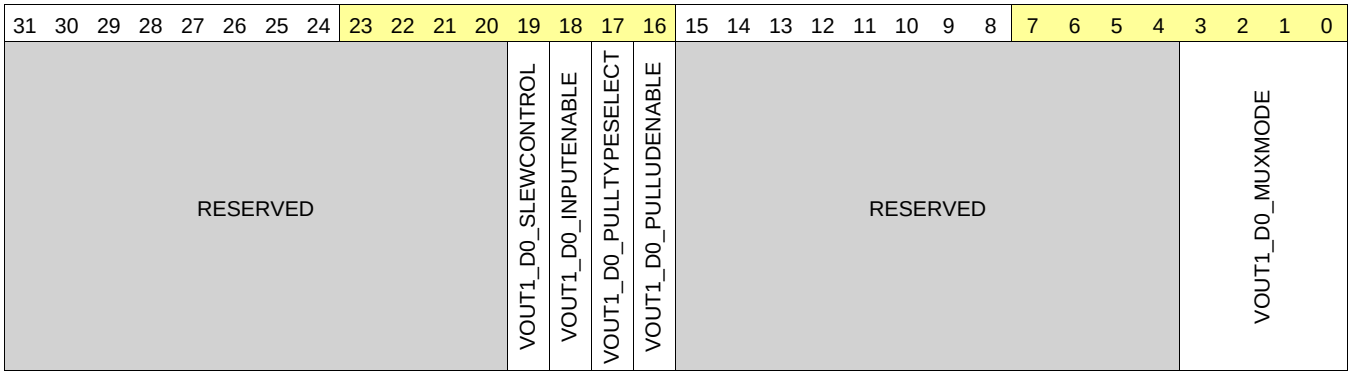
Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	VOUT1_VSYNC_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	VOUT1_VSYNC_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VOUT1_VSYNC_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VOUT1_VSYNC_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:4	RESERVED		R	0x0
3:0	VOUT1_VSYNC_MUXMODE	0x0: vout1_vsync 0x1: mcasp1_fsr 0x2: vin1a_d16 0x9: vin2a_fld0 0xE: gpio2_24 0xF: Driver off	RW	0xF

Table 13-711. Register Call Summary for Register CTRL_CORE_PAD_VOUT1_VSYNC

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-712. CTRL_CORE_PAD_VOUT1_D0

Address Offset	0x0000 14EC		
Physical Address	0x4A00 34EC	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		



Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	VOUT1_D0_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	VOUT1_D0_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VOUT1_D0_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VOUT1_D0_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:4	RESERVED		R	0x0
3:0	VOUT1_D0_MUXMODE	0x0: vout1_d0 0x1: mcasp1_axr0 0x5: mmc_clk 0xE: gpio2_25 0xF: Driver off	RW	0xF

Table 13-713. Register Call Summary for Register CTRL_CORE_PAD_VOUT1_D0

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-714. CTRL_CORE_PAD_VOUT1_D1

Address Offset	0x0000 14F0		
Physical Address	0x4A00 34F0	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												VOUT1_D1_SLEWCONTROL	VOUT1_D1_INPUTENABLE	VOUT1_D1_PULLTYPESELECT	VOUT1_D1_PULLUDENABLE	RESERVED								VOUT1_D1_MUXMODE							

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	VOUT1_D1_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	VOUT1_D1_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VOUT1_D1_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VOUT1_D1_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:4	RESERVED		R	0x0
3:0	VOUT1_D1_MUXMODE	0x0: vout1_d1 0x1: mcasp1_axr1 0x5: mmc_cmd 0xE: gpio2_26 0xF: Driver off	RW	0xF

Table 13-715. Register Call Summary for Register CTRL_CORE_PAD_VOUT1_D1

Control Module Register Manual

- CTRL_MODULE_CORE Register Summary: [0]

Table 13-716. CTRL_CORE_PAD_VOUT1_D2

Address Offset	0x0000 14F4		
Physical Address	0x4A00 34F4	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												VOUT1_D2_SLEWCONTROL	VOUT1_D2_INPUTENABLE	VOUT1_D2_PULLTYPESELECT	VOUT1_D2_PULLUDENABLE	RESERVED								VOUT1_D2_MUXMODE							

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	VOUT1_D2_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	VOUT1_D2_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VOUT1_D2_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VOUT1_D2_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:4	RESERVED		R	0x0
3:0	VOUT1_D2_MUXMODE	0x0: vout1_d2 0x1: mcasp1_axr2 0x4: mcasp1_axr8 0x5: mmc_dat0 0xE: gpio2_27 0xF: Driver off	RW	0xF if sysboot6 = 0, else 0x8

Table 13-717. Register Call Summary for Register CTRL_CORE_PAD_VOUT1_D2

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-718. CTRL_CORE_PAD_VOUT1_D3

Address Offset	0x0000 14F8		
Physical Address	0x4A00 34F8	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												VOUT1_D3_SLEWCONTROL	VOUT1_D3_INPUTENABLE	VOUT1_D3_PULLTYPESELECT	VOUT1_D3_PULLUDENABLE	RESERVED								VOUT1_D3_MUXMODE							

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	VOUT1_D3_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	VOUT1_D3_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VOUT1_D3_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VOUT1_D3_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:4	RESERVED		R	0x0
3:0	VOUT1_D3_MUXMODE	0x0: vout1_d3 0x1: mcasp1_axr3 0x4: mcasp1_axr9 0x5: mmc_dat1 0xE: gpio2_28 0xF: Driver off	RW	0xF if sysboot6 = 0, else 0x8

Table 13-719. Register Call Summary for Register CTRL_CORE_PAD_VOUT1_D3

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-720. CTRL_CORE_PAD_VOUT1_D4

Address Offset	0x0000 14FC		
Physical Address	0x4A00 34FC	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												VOUT1_D4_SLEWCONTROL	VOUT1_D4_INPUTENABLE	VOUT1_D4_PULLTYPESELECT	VOUT1_D4_PULLUDENABLE	RESERVED								VOUT1_D4_MUXMODE							

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	VOUT1_D4_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	VOUT1_D4_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VOUT1_D4_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VOUT1_D4_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:4	RESERVED		R	0x0
3:0	VOUT1_D4_MUXMODE	0x0: vout1_d4 0x1: mcasp1_axr4 0x4: mcasp1_axr10 0x5: mmc_dat2 0xE: gpio2_29 0xF: Driver off	RW	0xF if sysboot6 = 0, else 0x8

Table 13-721. Register Call Summary for Register CTRL_CORE_PAD_VOUT1_D4

Control Module Register Manual

- CTRL_MODULE_CORE Register Summary: [0]

Table 13-722. CTRL_CORE_PAD_VOUT1_D5

Address Offset	0x0000 1500		
Physical Address	0x4A00 3500	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												VOUT1_D5_SLEWCONTROL	VOUT1_D5_INPUTENABLE	VOUT1_D5_PULLTYPESELECT	VOUT1_D5_PULLUDENABLE	RESERVED								VOUT1_D5_MUXMODE							

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	VOUT1_D5_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	VOUT1_D5_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VOUT1_D5_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VOUT1_D5_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:4	RESERVED		R	0x0
3:0	VOUT1_D5_MUXMODE	0x0: vout1_d5 0x1: mcasp1_axr5 0x4: mcasp1_axr11 0x5: mmc_dat3 0x9: vin2a_clk0 0xE: gpio2_30 0xF: Driver off	RW	0xF if sysboot6 = 0, else 0x8

Table 13-723. Register Call Summary for Register CTRL_CORE_PAD_VOUT1_D5

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-724. CTRL_CORE_PAD_VOUT1_D6

Address Offset	0x0000 1504		
Physical Address	0x4A00 3504	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												VOUT1_D6_SLEWCONTROL	VOUT1_D6_INPUTENABLE	VOUT1_D6_PULLTYPESELECT	VOUT1_D6_PULLUDENABLE	RESERVED								VOUT1_D6_MUXMODE							

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	VOUT1_D6_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	VOUT1_D6_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VOUT1_D6_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VOUT1_D6_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:4	RESERVED		R	0x0
3:0	VOUT1_D6_MUXMODE	0x0: vout1_d6 0x1: mcasp1_axr6 0x4: mcasp1_axr12 0x5: esm_error Note: ESM is not supported on the DRA78x family of devices. 0x6: emu2 0x9: vin2a_de0 0xE: gpio2_31 0xF: Driver off	RW	0xF if sysboot6 = 0, else 0x8

Table 13-725. Register Call Summary for Register CTRL_CORE_PAD_VOUT1_D6

Control Module Register Manual

- CTRL_MODULE_CORE Register Summary: [0]

Table 13-726. CTRL_CORE_PAD_VOUT1_D7

Address Offset	0x0000 1508		
Physical Address	0x4A00 3508	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												VOUT1_D7_SLEWCONTROL	VOUT1_D7_INPUTENABLE	VOUT1_D7_PULLTYPESELECT	VOUT1_D7_PULLUDENABLE	RESERVED								VOUT1_D7_MUXMODE							

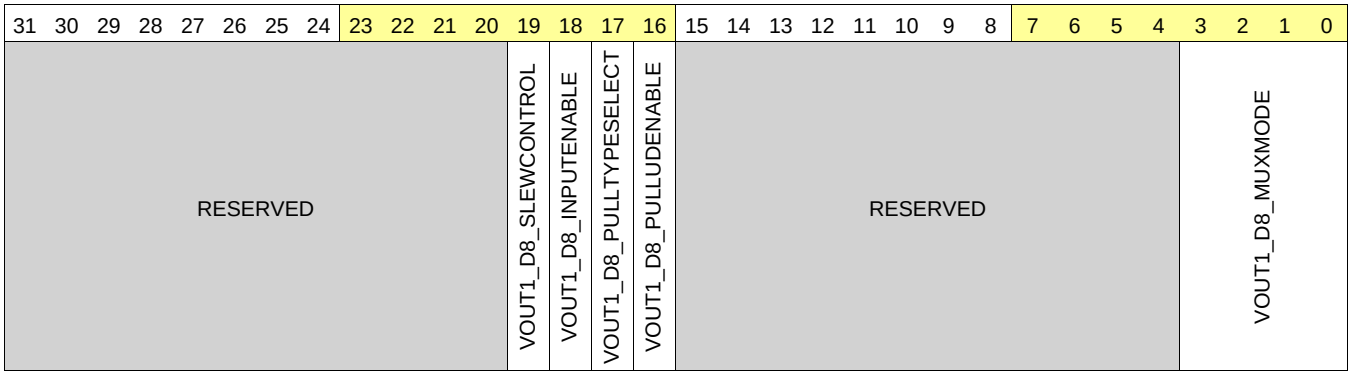
Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	VOUT1_D7_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	VOUT1_D7_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VOUT1_D7_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VOUT1_D7_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:4	RESERVED		R	0x0
3:0	VOUT1_D7_MUXMODE	0x0: vout1_d7 0x1: mcasp1_axr7 0x3: eCAP1_in_PWM1_out 0x4: mcasp1_axr13 0x6: emu3 0x9: vin2a_fld0 0xE: gpio3_0 0xF: Driver off	RW	0xF

Table 13-727. Register Call Summary for Register CTRL_CORE_PAD_VOUT1_D7

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-728. CTRL_CORE_PAD_VOUT1_D8

Address Offset	0x0000 150C		
Physical Address	0x4A00 350C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		



Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	VOUT1_D8_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	VOUT1_D8_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VOUT1_D8_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VOUT1_D8_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:4	RESERVED		R	0x0
3:0	VOUT1_D8_MUXMODE	0x0: vout1_d8 0x1: mcasp1_axr8 0x2: vin2a_d0 0x3: gpmc_a20 0x6: emu4 0xE: gpio3_1 0xF: Driver off	RW	0xF

Table 13-729. Register Call Summary for Register CTRL_CORE_PAD_VOUT1_D8

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-730. CTRL_CORE_PAD_VOUT1_D9

Address Offset	0x0000 1510		
Physical Address	0x4A00 3510	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												VOUT1_D9_SLEWCONTROL	VOUT1_D9_INPUTENABLE	VOUT1_D9_PULLTYPESELECT	VOUT1_D9_PULLUDENABLE	RESERVED								VOUT1_D9_MUXMODE							

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	VOUT1_D9_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	VOUT1_D9_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VOUT1_D9_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VOUT1_D9_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:4	RESERVED		R	0x0
3:0	VOUT1_D9_MUXMODE	0x0: vout1_d9 0x1: mcasp1_axr9 0x2: vin2a_d1 0x3: gpmc_a21 0x6: emu5 0xE: gpio3_2 0xF: Driver off	RW	0xF

Table 13-731. Register Call Summary for Register CTRL_CORE_PAD_VOUT1_D9

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-732. CTRL_CORE_PAD_VOUT1_D10

Address Offset	0x0000 1514		
Physical Address	0x4A00 3514	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												VOUT1_D10_SLEWCONTROL	VOUT1_D10_INPUTENABLE	VOUT1_D10_PULLTYPESELECT	VOUT1_D10_PULLUDENABLE	RESERVED								VOUT1_D10_MUXMODE							

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	VOUT1_D10_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	VOUT1_D10_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VOUT1_D10_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VOUT1_D10_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:4	RESERVED		R	0x0
3:0	VOUT1_D10_MUXMODE	0x0: vout1_d10 0x1: mcasp1_axr10 0x2: vin2a_d2 0x3: gpmc_a22 0x6: emu6 0xE: gpio3_3 0xF: Driver off	RW	0xF if sysboot6 = 0, else 0x8

Table 13-733. Register Call Summary for Register CTRL_CORE_PAD_VOUT1_D10

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-734. CTRL_CORE_PAD_VOUT1_D11

Address Offset	0x0000 1518		
Physical Address	0x4A00 3518	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												VOUT1_D11_SLEWCONTROL	VOUT1_D11_INPUTENABLE	VOUT1_D11_PULLTYPESELECT	VOUT1_D11_PULLUDENABLE	RESERVED								VOUT1_D11_MUXMODE							

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	VOUT1_D11_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	VOUT1_D11_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VOUT1_D11_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VOUT1_D11_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:4	RESERVED		R	0x0
3:0	VOUT1_D11_MUXMODE	0x0: vout1_d11 0x1: mcasp1_axr11 0x2: vin2a_d3 0x3: gpmc_a23 0x6: emu7 0xE: gpio3_4 0xF: Driver off	RW	0xF if sysboot6 = 0, else 0x8

Table 13-735. Register Call Summary for Register CTRL_CORE_PAD_VOUT1_D11

Control Module Register Manual

- CTRL_MODULE_CORE Register Summary: [0]

Table 13-736. CTRL_CORE_PAD_VOUT1_D12

Address Offset	0x0000 151C		
Physical Address	0x4A00 351C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												VOUT1_D12_SLEWCONTROL	VOUT1_D12_INPUTENABLE	VOUT1_D12_PULLTYPESELECT	VOUT1_D12_PULLUDENABLE	RESERVED								VOUT1_D12_MUXMODE							

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	VOUT1_D12_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	VOUT1_D12_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VOUT1_D12_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VOUT1_D12_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:4	RESERVED		R	0x0
3:0	VOUT1_D12_MUXMODE	0x0: vout1_d12 0x1: mcasp1_axr12 0x2: vin2a_d4 0x3: gpmc_a24 0x6: emu8 0xE: gpio3_5 0xF: Driver off	RW	0xF if sysboot6 = 0, else 0x8

Table 13-737. Register Call Summary for Register CTRL_CORE_PAD_VOUT1_D12

Control Module Register Manual

- CTRL_MODULE_CORE Register Summary: [0]

Table 13-738. CTRL_CORE_PAD_VOUT1_D13

Address Offset	0x0000 1520		
Physical Address	0x4A00 3520	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												VOUT1_D13_SLEWCONTROL	VOUT1_D13_INPUTENABLE	VOUT1_D13_PULLTYPESELECT	VOUT1_D13_PULLUDENABLE	RESERVED								VOUT1_D13_MUXMODE							

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	VOUT1_D13_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	VOUT1_D13_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VOUT1_D13_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VOUT1_D13_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:4	RESERVED		R	0x0
3:0	VOUT1_D13_MUXMODE	0x0: vout1_d13 0x1: mcasp1_axr13 0x2: vin2a_d5 0x3: gpmc_a25 0x6: emu9 0xE: gpio3_6 0xF: Driver off	RW	0xF if sysboot6 = 0, else 0x8

Table 13-739. Register Call Summary for Register CTRL_CORE_PAD_VOUT1_D13

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-740. CTRL_CORE_PAD_VOUT1_D14

Address Offset	0x0000 1524		
Physical Address	0x4A00 3524	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												VOUT1_D14_SLEWCONTROL	VOUT1_D14_INPUTENABLE	VOUT1_D14_PULLTYPESELECT	VOUT1_D14_PULLUDENABLE	RESERVED								VOUT1_D14_MUXMODE							

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	VOUT1_D14_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	VOUT1_D14_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VOUT1_D14_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VOUT1_D14_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:4	RESERVED		R	0x0
3:0	VOUT1_D14_MUXMODE	0x0: vout1_d14 0x1: mcasp1_axr14 0x2: vin2a_d6 0x3: gpmc_a26 0x6: emu10 0xE: gpio3_7 0xF: Driver off	RW	0xF if sysboot6 = 0, else 0x8

Table 13-741. Register Call Summary for Register CTRL_CORE_PAD_VOUT1_D14

Control Module Register Manual

- CTRL_MODULE_CORE Register Summary: [0]

Table 13-742. CTRL_CORE_PAD_VOUT1_D15

Address Offset	0x0000 1528		
Physical Address	0x4A00 3528	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												VOUT1_D15_SLEWCONTROL	VOUT1_D15_INPUTENABLE	VOUT1_D15_PULLTYPESELECT	VOUT1_D15_PULLUDENABLE	RESERVED								VOUT1_D15_MUXMODE							

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	VOUT1_D15_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	VOUT1_D15_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VOUT1_D15_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VOUT1_D15_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:4	RESERVED		R	0x0
3:0	VOUT1_D15_MUXMODE	0x0: vout1_d15 0x1: mcasp1_axr15 0x2: vin2a_d7 0x3: gpmc_a27 0x6: emu11 0xE: gpio3_8 0xF: Driver off	RW	0xF if sysboot6 = 0, else 0x8

Table 13-743. Register Call Summary for Register CTRL_CORE_PAD_VOUT1_D15

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-744. CTRL_CORE_PAD_VOUT1_D16

Address Offset	0x0000 152C		
Physical Address	0x4A00 352C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												VOUT1_D16_SLEWCONTROL	VOUT1_D16_INPUTENABLE	VOUT1_D16_PULLTYPESELECT	VOUT1_D16_PULLUDENABLE	RESERVED								VOUT1_D16_MUXMODE							

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	VOUT1_D16_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	VOUT1_D16_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VOUT1_D16_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VOUT1_D16_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:4	RESERVED		R	0x0
3:0	VOUT1_D16_MUXMODE	0x0: vout1_d16 0x1: mcasp1_ahclkx 0x2: vin2a_d8 0x3: gpmc_a0 0x4: mcasp1_axr8 0x5: vin2b_d0 0x6: emu12 0xE: gpio3_9 0xF: Driver off	RW	0xF

Table 13-745. Register Call Summary for Register CTRL_CORE_PAD_VOUT1_D16

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-746. CTRL_CORE_PAD_VOUT1_D17

Address Offset	0x0000 1530		
Physical Address	0x4A00 3530	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												VOUT1_D17_SLEWCONTROL	VOUT1_D17_INPUTENABLE	VOUT1_D17_PULLTYPESELECT	VOUT1_D17_PULLUDENABLE	RESERVED								VOUT1_D17_MUXMODE							

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	VOUT1_D17_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	VOUT1_D17_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VOUT1_D17_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VOUT1_D17_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:4	RESERVED		R	0x0
3:0	VOUT1_D17_MUXMODE	0x0: vout1_d17 0x2: vin2a_d9 0x3: gpmc_a1 0x4: mcasp1_axr9 0x5: vin2b_d1 0x6: emu13 0xE: gpio3_10 0xF: Driver off	RW	0xF

Table 13-747. Register Call Summary for Register CTRL_CORE_PAD_VOUT1_D17

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-748. CTRL_CORE_PAD_VOUT1_D18

Address Offset	0x0000 1534		
Physical Address	0x4A00 3534	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												VOUT1_D18_SLEWCONTROL	VOUT1_D18_INPUTENABLE	VOUT1_D18_PULLTYPESELECT	VOUT1_D18_PULLUDENABLE	RESERVED								VOUT1_D18_MUXMODE							

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	VOUT1_D18_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	VOUT1_D18_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VOUT1_D18_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VOUT1_D18_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:4	RESERVED		R	0x0
3:0	VOUT1_D18_MUXMODE	0x0: vout1_d18 0x2: vin2a_d10 0x3: gpmc_a2 0x4: mcasp1_axr10 0x5: vin2b_d2 0x6: emu14 0xE: gpio3_11 0xF: Driver off	RW	0xF if sysboot6 = 0, else 0x8

Table 13-749. Register Call Summary for Register CTRL_CORE_PAD_VOUT1_D18

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-750. CTRL_CORE_PAD_VOUT1_D19

Address Offset	0x0000 1538		
Physical Address	0x4A00 3538	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												VOUT1_D19_SLEWCONTROL	VOUT1_D19_INPUTENABLE	VOUT1_D19_PULLTYPESELECT	VOUT1_D19_PULLUDENABLE	RESERVED								VOUT1_D19_MUXMODE							

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	VOUT1_D19_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	VOUT1_D19_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VOUT1_D19_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VOUT1_D19_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:4	RESERVED		R	0x0
3:0	VOUT1_D19_MUXMODE	0x0: vout1_d19 0x2: vin2a_d11 0x3: gpmc_a3 0x4: mcasp1_axr11 0x5: vin2b_d3 0x6: emu15 0xE: gpio3_12 0xF: Driver off	RW	0xF if sysboot6 = 0, else 0x8

Table 13-751. Register Call Summary for Register CTRL_CORE_PAD_VOUT1_D19

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-752. CTRL_CORE_PAD_VOUT1_D20

Address Offset	0x0000 153C		
Physical Address	0x4A00 353C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												VOUT1_D20_SLEWCONTROL	VOUT1_D20_INPUTENABLE	VOUT1_D20_PULLTYPESELECT	VOUT1_D20_PULLUDENABLE	RESERVED								VOUT1_D20_MUXMODE							

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	VOUT1_D20_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	VOUT1_D20_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VOUT1_D20_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VOUT1_D20_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:4	RESERVED		R	0x0
3:0	VOUT1_D20_MUXMODE	0x0: vout1_d20 0x2: vin2a_d12 0x3: gpmc_a4 0x4: mcasp1_axr12 0x5: vin2b_d4 0x6: emu16 0xE: gpio3_13 0xF: Driver off	RW	0xF if sysboot6 = 0, else 0x8

Table 13-753. Register Call Summary for Register CTRL_CORE_PAD_VOUT1_D20

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-754. CTRL_CORE_PAD_VOUT1_D21

Address Offset	0x0000 1540		
Physical Address	0x4A00 3540	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												VOUT1_D21_SLEWCONTROL	VOUT1_D21_INPUTENABLE	VOUT1_D21_PULLTYPESELECT	VOUT1_D21_PULLUDENABLE	RESERVED								VOUT1_D21_MUXMODE							

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	VOUT1_D21_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	VOUT1_D21_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VOUT1_D21_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VOUT1_D21_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:4	RESERVED		R	0x0
3:0	VOUT1_D21_MUXMODE	0x0: vout1_d21 0x2: vin2a_d13 0x3: gpmc_a5 0x4: mcasp1_axr13 0x5: vin2b_d5 0x6: emu17 0xE: gpio3_14 0xF: Driver off	RW	0xF if sysboot6 = 0, else 0x8

Table 13-755. Register Call Summary for Register CTRL_CORE_PAD_VOUT1_D21

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-756. CTRL_CORE_PAD_VOUT1_D22

Address Offset	0x0000 1544		
Physical Address	0x4A00 3544	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												VOUT1_D22_SLEWCONTROL	VOUT1_D22_INPUTENABLE	VOUT1_D22_PULLTYPESELECT	VOUT1_D22_PULLUDENABLE	RESERVED								VOUT1_D22_MUXMODE							

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	VOUT1_D22_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	VOUT1_D22_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VOUT1_D22_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VOUT1_D22_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:4	RESERVED		R	0x0
3:0	VOUT1_D22_MUXMODE	0x0: vout1_d22 0x2: vin2a_d14 0x3: gpmc_a6 0x4: mcasp1_axr14 0x5: vin2b_d6 0x6: emu18 0xE: gpio3_15 0xF: Driver off	RW	0xF if sysboot6 = 0, else 0x8

Table 13-757. Register Call Summary for Register CTRL_CORE_PAD_VOUT1_D22

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-758. CTRL_CORE_PAD_VOUT1_D23

Address Offset	0x0000 1548		
Physical Address	0x4A00 3548	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												VOUT1_D23_SLEWCONTROL	VOUT1_D23_INPUTENABLE	VOUT1_D23_PULLTYPESELECT	VOUT1_D23_PULLUDENABLE	RESERVED								VOUT1_D23_MUXMODE							

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	VOUT1_D23_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	VOUT1_D23_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	VOUT1_D23_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	VOUT1_D23_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:4	RESERVED		R	0x0
3:0	VOUT1_D23_MUXMODE	0x0: vout1_d23 0x2: vin2a_d15 0x3: gpmc_a7 0x4: mcasp1_axr15 0x5: vin2b_d7 0x6: emu19 0xE: gpio3_16 0xF: Driver off	RW	0xF

Table 13-759. Register Call Summary for Register CTRL_CORE_PAD_VOUT1_D23

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-760. CTRL_CORE_PAD_MCAN_TX

Address Offset	0x0000 154C		
Physical Address	0x4A00 354C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												MCAN_TX_SLEWCONTROL	MCAN_TX_INPUTENABLE	MCAN_TX_PULLTYPESELECT	MCAN_TX_PULLUDENABLE	RESERVED								MCAN_TX_MUXMODE							

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	MCAN_TX_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x1
18	MCAN_TX_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	MCAN_TX_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x1
16	MCAN_TX_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:4	RESERVED		R	0x0
3:0	MCAN_TX_MUXMODE	0x0: mcan_tx 0x1: vin2a_de0 0x2: vin2a_hsync0 0x3: spi1_cs2 0x4: uart3_rxd 0x6: gpmc_wait1 0x7: vin1b_hsync1 0x8: vin1b_de1 0xE: gpio4_11 0xF: Driver off	RW	0xF

Table 13-761. Register Call Summary for Register CTRL_CORE_PAD_MCAN_TX

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-762. CTRL_CORE_PAD_MCAN_RX

Address Offset	0x0000 1550		
Physical Address	0x4A00 3550	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												MCAN_RX_SLEWCONTROL	MCAN_RX_INPUTENABLE	MCAN_RX_PULLTYPESELECT	MCAN_RX_PULLUDENABLE	RESERVED								MCAN_RX_MUXMODE							

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	MCAN_RX_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x1
18	MCAN_RX_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	MCAN_RX_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x1
16	MCAN_RX_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:4	RESERVED		R	0x0
3:0	MCAN_RX_MUXMODE	0x0: mcan_rx 0x1: cam_nreset 0x2: vin2a_vsync0 0x3: spi1_cs3 0x4: uart3_txd 0x5: gpmc_cs7 0x7: vin1b_vsync1 0xE: gpio4_12 0xF: Driver off	RW	0xF

Table 13-763. Register Call Summary for Register CTRL_CORE_PAD_MCAN_RX

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-764. CTRL_CORE_PAD_MDIO_MCLK

Address Offset	0x0000 1554		
Physical Address	0x4A00 3554	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												MDIO_MCLK_SLEWCONTROL	MDIO_MCLK_INPUTENABLE	MDIO_MCLK_PULLTYPESELECT	MDIO_MCLK_PULLUDENABLE	RESERVED								MDIO_MCLK_MUXMODE							

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	MDIO_MCLK_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x1
18	MDIO_MCLK_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	MDIO_MCLK_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x1
16	MDIO_MCLK_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:4	RESERVED		R	0x0
3:0	MDIO_MCLK_MUXMODE	0x0: mdio_mclk 0x4: spi4_d1 0xE: gpio3_17 0xF: Driver off	RW	0xF

Table 13-765. Register Call Summary for Register CTRL_CORE_PAD_MDIO_MCLK

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-766. CTRL_CORE_PAD_MDIO_D

Address Offset	0x0000 1558		
Physical Address	0x4A00 3558	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												MDIO_D_SLEWCONTROL	MDIO_D_INPUTENABLE	MDIO_D_PULLTYPESELECT	MDIO_D_PULLUDENABLE	RESERVED								MDIO_D_MUXMODE							

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	MDIO_D_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x1
18	MDIO_D_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	MDIO_D_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x1
16	MDIO_D_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:4	RESERVED		R	0x0
3:0	MDIO_D_MUXMODE	0x0: mdio_d 0x4: spi4_d0 0x5: esm_error Note: ESM is not supported on the DRA78x family of devices. 0xE: gpio3_18 0xF: Driver off	RW	0xF

Table 13-767. Register Call Summary for Register CTRL_CORE_PAD_MDIO_D

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-768. CTRL_CORE_PAD_RGMII0_TXC

Address Offset	0x0000 155C		
Physical Address	0x4A00 355C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												RGMII0_TXC_SLEWCONTROL	RGMII0_TXC_INPUTENABLE	RGMII0_TXC_PULLTYPESELECT	RGMII0_TXC_PULLUDENABLE	RESERVED								RGMII0_TXC_MUXMODE							

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	RGMII0_TXC_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	RGMII0_TXC_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	RGMII0_TXC_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	RGMII0_TXC_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:4	RESERVED		R	0x0
3:0	RGMII0_TXC_MUXMODE	0x0: rgmii0_txc 0x3: cam_strobe 0x4: spi4_sclk 0x5: mmc_clk 0xE: gpio3_19 0xF: Driver off	RW	0xF

Table 13-769. Register Call Summary for Register CTRL_CORE_PAD_RGMII0_TXC

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-770. CTRL_CORE_PAD_RGMII0_TXCTL

Address Offset	0x0000 1560		
Physical Address	0x4A00 3560	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												RGMII0_TXCTL_SLEWCONTROL	RGMII0_TXCTL_INPUTENABLE	RGMII0_TXCTL_PULLTYPESELECT	RGMII0_TXCTL_PULLUDENABLE	RESERVED												RGMII0_TXCTL_MUXMODE			

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	RGMII0_TXCTL_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	RGMII0_TXCTL_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	RGMII0_TXCTL_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	RGMII0_TXCTL_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:4	RESERVED		R	0x0
3:0	RGMII0_TXCTL_MUXMODE	0x0: rgmii0_txctl 0x3: cam_shutter 0x4: spi4_cs0 0x5: mmc_cmd 0xE: gpio3_20 0xF: Driver off	RW	0xF

Table 13-771. Register Call Summary for Register CTRL_CORE_PAD_RGMII0_TXCTL

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-772. CTRL_CORE_PAD_RGMII0_TXD3

Address Offset	0x0000 1564		
Physical Address	0x4A00 3564	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												RGMII0_TXD3_SLEWCONTROL	RGMII0_TXD3_INPUTENABLE	RGMII0_TXD3_PULLTYPESELECT	RGMII0_TXD3_PULLUDENABLE	RESERVED								RGMII0_TXD3_MUXMODE							

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	RGMII0_TXD3_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	RGMII0_TXD3_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	RGMII0_TXD3_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	RGMII0_TXD3_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:4	RESERVED		R	0x0
3:0	RGMII0_TXD3_MUXMODE	0x0: rgmii0_txd3 0x5: mmc_dat0 0xE: gpio3_21 0xF: Driver off	RW	0xF

Table 13-773. Register Call Summary for Register CTRL_CORE_PAD_RGMII0_TXD3

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-774. CTRL_CORE_PAD_RGMII0_TXD2

Address Offset	0x0000 1568		
Physical Address	0x4A00 3568	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												RGMII0_TXD2_SLEWCONTROL	RGMII0_TXD2_INPUTENABLE	RGMII0_TXD2_PULLTYPESELECT	RGMII0_TXD2_PULLUDENABLE	RESERVED								RGMII0_TXD2_MUXMODE							

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	RGMII0_TXD2_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	RGMII0_TXD2_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	RGMII0_TXD2_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	RGMII0_TXD2_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:4	RESERVED		R	0x0
3:0	RGMII0_TXD2_MUXMODE	0x0: rgmii0_txd2 0x3: eCAP1_in_PWM1_out 0x5: mmc_dat1 0xE: gpio3_22 0xF: Driver off	RW	0xF

Table 13-775. Register Call Summary for Register CTRL_CORE_PAD_RGMII0_TXD2

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-776. CTRL_CORE_PAD_RGMII0_TXD1

Address Offset	0x0000 156C		
Physical Address	0x4A00 356C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												RGMII0_TXD1_SLEWCONTROL	RGMII0_TXD1_INPUTENABLE	RGMII0_TXD1_PULLTYPESELECT	RGMII0_TXD1_PULLUDENABLE	RESERVED												RGMII0_TXD1_MUXMODE			

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	RGMII0_TXD1_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	RGMII0_TXD1_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	RGMII0_TXD1_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	RGMII0_TXD1_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:4	RESERVED		R	0x0
3:0	RGMII0_TXD1_MUXMODE	0x0: rgmii0_txd1 0x5: mmc_dat2 0xE: gpio3_23 0xF: Driver off	RW	0xF

Table 13-777. Register Call Summary for Register CTRL_CORE_PAD_RGMII0_TXD1

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-778. CTRL_CORE_PAD_RGMII0_TXD0

Address Offset	0x0000 1570		
Physical Address	0x4A00 3570	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												RGMII0_TXD0_SLEWCONTROL	RGMII0_TXD0_INPUTENABLE	RGMII0_TXD0_PULLTYPESELECT	RGMII0_TXD0_PULLUDENABLE	RESERVED												RGMII0_TXD0_MUXMODE			

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	RGMII0_TXD0_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	RGMII0_TXD0_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	RGMII0_TXD0_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	RGMII0_TXD0_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:4	RESERVED		R	0x0
3:0	RGMII0_TXD0_MUXMODE	0x0: rgmii0_txd0 0x5: mmc_dat3 0xE: gpio3_24 0xF: Driver off	RW	0xF

Table 13-779. Register Call Summary for Register CTRL_CORE_PAD_RGMII0_TXD0

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-780. CTRL_CORE_PAD_RGMII0_RXC

Address Offset	0x0000 1574		
Physical Address	0x4A00 3574	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												RGMII0_RXC_SLEWCONTROL	RGMII0_RXC_INPUTENABLE	RGMII0_RXC_PULLTYPESELECT	RGMII0_RXC_PULLUDENABLE	RESERVED								RGMII0_RXC_MUXMODE							

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	RGMII0_RXC_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	RGMII0_RXC_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	RGMII0_RXC_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	RGMII0_RXC_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:4	RESERVED		R	0x0
3:0	RGMII0_RXC_MUXMODE	0x0: rgmii0_rxc 0x3: cam_strobe 0x5: mmc_clk 0xE: gpio3_25 0xF: Driver off	RW	0xF

Table 13-781. Register Call Summary for Register CTRL_CORE_PAD_RGMII0_RXC

Control Module Register Manual

- CTRL_MODULE_CORE Register Summary: [0]

Table 13-782. CTRL_CORE_PAD_RGMII0_RXCTL

Address Offset	0x0000 1578		
Physical Address	0x4A00 3578	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												RGMII0_RXCTL_SLEWCONTROL	RGMII0_RXCTL_INPUTENABLE	RGMII0_RXCTL_PULLTYPESELECT	RGMII0_RXCTL_PULLUDENABLE	RESERVED								RGMII0_RXCTL_MUXMODE							

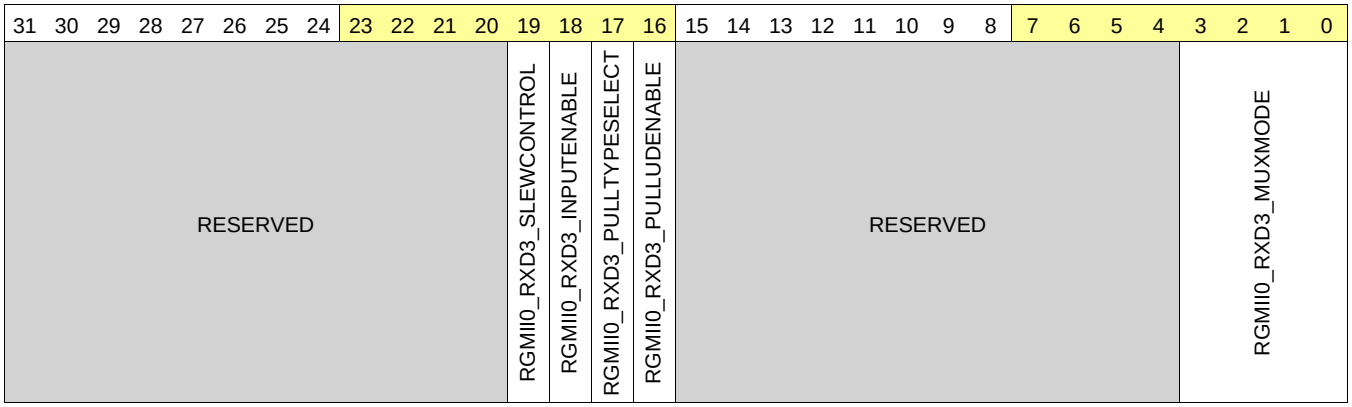
Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	RGMII0_RXCTL_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	RGMII0_RXCTL_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	RGMII0_RXCTL_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	RGMII0_RXCTL_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:4	RESERVED		R	0x0
3:0	RGMII0_RXCTL_MUXMODE	0x0: rgmii0_rxctl 0x3: cam_shutter 0x5: mmc_cmd 0xE: gpio3_26 0xF: Driver off	RW	0xF

Table 13-783. Register Call Summary for Register CTRL_CORE_PAD_RGMII0_RXCTL

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-784. CTRL_CORE_PAD_RGMII0_RXD3

Address Offset	0x0000 157C		
Physical Address	0x4A00 357C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		



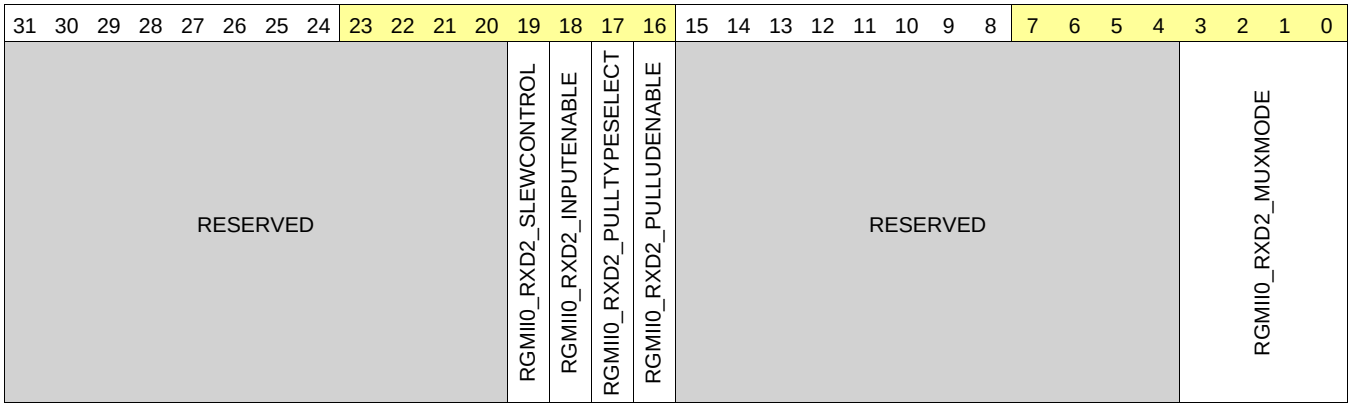
Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	RGMII0_RXD3_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	RGMII0_RXD3_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	RGMII0_RXD3_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	RGMII0_RXD3_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:4	RESERVED		R	0x0
3:0	RGMII0_RXD3_MUXMODE	0x0: rgmii0_rxd3 0x5: mmc_dat0 0xE: gpio3_27 0xF: Driver off	RW	0xF

Table 13-785. Register Call Summary for Register CTRL_CORE_PAD_RGMII0_RXD3

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-786. CTRL_CORE_PAD_RGMII0_RXD2

Address Offset	0x0000 1580		
Physical Address	0x4A00 3580	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		



Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	RGMII0_RXD2_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	RGMII0_RXD2_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	RGMII0_RXD2_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	RGMII0_RXD2_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:4	RESERVED		R	0x0
3:0	RGMII0_RXD2_MUXMODE	0x0: rgmii0_rxd2 0x5: mmc_dat1 0xE: gpio3_28 0xF: Driver off	RW	0xF

Table 13-787. Register Call Summary for Register CTRL_CORE_PAD_RGMII0_RXD2

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-788. CTRL_CORE_PAD_RGMII0_RXD1

Address Offset	0x0000 1584		
Physical Address	0x4A00 3584	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												RGMII0_RXD1_SLEWCONTROL	RGMII0_RXD1_INPUTENABLE	RGMII0_RXD1_PULLTYPESELECT	RGMII0_RXD1_PULLUDENABLE	RESERVED								RGMII0_RXD1_MUXMODE							

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	RGMII0_RXD1_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	RGMII0_RXD1_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	RGMII0_RXD1_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	RGMII0_RXD1_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:4	RESERVED		R	0x0
3:0	RGMII0_RXD1_MUXMODE	0x0: rgmii0_rxd1 0x5: mmc_dat2 0xE: gpio3_29 0xF: Driver off	RW	0xF

Table 13-789. Register Call Summary for Register CTRL_CORE_PAD_RGMII0_RXD1

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-790. CTRL_CORE_PAD_RGMII0_RXD0

Address Offset	0x0000 1588		
Physical Address	0x4A00 3588	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												RGMII0_RXD0_SLEWCONTROL	RGMII0_RXD0_INPUTENABLE	RGMII0_RXD0_PULLTYPESELECT	RGMII0_RXD0_PULLUDENABLE	RESERVED								RGMII0_RXD0_MUXMODE							

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	RGMII0_RXD0_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	RGMII0_RXD0_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	RGMII0_RXD0_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	RGMII0_RXD0_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:4	RESERVED		R	0x0
3:0	RGMII0_RXD0_MUXMODE	0x0: rgmii0_rxd0 0x5: mmc_dat3 0xE: gpio3_30 0xF: Driver off	RW	0xF

Table 13-791. Register Call Summary for Register CTRL_CORE_PAD_RGMII0_RXD0

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-792. CTRL_CORE_PAD_XREF_CLK0

Address Offset	0x0000 158C		
Physical Address	0x4A00 358C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												XREF_CLK0_SLEWCONTROL	XREF_CLK0_INPUTENABLE	XREF_CLK0_PULLTYPESELECT	XREF_CLK0_PULLUDENABLE	RESERVED								XREF_CLK0_MUXMODE							

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	XREF_CLK0_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	XREF_CLK0_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	XREF_CLK0_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	XREF_CLK0_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:4	RESERVED		R	0x0
3:0	XREF_CLK0_MUXMODE	0x0: xref_clk0 0x1: clkout0 0x4: spi3_cs0 0x5: spi2_cs1 0x6: spi1_cs0 0x7: spi1_cs1 0xE: gpio3_31 0xF: Driver off	RW	0xF

Table 13-793. Register Call Summary for Register CTRL_CORE_PAD_XREF_CLK0

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-794. CTRL_CORE_PAD_SPI1_SCLK

Address Offset	0x0000 1590		
Physical Address	0x4A00 3590	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												SPI1_SCLK_SLEWCONTROL	SPI1_SCLK_INPUTENABLE	SPI1_SCLK_PULLTYPESELECT	SPI1_SCLK_PULLUDENABLE	RESERVED								SPI1_SCLK_MUXMODE							

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	SPI1_SCLK_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	SPI1_SCLK_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	SPI1_SCLK_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	SPI1_SCLK_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:4	RESERVED		R	0x0
3:0	SPI1_SCLK_MUXMODE	0x0: spi1_sclk 0x1: uart3_rxd 0xE: gpio4_0 0xF: Driver off	RW	0xF

Table 13-795. Register Call Summary for Register CTRL_CORE_PAD_SPI1_SCLK

Control Module Register Manual

- CTRL_MODULE_CORE Register Summary: [0]

Table 13-796. CTRL_CORE_PAD_SPI1_D1

Address Offset	0x0000 1594		
Physical Address	0x4A00 3594	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												SPI1_D1_SLEWCONTROL	SPI1_D1_INPUTENABLE	SPI1_D1_PULLTYPESELECT	SPI1_D1_PULLUDENABLE	RESERVED								SPI1_D1_MUXMODE							

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	SPI1_D1_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	SPI1_D1_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	SPI1_D1_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	SPI1_D1_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:4	RESERVED		R	0x0
3:0	SPI1_D1_MUXMODE	0x0: spi1_d1 0x1: uart3_ctsn 0xE: gpio4_1 0xF: Driver off	RW	0xF

Table 13-797. Register Call Summary for Register CTRL_CORE_PAD_SPI1_D1

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-798. CTRL_CORE_PAD_SPI1_D0

Address Offset	0x0000 1598		
Physical Address	0x4A00 3598	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												SPI1_D0_SLEWCONTROL	SPI1_D0_INPUTENABLE	SPI1_D0_PULLTYPESELECT	SPI1_D0_PULLUDENABLE	RESERVED								SPI1_D0_MUXMODE							

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	SPI1_D0_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	SPI1_D0_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	SPI1_D0_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0

Bits	Field Name	Description	Type	Reset
16	SPI1_D0_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:4	RESERVED		R	0x0
3:0	SPI1_D0_MUXMODE	0x0: spi1_d0 0x1: uart3_rtsn 0xE: gpio4_2 0xF: Driver off	RW	0xF

Table 13-799. Register Call Summary for Register CTRL_CORE_PAD_SPI1_D0

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-800. CTRL_CORE_PAD_SPI1_CS0

Address Offset	0x0000 159C		
Physical Address	0x4A00 359C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												SPI1_CS0_SLEWCONTROL	SPI1_CS0_INPUTENABLE	SPI1_CS0_PULLTYPESELECT	SPI1_CS0_PULLUDENABLE	RESERVED								SPI1_CS0_MUXMODE							

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	SPI1_CS0_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	SPI1_CS0_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	SPI1_CS0_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x1
16	SPI1_CS0_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:4	RESERVED		R	0x0
3:0	SPI1_CS0_MUXMODE	0x0: spi1_cs0 0x1: uart3_txd 0xE: gpio4_3 0xF: Driver off	RW	0xF

Table 13-801. Register Call Summary for Register CTRL_CORE_PAD_SPI1_CS0

Control Module Register Manual

- CTRL_MODULE_CORE Register Summary: [0]

Table 13-802. CTRL_CORE_PAD_SPI1_CS1

Address Offset	0x0000 15A0		
Physical Address	0x4A00 35A0	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												SPI1_CS1_SLEWCONTROL	SPI1_CS1_INPUTENABLE	SPI1_CS1_PULLTYPESELECT	SPI1_CS1_PULLUDENABLE	RESERVED								SPI1_CS1_MUXMODE							

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	SPI1_CS1_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	SPI1_CS1_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	SPI1_CS1_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x1
16	SPI1_CS1_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:4	RESERVED		R	0x0
3:0	SPI1_CS1_MUXMODE	0x0: spi1_cs1 0x1: spi3_cs1 0x4: timer6 0x7: ehrpwm1_tripzone_input 0xE: gpio4_4 0xF: Driver off	RW	0xF

Table 13-803. Register Call Summary for Register CTRL_CORE_PAD_SPI1_CS1

Control Module Register Manual

- CTRL_MODULE_CORE Register Summary: [0]

Table 13-804. CTRL_CORE_PAD_SPI2_SCLK

Address Offset	0x0000 15A4		
Physical Address	0x4A00 35A4	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												SPI2_SCLK_SLEWCONTROL	SPI2_SCLK_INPUTENABLE	SPI2_SCLK_PULLTYPESELECT	SPI2_SCLK_PULLUDENABLE	RESERVED								SPI2_SCLK_MUXMODE							

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	SPI2_SCLK_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	SPI2_SCLK_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	SPI2_SCLK_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	SPI2_SCLK_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:4	RESERVED		R	0x0
3:0	SPI2_SCLK_MUXMODE	0x0: spi2_sclk 0x1: uart3_rxd 0x2: ehrpwm1A 0x3: timer3 0xE: gpio4_5 0xF: Driver off	RW	0xF

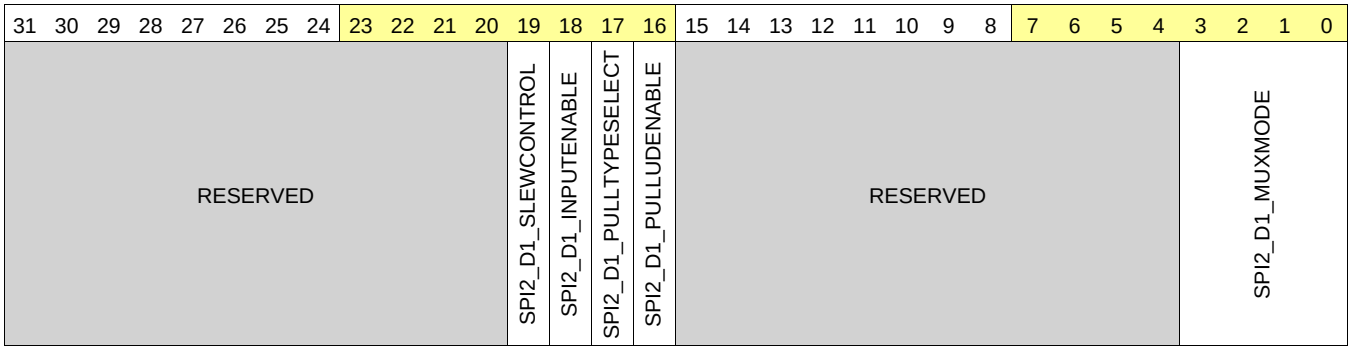
Table 13-805. Register Call Summary for Register CTRL_CORE_PAD_SPI2_SCLK

Control Module Register Manual

- CTRL_MODULE_CORE Register Summary: [0]

Table 13-806. CTRL_CORE_PAD_SPI2_D1

Address Offset	0x0000 15A8		
Physical Address	0x4A00 35A8	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		



Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	SPI2_D1_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x1
18	SPI2_D1_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	SPI2_D1_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	SPI2_D1_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:4	RESERVED		R	0x0
3:0	SPI2_D1_MUXMODE	0x0: spi2_d1 0x1: uart3_ctsn 0x3: timer5 0x7: eCAP1_in_PWM1_out 0xE: gpio4_6 0xF: Driver off	RW	0xF

Table 13-807. Register Call Summary for Register CTRL_CORE_PAD_SPI2_D1

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-808. CTRL_CORE_PAD_SPI2_D0

Address Offset	0x0000 15AC		
Physical Address	0x4A00 35AC	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												SPI2_D0_SLEWCONTROL	SPI2_D0_INPUTENABLE	SPI2_D0_PULLTYPESELECT	SPI2_D0_PULLUDENABLE	RESERVED								SPI2_D0_MUXMODE							

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	SPI2_D0_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x1
18	SPI2_D0_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	SPI2_D0_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	SPI2_D0_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:4	RESERVED		R	0x0
3:0	SPI2_D0_MUXMODE	0x0: spi2_d0 0x1: uart3_rtsn 0x3: timer1 0xE: gpio4_7 0xF: sysboot7	RW	0xF

Table 13-809. Register Call Summary for Register CTRL_CORE_PAD_SPI2_D0

Control Module Register Manual

- CTRL_MODULE_CORE Register Summary: [0]

Table 13-810. CTRL_CORE_PAD_SPI2_CS0

Address Offset	0x0000 15B0		
Physical Address	0x4A00 35B0	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												SPI2_CS0_SLEWCONTROL	SPI2_CS0_INPUTENABLE	SPI2_CS0_PULLTYPESELECT	SPI2_CS0_PULLUDENABLE	RESERVED								SPI2_CS0_MUXMODE							

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	SPI2_CS0_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x1
18	SPI2_CS0_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	SPI2_CS0_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x1
16	SPI2_CS0_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:4	RESERVED		R	0x0
3:0	SPI2_CS0_MUXMODE	0x0: spi2_cs0 0x1: uart3_txd 0x2: ehrpwm1B 0x3: timer4 0xE: gpio4_8 0xF: Driver off	RW	0xF

Table 13-811. Register Call Summary for Register CTRL_CORE_PAD_SPI2_CS0

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-812. CTRL_CORE_PAD_DCAN_TX

Address Offset	0x0000 15B4		
Physical Address	0x4A00 35B4	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												DCAN_TX_SLEWCONTROL	DCAN_TX_INPUTENABLE	DCAN_TX_PULLTYPESELECT	DCAN_TX_PULLUDENABLE	RESERVED								DCAN_TX_MUXMODE							

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	DCAN_TX_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x1
18	DCAN_TX_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	DCAN_TX_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x1
16	DCAN_TX_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:4	RESERVED		R	0x0
3:0	DCAN_TX_MUXMODE	0x0: dcan_tx 0xE: gpio4_9 0xF: Driver off	RW	0xF

Table 13-813. Register Call Summary for Register CTRL_CORE_PAD_DCAN_TX

Control Module Register Manual

- CTRL_MODULE_CORE Register Summary: [0]

Table 13-814. CTRL_CORE_PAD_DCAN_RX

Address Offset	0x0000 15B8		
Physical Address	0x4A00 35B8	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												DCAN_RX_SLEWCONTROL	DCAN_RX_INPUTENABLE	DCAN_RX_PULLTYPESELECT	DCAN_RX_PULLUDENABLE	RESERVED								DCAN_RX_MUXMODE							

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	DCAN_RX_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x1
18	DCAN_RX_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	DCAN_RX_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x1
16	DCAN_RX_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:4	RESERVED		R	0x0
3:0	DCAN_RX_MUXMODE	0x0: dcan_rx 0xE: gpio4_10 0xF: Driver off	RW	0xF

Table 13-815. Register Call Summary for Register CTRL_CORE_PAD_DCAN_RX

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-816. CTRL_CORE_PAD_UART1_RXD

Address Offset	0x0000 15BC		
Physical Address	0x4A00 35BC	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												UART1_RXD_SLEWCONTROL	UART1_RXD_INPUTENABLE	UART1_RXD_PULLTYPESELECT	UART1_RXD_PULLUDENABLE	RESERVED								UART1_RXD_MUXMODE							

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	UART1_RXD_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x1
18	UART1_RXD_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	UART1_RXD_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x1

Bits	Field Name	Description	Type	Reset
16	UART1_RXD_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:4	RESERVED		R	0x0
3:0	UART1_RXD_MUXMODE	0x0: uart1_rxd 0x4: spi4_d1 0x5: qspi1_rtclk 0xA: gpmc_a12 0xC: mcan_tx 0xE: gpio4_13 0xF: Driver off	RW	0xF

Table 13-817. Register Call Summary for Register CTRL_CORE_PAD_UART1_RXD

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-818. CTRL_CORE_PAD_UART1_TXD

Address Offset	0x0000 15C0		
Physical Address	0x4A00 35C0	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												UART1_TXD_SLEWCONTROL	UART1_TXD_INPUTENABLE	UART1_TXD_PULLTYPESELECT	UART1_TXD_PULLUDENABLE	RESERVED								UART1_TXD_MUXMODE							

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	UART1_TXD_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x1
18	UART1_TXD_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	UART1_TXD_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x1
16	UART1_TXD_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:4	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
3:0	UART1_TXD_MUXMODE	0x0: uart1_txd 0x4: spi4_d0 0xA: gpmc_a13 0xC: mcan_rx 0xE: gpio4_14 0xF: Driver off	RW	0xF

Table 13-819. Register Call Summary for Register CTRL_CORE_PAD_UART1_TXD

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-820. CTRL_CORE_PAD_UART1_CTSN

Address Offset	0x0000 15C4		
Physical Address	0x4A00 35C4	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												UART1_CTSN_SLEWCONTROL	UART1_CTSN_INPUTENABLE	UART1_CTSN_PULLTYPESELECT	UART1_CTSN_PULLUDENABLE	RESERVED								UART1_CTSN_MUXMODE							

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	UART1_CTSN_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	UART1_CTSN_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	UART1_CTSN_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x1
16	UART1_CTSN_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:4	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
3:0	UART1_CTSN_MUXMODE	0x0: uart1_ctsn 0x1: xref_clk1 0x2: uart3_rxd 0x3: gpmc_a16 0x4: spi4_sclk 0x5: spi1_cs2 0x6: timer3 0x7: ehrpwm1_synci 0x8: clkout0 0x9: vin2a_hsync0 0xA: gpmc_a12 0xB: gpmc_clk 0xC: dcan_tx 0xE: gpio4_15 0xF: Driver off	RW	0xF

Table 13-821. Register Call Summary for Register CTRL_CORE_PAD_UART1_CTSN

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-822. CTRL_CORE_PAD_UART1_RTSN

Address Offset	0x0000 15C8		
Physical Address	0x4A00 35C8	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												UART1_RTSN_SLEWCONTROL	UART1_RTSN_INPUTENABLE	UART1_RTSN_PULLTYPESELECT	UART1_RTSN_PULLUDENABLE	RESERVED								UART1_RTSN_MUXMODE							

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	UART1_RTSN_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	UART1_RTSN_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	UART1_RTSN_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x1

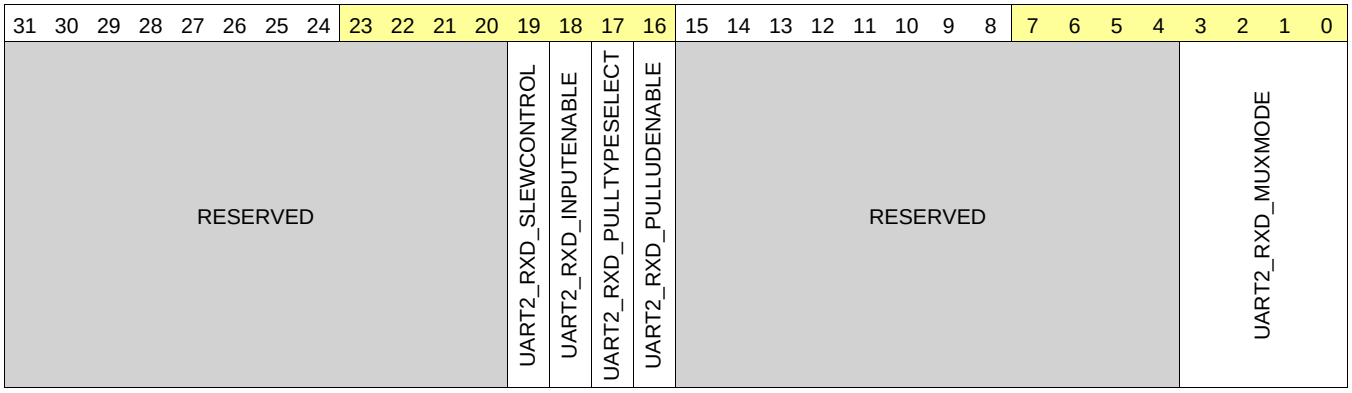
Bits	Field Name	Description	Type	Reset
16	UART1_RTSN_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:4	RESERVED		R	0x0
3:0	UART1_RTSN_MUXMODE	0x0: uart1_rtsn 0x2: uart3_txd 0x3: gpmc_a17 0x4: spi4_cs0 0x5: spi1_cs3 0x6: timer4 0x7: ehrpwm1_synco 0x8: qspi1_rtclk 0x9: vin2a_vsync0 0xA: gpmc_a13 0xC: dcan_rx 0xE: gpio4_16 0xF: Driver off	RW	0xF

Table 13-823. Register Call Summary for Register CTRL_CORE_PAD_UART1_RTSN

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-824. CTRL_CORE_PAD_UART2_RXD

Address Offset	0x0000 15CC		
Physical Address	0x4A00 35CC	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		



Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	UART2_RXD_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	UART2_RXD_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1

Bits	Field Name	Description	Type	Reset
17	UART2_RXD_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x1
16	UART2_RXD_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:4	RESERVED		R	0x0
3:0	UART2_RXD_MUXMODE	0x0: uart2_rxd 0x4: spi3_d1 0x6: timer1 0x7: ehrpwm1A 0xA: gpmc_clk 0xB: gpmc_a12 0xC: dcan_tx 0xE: gpio4_17 0xF: Driver off	RW	0xF

Table 13-825. Register Call Summary for Register CTRL_CORE_PAD_UART2_RXD

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-826. CTRL_CORE_PAD_UART2_TXD

Address Offset	0x0000 15D0		
Physical Address	0x4A00 35D0	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												UART2_TXD_SLEWCONTROL	UART2_TXD_INPUTENABLE	UART2_TXD_PULLTYPESELECT	UART2_TXD_PULLUDENABLE	RESERVED								UART2_TXD_MUXMODE							

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	UART2_TXD_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	UART2_TXD_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	UART2_TXD_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x1

Bits	Field Name	Description	Type	Reset
16	UART2_TXD_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:4	RESERVED		R	0x0
3:0	UART2_TXD_MUXMODE	0x0: uart2_txd 0x4: spi3_d0 0x6: timer2 0x7: ehrpwm1B 0xB: gpmc_a13 0xC: dcan_rx 0xE: gpio4_18 0xF: Driver off	RW	0xF

Table 13-827. Register Call Summary for Register CTRL_CORE_PAD_UART2_TXD

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-828. CTRL_CORE_PAD_UART2_CTSN

Address Offset	0x0000 15D4		
Physical Address	0x4A00 35D4	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												UART2_CTSN_SLEWCONTROL	UART2_CTSN_INPUTENABLE	UART2_CTSN_PULLTYPESELECT	UART2_CTSN_PULLUDENABLE	RESERVED								UART2_CTSN_MUXMODE							

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	UART2_CTSN_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	UART2_CTSN_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	UART2_CTSN_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x1
16	UART2_CTSN_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:4	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
3:0	UART2_CTSN_MUXMODE	0x0: uart2_ctsn 0x2: xref_clk1 0x3: gpmc_a18 0x4: spi3_sclk 0x5: qspi1_cs1 0x6: timer7 0x9: vin2a_hsync0 0xA: gpmc_clk 0xC: mcan_tx 0xE: gpio4_19 0xF: Driver off	RW	0xF

Table 13-829. Register Call Summary for Register CTRL_CORE_PAD_UART2_CTSN

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-830. CTRL_CORE_PAD_UART2_RTSN

Address Offset	0x0000 15D8		
Physical Address	0x4A00 35D8	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												UART2_RTSN_SLEWCONTROL	UART2_RTSN_INPUTENABLE	UART2_RTSN_PULLTYPESELECT	UART2_RTSN_PULLUDENABLE	RESERVED								UART2_RTSN_MUXMODE							

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	UART2_RTSN_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	UART2_RTSN_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	UART2_RTSN_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x1
16	UART2_RTSN_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:4	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
3:0	UART2_RTSN_MUXMODE	0x0: uart2_rtsn 0x1: eCAP1_in_PWM1_out 0x3: gpmc_a19 0x4: spi3_cs0 0x6: timer8 0x9: vin2a_vsync0 0xC: mcan_rx 0xE: gpio4_20 0xF: Driver off	RW	0xF

Table 13-831. Register Call Summary for Register CTRL_CORE_PAD_UART2_RTSN

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-832. CTRL_CORE_PAD_I2C1_SDA

Address Offset	0x0000 15DC		
Physical Address	0x4A00 35DC	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED													I2C1_SDA_INPUTENABLE			RESERVED															

Bits	Field Name	Description	Type	Reset
31:19	RESERVED		R	0x0
18	I2C1_SDA_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17:0	RESERVED		R	0x0

Table 13-833. Register Call Summary for Register CTRL_CORE_PAD_I2C1_SDA

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-834. CTRL_CORE_PAD_I2C1_SCL

Address Offset	0x0000 15E0		
Physical Address	0x4A00 35E0	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED													I2C1_SCL_INPUTENABLE			RESERVED															

Bits	Field Name	Description	Type	Reset
31:19	RESERVED		R	0x0
18	I2C1_SCL_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17:0	RESERVED		R	0x0

Table 13-835. Register Call Summary for Register CTRL_CORE_PAD_I2C1_SCL

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-836. CTRL_CORE_PAD_I2C2_SDA

Address Offset	0x0000 15E4		
Physical Address	0x4A00 35E4	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED													I2C2_SDA_INPUTENABLE			RESERVED															

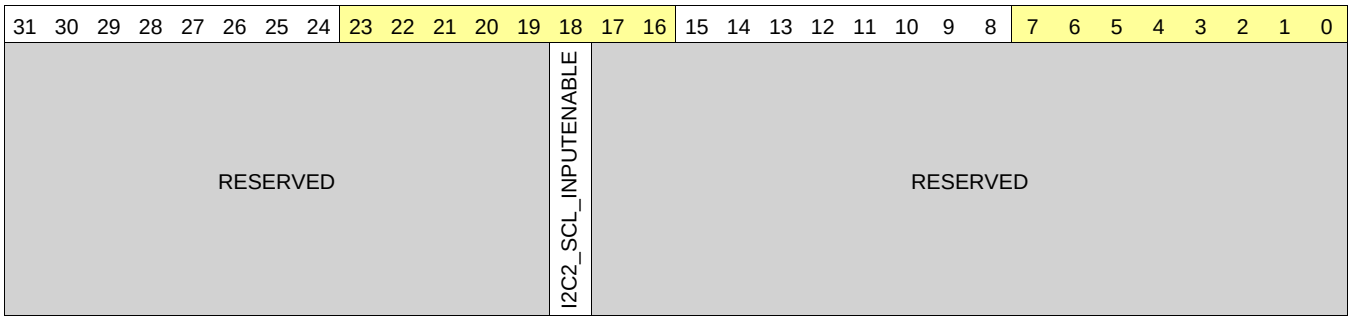
Bits	Field Name	Description	Type	Reset
31:19	RESERVED		R	0x0
18	I2C2_SDA_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17:0	RESERVED		R	0x0

Table 13-837. Register Call Summary for Register CTRL_CORE_PAD_I2C2_SDA

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-838. CTRL_CORE_PAD_I2C2_SCL

Address Offset	0x0000 15E8		
Physical Address	0x4A00 35E8	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		



Bits	Field Name	Description	Type	Reset
31:19	RESERVED		R	0x0
18	I2C2_SCL_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17:0	RESERVED		R	0x0

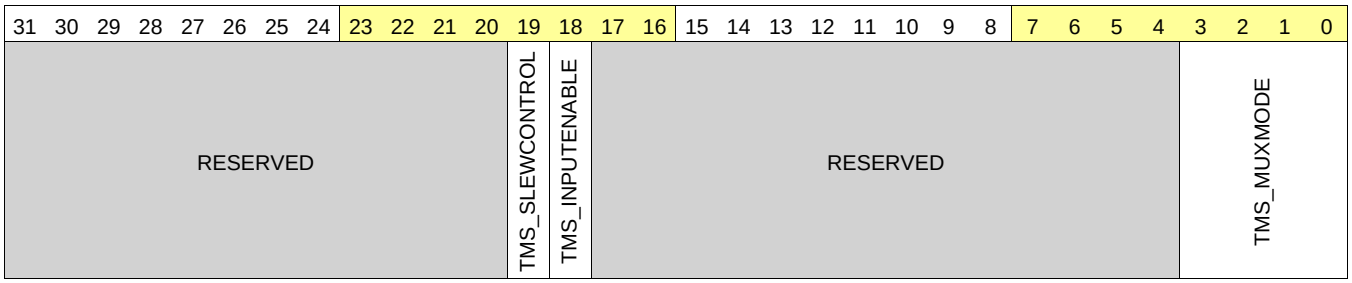
Table 13-839. Register Call Summary for Register CTRL_CORE_PAD_I2C2_SCL

Control Module Register Manual

- CTRL_MODULE_CORE Register Summary: [0]

Table 13-840. CTRL_CORE_PAD_TMS

Address Offset	0x0000 15EC		
Physical Address	0x4A00 35EC	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		



Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	TMS_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	TMS_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1

Bits	Field Name	Description	Type	Reset
17:4	RESERVED		R	0x0
3:0	TMS_MUXMODE	0x0: tms	RW	0x0

Table 13-841. Register Call Summary for Register CTRL_CORE_PAD_TMS

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-842. CTRL_CORE_PAD_TDI

Address Offset	0x0000 15F0		
Physical Address	0x4A00 35F0	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												TDI_SLEWCONTROL	TDI_INPUTENABLE	TDI_PULLTYPESELECT	TDI_PULLUDENABLE	RESERVED								TDI_MUXMODE							

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	TDI_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x1
18	TDI_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	TDI_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x1
16	TDI_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:4	RESERVED		R	0x0
3:0	TDI_MUXMODE	0x0: tdi 0xE: gpio4_25 0xF: Driver off	RW	0x0

Table 13-843. Register Call Summary for Register CTRL_CORE_PAD_TDI

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-844. CTRL_CORE_PAD_TDO

Address Offset	0x0000 15F4		
Physical Address	0x4A00 35F4	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												TDO_SLEWCONTROL	TDO_INPUTENABLE	TDO_PULLTYPESELECT	TDO_PULLUDENABLE	RESERVED								TDO_MUXMODE							

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	TDO_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	TDO_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	TDO_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x1
16	TDO_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:4	RESERVED		R	0x0
3:0	TDO_MUXMODE	0x0: tdo 0xE: gpio4_26 0xF: Driver off	RW	0x0

Table 13-845. Register Call Summary for Register CTRL_CORE_PAD_TDO

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-846. CTRL_CORE_PAD_TCLK

Address Offset	0x0000 15F8		
Physical Address	0x4A00 35F8	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED													TCLK_INPUTENABLE	TCLK_PULLTYPESELECT	TCLK_PULLUDENABLE	RESERVED												TCLK_MUXMODE			

Bits	Field Name	Description	Type	Reset
31:19	RESERVED		R	0x0
18	TCLK_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	TCLK_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x1
16	TCLK_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:4	RESERVED		R	0x0
3:0	TCLK_MUXMODE	0x0: tclk	RW	0x0

Table 13-847. Register Call Summary for Register CTRL_CORE_PAD_TCLK

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-848. CTRL_CORE_PAD_TRSTN

Address Offset	0x0000 15FC		
Physical Address	0x4A00 35FC	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												TRSTN_SLEWCONTROL	TRSTN_INPUTENABLE	TRSTN_PULLTYPESELECT	TRSTN_PULLUDENABLE	RESERVED															

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	TRSTN_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0

Bits	Field Name	Description	Type	Reset
18	TRSTN_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	TRSTN_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	TRSTN_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:0	RESERVED		R	0x0

Table 13-849. Register Call Summary for Register CTRL_CORE_PAD_TRSTN

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-850. CTRL_CORE_PAD_RTCK

Address Offset	0x0000 1600		
Physical Address	0x4A00 3600	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												RTCK_SLEWCONTROL	RTCK_INPUTENABLE	RTCK_PULLTYPESELECT	RTCK_PULLUDENABLE	RESERVED								RTCK_MUXMODE							

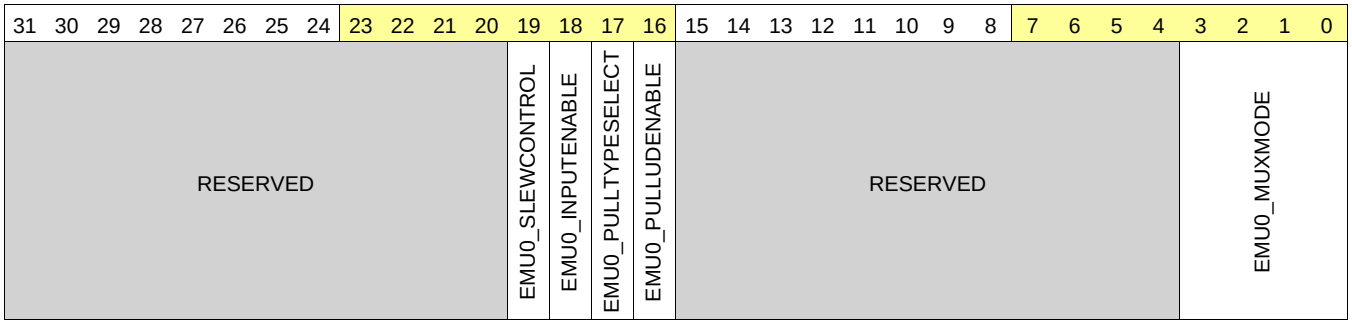
Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	RTCK_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	RTCK_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	RTCK_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	RTCK_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x1
15:4	RESERVED		R	0x0
3:0	RTCK_MUXMODE	0x0: rtck 0xE: gpio4_27 0xF: Driver off	RW	0x0

Table 13-851. Register Call Summary for Register CTRL_CORE_PAD_RTCK

Control Module Register Manual
• CTRL_MODULE_CORE Register Summary: [0]

Table 13-852. CTRL_CORE_PAD_EMU0

Address Offset	0x0000 1604		
Physical Address	0x4A00 3604	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		



Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	EMU0_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	EMU0_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	EMU0_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x1
16	EMU0_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:4	RESERVED		R	0x0
3:0	EMU0_MUXMODE	0x0: emu0 0xE: gpio4_28 0xF: Driver off	RW	0x0

Table 13-853. Register Call Summary for Register CTRL_CORE_PAD_EMU0

Control Module Register Manual
• CTRL_MODULE_CORE Register Summary: [0]

Table 13-854. CTRL_CORE_PAD_EMU1

Address Offset	0x0000 1608		
Physical Address	0x4A00 3608	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												EMU1_SLEWCONTROL	EMU1_INPUTENABLE	EMU1_PULLTYPESELECT	EMU1_PULLUDENABLE	RESERVED								EMU1_MUXMODE							

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19	EMU1_SLEWCONTROL	0x0: Fast slew is selected 0x1: Slow slew is selected	RW	0x0
18	EMU1_INPUTENABLE	0x0: Receive mode is disabled 0x1: Receive mode is enabled	RW	0x1
17	EMU1_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x1
16	EMU1_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:4	RESERVED		R	0x0
3:0	EMU1_MUXMODE	0x0: emu1 0xE: gpio4_29 0xF: Driver off	RW	0x0

Table 13-855. Register Call Summary for Register CTRL_CORE_PAD_EMU1

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-856. CTRL_CORE_PAD_RESETN

Address Offset	0x0000 160C		
Physical Address	0x4A00 360C	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														RESETN_PULLTYPESELECT	RESETN_PULLUDENABLE	RESERVED															

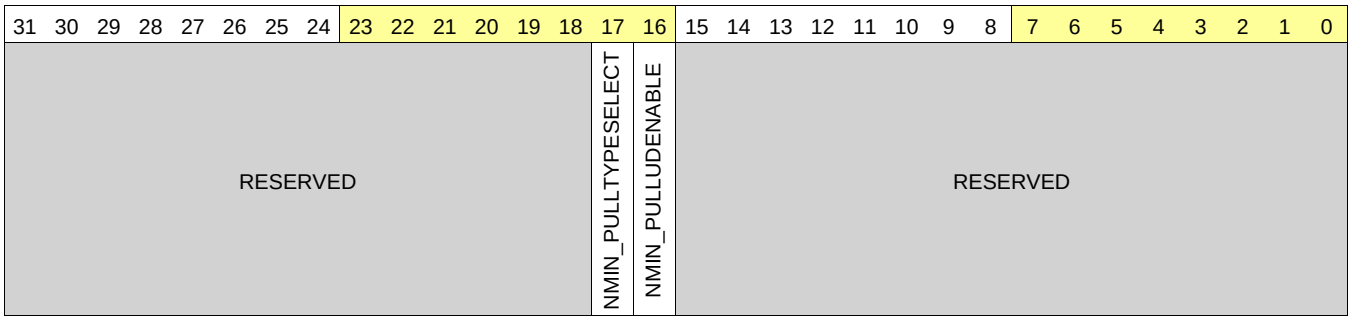
Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17	RESETN_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x1
16	RESETN_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:0	RESERVED		R	0x0

Table 13-857. Register Call Summary for Register CTRL_CORE_PAD_RESETN

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-858. CTRL_CORE_PAD_NMIN

Address Offset	0x0000 1610		
Physical Address	0x4A00 3610	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		



Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17	NMIN_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	NMIN_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:0	RESERVED		R	0x0

Table 13-859. Register Call Summary for Register CTRL_CORE_PAD_NMIN

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-860. CTRL_CORE_PAD_RSTOUTN

Address Offset	0x0000 1614		
Physical Address	0x4A00 3614	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														RSTOUTN_PULLTYPESELECT	RSTOUTN_PULLUDENABLE	RESERVED															

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17	RSTOUTN_PULLTYPESELECT	0x0: Pull Down is selected 0x1: Pull Up is selected	RW	0x0
16	RSTOUTN_PULLUDENABLE	0x0: Enables weak Pull Up/Down 0x1: Disables weak Pull Up/Down	RW	0x0
15:0	RESERVED		R	0x0

Table 13-861. Register Call Summary for Register CTRL_CORE_PAD_RSTOUTN

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-862. CTRL_CORE_STD_FUSE_OPP_VDD_DSPEVE_LVT_0

Address Offset	0x0000 1B38		
Physical Address	0x4A00 3B38	**Instance**	CTRL_MODULE_CORE
Description	Standard Fuse OPP VDD_GPU [31:0]. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STD_FUSE_OPP_VDD_DSPEVE_LVT_0																															

Bits	Field Name	Description	Type	Reset
31:0	STD_FUSE_OPP_VDD_DSPEVE_LVT_0		R	0x0

Table 13-863. Register Call Summary for Register CTRL_CORE_STD_FUSE_OPP_VDD_DSPEVE_LVT_0

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-864. CTRL_CORE_STD_FUSE_OPP_VDD_DSPEVE_LVT_1

Address Offset	0x0000 1B3C
Physical Address	0x4A00 3B3C **Instance** CTRL_MODULE_CORE
Description	Standard Fuse OPP VDD_GPU [63:32]. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.
Type	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STD_FUSE_OPP_VDD_DSPEVE_LVT_1																															

Bits	Field Name	Description	Type	Reset
31:0	STD_FUSE_OPP_VDD_DSPEVE_LVT_1		R	0x0

Table 13-865. Register Call Summary for Register CTRL_CORE_STD_FUSE_OPP_VDD_DSPEVE_LVT_1

Control Module Register Manual

- CTRL_MODULE_CORE Register Summary: [0]

Table 13-866. CTRL_CORE_STD_FUSE_OPP_VDD_DSPEVE_LVT_2

Address Offset	0x0000 1B40
Physical Address	0x4A00 3B40 **Instance** CTRL_MODULE_CORE
Description	Standard Fuse OPP VDD_GPU [95:64]. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.
Type	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STD_FUSE_OPP_VDD_DSPEVE_LVT_2																															

Bits	Field Name	Description	Type	Reset
31:0	STD_FUSE_OPP_VDD_DSPEVE_LVT_2		R	0x0

Table 13-867. Register Call Summary for Register CTRL_CORE_STD_FUSE_OPP_VDD_DSPEVE_LVT_2

Control Module Register Manual

- CTRL_MODULE_CORE Register Summary: [0]

Table 13-868. CTRL_CORE_STD_FUSE_OPP_VDD_DSPEVE_LVT_3

Address Offset	0x0000 1B44
Physical Address	0x4A00 3B44 **Instance** CTRL_MODULE_CORE
Description	Standard Fuse OPP VDD_GPU [127:96]. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.
Type	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STD_FUSE_OPP_VDD_DSPEVE_LVT_3																															

Bits	Field Name	Description	Type	Reset
31:0	STD_FUSE_OPP_VDD_DSPEVE_LVT_3		R	0x0

Table 13-869. Register Call Summary for Register CTRL_CORE_STD_FUSE_OPP_VDD_DSPEVE_LVT_3

Control Module Register Manual

- CTRL_MODULE_CORE Register Summary: [0]

Table 13-870. CTRL_CORE_STD_FUSE_OPP_VDD_DSPEVE_LVT_4

Address Offset	0x0000 1B48		
Physical Address	0x4A00 3B48	**Instance**	CTRL_MODULE_CORE
Description	Standard Fuse OPP VDD_GPU [159:128]. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STD_FUSE_OPP_VDD_DSPEVE_LVT_4																															

Bits	Field Name	Description	Type	Reset
31:0	STD_FUSE_OPP_VDD_DSPEVE_LVT_4		R	0x0

Table 13-871. Register Call Summary for Register CTRL_CORE_STD_FUSE_OPP_VDD_DSPEVE_LVT_4

Control Module Register Manual

- CTRL_MODULE_CORE Register Summary: [0]

Table 13-872. CTRL_CORE_STD_FUSE_OPP_VDD_CORE_LVT_0

Address Offset	0x0000 1B60		
Physical Address	0x4A00 3B60	**Instance**	CTRL_MODULE_CORE
Description	Standard Fuse OPP VDD_CORE [31:0]. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STD_FUSE_OPP_VDD_CORE_LVT_0																															

Bits	Field Name	Description	Type	Reset
31:0	STD_FUSE_OPP_VDD_CORE_LVT_0		R	0x0

Table 13-873. Register Call Summary for Register CTRL_CORE_STD_FUSE_OPP_VDD_CORE_LVT_0

Control Module Register Manual

- CTRL_MODULE_CORE Register Summary: [0]

Table 13-874. CTRL_CORE_STD_FUSE_OPP_VDD_CORE_LVT_1

Address Offset	0x0000 1B64
Physical Address	0x4A00 3B64 **Instance** CTRL_MODULE_CORE
Description	Standard Fuse OPP VDD_CORE [63:32]. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.
Type	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STD_FUSE_OPP_VDD_CORE_LVT_1																															

Bits	Field Name	Description	Type	Reset
31:0	STD_FUSE_OPP_VDD_CORE_LVT_1		R	0x0

Table 13-875. Register Call Summary for Register CTRL_CORE_STD_FUSE_OPP_VDD_CORE_LVT_1

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-876. CTRL_CORE_STD_FUSE_OPP_VDD_CORE_LVT_2

Address Offset	0x0000 1B68
Physical Address	0x4A00 3B68 **Instance** CTRL_MODULE_CORE
Description	Standard Fuse OPP VDD_CORE [95:64]. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.
Type	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STD_FUSE_OPP_VDD_CORE_LVT_2																															

Bits	Field Name	Description	Type	Reset
31:0	STD_FUSE_OPP_VDD_CORE_LVT_2		R	0x0

Table 13-877. Register Call Summary for Register CTRL_CORE_STD_FUSE_OPP_VDD_CORE_LVT_2

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-878. CTRL_CORE_STD_FUSE_OPP_VDD_CORE_LVT_3

Address Offset	0x0000 1B6C
Physical Address	0x4A00 3B6C **Instance** CTRL_MODULE_CORE
Description	Standard Fuse OPP VDD_CORE [127:96]. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.
Type	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STD_FUSE_OPP_VDD_CORE_LVT_3																															

Bits	Field Name	Description	Type	Reset
31:0	STD_FUSE_OPP_VDD_CORE_LVT_3		R	0x0

Table 13-879. Register Call Summary for Register CTRL_CORE_STD_FUSE_OPP_VDD_CORE_LVT_3

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-880. CTRL_CORE_STD_FUSE_OPP_VDD_CORE_LVT_4

Address Offset	0x0000 1B70		
Physical Address	0x4A00 3B70	**Instance**	CTRL_MODULE_CORE
Description	Standard Fuse OPP VDD_CORE [159:128]. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STD_FUSE_OPP_VDD_CORE_LVT_4																															

Bits	Field Name	Description	Type	Reset
31:0	STD_FUSE_OPP_VDD_CORE_LVT_4		R	0x0

Table 13-881. Register Call Summary for Register CTRL_CORE_STD_FUSE_OPP_VDD_CORE_LVT_4

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-882. CTRL_CORE_LDOSRAM_CORE_4_VOLTAGE_CTRL

Address Offset	0x0000 1B74		
Physical Address	0x4A00 3B74	**Instance**	CTRL_MODULE_CORE
Description	CORE 4th SRAM LDO Control register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED					LDOSRAMCORE_4_RETMODE_MUX_CTRL	LDOSRAMCORE_4_RETMODE_VSET_IN						LDOSRAMCORE_4_RETMODE_VSET_OUT				RESERVED					LDOSRAMCORE_4_ACTMODE_MUX_CTRL	LDOSRAMCORE_4_ACTMODE_VSET_IN						LDOSRAMCORE_4_ACTMODE_VSET_OUT						

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26	LDOSRAMCORE_4_RETMODE_MUX_CTRL	Override control of EFUSE Retention Mode Voltage value 0x0: eFuse value is used 0x1: Override value is used	RW	0x0
25:21	LDOSRAMCORE_4_RETMODE_VSET_IN	EFUSE Retention Mode Voltage value (vset[9:5])	R	0x0
20:16	LDOSRAMCORE_4_RETMODE_VSET_OUT	Override value for Retention Mode Voltage	RW	0x0
15:11	RESERVED		R	0x0
10	LDOSRAMCORE_4_ACTMODE_MUX_CTRL	Override control of EFUSE Active Mode Voltage value 0x0: eFuse value is used 0x1: Override value is used	RW	0x0
9:5	LDOSRAMCORE_4_ACTMODE_VSET_IN	EFUSE Active Mode Voltage value (vset[4:0])	R	0x0
4:0	LDOSRAMCORE_4_ACTMODE_VSET_OUT	Override value for Active Mode Voltage value	RW	0x0

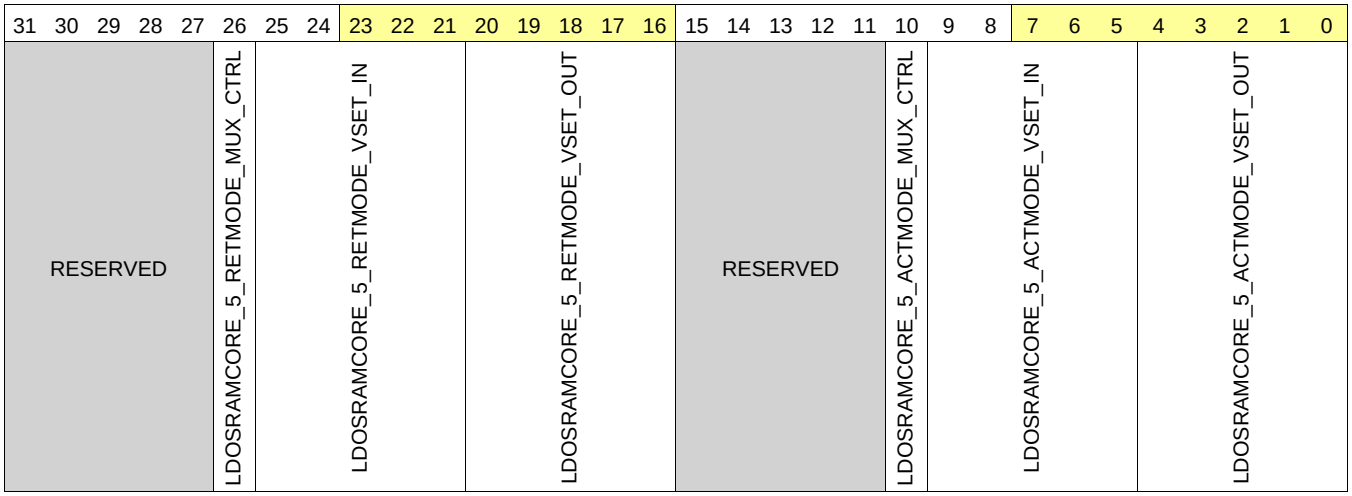
Table 13-883. Register Call Summary for Register CTRL_CORE_LDOSRAM_CORE_4_VOLTAGE_CTRL

Control Module Register Manual

- CTRL_MODULE_CORE Register Summary: [0]

Table 13-884. CTRL_CORE_LDOSRAM_CORE_5_VOLTAGE_CTRL

Address Offset	0x0000 1B78		
Physical Address	0x4A00 3B78	**Instance**	CTRL_MODULE_CORE
Description	CORE 5th SRAM LDO Control register		
Type	RW		



Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26	LDOSRAMCORE_5_RETMODE_MUX_CTRL	Override control of EFUSE Retention Mode Voltage value 0x0: eFuse value is used 0x1: Override value is used	RW	0x0
25:21	LDOSRAMCORE_5_RETMODE_VSET_IN	EFUSE Retention Mode Voltage value (vset[9:5])	R	0x0

Bits	Field Name	Description	Type	Reset
20:16	LDOSRAMCORE_5_RETMODE_VSET_OUT	Override value for Retention Mode Voltage	RW	0x0
15:11	RESERVED		R	0x0
10	LDOSRAMCORE_5_ACTMODE_MUX_CTRL	Override control of EFUSE Active Mode Voltage value 0x0: eFuse value is used 0x1: Override value is used	RW	0x0
9:5	LDOSRAMCORE_5_ACTMODE_VSET_IN	EFUSE Active Mode Voltage value (vset[4:0])	R	0x0
4:0	LDOSRAMCORE_5_ACTMODE_VSET_OUT	Override value for Active Mode Voltage value	RW	0x0

Table 13-885. Register Call Summary for Register CTRL_CORE_LDOSRAM_CORE_5_VOLTAGE_CTRL

Control Module Register Manual

- CTRL_MODULE_CORE Register Summary: [0]

Table 13-886. CTRL_CORE_LDOSRAM_DSPEVE_2_VOLTAGE_CTRL

Address Offset	0x0000 1B7C		
Physical Address	0x4A00 3B7C	**Instance**	CTRL_MODULE_CORE
Description	DSPEVE 2nd SRAM LDO Control register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED					LDOSRAMDSPEVE_2_RETMODE_MUX_CTRL	LDOSRAMDSPEVE_2_RETMODE_VSET_IN					LDOSRAMDSPEVE_2_RETMODE_VSET_OUT					RESERVED					LDOSRAMDSPEVE_2_ACTMODE_MUX_CTRL	LDOSRAMDSPEVE_2_ACTMODE_VSET_IN						LDOSRAMDSPEVE_2_ACTMODE_VSET_OUT					

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26	LDOSRAMDSPEVE_2_RETMODE_MUX_CTRL	Override control of EFUSE Retention Mode Voltage value 0x0: eFuse value is used 0x1: Override value is used	RW	0x0
25:21	LDOSRAMDSPEVE_2_RETMODE_VSET_IN	EFUSE Retention Mode Voltage value (vset[9:5])	R	0x0
20:16	LDOSRAMDSPEVE_2_RETMODE_VSET_OUT	Override value for Retention Mode Voltage	RW	0x0
15:11	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
10	LDOSRAMDSPEVE_2_ACTMODE_MUX_CTRL	Override control of EFUSE Active Mode Voltage value 0x0: eFuse value is used 0x1: Override value is used	RW	0x0
9:5	LDOSRAMDSPEVE_2_ACTMODE_VSET_IN	EFUSE Active Mode Voltage value (vset[4:0])	R	0x0
4:0	LDOSRAMDSPEVE_2_ACTMODE_VSET_OUT	Override value for Active Mode Voltage value	RW	0x0

Table 13-887. Register Call Summary for Register CTRL_CORE_LDOSRAM_DSPEVE_2_VOLTAGE_CTRL

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-888. CTRL_CORE_SMA_SW_2

Address Offset	0x0000 1C04		
Physical Address	0x4A00 3C04	**Instance**	CTRL_MODULE_CORE
Description	OCP Spare Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMA_SW_2																															

Bits	Field Name	Description	Type	Reset
31:0	SMA_SW_2	OCP spare register	RW	0x0

Table 13-889. Register Call Summary for Register CTRL_CORE_SMA_SW_2

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-890. CTRL_CORE_SMA_SW_3

Address Offset	0x0000 1C08		
Physical Address	0x4A00 3C08	**Instance**	CTRL_MODULE_CORE
Description	OCP Spare Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMA_SW_3																															

Bits	Field Name	Description	Type	Reset
31:0	SMA_SW_3	OCP spare register	RW	0x0

Table 13-891. Register Call Summary for Register CTRL_CORE_SMA_SW_3

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-892. CTRL_CORE_SMA_SW_6

Address Offset	0x0000 1C14		
Physical Address	0x4A00 3C14	**Instance**	CTRL_MODULE_CORE
Description	OCP Spare Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							RMII_CLK_SETTING	RESERVED							MUXSEL_32K_CLKIN

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8	RMII_CLK_SETTING	RMII CLK setting 0x0: Internal clock from DPLL_GMAC_DSP 0x1: Reserved	RW	0x0
7:1	RESERVED		R	0x0
0	MUXSEL_32K_CLKIN	Setting for mux to select 32KHz clock input to PRCM. This bit must NOT be modified by software. The 32kHz clock selection is done through the device sysboot[9:8] signals.	RW	0x0

Table 13-893. Register Call Summary for Register CTRL_CORE_SMA_SW_6

Control Module Register Manual

- CTRL_MODULE_CORE Register Summary: [0]

Table 13-894. CTRL_CORE_SMA_SW_7

Address Offset	0x0000 1C18		
Physical Address	0x4A00 3C18	**Instance**	CTRL_MODULE_CORE
Description	OCP Spare Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														MMU1_ABORT_ENABLE	RESERVED			EDMA_TC1_WR_MMU_ROUTE_ENABLE	EDMA_TC1_RD_MMU_ROUTE_ENABLE	EDMA_TC0_WR_MMU_ROUTE_ENABLE	EDMA_TC0_RD_MMU_ROUTE_ENABLE	RESERVED									

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17	MMU1_ABORT_ENABLE	MMU1 abort enable	RW	0x0
16:14	RESERVED		R	0x0
13	EDMA_TC1_WR_MMU_ROUTE_ENABLE	EDMA TC1 WR traffic MMU route enable	RW	0x0
12	EDMA_TC1_RD_MMU_ROUTE_ENABLE	EDMA TC1 RD traffic MMU route enable	RW	0x0
11	EDMA_TC0_WR_MMU_ROUTE_ENABLE	EDMA TC0 WR traffic MMU route enable	RW	0x0
10	EDMA_TC0_RD_MMU_ROUTE_ENABLE	EDMA TC0 RD traffic MMU route enable	RW	0x0
9:0	RESERVED		R	0x0

Table 13-895. Register Call Summary for Register CTRL_CORE_SMA_SW_7

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-896. CTRL_CORE_FIREWALL_CONNID_CONTROL_0

Address Offset	0x0000 1C48		
Physical Address	0x4A00 3C48	**Instance**	CTRL_MODULE_CORE
Description	Firewall ConnID control register		
Type	RW		

31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
MMU1_CONNID	EDMA_TC0_RD_CONNID	EDMA_TC0_WR_CONNID	EDMA_TC1_RD_CONNID	EDMA_TC1_WR_CONNID	VIP1_P0_CONNID	VIP1_P1_CONNID	EVE1_TC0_CONNID

Bits	Field Name	Description	Type	Reset
31:28	MMU1_CONNID		RW	0x8
27:24	EDMA_TC0_RD_CONNID		RW	0x7
23:20	EDMA_TC0_WR_CONNID		RW	0x7
19:16	EDMA_TC1_RD_CONNID		RW	0x7
15:12	EDMA_TC1_WR_CONNID		RW	0x7
11:8	VIP1_P0_CONNID		RW	0x9
7:4	VIP1_P1_CONNID		RW	0x9
3:0	EVE1_TC0_CONNID		RW	0x4

Table 13-897. Register Call Summary for Register CTRL_CORE_FIREWALL_CONNID_CONTROL_0

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-898. CTRL_CORE_FIREWALL_CONNID_CONTROL_1

Address Offset	0x0000 1C4C		
Physical Address	0x4A00 3C4C	**Instance**	CTRL_MODULE_CORE
Description	Firewall ConnID control register		
Type	RW		

31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
EVE1_TC1_CONNID	DSP1_EDMA_CONNID	DSP1_MDMA_CONNID	DSP2_EDMA_CONNID	DSP2_MDMA_CONNID	DSS_CONNID	RESERVED	IPU1_CONNID

Bits	Field Name	Description	Type	Reset
31:28	EVE1_TC1_CONNID		RW	0xD
27:24	DSP1_EDMA_CONNID		RW	0x2
23:20	DSP1_MDMA_CONNID		RW	0x2
19:16	DSP2_EDMA_CONNID		RW	0x2
15:12	DSP2_MDMA_CONNID		RW	0x3
11:8	DSS_CONNID		RW	0x8
7:4	RESERVED		R	0x0
3:0	IPU1_CONNID		RW	0x6

Table 13-899. Register Call Summary for Register CTRL_CORE_FIREWALL_CONNID_CONTROL_1

Control Module Register Manual

- CTRL_MODULE_CORE Register Summary: [0]

Table 13-900. CTRL_CORE_FIREWALL_CONNID_CONTROL_2

Address Offset	0x0000 1C50		
Physical Address	0x4A00 3C50	**Instance**	CTRL_MODULE_CORE
Description	Firewall ConnID control register		
Type	RW		

31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
RESERVED	DSP1_CFG_CONNID	DSP2_CFG_CONNID	GMAC_SW_CONNID	RESERVED	

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:24	DSP1_CFG_CONNID		RW	0x2
23:20	DSP2_CFG_CONNID		RW	0x3

Bits	Field Name	Description	Type	Reset
19:16	GMAC_SW_CONNID		RW	0xA
15:0	RESERVED		R	0x11

Table 13-901. Register Call Summary for Register CTRL_CORE_FIREWALL_CONNID_CONTROL_2

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-902. CTRL_CORE_FIREWALL_CONNID_CONTROL_3

Address Offset	0x0000 1C54		
Physical Address	0x4A00 3C54	**Instance**	CTRL_MODULE_CORE
Description	Firewall ConnID control register		
Type	RW		

31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4 3 2 1 0
RESERVED	ISS_RT_CONNID	RESERVED	ISS_NRT1_CONNID	ISS_NRT2_CONNID	RESERVED	

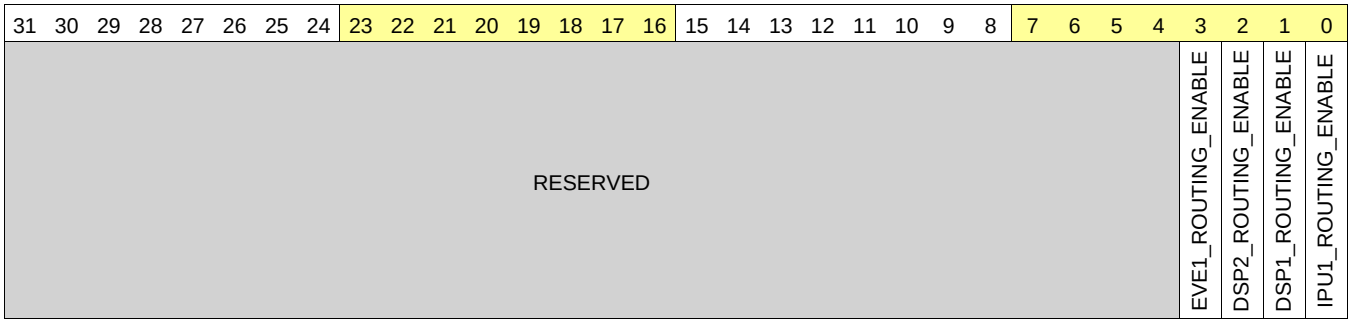
Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:24	ISS_RT_CONNID	Note: ISS is not supported on the DRA78x family of devices.	RW	0xC
23:20	RESERVED		R	0x0
19:16	ISS_NRT1_CONNID		RW	0xC
15:12	ISS_NRT2_CONNID		RW	0xC
11:0	RESERVED		R	0x0

Table 13-903. Register Call Summary for Register CTRL_CORE_FIREWALL_CONNID_CONTROL_3

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-904. CTRL_CORE_EMIF_MPU_ROUTING

Address Offset	0x0000 1C58		
Physical Address	0x4A00 3C58	**Instance**	CTRL_MODULE_CORE
Description	EMIF MPU traffic control register		
Type	RW		



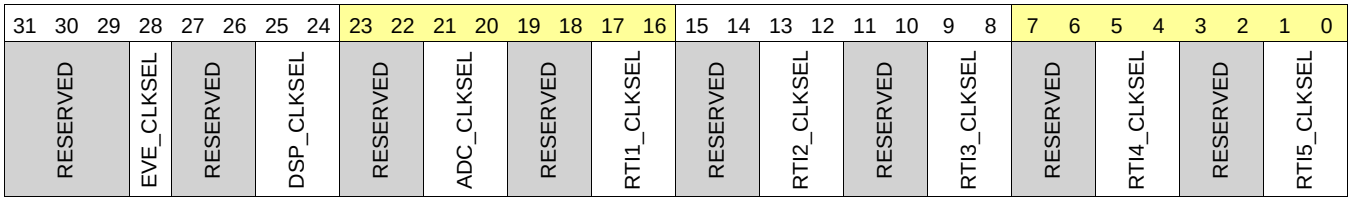
Bits	Field Name	Description	Type	Reset
31:4	RESERVED		R	0x0
3	EVE1_ROUTING_ENABLE		RW	0x0
2	DSP2_ROUTING_ENABLE		RW	0x0
1	DSP1_ROUTING_ENABLE		RW	0x0
0	IPU1_ROUTING_ENABLE		RW	0x0

Table 13-905. Register Call Summary for Register CTRL_CORE_EMIF_MPU_ROUTING

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-906. CTRL_CORE_PRCM_CLKSEL_CONTROL

Address Offset	0x0000 1C5C		
Physical Address	0x4A00 3C5C	**Instance**	CTRL_MODULE_CORE
Description	PRCM Clock selection control register		
Type	RW		



Bits	Field Name	Description	Type	Reset
31:29	RESERVED		R	0x0
28	EVE_CLKSEL	EVE Clock selection control 0x0: EVE1_GCLK 0x1: EVE2_GCLK	RW	0x0
27:26	RESERVED		R	0x0
25:24	DSP_CLKSEL	DSP Clock selection control 0x0: DSP2_CLK 0x1: DSP1_CLK 0x3: RESERVED 0x2: DSP0_CLK	RW	0x0
23:22	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
21:20	ADC_CLKSEL	ADC Clock selection control 0x0: SYS_CLK1 0x1: SYS_CLK2 0x3: RESERVED 0x2: XREF_CLK	RW	0x0
19:18	RESERVED		R	0x0
17:16	RTI1_CLKSEL	RTI1 Clock selection control 0x0: SYS_CLK1 div 4 0x1: SYS_CLK2 div 4 0x3: RESERVED 0x2: FUNC_32K_CLK	RW	0x2
15:14	RESERVED		R	0x0
13:12	RTI2_CLKSEL	RTI2 Clock selection control 0x0: SYS_CLK1 div 4 0x1: SYS_CLK2 div 4 0x3: RESERVED 0x2: FUNC_32K_CLK	RW	0x2
11:10	RESERVED		R	0x0
9:8	RTI3_CLKSEL	RTI3 Clock selection control 0x0: SYS_CLK1 div 4 0x1: SYS_CLK2 div 4 0x3: RESERVED 0x2: FUNC_32K_CLK	RW	0x2
7:6	RESERVED		R	0x0
5:4	RTI4_CLKSEL	RTI4 Clock selection control 0x0: SYS_CLK1 div 4 0x1: SYS_CLK2 div 4 0x3: RESERVED 0x2: FUNC_32K_CLK	RW	0x2
3:2	RESERVED		R	0x0
1:0	RTI5_CLKSEL	RTI5 Clock selection control 0x0: SYS_CLK1 div 4 0x1: SYS_CLK2 div 4 0x3: RESERVED 0x2: FUNC_32K_CLK	RW	0x2

Table 13-907. Register Call Summary for Register CTRL_CORE_PRCM_CLKSEL_CONTROL

Control Module Register Manual

- CTRL_MODULE_CORE Register Summary: [0]

Table 13-908. CTRL_CORE_PRCM_CLKDIV_CONTROL1

Address Offset	0x0000 1C60		
Physical Address	0x4A00 3C60	**Instance**	CTRL_MODULE_CORE
Description	PRCM Clock divider control1 register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED		DBG_STM_EXPT_CLK_HSDIV_CHANGE_ACK	DBG_STM_EXPT_CLK_HSDIV_EN_ACK	DBG_TRC_EXPT_CLK_HSDIV_CHANGE_ACK	DBG_TRC_EXPT_CLK_HSDIV_EN_ACK	DBG_ATB_CLK_HSDIV_CHANGE_ACK	DBG_ATB_CLK_HSDIV_EN_ACK	DBG_STM_EXPT_CLK_TENABLEDIV_SEL	DBG_STM_EXPT_CLK_TENABLEDIV	DBG_STM_EXPT_CLK_DIV						DBG_TRC_EXPT_CLK_TENABLEDIV_SEL	DBG_TRC_EXPT_CLK_TENABLEDIV	DBG_TRC_EXPT_CLK_DIV						RESERVED	DBG_ATB_CLK_TENABLEDIV	DBG_ATB_CLK_DIV					

Bits	Field Name	Description	Type	Reset
31:30	RESERVED	Read returns 0.	R	0x0
29	DBG_STM_EXPT_CLK_HSDIV_CHANGE_ACK	HSDIVIDER change ack	R	0x0
28	DBG_STM_EXPT_CLK_HSDIV_EN_ACK	EN_ACK from HSDIVIDER	R	0x0
27	DBG_TRC_EXPT_CLK_HSDIV_CHANGE_ACK	HSDIVIDER change ack	R	0x0
26	DBG_TRC_EXPT_CLK_HSDIV_EN_ACK	EN_ACK from HSDIVIDER	R	0x0
25	DBG_ATB_CLK_HSDIV_CHANGE_ACK	HSDIVIDER change ack	R	0x0
24	DBG_ATB_CLK_HSDIV_EN_ACK	EN_ACK from HSDIVIDER	R	0x0
23	DBG_STM_EXPT_CLK_TENABLEDIV_SEL	0: Control from PRCM 1: Control from Control Module - DBG_STM_EXPT_TENABLEDIV_CTRL	RW	0x0
22	DBG_STM_EXPT_CLK_TENABLEDIV	Needs to be pulsed (LO->HI->LO) for latching the divider value in to the DPLL.	RW	0x0
21:16	DBG_STM_EXPT_CLK_DIV	DebugSS STM Clock divider value. This field programs the H21 divider of DPLL_CORE. The actual division is equal to this register value+1. 0x0: H21 = /1 0x1: H21 = /2 **...** 0x3F: H21 = /64	RW	0xA
15	DBG_TRC_EXPT_CLK_TENABLEDIV_SEL	0: Control from PRCM 1: Control from Control Module - DBG_TRC_EXPT_TENABLEDIV_CTRL	RW	0x0
14	DBG_TRC_EXPT_CLK_TENABLEDIV	Needs to be pulsed (LO->HI->LO) for latching the divider value in to the DPLL.	RW	0x0
13:8	DBG_TRC_EXPT_CLK_DIV	DebugSS TRC Clock divider value. This field programs the H14 divider of DPLL_PER. The actual division is equal to this register value+1. 0x0: H14 = /1 0x1: H14 = /2 **...** 0x3F: H14 = /64	RW	0x3
7	RESERVED		RW	0x0

Bits	Field Name	Description	Type	Reset
6	DBG_ATB_CLK_TENABLEDIV	Needs to be pulsed (LO->HI->LO) for latching the divider value in to the DPLL.	RW	0x0
5:0	DBG_ATB_CLK_DIV	DebugSS ATB Clock divider value. This field programs the H11 divider of DPLL_CORE. The actual division is equal to this register value+1. 0x0: H11 = /1 0x1: H11 = /2 ... 0x3F: H11 = /64	RW	0x7

Table 13-909. Register Call Summary for Register CTRL_CORE_PRCM_CLKDIV_CONTROL1

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-910. CTRL_CORE_PRCM_CLKDIV_CONTROL2

Address Offset	0x0000 1C64		
Physical Address	0x4A00 3C64	**Instance**	CTRL_MODULE_CORE
Description	PRCM Clock divider control2 register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED			VID_PIX_CLK_TENABLEDIV_SEL	VID_PIX_CLK_HSDIV_CHANGE_ACK	VID_PIX_CLK_EXT_CLK_DIV		VID_PIX_CLK_HSDIV_EN_ACK	VID_PIX_CLK_HSDIV_EN	VID_PIX_CLK_HSDIV_LATCH_EN	VID_PIX_CLK_HSDIV						RESERVED		TESOC_DBG_ATB_CLK_TENABLEDIV_SEL		TESOC_HSDIV_CHANGE_ACK		TESOC_EXT_CLK_DIV		TESOC_HSDIV_EN_ACK	TESOC_HSDIV_EN	TESOC_HSDIV_LATCH_EN	TESOC_HSDIV				

Bits	Field Name	Description	Type	Reset
31:29	RESERVED	Read returns 0.	R	0x0
28	VID_PIX_CLK_TENABLEDIV_SEL	VID_PIX_CLK_TENABLEDIV control select: 0: Control from PRCM 1: Control from Control Module - VID_PIX_CLK_HSDIV_LATCH_EN	RW	0x0
27	VID_PIX_CLK_HSDIV_CHANGE_ACK	HSDIVIDER change ack	R	0x0
26:25	VID_PIX_CLK_EXT_CLK_DIV	Video Clock divider value 0x0: DIV1 0x1: DIV2 0x2: DIV4 0x3: Reserved	RW	0x0
24	VID_PIX_CLK_HSDIV_EN_ACK	EN_ACK from HSDIVIDER	R	0x0
23	VID_PIX_CLK_HSDIV_EN	Clock enable to the Video Clock HS divider	RW	0x0

Bits	Field Name	Description	Type	Reset
22	VID_PIX_CLK_HSDIV_LATCH_EN	Video Clock HS divider latch enable: To be toggled to latch DIV value in HSDIVIDER	RW	0x0
21:16	VID_PIX_CLK_HSDIV	Video Clock HS divider value. This field programs the HS divider of DPLL_EVE_VID_DSP. The actual division is equal to this register value+1. 0x0: HS = /1 0x1: HS = /2 ... 0x3F: HS = /64	RW	0x10
15:14	RESERVED	Reserved. Read returns 0.	R	0x0
13:12	TESOC_DBG_ATB_CLK_TENABLEDIV_SEL	Mux select for the TENABLE of Core DPLL HS divider 1: 0: PRCM 1: DBG_ATB_CLK 2: TESOC_EXT_CLK_HSDIV 3: Reserved Note: TESOC is not supported on the DRA78x family of devices.	RW	0x0
11	TESOC_HSDIV_CHANGE_ACK	HSDIVIDER change ack	R	0x0
10:9	TESOC_EXT_CLK_DIV	TesOC clock divider value 0x0: DIV1 0x1: DIV2 0x2: DIV4 0x3: Reserved	RW	0x0
8	TESOC_HSDIV_EN_ACK	EN_ACK from HSDIVIDER	R	0x0
7	TESOC_HSDIV_EN	Clock enable to TesOC clock HS divider	RW	0x0
6	TESOC_HSDIV_LATCH_EN	TesOC clock HS divider latch enable: To be toggled to latch DIV value in DPLL	RW	0x0
5:0	TESOC_HSDIV	TesOC clock HS divider value. This field programs the H13 divider of DPLL_CORE. The actual division is equal to this register value+1. Note: TESOC is not supported on the DRA78x family of devices. 0x0: H13 = /1 0x1: H13 = /2 ... 0x3F: H13 = /64	RW	0x3C

Table 13-911. Register Call Summary for Register CTRL_CORE_PRCM_CLKDIV_CONTROL2

Control Module Register Manual

- CTRL_MODULE_CORE Register Summary: [0]

Table 13-912. CTRL_CORE_SMA_SW_10

Address Offset	0x0000 1C68		
Physical Address	0x4A00 3C68	**Instance**	CTRL_MODULE_CORE
Description	OCP Spare Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMA_SW_10		IE_CSI2_0_Y4	IE_CSI2_0_X4	IE_CSI2_0_Y3	IE_CSI2_0_X3		IE_CSI2_0_Y2	IE_CSI2_0_X2	IE_CSI2_0_Y1	IE_CSI2_0_X1	IE_CSI2_0_Y0	IE_CSI2_0_X0	PIPD_CSI2_0_X4	PIPD_CSI2_0_Y3	PIPD_CSI2_0_X3	PIPD_CSI2_0_Y2	PIPD_CSI2_0_X2	PIPD_CSI2_0_Y1	PIPD_CSI2_0_X1	PIPD_CSI2_0_Y0	PIPD_CSI2_0_X0	PIPU_CSI2_0_Y4	PIPU_CSI2_0_X4	PIPU_CSI2_0_Y3	PIPU_CSI2_0_X3	PIPU_CSI2_0_Y2	PIPU_CSI2_0_X2	PIPU_CSI2_0_Y1	PIPU_CSI2_0_X1	PIPU_CSI2_0_Y0	PIPU_CSI2_0_X0

Bits	Field Name	Description	Type	Reset
31:30	SMA_SW_10	spare bits	R	0x0
29	IE_CSI2_0_Y4		RW	0x0
28	IE_CSI2_0_X4		RW	0x0
27	IE_CSI2_0_Y3		RW	0x0
26:25	IE_CSI2_0_X3		RW	0x0
24	IE_CSI2_0_Y2		RW	0x0
23	IE_CSI2_0_X2		RW	0x0
22	IE_CSI2_0_Y1		RW	0x0
21	IE_CSI2_0_X1		RW	0x0
20	IE_CSI2_0_Y0		RW	0x0
19	IE_CSI2_0_X0		RW	0x0
18	PIPD_CSI2_0_X4		RW	0x0
17	PIPD_CSI2_0_Y3		RW	0x0
16	PIPD_CSI2_0_X3		RW	0x0
15	PIPD_CSI2_0_Y2		RW	0x0
14	PIPD_CSI2_0_X2		RW	0x0
13	PIPD_CSI2_0_Y1		RW	0x0
12	PIPD_CSI2_0_X1		RW	0x0
11	PIPD_CSI2_0_Y0		RW	0x0
10	PIPD_CSI2_0_X0		RW	0x0
9	PIPU_CSI2_0_Y4		RW	0x0
8	PIPU_CSI2_0_X4		RW	0x0
7	PIPU_CSI2_0_Y3		RW	0x0
6	PIPU_CSI2_0_X3		RW	0x0
5	PIPU_CSI2_0_Y2		RW	0x0
4	PIPU_CSI2_0_X2		RW	0x0
3	PIPU_CSI2_0_Y1		RW	0x0
2	PIPU_CSI2_0_X1		RW	0x0
1	PIPU_CSI2_0_Y0		RW	0x0
0	PIPU_CSI2_0_X0		RW	0x0

Table 13-913. Register Call Summary for Register CTRL_CORE_SMA_SW_10

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-914. CTRL_CORE_SMA_SW_11

Address Offset	0x0000 1C6C		
Physical Address	0x4A00 3C6C	**Instance**	CTRL_MODULE_CORE
Description	OCP Spare Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMA_SW_11																															

Bits	Field Name	Description	Type	Reset
31:0	SMA_SW_11	OCP Spare Register	RW	0x0

Table 13-915. Register Call Summary for Register CTRL_CORE_SMA_SW_11

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-916. CTRL_CORE_SMA_SW_12

Address Offset	0x0000 1C70		
Physical Address	0x4A00 3C70	**Instance**	CTRL_MODULE_CORE
Description	OCP Spare Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMA_SW_12																															

Bits	Field Name	Description	Type	Reset
31:0	SMA_SW_12	OCP Spare Register	RW	0x0

Table 13-917. Register Call Summary for Register CTRL_CORE_SMA_SW_12

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-918. CTRL_CORE_SMA_SW_13

Address Offset	0x0000 1C74		
Physical Address	0x4A00 3C74	**Instance**	CTRL_MODULE_CORE
Description	OCP Spare Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMA_SW_13																															

Bits	Field Name	Description	Type	Reset
31:0	SMA_SW_13	OCP Spare Register	RW	0x0

Table 13-919. Register Call Summary for Register CTRL_CORE_SMA_SW_13

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-920. CTRL_CORE_TESOC_LAST_RESET_INDICATOR

Address Offset	0x0000 1C78		
Physical Address	0x4A00 3C78	**Instance**	CTRL_MODULE_CORE
Description	TESOC last reset indicator register Note: TESOC is not supported on the DRA78x family of devices.		
Type	RW		

31 30 29 28	27 26 25 24	23 22 21 20	19 18 17 16	15 14 13 12	11 10 9 8	7 6 5 4	3 2 1 0
RESERVED	IPU_LAST_RESET_INDICATOR	RESERVED	DSP1_LAST_RESET_INDICATOR	RESERVED	DSP2_LAST_RESET_INDICATOR	RESERVED	EVE_LAST_RESET_INDICATOR

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:24	IPU_LAST_RESET_INDICATOR	IPU resert indicator. IPU writes into this bit field when it initiates self-test. IPU reads this bit field after coming up from reset after field-test and understands that the reset was due to TESOC. Software can choose to do partial boot using this indicator. For details on TESOC, refer to Tester On Chip (TESOC) in the device Safety Subsystem.	RW	0x0
23:20	RESERVED		R	0x0
19:16	DSP1_LAST_RESET_INDICATOR	DSP1 resert indicator. DSP1 writes into this bit field when it initiates self-test. DSP1 reads this bit field after coming up from reset after field-test and understands that the reset was due to TESOC. Software can choose to do partial boot using this indicator. For details on TESOC, refer to Tester On Chip (TESOC) in the device Safety Subsystem.	RW	0x0
15:12	RESERVED		R	0x0
11:8	DSP2_LAST_RESET_INDICATOR	DSP2 resert indicator. DSP2 writes into this bit field when it initiates self-test. DSP reads this bit field after coming up from reset after field-test and understands that the reset was due to TESOC. Software can choose to do partial boot using this indicator. For details on TESOC, refer to Tester On Chip (TESOC) in the device Safety Subsystem.	RW	0x0
7:4	RESERVED		R	0x0
3:0	EVE_LAST_RESET_INDICATOR	EVE resert indicator. EVE writes into this bit field when it initiates self-test. EVE reads this bit field after coming up from reset after field-test and understands that the reset was due to TESOC. Software can choose to do partial boot using this indicator. For details on TESOC, refer to Tester On Chip (TESOC) in the device Safety Subsystem.	RW	0x0

Table 13-921. Register Call Summary for Register CTRL_CORE_TESOC_LAST_RESET_INDICATOR

Control Module Functional Description • Control Module Resets: [0]
Control Module Register Manual • CTRL_MODULE_CORE Register Summary: [1]

Table 13-922. CTRL_CORE_SD_DAC_CONTROL

Address Offset	0x0000 1C7C		
Physical Address	0x4A00 3C7C	**Instance**	CTRL_MODULE_CORE
Description	SD DAC control register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SD_DAC_CALSET								SD_DAC_CAL								CTL										CTL_WR_ACK	CTL_ASYNC_EN	TVOUTBYPASS	ACEN	RESERVED			

Bits	Field Name	Description	Type	Reset
31	SD_DAC_CALSET	0x0: Data from VENC is provided to SD_DAC 0x1: Data from the SD_DAC_CAL bit field is provided to SD_DAC	RW	0x0
30:21	SD_DAC_CAL	10-bit data for SD_DAC test or debug purposes.	RW	0x0
20:6	CTL	SD_DAC Control interface for reconfiguration of the module in functional mode through internal configuration registers.	RW	0x0
5	CTL_WR_ACK	Used only for synchronous mode of the CTL interface. It toggles high and then low when an asynchronous write on the CTL bus has been acknowledged by the SD_DAC module.	RW W1toClr	0x0
4	CTL_ASYNC_EN	Asynchronous mode enable for CTL interface. 0x0: Synchronous mode 0x1: Asynchronous mode	RW	0x0
3	TVOUTBYPASS	TVOUT Bypass enable. 0x0: Normal mode. Video Buffer is not bypassed 0x1: Bypass mode. Video Buffer is bypassed	RW	0x0
2	ACEN	AC coupling enable. 0x0: DC coupling selected 0x1: AC coupling selected	RW	0x0
1:0	RESERVED		R	0x0

Table 13-923. Register Call Summary for Register CTRL_CORE_SD_DAC_CONTROL

Control Module Register Manual • CTRL_MODULE_CORE Register Summary: [0]

Table 13-924. CTRL_CORE_SD_DAC_TRIM_VALUE

Address Offset	0x0000 1C80		
Physical Address	0x4A00 3C80	**Instance**	CTRL_MODULE_CORE
Description	SD DAC trim value		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
COMP_EN	RESERVED															SD_DAC_TRIM															

Bits	Field Name	Description	Type	Reset
31	COMP_EN	Optional control for lower output swing. 0x0: High full-scale output swing 0x1: Low full-scale output swing	RW	0x0
30:23	RESERVED		R	0x0
22:0	SD_DAC_TRIM	SD DAC trim value	RW	0x0

Table 13-925. Register Call Summary for Register CTRL_CORE_SD_DAC_TRIM_VALUE

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-926. CTRL_CORE_ADC_ERROR_OFFSET

Address Offset	0x0000 1C84		
Physical Address	0x4A00 3C84	**Instance**	CTRL_MODULE_CORE
Description	ADC Error offset register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								ERROR_OFFSET							

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x0
7:0	ERROR_OFFSET	Error offset value driven from efuse with optional override	RW	0x0

Table 13-927. Register Call Summary for Register CTRL_CORE_ADC_ERROR_OFFSET

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-928. CTRL_CORE_IPU_WAKEUP

Address Offset	0x0000 1C88		
Physical Address	0x4A00 3C88	**Instance**	CTRL_MODULE_CORE
Description	IPU wakeup enable		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															IPU_WKUP_EN

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	IPU_WKUP_EN	IPU wakeup enable	RW	0x1

Table 13-929. Register Call Summary for Register CTRL_CORE_IPU_WAKEUP

Control Module Functional Description • Control Module Resets: [0]
Control Module Register Manual • CTRL_MODULE_CORE Register Summary: [1]

Table 13-930. CTRL_CORE_ISS_EFUSE

Address Offset	0x0000 1C8C		
Physical Address	0x4A00 3C8C	**Instance**	CTRL_MODULE_CORE
Description	ADC Error offset register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								STD_FUSE_ISS_EFUSE1_EN	STD_FUSE_ISS_EFUSE2_EN	STD_FUSE_ISS_EFUSE3_EN	STD_FUSE_ISS_EFUSE4_EN	STD_FUSE_ISS_EFUSE5_EN	STD_FUSE_ISS_EFUSE6_EN	STD_FUSE_ISS_EFUSE7_EN	

Bits	Field Name	Description	Type	Reset
31:7	RESERVED		R	0x0
6	STD_FUSE_ISS_EFUSE1_EN	Error offset value driven from efuse with optional override	RW	0x0
5	STD_FUSE_ISS_EFUSE2_EN	Error offset value driven from efuse with optional override	RW	0x0
4	STD_FUSE_ISS_EFUSE3_EN	Error offset value driven from efuse with optional override	RW	0x0
3	STD_FUSE_ISS_EFUSE4_EN	Error offset value driven from efuse with optional override	RW	0x0
2	STD_FUSE_ISS_EFUSE5_EN	Error offset value driven from efuse with optional override	RW	0x0
1	STD_FUSE_ISS_EFUSE6_EN	Error offset value driven from efuse with optional override	RW	0x0
0	STD_FUSE_ISS_EFUSE7_EN	Error offset value driven from efuse with optional override	RW	0x0

Table 13-931. Register Call Summary for Register CTRL_CORE_ISS_EFUSE

Control Module Register Manual • CTRL_MODULE_CORE Register Summary: [0]

Table 13-932. CTRL_CORE_SMA_SW_14

Address Offset	0x0000 1C90		
Physical Address	0x4A00 3C90	**Instance**	CTRL_MODULE_CORE
Description	This register controls McASP2, McASP3 and ATL signal muxing.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MCASP_MODE_17_VOUT1_D22_MUX_CONTROL_0	MCASP_MODE_17_VOUT1_D21_MUX_CONTROL_0	MCASP_MODE_17_VOUT1_D20_MUX_CONTROL_0	MCASP_MODE_17_VOUT1_D19_MUX_CONTROL_0	MCASP_MODE_17_VOUT1_D18_MUX_CONTROL_0	MCASP_MODE_17_VOUT1_D17_MUX_CONTROL_0	MCASP_MODE_17_VOUT1_D16_MUX_CONTROL_0	MCASP_MODE_17_VOUT1_D15_MUX_CONTROL_0	MCASP_MODE_17_VOUT1_D14_MUX_CONTROL_0	MCASP_MODE_17_VOUT1_D13_MUX_CONTROL_0	MCASP_MODE_17_GPMC_AD7_MUX_CONTROL_0	MCASP_MODE_17_VIN1A_D5_MUX_CONTROL_0	MCASP_MODE_17_VIN1A_D4_MUX_CONTROL_0	MCASP_MODE_17_VIN1A_D3_MUX_CONTROL_0	MCASP_MODE_17_VIN1A_D2_MUX_CONTROL_0	MCASP_MODE_17_VIN1A_D1_MUX_CONTROL_0	MCASP_MODE_17_VIN1A_D0_MUX_CONTROL_0	MCASP_MODE_17_VIN1A_VSYNC0_MUX_CONTROL_0	MCASP_MODE_17_VIN1A_HSYNC0_MUX_CONTROL_0	MCASP_MODE_17_VIN1A_FLD0_MUX_CONTROL_0	MCASP_MODE_17_VIN1A_DE0_MUX_CONTROL_0	MCASP_MODE_17_VIN1A_CLK0_MUX_CONTROL_0	MCASP_MODE_17_GPMC_AD15_MUX_CONTROL_0	MCASP_MODE_17_GPMC_AD14_MUX_CONTROL_0	MCASP_MODE_17_GPMC_AD13_MUX_CONTROL_0	MCASP_MODE_17_GPMC_AD12_MUX_CONTROL_0	MCASP_MODE_17_GPMC_AD11_MUX_CONTROL_0	MCASP_MODE_17_GPMC_AD10_MUX_CONTROL_0	MCASP_MODE_17_GPMC_AD9_MUX_CONTROL_0	MCASP_MODE_17_GPMC_AD8_MUX_CONTROL_0	MCASP_MODE_17_GPMC_AD6_MUX_CONTROL_0	MCASP_MODE_17_GPMC_AD5_MUX_CONTROL_0

Bits	Field Name	Description	Type	Reset
31	MCASP_MODE_17_VOUT1_D22_MUX_CONTROL_0	0x0: Function 17 for pad vout1_d22 not selected 0x1: mcasp2_axr4 selected on pad vout1_d22 as function 17 **NOTE:** When this bit is set to 0x1, the MUXMODE field of the corresponding CTRL_CORE_PAD_x register must be set to 0xF.	RW	0x0
30	MCASP_MODE_17_VOUT1_D21_MUX_CONTROL_0	0x0: Function 17 for pad vout1_d21 not selected 0x1: mcasp2_axr3 selected on pad vout1_d21 as function 17 **NOTE:** When this bit is set to 0x1, the MUXMODE field of the corresponding CTRL_CORE_PAD_x register must be set to 0xF.	RW	0x0
29	MCASP_MODE_17_VOUT1_D20_MUX_CONTROL_0	0x0: Function 17 for pad vout1_d20 not selected 0x1: mcasp2_axr2 selected on pad vout1_d20 as function 17 **NOTE:** When this bit is set to 0x1, the MUXMODE field of the corresponding CTRL_CORE_PAD_x register must be set to 0xF.	RW	0x0
28	MCASP_MODE_17_VOUT1_D19_MUX_CONTROL_0	0x0: Function 17 for pad vout1_d19 not selected 0x1: mcasp2_axr1 selected on pad vout1_d19 as function 17 **NOTE:** When this bit is set to 0x1, the MUXMODE field of the corresponding CTRL_CORE_PAD_x register must be set to 0xF.	RW	0x0
27	MCASP_MODE_17_VOUT1_D18_MUX_CONTROL_0	0x0: Function 17 for pad vout1_d18 not selected 0x1: mcasp2_axr0 selected on pad vout1_d18 as function 17 **NOTE:** When this bit is set to 0x1, the MUXMODE field of the corresponding CTRL_CORE_PAD_x register must be set to 0xF.	RW	0x0
26	MCASP_MODE_17_VOUT1_D17_MUX_CONTROL_0	0x0: Function 17 for pad vout1_d17 not selected 0x1: mcasp2_fsr selected on pad vout1_d17 as function 17 **NOTE:** When this bit is set to 0x1, the MUXMODE field of the corresponding CTRL_CORE_PAD_x register must be set to 0xF.	RW	0x0

Bits	Field Name	Description	Type	Reset
25	MCASP_MODE_17_VOUT1_D16_MUX_CONTROL_0	0x0: Function 17 for pad vout1_d16 not selected 0x1: atl_clk0 selected on pad vout1_d16 as function 17 **NOTE:** When this bit is set to 0x1, the MUXMODE field of the corresponding CTRL_CORE_PAD_x register must be set to 0xF.	RW	0x0
24	MCASP_MODE_17_VOUT1_D15_MUX_CONTROL_0	0x0: Function 17 for pad vout1_d15 not selected 0x1: mcasp2_fsx selected on pad vout1_d15 as function 17 **NOTE:** When this bit is set to 0x1, the MUXMODE field of the corresponding CTRL_CORE_PAD_x register must be set to 0xF.	RW	0x0
23	MCASP_MODE_17_VOUT1_D14_MUX_CONTROL_0	0x0: Function 17 for pad vout1_d14 not selected 0x1: mcasp2_aclkx selected on pad vout1_d14 as function 17 **NOTE:** When this bit is set to 0x1, the MUXMODE field of the corresponding CTRL_CORE_PAD_x register must be set to 0xF.	RW	0x0
22	MCASP_MODE_17_VOUT1_D13_MUX_CONTROL_0	0x0: Function 17 for pad vout1_d13 not selected 0x1: mcasp2_aclkr selected on pad vout1_d13 as function 17 **NOTE:** When this bit is set to 0x1, the MUXMODE field of the corresponding CTRL_CORE_PAD_x register must be set to 0xF.	RW	0x0
21	MCASP_MODE_17_GPMC_AD7_MUX_CONTROL_0	0x0: Function 17 for pad gpmc_ad7 not selected 0x1: mcasp2_ahclkx selected on pad gpmc_ad7 as function 17 **NOTE:** When this bit is set to 0x1, the MUXMODE field of the corresponding CTRL_CORE_PAD_x register must be set to 0xF.	RW	0x0
20	MCASP_MODE_17_VIN1A_D5_MUX_CONTROL_0	0x0: Function 17 for pad vin1a_d5 not selected 0x1: mcasp3_ahclkx selected on pad vin1a_d5 as function 17 **NOTE:** When this bit is set to 0x1, the MUXMODE field of the corresponding CTRL_CORE_PAD_x register must be set to 0xF.	RW	0x0
19	MCASP_MODE_17_VIN1A_D4_MUX_CONTROL_0	0x0: Function 17 for pad vin1a_d4 not selected 0x1: mcasp3_axr5 selected on pad vin1a_d4 as function 17 **NOTE:** When this bit is set to 0x1, the MUXMODE field of the corresponding CTRL_CORE_PAD_x register must be set to 0xF.	RW	0x0
18	MCASP_MODE_17_VIN1A_D3_MUX_CONTROL_0	0x0: Function 17 for pad vin1a_d3 not selected 0x1: mcasp3_axr4 selected on pad vin1a_d3 as function 17 **NOTE:** When this bit is set to 0x1, the MUXMODE field of the corresponding CTRL_CORE_PAD_x register must be set to 0xF.	RW	0x0
17	MCASP_MODE_17_VIN1A_D2_MUX_CONTROL_0	0x0: Function 17 for pad vin1a_d2 not selected 0x1: mcasp3_axr3 selected on pad vin1a_d2 as function 17 **NOTE:** When this bit is set to 0x1, the MUXMODE field of the corresponding CTRL_CORE_PAD_x register must be set to 0xF.	RW	0x0

Bits	Field Name	Description	Type	Reset
16	MCASP_MODE_17_VIN1A_D1_MUX_CONTROL_0	0x0: Function 17 for pad vin1a_d1 not selected 0x1: mcasp3_axr2 selected on pad vin1a_d1 as function 17 **NOTE:** When this bit is set to 0x1, the MUXMODE field of the corresponding CTRL_CORE_PAD_x register must be set to 0xF.	RW	0x0
15	MCASP_MODE_17_VIN1A_D0_MUX_CONTROL_0	0x0: Function 17 for pad vin1a_d0 not selected 0x1: mcasp3_axr1 selected on pad vin1a_d0 as function 17 **NOTE:** When this bit is set to 0x1, the MUXMODE field of the corresponding CTRL_CORE_PAD_x register must be set to 0xF.	RW	0x0
14	MCASP_MODE_17_VIN1A_VSYNC0_MUX_CONTROL_0	0x0: Function 17 for pad vin1a_vsync0 not selected 0x1: mcasp3_axr0 selected on pad vin1a_vsync0 as function 17 **NOTE:** When this bit is set to 0x1, the MUXMODE field of the corresponding CTRL_CORE_PAD_x register must be set to 0xF.	RW	0x0
13	MCASP_MODE_17_VIN1A_HSYNC0_MUX_CONTROL_0	0x0: Function 17 for pad vin1a_hsync0 not selected 0x1: mcasp3_fsr selected on pad vin1a_hsync0 as function 17 **NOTE:** When this bit is set to 0x1, the MUXMODE field of the corresponding CTRL_CORE_PAD_x register must be set to 0xF.	RW	0x0
12	MCASP_MODE_17_VIN1A_FLD0_MUX_CONTROL_0	0x0: Function 17 for pad vin1a_fld0 not selected 0x1: mcasp3_aclkr selected on pad vin1a_fld0 as function 17 **NOTE:** When this bit is set to 0x1, the MUXMODE field of the corresponding CTRL_CORE_PAD_x register must be set to 0xF.	RW	0x0
11	MCASP_MODE_17_VIN1A_DE0_MUX_CONTROL_0	0x0: Function 17 for pad vin1a_de0 not selected 0x1: atl_clk1 selected on pad vin1a_de0 as function 17 **NOTE:** When this bit is set to 0x1, the MUXMODE field of the corresponding CTRL_CORE_PAD_x register must be set to 0xF.	RW	0x0
10	MCASP_MODE_17_VIN1A_CLK0_MUX_CONTROL_0	0x0: Function 17 for pad vin1a_clk0 not selected 0x1: mcasp3_aclkx selected on pad vin1a_clk0 as function 17 **NOTE:** When this bit is set to 0x1, the MUXMODE field of the corresponding CTRL_CORE_PAD_x register must be set to 0xF.	RW	0x0
9	MCASP_MODE_17_GPMC_AD15_MUX_CONTROL_0	0x0: Function 17 for pad gpmc_ad15 not selected 0x1: mcasp2_axr5 selected on pad gpmc_ad15 as function 17 **NOTE:** When this bit is set to 0x1, the MUXMODE field of the corresponding CTRL_CORE_PAD_x register must be set to 0xF.	RW	0x0
8	MCASP_MODE_17_GPMC_AD14_MUX_CONTROL_0	0x0: Function 17 for pad gpmc_ad14 not selected 0x1: mcasp2_axr4 selected on pad gpmc_ad14 as function 17 **NOTE:** When this bit is set to 0x1, the MUXMODE field of the corresponding CTRL_CORE_PAD_x register must be set to 0xF.	RW	0x0

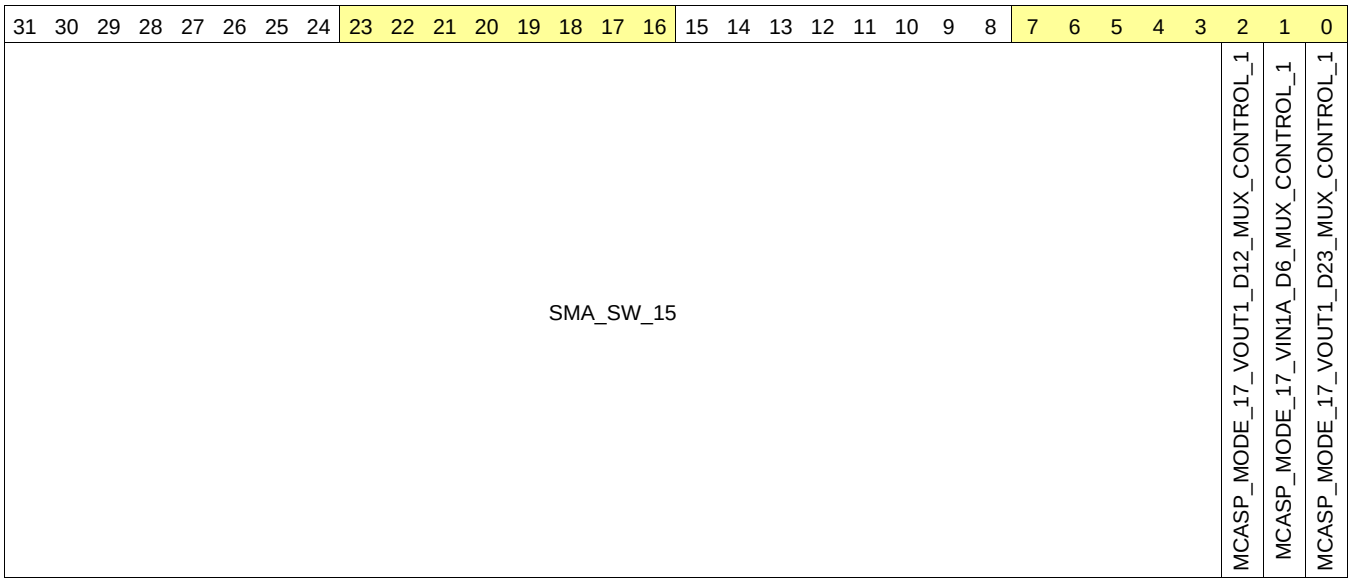
Bits	Field Name	Description	Type	Reset
7	MCASP_MODE_17_GPMC_AD13_MUX_CONTROL_0	0x0: Function 17 for pad gpmc_ad13 not selected 0x1: mcasp2_axr3 selected on pad gpmc_ad13 as function 17 **NOTE:** When this bit is set to 0x1, the MUXMODE field of the corresponding CTRL_CORE_PAD_x register must be set to 0xF.	RW	0x0
6	MCASP_MODE_17_GPMC_AD12_MUX_CONTROL_0	0x0: Function 17 for pad gpmc_ad12 not selected 0x1: mcasp2_axr2 selected on pad gpmc_ad12 as function 17 **NOTE:** When this bit is set to 0x1, the MUXMODE field of the corresponding CTRL_CORE_PAD_x register must be set to 0xF.	RW	0x0
5	MCASP_MODE_17_GPMC_AD11_MUX_CONTROL_0	0x0: Function 17 for pad gpmc_ad11 not selected 0x1: mcasp2_axr1 selected on pad gpmc_ad11 as function 17 **NOTE:** When this bit is set to 0x1, the MUXMODE field of the corresponding CTRL_CORE_PAD_x register must be set to 0xF.	RW	0x0
4	MCASP_MODE_17_GPMC_AD10_MUX_CONTROL_0	0x0: Function 17 for pad gpmc_ad10 not selected 0x1: mcasp2_axr0 selected on pad gpmc_ad10 as function 17 **NOTE:** When this bit is set to 0x1, the MUXMODE field of the corresponding CTRL_CORE_PAD_x register must be set to 0xF.	RW	0x0
3	MCASP_MODE_17_GPMC_AD9_MUX_CONTROL_0	0x0: Function 17 for pad gpmc_ad9 not selected 0x1: mcasp2_fsr selected on pad gpmc_ad9 as function 17 **NOTE:** When this bit is set to 0x1, the MUXMODE field of the corresponding CTRL_CORE_PAD_x register must be set to 0xF.	RW	0x0
2	MCASP_MODE_17_GPMC_AD8_MUX_CONTROL_0	0x0: Function 17 for pad gpmc_ad8 not selected 0x1: mcasp2_aclkr selected on pad gpmc_ad8 as function 17 **NOTE:** When this bit is set to 0x1, the MUXMODE field of the corresponding CTRL_CORE_PAD_x register must be set to 0xF.	RW	0x0
1	MCASP_MODE_17_GPMC_AD6_MUX_CONTROL_0	0x0: Function 17 for pad gpmc_ad6 not selected 0x1: mcasp2_fsx selected on pad gpmc_ad6 as function 17 **NOTE:** When this bit is set to 0x1, the MUXMODE field of the corresponding CTRL_CORE_PAD_x register must be set to 0xF.	RW	0x0
0	MCASP_MODE_17_GPMC_AD5_MUX_CONTROL_0	0x0: Function 17 for pad gpmc_ad5 not selected 0x1: mcasp2_aclkx selected on pad gpmc_ad5 as function 17 **NOTE:** When this bit is set to 0x1, the MUXMODE field of the corresponding CTRL_CORE_PAD_x register must be set to 0xF.	RW	0x0

Table 13-933. Register Call Summary for Register CTRL_CORE_SMA_SW_14

Control Module Functional Description • Pad Configuration Registers: [0]
Control Module Register Manual • CTRL_MODULE_CORE Register Summary: [65]

Table 13-934. CTRL_CORE_SMA_SW_15

Address Offset	0x0000 1C94		
Physical Address	0x4A00 3C94	**Instance**	CTRL_MODULE_CORE
Description	Several bits of this register control McASP2 and McASP3 signal muxing.		
Type	RW		



Bits	Field Name	Description	Type	Reset
31:3	SMA_SW_15	OCP Spare Bits	RW	0x0
2	MCASP_MODE_17_VOUT1_D12_MUX_CONTROL_1	0x0: Function 17 for pad vout1_d12 not selected 0x1: mcasp2_ahclkx selected on pad vout1_d12 as function 17 **NOTE:** When this bit is set to 0x1, the MUXMODE field of the corresponding CTRL_CORE_PAD_x register must be set to 0xF.	RW	0x0
1	MCASP_MODE_17_VIN1A_D6_MUX_CONTROL_1	0x0: Function 17 for pad vin1a_d6 not selected 0x1: mcasp3_fsx selected on pad vin1a_d6 as function 17 **NOTE:** When this bit is set to 0x1, the MUXMODE field of the corresponding CTRL_CORE_PAD_x register must be set to 0xF.	RW	0x0
0	MCASP_MODE_17_VOUT1_D23_MUX_CONTROL_1	0x0: Function 17 for pad vout1_d23 not selected 0x1: mcasp2_axr5 selected on pad vout1_d23 as function 17 **NOTE:** When this bit is set to 0x1, the MUXMODE field of the corresponding CTRL_CORE_PAD_x register must be set to 0xF.	RW	0x0

Table 13-935. Register Call Summary for Register CTRL_CORE_SMA_SW_15

Control Module Functional Description • Pad Configuration Registers: [0]
Control Module Register Manual • CTRL_MODULE_CORE Register Summary: [7]

Table 13-936. CTRL_CORE_SMA_SW_16

Address Offset	0x0000 1C98		
Physical Address	0x4A00 3C98	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMA_SW_16																															CORE_DPLL_INPUT_CLK_SELECTION

Bits	Field Name	Description	Type	Reset
31:1	SMA_SW_16	OCP Spare Register	RW	0x0
0	CORE_DPLL_INPUT_CLK_SELECTION	DPLL_CORE input clock selection. 0x0: SYS_CLK1 selected as input clock (REF_CLK) for DPLL_CORE 0x1: SYS_CLK2 selected as input clock (REF_CLK) for DPLL_CORE	RW	0x0

Table 13-937. Register Call Summary for Register CTRL_CORE_SMA_SW_16

Control Module Register Manual

- CTRL_MODULE_CORE Register Summary: [0]

Table 13-938. CTRL_CORE_SMA_SW_17

Address Offset	0x0000 1C9C		
Physical Address	0x4A00 3C9C	**Instance**	CTRL_MODULE_CORE
Description	This register holds some controls and status bits for H14 HS divider of DPLL_GMAC_DSP which is the source of MCAN functional clock (MCAN_CLK).		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMA_SW_17																				MCAN_CLK_HSDIV_CHANGE_ACK	MCAN_CLK_HSDIV_EN_ACK	MCAN_CLK_TENABLEDIV_SEL	RESERVED	MCAN_CLK_HSDIV_EN	MCAN_CLK_HSDIV_LATCH_EN	MCAN_CLK_HSDIV							

Bits	Field Name	Description	Type	Reset
31:12	SMA_SW_17	OCP Spare Bits	RW	0x0
11	MCAN_CLK_HSDIV_CHANGE_ACK	Acknowledge flag which indicates that the on-the-fly change of H14 divider of DPLL_GMAC_DSP is done. This bit changes its state (0 to 1 or 1 to 0) on each on-the-fly change of H14.	R	0x1
10	MCAN_CLK_HSDIV_EN_ACK	Indicates whether CLKOUTX2_H14 of DPLL_GMAC_DSP is enabled or not. 0x0: CLKOUTX2_H14 is disabled 0x1: CLKOUTX2_H14 is enabled	R	0x0
9	MCAN_CLK_TENABLEDIV_SEL	TENABLEDIV control select for H14 of DPLL_GMAC_DSP 0x0: Control from PRCM 0x1: Control from Control Module (bit CTRL_CORE_SMA_SW_17[6] MCAN_CLK_HSDIV_LATCH_EN)	RW	0x0
8	RESERVED	Reserved	RW	0x0
7	MCAN_CLK_HSDIV_EN	Output clock (CLKOUTX2_H14) enable for H14 of DPLL_GMAC_DSP 0x0: Output clock disabled 0x1: Output clock enabled	RW	0x0
6	MCAN_CLK_HSDIV_LATCH_EN	To be toggled (LO->HI->LO) to latch MCAN_CLK_HSDIV value in H14 of DPLL_GMAC_DSP. This bit takes effect only when bit CTRL_CORE_SMA_SW_17[9] MCAN_CLK_TENABLEDIV_SEL is set to 0x1.	RW	0x0
5:0	MCAN_CLK_HSDIV	This field programs the H14 divider of DPLL_GMAC_DSP. The actual division is equal to this register value+1. 0x0: H14 = /1 0x1: H14 = /2 ... 0x3F: H14 = /64	RW	0x18

Table 13-939. Register Call Summary for Register CTRL_CORE_SMA_SW_17

Control Module Register Manual

- CTRL_MODULE_CORE Register Summary: [0]
- CTRL_MODULE_CORE Register Description: [1][2]

Table 13-940. CTRL_CORE_ROM_CPU0_BRANCH

Address Offset	0x0000 1CA0		
Physical Address	0x4A00 3CA0	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ROM_CPU0_BRANCH																															

Bits	Field Name	Description	Type	Reset
31:0	ROM_CPU0_BRANCH		RW	0x0

Table 13-941. Register Call Summary for Register CTRL_CORE_ROM_CPU0_BRANCH

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-942. CTRL_CORE_ROM_CPU1_BRANCH

Address Offset	0x0000 1CA4		
Physical Address	0x4A00 3CA4	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ROM_CPU1_BRANCH																															

Bits	Field Name	Description	Type	Reset
31:0	ROM_CPU1_BRANCH		RW	0x0

Table 13-943. Register Call Summary for Register CTRL_CORE_ROM_CPU1_BRANCH

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-944. CTRL_CORE_ROM_AUXBOOT0

Address Offset	0x0000 1CA8		
Physical Address	0x4A00 3CA8	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ROM_AUXBOOT0																															

Bits	Field Name	Description	Type	Reset
31:0	ROM_AUXBOOT0		RW	0x0

Table 13-945. Register Call Summary for Register CTRL_CORE_ROM_AUXBOOT0

Control Module Functional Description
• Control Module Resets: [0]
Control Module Register Manual
• CTRL_MODULE_CORE Register Summary: [1]

Table 13-946. CTRL_CORE_ROM_AUXBOOT1

Address Offset	0x0000 1CAC		
Physical Address	0x4A00 3CAC	**Instance**	CTRL_MODULE_CORE
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ROM_AUXBOOT1																															

Bits	Field Name	Description	Type	Reset
31:0	ROM_AUXBOOT1		RW	0x0

Table 13-947. Register Call Summary for Register CTRL_CORE_ROM_AUXBOOT1

Control Module Functional Description
• Control Module Resets: [0]
Control Module Register Manual
• CTRL_MODULE_CORE Register Summary: [1]

Table 13-948. CTRL_CORE_SMA_SW_18

Address Offset	0x0000 1CB0		
Physical Address	0x4A00 3CB0	**Instance**	CTRL_MODULE_CORE
Description	OCP Spare Register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMA_SW_18																															

Bits	Field Name	Description	Type	Reset
31:0	SMA_SW_18	OCP Spare Register	R	0x0

Table 13-949. Register Call Summary for Register CTRL_CORE_SMA_SW_18

Control Module Register Manual
• CTRL_MODULE_CORE Register Summary: [0]

Table 13-950. CTRL_CORE_SMA_SW_19

Address Offset	0x0000 1CB4		
Physical Address	0x4A00 3CB4	**Instance**	CTRL_MODULE_CORE
Description	OCP Spare Register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMA_SW_19																															

Bits	Field Name	Description	Type	Reset
31:0	SMA_SW_19	OCP Spare Register	R	0x0

Table 13-951. Register Call Summary for Register CTRL_CORE_SMA_SW_19

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-952. CTRL_CORE_SMA_SW_20

Address Offset	0x0000 1CB8		
Physical Address	0x4A00 3CB8	**Instance**	CTRL_MODULE_CORE
Description	OCP Spare Register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMA_SW_20																															

Bits	Field Name	Description	Type	Reset
31:0	SMA_SW_20	OCP Spare Register	R	0x0

Table 13-953. Register Call Summary for Register CTRL_CORE_SMA_SW_20

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-954. CTRL_CORE_SMA_SW_21

Address Offset	0x0000 1CBC		
Physical Address	0x4A00 3CBC	**Instance**	CTRL_MODULE_CORE
Description	OCP Spare Register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMA_SW_21																															

Bits	Field Name	Description	Type	Reset
31:0	SMA_SW_21	OCP Spare Register	R	0x0

Table 13-955. Register Call Summary for Register CTRL_CORE_SMA_SW_21

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-956. CTRL_CORE_SMA_SW_22

Address Offset	0x0000 1CC0		
Physical Address	0x4A00 3CC0	**Instance**	CTRL_MODULE_CORE
Description			

Table 13-956. CTRL_CORE_SMA_SW_22 (continued)

Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMA_SW_22																CORE_DPLL_REGMF_CONTROL															

Bits	Field Name	Description	Type	Reset
31:18	SMA_SW_22	OCP Spare Register	RW	0x0
17:0	CORE_DPLL_REGMF_CONTROL	Fractional part of the software-configured multiplication ratio M of DPLL_CORE. To disable the fractional part this bit field must be set to 0x0.	RW	0x0

Table 13-957. Register Call Summary for Register CTRL_CORE_SMA_SW_22

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-958. CTRL_CORE_SMA_SW_23

Address Offset	0x0000 1CC4		
Physical Address	0x4A00 3CC4	**Instance**	CTRL_MODULE_CORE
Description	OCP Spare Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMA_SW_23																															

Bits	Field Name	Description	Type	Reset
31:0	SMA_SW_23	OCP Spare Register	RW	0x0

Table 13-959. Register Call Summary for Register CTRL_CORE_SMA_SW_23

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-960. CTRL_CORE_SMA_SW_24

Address Offset	0x0000 1CC8		
Physical Address	0x4A00 3CC8	**Instance**	CTRL_MODULE_CORE
Description	OCP Spare Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMA_SW_24																															

Bits	Field Name	Description	Type	Reset
31:0	SMA_SW_24	OCP Spare Register	RW	0x0

Table 13-961. Register Call Summary for Register CTRL_CORE_SMA_SW_24

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-962. CTRL_CORE_SMA_SW_25

Address Offset	0x0000 1CCC		
Physical Address	0x4A00 3CCC	**Instance**	CTRL_MODULE_CORE
Description	OCP Spare Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMA_SW_25																															

Bits	Field Name	Description	Type	Reset
31:0	SMA_SW_25	OCP Spare Register	RW	0x0

Table 13-963. Register Call Summary for Register CTRL_CORE_SMA_SW_25

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-964. CTRL_CORE_SMA_SW_26

Address Offset	0x0000 1CD0		
Physical Address	0x4A00 3CD0	**Instance**	CTRL_MODULE_CORE
Description	OCP Spare Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMA_SW_26																															

Bits	Field Name	Description	Type	Reset
31:0	SMA_SW_26	OCP Spare Register	RW	0x0

Table 13-965. Register Call Summary for Register CTRL_CORE_SMA_SW_26

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

Table 13-966. CTRL_CORE_SMA_SW_27

Address Offset	0x0000 1CD4		
Physical Address	0x4A00 3CD4	**Instance**	CTRL_MODULE_CORE
Description	OCP Spare Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SMA_SW_27																															

Bits	Field Name	Description	Type	Reset
31:0	SMA_SW_27	OCP Spare Register	RW	0x0

Table 13-967. Register Call Summary for Register CTRL_CORE_SMA_SW_27

Control Module Register Manual
- CTRL_MODULE_CORE Register Summary: [0]

13.5.3 CTRL_MODULE_WKUP Registers

13.5.3.1 CTRL_MODULE_WKUP Register Summary

Table 13-968. CTRL_MODULE_WKUP Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	CTRL_MODULE_WKUP Base Address
RESERVED_a (a = 0 to 63)	R	32	0x0000 0000 + (a*4)	0x4AE0 C000 + (a*4)
CTRL_WKUP_SEC_CTRL	RW	32	0x0000 0100	0x4AE0 C100
RESERVED	R	32	0x0000 0104	0x4AE0 C104
CTRL_WKUP_SEC_TAP	RW	32	0x0000 0108	0x4AE0 C108
CTRL_WKUP_OCPREG_SPARE	RW	32	0x0000 010C	0x4AE0 C10C
CTRL_WKUP_SECURE_EMIF1_SDRAM_CONFIG	RW	32	0x0000 0110	0x4AE0 C110
RESERVED_b (b = 0 to 9)	R	32	0x0000 0114 + (b*4)	0x4AE0 C114 + b*4)
CTRL_WKUP_STD_FUSE_CONF	R	32	0x0000 013C	0x4AE0 C13C
RESERVED	R	32	0x0000 0140	0x4AE0 C140
CTRL_WKUP_EMIF1_SDRAM_CONFIG_EXT	RW	32	0x0000 0144	0x4AE0 C144
RESERVED	R	32	0x0000 0148	0x4AE0 C148
CTRL_WKUP_EMIF1_SDRAM_CONFIG_EXT_1	R	32	0x0000 014C	0x4AE0 C14C
RESERVED	R	32	0x0000 0150	0x4AE0 C150
RESERVED	R	32	0x0000 0154	0x4AE0 C154
RESERVED	R	32	0x0000 0158	0x4AE0 C158
RESERVED	R	32	0x0000 015C	0x4AE0 C15C
RESERVED	R	32	0x0000 0160	0x4AE0 C160
CTRL_WKUP_LDOSRAM_CORE_VOLTAGE_CTRL	RW	32	0x0000 0164	0x4AE0 C164
RESERVED_c (c = 0 to 37)	R	32	0x0000 0168 + (c*4)	0x4AE0 C168 + (c*4)
CTRL_WKUP_STD_FUSE_DIE_ID_0	R	32	0x0000 0200	0x4AE0 C200
CTRL_WKUP_ID_CODE	R	32	0x0000 0204	0x4AE0 C204
CTRL_WKUP_STD_FUSE_DIE_ID_1	R	32	0x0000 0208	0x4AE0 C208
CTRL_WKUP_STD_FUSE_DIE_ID_2	R	32	0x0000 020C	0x4AE0 C20C
CTRL_WKUP_STD_FUSE_DIE_ID_3	R	32	0x0000 0210	0x4AE0 C210
CTRL_WKUP_STD_FUSE_PROD_ID_0	R	32	0x0000 0214	0x4AE0 C214
RESERVED_d (d = 0 to 292)	R	32	0x0000 0218 + (d*4)	0x4AE0 C218 + (d*4)
CTRL_WKUP_CONTROL_XTAL_OSCILLATOR	RW	32	0x0000 05AC	0x4AE0 C5AC
RESERVED	R	32	0x0000 05B0	0x4AE0 C5B0
RESERVED	R	32	0x0000 05B4	0x4AE0 C5B4
RESERVED	R	32	0x0000 05B8	0x4AE0 C5B8
RESERVED	R	32	0x0000 05BC	0x4AE0 C5BC
RESERVED	R	32	0x0000 05C0	0x4AE0 C5C0

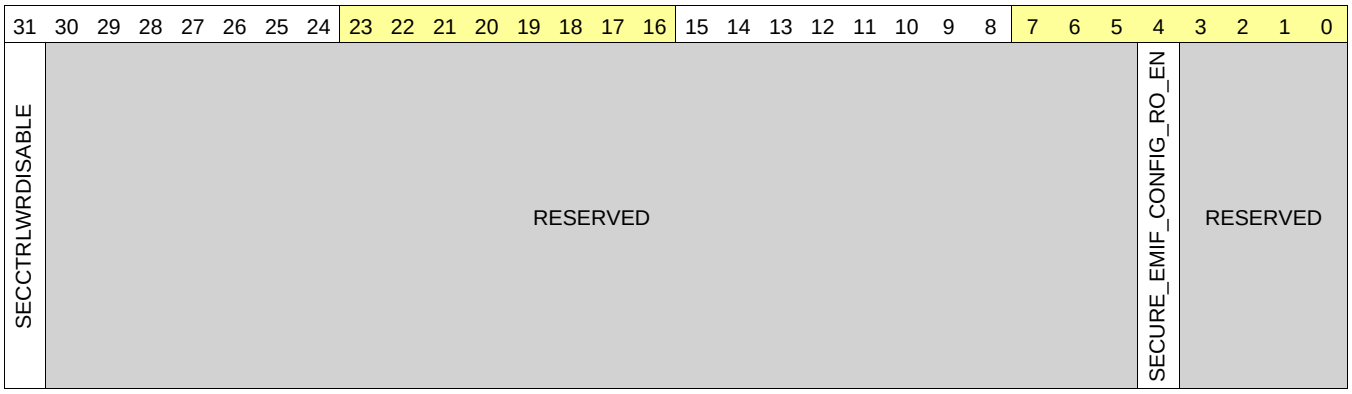
Table 13-968. CTRL_MODULE_WKUP Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	CTRL_MODULE_WKUP Base Address
RESERVED	R	32	0x0000 05C4	0x4AE0 C5C4
CTRL_WKUP_EFUSE_1	RW	32	0x0000 05C8	0x4AE0 C5C8
CTRL_WKUP_EFUSE_2	RW	32	0x0000 05CC	0x4AE0 C5CC
CTRL_WKUP_EFUSE_3	RW	32	0x0000 05D0	0x4AE0 C5D0
CTRL_WKUP_EFUSE_4	RW	32	0x0000 05D4	0x4AE0 C5D4
RESERVED_e (e = 0 to 7)	R	32	0x0000 05D8 + (e*4)	0x4AE0 C5D8 + (e*4)
CTRL_WKUP_EFUSE_13	RW	32	0x0000 05F8	0x4AE0 C5F8
RESERVED_f (f = 0 to 64)	R	32	0x0000 05FC + (f*4)	0x4AE0 C5FC + (f*4)
CTRL_WKUP_CONF_DEBUG_SEL_TST_0	RW	32	0x0000 0700	0x4AE0 C700
CTRL_WKUP_CONF_DEBUG_SEL_TST_1	RW	32	0x0000 0704	0x4AE0 C704
CTRL_WKUP_CONF_DEBUG_SEL_TST_2	RW	32	0x0000 0708	0x4AE0 C708
CTRL_WKUP_CONF_DEBUG_SEL_TST_3	RW	32	0x0000 070C	0x4AE0 C70C
CTRL_WKUP_CONF_DEBUG_SEL_TST_4	RW	32	0x0000 0710	0x4AE0 C710
CTRL_WKUP_CONF_DEBUG_SEL_TST_5	RW	32	0x0000 0714	0x4AE0 C714
CTRL_WKUP_CONF_DEBUG_SEL_TST_6	RW	32	0x0000 0718	0x4AE0 C718
CTRL_WKUP_CONF_DEBUG_SEL_TST_7	RW	32	0x0000 071C	0x4AE0 C71C
CTRL_WKUP_CONF_DEBUG_SEL_TST_8	RW	32	0x0000 0720	0x4AE0 C720
CTRL_WKUP_CONF_DEBUG_SEL_TST_9	RW	32	0x0000 0724	0x4AE0 C724
CTRL_WKUP_CONF_DEBUG_SEL_TST_10	RW	32	0x0000 0728	0x4AE0 C728
CTRL_WKUP_CONF_DEBUG_SEL_TST_11	RW	32	0x0000 072C	0x4AE0 C72C
CTRL_WKUP_CONF_DEBUG_SEL_TST_12	RW	32	0x0000 0730	0x4AE0 C730
CTRL_WKUP_CONF_DEBUG_SEL_TST_13	RW	32	0x0000 0734	0x4AE0 C734
CTRL_WKUP_CONF_DEBUG_SEL_TST_14	RW	32	0x0000 0738	0x4AE0 C738
CTRL_WKUP_CONF_DEBUG_SEL_TST_15	RW	32	0x0000 073C	0x4AE0 C73C
CTRL_WKUP_CONF_DEBUG_SEL_TST_16	RW	32	0x0000 0740	0x4AE0 C740
CTRL_WKUP_CONF_DEBUG_SEL_TST_17	RW	32	0x0000 0744	0x4AE0 C744
CTRL_WKUP_CONF_DEBUG_SEL_TST_18	RW	32	0x0000 0748	0x4AE0 C748
CTRL_WKUP_CONF_DEBUG_SEL_TST_19	RW	32	0x0000 074C	0x4AE0 C74C
CTRL_WKUP_CONF_DEBUG_SEL_TST_20	RW	32	0x0000 0750	0x4AE0 C750
CTRL_WKUP_CONF_DEBUG_SEL_TST_21	RW	32	0x0000 0754	0x4AE0 C754
CTRL_WKUP_CONF_DEBUG_SEL_TST_22	RW	32	0x0000 0758	0x4AE0 C758
CTRL_WKUP_CONF_DEBUG_SEL_TST_23	RW	32	0x0000 075C	0x4AE0 C75C
CTRL_WKUP_CONF_DEBUG_SEL_TST_24	RW	32	0x0000 0760	0x4AE0 C760
CTRL_WKUP_CONF_DEBUG_SEL_TST_25	RW	32	0x0000 0764	0x4AE0 C764
CTRL_WKUP_CONF_DEBUG_SEL_TST_26	RW	32	0x0000 0768	0x4AE0 C768
CTRL_WKUP_CONF_DEBUG_SEL_TST_27	RW	32	0x0000 076C	0x4AE0 C76C
CTRL_WKUP_CONF_DEBUG_SEL_TST_28	RW	32	0x0000 0770	0x4AE0 C770
CTRL_WKUP_CONF_DEBUG_SEL_TST_29	RW	32	0x0000 0774	0x4AE0 C774
CTRL_WKUP_CONF_DEBUG_SEL_TST_30	RW	32	0x0000 0778	0x4AE0 C778
CTRL_WKUP_CONF_DEBUG_SEL_TST_31	RW	32	0x0000 077C	0x4AE0 C77C

13.5.3.2 CTRL_MODULE_WKUP Register Description

Table 13-969. CTRL_WKUP_SEC_CTRL

Address Offset	0x0000 0100		
Physical Address	0x4AE0 C100	**Instance**	CTRL_MODULE_WKUP
Description	Control Register		
Type	RW		



Bits	Field Name	Description	Type	Reset
31	SECCTRLWRDISABLE	Control Register write disable control. 0x0 = Write in this register is allowed 0x1 = Write in this register is forbidden	RW	0x0
30:5	RESERVED		R	0x0
4	SECURE_EMIF_CONFIG_RO_EN	Access mode for registers: CTRL_WKUP_EMIF1_SDRAM_CONFIG CTRL_WKUP_EMIF2_SDRAM_CONFIG 0x0 = These registers are RW 0x1 = These registers are RO	RW	0x0
3:0	RESERVED		R	0x0

Table 13-970. Register Call Summary for Register CTRL_WKUP_SEC_CTRL

Control Module Register Manual
- CTRL_MODULE_WKUP Register Summary: [0]
- CTRL_MODULE_WKUP Register Description: [1]

Table 13-971. CTRL_WKUP_SEC_TAP

Address Offset	0x0000 0108		
Physical Address	0x4AE0 C108	**Instance**	CTRL_MODULE_WKUP
Description	TAP controllers register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SECTAPWR_DISABLE	RESERVED				RESERVED	RESERVED													DSP2_TAPENABLE	JTAGEXT_TAPENABLE	RESERVED						IEEE1500_ENABLE	P1500_ENABLE	IPU1_TAPENABLE	DSP1_TAPENABLE	DAP_TAPENABLE

Bits	Field Name	Description	Type	Reset
31	SECTAPWR_DISABLE	TAP controllers register write disable control	RW	0x0
		0x0: Write in this register is allowed	Woco	
		0x1: Write in this register is forbidden		
30:27	RESERVED		R	0x0
26	RESERVED	Reserved. This bit must not be modified.	RW	0x1
25:13	RESERVED		R	0x0
12	DSP2_TAPENABLE	DSP2 TAP control	RW	0x1
		0x0: DSP2 TAP controller is disabled		
		0x1: DSP2 TAP controller is enabled		
11	JTAGEXT_TAPENABLE	External JTAG expansion TAP control.	RW	0x1
		0x0: external JTAG TAP controller is disabled		
		0x1: external JTAG TAP controller is enabled		
10:5	RESERVED		R	0x0
4	IEEE1500_ENABLE	IEEE1500 and P1500 access enable	RW	0x1
		0x0: P1500 controller is disabled	W1toClr	
		0x1: P1500 controller is enabled		
3	P1500_ENABLE	P1500 access enable	RW	0x1
		0x0: P1500 controller is disabled		
		0x1: P1500 controller is enabled		
2	IPU1_TAPENABLE	IPU1 TAP control	RW	0x1
		0x0: IPU1 TAP controller is disabled		
		0x1: IPU1 TAP controller is enabled		
1	DSP1_TAPENABLE	DSP1 TAP control	RW	0x1
		0x0: DSP1 TAP controller is disabled		
		0x1: DSP1 TAP controller is enabled		
0	DAP_TAPENABLE	DAP TAP control	RW	0x1
		0x0: DAP TAP controller is disabled		
		0x1: DAP TAP controller is enabled		

Table 13-972. Register Call Summary for Register CTRL_WKUP_SEC_TAP

Control Module Register Manual

- CTRL_MODULE_WKUP Register Summary: [0]

Table 13-973. CTRL_WKUP_OCPREG_SPARE

Address Offset	0x0000 010C		
Physical Address	0x4AE0 C10C	**Instance**	CTRL_MODULE_WKUP
Description	OCP Spare Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OCPREG_SPARE31	OCPREG_SPARE30	OCPREG_SPARE29	OCPREG_SPARE28	OCPREG_SPARE27	OCPREG_SPARE26	OCPREG_SPARE25	OCPREG_SPARE24	OCPREG_SPARE23	OCPREG_SPARE22	OCPREG_SPARE21	OCPREG_SPARE20	OCPREG_SPARE19	OCPREG_SPARE18	OCPREG_SPARE17	OCPREG_SPARE16	OCPREG_SPARE15	OCPREG_SPARE14	OCPREG_SPARE13	OCPREG_SPARE12	OCPREG_SPARE11	OCPREG_SPARE10	OCPREG_SPARE9	OCPREG_SPARE8	OCPREG_SPARE7	OCPREG_SPARE6	OCPREG_SPARE5	OCPREG_SPARE4	OCPREG_SPARE3	OCPREG_SPARE2	OCPREG_SPARE1	RESERVED

Bits	Field Name	Description	Type	Reset
31	OCPREG_SPARE31	OCP spare register 31	RW	0x0
30	OCPREG_SPARE30	OCP spare register 30	RW	0x0
29	OCPREG_SPARE29	OCP spare register 29	RW	0x0
28	OCPREG_SPARE28	OCP spare register 28	RW	0x0
27	OCPREG_SPARE27	OCP spare register 27	RW	0x0
26	OCPREG_SPARE26	OCP spare register 26	RW	0x0
25	OCPREG_SPARE25	OCP spare register 25	RW	0x0
24	OCPREG_SPARE24	OCP spare register 24	RW	0x0
23	OCPREG_SPARE23	OCP spare register 23	RW	0x0
22	OCPREG_SPARE22	OCP spare register 22	RW	0x0
21	OCPREG_SPARE21	OCP spare register 21	RW	0x0
20	OCPREG_SPARE20	OCP spare register 20	RW	0x0
19	OCPREG_SPARE19	OCP spare register 19	RW	0x0
18	OCPREG_SPARE18	OCP spare register 18	RW	0x0
17	OCPREG_SPARE17	OCP spare register 17	RW	0x0
16	OCPREG_SPARE16	OCP spare register 16	RW	0x0
15	OCPREG_SPARE15	OCP spare register 15	RW	0x0
14	OCPREG_SPARE14	OCP spare register 14	RW	0x0
13	OCPREG_SPARE13	OCP spare register 13	RW	0x0
12	OCPREG_SPARE12	OCP spare register 12	RW	0x0
11	OCPREG_SPARE11	OCP spare register 11	RW	0x0
10	OCPREG_SPARE10	OCP spare register 10	RW	0x0
9	OCPREG_SPARE9	OCP spare register 9	RW	0x0
8	OCPREG_SPARE8	OCP spare register 8	RW	0x0
7	OCPREG_SPARE7	OCP spare register 7	RW	0x0
6	OCPREG_SPARE6	OCP spare register 6	RW	0x0
5	OCPREG_SPARE5	OCP spare register 5	RW	0x0
4	OCPREG_SPARE4	OCP spare register 4	RW	0x0
3	OCPREG_SPARE3	OCP spare register 3	RW	0x0

Bits	Field Name	Description	Type	Reset
2	OCPREG_SPARE2	OCP spare register 2	RW	0x0
1	OCPREG_SPARE1	OCP spare register 1	RW	0x0
0	RESERVED		R	0x0

Table 13-974. Register Call Summary for Register CTRL_WKUP_OCPREG_SPARE

Control Module Register Manual
- CTRL_MODULE_WKUP Register Summary: [0]

Table 13-975. CTRL_WKUP_SECURE_EMIF1_SDRAM_CONFIG

Address Offset	0x0000 0110
Physical Address	0x4AE0 C110 **Instance** CTRL_MODULE_WKUP
Description	EMIF1 SDRAM configuration register. Its values are exported to EMIF_SDRAM_CONFIG register at POR. For bit field descriptions see EMIF_SDRAM_CONFIG register in Section 10.2, *EMIF Controller*, in Chapter 10, *Memory Subsystem*. Write to this register is allowed if the CTRL_WKUP_SEC_CTRL[4] SECURE_EMIF_CONFIG_RO_EN bit is set to 0x0 (default).
Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED			EMIF1_SDRAM_IBANK_POS		EMIF1_SDRAM_DDR_TERM			EMIF1_SDRAM_DDR2_DDQS		EMIF1_SDRAM_DYN_ODT		EMIF1_SDRAM_DDR_DISABLE_DLL		EMIF1_SDRAM_DRIVE		EMIF1_SDRAM_CWL		RESERVED		EMIF1_SDRAM_CL		EMIF1_SDRAM_ROWSIZE		EMIF1_SDRAM_IBANK			RESERVED		EMIF1_SDRAM_PAGESIZE		

Bits	Field Name	Description	Type	Reset
31:29	RESERVED		R	0x0
28:27	EMIF1_SDRAM_IBANK_POS	Internal bank position.	RW	0x0
26:24	EMIF1_SDRAM_DDR_TERM	DDR2 and DDR3 termination resistor value.	RW	0x0
23	EMIF1_SDRAM_DDR2_DDQS	DDR2 differential DQS enable.	RW	0x1
22:21	EMIF1_SDRAM_DYN_ODT	DDR3 Dynamic ODT.	RW	0x0
20	EMIF1_SDRAM_DDR_DISABLE_DLL	Disable DLL select.	RW	0x0
19:18	EMIF1_SDRAM_DRIVE	SDRAM drive strength.	RW	0x0
17:16	EMIF1_SDRAM_CWL	DDR3 CAS Write latency.	RW	0x0
15:14	RESERVED		R	0x0
13:10	EMIF1_SDRAM_CL	CAS Latency.	RW	0x0
9:7	EMIF1_SDRAM_ROWSIZE	Row Size.	RW	0x0
6:4	EMIF1_SDRAM_IBANK	Internal Bank setup.	RW	0x0
3	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
2:0	EMIF1_SDRAM_PAGESIZE	Page Size.	RW	0x0

Table 13-976. Register Call Summary for Register CTRL_WKUP_SECURE_EMIF1_SDRAM_CONFIG

Control Module Functional Description
- Registers For Basic EMIF configuration: [0][1]

Control Module Register Manual
- CTRL_MODULE_WKUP Register Summary: [2]

Table 13-977. CTRL_WKUP_STD_FUSE_CONF

Address Offset	0x0000 013C		
Physical Address	0x4AE0 C13C	**Instance**	CTRL_MODULE_WKUP
Description	Standard Fuse conf [63:32]. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												STD_FUSE_EMIF1_INITREF_DEF_DIS	STD_FUSE_EMIF1_DDR3_LPDDR2N	RESERVED						STD_FUSE_CH_SPEEDUP_DISABLE	RESERVED												

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x-
19	STD_FUSE_EMIF1_INITREF_DEF_DIS	Disable EMIF1 DDR refresh and initialization sequence 0x0 = refresh and initialization sequence is enabled 0x1 = refresh and initialization sequence is disabled	R	0x-
18	STD_FUSE_EMIF1_DDR3_LPDDR2N	EMIF1 DDR3 0x0 = LPDDR2 configured 0x1 = DDR3 configured **NOTE:** LPDDR2 is not supported on the DRA78x family of devices.	R	0x-
17:13	RESERVED		R	0x-
12	STD_FUSE_CH_SPEEDUP_DISABLE	ROM code settings for configuration header block and speedup block. Only SW access (no hardware access). 0x0 = enables CH and speedup 0x1 = disables CH and speedup	R	0x-
11:0	RESERVED		R	0x-

Table 13-978. Register Call Summary for Register CTRL_WKUP_STD_FUSE_CONF

Control Module Register Manual
- CTRL_MODULE_WKUP Register Summary: [0]

Table 13-979. CTRL_WKUP_EMIF1_SDRAM_CONFIG_EXT

Address Offset	0x0000 0144		
Physical Address	0x4AE0 C144	**Instance**	CTRL_MODULE_WKUP
Description	SLICE register for emif1 and emif2		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														EMIF1_NARROW_ONLY	EMIF1_EN_ECC	EMIF1_REG_PHY_NUM_OF_SAMPLES		EMIF1_REG_PHY_SEL_LOGIC	EMIF1_REG_PHY_ALL_DQ_MPR_RD_RESP	EMIF1_REG_PHY_OUTPUT_STATUS_SELECT			RESERVED	EMIF1_SDRAM_DISABLE_RESET		EMIF1_PHY_RD_LOCAL_ODT		RESERVED		EMIF1_DFI_CLOCK_PHASE_CTRL	EMIF1_EN_SLICE_2	EMIF1_EN_SLICE_1	EMIF1_EN_SLICE_0

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		R	0x0
17	EMIF1_NARROW_ONLY	EMIF1 operates in narrow mode, to allow for data macros to be powered down to save power 0x0 = narrow mode disabled 0x1 = narrow mode enabled	RW	0x0
16	EMIF1_EN_ECC	EMIF1 ECC enable 0x0 = ECC is disabled 0x1 = ECC is enabled	RW	0x0
15:14	EMIF1_REG_PHY_NUM_OF_SAMPLES	Controls the number of DQ samples required for read leveling. The recommended setting for full leveling is 0x3 (128 samples) and for incremental leveling is 0x0 (4 samples). 0x0 = 4 samples 0x1 = 8 samples. 0x2 = 16 samples 0x3 = 128 samples	RW	0x0
13	EMIF1_REG_PHY_SEL_LOGIC	Selects an algorithm for read leveling. The use of algorithm 1 (set by default) is recommended. 0x0 = Algorithm 1 is used 0x1 = Algorithm 2 is used	RW	0x0
12	EMIF1_REG_PHY_ALL_DQ_MPR_RD_RESP	Analysis method of DQ bits during read leveling. 0x0: if the DRAM provides a read response on only one DQ bit (this can be any bit, since in this mode all 8 DQ bits are OR-ed together). This is the default setting and works with all memory types (memories send responses on all DQ bits or on a single DQ bit). 0x1: if the DRAM provides a read response on all DQ bits.	RW	0x0
11:9	EMIF1_REG_PHY_OUTPUT_STATUS_SELECT	Selects the status to be observed on the outputs of the DDR PHYs through CTRL_WKUP_EMIF1_SDRAM_CONFIG_EXT_1 register. 0x0 = selects phy_reg_rdlvl_start_ratio[7:0] 0x1 = selects phy_reg_rdlvl_start_ratio[15:8] 0x2 = selects phy_reg_rdlvl_end_ratio[7:0] 0x3 = selects phy_reg_rdlvl_end_ratio[15:8]	RW	0x0

Bits	Field Name	Description	Type	Reset
8	RESERVED		R	0x1
7	EMIF1_SDRAM_DISABLE_RESET	DDR3 SDRAM reset disable. 0x0 = DDR3 SDRAM reset signal is enabled. It can be asserted by EMIF 0x1 = DDR3 SDRAM reset signal is disabled. It is forbidden to EMIF to assert it.	RW	0x0
6:5	EMIF1_PHY_RD_LOCAL_ODT	Control of ODT (on – die termination) settings for the device DDR I/Os. ODT is enabled only during read operations when termination is required. 0x0 = ODT disabled 0x1= 60 Ohms 0x2 = 80 Ohms 0x3 =120 Ohms	RW	0x0
4	RESERVED		RW	0x0
3	EMIF1_DFI_CLOCK_PHASE_CTRL	EMIF_FICLK clock phase control (shifting by 180°). For normal operation this bit must always be set to 0x0 (disabled).	RW	0x0
2	EMIF1_EN_SLICE_2	Enable command PHY 2. When using DDR3 this bit can be set to 0x0 or 0x1. For lower power consumption 0x0 is used.	RW	0x1
1	EMIF1_EN_SLICE_1	Enable command PHY 1. 0x1 is the mandatory setting if DDR3 is used. EMIF1_EN_SLICE_0 and EMIF1_EN_SLICE_1 have to be programmed with the same value.	RW	0x1
0	EMIF1_EN_SLICE_0	Enable command PHY 0. 0x1 is the mandatory setting if DDR3 is used. EMIF1_EN_SLICE_0 and EMIF1_EN_SLICE_1 have to be programmed with the same value.	RW	0x1

Table 13-980. Register Call Summary for Register CTRL_WKUP_EMIF1_SDRAM_CONFIG_EXT

Control Module Functional Description
- Registers For Basic EMIF configuration: [0]

Control Module Register Manual
- CTRL_MODULE_WKUP Register Summary: [1]

Table 13-981. CTRL_WKUP_EMIF1_SDRAM_CONFIG_EXT_1

Address Offset	0x0000 014C		
Physical Address	0x4AE0 C14C	**Instance**	CTRL_MODULE_WKUP
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EMIF1_PHY_REG_READ_DATA_EYE_LVL																															

Bits	Field Name	Description	Type	Reset
31:0	EMIF1_PHY_REG_READ_DATA_EYE_LVL		R	0x0

Table 13-982. Register Call Summary for Register CTRL_WKUP_EMIF1_SDRAM_CONFIG_EXT_1

Control Module Register Manual
- CTRL_MODULE_WKUP Register Summary: [0]
- CTRL_MODULE_WKUP Register Description: [1]

Table 13-983. CTRL_WKUP_LDOSRAM_CORE_VOLTAGE_CTRL

Address Offset	0x0000 0164		
Physical Address	0x4AE0 C164	**Instance**	CTRL_MODULE_WKUP
Description	Core SRAM LDO Control register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED					LDOSRAMCORE_RETMODE_MUX_CTRL			LDOSRAMCORE_RETMODE_VSET_IN					LDOSRAMCORE_RETMODE_VSET_OUT			RESERVED						LDOSRAMCORE_ACTMODE_MUX_CTRL			LDOSRAMCORE_ACTMODE_VSET_IN					LDOSRAMCORE_ACTMODE_VSET_OUT			

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x0
26	LDOSRAMCORE_RETMODE_MUX_CTRL	Override control of EFUSE Retention Mode Voltage value 0x0: eFuse value is used 0x1: Override value is used	RW	0x0
25:21	LDOSRAMCORE_RETMODE_VSET_IN	EFUSE Retention Mode Voltage value (vset[9:5])	R	0x0
20:16	LDOSRAMCORE_RETMODE_VSET_OUT	Override value for Retention Mode Voltage	RW	0x0
15:11	RESERVED		R	0x0
10	LDOSRAMCORE_ACTMODE_MUX_CTRL	Override control of EFUSE Active Mode Voltage value 0x0: eFuse value is used 0x1: Override value is used	RW	0x0
9:5	LDOSRAMCORE_ACTMODE_VSET_IN	EFUSE Active Mode Voltage value (vset[4:0])	R	0x0
4:0	LDOSRAMCORE_ACTMODE_VSET_OUT	Override value for Active Mode Voltage value	RW	0x0

Table 13-984. Register Call Summary for Register CTRL_WKUP_LDOSRAM_CORE_VOLTAGE_CTRL

Control Module Register Manual

- CTRL_MODULE_WKUP Register Summary: [0]

Table 13-985. CTRL_WKUP_STD_FUSE_DIE_ID_0

Address Offset	0x0000 0200		
Physical Address	0x4AE0 C200	**Instance**	CTRL_MODULE_WKUP
Description	Die ID Register : Part 0. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STD_FUSE_DIE_ID_0																															

Bits	Field Name	Description	Type	Reset
31:0	STD_FUSE_DIE_ID_0		R	0x0

Table 13-986. Register Call Summary for Register CTRL_WKUP_STD_FUSE_DIE_ID_0

Control Module Register Manual
- CTRL_MODULE_WKUP Register Summary: [0]

Table 13-987. CTRL_WKUP_ID_CODE

Address Offset	0x0000 0204		
Physical Address	0x4AE0 C204	**Instance**	CTRL_MODULE_WKUP
Description	ID_CODE Key Register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STD_FUSE_IDCODE																															

Bits	Field Name	Description	Type	Reset
31:0	STD_FUSE_IDCODE		R	0x0

Table 13-988. Register Call Summary for Register CTRL_WKUP_ID_CODE

Control Module Register Manual
- CTRL_MODULE_WKUP Register Summary: [0]

Table 13-989. CTRL_WKUP_STD_FUSE_DIE_ID_1

Address Offset	0x0000 0208		
Physical Address	0x4AE0 C208	**Instance**	CTRL_MODULE_WKUP
Description	Die ID Register : Part 1. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STD_FUSE_DIE_ID_1																															

Bits	Field Name	Description	Type	Reset
31:0	STD_FUSE_DIE_ID_1		R	0x0

Table 13-990. Register Call Summary for Register CTRL_WKUP_STD_FUSE_DIE_ID_1

Control Module Register Manual

- CTRL_MODULE_WKUP Register Summary: [0]

Table 13-991. CTRL_WKUP_STD_FUSE_DIE_ID_2

Address Offset	0x0000 020C
Physical Address	0x4AE0 C20C **Instance** CTRL_MODULE_WKUP
Description	Die ID Register : Part 2. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.
Type	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STD_FUSE_DIE_ID_2																															

Bits	Field Name	Description	Type	Reset
31:0	STD_FUSE_DIE_ID_2		R	0x0

Table 13-992. Register Call Summary for Register CTRL_WKUP_STD_FUSE_DIE_ID_2

Control Module Register Manual

- CTRL_MODULE_WKUP Register Summary: [0]

Table 13-993. CTRL_WKUP_STD_FUSE_DIE_ID_3

Address Offset	0x0000 0210
Physical Address	0x4AE0 C210 **Instance** CTRL_MODULE_WKUP
Description	Die ID Register : Part 3. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.
Type	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STD_FUSE_DIE_ID_3																															

Bits	Field Name	Description	Type	Reset
31:0	STD_FUSE_DIE_ID_3		R	0x0

Table 13-994. Register Call Summary for Register CTRL_WKUP_STD_FUSE_DIE_ID_3

Control Module Register Manual

- CTRL_MODULE_WKUP Register Summary: [0]

Table 13-995. CTRL_WKUP_STD_FUSE_PROD_ID_0

Address Offset	0x0000 0214		
Physical Address	0x4AE0 C214	**Instance**	CTRL_MODULE_WKUP
Description	Prod ID Register : Part 0. Register shows part of the chip eFuse configuration on the OCP interface. Reading at the address of one of these registers provides a direct view into a part of the eFuse chain.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
STD_FUSE_PROD_ID																															

Bits	Field Name	Description	Type	Reset
31:0	STD_FUSE_PROD_ID		R	0x0

Table 13-996. Register Call Summary for Register CTRL_WKUP_STD_FUSE_PROD_ID_0

Control Module Register Manual

- CTRL_MODULE_WKUP Register Summary: [0]

Table 13-997. CTRL_WKUP_CONTROL_XTAL_OSCILLATOR

Address Offset	0x0000 05AC		
Physical Address	0x4AE0 C5AC	**Instance**	CTRL_MODULE_WKUP
Description	XTAL OSCILLATOR control		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OSCILLATOR0_BOOST	OSCILLATOR0_OS_OUT	OSCILLATOR1_BOOST	OSCILLATOR1_OS_OUT	RESERVED																											

Bits	Field Name	Description	Type	Reset
31	OSCILLATOR0_BOOST	Fast startup control of OSC0 0x0 = Fast startup is disabled 0x1 = Fast startup is enabled	RW	0x1
30	OSCILLATOR0_OS_OUT	Oscillator output of OSC0 0x0 = low to high transition in BOOST mode 0x1 = BOOST is disabled	R	0x0
29	OSCILLATOR1_BOOST	Fast startup control of OSC1 0x0 = Fast startup is disabled 0x1 = Fast startup is enabled	RW	0x1
28	OSCILLATOR1_OS_OUT	Oscillator output of OSC1 0x0 = low to high transition in BOOST mode 0x1 = BOOST is disabled	R	0x0
27:0	RESERVED		R	0x0

Table 13-998. Register Call Summary for Register CTRL_WKUP_CONTROL_XTAL_OSCILLATOR

Control Module Register Manual

- CTRL_MODULE_WKUP Register Summary: [0]

Table 13-999. CTRL_WKUP_EFUSE_1

Address Offset	0x0000 05C8		
Physical Address	0x4AE0 C5C8	**Instance**	CTRL_MODULE_WKUP
Description	EFUSE compensation 1		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DDRDIFF_PTV_NORTH_SIDE_N5	DDRDIFF_PTV_NORTH_SIDE_N4	DDRDIFF_PTV_NORTH_SIDE_N3	DDRDIFF_PTV_NORTH_SIDE_N2	DDRDIFF_PTV_NORTH_SIDE_N1	DDRDIFF_PTV_NORTH_SIDE_N0	DDRDIFF_PTV_NORTH_SIDE_P5	DDRDIFF_PTV_NORTH_SIDE_P4	DDRDIFF_PTV_NORTH_SIDE_P3	DDRDIFF_PTV_NORTH_SIDE_P2	DDRDIFF_PTV_NORTH_SIDE_P1	DDRDIFF_PTV_NORTH_SIDE_P0	DDRDIFF_PTV_EAST_SIDE_N5	DDRDIFF_PTV_EAST_SIDE_N4	DDRDIFF_PTV_EAST_SIDE_N3	DDRDIFF_PTV_EAST_SIDE_N2	DDRDIFF_PTV_EAST_SIDE_N1	DDRDIFF_PTV_EAST_SIDE_N0	DDRDIFF_PTV_EAST_SIDE_P5	DDRDIFF_PTV_EAST_SIDE_P4	DDRDIFF_PTV_EAST_SIDE_P3	DDRDIFF_PTV_EAST_SIDE_P2	DDRDIFF_PTV_EAST_SIDE_P1	DDRDIFF_PTV_EAST_SIDE_P0	RESERVED							

Bits	Field Name	Description	Type	Reset
31	DDRDIFF_PTV_NORTH_SIDE_N5		RW	0x0
30	DDRDIFF_PTV_NORTH_SIDE_N4		RW	0x0
29	DDRDIFF_PTV_NORTH_SIDE_N3		RW	0x0
28	DDRDIFF_PTV_NORTH_SIDE_N2		RW	0x0
27	DDRDIFF_PTV_NORTH_SIDE_N1		RW	0x0
26	DDRDIFF_PTV_NORTH_SIDE_N0		RW	0x0
25	DDRDIFF_PTV_NORTH_SIDE_P5		RW	0x0
24	DDRDIFF_PTV_NORTH_SIDE_P4		RW	0x0
23	DDRDIFF_PTV_NORTH_SIDE_P3		RW	0x0
22	DDRDIFF_PTV_NORTH_SIDE_P2		RW	0x0
21	DDRDIFF_PTV_NORTH_SIDE_P1		RW	0x0
20	DDRDIFF_PTV_NORTH_SIDE_P0		RW	0x0
19	DDRDIFF_PTV_EAST_SIDE_N5		RW	0x0
18	DDRDIFF_PTV_EAST_SIDE_N4		RW	0x0
17	DDRDIFF_PTV_EAST_SIDE_N3		RW	0x0

Bits	Field Name	Description	Type	Reset
16	DDRDIFF_PTV_EAST_SIDE_N2		RW	0x0
15	DDRDIFF_PTV_EAST_SIDE_N1		RW	0x0
14	DDRDIFF_PTV_EAST_SIDE_N0		RW	0x0
13	DDRDIFF_PTV_EAST_SIDE_P5		RW	0x0
12	DDRDIFF_PTV_EAST_SIDE_P4		RW	0x0
11	DDRDIFF_PTV_EAST_SIDE_P3		RW	0x0
10	DDRDIFF_PTV_EAST_SIDE_P2		RW	0x0
9	DDRDIFF_PTV_EAST_SIDE_P1		RW	0x0
8	DDRDIFF_PTV_EAST_SIDE_P0		RW	0x0
7:0	RESERVED		R	0x0

Table 13-1000. Register Call Summary for Register CTRL_WKUP_EFUSE_1

Control Module Register Manual

- CTRL_MODULE_WKUP Register Summary: [0]

Table 13-1001. CTRL_WKUP_EFUSE_2

Address Offset	0x0000 05CC		
Physical Address	0x4AE0 C5CC	**Instance**	CTRL_MODULE_WKUP
Description	EFUSE compensation 2		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DDRDIFF_PTV_SOUTH_SIDE_N5	DDRDIFF_PTV_SOUTH_SIDE_N4	DDRDIFF_PTV_SOUTH_SIDE_N3	DDRDIFF_PTV_SOUTH_SIDE_N2	DDRDIFF_PTV_SOUTH_SIDE_N1	DDRDIFF_PTV_SOUTH_SIDE_N0	DDRDIFF_PTV_SOUTH_SIDE_P5	DDRDIFF_PTV_SOUTH_SIDE_P4	DDRDIFF_PTV_SOUTH_SIDE_P3	DDRDIFF_PTV_SOUTH_SIDE_P2	DDRDIFF_PTV_SOUTH_SIDE_P1	DDRDIFF_PTV_SOUTH_SIDE_P0	DDRDIFF_PTV_WEST_SIDE_N5	DDRDIFF_PTV_WEST_SIDE_N4	DDRDIFF_PTV_WEST_SIDE_N3	DDRDIFF_PTV_WEST_SIDE_N2	DDRDIFF_PTV_WEST_SIDE_N1	DDRDIFF_PTV_WEST_SIDE_N0	DDRDIFF_PTV_WEST_SIDE_P5	DDRDIFF_PTV_WEST_SIDE_P4	DDRDIFF_PTV_WEST_SIDE_P3	DDRDIFF_PTV_WEST_SIDE_P2	DDRDIFF_PTV_WEST_SIDE_P1	DDRDIFF_PTV_WEST_SIDE_P0	RESERVED							

Bits	Field Name	Description	Type	Reset
31	DDRDIFF_PTV_SOUTH_SIDE_N5		RW	0x0
30	DDRDIFF_PTV_SOUTH_SIDE_N4		RW	0x0
29	DDRDIFF_PTV_SOUTH_SIDE_N3		RW	0x0
28	DDRDIFF_PTV_SOUTH_SIDE_N2		RW	0x0
27	DDRDIFF_PTV_SOUTH_SIDE_N1		RW	0x0

Bits	Field Name	Description	Type	Reset
26	DDRDIFF_PTV_SOUTH_SIDE_N0		RW	0x0
25	DDRDIFF_PTV_SOUTH_SIDE_P5		RW	0x0
24	DDRDIFF_PTV_SOUTH_SIDE_P4		RW	0x0
23	DDRDIFF_PTV_SOUTH_SIDE_P3		RW	0x0
22	DDRDIFF_PTV_SOUTH_SIDE_P2		RW	0x0
21	DDRDIFF_PTV_SOUTH_SIDE_P1		RW	0x0
20	DDRDIFF_PTV_SOUTH_SIDE_P0		RW	0x0
19	DDRDIFF_PTV_WEST_SIDE_N5		RW	0x0
18	DDRDIFF_PTV_WEST_SIDE_N4		RW	0x0
17	DDRDIFF_PTV_WEST_SIDE_N3		RW	0x0
16	DDRDIFF_PTV_WEST_SIDE_N2		RW	0x0
15	DDRDIFF_PTV_WEST_SIDE_N1		RW	0x0
14	DDRDIFF_PTV_WEST_SIDE_N0		RW	0x0
13	DDRDIFF_PTV_WEST_SIDE_P5		RW	0x0
12	DDRDIFF_PTV_WEST_SIDE_P4		RW	0x0
11	DDRDIFF_PTV_WEST_SIDE_P3		RW	0x0
10	DDRDIFF_PTV_WEST_SIDE_P2		RW	0x0
9	DDRDIFF_PTV_WEST_SIDE_P1		RW	0x0
8	DDRDIFF_PTV_WEST_SIDE_P0		RW	0x0
7:0	RESERVED		R	0x0

Table 13-1002. Register Call Summary for Register CTRL_WKUP_EFUSE_2

Control Module Register Manual

- CTRL_MODULE_WKUP Register Summary: [0]

Table 13-1003. CTRL_WKUP_EFUSE_3

Address Offset	0x0000 05D0		
Physical Address	0x4AE0 C5D0	**Instance**	CTRL_MODULE_WKUP
Description	EFUSE compensation 3		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DDRSE_PTV_NORTH_SIDE_N5	DDRSE_PTV_NORTH_SIDE_N4	DDRSE_PTV_NORTH_SIDE_N3	DDRSE_PTV_NORTH_SIDE_N2	DDRSE_PTV_NORTH_SIDE_N1	DDRSE_PTV_NORTH_SIDE_N0	DDRSE_PTV_NORTH_SIDE_P5	DDRSE_PTV_NORTH_SIDE_P4	DDRSE_PTV_NORTH_SIDE_P3	DDRSE_PTV_NORTH_SIDE_P2	DDRSE_PTV_NORTH_SIDE_P1	DDRSE_PTV_NORTH_SIDE_P0	DDRSE_PTV_EAST_SIDE_N5	DDRSE_PTV_EAST_SIDE_N4	DDRSE_PTV_EAST_SIDE_N3	DDRSE_PTV_EAST_SIDE_N2	DDRSE_PTV_EAST_SIDE_N1	DDRSE_PTV_EAST_SIDE_N0	DDRSE_PTV_EAST_SIDE_P5	DDRSE_PTV_EAST_SIDE_P4	DDRSE_PTV_EAST_SIDE_P3	DDRSE_PTV_EAST_SIDE_P2	DDRSE_PTV_EAST_SIDE_P1	DDRSE_PTV_EAST_SIDE_P0	RESERVED							

Bits	Field Name	Description	Type	Reset
31	DDRSE_PTV_NORTH_SIDE_N5		RW	0x0
30	DDRSE_PTV_NORTH_SIDE_N4		RW	0x0
29	DDRSE_PTV_NORTH_SIDE_N3		RW	0x0
28	DDRSE_PTV_NORTH_SIDE_N2		RW	0x0
27	DDRSE_PTV_NORTH_SIDE_N1		RW	0x0
26	DDRSE_PTV_NORTH_SIDE_N0		RW	0x0
25	DDRSE_PTV_NORTH_SIDE_P5		RW	0x0
24	DDRSE_PTV_NORTH_SIDE_P4		RW	0x0
23	DDRSE_PTV_NORTH_SIDE_P3		RW	0x0
22	DDRSE_PTV_NORTH_SIDE_P2		RW	0x0
21	DDRSE_PTV_NORTH_SIDE_P1		RW	0x0
20	DDRSE_PTV_NORTH_SIDE_P0		RW	0x0
19	DDRSE_PTV_EAST_SIDE_N5		RW	0x0
18	DDRSE_PTV_EAST_SIDE_N4		RW	0x0
17	DDRSE_PTV_EAST_SIDE_N3		RW	0x0
16	DDRSE_PTV_EAST_SIDE_N2		RW	0x0
15	DDRSE_PTV_EAST_SIDE_N1		RW	0x0
14	DDRSE_PTV_EAST_SIDE_N0		RW	0x0

Bits	Field Name	Description	Type	Reset
13	DDRSE_PTV_EAST_SIDE_P5		RW	0x0
12	DDRSE_PTV_EAST_SIDE_P4		RW	0x0
11	DDRSE_PTV_EAST_SIDE_P3		RW	0x0
10	DDRSE_PTV_EAST_SIDE_P2		RW	0x0
9	DDRSE_PTV_EAST_SIDE_P1		RW	0x0
8	DDRSE_PTV_EAST_SIDE_P0		RW	0x0
7:0	RESERVED		R	0x0

Table 13-1004. Register Call Summary for Register CTRL_WKUP_EFUSE_3

Control Module Register Manual
- CTRL_MODULE_WKUP Register Summary: [0]

Table 13-1005. CTRL_WKUP_EFUSE_4

Address Offset	0x0000 05D4		
Physical Address	0x4AE0 C5D4	**Instance**	CTRL_MODULE_WKUP
Description	EFUSE compensation 4		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DDRSE_PTV_SOUTH_SIDE_N5	DDRSE_PTV_SOUTH_SIDE_N4	DDRSE_PTV_SOUTH_SIDE_N3	DDRSE_PTV_SOUTH_SIDE_N2	DDRSE_PTV_SOUTH_SIDE_N1	DDRSE_PTV_SOUTH_SIDE_N0	DDRSE_PTV_SOUTH_SIDE_P5	DDRSE_PTV_SOUTH_SIDE_P4	DDRSE_PTV_SOUTH_SIDE_P3	DDRSE_PTV_SOUTH_SIDE_P2	DDRSE_PTV_SOUTH_SIDE_P1	DDRSE_PTV_SOUTH_SIDE_P0	DDRSE_PTV_WEST_SIDE_N5	DDRSE_PTV_WEST_SIDE_N4	DDRSE_PTV_WEST_SIDE_N3	DDRSE_PTV_WEST_SIDE_N2	DDRSE_PTV_WEST_SIDE_N1	DDRSE_PTV_WEST_SIDE_N0	DDRSE_PTV_WEST_SIDE_P5	DDRSE_PTV_WEST_SIDE_P4	DDRSE_PTV_WEST_SIDE_P3	DDRSE_PTV_WEST_SIDE_P2	DDRSE_PTV_WEST_SIDE_P1	DDRSE_PTV_WEST_SIDE_P0	RESERVED							

Bits	Field Name	Description	Type	Reset
31	DDRSE_PTV_SOUTH_SIDE_N5		RW	0x0
30	DDRSE_PTV_SOUTH_SIDE_N4		RW	0x0
29	DDRSE_PTV_SOUTH_SIDE_N3		RW	0x0
28	DDRSE_PTV_SOUTH_SIDE_N2		RW	0x0
27	DDRSE_PTV_SOUTH_SIDE_N1		RW	0x0
26	DDRSE_PTV_SOUTH_SIDE_N0		RW	0x0
25	DDRSE_PTV_SOUTH_SIDE_P5		RW	0x0
24	DDRSE_PTV_SOUTH_SIDE_P4		RW	0x0

Bits	Field Name	Description	Type	Reset
23	DDRSE_PTV_SOUTH_SIDE_P3		RW	0x0
22	DDRSE_PTV_SOUTH_SIDE_P2		RW	0x0
21	DDRSE_PTV_SOUTH_SIDE_P1		RW	0x0
20	DDRSE_PTV_SOUTH_SIDE_P0		RW	0x0
19	DDRSE_PTV_WEST_SIDE_N5		RW	0x0
18	DDRSE_PTV_WEST_SIDE_N4		RW	0x0
17	DDRSE_PTV_WEST_SIDE_N3		RW	0x0
16	DDRSE_PTV_WEST_SIDE_N2		RW	0x0
15	DDRSE_PTV_WEST_SIDE_N1		RW	0x0
14	DDRSE_PTV_WEST_SIDE_N0		RW	0x0
13	DDRSE_PTV_WEST_SIDE_P5		RW	0x0
12	DDRSE_PTV_WEST_SIDE_P4		RW	0x0
11	DDRSE_PTV_WEST_SIDE_P3		RW	0x0
10	DDRSE_PTV_WEST_SIDE_P2		RW	0x0
9	DDRSE_PTV_WEST_SIDE_P1		RW	0x0
8	DDRSE_PTV_WEST_SIDE_P0		RW	0x0
7:0	RESERVED		R	0x0

Table 13-1006. Register Call Summary for Register CTRL_WKUP_EFUSE_4

Control Module Register Manual

- CTRL_MODULE_WKUP Register Summary: [0]

Table 13-1007. CTRL_WKUP_EFUSE_13

Address Offset	0x0000 05F8		
Physical Address	0x4AE0 C5F8	**Instance**	CTRL_MODULE_WKUP
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDIO1833_PTV_N5	SDIO1833_PTV_N4	SDIO1833_PTV_N3	SDIO1833_PTV_N2	SDIO1833_PTV_N1	SDIO1833_PTV_N0	SDIO1833_PTV_P5	SDIO1833_PTV_P4	SDIO1833_PTV_P3	SDIO1833_PTV_P2	SDIO1833_PTV_P1	SDIO1833_PTV_P0	RESERVED																			

Bits	Field Name	Description	Type	Reset
31	SDIO1833_PTV_N5		RW	0x0
30	SDIO1833_PTV_N4		RW	0x0
29	SDIO1833_PTV_N3		RW	0x0
28	SDIO1833_PTV_N2		RW	0x0
27	SDIO1833_PTV_N1		RW	0x0
26	SDIO1833_PTV_N0		RW	0x0
25	SDIO1833_PTV_P5		RW	0x0
24	SDIO1833_PTV_P4		RW	0x0
23	SDIO1833_PTV_P3		RW	0x0
22	SDIO1833_PTV_P2		RW	0x0
21	SDIO1833_PTV_P1		RW	0x0
20	SDIO1833_PTV_P0		RW	0x0
19:0	RESERVED		R	0x0

Table 13-1008. Register Call Summary for Register CTRL_WKUP_EFUSE_13

Control Module Register Manual
- CTRL_MODULE_WKUP Register Summary: [0]

Table 13-1009. CTRL_WKUP_CONF_DEBUG_SEL_TST_0

Address Offset	0x0000 0700		
Physical Address	0x4AE0 C700	**Instance**	CTRL_MODULE_WKUP
Description	This register is used to select an observable signal for WKUP observability line 0.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															MODE

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	MODE	Selects one of the following signals to be available on WKUP observability line 0. This signal can then be mapped to obs0 line through CTRL_CORE_HWOBS_FINAL_MUX_SEL[0] bit. 0x0: hwobs_int_prm(0) 0x1: hwobs_int_cm1(0)	RW	0x0

Table 13-1010. Register Call Summary for Register CTRL_WKUP_CONF_DEBUG_SEL_TST_0

Control Module Functional Description
- Hardware Observability Related Registers: [0]

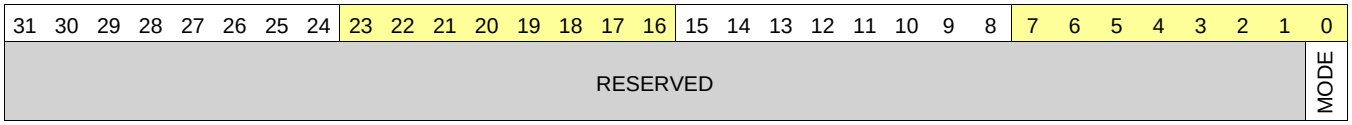
Control Module Register Manual
- CTRL_MODULE_WKUP Register Summary: [1]

Table 13-1011. CTRL_WKUP_CONF_DEBUG_SEL_TST_1

Address Offset	0x0000 0704		
Physical Address	0x4AE0 C704	**Instance**	CTRL_MODULE_WKUP
Description	This register is used to select an observable signal for WKUP observability line 1.		

Table 13-1011. CTRL_WKUP_CONF_DEBUG_SEL_TST_1 (continued)

Type	RW



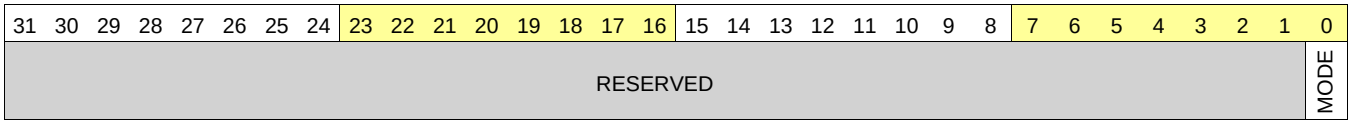
Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	MODE	Selects one of the following signals to be available on WKUP observability line 1. This signal can then be mapped to obs1 line through CTRL_CORE_HWOBS_FINAL_MUX_SEL[1] bit. 0x0: hwobs_int_prm(1) 0x1: hwobs_int_cm1(1)	RW	0x0

Table 13-1012. Register Call Summary for Register CTRL_WKUP_CONF_DEBUG_SEL_TST_1

Control Module Register Manual
- CTRL_MODULE_WKUP Register Summary: [0]

Table 13-1013. CTRL_WKUP_CONF_DEBUG_SEL_TST_2

Address Offset	0x0000 0708		
Physical Address	0x4AE0 C708	**Instance**	CTRL_MODULE_WKUP
Description	This register is used to select an observable signal for WKUP observability line 2.		
Type	RW		



Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	MODE	Selects one of the following signals to be available on WKUP observability line 2. This signal can then be mapped to obs2 line through CTRL_CORE_HWOBS_FINAL_MUX_SEL[2] bit. 0x0: hwobs_int_prm(2) 0x1: hwobs_int_cm1(2)	RW	0x0

Table 13-1014. Register Call Summary for Register CTRL_WKUP_CONF_DEBUG_SEL_TST_2

Control Module Register Manual
- CTRL_MODULE_WKUP Register Summary: [0]

Table 13-1015. CTRL_WKUP_CONF_DEBUG_SEL_TST_3

Address Offset	0x0000 070C		
Physical Address	0x4AE0 C70C	**Instance**	CTRL_MODULE_WKUP
Description	This register is used to select an observable signal for WKUP observability line 3.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															MODE

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	MODE	Selects one of the following signals to be available on WKUP observability line 3. This signal can then be mapped to obs3 line through CTRL_CORE_HWOBS_FINAL_MUX_SEL[3] bit. 0x0: hwobs_int_prm(3) 0x1: hwobs_int_cm1(3)	RW	0x0

Table 13-1016. Register Call Summary for Register CTRL_WKUP_CONF_DEBUG_SEL_TST_3

Control Module Register Manual
- CTRL_MODULE_WKUP Register Summary: [0]

Table 13-1017. CTRL_WKUP_CONF_DEBUG_SEL_TST_4

Address Offset	0x0000 0710		
Physical Address	0x4AE0 C710	**Instance**	CTRL_MODULE_WKUP
Description	This register is used to select an observable signal for WKUP observability line 4.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															MODE

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	MODE	Selects one of the following signals to be available on WKUP observability line 4. This signal can then be mapped to obs4 line through CTRL_CORE_HWOBS_FINAL_MUX_SEL[4] bit. 0x0: hwobs_int_prm(4) 0x1: hwobs_int_cm1(4)	RW	0x0

Table 13-1018. Register Call Summary for Register CTRL_WKUP_CONF_DEBUG_SEL_TST_4

Control Module Register Manual
- CTRL_MODULE_WKUP Register Summary: [0]

Table 13-1019. CTRL_WKUP_CONF_DEBUG_SEL_TST_5

Address Offset	0x0000 0714		
Physical Address	0x4AE0 C714	**Instance**	CTRL_MODULE_WKUP
Description	This register is used to select an observable signal for WKUP observability line 5.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															MODE

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	MODE	Selects one of the following signals to be available on WKUP observability line 5. This signal can then be mapped to obs5 line through CTRL_CORE_HWOBS_FINAL_MUX_SEL[5] bit. 0x0: hwobs_int_prm(5) 0x1: hwobs_int_cm1(5)	RW	0x0

Table 13-1020. Register Call Summary for Register CTRL_WKUP_CONF_DEBUG_SEL_TST_5

Control Module Register Manual
- CTRL_MODULE_WKUP Register Summary: [0]

Table 13-1021. CTRL_WKUP_CONF_DEBUG_SEL_TST_6

Address Offset	0x0000 0718		
Physical Address	0x4AE0 C718	**Instance**	CTRL_MODULE_WKUP
Description	This register is used to select an observable signal for WKUP observability line 6.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															MODE

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	MODE	Selects one of the following signals to be available on WKUP observability line 6. This signal can then be mapped to obs6 line through CTRL_CORE_HWOBS_FINAL_MUX_SEL[6] bit. 0x0: hwobs_int_prm(6) 0x1: hwobs_int_cm1(6)	RW	0x0

Table 13-1022. Register Call Summary for Register CTRL_WKUP_CONF_DEBUG_SEL_TST_6

Control Module Register Manual
- CTRL_MODULE_WKUP Register Summary: [0]

Table 13-1023. CTRL_WKUP_CONF_DEBUG_SEL_TST_7

Address Offset	0x0000 071C		
Physical Address	0x4AE0 C71C	**Instance**	CTRL_MODULE_WKUP
Description	This register is used to select an observable signal for WKUP observability line 7.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															MODE

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	MODE	Selects one of the following signals to be available on WKUP observability line 7. This signal can then be mapped to obs7 line through CTRL_CORE_HWOBS_FINAL_MUX_SEL[7] bit. 0x0: hwobs_int_prm(7) 0x1: hwobs_int_cm1(7)	RW	0x0

Table 13-1024. Register Call Summary for Register CTRL_WKUP_CONF_DEBUG_SEL_TST_7

Control Module Register Manual
- CTRL_MODULE_WKUP Register Summary: [0]

Table 13-1025. CTRL_WKUP_CONF_DEBUG_SEL_TST_8

Address Offset	0x0000 0720		
Physical Address	0x4AE0 C720	**Instance**	CTRL_MODULE_WKUP
Description	This register is used to select an observable signal for WKUP observability line 8.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															MODE

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	MODE	Selects one of the following signals to be available on WKUP observability line 8. This signal can then be mapped to obs8 line through CTRL_CORE_HWOBS_FINAL_MUX_SEL[8] bit. 0x0: hwobs_int_prm(8) 0x1: hwobs_int_cm1(8)	RW	0x0

Table 13-1026. Register Call Summary for Register CTRL_WKUP_CONF_DEBUG_SEL_TST_8

Control Module Register Manual
- CTRL_MODULE_WKUP Register Summary: [0]

Table 13-1027. CTRL_WKUP_CONF_DEBUG_SEL_TST_9

Address Offset	0x0000 0724		
Physical Address	0x4AE0 C724	**Instance**	CTRL_MODULE_WKUP
Description	This register is used to select an observable signal for WKUP observability line 9.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															MODE

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	MODE	Selects one of the following signals to be available on WKUP observability line 9. This signal can then be mapped to obs9 line through CTRL_CORE_HWOBS_FINAL_MUX_SEL[9] bit. 0x0: hwobs_int_prm(9) 0x1: hwobs_int_cm1(9)	RW	0x0

Table 13-1028. Register Call Summary for Register CTRL_WKUP_CONF_DEBUG_SEL_TST_9

Control Module Register Manual
- CTRL_MODULE_WKUP Register Summary: [0]

Table 13-1029. CTRL_WKUP_CONF_DEBUG_SEL_TST_10

Address Offset	0x0000 0728		
Physical Address	0x4AE0 C728	**Instance**	CTRL_MODULE_WKUP
Description	This register is used to select an observable signal for WKUP observability line 10.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															MODE

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	MODE	Selects one of the following signals to be available on WKUP observability line 10. This signal can then be mapped to obs10 line through CTRL_CORE_HWOBS_FINAL_MUX_SEL[10] bit. 0x0: hwobs_int_prm(10) 0x1: hwobs_int_cm1(10)	RW	0x0

Table 13-1030. Register Call Summary for Register CTRL_WKUP_CONF_DEBUG_SEL_TST_10

Control Module Register Manual
- CTRL_MODULE_WKUP Register Summary: [0]

Table 13-1031. CTRL_WKUP_CONF_DEBUG_SEL_TST_11

Address Offset	0x0000 072C		
Physical Address	0x4AE0 C72C	**Instance**	CTRL_MODULE_WKUP
Description	This register is used to select an observable signal for WKUP observability line 11.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															MODE

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	MODE	Selects one of the following signals to be available on WKUP observability line 11. This signal can then be mapped to obs11 line through CTRL_CORE_HWOBS_FINAL_MUX_SEL[11] bit. 0x0: hwobs_int_prm(11) 0x1: hwobs_int_cm1(11)	RW	0x0

Table 13-1032. Register Call Summary for Register CTRL_WKUP_CONF_DEBUG_SEL_TST_11

Control Module Register Manual
- CTRL_MODULE_WKUP Register Summary: [0]

Table 13-1033. CTRL_WKUP_CONF_DEBUG_SEL_TST_12

Address Offset	0x0000 0730		
Physical Address	0x4AE0 C730	**Instance**	CTRL_MODULE_WKUP
Description	This register is used to select an observable signal for WKUP observability line 12.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															MODE

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	MODE	Selects one of the following signals to be available on WKUP observability line 12. This signal can then be mapped to obs12 line through CTRL_CORE_HWOBS_FINAL_MUX_SEL[12] bit. 0x0: hwobs_int_prm(12) 0x1: hwobs_int_cm1(12)	RW	0x0

Table 13-1034. Register Call Summary for Register CTRL_WKUP_CONF_DEBUG_SEL_TST_12

Control Module Register Manual
- CTRL_MODULE_WKUP Register Summary: [0]

Table 13-1035. CTRL_WKUP_CONF_DEBUG_SEL_TST_13

Address Offset	0x0000 0734		
Physical Address	0x4AE0 C734	**Instance**	CTRL_MODULE_WKUP
Description	This register is used to select an observable signal for WKUP observability line 13.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															MODE

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	MODE	Selects one of the following signals to be available on WKUP observability line 13. This signal can then be mapped to obs13 line through CTRL_CORE_HWOBS_FINAL_MUX_SEL[13] bit. 0x0: hwobs_int_prm(13) 0x1: hwobs_int_cm1(13)	RW	0x0

Table 13-1036. Register Call Summary for Register CTRL_WKUP_CONF_DEBUG_SEL_TST_13

Control Module Register Manual
- CTRL_MODULE_WKUP Register Summary: [0]

Table 13-1037. CTRL_WKUP_CONF_DEBUG_SEL_TST_14

Address Offset	0x0000 0738		
Physical Address	0x4AE0 C738	**Instance**	CTRL_MODULE_WKUP
Description	This register is used to select an observable signal for WKUP observability line 14.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															MODE

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	MODE	Selects one of the following signals to be available on WKUP observability line 14. This signal can then be mapped to obs14 line through CTRL_CORE_HWOBS_FINAL_MUX_SEL[14] bit. 0x0: hwobs_int_prm(14) 0x1: hwobs_int_cm1(14)	RW	0x0

Table 13-1038. Register Call Summary for Register CTRL_WKUP_CONF_DEBUG_SEL_TST_14

Control Module Register Manual
- CTRL_MODULE_WKUP Register Summary: [0]

Table 13-1039. CTRL_WKUP_CONF_DEBUG_SEL_TST_15

Address Offset	0x0000 073C		
Physical Address	0x4AE0 C73C	**Instance**	CTRL_MODULE_WKUP
Description	This register is used to select an observable signal for WKUP observability line 15.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															MODE

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	MODE	Selects one of the following signals to be available on WKUP observability line 15. This signal can then be mapped to obs15 line through CTRL_CORE_HWOBS_FINAL_MUX_SEL[15] bit. 0x0: hwobs_int_prm(15) 0x1: hwobs_int_cm1(15)	RW	0x0

Table 13-1040. Register Call Summary for Register CTRL_WKUP_CONF_DEBUG_SEL_TST_15

Control Module Register Manual
- CTRL_MODULE_WKUP Register Summary: [0]

Table 13-1041. CTRL_WKUP_CONF_DEBUG_SEL_TST_16

Address Offset	0x0000 0740		
Physical Address	0x4AE0 C740	**Instance**	CTRL_MODULE_WKUP
Description	This register is used to select an observable signal for WKUP observability line 16.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															MODE

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	MODE	Selects one of the following signals to be available on WKUP observability line 16. This signal can then be mapped to obs16 line through CTRL_CORE_HWOBS_FINAL_MUX_SEL[16] bit. 0x0: hwobs_int_prm(16) 0x1: hwobs_int_cm1(16)	RW	0x0

Table 13-1042. Register Call Summary for Register CTRL_WKUP_CONF_DEBUG_SEL_TST_16

Control Module Register Manual
- CTRL_MODULE_WKUP Register Summary: [0]

Table 13-1043. CTRL_WKUP_CONF_DEBUG_SEL_TST_17

Address Offset	0x0000 0744		
Physical Address	0x4AE0 C744	**Instance**	CTRL_MODULE_WKUP
Description	This register is used to select an observable signal for WKUP observability line 17.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															MODE

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	MODE	Selects one of the following signals to be available on WKUP observability line 17. This signal can then be mapped to obs17 line through CTRL_CORE_HWOBS_FINAL_MUX_SEL[17] bit. 0x0: hwobs_int_prm(17) 0x1: hwobs_int_cm1(17)	RW	0x0

Table 13-1044. Register Call Summary for Register CTRL_WKUP_CONF_DEBUG_SEL_TST_17

Control Module Register Manual
- CTRL_MODULE_WKUP Register Summary: [0]

Table 13-1045. CTRL_WKUP_CONF_DEBUG_SEL_TST_18

Address Offset	0x0000 0748		
Physical Address	0x4AE0 C748	**Instance**	CTRL_MODULE_WKUP
Description	This register is used to select an observable signal for WKUP observability line 18.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															MODE

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	MODE	Selects one of the following signals to be available on WKUP observability line 18. This signal can then be mapped to obs18 line through CTRL_CORE_HWOBS_FINAL_MUX_SEL[18] bit. 0x0: hwobs_int_prm(18) 0x1: hwobs_int_cm1(18)	RW	0x0

Table 13-1046. Register Call Summary for Register CTRL_WKUP_CONF_DEBUG_SEL_TST_18

Control Module Register Manual
- CTRL_MODULE_WKUP Register Summary: [0]

Table 13-1047. CTRL_WKUP_CONF_DEBUG_SEL_TST_19

Address Offset	0x0000 074C		
Physical Address	0x4AE0 C74C	**Instance**	CTRL_MODULE_WKUP
Description	This register is used to select an observable signal for WKUP observability line 19.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															MODE

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	MODE	Selects one of the following signals to be available on WKUP observability line 19. This signal can then be mapped to obs19 line through CTRL_CORE_HWOBS_FINAL_MUX_SEL[19] bit. 0x0: hwobs_int_prm(19) 0x1: hwobs_int_cm1(19)	RW	0x0

Table 13-1048. Register Call Summary for Register CTRL_WKUP_CONF_DEBUG_SEL_TST_19

Control Module Register Manual
- CTRL_MODULE_WKUP Register Summary: [0]

Table 13-1049. CTRL_WKUP_CONF_DEBUG_SEL_TST_20

Address Offset	0x0000 0750		
Physical Address	0x4AE0 C750	**Instance**	CTRL_MODULE_WKUP
Description	This register is used to select an observable signal for WKUP observability line 20.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															MODE

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	MODE	Selects one of the following signals to be available on WKUP observability line 20. This signal can then be mapped to obs20 line through CTRL_CORE_HWOBS_FINAL_MUX_SEL[20] bit. 0x0: hwobs_int_prm(20) 0x1: hwobs_int_cm1(20)	RW	0x0

Table 13-1050. Register Call Summary for Register CTRL_WKUP_CONF_DEBUG_SEL_TST_20

Control Module Register Manual
- CTRL_MODULE_WKUP Register Summary: [0]

Table 13-1051. CTRL_WKUP_CONF_DEBUG_SEL_TST_21

Address Offset	0x0000 0754		
Physical Address	0x4AE0 C754	**Instance**	CTRL_MODULE_WKUP
Description	This register is used to select an observable signal for WKUP observability line 21.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															MODE

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	MODE	Selects one of the following signals to be available on WKUP observability line 21. This signal can then be mapped to obs21 line through CTRL_CORE_HWOBS_FINAL_MUX_SEL[21] bit. 0x0: hwobs_int_prm(21) 0x1: hwobs_int_cm1(21)	RW	0x0

Table 13-1052. Register Call Summary for Register CTRL_WKUP_CONF_DEBUG_SEL_TST_21

Control Module Register Manual
- CTRL_MODULE_WKUP Register Summary: [0]

Table 13-1053. CTRL_WKUP_CONF_DEBUG_SEL_TST_22

Address Offset	0x0000 0758		
Physical Address	0x4AE0 C758	**Instance**	CTRL_MODULE_WKUP
Description	This register is used to select an observable signal for WKUP observability line 22.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															MODE

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	MODE	Selects one of the following signals to be available on WKUP observability line 22. This signal can then be mapped to obs22 line through CTRL_CORE_HWOBS_FINAL_MUX_SEL[22] bit. 0x0: hwobs_int_prm(22) 0x1: hwobs_int_cm1(22)	RW	0x0

Table 13-1054. Register Call Summary for Register CTRL_WKUP_CONF_DEBUG_SEL_TST_22

Control Module Register Manual
- CTRL_MODULE_WKUP Register Summary: [0]

Table 13-1055. CTRL_WKUP_CONF_DEBUG_SEL_TST_23

Address Offset	0x0000 075C		
Physical Address	0x4AE0 C75C	**Instance**	CTRL_MODULE_WKUP
Description	This register is used to select an observable signal for WKUP observability line 23.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															MODE

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	MODE	Selects one of the following signals to be available on WKUP observability line 23. This signal can then be mapped to obs23 line through CTRL_CORE_HWOBS_FINAL_MUX_SEL[23] bit. 0x0: hwobs_int_prm(23) 0x1: hwobs_int_cm1(23)	RW	0x0

Table 13-1056. Register Call Summary for Register CTRL_WKUP_CONF_DEBUG_SEL_TST_23

Control Module Register Manual
- CTRL_MODULE_WKUP Register Summary: [0]

Table 13-1057. CTRL_WKUP_CONF_DEBUG_SEL_TST_24

Address Offset	0x0000 0760		
Physical Address	0x4AE0 C760	**Instance**	CTRL_MODULE_WKUP
Description	This register is used to select an observable signal for WKUP observability line 24.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															MODE

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	MODE	Selects one of the following signals to be available on WKUP observability line 24. This signal can then be mapped to obs24 line through CTRL_CORE_HWOBS_FINAL_MUX_SEL[24] bit. 0x0: hwobs_int_prm(24) 0x1: hwobs_int_cm1(24)	RW	0x0

Table 13-1058. Register Call Summary for Register CTRL_WKUP_CONF_DEBUG_SEL_TST_24

Control Module Register Manual
- CTRL_MODULE_WKUP Register Summary: [0]

Table 13-1059. CTRL_WKUP_CONF_DEBUG_SEL_TST_25

Address Offset	0x0000 0764		
Physical Address	0x4AE0 C764	**Instance**	CTRL_MODULE_WKUP
Description	This register is used to select an observable signal for WKUP observability line 25.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															MODE

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	MODE	Selects one of the following signals to be available on WKUP observability line 25. This signal can then be mapped to obs25 line through CTRL_CORE_HWOBS_FINAL_MUX_SEL[25] bit. 0x0: hwobs_int_prm(25) 0x1: hwobs_int_cm1(25)	RW	0x0

Table 13-1060. Register Call Summary for Register CTRL_WKUP_CONF_DEBUG_SEL_TST_25

Control Module Register Manual
- CTRL_MODULE_WKUP Register Summary: [0]

Table 13-1061. CTRL_WKUP_CONF_DEBUG_SEL_TST_26

Address Offset	0x0000 0768		
Physical Address	0x4AE0 C768	**Instance**	CTRL_MODULE_WKUP
Description	This register is used to select an observable signal for WKUP observability line 26.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															MODE

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	MODE	Selects one of the following signals to be available on WKUP observability line 26. This signal can then be mapped to obs26 line through CTRL_CORE_HWOBS_FINAL_MUX_SEL[26] bit. 0x0: hwobs_int_prm(26) 0x1: hwobs_int_cm1(26)	RW	0x0

Table 13-1062. Register Call Summary for Register CTRL_WKUP_CONF_DEBUG_SEL_TST_26

Control Module Register Manual
- CTRL_MODULE_WKUP Register Summary: [0]

Table 13-1063. CTRL_WKUP_CONF_DEBUG_SEL_TST_27

Address Offset	0x0000 076C		
Physical Address	0x4AE0 C76C	**Instance**	CTRL_MODULE_WKUP
Description	This register is used to select an observable signal for WKUP observability line 27.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															MODE

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	MODE	Selects one of the following signals to be available on WKUP observability line 27. This signal can then be mapped to obs27 line through CTRL_CORE_HWOBS_FINAL_MUX_SEL[27] bit. 0x0: hwobs_int_prm(27) 0x1: hwobs_int_cm1(27)	RW	0x0

Table 13-1064. Register Call Summary for Register CTRL_WKUP_CONF_DEBUG_SEL_TST_27

Control Module Register Manual
- CTRL_MODULE_WKUP Register Summary: [0]

Table 13-1065. CTRL_WKUP_CONF_DEBUG_SEL_TST_28

Address Offset	0x0000 0770		
Physical Address	0x4AE0 C770	**Instance**	CTRL_MODULE_WKUP
Description	This register is used to select an observable signal for WKUP observability line 28.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															MODE

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	MODE	Selects one of the following signals to be available on WKUP observability line 28. This signal can then be mapped to obs28 line through CTRL_CORE_HWOBS_FINAL_MUX_SEL[28] bit. 0x0: hwobs_int_prm(28) 0x1: hwobs_int_cm1(28)	RW	0x0

Table 13-1066. Register Call Summary for Register CTRL_WKUP_CONF_DEBUG_SEL_TST_28

Control Module Register Manual
- CTRL_MODULE_WKUP Register Summary: [0]

Table 13-1067. CTRL_WKUP_CONF_DEBUG_SEL_TST_29

Address Offset	0x0000 0774		
Physical Address	0x4AE0 C774	**Instance**	CTRL_MODULE_WKUP
Description	This register is used to select an observable signal for WKUP observability line 29.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															MODE

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	MODE	Selects one of the following signals to be available on WKUP observability line 29. This signal can then be mapped to obs29 line through CTRL_CORE_HWOBS_FINAL_MUX_SEL[29] bit. 0x0: hwobs_int_prm(29) 0x1: hwobs_int_cm1(29)	RW	0x0

Table 13-1068. Register Call Summary for Register CTRL_WKUP_CONF_DEBUG_SEL_TST_29

Control Module Register Manual
- CTRL_MODULE_WKUP Register Summary: [0]

Table 13-1069. CTRL_WKUP_CONF_DEBUG_SEL_TST_30

Address Offset	0x0000 0778		
Physical Address	0x4AE0 C778	**Instance**	CTRL_MODULE_WKUP
Description	This register is used to select an observable signal for WKUP observability line 30.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															MODE

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	MODE	Selects one of the following signals to be available on WKUP observability line 30. This signal can then be mapped to obs30 line through CTRL_CORE_HWOBS_FINAL_MUX_SEL[30] bit. 0x0: hwobs_int_prm(30) 0x1: hwobs_int_cm1(30)	RW	0x0

Table 13-1070. Register Call Summary for Register CTRL_WKUP_CONF_DEBUG_SEL_TST_30

Control Module Register Manual
- CTRL_MODULE_WKUP Register Summary: [0]

Table 13-1071. CTRL_WKUP_CONF_DEBUG_SEL_TST_31

Address Offset	0x0000 077C		
Physical Address	0x4AE0 C77C	**Instance**	CTRL_MODULE_WKUP
Description	This register is used to select an observable signal for WKUP observability line 31.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															MODE

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	MODE	Selects one of the following signals to be available on WKUP observability line 31. This signal can then be mapped to obs31 line through CTRL_CORE_HWOBS_FINAL_MUX_SEL[31] bit. 0x0: hwobs_int_prm(31) 0x1: hwobs_int_cm1(31)	RW	0x0

Table 13-1072. Register Call Summary for Register CTRL_WKUP_CONF_DEBUG_SEL_TST_31

Control Module Functional Description • Hardware Observability Related Registers: [0]
Control Module Register Manual • CTRL_MODULE_WKUP Register Summary: [1]

Mailbox

This chapter describes the mailbox module in the device.

Topic **Page**

14.1 Mailbox Overview

Communication between the on-chip processors of the device uses a queued mailbox-interrupt mechanism.

The queued mailbox-interrupt mechanism allows the software to establish a communication channel between two processors through a set of registers and associated interrupt signals by sending and receiving messages (mailboxes).

The device implements the following mailbox types:

- System mailbox:
  - Number of instances: 2
  - Used for communication between: DSP1, DSP2, and IPU subsystems
  - Reference names: MAILBOX1, MAILBOX2
- EVE mailbox:
  - Number of instances: 2
  - Used for communication between: EVE local user (ARP32) and three external users (DSP1, DSP2, and IPU) – EVE_MBOX0 and EVE_MBOX1 are dedicated for this communication
  - Reference names: EVE_MBOX0, EVE_MBOX1

Each mailbox module supports the following features:

- Parameters configurable at design time (see Table 14-1):
  - Number of users
  - Number of mailbox message queues
  - Number of messages (FIFO depth) for each message queue
- 32-bit message width
- Message reception and queue-not-full notification using interrupts
- Support of 16-/32-bit addressing scheme
- Power management support

Table 14-1 shows the configuartion of the mailbox modules in the device.

Table 14-1. Mailbox Configuration in the Device

Module Parameters	Mailbox Type		
	System Mailbox		EVE Mailbox (0, 1)
	MAILBOX1	MAILBOX2	
Number of users	3	4	4
Number of mailbox message queues	8	12	16
Number of messages (FIFO depth) for each message queue	4		4

14.2 Mailbox Integration

This section describes the mailbox integration in the device, including information about clocks, resets, and hardware requests.

14.2.1 System MAILBOX Integration

Figure 14-1 and Figure 14-2 show the MAILBOX1 and MAILBOX2 integration, respectively.

Figure 14-1. MAILBOX1 Integration

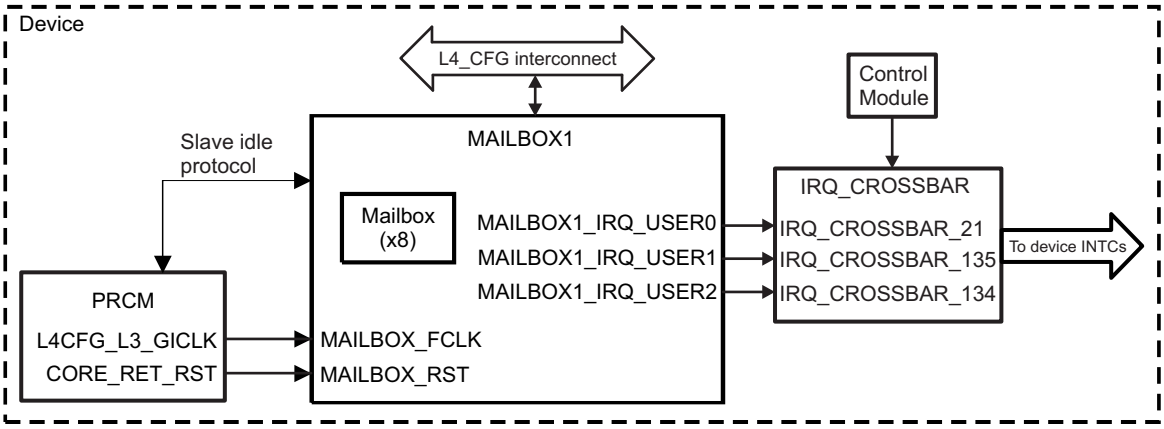
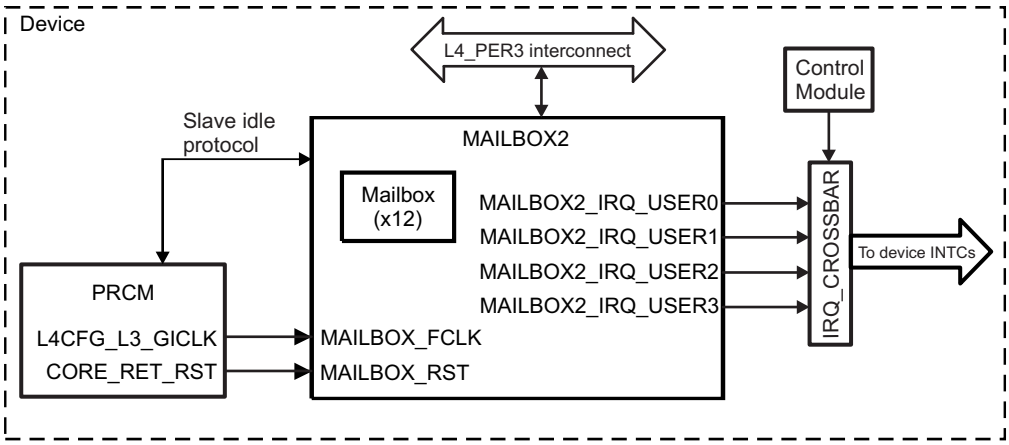


Figure 14-2. MAILBOX2 Integration



NOTE: For more information about the Slave idle protocol, see Section 3.1.1.1.2, *Module-Level Clock Management* in Chapter 3, *Power, Reset, and Clock Management*.

Table 14-2 through Table 14-4 summarize the MAILBOX1/2 integration in the device.

Table 14-2. MAILBOX Integration Attributes

Module Instance	Attributes		
	Power Domain	Wake-Up Capability	Interconnect
MAILBOX1	PD_COREAON	N/A	L4_CFG
MAILBOX2	PD_COREAON	N/A	L4_PER3

Table 14-3. MAILBOX Clocks and Resets

Clocks				
Module Instance	**Destination Signal Name**	**Source Signal Name**	**Source**	**Description**
MAILBOX	MAILBOX_FCLK	L4CFG_L3_GICLK	PRCM	MAILBOX interface clock. This clock is used for all interface and functional operations.
Resets				
Module Instance	**Destination Signal Name**	**Source Signal Name**	**Source**	**Description**

Table 14-3. MAILBOX Clocks and Resets (continued)

MAILBOX	MAILBOX_RST	CORE_RET_RST	PRCM	MAILBOX hardware reset. This reset is asynchronously applied to the MAILBOX registers.

Table 14-4. MAILBOX Hardware Requests

Interrupt Requests							
Module Instance	**Interrupt Name (Source)**	**IRQ_CROSSBAR Input (Destination)**	**Default Mapping**	**Description**			
MAILBOX1	MAILBOX1_IRQ_USER0	IRQ_CROSSBAR_21	DSP1_IRQ_52; DSP2_IRQ_52;	MAILBOX1 request.	user	0	interrupt
	MAILBOX1_IRQ_USER1	IRQ_CROSSBAR_135	–	MAILBOX1 request.	user	1	interrupt
	MAILBOX1_IRQ_USER2	IRQ_CROSSBAR_134	IPU_IRQ_50;	MAILBOX1 request.	user	2	interrupt
MAILBOX2	MAILBOX2_IRQ_USER0	IRQ_CROSSBAR_237	–	MAILBOX2 request.	user	0	interrupt
	MAILBOX2_IRQ_USER1	IRQ_CROSSBAR_238	–	MAILBOX2 request.	user	1	interrupt
	MAILBOX2_IRQ_USER2	IRQ_CROSSBAR_239	–	MAILBOX2 request.	user	2	interrupt
	MAILBOX2_IRQ_USER3	IRQ_CROSSBAR_240	–	MAILBOX2 request.	user	3	interrupt
No DMA Requests							

NOTE: The "**Default Mapping**" column in Table 14-4 and Table 14-7 shows the default mapping of module interrupts. These interrupts can also be mapped to other input lines of each device interrupt controller through the IRQ_CROSSBAR module.

For more information about the IRQ_CROSSBAR module, see Chapter 13, *Control Module*.

For more information about the device interrupt controllers, see Chapter 12, *Interrupt Controllers*.

NOTE: For information about interrupt source description, see Section 14.3.4, *Mailbox Interrupt Requests*.

14.2.2 EVE Mailbox Integration

Figure 14-3 shows the EVE_MBOX integration.

Figure 14-3. EVE_MBOX Integration

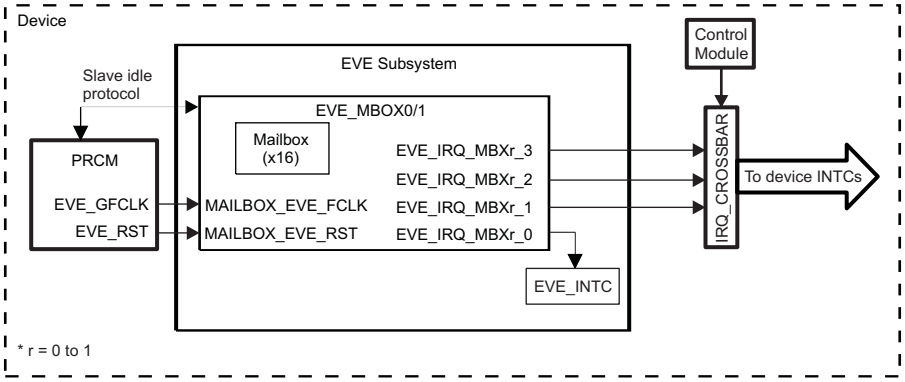


Table 14-5 through Table 14-7 summarize the EVE_MBOX integration in the device.

Table 14-5. EVE_MBOX Integration Attributes

Module Instance	Attributes		
	Power Domain	Wake-Up Capability	Interconnect
EVE_MBOX	PD_EVE	N/A	EVE local interconnect

Table 14-6. EVE_MBOX Clocks and Resets

Clocks				
Module Instance	**Destination Signal Name**	**Source Signal Name**	**Source**	**Description**
EVE_MBOX	MAILBOX_EVE_FCLK	EVE_GFCLK	PRCM	EVE_MBOX interface clock. This clock is used for all interface and functional operations.
Resets				
Module Instance	**Destination Signal Name**	**Source Signal Name**	**Source**	**Description**
EVE_MBOX	MAILBOX_EVE_RST	EVE_RST	PRCM	EVE_MBOX hardware reset. This reset is asynchronously applied to the EVE_MBOX registers.

Table 14-7. EVE_MBOX Hardware Requests

Interrupt Requests				
Module Instance	**Interrupt Name (Source)**	**IRQ_CROSSBAR Input (Destination)**	**Default Mapping**	**Description**
EVE_MBOX0	EVE1_IRQ_MBX0_USER0	TBD	–	EVE_MBOX0 user 0 interrupt request.
	EVE1_IRQ_MBX0_USER1	IRQ_CROSSBAR_284	–	EVE_MBOX0 user 1 interrupt request.
	EVE1_IRQ_MBX0_USER2	IRQ_CROSSBAR_285	–	EVE_MBOX0 user 2 interrupt request.
	EVE1_IRQ_MBX0_USER3	IRQ_CROSSBAR_286	–	EVE_MBOX0 user 3 interrupt request.
EVE_MBOX1	EVE1_IRQ_MBX1_USER0	TBD	–	EVE_MBOX1 user 0 interrupt request.
	EVE1_IRQ_MBX1_USER1	IRQ_CROSSBAR_287	–	EVE_MBOX1 user 1 interrupt request.
	EVE1_IRQ_MBX1_USER2	IRQ_CROSSBAR_288	–	EVE_MBOX1 user 2 interrupt request.
	EVE1_IRQ_MBX1_USER3	IRQ_CROSSBAR_289	–	EVE_MBOX1 user 3 interrupt request.
No DMA Requests				

14.3 Mailbox Functional Description

NOTE: The functionality of all mailbox instances in the device is the same and is described in this section.

NOTE: In this chapter, *u* is the user number and *m* is the mailbox number as follows:

- For MAILBOX1: *u* = 0 to 2 and *m* = 0 to 7;
- For MAILBOX2: *u* = 0 to 3 and *m* = 0 to 11;
- For EVE_MBOX: *u* = 0 to 3 and *m* = 0 to 15;

The mailbox module provides a means of communication through message queues among the users. The individual mailbox modules, or FIFOs, can associate (or de-associate) with any of the processors using the MAILBOX_IRQENABLE_SET_u (or MAILBOX_IRQENABLE_CLR_u) register.

Table 14-8 shows the potential users of the mailbox modules in the device.

Table 14-8. Mailbox Users in the Device

Mailbox Type		Users			
		User 0	User 1	User 2	User 3
System Mailbox	MAILBOX1	Any of: DSP1, DSP2, and IPU[1]			–
	MAILBOX2	Any of: DSP1, DSP2, and IPU[1]			
EVE Mailbox	EVE_MBOX0, EVE_MBOX1	TBD	Any of: DSP1, DSP2, and IPU[1]		

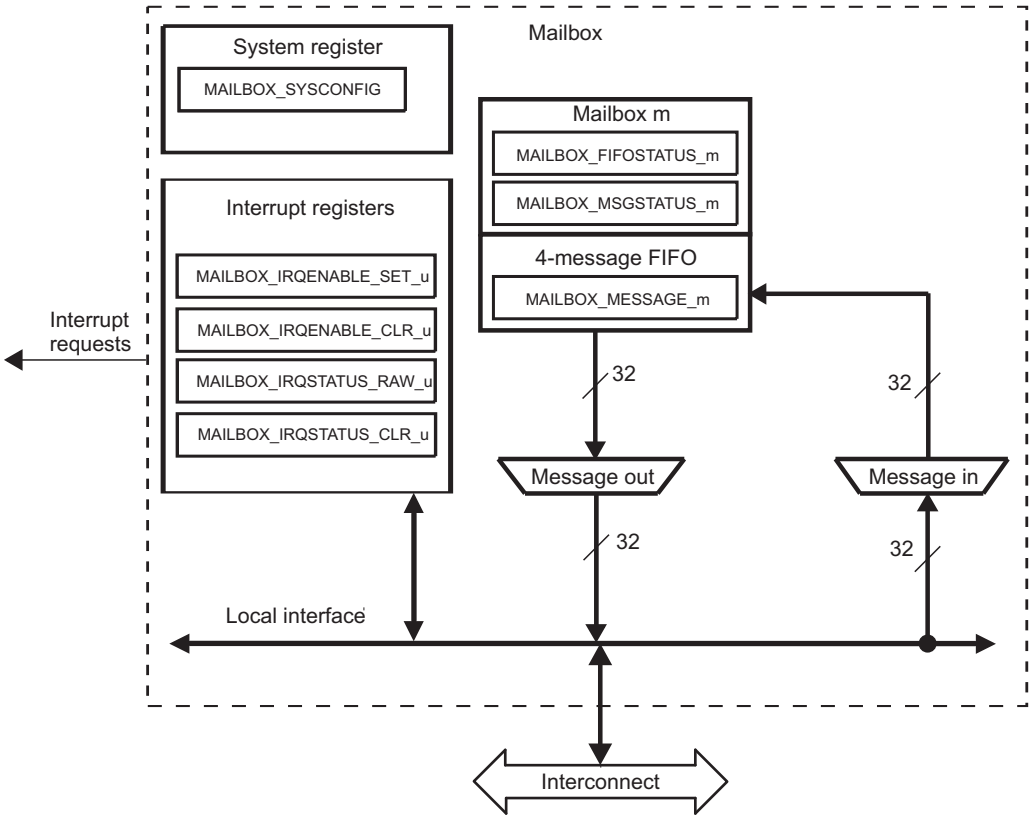
(1) It is software responsibility to select a user by mapping (via IRQ_CROSSBAR) the corresponding mailbox interrupt to the interrupt controller of the appropriate processor subsystem.

Each user has a dedicated interrupt signal from the corresponding mailbox module instance and dedicated interrupt enabling and status registers.
Each MAILBOX_IRQSTATUS_RAW_u/MAILBOX_IRQSTATUS_CLR_u interrupt status register corresponds to a particular user.

14.3.1 Mailbox Block Diagram

Figure 14-4 shows the mailbox block diagram.

Figure 14-4. Mailbox Block Diagram



14.3.2 Mailbox Software Reset

The mailbox module supports a software reset through the MAILBOX_SYSCONFIG[0] SOFTRESET bit. Setting this bit to 1 enables an active software reset that is functionally equivalent to a hardware reset. Reading the MAILBOX_SYSCONFIG[0] SOFTRESET bit gives the status of the software reset:

- Read 1: the software reset is on-going.
- Read 0: the software reset is complete.

The software must ensure that the software reset completes before doing mailbox operations.

14.3.3 Mailbox Power Management

Table 14-9 describes power-management features available for the mailbox module.

Table 14-9. Local Power Management Features

Feature	Registers	Description
Clock autogating	N/A	Feature not available
Slave idle modes	MAILBOX_SYSCONFIG[3:2] SIDLEMODE bit field	Force-idle, no-idle and smart-idle modes are available
Clock activity	N/A	Feature not available
Master standby modes	N/A	Feature not available
Global wake-up enable	N/A	Feature not available
Wake-up sources enable	N/A	Feature not available

The mailbox module can be configured using the MAILBOX_SYSCONFIG[3:2] SIDLEMODE bit field to one of the following acknowledgment modes:

- Force-idle mode (SIDLEMODE = 0x0): The mailbox module immediately enters the idle state on receiving a low-power-mode request from the PRCM module. In this mode, the software must ensure that there are no asserted output interrupts before requesting this mode to go into the idle state.
- No-idle mode (SIDLEMODE = 0x1): The mailbox module never enters the idle state.
- Smart-idle mode (SIDLEMODE = 0x2): After receiving a low-power-mode request from the PRCM module, the mailbox module enters the idle state only after all asserted output interrupts are acknowledged.

14.3.4 Mailbox Interrupt Requests

An interrupt request allows the user of the mailbox to be notified when a message is received or when the message queue is not full. There is one interrupt per user.

Table 14-10 lists the event flags, and their mask, that can cause module interrupts.

Table 14-10. Interrupt Events

Non-Maskable Event Flag[1]	Maskable Event Flag	Event Mask Bit	Event Unmask Bit	Description
MAILBOX_IRQSTATUS_RAW_u[0+m*2] NEWMSGSTATUSUUMBm	MAILBOX_IRQSTATUS_CLR_u[0+m*2] NEWMSGSTATUSUUMBm	MAILBOX_IRQENABLE_CLR_u[0+m*2] NEWMSGSTATUSUUMBm	MAILBOX_IRQENABLE_SET_u[0+m*2] NEWMSGSTATUSUUMBm	Mailbox *m* receives a new message.
MAILBOX_IRQSTATUS_RAW_u[1+m*2] NOTFULLSTATUSUUMBm	MAILBOX_IRQSTATUS_CLR_u[1+m*2] NOTFULLSTATUSUUMBm	MAILBOX_IRQENABLE_CLR_u[1+m*2] NOTFULLSTATUSUUMBm	MAILBOX_IRQENABLE_SET_u[1+m*2] NOTFULLSTATUSUUMBm	Mailbox *m* message queue is not full.

(1) MAILBOX_IRQSTATUS_RAW_u register is mostly used for debug purposes.

CAUTION

Once an event generating the interrupt request has been processed by the software, it must be cleared by writing a logical 1 in the corresponding bit of the MAILBOX_IRQSTATUS_CLR_u register.

Writing a logical 1 in a bit of the MAILBOX_IRQSTATUS_CLR_u register will also clear to 0 the corresponding bit in the appropriate MAILBOX_IRQSTATUS_RAW_u register.

An event can generate an interrupt request when a logical 1 is written to the corresponding unmask bit in the MAILBOX_IRQENABLE_SET_u register. Events are reported in the appropriate MAILBOX_IRQSTATUS_CLR_u and MAILBOX_IRQSTATUS_RAW_u registers.

An event stops generating interrupt requests when a logical 1 is written to the corresponding mask bit in the MAILBOX_IRQENABLE_CLR_u register. Events are only reported in the appropriate MAILBOX_IRQSTATUS_RAW_u register.

In case of the MAILBOX_IRQSTATUS_RAW_u register, the event is reported in the corresponding bit even if the interrupt request generation is disabled for this event.

14.3.5 Mailbox Assignment

14.3.5.1 Description

To assign a receiver to a mailbox, set the new message interrupt enable bit corresponding to the desired mailbox in the MAILBOX_IRQENABLE_SET_u register. The receiver reads the MAILBOX_MESSAGE_m register to retrieve a message from the mailbox.

An alternate method for the receiver that does not use the interrupts is to poll the MAILBOX_FIFOSTATUS_m and/or MAILBOX_MSGSTATUS_m registers to know when to send or retrieve a message to or from the mailbox. This method does not require assigning a receiver to a mailbox. Because this method does not include the explicit assignment of the mailbox, the software must avoid having multiple receivers use the same mailbox, which can result in incoherency.

To assign a sender to a mailbox, set the queue-not-full interrupt enable bit of the desired mailbox in the MAILBOX_IRQENABLE_SET_u register, where *u* is the number of the sending user. However, direct allocation of a mailbox to a sender is not recommended because it can cause the sending processor to be constantly interrupted.

It is recommended that register polling be used to:

- Check the status of either the MAILBOX_FIFOSTATUS_m or MAILBOX_MSGSTATUS_m registers
- Write the message to the corresponding MAILBOX_MESSAGE_m register, if space is available.

The sender might use the queue-not-full interrupt when the initial mailbox status check indicates the mailbox is full. In this case, the sender can enable the queue-not-full interrupt for its mailbox in the appropriate MAILBOX_IRQENABLE_SET_u register. This allows the sender to be notified by interrupt only when a FIFO queue has at least one available entry.

Reading the MAILBOX_IRQSTATUS_CLR_u register determines the status of the new message and the queue-not-full interrupts for a particular user. Writing 1 to the corresponding bit in the MAILBOX_IRQSTATUS_CLR_u register acknowledges, and subsequently clears, an interrupt.

CAUTION

Assigning multiple senders or multiple receivers to the same mailbox is not recommended.

14.3.6 Sending and Receiving Messages

14.3.6.1 Description

When a 32-bit message is written to the MAILBOX_MESSAGE_m register, the message is appended into the FIFO queue. This queue holds four messages. If the queue is full, the message is discarded.

Queue overflow can be avoided by first reading the MAILBOX_FIFOSTATUS_m register to check that the mailbox message queue is not full before writing a new message to it.

Reading the MAILBOX_MESSAGE_m register returns the message at the beginning of the FIFO queue and removes it from the queue. If the FIFO queue is empty when the MAILBOX_MESSAGE_m register is read, the value 0 is returned.

The new message interrupt is asserted when at least one message is in the mailbox message FIFO queue. To determine the number of messages in the mailbox message FIFO queue, read the MAILBOX_MSGSTATUS_m register.

14.3.7 16-Bit Register Access

14.3.7.1 Description

So that 16-bit processors can access the mailbox module, the module allows 16-bit register read and write access, with restrictions for the MAILBOX_MESSAGE_m registers. The 16-bit half-words are organized in little endian fashion; that is, the least-significant 16 bits are at the low address and the most-significant 16 bits are at the high address (low address + 0x02).

All mailbox module registers can be read or written to directly using individual 16-bit accesses with no restriction on interleaving, except the MAILBOX_MESSAGE_m registers, which must always be accessed by either single 32-bit accesses or two consecutive 16-bit accesses.

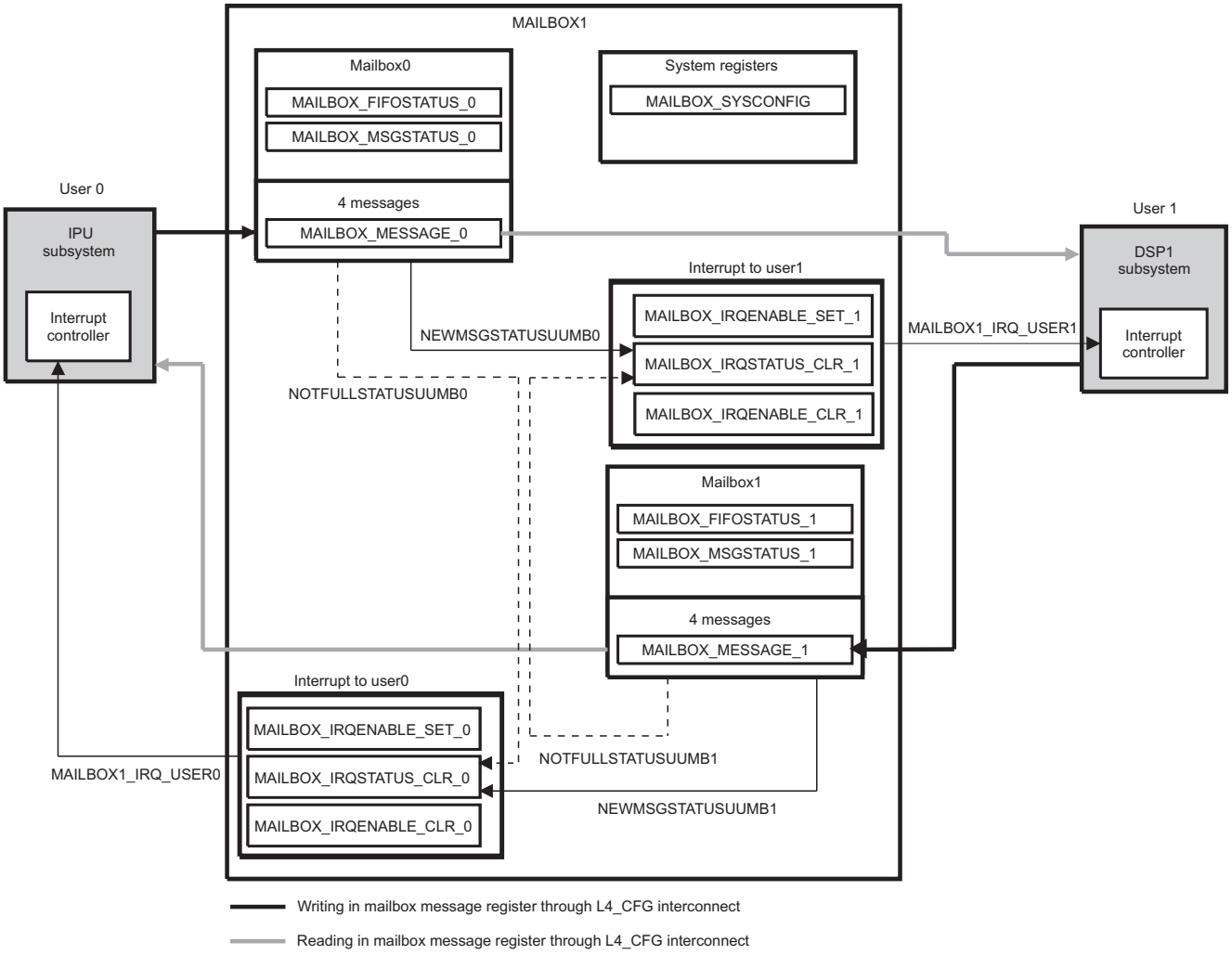
CAUTION

When using 16-bit accesses to the MAILBOX_MESSAGE_m registers, the order of access must be the least-significant half-word first (low address) and the most-significant half-word last (high address). This requirement is because of the update operation by the message FIFO of the MAILBOX_MSGSTATUS_m registers. The update of the FIFO queue contents and the associated status registers and possible interrupt generation occurs only when the most-significant 16 bits of a MAILBOX_MESSAGE_m are accessed.

14.3.8 Example of Communication

Figure 14-5 shows an example of communication between IPU and DSP1 subsystems.

Figure 14-5. Example of Communication



14.4 Mailbox Programming Guide

14.4.1 Mailbox Low-level Programming Models

This section covers the low-level hardware programming sequences for configuration and usage of the mailbox module.

14.4.1.1 Global Initialization

14.4.1.1.1 Surrounding Modules Global Initialization

This section identifies the requirements of initializing the surrounding modules when the mailbox module is to be used for the first time after a device reset. This initialization of surrounding modules is based on the integration of the mailbox.

See Section 14.2, *Mailbox Integration*, for further information.

Table 14-11. Global Initialization of Surrounding Modules for MAILBOX

Surrounding Modules	Comments
PRCM	MAILBOX functional/interface clock must be enabled.
Interrupt Controllers	IPU, or DSP1, or DSP2 interrupt controller must be configured to enable the interrupt request generation to the corresponding subsystem.
IRQ_CROSSBAR	IRQ_CROSSBAR configuration must be done to allow module IRQs to be mapped to certain device INTC line. For more information see Section 13.4.6.4, *IRQ_CROSSBAR Module Functional Description*, in Chapter 13, *Control Module*.

Table 14-12. Global Initialization of Surrounding Modules for EVE_MBOX

Surrounding Modules	Comments
PRCM	EVEx_MBOX functional/interface clock must be enabled.
Interrupt Controllers	IPU, or DSP1, or DSP2, or EVE interrupt controller must be configured to enable the interrupt request generation to the corresponding subsystem.
IRQ_CROSSBAR	IRQ_CROSSBAR configuration must be done to allow module IRQs to be mapped to certain device INTC line. For more information see Section 13.4.6.4, *IRQ_CROSSBAR Module Functional Description*, in Chapter 13, *Control Module*.

14.4.1.1.2 Mailbox Global Initialization

14.4.1.1.2.1 Main Sequence - Mailbox Global Initialization

This procedure initializes the mailbox module after a power-on or software reset.

Table 14-13. Mailbox Global Initialization

Step	Register/ Bit Field / Programming Model	Value
Perform a software reset	MAILBOX_SYSCONFIG[0] SOFTRESET	0x1
Wait until reset is complete	MAILBOX_SYSCONFIG[0] SOFTRESET	= 0x0
Set idle mode configuration	MAILBOX_SYSCONFIG[3:2] SIDLEMODE	0x-

14.4.1.2 Mailbox Operational Modes Configuration

14.4.1.2.1 Mailbox Processing modes

14.4.1.2.1.1 Main Sequence - Sending a Message (Polling Method)

Table 14-14. Sending a Message (Polling Method)

Step	Register/ Bit Field / Programming Model	Value
IF : Is FIFO full ?	MAILBOX_FIFOSTATUS_m[0] FIFOFULLMB	= 0x1
Wait until at least one message slot is available	MAILBOX_FIFOSTATUS_m[0] FIFOFULLMB	= 0x0
ELSE		
Write message	MAILBOX_MESSAGE_m[31:0] MESSAGEVALUEMBM	0x----
ENDIF		

14.4.1.2.1.2 Main Sequence - Sending a Message (Interrupt Method)

Table 14-15. Sending a Message (Interrupt Method)

Step	Register/ Bit Field / Programming Model	Value
IF : Is FIFO full ?	MAILBOX_FIFOSTATUS_m[0] FIFOFULLMB	= 0x1
Enable interrupt event	MAILBOX_IRQENABLE_SET_u[1+ m*2]	0x1
User (processor) can perform another task until interrupt occurs See Section 14.4.1.3.1 for interrupt handling in sending mode		
ELSE		
Write message	MAILBOX_MESSAGE_m[31:0] MESSAGEVALUEMBM	0x----
ENDIF		

14.4.1.2.1.3 Main Sequence - Receiving a Message (Polling Method)

Table 14-16. Receiving a Message (Polling Method)

Step	Register/ Bit Field / Programming Model	Value
IF : Number of messages is not equal to 0	MAILBOX_MSGSTATUS_m[2:0] NBOFMSGMB	!= 0x0
Read message	MAILBOX_MESSAGE_m[31:0] MESSAGEVALUEMBM	0x----
ENDIF		

14.4.1.2.1.4 Main Sequence - Receiving a Message (Interrupt Method)

Table 14-17. Receiving a Message (Interrupt Method)

Step	Register/ Bit Field / Programming Model	Value
Enable interrupt event	MAILBOX_IRQENABLE_SET_u[0 + m*2]	0x1
User (processor) can perform another task until interrupt occurs See Section 14.4.1.3.2 for interrupt handling in receiving mode		

14.4.1.3 Mailbox Events Servicing

14.4.1.3.1 Events Servicing in Sending Mode

Table 14-18 describes the events servicing in sending mode.

Table 14-18. Events Servicing in Sending Mode

Step	Register/ Bit Field / Programming Model	Value
Read interrupt status bit	MAILBOX_IRQSTATUS_CLR_u[1 + m*2]	0x1
Write message	MAILBOX_MESSAGE_m[31:0] MESSAGEVALUEMBM	0x----
Write 1 to acknowledge interrupt	MAILBOX_IRQSTATUS_CLR_u[1 + m*2]	0x1

14.4.1.3.2 Events Servicing in Receiving Mode

Table 14-19 describes the events servicing in receiving mode.

Table 14-19. Events Servicing in Receiving Mode

Step	Register/ Bit Field / Programming Model	Value
Read interrupt status bit	MAILBOX_IRQSTATUS_CLR_u[0 + m*2]	0x1
IF : Number of messages is not equal to 0 ?	MAILBOX_MSGSTATUS_m[2:0] NBOFMSGMB	!= 0x0
Read message	MAILBOX_MESSAGE_m[31:0] MESSAGEVALUEMBM	0x----
ELSE		
Write 1 to acknowledge interrupt	MAILBOX_IRQSTATUS_CLR_u[0 + m*2]	0x1
ENDIF		

14.5 Mailbox Register Manual

14.5.1 Mailbox Instance Summary

Table 14-20. Mailbox Instance Summary

Module Name	L3_MAIN Base Address	L4_CFG Base Address	L4_PER3 Base Address	Size
MAILBOX1	–	0x4A0F 4000	–	4 KiB
MAILBOX2	–	–	0x4883 A000	4 KiB
EVE_MBOX0	0x4208 B000	–	–	4 KiB
EVE_MBOX1	0x4208 C000	–	–	4 KiB

14.5.2 Mailbox Registers

14.5.2.1 Mailbox Register Summary

Table 14-21. MAILBOX Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	MAILBOX1 L4_CFG Physical Address	MAILBOX2 L4_PER3 Physical Address
MAILBOX_REVISION	R	32	0x0000 0000	0x4A0F 4000	0x4883 A000
MAILBOX_SYSCONFIG	RW	32	0x0000 0010	0x4A0F 4010	0x4883 A010
MAILBOX_MESSAGE_m (1)	RW	32	0x0000 0040 + (0x4 * m)	0x4A0F 4040 + (0x4 * m)	0x4883 A040 + (0x4 * m)
MAILBOX_FIFOSTATUS_m (1)	R	32	0x0000 0080 + (0x4 * m)	0x4A0F 4080 + (0x4 * m)	0x4883 A080 + (0x4 * m)
MAILBOX_MSGSTATUS_m (1)	R	32	0x0000 00C0 + (0x4 * m)	0x4A0F 40C0 + (0x4 * m)	0x4883 A0C0 + (0x4 * m)
MAILBOX_IRQSTATUS_RAW_u (2)	RW	32	0x0000 0100 + (0x10 * u)	0x4A0F 4100 + (0x10 * u)	0x4883 A100 + (0x10 * u)
MAILBOX_IRQSTATUS_CLR_u (2)	RW	32	0x0000 0104 + (0x10 * u)	0x4A0F 4104 + (0x10 * u)	0x4883 A104 + (0x10 * u)
MAILBOX_IRQENABLE_SET_u (2)	RW	32	0x0000 0108 + (0x10 * u)	0x4A0F 4108 + (0x10 * u)	0x4883 A108 + (0x10 * u)
MAILBOX_IRQENABLE_CLR_u (2)	RW	32	0x0000 010C + (0x10 * u)	0x4A0F 410C + (0x10 * u)	0x4883 A10C + (0x10 * u)
MAILBOX_IRQ_EOI	W	32	0x0000 0140	0x4A0F 4140	0x4883 A140

(1) m = 0 to 7
(2) u = 0 to 2

Table 14-22. EVE_MBOX Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	EVE_MBOX0 L3_MAIN Physical Address	EVE_MBOX1 L3_MAIN Physical Address
MAILBOX_REVISION	R	32	0x0000 0000	0x4208 B000	0x4208 C000
MAILBOX_SYSCONFIG	RW	32	0x0000 0010	0x4208 B010	0x4208 C010
MAILBOX_MESSAGE_m (1)	RW	32	0x0000 0040 + (0x4 * m)	0x4208 B040 + (0x4 * m)	0x4208 C040 + (0x4 * m)
MAILBOX_FIFOSTATUS_m (1)	R	32	0x0000 0080 + (0x4 * m)	0x4208 B080 + (0x4 * m)	0x4208 C080 + (0x4 * m)
MAILBOX_MSGSTATUS_m (1)	R	32	0x0000 00C0 + (0x4 * m)	0x4208 B0C0 + (0x4 * m)	0x4208 C0C0 + (0x4 * m)

(1) m = 0 to 15

Table 14-22. EVE_MBOX Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	EVE_MBOX0 L3_MAIN Physical Address	EVE_MBOX1 L3_MAIN Physical Address
MAILBOX_IRQSTATUS_RAW_u [2]	RW	32	0x0000 0100 + (0x10 * u)	0x4208 B100 + (0x10 * u)	0x4208 C100 + (0x10 * u)
MAILBOX_IRQSTATUS_CLR_u [2]	RW	32	0x0000 0104 + (0x10 * u)	0x4208 B104 + (0x10 * u)	0x4208 C104 + (0x10 * u)
MAILBOX_IRQENABLE_SET_u [2]	RW	32	0x0000 0108 + (0x10 * u)	0x4208 B108 + (0x10 * u)	0x4208 C108 + (0x10 * u)
MAILBOX_IRQENABLE_CLR_u [2]	RW	32	0x0000 010C + (0x10 * u)	0x4208 B10C + (0x10 * u)	0x4208 C10C + (0x10 * u)
MAILBOX_IRQ_EOI	W	32	0x0000 0140	0x4208 B140	0x4208 C140

[2] u = 0 to 3

14.5.2.2 Mailbox Register Description

Table 14-23. MAILBOX_REVISION

Address Offset	0x0000 0000		
Physical Address	0x4A0F 4000 0x4883 A000 0x4208 B000 0x4208 C000	**Instance**	MAILBOX1 MAILBOX2 EVE_MBOX0 EVE_MBOX1
Description	This register contains the IP revision code		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REVISION																															

Bits	Field Name	Description	Type	Reset
31:0	REVISION	IP Revision	R	TI internal data

Table 14-24. Register Call Summary for Register MAILBOX_REVISION

Mailbox Register Manual
- Mailbox Register Summary: [0][1]

Table 14-25. MAILBOX_SYSCONFIG

Address Offset	0x0000 0010		
Physical Address	0x4A0F 4010 0x4883 A010 0x4208 B010 0x4208 C010	**Instance**	MAILBOX1 MAILBOX2 EVE_MBOX0 EVE_MBOX1
Description	This register controls the various parameters of the communication interface		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												SIDLEMODE		RESERVED	SOFTRESET

Bits	Field Name	Description	Type	Reset
31:4	RESERVED	Reserved	RW	0x0000000
3:2	SIDLEMODE	Idle Mode 0x0: Force-idle. An idle request is acknowledged unconditionally 0x1: No-idle. An idle request is never acknowledged 0x2: Smart-idle. Acknowledgement to an idle request is given based on the internal activity of the module. 0x3: reserved do not use	RW	0x2
1	RESERVED	Reserved	RW	0
0	SOFTRESET	Softreset Read 0x0: Soft/Hard reset done Write 0x0: No action Read 0x1: Reset is ongoing Write 0x1: Start the soft reset sequence	RW	0

Table 14-26. Register Call Summary for Register MAILBOX_SYSCONFIG

Mailbox Functional Description • Mailbox Software Reset: [0][1] • Mailbox Power Management: [2][3]
Mailbox Programming Guide • Mailbox Global Initialization: [4][5][6]
Mailbox Register Manual • Mailbox Register Summary: [7][8]

Table 14-27. MAILBOX_MESSAGE_m

Address Offset	0x0000 0040 + (0x4 * m)	**Index**	m = 0 to 7 (MAILBOX1), or m = 0 to 11 (MAILBOX2), or m = 0 to 15 (EVE_MBOX)
Physical Address	0x4A0F 4040 + (0x4 * m) 0x4883 A040 + (0x4 * m) 0x4208 B040 + (0x4 * m) 0x4208 C040 + (0x4 * m)	**Instance**	MAILBOX1 MAILBOX2 EVE_MBOX0 EVE_MBOX1
Description	The message register stores the next to be read message of the mailbox. Reads remove the message from the FIFO queue.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MESSAGEVALUEMBM																															

Bits	Field Name	Description	Type	Reset
31:0	MESSAGEVALUEMBM	Message in Mailbox	RW	0x0000 0000

Table 14-28. Register Call Summary for Register MAILBOX_MESSAGE_m

Mailbox Functional Description • Description: [0][1] • Description: [2][3][4] • Description: [5][6][7][8]
Mailbox Programming Guide • Mailbox Processing modes: [9][10][11] • Events Servicing in Sending Mode: [12] • Events Servicing in Receiving Mode: [13]

Table 14-28. Register Call Summary for Register MAILBOX_MESSAGE_m (continued)

Mailbox Register Manual • Mailbox Register Summary: [14][15]

Table 14-29. MAILBOX_FIFOSTATUS_m

Address Offset	0x0000 0080 + (0x4 * m)	**Index**	m = 0 to 7 (MAILBOX1), or m = 0 to 11 (MAILBOX2), or m = 0 to 15 (EVE_MBOX)
Physical Address	0x4A0F 4080 + (0x4 * m) 0x4883 A080 + (0x4 * m) 0x4208 B080 + (0x4 * m) 0x4208 C080 + (0x4 * m)	**Instance**	MAILBOX1 MAILBOX2 EVE_MBOX0 EVE_MBOX1
Description	The FIFO status register has the status related to the mailbox internal FIFO		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															FIFOFULLMBM

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reads returns 0	R	0x0000 0000
0	FIFOFULLMBM	Full flag for Mailbox Read 0x0: Mailbox FIFO is not full Read 0x1: Mailbox FIFO is full	R	0

Table 14-30. Register Call Summary for Register MAILBOX_FIFOSTATUS_m

Mailbox Functional Description • Description: [0][1] • Description: [2]
Mailbox Programming Guide • Mailbox Processing modes: [3][4][5]
Mailbox Register Manual • Mailbox Register Summary: [6][7]

Table 14-31. MAILBOX_MSGSTATUS_m

Address Offset	0x0000 00C0 + (0x4 * m)	**Index**	m = 0 to 7 (MAILBOX1), or m = 0 to 11 (MAILBOX2), or m = 0 to 15 (EVE_MBOX)
Physical Address	0x4A0F 40C0 + (0x4 * m) 0x4883 A0C0 + (0x4 * m) 0x4208 B0C0 + (0x4 * m) 0x4208 C0C0 + (0x4 * m)	**Instance**	MAILBOX1 MAILBOX2 EVE_MBOX0 EVE_MBOX1
Description	The message status register has the status of the messages in the mailbox.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														NBOFMSGMBM	

Bits	Field Name	Description	Type	Reset
31:3	RESERVED	Reserved. Read returns 0	R	0x0000 0000
2:0	NBOFMSGMBM	Number of unread messages in Mailbox Note: Limited to four messages per mailbox.	R	0x00

Table 14-32. Register Call Summary for Register MAILBOX_MSGSTATUS_m

Mailbox Functional Description
- Description: [0][1]
- Description: [2]
- Description: [3]

Mailbox Programming Guide
- Mailbox Processing modes: [4]
- Events Servicing in Receiving Mode: [5]

Mailbox Register Manual
- Mailbox Register Summary: [6][7]

Table 14-33. MAILBOX_IRQSTATUS_RAW_u

Address Offset	0x0000 0100 + (0x10 * u)	**Index**	u = 0 to 2 (MAILBOX1), or u = 0 to 3 (MAILBOX2, EVE_MBOX)
Physical Address	0x4A0F 4100 + (0x10 * u) 0x4883 A100 + (0x10 * u) 0x4208 B100 + (0x10 * u) 0x4208 C100 + (0x10 * u)	**Instance**	MAILBOX1 MAILBOX2 EVE_MBOX0 EVE_MBOX1
Description	The interrupt status register has the raw status for each event that may be responsible for the generation of an interrupt to the corresponding user - write 1 to a given bit sets this bit. This register is mainly used for debug purpose.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NOTFULLSTATUSUUMB15	NEWMSGSTATUSUUMB15	NOTFULLSTATUSUUMB14	NEWMSGSTATUSUUMB14	NOTFULLSTATUSUUMB13	NEWMSGSTATUSUUMB13	NOTFULLSTATUSUUMB12	NEWMSGSTATUSUUMB12	NOTFULLSTATUSUUMB11	NEWMSGSTATUSUUMB11	NOTFULLSTATUSUUMB10	NEWMSGSTATUSUUMB10	NOTFULLSTATUSUUMB9	NEWMSGSTATUSUUMB9	NOTFULLSTATUSUUMB8	NEWMSGSTATUSUUMB8	NOTFULLSTATUSUUMB7	NEWMSGSTATUSUUMB7	NOTFULLSTATUSUUMB6	NEWMSGSTATUSUUMB6	NOTFULLSTATUSUUMB5	NEWMSGSTATUSUUMB5	NOTFULLSTATUSUUMB4	NEWMSGSTATUSUUMB4	NOTFULLSTATUSUUMB3	NEWMSGSTATUSUUMB3	NOTFULLSTATUSUUMB2	NEWMSGSTATUSUUMB2	NOTFULLSTATUSUUMB1	NEWMSGSTATUSUUMB1	NOTFULLSTATUSUUMB0	NEWMSGSTATUSUUMB0

Bits	Field Name	Description	Type	Reset
31	NOTFULLSTATUSUUMB15	NotFull Status bit for User u, Mailbox 15 Read 0x0: No event pending (message queue full) Write 0x0: No action Read 0x1: Event pending (message queue not full) Write 0x1: Set the event (for debug)	RW	1
30	NEWMSGSTATUSUUMB15	NewMessage Status bit for User u, Mailbox 15 Read 0x0: No event (message) pending Write 0x0: No action Read 0x1: Event (message) pending Write 0x1: Set the event (for debug)	RW	0
29	NOTFULLSTATUSUUMB14	NotFull Status bit for User u, Mailbox 14 Read 0x0: No event pending (message queue full) Write 0x0: No action Read 0x1: Event pending (message queue not full) Write 0x1: Set the event (for debug)	RW	1
28	NEWMSGSTATUSUUMB14	NewMessage Status bit for User u, Mailbox 14 Read 0x0: No event (message) pending Write 0x0: No action Read 0x1: Event (message) pending Write 0x1: Set the event (for debug)	RW	0
27	NOTFULLSTATUSUUMB13	NotFull Status bit for User u, Mailbox 13 Read 0x0: No event pending (message queue full) Write 0x0: No action Read 0x1: Event pending (message queue not full) Write 0x1: Set the event (for debug)	RW	1

Bits	Field Name	Description	Type	Reset
26	NEWMSGSTATUSUUMB13	NewMessage Status bit for User u, Mailbox 13 Read 0x0: No event (message) pending Write 0x0: No action Read 0x1: Event (message) pending Write 0x1: Set the event (for debug)	RW	0
25	NOTFULLSTATUSUUMB12	NotFull Status bit for User u, Mailbox 12 Read 0x0: No event pending (message queue full) Write 0x0: No action Read 0x1: Event pending (message queue not full) Write 0x1: Set the event (for debug)	RW	1
24	NEWMSGSTATUSUUMB12	NewMessage Status bit for User u, Mailbox 12 Read 0x0: No event (message) pending Write 0x0: No action Read 0x1: Event (message) pending Write 0x1: Set the event (for debug)	RW	0
23	NOTFULLSTATUSUUMB11	NotFull Status bit for User u, Mailbox 11 Read 0x0: No event pending (message queue full) Write 0x0: No action Read 0x1: Event pending (message queue not full) Write 0x1: Set the event (for debug)	RW	1
22	NEWMSGSTATUSUUMB11	NewMessage Status bit for User u, Mailbox 11 Read 0x0: No event (message) pending Write 0x0: No action Read 0x1: Event (message) pending Write 0x1: Set the event (for debug)	RW	0
21	NOTFULLSTATUSUUMB10	NotFull Status bit for User u, Mailbox 10 Read 0x0: No event pending (message queue full) Write 0x0: No action Read 0x1: Event pending (message queue not full) Write 0x1: Set the event (for debug)	RW	1
20	NEWMSGSTATUSUUMB10	NewMessage Status bit for User u, Mailbox 10 Read 0x0: No event (message) pending Write 0x0: No action Read 0x1: Event (message) pending Write 0x1: Set the event (for debug)	RW	0
19	NOTFULLSTATUSUUMB9	NotFull Status bit for User u, Mailbox 9 Read 0x0: No event pending (message queue full) Write 0x0: No action Read 0x1: Event pending (message queue not full) Write 0x1: Set the event (for debug)	RW	1
18	NEWMSGSTATUSUUMB9	NewMessage Status bit for User u, Mailbox 9 Read 0x0: No event (message) pending Write 0x0: No action Read 0x1: Event (message) pending Write 0x1: Set the event (for debug)	RW	0

Bits	Field Name	Description	Type	Reset
17	NOTFULLSTATUSUUMB8	NotFull Status bit for User u, Mailbox 8 Read 0x0: No event pending (message queue full) Write 0x0: No action Read 0x1: Event pending (message queue not full) Write 0x1: Set the event (for debug)	RW	1
16	NEWMSGSTATUSUUMB8	NewMessage Status bit for User u, Mailbox 8 Read 0x0: No event (message) pending Write 0x0: No action Read 0x1: Event (message) pending Write 0x1: Set the event (for debug)	RW	0
15	NOTFULLSTATUSUUMB7	NotFull Status bit for User u, Mailbox 7 Read 0x0: No event pending (message queue full) Write 0x0: No action Read 0x1: Event pending (message queue not full) Write 0x1: Set the event (for debug)	RW	1
14	NEWMSGSTATUSUUMB7	NewMessage Status bit for User u, Mailbox 7 Read 0x0: No event (message) pending Write 0x0: No action Read 0x1: Event (message) pending Write 0x1: Set the event (for debug)	RW	0
13	NOTFULLSTATUSUUMB6	NotFull Status bit for User u, Mailbox 6 Read 0x0: No event pending (message queue full) Write 0x0: No action Read 0x1: Event pending (message queue not full) Write 0x1: Set the event (for debug)	RW	1
12	NEWMSGSTATUSUUMB6	NewMessage Status bit for User u, Mailbox 6 Read 0x0: No event (message) pending Write 0x0: No action Read 0x1: Event (message) pending Write 0x1: Set the event (for debug)	RW	0
11	NOTFULLSTATUSUUMB5	NotFull Status bit for User u, Mailbox 5 Read 0x0: No event pending (message queue full) Write 0x0: No action Read 0x1: Event pending (message queue not full) Write 0x1: Set the event (for debug)	RW	1
10	NEWMSGSTATUSUUMB5	NewMessage Status bit for User u, Mailbox 5 Read 0x0: No event (message) pending Write 0x0: No action Read 0x1: Event (message) pending Write 0x1: Set the event (for debug)	RW	0
9	NOTFULLSTATUSUUMB4	NotFull Status bit for User u, Mailbox 4 Read 0x0: No event pending (message queue full) Write 0x0: No action Read 0x1: Event pending (message queue not full) Write 0x1: Set the event (for debug)	RW	1

Bits	Field Name	Description	Type	Reset
8	NEWMSGSTATUSUUMB4	NewMessage Status bit for User u, Mailbox 4 Read 0x0: No event (message) pending Write 0x0: No action Read 0x1: Event (message) pending Write 0x1: Set the event (for debug)	RW	0
7	NOTFULLSTATUSUUMB3	NotFull Status bit for User u, Mailbox 3 Read 0x0: No event pending (message queue full) Write 0x0: No action Read 0x1: Event pending (message queue not full) Write 0x1: Set the event (for debug)	RW	1
6	NEWMSGSTATUSUUMB3	NewMessage Status bit for User u, Mailbox 3 Read 0x0: No event (message) pending Write 0x0: No action Read 0x1: Event (message) pending Write 0x1: Set the event (for debug)	RW	0
5	NOTFULLSTATUSUUMB2	NotFull Status bit for User u, Mailbox 2 Read 0x0: No event pending (message queue full) Write 0x0: No action Read 0x1: Event pending (message queue not full) Write 0x1: Set the event (for debug)	RW	1
4	NEWMSGSTATUSUUMB2	NewMessage Status bit for User u, Mailbox 2 Read 0x0: No event (message) pending Write 0x0: No action Read 0x1: Event (message) pending Write 0x1: Set the event (for debug)	RW	0
3	NOTFULLSTATUSUUMB1	NotFull Status bit for User u, Mailbox 1 Read 0x0: No event pending (message queue full) Write 0x0: No action Read 0x1: Event pending (message queue not full) Write 0x1: Set the event (for debug)	RW	1
2	NEWMSGSTATUSUUMB1	NewMessage Status bit for User u, Mailbox 1 Read 0x0: No event (message) pending Write 0x0: No action Read 0x1: Event (message) pending Write 0x1: Set the event (for debug)	RW	0
1	NOTFULLSTATUSUUMB0	NotFull Status bit for User u, Mailbox 0 Read 0x0: No event pending (message queue full) Write 0x0: No action Read 0x1: Event pending (message queue not full) Write 0x1: Set the event (for debug)	RW	1
0	NEWMSGSTATUSUUMB0	NewMessage Status bit for User u, Mailbox 0 Read 0x0: No event (message) pending Write 0x0: No action Read 0x1: Event (message) pending Write 0x1: Set the event (for debug)	RW	0

Table 14-34. Register Call Summary for Register MAILBOX_IRQSTATUS_RAW_u

Mailbox Functional Description

- Mailbox Functional Description: [0]
- Mailbox Interrupt Requests: [1][2][3][4][5][6][7]

Mailbox Register Manual

- Mailbox Register Summary: [8][9]
- Mailbox Register Description: [10]

Table 14-35. MAILBOX_IRQSTATUS_CLR_u

Address Offset	0x0000 0104 + (0x10 * u)	**Index**	u = 0 to 2 (MAILBOX1), or u = 0 to 3 (MAILBOX2, EVE_MBOX)
Physical Address	0x4A0F 4104 + (0x10 * u) 0x4883 A104 + (0x10 * u) 0x4208 B104 + (0x10 * u) 0x4208 C104 + (0x10 * u)	**Instance**	MAILBOX1 MAILBOX2 EVE_MBOX0 EVE_MBOX1
Description	The interrupt status register has the status combined with irq-enable for each event that may be responsible for the generation of an interrupt to the corresponding user - write 1 to a given bit resets this bit		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NOTFULLSTATUSENUUMB15	NEWMSGSTATUSENUUMB15	NOTFULLSTATUSENUUMB14	NEWMSGSTATUSENUUMB14	NOTFULLSTATUSENUUMB13	NEWMSGSTATUSENUUMB13	NOTFULLSTATUSENUUMB12	NEWMSGSTATUSENUUMB12	NOTFULLSTATUSENUUMB11	NEWMSGSTATUSENUUMB11	NOTFULLSTATUSENUUMB10	NEWMSGSTATUSENUUMB10	NOTFULLSTATUSENUUMB9	NEWMSGSTATUSENUUMB9	NOTFULLSTATUSENUUMB8	NEWMSGSTATUSENUUMB8	NOTFULLSTATUSENUUMB7	NEWMSGSTATUSENUUMB7	NOTFULLSTATUSENUUMB6	NEWMSGSTATUSENUUMB6	NOTFULLSTATUSENUUMB5	NEWMSGSTATUSENUUMB5	NOTFULLSTATUSENUUMB4	NEWMSGSTATUSENUUMB4	NOTFULLSTATUSENUUMB3	NEWMSGSTATUSENUUMB3	NOTFULLSTATUSENUUMB2	NEWMSGSTATUSENUUMB2	NOTFULLSTATUSENUUMB1	NEWMSGSTATUSENUUMB1	NOTFULLSTATUSENUUMB0	NEWMSGSTATUSENUUMB0

Bits	Field Name	Description	Type	Reset
31	NOTFULLSTATUSENUUMB15	NotFull Status bit for User u, Mailbox 15 Read 0x0: No event pending (message queue full) Write 0x0: No action Read 0x1: Event pending (message queue not full) Write 0x1: Clear pending event, if any	RW	0
30	NEWMSGSTATUSENUUMB15	NewMessage Status bit for User u, Mailbox 15 Read 0x0: No event (message) pending Write 0x0: No action Read 0x1: Event (message) pending Write 0x1: Clear pending event, if any	RW	0
29	NOTFULLSTATUSENUUMB14	NotFull Status bit for User u, Mailbox 14 Read 0x0: No event pending (message queue full) Write 0x0: No action Read 0x1: Event pending (message queue not full) Write 0x1: Clear pending event, if any	RW	0

Bits	Field Name	Description	Type	Reset
28	NEWMSGSTATUSENUUMB14	NewMessage Status bit for User u, Mailbox 14 Read 0x0: No event (message) pending Write 0x0: No action Read 0x1: Event (message) pending Write 0x1: Clear pending event, if any	RW	0
27	NOTFULLSTATUSENUUMB13	NotFull Status bit for User u, Mailbox 13 Read 0x0: No event pending (message queue full) Write 0x0: No action Read 0x1: Event pending (message queue not full) Write 0x1: Clear pending event, if any	RW	0
26	NEWMSGSTATUSENUUMB13	NewMessage Status bit for User u, Mailbox 13 Read 0x0: No event (message) pending Write 0x0: No action Read 0x1: Event (message) pending Write 0x1: Clear pending event, if any	RW	0
25	NOTFULLSTATUSENUUMB12	NotFull Status bit for User u, Mailbox 12 Read 0x0: No event pending (message queue full) Write 0x0: No action Read 0x1: Event pending (message queue not full) Write 0x1: Clear pending event, if any	RW	0
24	NEWMSGSTATUSENUUMB12	NewMessage Status bit for User u, Mailbox 12 Read 0x0: No event (message) pending Write 0x0: No action Read 0x1: Event (message) pending Write 0x1: Clear pending event, if any	RW	0
23	NOTFULLSTATUSENUUMB11	NotFull Status bit for User u, Mailbox 11 Read 0x0: No event pending (message queue full) Write 0x0: No action Read 0x1: Event pending (message queue not full) Write 0x1: Clear pending event, if any	RW	0
22	NEWMSGSTATUSENUUMB11	NewMessage Status bit for User u, Mailbox 11 Read 0x0: No event (message) pending Write 0x0: No action Read 0x1: Event (message) pending Write 0x1: Clear pending event, if any	RW	0
21	NOTFULLSTATUSENUUMB10	NotFull Status bit for User u, Mailbox 10 Read 0x0: No event pending (message queue full) Write 0x0: No action Read 0x1: Event pending (message queue not full) Write 0x1: Clear pending event, if any	RW	0
20	NEWMSGSTATUSENUUMB10	NewMessage Status bit for User u, Mailbox 10 Read 0x0: No event (message) pending Write 0x0: No action Read 0x1: Event (message) pending Write 0x1: Clear pending event, if any	RW	0

Bits	Field Name	Description	Type	Reset
19	NOTFULLSTATUSENUUMB9	NotFull Status bit for User u, Mailbox 9 Read 0x0: No event pending (message queue full) Write 0x0: No action Read 0x1: Event pending (message queue not full) Write 0x1: Clear pending event, if any	RW	0
18	NEWMSGSTATUSENUUMB9	NewMessage Status bit for User u, Mailbox 9 Read 0x0: No event (message) pending Write 0x0: No action Read 0x1: Event (message) pending Write 0x1: Clear pending event, if any	RW	0
17	NOTFULLSTATUSENUUMB8	NotFull Status bit for User u, Mailbox 8 Read 0x0: No event pending (message queue full) Write 0x0: No action Read 0x1: Event pending (message queue not full) Write 0x1: Clear pending event, if any	RW	0
16	NEWMSGSTATUSENUUMB8	NewMessage Status bit for User u, Mailbox 8 Read 0x0: No event (message) pending Write 0x0: No action Read 0x1: Event (message) pending Write 0x1: Clear pending event, if any	RW	0
15	NOTFULLSTATUSENUUMB7	NotFull Status bit for User u, Mailbox 7 Read 0x0: No event pending (message queue full) Write 0x0: No action Read 0x1: Event pending (message queue not full) Write 0x1: Clear pending event, if any	RW	0
14	NEWMSGSTATUSENUUMB7	NewMessage Status bit for User u, Mailbox 7 Read 0x0: No event (message) pending Write 0x0: No action Read 0x1: Event (message) pending Write 0x1: Clear pending event, if any	RW	0
13	NOTFULLSTATUSENUUMB6	NotFull Status bit for User u, Mailbox 6 Read 0x0: No event pending (message queue full) Write 0x0: No action Read 0x1: Event pending (message queue not full) Write 0x1: Clear pending event, if any	RW	0
12	NEWMSGSTATUSENUUMB6	NewMessage Status bit for User u, Mailbox 6 Read 0x0: No event (message) pending Write 0x0: No action Read 0x1: Event (message) pending Write 0x1: Clear pending event, if any	RW	0
11	NOTFULLSTATUSENUUMB5	NotFull Status bit for User u, Mailbox 5 Read 0x0: No event pending (message queue full) Write 0x0: No action Read 0x1: Event pending (message queue not full) Write 0x1: Clear pending event, if any	RW	0

Bits	Field Name	Description	Type	Reset
10	NEWMSGSTATUSENUUMB5	NewMessage Status bit for User u, Mailbox 5 Read 0x0: No event (message) pending Write 0x0: No action Read 0x1: Event (message) pending Write 0x1: Clear pending event, if any	RW	0
9	NOTFULLSTATUSENUUMB4	NotFull Status bit for User u, Mailbox 4 Read 0x0: No event pending (message queue full) Write 0x0: No action Read 0x1: Event pending (message queue not full) Write 0x1: Clear pending event, if any	RW	0
8	NEWMSGSTATUSENUUMB4	NewMessage Status bit for User u, Mailbox 4 Read 0x0: No event (message) pending Write 0x0: No action Read 0x1: Event (message) pending Write 0x1: Clear pending event, if any	RW	0
7	NOTFULLSTATUSENUUMB3	NotFull Status bit for User u, Mailbox 3 Read 0x0: No event pending (message queue full) Write 0x0: No action Read 0x1: Event pending (message queue not full) Write 0x1: Clear pending event, if any	RW	0
6	NEWMSGSTATUSENUUMB3	NewMessage Status bit for User u, Mailbox 3 Read 0x0: No event (message) pending Write 0x0: No action Read 0x1: Event (message) pending Write 0x1: Clear pending event, if any	RW	0
5	NOTFULLSTATUSENUUMB2	NotFull Status bit for User u, Mailbox 2 Read 0x0: No event pending (message queue full) Write 0x0: No action Read 0x1: Event pending (message queue not full) Write 0x1: Clear pending event, if any	RW	0
4	NEWMSGSTATUSENUUMB2	NewMessage Status bit for User u, Mailbox 2 Read 0x0: No event (message) pending Write 0x0: No action Read 0x1: Event (message) pending Write 0x1: Clear pending event, if any	RW	0
3	NOTFULLSTATUSENUUMB1	NotFull Status bit for User u, Mailbox 1 Read 0x0: No event pending (message queue full) Write 0x0: No action Read 0x1: Event pending (message queue not full) Write 0x1: Clear pending event, if any	RW	0
2	NEWMSGSTATUSENUUMB1	NewMessage Status bit for User u, Mailbox 1 Read 0x0: No event (message) pending Write 0x0: No action Read 0x1: Event (message) pending Write 0x1: Clear pending event, if any	RW	0

Bits	Field Name	Description	Type	Reset
1	NOTFULLSTATUSENUUMB0	NotFull Status bit for User u, Mailbox 0 Read 0x0: No event pending (message queue full) Write 0x0: No action Read 0x1: Event pending (message queue not full) Write 0x1: Clear pending event, if any	RW	0
0	NEWMSGSTATUSENUUMB0	NewMessage Status bit for User u, Mailbox 0 Read 0x0: No event (message) pending Write 0x0: No action Read 0x1: Event (message) pending Write 0x1: Clear pending event, if any	RW	0

Table 14-36. Register Call Summary for Register MAILBOX_IRQSTATUS_CLR_u

Mailbox Functional Description
- Mailbox Functional Description: [0]
- Mailbox Interrupt Requests: [1][2][3][4][5]
- Description: [6][7]

Mailbox Programming Guide
- Events Servicing in Sending Mode: [8][9]
- Events Servicing in Receiving Mode: [10][11]

Mailbox Register Manual
- Mailbox Register Summary: [12][13]
- Mailbox Register Description: [14]

Table 14-37. MAILBOX_IRQENABLE_SET_u

Address Offset	0x0000 0108 + (0x10 * u)	**Index**	u = 0 to 2 (MAILBOX1), or u = 0 to 3 (MAILBOX2, EVE_MBOX)
Physical Address	0x4A0F 4108 + (0x10 * u) 0x4883 A108 + (0x10 * u) 0x4208 B108 + (0x10 * u) 0x4208 C108 + (0x10 * u)	**Instance**	MAILBOX1 MAILBOX2 EVE_MBOX0 EVE_MBOX1
Description	The interrupt enable register enables to unmask the module internal source of interrupt to the corresponding user. This register is write 1 to set.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NOTFULLENABLEUUMB15	NEWMSGENABLEUUMB15	NOTFULLENABLEUUMB14	NEWMSGENABLEUUMB14	NOTFULLENABLEUUMB13	NEWMSGENABLEUUMB13	NOTFULLENABLEUUMB12	NEWMSGENABLEUUMB12	NOTFULLENABLEUUMB11	NEWMSGENABLEUUMB11	NOTFULLENABLEUUMB10	NEWMSGENABLEUUMB10	NOTFULLENABLEUUMB9	NEWMSGENABLEUUMB9	NOTFULLENABLEUUMB8	NEWMSGENABLEUUMB8	NOTFULLENABLEUUMB7	NEWMSGENABLEUUMB7	NOTFULLENABLEUUMB6	NEWMSGENABLEUUMB6	NOTFULLENABLEUUMB5	NEWMSGENABLEUUMB5	NOTFULLENABLEUUMB4	NEWMSGENABLEUUMB4	NOTFULLENABLEUUMB3	NEWMSGENABLEUUMB3	NOTFULLENABLEUUMB2	NEWMSGENABLEUUMB2	NOTFULLENABLEUUMB1	NEWMSGENABLEUUMB1	NOTFULLENABLEUUMB0	NEWMSGENABLEUUMB0

Bits	Field Name	Description	Type	Reset
31	NOTFULLENABLEUUMB15	NotFull Enable bit for User u, Mailbox 15 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Enable interrupt	RW	0

Bits	Field Name	Description	Type	Reset
30	NEWMSGENABLEUUMB15	NewMessage Enable bit for User u, Mailbox 15 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Enable interrupt	RW	0
29	NOTFULLENABLEUUMB14	NotFull Enable bit for User u, Mailbox 14 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Enable interrupt	RW	0
28	NEWMSGENABLEUUMB14	NewMessage Enable bit for User u, Mailbox 14 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Enable interrupt	RW	0
27	NOTFULLENABLEUUMB13	NotFull Enable bit for User u, Mailbox 13 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Enable interrupt	RW	0
26	NEWMSGENABLEUUMB13	NewMessage Enable bit for User u, Mailbox 13 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Enable interrupt	RW	0
25	NOTFULLENABLEUUMB12	NotFull Enable bit for User u, Mailbox 12 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Enable interrupt	RW	0
24	NEWMSGENABLEUUMB12	NewMessage Enable bit for User u, Mailbox 12 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Enable interrupt	RW	0
23	NOTFULLENABLEUUMB11	NotFull Enable bit for User u, Mailbox 11 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Enable interrupt	RW	0
22	NEWMSGENABLEUUMB11	NewMessage Enable bit for User u, Mailbox 11 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Enable interrupt	RW	0

Bits	Field Name	Description	Type	Reset
21	NOTFULLENABLEUUMB10	NotFull Enable bit for User u, Mailbox 10 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Enable interrupt	RW	0
20	NEWMSGENABLEUUMB10	NewMessage Enable bit for User u, Mailbox 10 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Enable interrupt	RW	0
19	NOTFULLENABLEUUMB9	NotFull Enable bit for User u, Mailbox 9 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Enable interrupt	RW	0
18	NEWMSGENABLEUUMB9	NewMessage Enable bit for User u, Mailbox 9 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Enable interrupt	RW	0
17	NOTFULLENABLEUUMB8	NotFull Enable bit for User u, Mailbox 8 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Enable interrupt	RW	0
16	NEWMSGENABLEUUMB8	NewMessage Enable bit for User u, Mailbox 8 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Enable interrupt	RW	0
15	NOTFULLENABLEUUMB7	NotFull Enable bit for User u, Mailbox 7 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Enable interrupt	RW	0
14	NEWMSGENABLEUUMB7	NewMessage Enable bit for User u, Mailbox 7 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Enable interrupt	RW	0
13	NOTFULLENABLEUUMB6	NotFull Enable bit for User u, Mailbox 6 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Enable interrupt	RW	0

Bits	Field Name	Description	Type	Reset
12	NEWMSGENABLEUUMB6	NewMessage Enable bit for User u, Mailbox 6 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Enable interrupt	RW	0
11	NOTFULLENABLEUUMB5	NotFull Enable bit for User u, Mailbox 5 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Enable interrupt	RW	0
10	NEWMSGENABLEUUMB5	NewMessage Enable bit for User u, Mailbox 5 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Enable interrupt	RW	0
9	NOTFULLENABLEUUMB4	NotFull Enable bit for User u, Mailbox 4 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Enable interrupt	RW	0
8	NEWMSGENABLEUUMB4	NewMessage Enable bit for User u, Mailbox 4 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Enable interrupt	RW	0
7	NOTFULLENABLEUUMB3	NotFull Enable bit for User u, Mailbox 3 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Enable interrupt	RW	0
6	NEWMSGENABLEUUMB3	NewMessage Enable bit for User u, Mailbox 3 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Enable interrupt	RW	0
5	NOTFULLENABLEUUMB2	NotFull Enable bit for User u, Mailbox 2 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Enable interrupt	RW	0
4	NEWMSGENABLEUUMB2	NewMessage Enable bit for User u, Mailbox 2 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Enable interrupt	RW	0

Bits	Field Name	Description	Type	Reset
3	NOTFULLENABLEUUMB1	NotFull Enable bit for User u, Mailbox 1 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Enable interrupt	RW	0
2	NEWMSGENABLEUUMB1	NewMessage Enable bit for User u, Mailbox 1 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Enable interrupt	RW	0
1	NOTFULLENABLEUUMB0	NotFull Enable bit for User u, Mailbox 0 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Enable interrupt	RW	0
0	NEWMSGENABLEUUMB0	NewMessage Enable bit for User u, Mailbox 0 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Enable interrupt	RW	0

Table 14-38. Register Call Summary for Register MAILBOX_IRQENABLE_SET_u

Mailbox Functional Description • Mailbox Functional Description: [0] • Mailbox Interrupt Requests: [1][2][3] • Description: [4][5][6]
Mailbox Programming Guide • Mailbox Processing modes: [7][8]
Mailbox Register Manual • Mailbox Register Summary: [9][10] • Mailbox Register Description: [11]

Table 14-39. MAILBOX_IRQENABLE_CLR_u

Address Offset	0x0000 010C + (0x10 * u)	**Index**	u = 0 to 2 (MAILBOX1), or u = 0 to 3 (MAILBOX2, EVE_MBOX)
Physical Address	0x4A0F 410C + (0x10 * u) 0x4883 A10C + (0x10 * u) 0x4208 B10C + (0x10 * u) 0x4208 C10C + (0x10 * u)	**Instance**	MAILBOX1 MAILBOX2 EVE_MBOX0 EVE_MBOX1
Description	The interrupt enable register enables to mask the module internal source of interrupt to the corresponding user. This register is write 1 to clear.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NOTFULLENABLEUUMB15	NEWMSGENABLEUUMB15	NOTFULLENABLEUUMB14	NEWMSGENABLEUUMB14	NOTFULLENABLEUUMB13	NEWMSGENABLEUUMB13	NOTFULLENABLEUUMB12	NEWMSGENABLEUUMB12	NOTFULLENABLEUUMB11	NEWMSGENABLEUUMB11	NOTFULLENABLEUUMB10	NEWMSGENABLEUUMB10	NOTFULLENABLEUUMB9	NEWMSGENABLEUUMB9	NOTFULLENABLEUUMB8	NEWMSGENABLEUUMB8	NOTFULLENABLEUUMB7	NEWMSGENABLEUUMB7	NOTFULLENABLEUUMB6	NEWMSGENABLEUUMB6	NOTFULLENABLEUUMB5	NEWMSGENABLEUUMB5	NOTFULLENABLEUUMB4	NEWMSGENABLEUUMB4	NOTFULLENABLEUUMB3	NEWMSGENABLEUUMB3	NOTFULLENABLEUUMB2	NEWMSGENABLEUUMB2	NOTFULLENABLEUUMB1	NEWMSGENABLEUUMB1	NOTFULLENABLEUUMB0	NEWMSGENABLEUUMB0

Bits	Field Name	Description	Type	Reset
31	NOTFULLENABLEUUMB15	NotFull Enable bit for User u, Mailbox 15 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Disable interrupt	RW	0
30	NEWMSGENABLEUUMB15	NewMessage Enable bit for User u, Mailbox 15 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Disable interrupt	RW	0
29	NOTFULLENABLEUUMB14	NotFull Enable bit for User u, Mailbox 14 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Disable interrupt	RW	0
28	NEWMSGENABLEUUMB14	NewMessage Enable bit for User u, Mailbox 14 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Disable interrupt	RW	0
27	NOTFULLENABLEUUMB13	NotFull Enable bit for User u, Mailbox 13 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Disable interrupt	RW	0

Bits	Field Name	Description	Type	Reset
26	NEWMSGENABLEUUMB13	NewMessage Enable bit for User u, Mailbox 13 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Disable interrupt	RW	0
25	NOTFULLENABLEUUMB12	NotFull Enable bit for User u, Mailbox 12 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Disable interrupt	RW	0
24	NEWMSGENABLEUUMB12	NewMessage Enable bit for User u, Mailbox 12 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Disable interrupt	RW	0
23	NOTFULLENABLEUUMB11	NotFull Enable bit for User u, Mailbox 11 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Disable interrupt	RW	0
22	NEWMSGENABLEUUMB11	NewMessage Enable bit for User u, Mailbox 11 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Disable interrupt	RW	0
21	NOTFULLENABLEUUMB10	NotFull Enable bit for User u, Mailbox 10 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Disable interrupt	RW	0
20	NEWMSGENABLEUUMB10	NewMessage Enable bit for User u, Mailbox 10 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Disable interrupt	RW	0
19	NOTFULLENABLEUUMB9	NotFull Enable bit for User u, Mailbox 9 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Disable interrupt	RW	0
18	NEWMSGENABLEUUMB9	NewMessage Enable bit for User u, Mailbox 9 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Disable interrupt	RW	0

Bits	Field Name	Description	Type	Reset
17	NOTFULLENABLEUUMB8	NotFull Enable bit for User u, Mailbox 8 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Disable interrupt	RW	0
16	NEWMSGENABLEUUMB8	NewMessage Enable bit for User u, Mailbox 8 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Disable interrupt	RW	0
15	NOTFULLENABLEUUMB7	NotFull Enable bit for User u, Mailbox 7 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Disable interrupt	RW	0
14	NEWMSGENABLEUUMB7	NewMessage Enable bit for User u, Mailbox 7 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Disable interrupt	RW	0
13	NOTFULLENABLEUUMB6	NotFull Enable bit for User u, Mailbox 6 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Disable interrupt	RW	0
12	NEWMSGENABLEUUMB6	NewMessage Enable bit for User u, Mailbox 6 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Disable interrupt	RW	0
11	NOTFULLENABLEUUMB5	NotFull Enable bit for User u, Mailbox 5 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Disable interrupt	RW	0
10	NEWMSGENABLEUUMB5	NewMessage Enable bit for User u, Mailbox 5 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Disable interrupt	RW	0
9	NOTFULLENABLEUUMB4	NotFull Enable bit for User u, Mailbox 4 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Disable interrupt	RW	0

Bits	Field Name	Description	Type	Reset
8	NEWMSGENABLEUUMB4	NewMessage Enable bit for User u, Mailbox 4 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Disable interrupt	RW	0
7	NOTFULLENABLEUUMB3	NotFull Enable bit for User u, Mailbox 3 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Disable interrupt	RW	0
6	NEWMSGENABLEUUMB3	NewMessage Enable bit for User u, Mailbox 3 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Disable interrupt	RW	0
5	NOTFULLENABLEUUMB2	NotFull Enable bit for User u, Mailbox 2 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Disable interrupt	RW	0
4	NEWMSGENABLEUUMB2	NewMessage Enable bit for User u, Mailbox 2 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Disable interrupt	RW	0
3	NOTFULLENABLEUUMB1	NotFull Enable bit for User u, Mailbox 1 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Disable interrupt	RW	0
2	NEWMSGENABLEUUMB1	NewMessage Enable bit for User u, Mailbox 1 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Disable interrupt	RW	0
1	NOTFULLENABLEUUMB0	NotFull Enable bit for User u, Mailbox 0 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Disable interrupt	RW	0
0	NEWMSGENABLEUUMB0	NewMessage Enable bit for User u, Mailbox 0 Read 0x0: Interrupt disabled Write 0x0: No action Read 0x1: Interrupt enabled Write 0x1: Disable interrupt	RW	0

Table 14-40. Register Call Summary for Register MAILBOX_IRQENABLE_CLR_u

Mailbox Functional Description • Mailbox Functional Description: [0] • Mailbox Interrupt Requests: [1][2][3]
Mailbox Register Manual • Mailbox Register Summary: [4][5] • Mailbox Register Description: [6]

NOTE: For each interrupt status and enable register (MAILBOX_IRQSTATUS_RAW_u, MAILBOX_IRQSTATUS_CLR_u, MAILBOX_IRQENABLE_SET_u and MAILBOX_IRQENABLE_CLR_u):

- Bits [31:0] have the given meaning for the EVE_MBOX0 and EVE_MBOX1 instances.
- Bits [31:24] are RESERVED for the MAILBOX2 instance.
- Bits [31:16] are RESERVED for the MAILBOX1 instance.

Table 14-41. MAILBOX_IRQ_EOI

Address Offset	0x0000 0140		
Physical Address	0x4A0F 4140 0x4883 A140 0x4208 B140 0x4208 C140	**Instance**	MAILBOX1 MAILBOX2 EVE_MBOX0 EVE_MBOX1
Description	This register is used for the software EOI clearance of the pulse. This register being write only gives 0 on read.		
Type	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														EOIVAL	

Bits	Field Name	Description	Type	Reset
31:2	RESERVED	Write 0's for future compatibility. Reads returns 0	W	0x0
1:0	EOIVAL	EOI value 0x0: EOI val first bit 0x1: EOI val second bit	W	0x0

Table 14-42. Register Call Summary for Register MAILBOX_IRQ_EOI

Mailbox Register Manual • Mailbox Register Summary: [0][1]

Memory Management Units

This chapter describes the memory management units (MMUs) in the device.

Topic **Page**

15.1 MMU Overview

A memory management unit (MMU) is a hardware component responsible for handling accesses to memory requested by a processing unit, DMA controller, or other bus requestor. MMU functions include:

- Translation of initiator internal (virtual) addresses to physical addresses (that is, virtual memory management)
- Preventing an initiator from making accesses to unmapped pages of the system memory

This device includes the following MMUs:

- One top-level (system) MMU, dedicated to EDMA Transfer Controller 0 (TC0), and EDMA Transfer Controller 1 (TC1):
- Two MMUs inside each of the DSP1, and DSP2 subsytems:
  - DSP1 subsystem: DSP1_MMU0, and DSP1_MMU1
  - DSP2 subsystem: DSP2_MMU0, and DSP2_MMU1
- Two MMUs inside the EVE subsystem: EVE_MMU0, and EVE_MMU1
- Two MMUs inside the IPU subsystem:
  - L1 unicache MMU – IPU_UNICACHE_MMU
  - L2 MMU – IPU_MMU

NOTE: This chapter provides a detailed description of the following MMUs:

- System MMU
- DSP MMUs
- EVE MMUs
- IPU L2 MMU

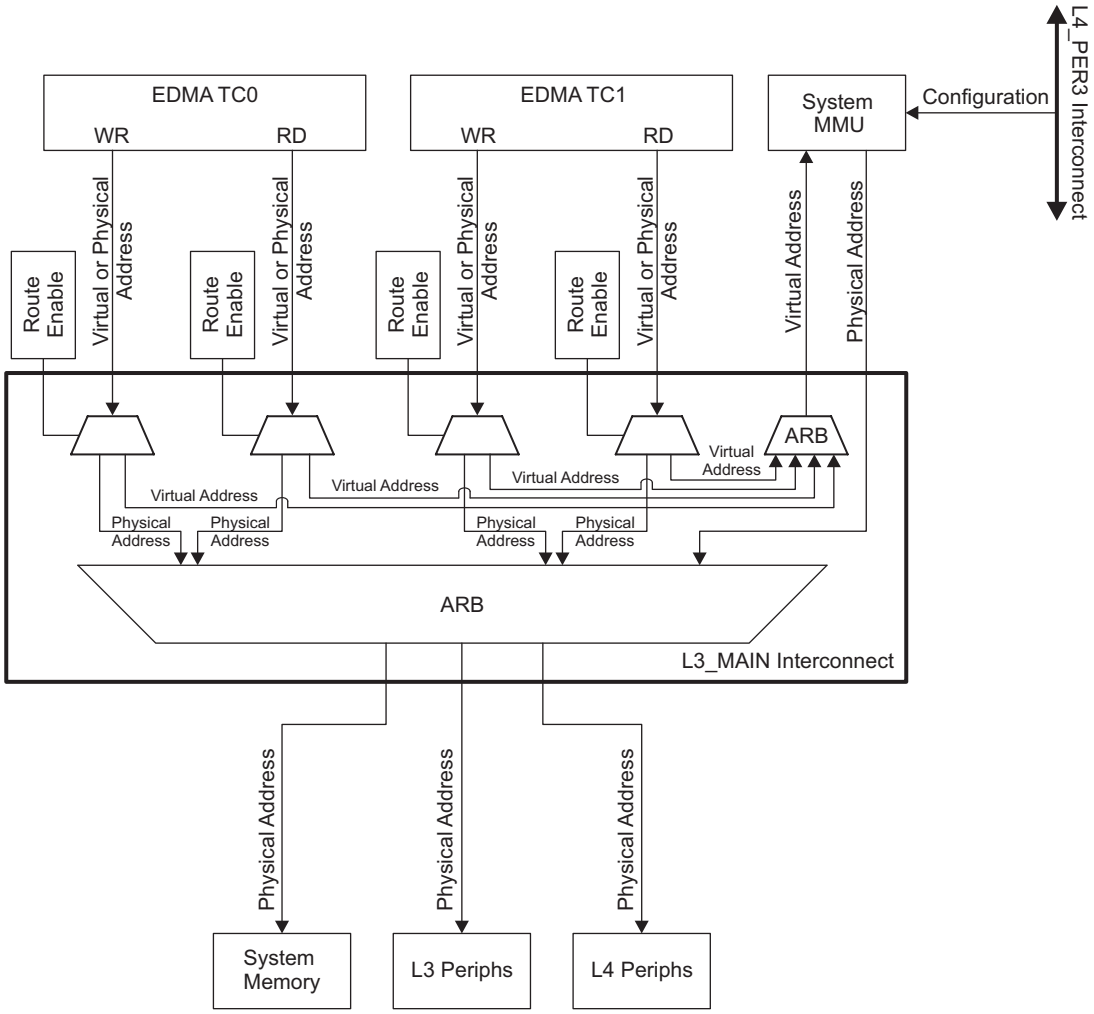
System MMUs, DSP MMUs and EVE MMUs are fully identical from functional perspective. IPU L2 MMU is different in that it does not support “bypass” functionality.

For more information about IPU unicache MMU, see Chapter 5, *Dual Cortex-M4 IPU Subsystem*.

Figure 15-1 shows an overview of system MMU. In summary, requests initiated by a given requestor (EDMA TC0 or TC1 [both read and write ports]) can optionally be routed through the system MMU. Each requestor’s use (or not) of the MMU is independently controllable via the Control Module CTRL_CORE_SMA_SW_7 register bitfields. It is recommended that this register is set during system initialization and remain static.

If the MMU loopback path is enabled for a given requestor, requests will be routed via the L3_MAIN interconnect to the MMU and will again go through the L3_MAIN interconnect to the requested physical address. If the MMU loopback path is disabled for a given requestor, those bus requests go directly through the L3_MAIN interconnect to the requested physical address, thus minimizing bus request latency.

Figure 15-1. System MMU1 Overview



15.2 MMU Integration

This section describes the system MMUs integration in the device, including information about clocks, resets, and hardware requests. For more information about DSP, EVE, and IPU MMUs integration, refer to their respective chapters.

Figure 15-2 shows system MMU integration.

Figure 15-2. System MMU1 Integration

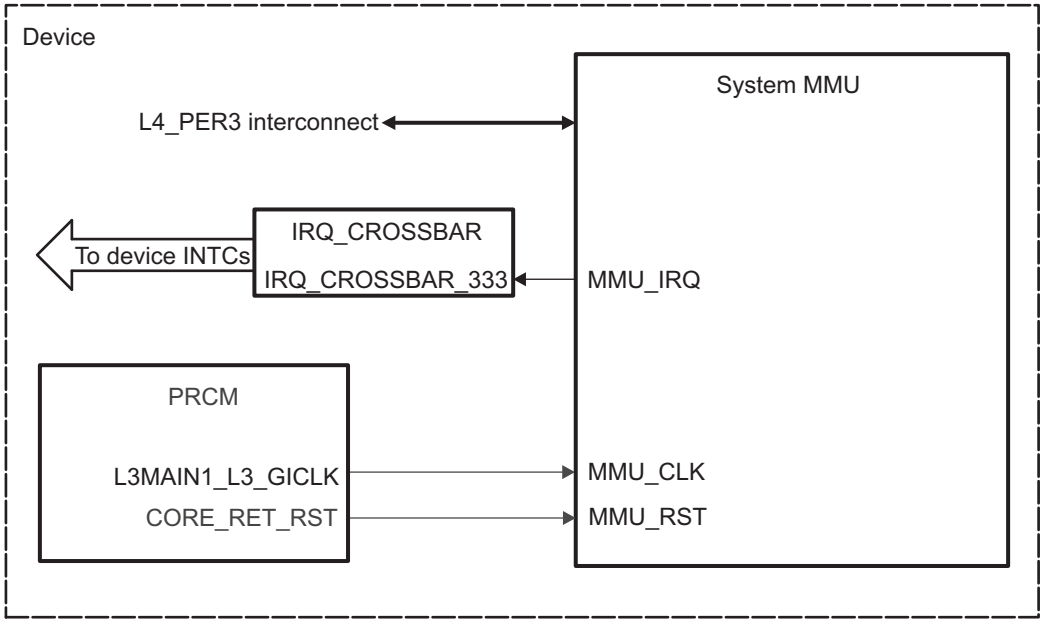


Table 15-1 through Table 15-3 summarize the system MMU integration.

Table 15-1. MMU Integration Attributes

Module Instance	Attributes	
	Power Domain	Interconnect
System MMU	PD_COREAON	L4_PER3

Table 15-2. MMU Clocks and Resets

Clocks				
Module Instance	**Destination Signal Name**	**Source Signal Name**	**Source**	**Description**
System MMU	MMU_CLK	L3MAIN1_L3_GICLK	PRCM	System MMU interface/functional clock. This clock is used for all interface and functional operations.
Resets				
Module Instance	**Destination Signal Name**	**Source Signal Name**	**Source**	**Description**
System MMU	MMU_RST	CORE_RET_RST	PRCM	System MMU hardware reset. This reset is asynchronously applied to the MMU internal registers.

Table 15-3. MMU Hardware Requests

Interrupt Requests				
Module Instance	**Interrupt Name (Source)**	**IRQ_CROSSBAR Input (Destination)**	**Default Mapping**	**Description**
System MMU	MMU_IRQ	IRQ_CROSSBAR_333	–	System MMU interrupt.
No DMA Requests				

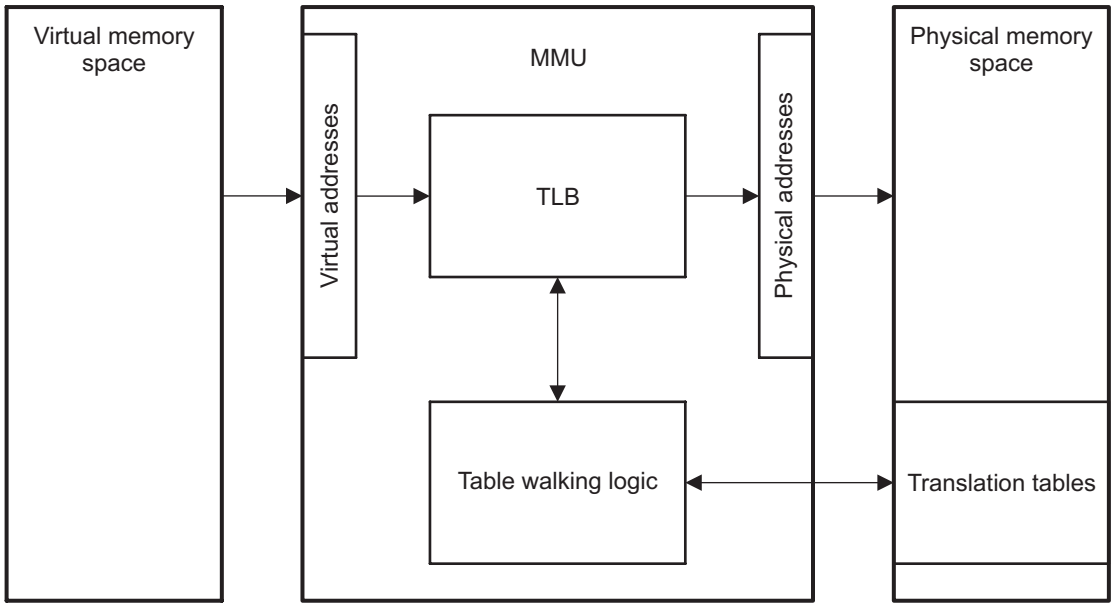
NOTE: For a description of the interrupt sources, see Section 15.3.4, *MMU Interrupt Requests*.

15.3 MMU Functional Description

15.3.1 MMU Block Diagram

The MMU manages the virtual to physical address translation for external addresses. Figure 15-3 shows the MMU block diagram.

Figure 15-3. MMU Block Diagram



Each table entry describes the translation of one contiguous memory region. For a description of the structure of these tables, see Section 15.3.1.2, *Translation Tables*.

Two major functional units exist in the MMU to provide address translation automatically based on the table entries:

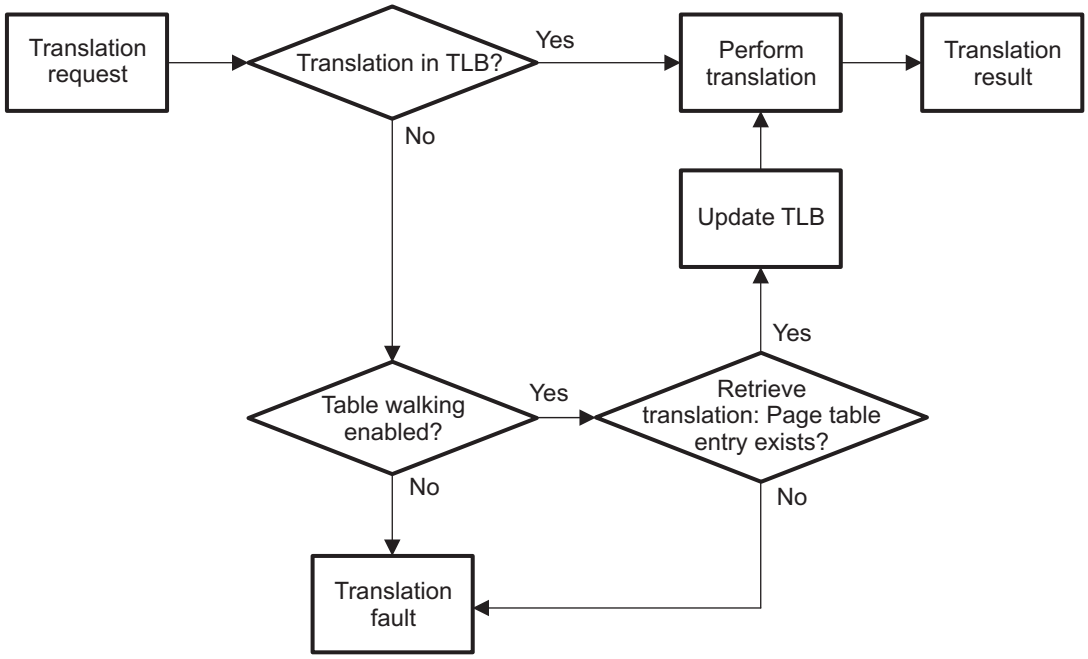
- The table walker automatically retrieves the correct translation table entry for a requested translation. If two-level translation is used (for the translation of small memory pages), the table walker also automatically reads the required second-level translation table entry. The two-level translation is described later in the chapter.
- The translation look-aside buffer (TLB) stores recently used translation entries, acting like a cache of the translation table.

15.3.1.1 MMU Address Translation Process

Whenever an address translation is requested (that is, for every access with the MMU enabled), the MMU first checks whether the translation is contained in the TLB, which acts like a cache storing recent translations. The TLB can also be programmed manually to ensure that time-critical data can be translated without delay.

If the requested translation is not in the TLB, the table-walking logic retrieves this translation from the translation table(s), and then updates the TLB. The address translation is then performed. Figure 15-4 summarizes the process.

Figure 15-4. Translation Process



15.3.1.2 Translation Tables

The translation of virtual to physical addresses is based on entries in translation tables that define the following properties:

- Address translation, that is, the correspondence between virtual and physical addresses
- Size of the memory region the entry translates

The virtual addresses index the translation tables. Each virtual address corresponds to exactly one entry in the translation table.

15.3.1.2.1 Translation Table Hierarchy

When developing a table-based address translation scheme, one of the most important design parameters is the memory page size described by each translation table entry. MMU instances support 4-KiB and 64-KiB pages, a 1-MiB section, and a 16-MiB supersection. Using bigger page sizes means a smaller translation table.

Using a smaller page size greatly increases the efficiency of dynamic memory allocation and defragmentation. That is why many operating systems (OSs) can operate on memory blocks as small as 4 KiB; however, the smaller size implies a more complex table structure.

A quick calculation shows that using 4 KiB memory pages with one translation table would require one million entries to span the entire 4-GiB address range. The table itself would be 32 MiB, a size that is not feasible.

However, using bigger pages reduces the flexibility of typical OS memory management. Implementing a two-level hierarchy reconciles these two requirements. Within this hierarchy, one first-level translation table describes the translation properties based on 1 MiB memory regions.

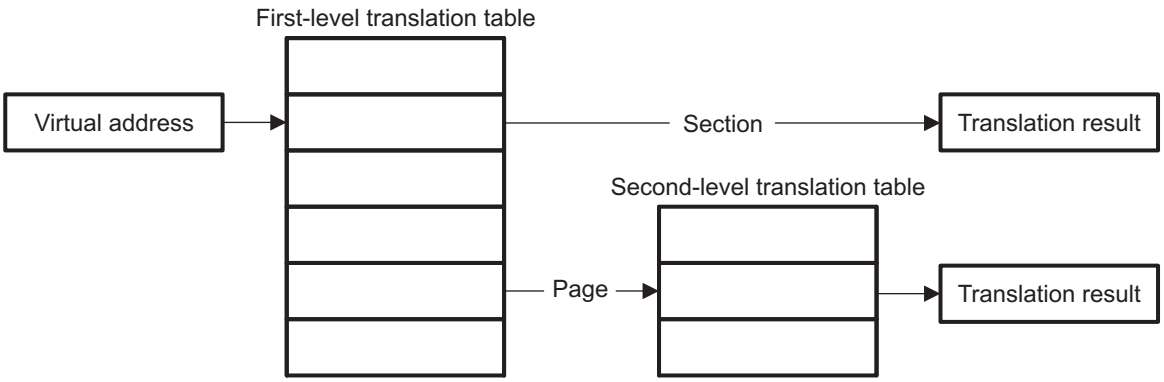
Each of the entries in this first-level translation table can specify the following:

- The translation properties for a big memory section. This memory section can be either 1 MiB (section) or 16 MiB (supersection). In this case, all translation parameters are specified in the first-level translation table entry.
- A pointer to a second-level translation table that specifies individual translation properties based on smaller pages within the 1-MiB page of memory. These pages can be either 64 KiB (large page) or 4

KiB (small page). In this case, the actual translation parameters are specified in the second-level translation table entry. The first-level translation table entry specifies only the base address of the second-level translation table.

This hierarchical approach means that additional translation information for smaller pages must be provided only when the pages are actually used. Figure 15-5 shows the hierarchy.

Figure 15-5. Translation Hierarchy



The structure of the first and second-level translation tables and their entries are described in more detail in Section 15.3.1.2.2, *First-Level Translation Table*, and Section 15.3.1.2.3, *Two-Level Translation*.

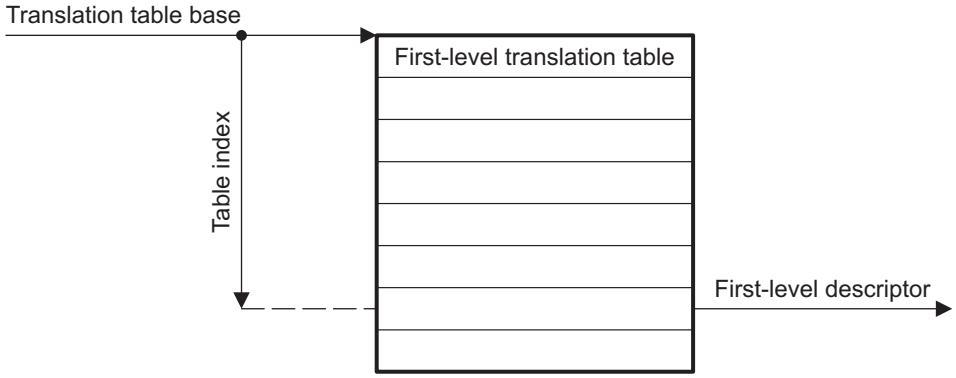
15.3.1.2.2 First-Level Translation Table

The first-level translation table describes the translation properties for 1-MiB sections. To describe a 4-GiB address range requires 4096 32-bit entries (so-called first-level descriptors).

The first-level translation table start address must be aligned on a multiple of the table size with a 128-byte minimum. Consequently, an alignment of at least 16K bytes is required for a complete 4096-entry table; that is, at least the last fourteen address bits must be zero.

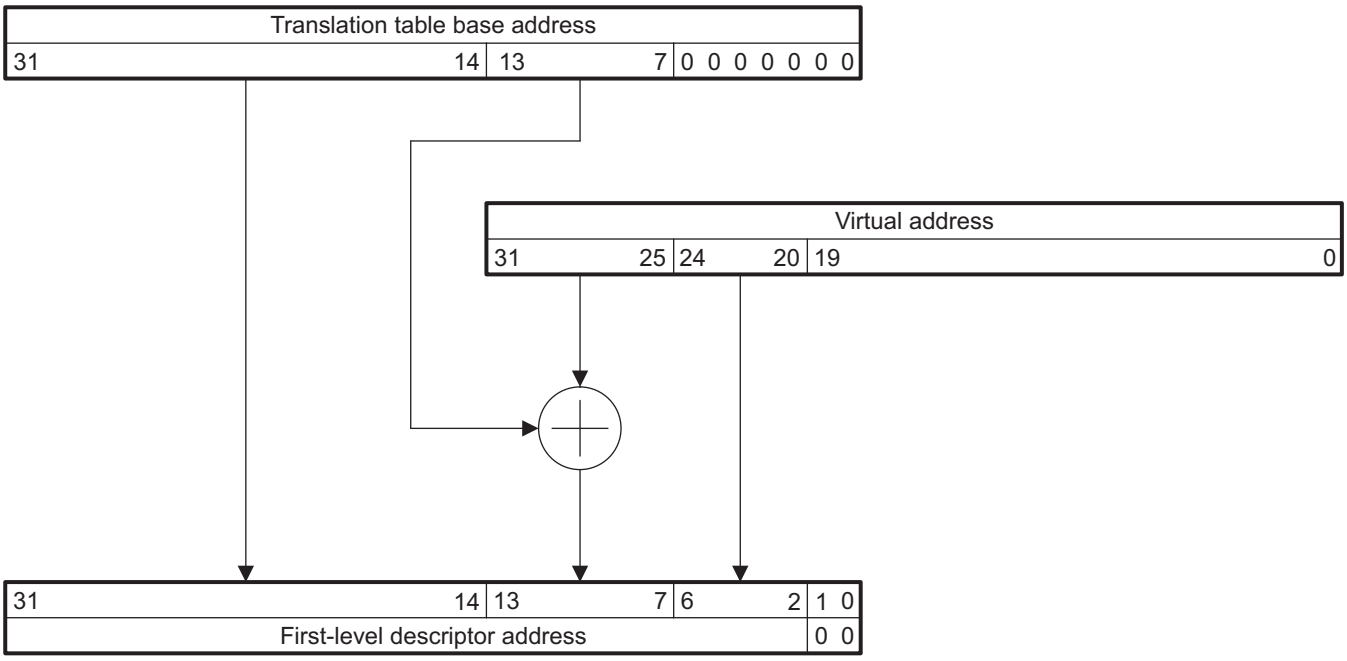
The start address of the first-level translation table is specified by the so-called translation table base. The table is indexed by the upper 12-bits of the virtual address. Figure 15-6 shows this mechanism.

Figure 15-6. First-level Descriptor Address Calculation



To summarize, the translation table base and the translation table index together define the first-level descriptor address. Figure 15-7 outlines the precise mechanism used to calculate this address.

Figure 15-7. Detailed First-Level Descriptor Address Calculation



As an example of this mechanism, consider a translation table base address of 0x8000:0000 and a virtual address of 0x1234:5678. In this case, the first-level descriptor address is 0x8000:0000 + (0x123 << 2) = 0x8000:048C.

15.3.1.2.2.1 First-Level Descriptor Format

Each first-level descriptor provides either the complete address translation for 1-MiB or 16-MiB sections or provides a pointer to a second-level translation table for 4 KiB or 64 KiB pages. Table 15-4 shows the first-level descriptor format.

Table 15-4. First-Level Descriptor Format

First-Level Descriptor Format								
31:24	23:20	19	18	17:10	9:2	1	0	
X						0	0	Fault
Second-Level Translation Table Base Address					X	0	1	Page
Section Base Address		X	0	X		1	0	Section
Supersection Base Address	X		1	X		1	0	Supersection
X						1	1	Fault

X = Don't care. Set to 0 for future compatibility.

15.3.1.2.2.2 First-Level Page Descriptor Format

If a translation granularity smaller than 1 MiB is required, a two-level translation process is used. In this case, the first-level block descriptor specifies only the start address of a second-level translation table. The second-level translation table entries specify the actual translation properties.

15.3.1.2.2.3 First-Level Section Descriptor Format

Each section descriptor in the first-level translation table specifies the complete translation properties for a 1-MiB section or a 16-MiB supersection.

NOTE: Supersection descriptors must be repeated 16 times, because each descriptor in the first-level translation table describes 1 MiB of memory. If an access points to a descriptor that is not initialized, the MMU will behave in an unpredictable way.

15.3.1.2.2.4 Section Translation Summary

Sections and supersections can be translated based solely on the information in the first-level translation table. Figure 15-8 summarizes the address translation process for a section.

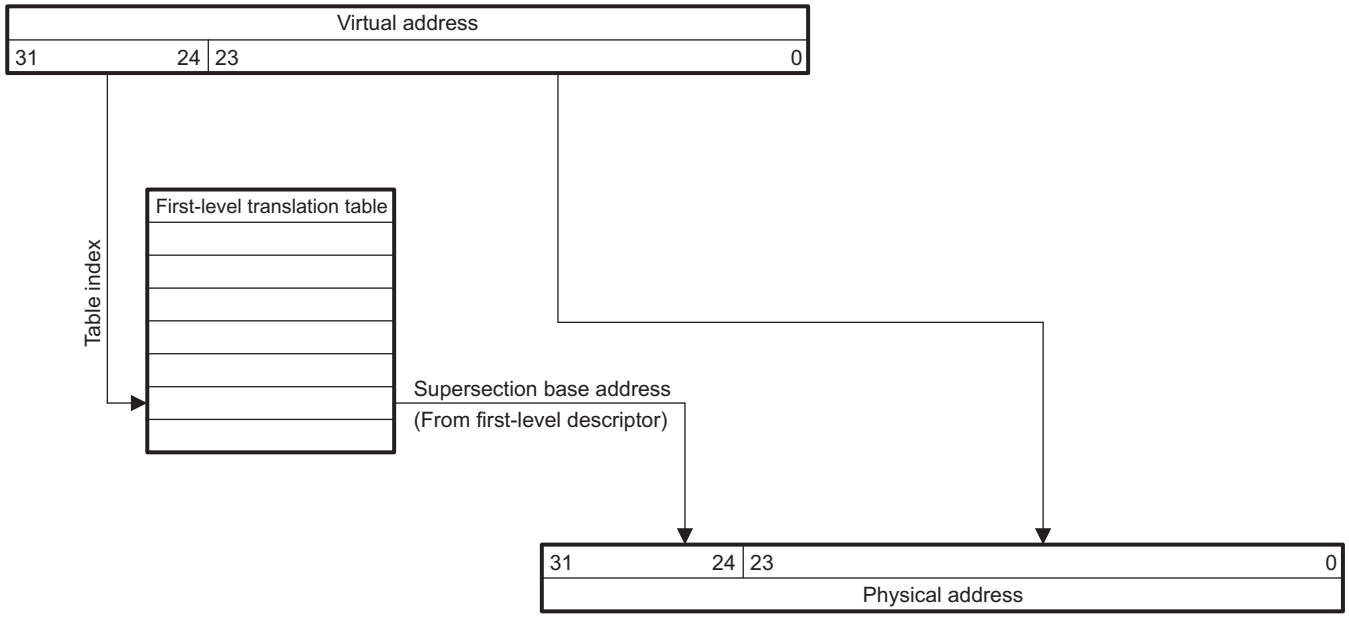
Figure 15-8. Section Translation Summary

15.3.1.2.2.5 Supersection Translation Summary

The translation of a supersection is similar to the translation of a section. The difference is that for a supersection only bits 31 to 24 index into the first-level translation table. The last four bits of the table index are implicitly assumed to be zero as there are 16 identical consecutive entries for a supersection.

Figure 15-9 shows the translation mechanism for a supersection.

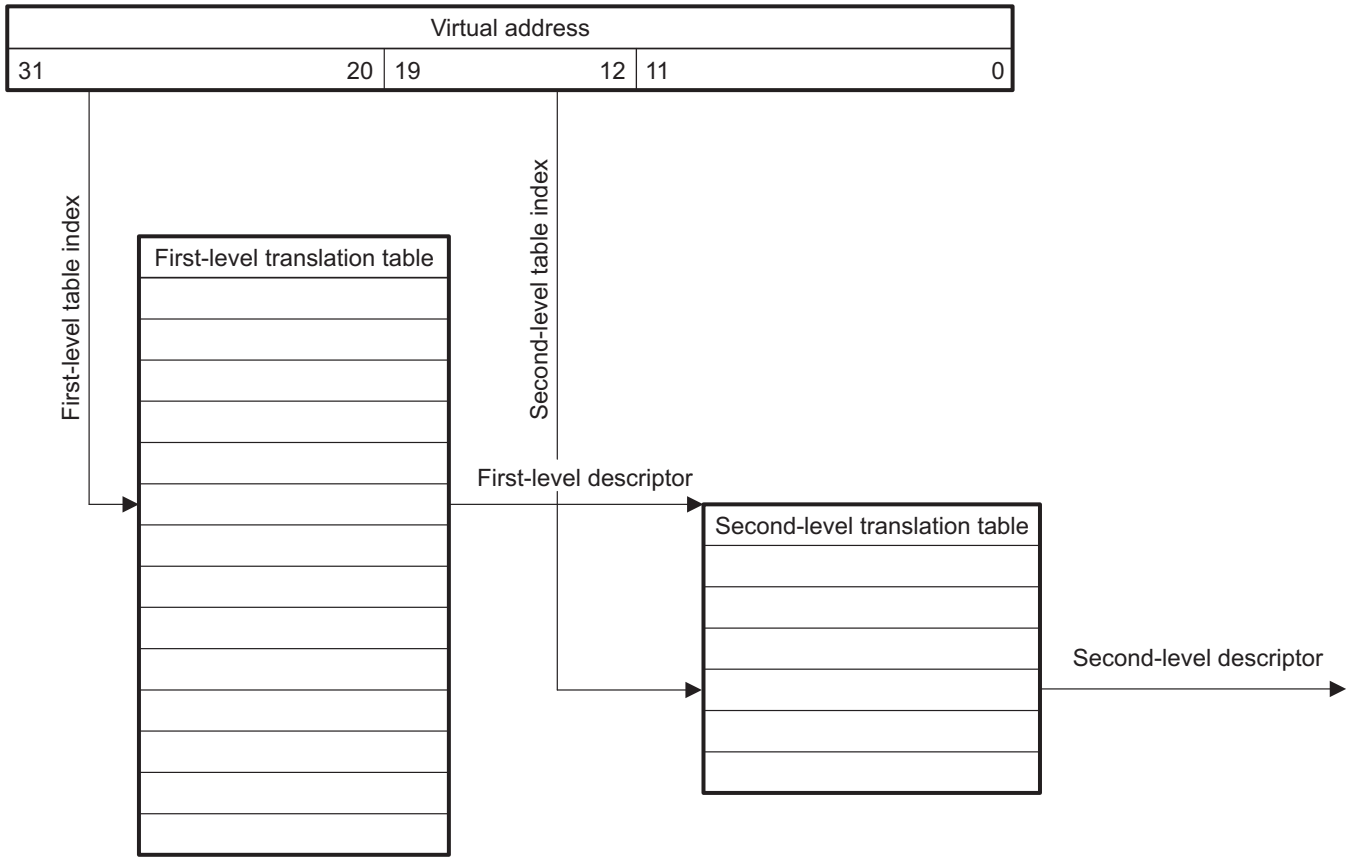
Figure 15-9. Supersection Translation Summary



15.3.1.2.3 Two-Level Translation

Two-level translation is used when fine-grain granularity is required, that is, when memory sections smaller than 1 MiB are needed. In this case, the first-level descriptor provides a pointer to the base address of a second-level translation table. This second-level table is indexed by bits 19 to 12 of the virtual address. Figure 15-10 shows this indexing mechanism.

Figure 15-10. Two-Level Translation



Each second-level translation table describes the translation of 1 MiB of address space in pages of 64 KiB (large page) or 4 KiB (small page). It consists of 256 second-level descriptors describing 4 KiB each.

NOTE: In the case of a large page, the same descriptor must be repeated 16 times. If an access points to a descriptor that is not initialized, the MMU will behave in an unpredictable way.

15.3.1.2.3.1 Second-Level Descriptor Format

Similar to first-level section descriptors, second-level descriptors provide all of the necessary information for the translation of a large or small page. Table 15-5 shows the format of second-level descriptors.

Table 15-5. Second-Level Descriptor Format

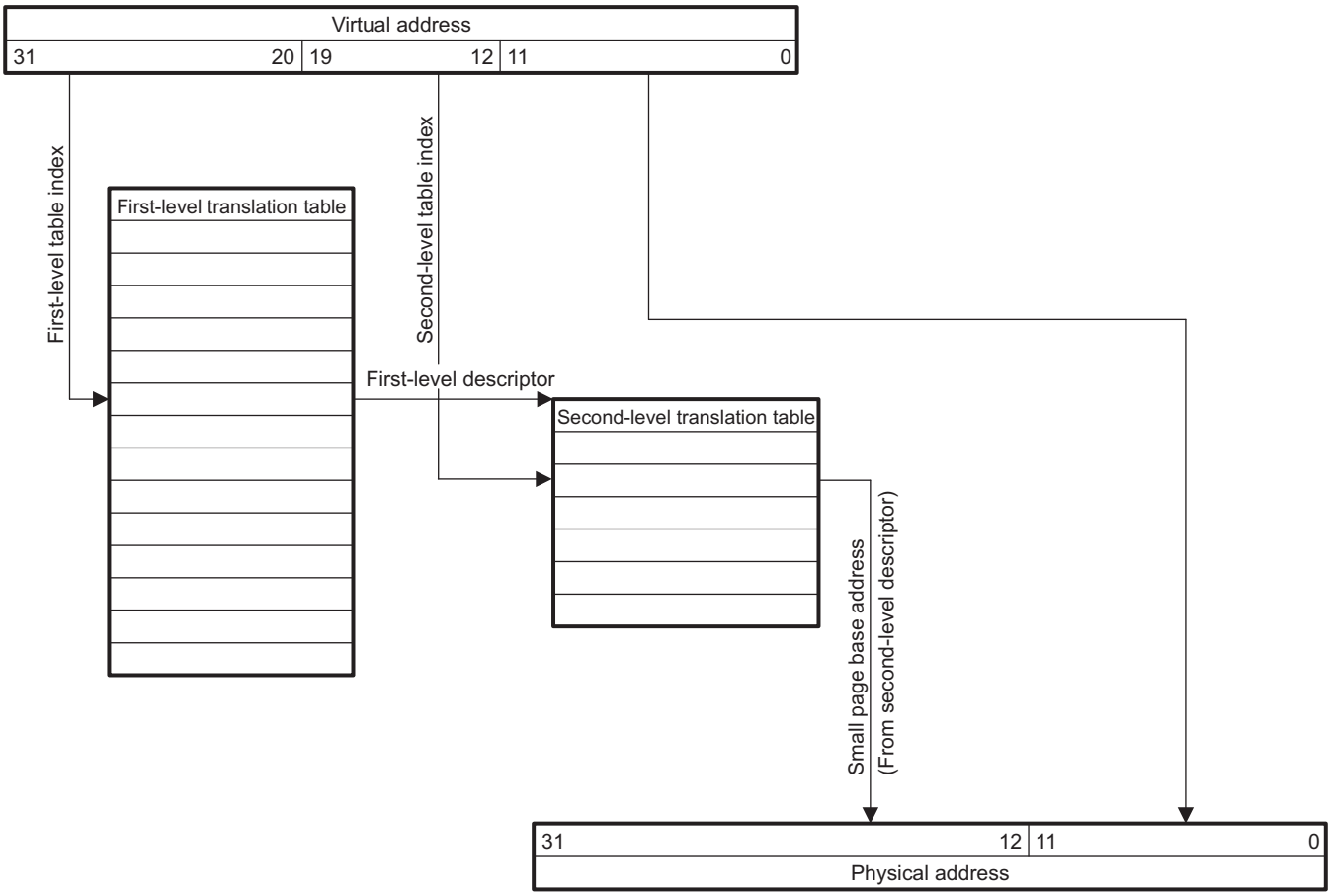
Second-Level Descriptor Format					
31:16	15:12	11:2	1	0	
X			0	0	Fault
Large Page Base Address	X		0	1	Large Page
Small Page Base Address		X	1	X	Small Page

X = Don't care. Set to 0 for future compatibility.

15.3.1.2.3.2 Small Page Translation Summary

Figure 15-11 summarizes the translation process for small pages.

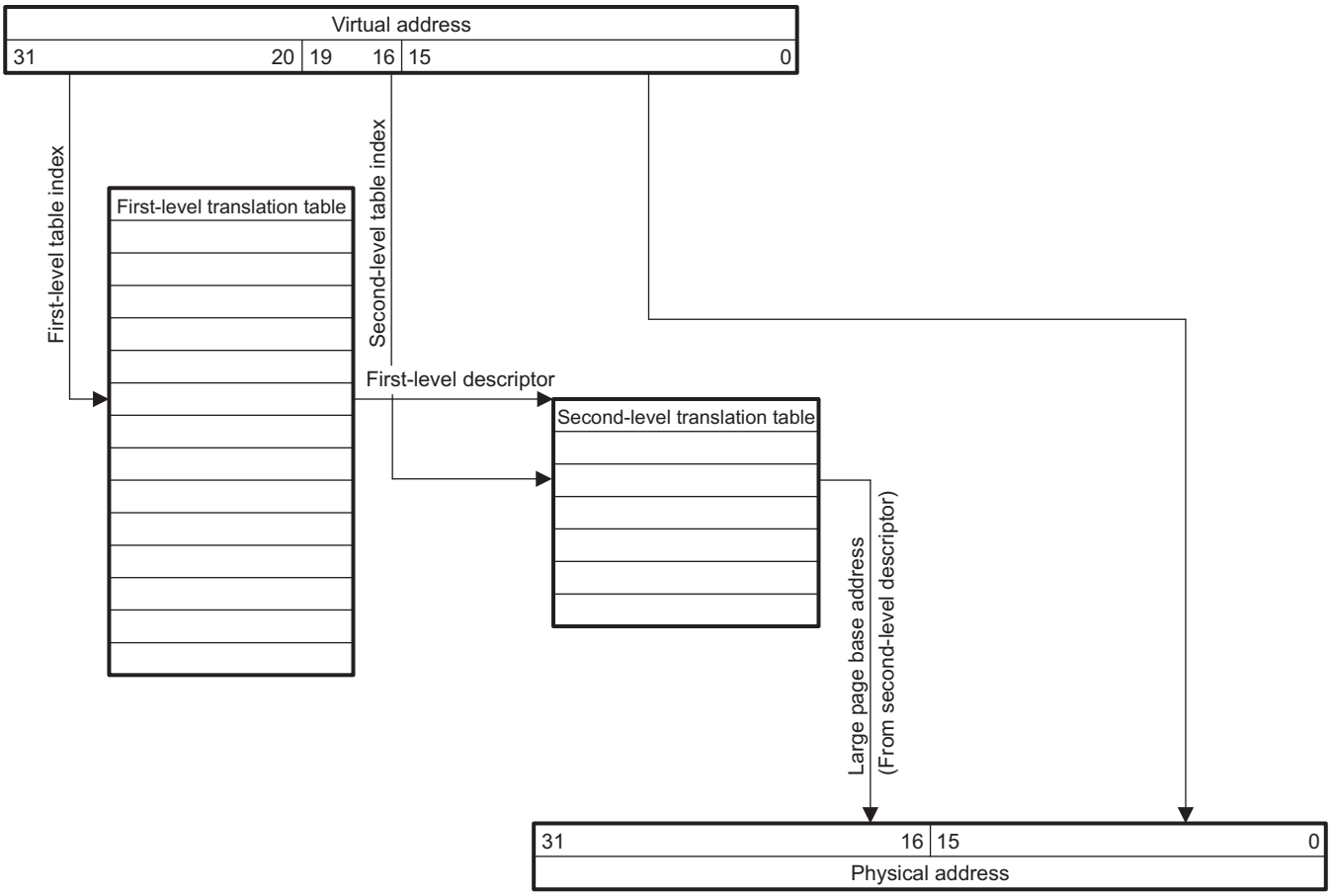
Figure 15-11. Small Page Translation Summary



15.3.1.2.3.3 Large Page Translation Summary

The translation of a large page is similar to the translation of a small page. The difference is that, for a large page, only bits 19 to 16 index into the second-level translation table. The last four bits of the table index are implicitly assumed to be zero as there are 16 identical consecutive entries for a large page. This is shown in Figure 15-12.

Figure 15-12. Large Page Translation Summary



15.3.1.3 Translation Lookaside Buffer

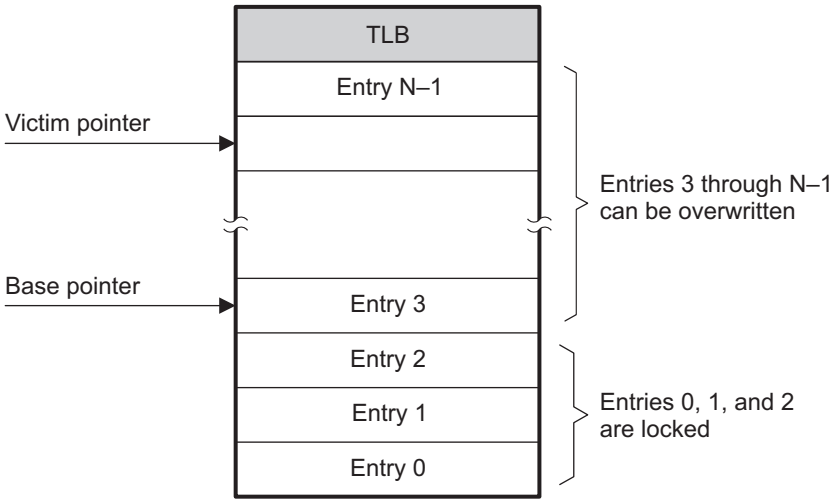
Translating virtual addresses to physical addresses is required for each memory access in systems using an MMU. To accelerate this translation process, a cache, or TLB, holds the result of recent translations.

For every translation, the MMU internal logic first checks whether the requested translation is already cached in the TLB. If the translation is cached, this translation is used; otherwise the translation is retrieved from the translation tables and the TLB is updated. If the TLB is full, one of its entries must be replaced. This entry is selected on a random basis.

The first *n* TLB entries, where *n* < *Total Number N of TLB Entries*, can be protected (locked) against being overwritten by setting the TLB base pointer to *n*. When this mechanism is used, only unprotected entries can be overwritten. The victim pointer indicates the next TLB entry to be written. Figure 15-13 shows an example of the TLB with N TLB entries (ranging from 0 to N-1). The base pointer contains the value "3" protecting Entry 0, Entry 1, and Entry 2 and the victim pointer points to the next TLB entry to be updated.

NOTE: The last TLB entry (Entry N-1) always remains unprotected.

Figure 15-13. TLB Entry Lock Mechanism



The table walking logic automatically writes the TLB entries. The entries can also be manually written, which is done typically to ensure that the translation of time-critical data accesses is already present in the TLB so that they execute as fast as possible. The entries must be locked to prevent them from being overwritten.

15.3.1.3.1 TLB Entry Format

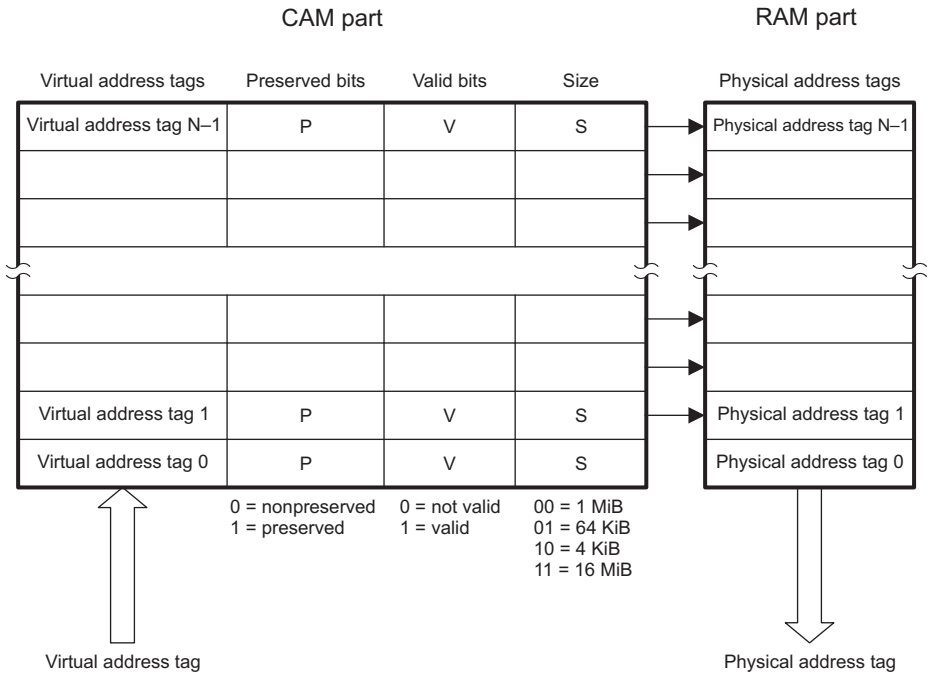
TLB entries consist of two parts:

- The Content Addressable Memory (CAM) part contains the virtual address tag used to determine if a virtual address translation is in the TLB. The TLB acts like a fully associative cache addressed by the virtual address tag. The CAM part also contains the section/page size, as well as the preserved and the valid parameters. See the MMU_CAM register table for more details.
- The Random Addressable Memory (RAM) part contains the address translation that belongs to the virtual address tag. See the MMU_RAM register table for more details.

The valid parameter specifies whether an entry is valid or not. The preserved parameter determines the behavior of an entry in the event of a TLB flush. If an entry is set as preserved, it is not deleted when a TLB is flushed, that is, when MMU_GFLUSH[0] GLOBALFLUSH is set to 1. Preserved entries must be deleted manually. Section 15.3.1.2.2, *First-Level Translation Table* describes the procedure to delete TLB entries.

Figure 15-15 shows the TLB entry structure.

Figure 15-14. TLB Entry Structure



15.3.1.4 No Translation (Bypass) Regions

The MMU provides support for up to four user programmable regions where there is no address translation. Any access to a region specified by the MMU_BYPASS_REGIONx_ADDR and MMU_BYPASS_REGIONx_SIZE registers (where x = 1 to 4) will have no virtual to physical address translation.

15.3.2 MMU Software Reset

To perform a software reset, write 1 in the MMU_SYSCONFIG[1] SOFTRESET bit. The MMU_SYSSTATUS[0] RESETDONE bit indicates that the software reset is complete when its value is 1. When the software reset completes, the MMU_SYSCONFIG[1] SOFTRESET bit is automatically reset. The software must ensure that the software reset completes before doing MMU operations. When an MMU instance is released from reset, its TLB is empty and the MMU is disabled.

15.3.3 MMU Power Management

Table 15-6 describes the power-management features available for the MMU modules.

Table 15-6. MMU Local Power Management Features

Feature	Register
Idle modes	MMU_SYSCONFIG[4:3] IDLEMODE
Clock activity	MMU_SYSCONFIG[9:8] CLOCKACTIVITY
Clock autogating	MMU_SYSCONFIG[0] AUTOIDLE

NOTE: The MMU_SYSCONFIG[9:8] CLOCKACTIVITY bit field is read only.

15.3.4 MMU Interrupt Requests

Table 15-7. MMU Events

Event Flag	Event Mask	Description
MMU_IRQSTATUS[4] MULTIHITFAULT	MMU_IRQENABLE[4] MULTIHITFAULT	Error due to multiple matches in the TLB
MMU_IRQSTATUS[3] TABLEWALKFAULT	MMU_IRQENABLE[3] TABLEWALKFAULT	Error due to error response received during a Table Walk
MMU_IRQSTATUS[2] EMUMISS	MMU_IRQENABLE[2] EMUMISS	Error due to unrecoverable TLB miss during debug (hardware TWL disabled)
MMU_IRQSTATUS[1] TRANSLATIONFAULT	MMU_IRQENABLE[1] TRANSLATIONFAULT	Error due to invalid descriptor in the translation tables (translation fault)
MMU_IRQSTATUS[0] TLBMISS	MMU_IRQENABLE[0] TLBMISS	Error due to unrecoverable TLB miss (hardware TWL disabled)

15.3.5 MMU Error Handling

Table 15-8 summarizes the intended operation for real and potential error conditions.

Table 15-8. Error Handling

Item	Condition	Action
1	Table-walk read has an error response.	Treat generally the same as a translation fault, but set the TableWalkFault interrupt status bit to aid in diagnosis
2	MMU is disabled during table-walk.	Not permitted; can result in loss of the current transaction but must not deadlock the MMU. Avoid this condition by first disabling the table-walk logic and then polling the TWLRunning bit to ensure that no table walk is pending
3	MMU is disabled during an address translation.	Not permitted; can result in access to an unintended location, but must not deadlock MMU. This condition should be avoided by ensuring that no accesses are pending.
4	TLB is accessed during an address translation or a table walk.	Reading permitted; write should be done with care to ensure that the TLB is self-consistent at all times that a translation can occur.
5	TLB is flushed during address translation or a table walk.	Permitted; the flush is processed first, followed by the TWL update.
6	MMU is disabled while an interrupt is pending.	Not permitted; all pending interrupts should be processed before disabling the MMU.
7	Interrupt is not enabled and a fault/miss happens during translation.	If MMU_GPR[0] FAULT_INTR_DIS = 1 : Error response is sent back. If MMU_GPR[0] FAULT_INTR_DIS = 0 : • Mreqdebug = 1 (debug access) : Error response is sent back • Mreqdebug = 0 (application access) : Error response is not sent. MMU is waiting for TLB to be updated through config port. But since the interrupt is not asserted, system does not know there was a fault. This results in deadlock. Software must take care of this by enabling interrupts.

15.4 MMU Low-level Programming Models

This section covers the low-level hardware programming sequences for configuration and usage of the module.

15.4.1 Global Initialization

15.4.1.1 Surrounding Modules Global Initialization

This section identifies the requirements of initializing the surrounding modules when the MMU module is to be used for the first time after a device reset. This initialization of surrounding modules is based on the integration and environment of the MMU. For more information, see Section 15.2, *MMU Module Integration* .

Table 15-9. Global Initialization of Surrounding Modules

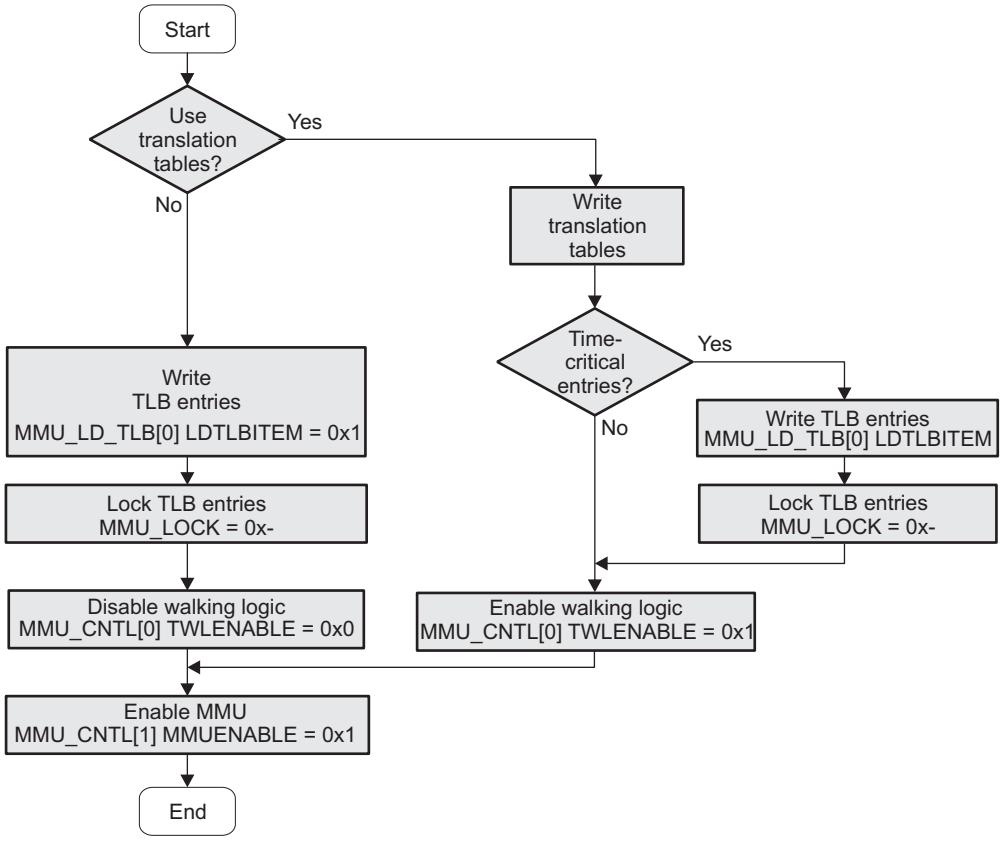
Surrounding Modules	Comments
PRCM	Enable MMU interface/functional clock.
(optional) Interrupt controller(s)	Configure device interrupt controller(s) to enable the interrupts from MMU.
IRQ_CROSSBAR	IRQ_CROSSBAR configuration must be done to allow module IRQs to be mapped to certain device INTC line. For more information see Section 13.4.6.4, *IRQ_CROSSBAR Module Functional Description*, in Chapter 13, *Control Module*.

15.4.1.2 MMU Global Initialization

15.4.1.2.1 Main Sequence - MMU Global Initialization

Figure 15-15 shows the procedure to initialize the MMU after a power-on or software reset.

Figure 15-15. MMU Global Initialization



15.4.1.2.2 Subsequence - Configure a TLB entry

Table 15-10. Configure a TLB Entry

Step	Register / Bit Field / Programming Model	Value
Load the Virtual Address Tag	MMU_CAM[31:12] VATAG	0x-
Protect the TLB entry against flush	MMU_CAM[3] P	0x1
Validate the TLB entry	MMU_CAM[2] V	0x1
Define the page size	MMU_CAM[1:0] PAGESIZE	0x-

15.4.1.3 Operational Modes Configuration

15.4.1.3.1 Main Sequence - Writing TLB Entries Statically

Writing TLB entries statically avoids the need to write translation tables in memory and is commonly used for relatively small address spaces. This method ensures that the translation of time-critical data accesses execute as fast as possible with entries already present in the TLB. These entries must be locked to prevent them from being overwritten.

Table 15-11. MMU Writing TLB Entries Statically

Step	Register/ Bit Field / Programming Model	Value
Execute software reset	MMU_SYSCONFIG[1] SOFTRESET	0x1
Wait for reset to complete	MMU_SYSSTATUS[0] RESETDONE	=0x1
Enable power saving via automatic interface clock gating	MMU_SYSCONFIG[0] AUTOIDLE	0x1
Configure TLB entries	See Table 15-10	
Load the physical Address of the page	MMU_RAM[31:12] PHYSICALADDRESS	0x-
Specify the TLB entry you want to write	MMU_LOCK[8:4] CURRENTVICTIM	0x-
Load the specified entry in the TLB	MMU_LD_TLB[0] LDTLBITEM	0x1
Enable multihit fault and TLB miss	MMU_IRQENABLE[4] MULTIHITFAULT	0x1
	MMU_IRQENABLE[0] TLBMISS	0x1
Enable memory translations	MMU_CNTL[1] MMUENABLE	0x1

15.4.1.3.2 Main Sequence - Protecting TLB Entries

The first *n* TLB entries (with *n* < total number of TLB entries) can be protected from being overwritten with new translations. This is useful to ensure that certain commonly used or time-critical translations are always in the TLB and do not require retrieval using the table walking process.

Table 15-12. Protecting TLB Entries

Step	Registe/Bit Field/Programming Model	Value
Locks the TLB entries	MMU_LOCK[14:10] BASEVALUE	0x-

15.4.1.3.3 Main Sequence - Deleting TLB Entries

Two mechanisms exist to delete TLB entries. All unpreserved TLB entries, that is, TLB entries written with the preserved bit set to zero, can be deleted by invoking a TLB flush. The preserved bit should only be used on protected TLB entries, as it does not prevent replacement by the table walking logic.

Table 15-13. Deleting TLB Entries

Step	Registet / Bit Field / Programming Model	Value
Flush all nonprotected TLB entries	MMU_GFLUSH[0] GLOBALFLUSH	0x1
Flush all TLB entries specified by the CAM register	MMU_FLUSH_ENTRY[0] FLUSHENTRY	0x1

15.4.1.3.4 Main Sequence - Read TLB Entries

TLB entries can be read by the programmer to determine the TLB content at runtime.

Table 15-14. Read TLB Entries

Step	Register / Bit Field / Programming Model	Value
Set the current victim pointer	MMU_LOCK[8:4] CURRENTVICTIM	0x-
Read RAM parts of the TLB entry	MMU_READ_RAM	

Table 15-14. Read TLB Entries (continued)

Step	Register / Bit Field / Programming Model	Value
Read CAM parts of the TLB entry	MMU_READ_CAM	

15.5 MMU Register Manual

15.5.1 MMU Instance Summary

Table 15-15. MMU Instance Summary

Module Name	Base Address (L3/L4 Access)	Base Address (CPU Private Access)	Size
System MMU	0x4881 C000	–	176 Bytes
DSP1_MMU0	0x40D0 1000	0x01D0 1000	176 Bytes
DSP1_MMU1	0x40D0 2000	0x01D0 2000	176 Bytes
DSP2_MMU0	0x4150 1000	0x01D0 1000	176 Bytes
DSP2_MMU1	0x4150 2000	0x01D0 2000	176 Bytes
EVE_MMU0	0x4208 1000	0x4008 1000	176 Bytes
EVE_MMU1	0x4208 2000	0x4008 2000	176 Bytes
IPU_MMU	0x5888 2000	0x5508 2000	176 Bytes

15.5.2 MMU Registers

15.5.2.1 MMU Register Summary

Table 15-16. System MMU Register Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	System MMU Physical Address (L4_PER3 Access)
MMU_REVISION	R	32	0x0000 0000	0x4881 C000
MMU_SYSCONFIG	RW	32	0x0000 0010	0x4881 C010
MMU_SYSSTATUS	R	32	0x0000 0014	0x4881 C014
MMU_IRQSTATUS	RW	32	0x0000 0018	0x4881 C018
MMU_IRQENABLE	RW	32	0x0000 001C	0x4881 C01C
MMU_WALKING_ST	R	32	0x0000 0040	0x4881 C040
MMU_CNTL	RW	32	0x0000 0044	0x4881 C044
MMU_FAULT_AD	R	32	0x0000 0048	0x4881 C048
MMU_TTB	RW	32	0x0000 004C	0x4881 C04C
MMU_LOCK	RW	32	0x0000 0050	0x4881 C050
MMU_LD_TLB	RW	32	0x0000 0054	0x4881 C054
MMU_CAM	RW	32	0x0000 0058	0x4881 C058
MMU_RAM	RW	32	0x0000 005C	0x4881 C05C
MMU_GFLUSH	RW	32	0x0000 0060	0x4881 C060
MMU_FLUSH_ENTRY	RW	32	0x0000 0064	0x4881 C064
MMU_READ_CAM	R	32	0x0000 0068	0x4881 C068
MMU_READ_RAM	R	32	0x0000 006C	0x4881 C06C
MMU_EMU_FAULT_AD	R	32	0x0000 0070	0x4881 C070
RESERVED	R	32	0x0000 0074	0x4881 C074
RESERVED	R	32	0x0000 0078	0x4881 C078
RESERVED	R	32	0x0000 007C	0x4881 C07C
MMU_FAULT_PC	R	32	0x0000 0080	0x4881 C080
MMU_FAULT_STATUS	RW	32	0x0000 0084	0x4881 C084
MMU_GPR	RW	32	0x0000 0088	0x4881 C088
MMU_BYPASS_REGION1_ADDR	RW	32	0x0000 0090	0x4881 C090

Table 15-16. System MMU Register Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	System MMU Physical Address (L4_PER3 Access)
MMU_BYPASS_REGION1_SIZE	RW	32	0x0000 0094	0x4881 C094
MMU_BYPASS_REGION2_ADDR	RW	32	0x0000 0098	0x4881 C098
MMU_BYPASS_REGION2_SIZE	RW	32	0x0000 009C	0x4881 C09C
MMU_BYPASS_REGION3_ADDR	RW	32	0x0000 00A0	0x4881 C0A0
MMU_BYPASS_REGION3_SIZE	RW	32	0x0000 00A4	0x4881 C0A4
MMU_BYPASS_REGION4_ADDR	RW	32	0x0000 00A8	0x4881 C0A8
MMU_BYPASS_REGION4_SIZE	RW	32	0x0000 00AC	0x4881 C0AC

Table 15-17. DSP1 MMU Register Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	DSP1_MMU0 Physical Address (L3_MAIN Access)	DSP1_MMU0 Physical Address (DSP1 Private Access)	DSP1_MMU1 Physical Address (L3_MAIN Access)	DSP1_MMU1 Physical Address (DSP1 Private Access)
MMU_REVISION	R	32	0x0000 0000	0x40D0 1000	0x01D0 1000	0x40D0 2000	0x01D0 2000
MMU_SYSCONFIG	RW	32	0x0000 0010	0x40D0 1010	0x01D0 1010	0x40D0 2010	0x01D0 2010
MMU_SYSSTATUS	R	32	0x0000 0014	0x40D0 1014	0x01D0 1014	0x40D0 2014	0x01D0 2014
MMU_IRQSTATUS	RW	32	0x0000 0018	0x40D0 1018	0x01D0 1018	0x40D0 2018	0x01D0 2018
MMU_IRQENABLE	RW	32	0x0000 001C	0x40D0 101C	0x01D0 101C	0x40D0 201C	0x01D0 201C
MMU_WALKING_ST	R	32	0x0000 0040	0x40D0 1040	0x01D0 1040	0x40D0 2040	0x01D0 2040
MMU_CNTL	RW	32	0x0000 0044	0x40D0 1044	0x01D0 1044	0x40D0 2044	0x01D0 2044
MMU_FAULT_AD	R	32	0x0000 0048	0x40D0 1048	0x01D0 1048	0x40D0 2048	0x01D0 2048
MMU_TTB	RW	32	0x0000 004C	0x40D0 104C	0x01D0 104C	0x40D0 204C	0x01D0 204C
MMU_LOCK	RW	32	0x0000 0050	0x40D0 1050	0x01D0 1050	0x40D0 2050	0x01D0 2050
MMU_LD_TLB	RW	32	0x0000 0054	0x40D0 1054	0x01D0 1054	0x40D0 2054	0x01D0 2054
MMU_CAM	RW	32	0x0000 0058	0x40D0 1058	0x01D0 1058	0x40D0 2058	0x01D0 2058
MMU_RAM	RW	32	0x0000 005C	0x40D0 105C	0x01D0 105C	0x40D0 205C	0x01D0 205C
MMU_GFLUSH	RW	32	0x0000 0060	0x40D0 1060	0x01D0 1060	0x40D0 2060	0x01D0 2060
MMU_FLUSH_ENTRY	RW	32	0x0000 0064	0x40D0 1064	0x01D0 1064	0x40D0 2064	0x01D0 2064
MMU_READ_CAM	R	32	0x0000 0068	0x40D0 1068	0x01D0 1068	0x40D0 2068	0x01D0 2068
MMU_READ_RAM	R	32	0x0000 006C	0x40D0 106C	0x01D0 106C	0x40D0 206C	0x01D0 206C
MMU_EMU_FAULT_AD	R	32	0x0000 0070	0x40D0 1070	0x01D0 1070	0x40D0 2070	0x01D0 2070
RESERVED	R	32	0x0000 0074	0x40D0 1074	0x01D0 1074	0x40D0 2074	0x01D0 2074
RESERVED	R	32	0x0000 0078	0x40D0 1078	0x01D0 1078	0x40D0 2078	0x01D0 2078
RESERVED	R	32	0x0000 007C	0x40D0 107C	0x01D0 107C	0x40D0 207C	0x01D0 207C
MMU_FAULT_PC	R	32	0x0000 0080	0x40D0 1080	0x01D0 1080	0x40D0 2080	0x01D0 2080
MMU_FAULT_STATUS	RW	32	0x0000 0084	0x40D0 1084	0x01D0 1084	0x40D0 2084	0x01D0 2084
MMU_GPR	RW	32	0x0000 0088	0x40D0 1088	0x01D0 1088	0x40D0 2088	0x01D0 2088
MMU_BYPASS_REGION1_ADDR	RW	32	0x0000 0090	0x40D0 1090	0x01D0 1090	0x40D0 2090	0x01D0 2090
MMU_BYPASS_REGION1_SIZE	RW	32	0x0000 0094	0x40D0 1094	0x01D0 1094	0x40D0 2094	0x01D0 2094
MMU_BYPASS_REGION2_ADDR	RW	32	0x0000 0098	0x40D0 1098	0x01D0 1098	0x40D0 2098	0x01D0 2098
MMU_BYPASS_REGION2_SIZE	RW	32	0x0000 009C	0x40D0 109C	0x01D0 109C	0x40D0 209C	0x01D0 209C
MMU_BYPASS_REGION3_ADDR	RW	32	0x0000 00A0	0x40D0 10A0	0x01D0 10A0	0x40D0 20A0	0x01D0 20A0
MMU_BYPASS_REGION3_SIZE	RW	32	0x0000 00A4	0x40D0 10A4	0x01D0 10A4	0x40D0 20A4	0x01D0 20A4
MMU_BYPASS_REGION4_ADDR	RW	32	0x0000 00A8	0x40D0 10A8	0x01D0 10A8	0x40D0 20A8	0x01D0 20A8
MMU_BYPASS_REGION4_SIZE	RW	32	0x0000 00AC	0x40D0 10AC	0x01D0 10AC	0x40D0 20AC	0x01D0 20AC

Table 15-18. DSP2 MMU Register Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	DSP2_MMU0 Physical Address (L3_MAIN Access)	DSP2_MMU0 Physical Address (DSP2 Private Access)	DSP2_MMU1 Physical Address (L3_MAIN Access)	DSP2_MMU1 Physical Address (DSP2 Private Access)
MMU_REVISION	R	32	0x0000 0000	0x4150 1000	0x01D0 1000	0x4150 2000	0x01D0 2000
MMU_SYSCONFIG	RW	32	0x0000 0010	0x4150 1010	0x01D0 1010	0x4150 2010	0x01D0 2010
MMU_SYSSTATUS	R	32	0x0000 0014	0x4150 1014	0x01D0 1014	0x4150 2014	0x01D0 2014
MMU_IRQSTATUS	RW	32	0x0000 0018	0x4150 1018	0x01D0 1018	0x4150 2018	0x01D0 2018
MMU_IRQENABLE	RW	32	0x0000 001C	0x4150 101C	0x01D0 101C	0x4150 201C	0x01D0 201C
MMU_WALKING_ST	R	32	0x0000 0040	0x4150 1040	0x01D0 1040	0x4150 2040	0x01D0 2040
MMU_CNTL	RW	32	0x0000 0044	0x4150 1044	0x01D0 1044	0x4150 2044	0x01D0 2044
MMU_FAULT_AD	R	32	0x0000 0048	0x4150 1048	0x01D0 1048	0x4150 2048	0x01D0 2048
MMU_TTB	RW	32	0x0000 004C	0x4150 104C	0x01D0 104C	0x4150 204C	0x01D0 204C
MMU_LOCK	RW	32	0x0000 0050	0x4150 1050	0x01D0 1050	0x4150 2050	0x01D0 2050
MMU_LD_TLB	RW	32	0x0000 0054	0x4150 1054	0x01D0 1054	0x4150 2054	0x01D0 2054
MMU_CAM	RW	32	0x0000 0058	0x4150 1058	0x01D0 1058	0x4150 2058	0x01D0 2058
MMU_RAM	RW	32	0x0000 005C	0x4150 105C	0x01D0 105C	0x4150 205C	0x01D0 205C
MMU_GFLUSH	RW	32	0x0000 0060	0x4150 1060	0x01D0 1060	0x4150 2060	0x01D0 2060
MMU_FLUSH_ENTRY	RW	32	0x0000 0064	0x4150 1064	0x01D0 1064	0x4150 2064	0x01D0 2064
MMU_READ_CAM	R	32	0x0000 0068	0x4150 1068	0x01D0 1068	0x4150 2068	0x01D0 2068
MMU_READ_RAM	R	32	0x0000 006C	0x4150 106C	0x01D0 106C	0x4150 206C	0x01D0 206C
MMU_EMU_FAULT_AD	R	32	0x0000 0070	0x4150 1070	0x01D0 1070	0x4150 2070	0x01D0 2070
RESERVED	R	32	0x0000 0074	0x4150 1074	0x01D0 1074	0x4150 2074	0x01D0 2074
RESERVED	R	32	0x0000 0078	0x4150 1078	0x01D0 1078	0x4150 2078	0x01D0 2078
RESERVED	R	32	0x0000 007C	0x4150 107C	0x01D0 107C	0x4150 207C	0x01D0 207C
MMU_FAULT_PC	R	32	0x0000 0080	0x4150 1080	0x01D0 1080	0x4150 2080	0x01D0 2080
MMU_FAULT_STATUS	RW	32	0x0000 0084	0x4150 1084	0x01D0 1084	0x4150 2084	0x01D0 2084
MMU_GPR	RW	32	0x0000 0088	0x4150 1088	0x01D0 1088	0x4150 2088	0x01D0 2088
MMU_BYPASS_REGION1_ADDR	RW	32	0x0000 0090	0x4150 1090	0x01D0 1090	0x4150 2090	0x01D0 2090
MMU_BYPASS_REGION1_SIZE	RW	32	0x0000 0094	0x4150 1094	0x01D0 1094	0x4150 2094	0x01D0 2094
MMU_BYPASS_REGION2_ADDR	RW	32	0x0000 0098	0x4150 1098	0x01D0 1098	0x4150 2098	0x01D0 2098
MMU_BYPASS_REGION2_SIZE	RW	32	0x0000 009C	0x4150 109C	0x01D0 109C	0x4150 209C	0x01D0 209C
MMU_BYPASS_REGION3_ADDR	RW	32	0x0000 00A0	0x4150 10A0	0x01D0 10A0	0x4150 20A0	0x01D0 20A0
MMU_BYPASS_REGION3_SIZE	RW	32	0x0000 00A4	0x4150 10A4	0x01D0 10A4	0x4150 20A4	0x01D0 20A4
MMU_BYPASS_REGION4_ADDR	RW	32	0x0000 00A8	0x4150 10A8	0x01D0 10A8	0x4150 20A8	0x01D0 20A8
MMU_BYPASS_REGION4_SIZE	RW	32	0x0000 00AC	0x4150 10AC	0x01D0 10AC	0x4150 20AC	0x01D0 20AC

Table 15-19. EVE MMU Register Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	EVE_MMU0 Physical Address (L3_MAIN Access)	EVE_MMU0 Physical Address (EVE Private Access)	EVE_MMU1 Physical Address (L3_MAIN Access)	EVE_MMU1 Physical Address (EVE Private Access)
MMU_REVISION	R	32	0x0000 0000	0x4208 1000	0x4008 1000	0x4208 2000	0x4008 2000
MMU_SYSCONFIG	RW	32	0x0000 0010	0x4208 1010	0x4008 1010	0x4208 2010	0x4008 2010
MMU_SYSSTATUS	R	32	0x0000 0014	0x4208 1014	0x4008 1014	0x4208 2014	0x4008 2014
MMU_IRQSTATUS	RW	32	0x0000 0018	0x4208 1018	0x4008 1018	0x4208 2018	0x4008 2018
MMU_IRQENABLE	RW	32	0x0000 001C	0x4208 101C	0x4008 101C	0x4208 201C	0x4008 201C
MMU_WALKING_ST	R	32	0x0000 0040	0x4208 1040	0x4008 1040	0x4208 2040	0x4008 2040

Table 15-19. EVE MMU Register Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	EVE_MMU0 Physical Address (L3_MAIN Access)	EVE_MMU0 Physical Address (EVE Private Access)	EVE_MMU1 Physical Address (L3_MAIN Access)	EVE_MMU1 Physical Address (EVE Private Access)
MMU_CNTL	RW	32	0x0000 0044	0x4208 1044	0x4008 1044	0x4208 2044	0x4008 2044
MMU_FAULT_AD	R	32	0x0000 0048	0x4208 1048	0x4008 1048	0x4208 2048	0x4008 2048
MMU_TTB	RW	32	0x0000 004C	0x4208 104C	0x4008 104C	0x4208 204C	0x4008 204C
MMU_LOCK	RW	32	0x0000 0050	0x4208 1050	0x4008 1050	0x4208 2050	0x4008 2050
MMU_LD_TLB	RW	32	0x0000 0054	0x4208 1054	0x4008 1054	0x4208 2054	0x4008 2054
MMU_CAM	RW	32	0x0000 0058	0x4208 1058	0x4008 1058	0x4208 2058	0x4008 2058
MMU_RAM	RW	32	0x0000 005C	0x4208 105C	0x4008 105C	0x4208 205C	0x4008 205C
MMU_GFLUSH	RW	32	0x0000 0060	0x4208 1060	0x4008 1060	0x4208 2060	0x4008 2060
MMU_FLUSH_ENTRY	RW	32	0x0000 0064	0x4208 1064	0x4008 1064	0x4208 2064	0x4008 2064
MMU_READ_CAM	R	32	0x0000 0068	0x4208 1068	0x4008 1068	0x4208 2068	0x4008 2068
MMU_READ_RAM	R	32	0x0000 006C	0x4208 106C	0x4008 106C	0x4208 206C	0x4008 206C
MMU_EMU_FAULT_AD	R	32	0x0000 0070	0x4208 1070	0x4008 1070	0x4208 2070	0x4008 2070
RESERVED	R	32	0x0000 0074	0x4208 1074	0x4008 1074	0x4208 2074	0x4008 2074
RESERVED	R	32	0x0000 0078	0x4208 1078	0x4008 1078	0x4208 2078	0x4008 2078
RESERVED	R	32	0x0000 007C	0x4208 107C	0x4008 107C	0x4208 207C	0x4008 207C
MMU_FAULT_PC	R	32	0x0000 0080	0x4208 1080	0x4008 1080	0x4208 2080	0x4008 2080
MMU_FAULT_STATUS	RW	32	0x0000 0084	0x4208 1084	0x4008 1084	0x4208 2084	0x4008 2084
MMU_GPR	RW	32	0x0000 0088	0x4208 1088	0x4008 1088	0x4208 2088	0x4008 2088
MMU_BYPASS_REGION1_ADDR	RW	32	0x0000 0090	0x4208 1090	0x4008 1090	0x4208 2090	0x4008 2090
MMU_BYPASS_REGION1_SIZE	RW	32	0x0000 0094	0x4208 1094	0x4008 1094	0x4208 2094	0x4008 2094
MMU_BYPASS_REGION2_ADDR	RW	32	0x0000 0098	0x4208 1098	0x4008 1098	0x4208 2098	0x4008 2098
MMU_BYPASS_REGION2_SIZE	RW	32	0x0000 009C	0x4208 109C	0x4008 109C	0x4208 209C	0x4008 209C
MMU_BYPASS_REGION3_ADDR	RW	32	0x0000 00A0	0x4208 10A0	0x4008 10A0	0x4208 20A0	0x4008 20A0
MMU_BYPASS_REGION3_SIZE	RW	32	0x0000 00A4	0x4208 10A4	0x4008 10A4	0x4208 20A4	0x4008 20A4
MMU_BYPASS_REGION4_ADDR	RW	32	0x0000 00A8	0x4208 10A8	0x4008 10A8	0x4208 20A8	0x4008 20A8
MMU_BYPASS_REGION4_SIZE	RW	32	0x0000 00AC	0x4208 10AC	0x4008 10AC	0x4208 20AC	0x4008 20AC

Table 15-20. IPU MMU Register Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	IPU_MMU Physical Address (L3_MAIN Access)	IPU_MMU Physical Address (IPU Private Access)
MMU_REVISION	R	32	0x0000 0000	0x5888 2000	0x5508 2000
MMU_SYSCONFIG	RW	32	0x0000 0010	0x5888 2010	0x5508 2010
MMU_SYSSTATUS	R	32	0x0000 0014	0x5888 2014	0x5508 2014
MMU_IRQSTATUS	RW	32	0x0000 0018	0x5888 2018	0x5508 2018
MMU_IRQENABLE	RW	32	0x0000 001C	0x5888 201C	0x5508 201C
MMU_WALKING_ST	R	32	0x0000 0040	0x5888 2040	0x5508 2040
MMU_CNTL	RW	32	0x0000 0044	0x5888 2044	0x5508 2044
MMU_FAULT_AD	R	32	0x0000 0048	0x5888 2048	0x5508 2048
MMU_TTB	RW	32	0x0000 004C	0x5888 204C	0x5508 204C
MMU_LOCK	RW	32	0x0000 0050	0x5888 2050	0x5508 2050
MMU_LD_TLB	RW	32	0x0000 0054	0x5888 2054	0x5508 2054
MMU_CAM	RW	32	0x0000 0058	0x5888 2058	0x5508 2058

Table 15-20. IPU MMU Register Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	IPU_MMU Physical Address (L3_MAIN Access)	IPU_MMU Physical Address (IPU Private Access)
MMU_RAM	RW	32	0x0000 005C	0x5888 205C	0x5508 205C
MMU_GFLUSH	RW	32	0x0000 0060	0x5888 2060	0x5508 2060
MMU_FLUSH_ENTRY	RW	32	0x0000 0064	0x5888 2064	0x5508 2064
MMU_READ_CAM	R	32	0x0000 0068	0x5888 2068	0x5508 2068
MMU_READ_RAM	R	32	0x0000 006C	0x5888 206C	0x5508 206C
MMU_EMU_FAULT_AD	R	32	0x0000 0070	0x5888 2070	0x5508 2070
RESERVED	R	32	0x0000 0074	0x5888 2074	0x5508 2074
RESERVED	R	32	0x0000 0078	0x5888 2078	0x5508 2078
RESERVED	R	32	0x0000 007C	0x5888 207C	0x5508 207C
MMU_FAULT_PC	R	32	0x0000 0080	0x5888 2080	0x5508 2080
MMU_FAULT_STATUS	RW	32	0x0000 0084	0x5888 2084	0x5508 2084
MMU_GPR	RW	32	0x0000 0088	0x5888 2088	0x5508 2088

15.5.2.2 MMU Register Description

Table 15-21. MMU_REVISION

Address Offset	0x0000 0000		
Physical Address	See Section 15.5.2.1	**Instance**	See Table 15-15
Description	This register contains the IP revision code		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REVISION																															

Bits	Field Name	Description	Type	Reset
31:0	REVISION	IP Revision	R	See (1)

(1) TI internal data

Table 15-22. Register Call Summary for Register MMU_REVISION

MMU Register Manual
- MMU Register Summary: [0][1][2][3][4]

Table 15-23. MMU_SYSCONFIG

Address Offset	0x0000 0010		
Physical Address	See Section 15.5.2.1	**Instance**	See Table 15-15
Description	This register controls the various parameters of the OCP interface		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						CLOCKACTIVITY		RESERVED				IDLEMODE		RESERVED	SOFTRESET	AUTOIDLE

Bits	Field Name	Description	Type	Reset
31:10	RESERVED	Write 0's for future compatibility. Reads return 0	R	0x000000
9:8	CLOCKACTIVITY	Clock activity during wake-up mode 0x0: Functional and OCP clocks can be switched off	R	0x0
7:5	RESERVED	Write 0's for future compatibility Reads returns 0	R	0x0
4:3	IDLEMODE	Idle mode 0x0: Force-idle. An idle request is acknowledged unconditionally 0x1: No-idle. An idle request is never acknowledged 0x2: Smart-idle. Acknowledgement to an idle request is given based on the internal activity of the module 0x3: Reserved. Do not use	RW	0x0
2	RESERVED	Write 0's for future compatibility Reads returns 0	R	0x0
1	SOFTRESET	Software reset. This bit is automatically reset by the hardware. During reads, it always return 0 Write 0x0: No functional effect Write 0x1: The module is reset	W	0x0
0	AUTOIDLE	Internal OCP clock gating strategy 0x0: OCP clock is free-running 0x1: Automatic interconnect clock gating strategy is applied, based on the interconnect interface activity	RW	0x0

Table 15-24. Register Call Summary for Register MMU_SYSCONFIG

MMU Functional Description
- MMU Software Reset: [0][1]
- MMU Power Management: [2][3][4][5]

MMU Low-level Programming Models
- Operational Modes Configuration: [6][7]

MMU Register Manual
- MMU Register Summary: [8][9][10][11][12]

Table 15-25. MMU_SYSSTATUS

Address Offset	0x0000 0014		
Physical Address	See Section 15.5.2.1	**Instance**	See Table 15-15
Description	This register provides status information about the module, excluding the interrupt status information		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															RESETDONE

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reads returns 0	R	0x000000
0	RESETDONE	Internal reset monitoring Read 0x0: Internal module reset in on-going Read 0x1: Reset completed	R	-

Table 15-26. Register Call Summary for Register MMU_SYSSTATUS

MMU Functional Description • MMU Software Reset: [0]
MMU Low-level Programming Models • Operational Modes Configuration: [1]
MMU Register Manual • MMU Register Summary: [2][3][4][5][6]

Table 15-27. MMU_IRQSTATUS

Address Offset	0x0000 0018		
Physical Address	See Section 15.5.2.1	**Instance**	See Table 15-15
Description	This interrupt status register regroups all the status of the module internal events that can generate an interrupt.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																											MULTIHITFAULT	TABLEWALKFAULT	EMUMISS	TRANSLATIONFAULT	TLBMISS

Bits	Field Name	Description	Type	Reset
31:5	RESERVED	Write 0's for future compatibility. Read returns 0	R	0x0000000
4	MULTIHITFAULT	Error due to multiple matches in the TLB Read 0x0: MultiHitFault false Write 0x0: MultiHitFault status bit unchanged Write 0x1: MultiHitFault status bit is reset Read 0x1: MultiHitFault is true ('pending')	RW (W1toClr)	0x0

Bits	Field Name	Description	Type	Reset
3	TABLEWALKFAULT	Error response received during a Table Walk Read 0x0: TableWalkFault false Write 0x0: TableWalkFault status bit unchanged Write 0x1: TableWalkFault status bit is reset Read 0x1: TableWalkFault is true ('pending')	RW (W1toClr)	0x0
2	EMUMISS	Unrecoverable TLB miss during debug (hardware TWL disabled) Read 0x0: EMUMiss false Write 0x0: EMUMiss status bit unchanged Write 0x1: EMUMiss status bit is reset Read 0x1: EMUMiss is true ('pending')	RW (W1toClr)	0x0
1	TRANSLATIONFAULT	Invalid descriptor in translation tables (translation fault) Read 0x0: TranslationFault false Write 0x0: TranslationFault status bit unchanged Write 0x1: TranslationFault status bit is reset Read 0x1: TranslationFault is true ('pending')	RW (W1toClr)	0x0
0	TLBMISS	Unrecoverable TLB miss (hardware TWL disabled) Read 0x0: TLBMiss false Write 0x0: TLBMiss status bit unchanged Write 0x1: TLBMiss status bit is reset Read 0x1: TLBMiss is true ('pending')	RW (W1toClr)	0x0

Table 15-28. Register Call Summary for Register MMU_IRQSTATUS

MMU Functional Description
- MMU Interrupt Requests: [0][1][2][3][4]

MMU Register Manual
- MMU Register Summary: [5][6][7][8][9]

Table 15-29. MMU_IRQENABLE

Address Offset	0x0000 001C		
Physical Address	See Section 15.5.2.1	**Instance**	See Table 15-15
Description	The interrupt enable register allows to mask/unmask the module internal sources of interrupt, on a event-by-event basis.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																											MULTIHITFAULT	TABLEWALKFAULT	EMUMISS	TRANSLATIONFAULT	TLBMISS

Bits	Field Name	Description	Type	Reset
31:5	RESERVED	Write 0's for future compatibility Read returns 0	R	0x0000000
4	MULTIHITFAULT	Error due to multiple matches in the TLB 0x0: MultiHitFault is masked 0x1: MultiHitFault event generates an interrupt if occurs	RW	0x0

Bits	Field Name	Description	Type	Reset
3	TABLEWALKFAULT	Error response received during a Table Walk 0x0: TableWalkFault is masked 0x1: TableWalkFault event generates an interrupt if occurs	RW	0x0
2	EMUMISS	Unrecoverable TLB miss during debug (hardware TWL disabled) 0x0: EMUMiss interrupt is masked 0x1: EMUMiss event generates an interrupt when it occurs	RW	0x0
1	TRANSLATIONFAULT	Invalid descriptor in translation tables (translation fault) 0x0: TranslationFault is masked 0x1: TranslationFault event generates an interrupt if occurs	RW	0x0
0	TLBMISS	Unrecoverable TLB miss (hardware TWL disabled) 0x0: TLBMiss interrupt is masked 0x1: TLBMiss event generates an interrupt when if occurs	RW	0x0

Table 15-30. Register Call Summary for Register MMU_IRQENABLE

MMU Functional Description • MMU Interrupt Requests: [0][1][2][3][4]
MMU Low-level Programming Models • Operational Modes Configuration: [5][6]
MMU Register Manual • MMU Register Summary: [7][8][9][10][11]

Table 15-31. MMU_WALKING_ST

Address Offset	0x0000 0040		
Physical Address	See Section 15.5.2.1	**Instance**	See Table 15-15
Description	This register provides status information about the table walking logic		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															TWLRUNNING

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reads return 0	R	0x0000 0000
0	TWLRUNNING	Table Walking Logic is running Read 0x0: TWL Completed Read 0x1: TWL Running	R	0x0

Table 15-32. Register Call Summary for Register MMU_WALKING_ST

MMU Register Manual • MMU Register Summary: [0][1][2][3][4]

Table 15-33. MMU_CNTL

Address Offset	0x0000 0044		
Physical Address	See Section 15.5.2.1	**Instance**	See Table 15-15
Description	This register programs the MMU features		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												EMUTLBUPDATE	TWLENABLE	MMUENABLE	RESERVED

Bits	Field Name	Description	Type	Reset
31:4	RESERVED	Write 0's for future compatibility. Reads return 0	R	0x0000000
3	EMUTLBUPDATE	Enable TLB update on emulator table walk 0x0: Emulator TLB update disabled 0x1: Emulator TLB update enabled	RW	0x0
2	TWLENABLE	Table Walking Logic enable 0x0: TWL disabled 0x1: TWL enabled	RW	0x0
1	MMUENABLE	MMU enable 0x0: MMU disabled 0x1: MMU enabled	RW	0x0
0	RESERVED	Write 0's for future compatibility. Reads return 0	R	0x0

Table 15-34. Register Call Summary for Register MMU_CNTL

MMU Low-level Programming Models
- Operational Modes Configuration: [0]

MMU Register Manual
- MMU Register Summary: [1][2][3][4][5]

Table 15-35. MMU_FAULT_AD

Address Offset	0x0000 0048		
Physical Address	See Section 15.5.2.1	**Instance**	See Table 15-15
Description	This register contains the virtual address that generated the interrupt		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FAULTADDRESS																															

Bits	Field Name	Description	Type	Reset
31:0	FAULTADDRESS	Virtual address of the access that generated a fault	R	0x0000 0000

Table 15-36. Register Call Summary for Register MMU_FAULT_AD

MMU Register Manual
- MMU Register Summary: [0][1][2][3][4]

Table 15-37. MMU_TTB

Address Offset	0x0000 004C		
Physical Address	See Section 15.5.2.1	**Instance**	See Table 15-15
Description	This register contains the Translation Table Base address		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TTBADDRESS																								RESERVED							

Bits	Field Name	Description	Type	Reset
31:7	TTBADDRESS	Translation Table Base Address	RW	0x0000000
6:0	RESERVED	Write 0's for future compatibility. Reads retiurn 0	R	0x00

Table 15-38. Register Call Summary for Register MMU_TTB

MMU Register Manual
- MMU Register Summary: [0][1][2][3][4]

Table 15-39. MMU_LOCK

Address Offset	0x0000 0050		
Physical Address	See Section 15.5.2.1	**Instance**	See Table 15-15
Description	This register locks some of the TLB entries		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																BASEVALUE						RESERVED		CURRENTVICTIM						RESERVED	

Bits	Field Name	Description	Type	Reset
31:15	RESERVED	Write 0's for future compatibility. Reads return 0	R	0x00000
14:10	BASEVALUE	Locked entries base value.	RW	0x00
9	RESERVED	Write 0's for future compatibility. Reads return 0	R	0x0
8:4	CURRENTVICTIM	Current entry to be updated either by the TWL or by the software. Write value : TLB entry to be updated by software Read value : TLB entry that will be updated by table walk logic. This will be same as BASEVALUE when there are no tablewalks.	RW	0x00
3:0	RESERVED	Write 0's for future compatibility. Reads return 0	R	0x0

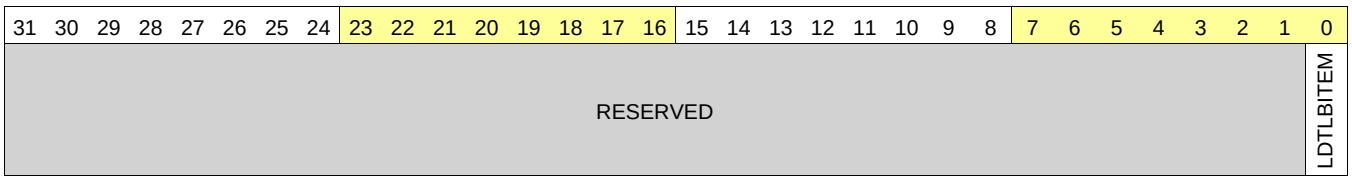
Table 15-40. Register Call Summary for Register MMU_LOCK

MMU Low-level Programming Models
- Operational Modes Configuration: [0][1][2]

MMU Register Manual
- MMU Register Summary: [3][4][5][6][7]
- MMU Register Description: [8][9]

Table 15-41. MMU_LD_TLB

Address Offset	0x0000 0054		
Physical Address	See Section 15.5.2.1	**Instance**	See Table 15-15
Description	This register loads a TLB entry (CAM+RAM)		
Type	RW		



Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Write 0's for future compatibility. Reads return 0	R	0x0000 0000
0	LDTLBITEM	Write (load) data in the TLB. Reads return 0. Write 0x0: No functional effect Write 0x1: Load TLB data	W	0x0

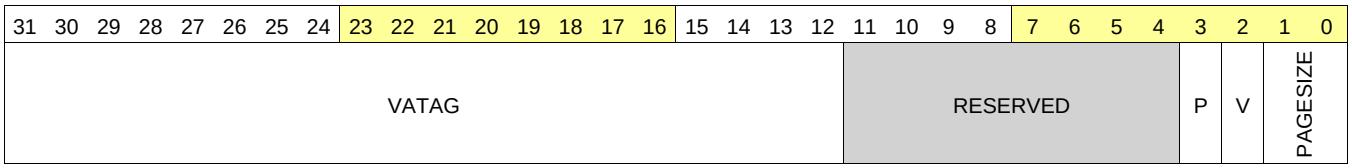
Table 15-42. Register Call Summary for Register MMU_LD_TLB

MMU Low-level Programming Models
- Operational Modes Configuration: [0]

MMU Register Manual
- MMU Register Summary: [1][2][3][4][5]

Table 15-43. MMU_CAM

Address Offset	0x0000 0058		
Physical Address	See Section 15.5.2.1	**Instance**	See Table 15-15
Description	This register holds a CAM entry		
Type	RW		



Bits	Field Name	Description	Type	Reset
31:12	VATAG	Virtual address tag	RW	0x00000
11:4	RESERVED	Write 0's for future compatibility. Reads return 0	R	0x00
3	P	Preserved bit 0x0: TLB entry may be flushed 0x1: TLB entry is protected against flush	RW	0x0
2	V	Valid bit 0x0: TLB entry is invalid 0x1: TLB entry is valid	RW	0x0

Bits	Field Name	Description	Type	Reset
1:0	PAGESIZE	Page size 0x0: Section (1 MiB) 0x1: Large page (64 KiB) 0x2: Small page (4 KiB) 0x3: Supersection (16 MiB)	RW	0x0

Table 15-44. Register Call Summary for Register MMU_CAM

MMU Functional Description • Translation Lookaside Buffer: [0]
MMU Low-level Programming Models • MMU Global Initialization: [1][2][3][4]
MMU Register Manual • MMU Register Summary: [5][6][7][8][9] • MMU Register Description: [10]

Table 15-45. MMU_RAM

Address Offset	0x0000 005C		
Physical Address	See Section 15.5.2.1	**Instance**	See Table 15-15
Description	This register contains bits [31:12] of the physical address to be written to a TLB entry pointed to by CURRENTVICTIM field of MMU_LOCK register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PHYSICALADDRESS																				RESERVED											

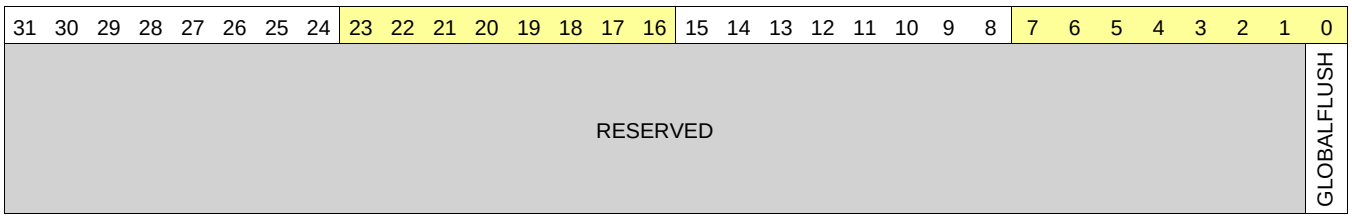
Bits	Field Name	Description	Type	Reset
31:12	PHYSICALADDRESS	Physical address of the page	RW	0x00000
11:0	RESERVED	Write 0's for future compatibility. Reads return 0	R	0x0

Table 15-46. Register Call Summary for Register MMU_RAM

MMU Functional Description • Translation Lookaside Buffer: [0]
MMU Low-level Programming Models • Operational Modes Configuration: [1]
MMU Register Manual • MMU Register Summary: [2][3][4][5][6]

Table 15-47. MMU_GFLUSH

Address Offset	0x0000 0060		
Physical Address	See Section 15.5.2.1	**Instance**	See Table 15-15
Description	This register flushes all the non-protected TLB entries		
Type	RW		



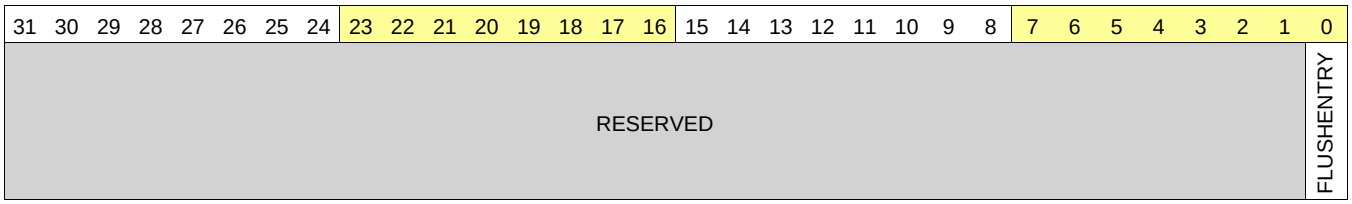
Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Write 0's for future compatibility. Reads return 0	R	0x0000 0000
0	GLOBALFLUSH	Flush all the non-protected TLB entries when set. Reads return 0. Write 0x0: No functional effect Write 0x1: Flush all the non-protected TLB entries	W	0x0

Table 15-48. Register Call Summary for Register MMU_GFLUSH

MMU Functional Description • Translation Lookaside Buffer: [0]
MMU Low-level Programming Models • Operational Modes Configuration: [1]
MMU Register Manual • MMU Register Summary: [2][3][4][5][6]

Table 15-49. MMU_FLUSH_ENTRY

Address Offset	0x0000 0064		
Physical Address	See Section 15.5.2.1	**Instance**	See Table 15-15
Description	This register flushes the entry pointed to by the CAM virtual address		
Type	RW		



Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Write 0's for future compatibility. Reads return 0	R	0x0000 0000
0	FLUSHENTRY	Flush the TLB entry pointed by the virtual address (VATag) in MMU_CAM register, even if this entry is set protected. Reads return 0. Write 0x0: No functional effect Write 0x1: Flush all the TLB entries specified by the CAM register	W	0x0

Table 15-50. Register Call Summary for Register MMU_FLUSH_ENTRY

MMU Low-level Programming Models • Operational Modes Configuration: [0]
MMU Register Manual • MMU Register Summary: [1][2][3][4][5]

Table 15-51. MMU_READ_CAM

Address Offset	0x0000 0068		
Physical Address	See Section 15.5.2.1	**Instance**	See Table 15-15
Description	This register reads CAM data from a CAM entry		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VATAG																RESERVED								P	V	PAGESIZE					

Bits	Field Name	Description	Type	Reset
31:12	VATAG	Virtual address tag	R	0x00000
11:4	RESERVED	Reads return 0	R	0x00
3	P	Preserved bit	R	0x0
		Read 0x0: TLB entry may be flushed		
		Read 0x1: TLB entry is protected against flush		
2	V	Valid bit	R	0x0
		Read 0x0: TLB entry is invalid		
		Read 0x1: TLB entry is valid		
1:0	PAGESIZE	Page size	R	0x0
		Read 0x0: Section (1 MiB)		
		Read 0x1: Large page (64 KiB)		
		Read 0x2: Small page (4 KiB)		
		Read 0x3: Supersection (16 MiB)		

Table 15-52. Register Call Summary for Register MMU_READ_CAM

MMU Low-level Programming Models • Operational Modes Configuration: [0]
MMU Register Manual • MMU Register Summary: [1][2][3][4][5]

Table 15-53. MMU_READ_RAM

Address Offset	0x0000 006C		
Physical Address	See Section 15.5.2.1	**Instance**	See Table 15-15
Description	This register reads bits [31:12] of the physical address from the TLB entry pointed to by CURRENTVICTIM field of the MMU_LOCK register.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PHYSICALADDRESS																RESERVED															

Bits	Field Name	Description	Type	Reset
31:12	PHYSICALADDRESS	Physical address of the page	R	0x00000
11:0	RESERVED	Reads return 0	R	0x0

Table 15-54. Register Call Summary for Register MMU_READ_RAM

MMU Low-level Programming Models
- Operational Modes Configuration: [0]

MMU Register Manual
- MMU Register Summary: [1][2][3][4][5]

Table 15-55. MMU_EMU_FAULT_AD

Address Offset	0x0000 0070		
Physical Address	See Section 15.5.2.1	**Instance**	See Table 15-15
Description	This register contains the last virtual address of a fault caused by the debugger		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EMUFAULTADDRESS																															

Bits	Field Name	Description	Type	Reset
31:0	EMUFAULTADDRESS	Virtual address of the last emulator access that generated a fault	R	0x0000 0000

Table 15-56. Register Call Summary for Register MMU_EMU_FAULT_AD

MMU Register Manual
- MMU Register Summary: [0][1][2][3][4]
- MMU Register Description: [5]

Table 15-57. MMU_FAULT_PC

Address Offset	0x0000 0080		
Physical Address	See Section 15.5.2.1	**Instance**	See Table 15-15
Description	Typically CPU program counter value of instruction generating MMU fault. The address value is captured at MMU_EMU_FAULT_AD[31:0] EMUFAULTADDRESS. Data-Read-access : corresponding PC. Data-write-access : not perfect accuracy due to posted-write.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PC																															

Bits	Field Name	Description	Type	Reset
31:0	PC	Typically CPU program counter value of instruction generating MMU fault	R	0x0000 0000

Table 15-58. Register Call Summary for Register MMU_FAULT_PC

MMU Register Manual
- MMU Register Summary: [0][1][2][3][4]

Table 15-59. MMU_FAULT_STATUS

Address Offset	0x0000 0084		
Physical Address	See Section 15.5.2.1	**Instance**	See Table 15-15
Description	Fault status register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								MMU_FAULT_TRANS_ID				RD_WR	MMU_FAULT_TYPE		FAULTINDICATION

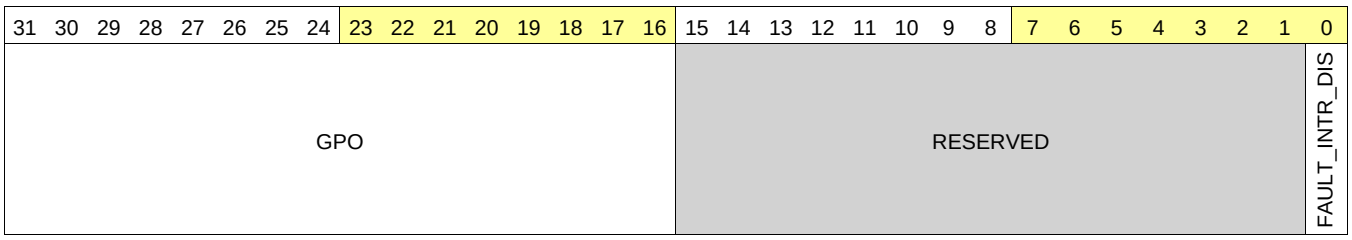
Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reserved	R	0x000000
8:4	MMU_FAULT_TRANS_ID	MtagID of the transaction that caused fault	R	0x0
3	RD_WR	Indicates read or write 0x0: Write 0x1: Read	R	0x0
2:1	MMU_FAULT_TYPE	MReqInfo[1:0] is captured as fault type	R	0x0
0	FAULTINDICATION	Indicates an MMU fault	RW (W1toClr)	0x0

Table 15-60. Register Call Summary for Register MMU_FAULT_STATUS

MMU Register Manual
- MMU Register Summary: [0][1][2][3][4]

Table 15-61. MMU_GPR

Address Offset	0x0000 0088		
Physical Address	See Section 15.5.2.1	**Instance**	See Table 15-15
Description	General purpose register		
Type	RW		



Bits	Field Name	Description	Type	Reset
31:16	GPO	General purpose output sent out as MMU output	RW	0x0000
15:1	RESERVED	Reserved	R	0x0000
0	FAULT_INTR_DIS	Disable generation of interrupt on fault. Error response is returned instead on the slave port	RW	0x0

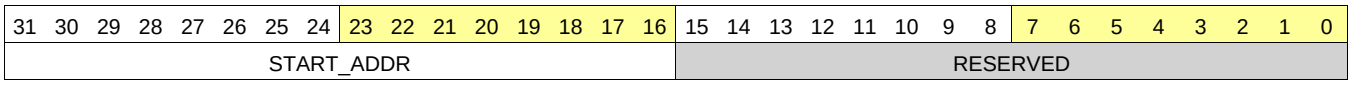
Table 15-62. Register Call Summary for Register MMU_GPR

MMU Functional Description
- MMU Error Handling: [0][1]

MMU Register Manual
- MMU Register Summary: [2][3][4][5][6]

Table 15-63. MMU_BYPASS_REGION1_ADDR

Address Offset	0x0000 0090		
Physical Address	See Section 15.5.2.1	**Instance**	See Table 15-15
Description	This register contains the start address of the first NO TRANSLATION REGION		
Type	RW		



Bits	Field Name	Description	Type	Reset
31:16	START_ADDR	Start address of NO TRANSLATION REGION. This has to be aligned to SIZE in MMU_BYPASS_REGION1_SIZE	RW	0x0
15:0	RESERVED	Reserved	R	0x0

Table 15-64. Register Call Summary for Register MMU_BYPASS_REGION1_ADDR

MMU Register Manual
- MMU Register Summary: [0][1][2][3]

Table 15-65. MMU_BYPASS_REGION1_SIZE

Address Offset	0x0000 0094		
Physical Address	See Section 15.5.2.1	**Instance**	See Table 15-15
Description	This register contains the size of first NO TRANSLATION REGION		

Table 15-65. MMU_BYPASS_REGION1_SIZE (continued)

Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												SIZE			

Bits	Field Name	Description	Type	Reset
31:4	RESERVED	Reserved	R	0x0
3:0	SIZE	Size of the NO TRANSLATION REGION. 0x0 = region not valid 0x1 = 64K bytes 0x2 = 128K bytes 0x3 = 256K bytes 0x4 = 512K bytes 0x5 = 1M bytes 0x6 = 2M bytes 0x7 = 4M bytes 0x8 = 8M bytes 0x9 = 16M bytes 0xA = 32M bytes 0xB = 64M bytes 0xC = 128M bytes 0xD = 256M bytes 0xE = 512M bytes 0xF = 1G bytes	RW	0x0

Table 15-66. Register Call Summary for Register MMU_BYPASS_REGION1_SIZE

MMU Register Manual
- MMU Register Summary: [0][1][2][3]
- MMU Register Description: [4]

Table 15-67. MMU_BYPASS_REGION2_ADDR

Address Offset	0x0000 0098		
Physical Address	See Section 15.5.2.1	**Instance**	See Table 15-15
Description	This register contains the start address of the second NO TRANSLATION REGION		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
START_ADDR																RESERVED															

Bits	Field Name	Description	Type	Reset
31:16	START_ADDR	Start address of NO TRANSLATION REGION. This has to be aligned to SIZE in MMU_BYPASS_REGION2_SIZE.	RW	0x0
15:0	RESERVED	Reserved	R	0x0

Table 15-68. Register Call Summary for Register MMU_BYPASS_REGION2_ADDR

MMU Register Manual
- MMU Register Summary: [0][1][2][3]

Table 15-69. MMU_BYPASS_REGION2_SIZE

Address Offset	0x0000 009C		
Physical Address	See Section 15.5.2.1	**Instance**	See Table 15-15
Description	This register contains the size of second NO TRANSLATION REGION		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												SIZE			

Bits	Field Name	Description	Type	Reset
31:4	RESERVED	Reserved	R	0x0
3:0	SIZE	Size of the NO TRANSLATION REGION. 0x0 = region not valid 0x1 = 64K bytes 0x2 = 128K bytes 0x3 = 256K bytes 0x4 = 512K bytes 0x5 = 1M bytes 0x6 = 2M bytes 0x7 = 4M bytes 0x8 = 8M bytes 0x9 = 16M bytes 0xA = 32M bytes 0xB = 64M bytes 0xC = 128M bytes 0xD = 256M bytes 0xE = 512M bytes 0xF = 1G bytes	RW	0x0

Table 15-70. Register Call Summary for Register MMU_BYPASS_REGION2_SIZE

MMU Register Manual
- MMU Register Summary: [0][1][2][3]
- MMU Register Description: [4][5][6]

Table 15-71. MMU_BYPASS_REGION3_ADDR

Address Offset	0x0000 00A0		
Physical Address	See Section 15.5.2.1	**Instance**	See Table 15-15
Description	This register contains the start address of the third NO TRANSLATION REGION		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
START_ADDR																RESERVED															

Bits	Field Name	Description	Type	Reset
31:16	START_ADDR	Start address of NO TRANSLATION REGION. This has to be aligned to SIZE in MMU_BYPASS_REGION2_SIZE.	RW	0x0
15:0	RESERVED	Reserved	R	0x0

Table 15-72. Register Call Summary for Register MMU_BYPASS_REGION3_ADDR

MMU Register Manual
- MMU Register Summary: [0][1][2][3]

Table 15-73. MMU_BYPASS_REGION3_SIZE

Address Offset	0x0000 00A4		
Physical Address	See Section 15.5.2.1	**Instance**	See Table 15-15
Description	This register contains the size of third NO TRANSLATION REGION		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												SIZE			

Bits	Field Name	Description	Type	Reset
31:4	RESERVED	Reserved	R	0x0
3:0	SIZE	Size of the NO TRANSLATION REGION. 0x0 = region not valid 0x1 = 64K bytes 0x2 = 128K bytes 0x3 = 256K bytes 0x4 = 512K bytes 0x5 = 1M bytes 0x6 = 2M bytes 0x7 = 4M bytes 0x8 = 8M bytes 0x9 = 16M bytes 0xA = 32M bytes 0xB = 64M bytes 0xC = 128M bytes 0xD = 256M bytes 0xE = 512M bytes 0xF = 1G bytes	RW	0x0

Table 15-74. Register Call Summary for Register MMU_BYPASS_REGION3_SIZE

MMU Register Manual
- MMU Register Summary: [0][1][2][3]

Table 15-75. MMU_BYPASS_REGION4_ADDR

Address Offset	0x0000 00A8		
Physical Address	See Section 15.5.2.1	**Instance**	See Table 15-15
Description	This register contains the start address of the fourth NO TRANSLATION REGION		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
START_ADDR																RESERVED															

Bits	Field Name	Description	Type	Reset
31:16	START_ADDR	Start address of NO TRANSLATION REGION. This has to be aligned to SIZE in MMU_BYPASS_REGION2_SIZE.	RW	0x0
15:0	RESERVED	Reserved	R	0x0

Table 15-76. Register Call Summary for Register MMU_BYPASS_REGION4_ADDR

MMU Register Manual
- MMU Register Summary: [0][1][2][3]

Table 15-77. MMU_BYPASS_REGION4_SIZE

Address Offset	0x0000 00AC		
Physical Address	See Section 15.5.2.1	**Instance**	See Table 15-15
Description	This register contains the size of fourth NO TRANSLATION REGION		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												SIZE			

Bits	Field Name	Description	Type	Reset
31:4	RESERVED	Reserved	R	0x0
3:0	SIZE	Size of the NO TRANSLATION REGION. 0x0 = region not valid 0x1 = 64K bytes 0x2 = 128K bytes 0x3 = 256K bytes 0x4 = 512K bytes 0x5 = 1M bytes 0x6 = 2M bytes 0x7 = 4M bytes 0x8 = 8M bytes 0x9 = 16M bytes 0xA = 32M bytes 0xB = 64M bytes 0xC = 128M bytes 0xD = 256M bytes 0xE = 512M bytes 0xF = 1G bytes	RW	0x0

Table 15-78. Register Call Summary for Register MMU_BYPASS_REGION4_SIZE

MMU Register Manual

- MMU Register Summary: [0][1][2][3]

Spinlock

This chapter describes the Spinlock module of the device.

Topic **Page**

16.1 Spinlock Overview

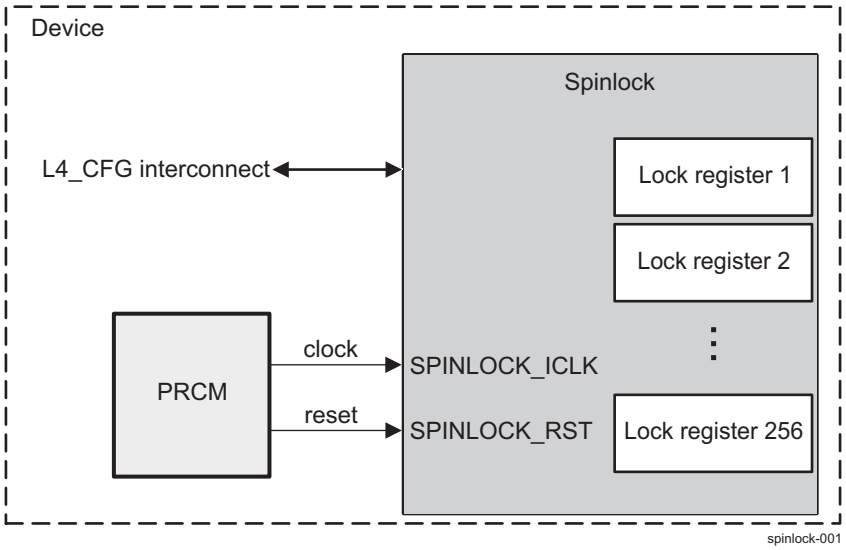
The Spinlock module provides hardware assistance for synchronizing the processes running on multiple processors in the device:

- Digital signal processor (DSP) subsystems – DSP1 and DSP2
- Dual Cortex-M4 image processing unit (IPU) subsystem – IPU

The Spinlock module implements 256 spinlocks (or hardware semaphores), which provide an efficient way to perform a lock operation of a device resource using a single read-access, avoiding the need of a read-modify-write bus transfer that the programmable cores are not capable of.

Figure 16-1 shows an overview of the Spinlock module.

Figure 16-1. Spinlock Overview

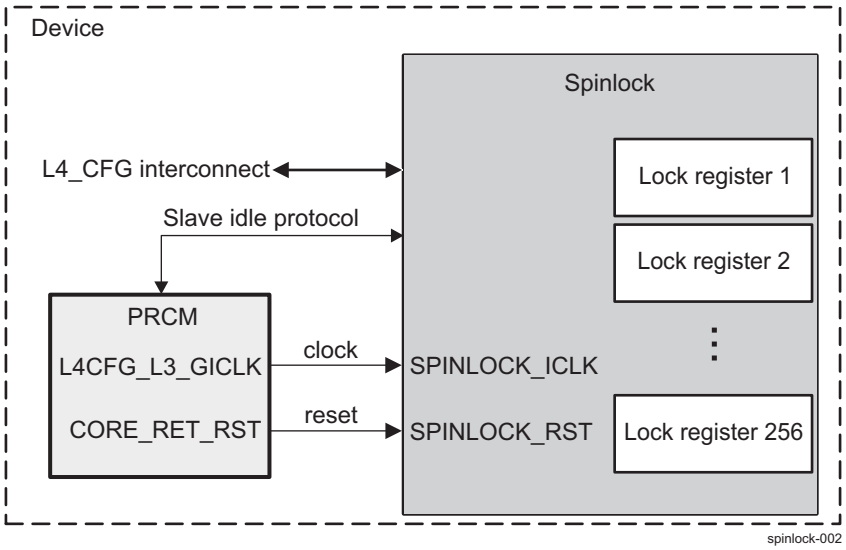


16.2 Spinlock Integration

This section describes module integration in the device, including information about clocks, resets, and hardware requests.

Figure 16-2 shows the Spinlock integration.

Figure 16-2. Spinlock Integration



NOTE: For more information about the Slave idle protocol and the wake-up request, see Section 3.1.1.1.2, *Module-Level Clock Management*, in Chapter 3, *Power, Reset, and Clock Management*.

Table 16-1 and Table 16-2 summarize the integration of the module in the device.

Table 16-1. Spinlock Integration Attributes

Module Instance	Attributes	
	Power Domain	Interconnect
SPINLOCK	PD_COREAON	L4_CFG

Table 16-2. Spinlock Clocks and Resets

Clocks				
Module Instance	**Destination Signal Name**	**Source Signal Name**	**Source**	**Description**
SPINLOCK	SPINLOCK_ICLK	L4CFG_L3_GICLK	PRCM	Spinlock interface/functional clock. This clock is used for all interface and functional operations.
Resets				
Module Instance	**Destination Signal Name**	**Source Signal Name**	**Source**	**Description**
SPINLOCK	SPINLOCK_RST	CORE_RET_RST	PRCM	Spinlock hardware reset. This reset is asynchronously applied to the Spinlock internal registers.

NOTE: The Spinlock module does not support any interrupt and DMA requests.

16.3 Spinlock Functional Description

16.3.1 Spinlock Software Reset

The Spinlock module can be reset by software through the SPINLOCK_SYSCONFIG[1] SOFTRESET bit. Setting this bit to 1 enables an active software reset that is functionally equivalent to a hardware reset. The SPINLOCK_SYSTATUS[0] RESETDONE bit can be polled to check the reset status (reading 1 indicates that reset sequence is done; reading 0 indicates that reset sequence is in progress). The software must ensure that the software reset completes before doing Spinlock operations.

16.3.2 Spinlock Power Management

Table 16-3 describes power-management features available to the Spinlock module.

Table 16-3. Spinlock Local Power Management Features

Feature	Registers	Description
Clock auto gating	SPINLOCK_SYSCONFIG[0] AUTOGATING bit	This bit indicates that the Spinlock module uses an automatic internal interface clock gating strategy, based on interface activity.
Global wake-up enable	SPINLOCK_SYSCONFIG[2] ENAWAKEUP bit	This bit indicates that the wake-up generation feature (at Spinlock module level) is disabled.
Slave idle modes	SPINLOCK_SYSCONFIG[4:3] SIDLEMODE bit field	This bit field indicates that the Spinlock module uses smart-idle mode.

NOTE: All Spinlock local power management features are non-configurable – that is, their respective bit fields are read-only and only show the actual hardware implementation.

For information about source clock gating and sleep/wake-up transitions description, see Section 3.1.1.1.4, *Clock Domain-Level Clock Management*, in Chapter 3, *Power, Reset, and Clock Management*. For descriptions of EnaWakeUp, and IdleMode features, see Section 3.1.1.1.2, *Module-Level Clock Management*, in Chapter 3, *Power, Reset, and Clock Management*.

The Spinlock module is normally idle, except when processing a request from its slave interface port. The smart-idle mode acknowledges idle requests from the PRCM only when the module is prepared to go idle. The Spinlock module is always ready to go idle if it does not have any request that it is processing.

16.3.3 About Spinlocks

Spinlocks are present to solve the need for synchronization and mutual exclusion between heterogeneous processors and those not operating under a single, shared operating system. There is no alternative mechanism to accomplish these operations between processors in separate subsystems.

Spinlocks are not the best way to synchronize between tasks or threads on one CPU. Instead, spinlocks are for use in synchronization between different subsystems in the device that don't have any other means of hardware-based synchronization.

Spinlocks do not solve all system synchronization issues. They have limited applicability and should be used with care to implement higher level synchronization protocols.

A spinlock is appropriate for mutual exclusion for access to a shared data structure. It should be used only when:

1. The time to hold the lock is predictable and small (for example, a maximum hold time of less than 200 CPU cycles may be acceptable).
2. The locking task cannot be preempted, suspended, or interrupted while holding the lock (this would make the hold time large and unpredictable).
3. The lock is lightly contended, that is the chance of any other process (or processor) trying to acquire the lock while it is held is small.

If these conditions are met, then the locking code can retry a failed attempt to acquire the lock until success.

If the conditions are not met, then a spinlock is not a good candidate. One alternative is to use a spinlock for critical section control (engineered to meet the conditions) to implement a higher level semaphore that can support preemption, notification, timeout or other higher level properties.

16.3.4 Spinlock Functional Operation

The Spinlock module supports 256 spinlocks. It accepts only a single command at a time and processes the command fully before accepting the next command. A lock is requested by reading the SPINLOCK_LOCK_REG_i[0] TAKEN bit. There are two states: Taken (SPINLOCK_LOCK_REG_i[0] TAKEN = 1) or Not Taken (SPINLOCK_LOCK_REG_i[0] TAKEN = 0).

When the status of lock *i* (where i = 0 to 255) is Not Taken (free), a read from the SPINLOCK_LOCK_REG_i register returns 0 and sets the lock to Taken (locked). When the status of lock *i* is Taken, a read returns 1 and does not change the state of the lock.

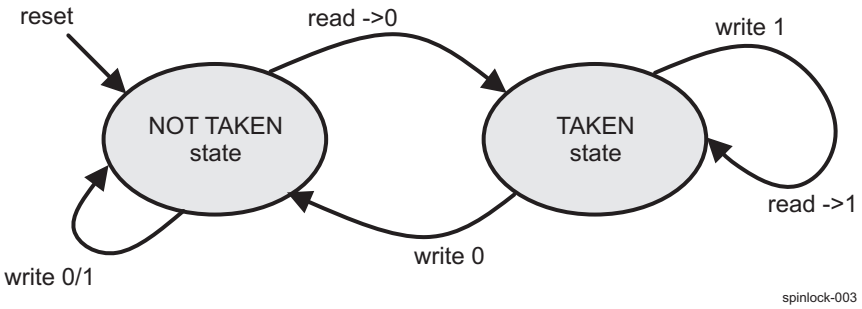
A write to the SPINLOCK_LOCK_REG_i register does not change the state of lock, unless when writing 0 when the lock is in Taken state. By doing this, the requester frees the lock.

CAUTION

Only 32-bit reads and writes are supported.

Figure 16-3 shows the SPINLOCK_LOCK_REG_i register state diagram.

Figure 16-3. Lock Register State Diagram



NOTE:

- There is no support to ensure that a lock register is locked and unlocked by the same process. This must be ensured in software.
- There is no support to check that the same initiator that acquired the lock is the one that is freeing the lock.

16.4 Spinlock Programming Guide

16.4.1 Spinlock Low-level Programming Models

This section covers the low-level hardware programming sequences for configuration and usage of the module.

16.4.1.1 Surrounding Modules Global Initialization

This procedure initializes the surrounding modules when the Spinlock module is used for the first time after a device reset.

Table 16-4. Global Initialization of Surrounding Modules

Surrounding Modules	Comments
PRCM	Spinlock interface clock must be enabled. For more information, see Section 3.6.4.11, *CD_L4_CFG Clock Domain*, in Chapter 3, *Power, Reset, and Clock Management*.
Interconnect	For more information about the L4_CFG interconnect configuration, see Section 9.3, *L4 Interconnects*.

16.4.1.2 Basic Spinlock Operations

The main spinlock operations are:

- Clear all the Taken spinlocks (only after a system bug recovery)
- Take a spinlock
- Release spinlock

16.4.1.2.1 Spinlocks Clearing After a System Bug Recovery

Module initialization (after reset) is not needed, except after system bug recovery. The following table presents the Spinlock initialization after a system bug recovery. Software should store 0 into each of the SPINLOCK_LOCK_REG_i registers at system startup to insure that all locks are initialized to Not Taken.

Table 16-5. Spinlock System Bug Recovery

Step	Register	Value
IF: SPINLOCK_SYSTATUS[0] IU0 = = 1?	SPINLOCK_SYSTATUS[0] IU0	
Free the 256 locks	SPINLOCK_LOCK_REG_i[0] TAKEN (i = 0 to 255)	0x0
END		

16.4.1.2.2 Take and Release Spinlock

This procedure configures the take and release (free) operations for the Spinlock module. A spinlock should only be held with interrupts disabled. So, before attempting to obtain the spinlock, software should disable interrupts. Then it should read the SPINLOCK_LOCK_REG_i[0] TAKEN bit to attempt to obtain the lock. If it succeeds, it should proceed directly through the critical section then unlock and re-enable interrupts. If the acquisition attempt fails, the acquisition should be reattempted. To prevent unknown interrupt disabled time, interrupts should be re-enabled and then disabled before reattempting to acquire the lock. Figure 16-4 shows the described above procedure.

Figure 16-4. Take and Release Spinlock

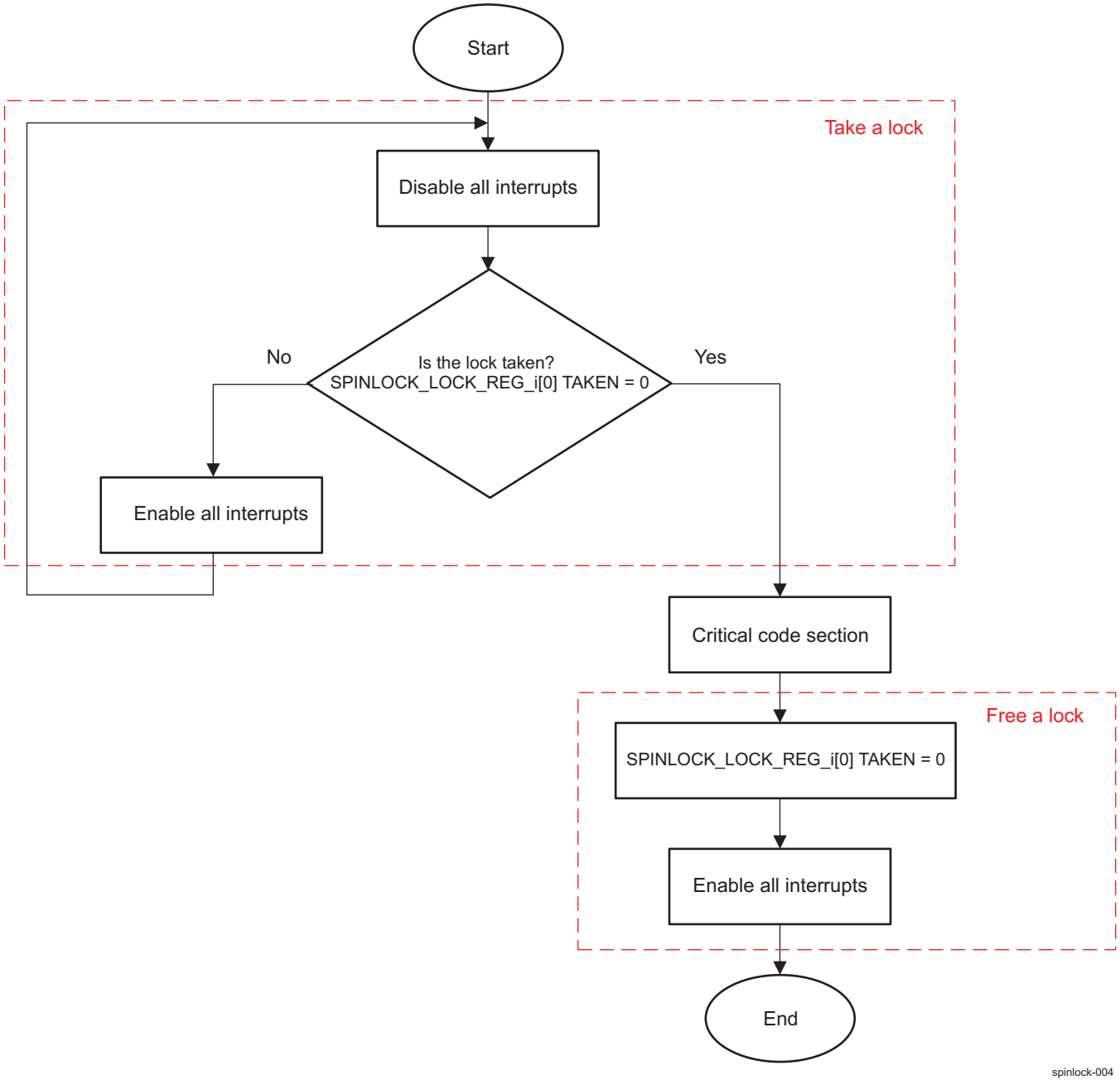


Table 16-6. Register Call Summary

Register Name
SPINLOCK_LOCK_REG_i[0] TAKEN

Table 16-7. Subprocess Call Summary

Subprocess Name	Cross Reference
Disable Interrupts	For information about disabling/enabling interrupts in IPU_INTC, see the Arm *Cortex-M4 Technical Reference Manual* (available at infocenter.arm.com/help/index.jsp). For information about disabling/enabling interrupts in DSP_INTC, see Chapter 4, *DSP Subsystem*.
Enable Interrupts	

16.5 Spinlock Register Manual

16.5.1 Spinlock Instance Summary

Table 16-8. Spinlock Instance Summary

Module Name	L4_CFG Base Address	Size
Spinlock	0x4A0F 6000	4KiB

16.5.2 Spinlock Registers

16.5.2.1 Spinlock Register Summary

Table 16-9. Spinlock Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	L4_CFG Physical Address
SPINLOCK_REVISION	R	32	0x0000 0000	0x4A0F 6000
SPINLOCK_SYSCONFIG	RW	32	0x0000 0010	0x4A0F 6010
SPINLOCK_SYSTATUS	R	32	0x0000 0014	0x4A0F 6014
SPINLOCK_LOCK_REG_i [(1)]	RW	32	0x0000 0800 + (0x4 * i)	0x4A0F 6800 + (0x4 * i)

(1) i = 0 to 255

16.5.2.2 Spinlock Register Description

Table 16-10. SPINLOCK_REVISION

Address Offset	0x0000 0000		
Physical Address	0x4A0F 6000	**Instance**	Spinlock
Description	This register contains the IP revision code		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REVISION																															

Bits	Field Name	Description	Type	Reset
31:0	REVISION	IP revision	R	TI internal data

Table 16-11. Register Call Summary for Register SPINLOCK_REVISION

Spinlock Register Manual
- Spinlock Register Summary: [0]

Table 16-12. SPINLOCK_SYSCONFIG

Address Offset	0x0000 0010		
Physical Address	0x4A0F 6010	**Instance**	Spinlock
Description	This register controls the various parameters of the OCP interface. Note that most fields are read-only.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																											SIDLEMODE			ENWAKEUP	SOFTRESET	AUTOGATING

Bits	Field Name	Description	Type	Reset
31:5	RESERVED	Reserved. Reads return 0.	R	0x0000000
4:3	SIDLEMODE	Slave interface power management (IDLE request/acknowledgement control). Read 0x0: Force-idle. IDLE request is acknowledged unconditionally and immediately. Read 0x1: No-idle. IDLE request is never acknowledged. Read 0x2: Smart-idle. IDLE request acknowledgement is based on the internal module activity. Read 0x3: Reserved. Do not use.	R	0x2
2	ENWAKEUP	Asynchronous wakeup gereration. Read 0x0: Wakeup generation is disabled. Read 0x1: Wakeup generation is enabled.	R	0
1	SOFTRESET	Module software reset. Write 0x0: No action Write 0x1: Start soft reset sequence	W	0
0	AUTOGATING	Internal interface clock gating strategy. Read 0x0: Interface clock is not gated when the interface is idle. Read 0x1: Automatic internal OCP clock gating strategy is applied, based on the OCP interface activity.	R	1

Table 16-13. Register Call Summary for Register SPINLOCK_SYSCONFIG

Spinlock Functional Description
- Spinlock Software Reset: [0]
- Spinlock Power Management: [1][2][3]

Spinlock Register Manual
- Spinlock Register Summary: [4]

Table 16-14. SPINLOCK_SYSTATUS

Address Offset	0x0000 0014		
Physical Address	0x4A0F 6014	**Instance**	Spinlock
Description	This register provides status information about this instance of the Spinlock module.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NUMLOCKS								RESERVED								IU7	IU6	IU5	IU4	IU3	IU2	IU1	IU0	RESERVED							RESETDONE

Bits	Field Name	Description	Type	Reset
31:24	NUMLOCKS	Number of lock registers implemeted. Read 0x1: This instance has 32 lock registers. Read 0x2: This instance has 64 lock registers. Read 0x4: This instance has 128 lock registers. Read 0x8: This instance has 256 lock registers.	R	0x08
23:16	RESERVED	Reserved. Reads return 0.	R	0x00
15	IU7	In-Use flag 0, covering lock registers 224 - 255. Read 0x0: All lock registers 224 - 255 are in the Not Taken state. Read 0x1: At least one of the lock registers 224 - 255 is in the Taken state.	R	0
14	IU6	In-Use flag 0, covering lock registers 192 - 223. Read 0x0: All lock registers 192 - 223 are in the Not Taken state. Read 0x1: At least one of the lock registers 192 - 223 is in the Taken state.	R	0
13	IU5	In-Use flag 0, covering lock registers 160 - 191. Read 0x0: All lock registers 160 - 191 are in the Not Taken state. Read 0x1: At least one of the lock registers 160 - 191 is in the Taken state.	R	0
12	IU4	In-Use flag 0, covering lock registers 128 - 159. Read 0x0: All lock registers 128 - 159 are in the Not Taken state. Read 0x1: At least one of the lock registers 128 - 159 is in the Taken state.	R	0
11	IU3	In-Use flag 0, covering lock registers 96 - 127. Read 0x0: All lock registers 96 - 127 are in the Not Taken state. Read 0x1: At least one of the lock registers 96 - 127 is in the Taken state.	R	0
10	IU2	In-Use flag 0, covering lock registers 64 - 95. Read 0x0: All lock registers 64 - 95 are in the Not Taken state. Read 0x1: At least one of the lock registers 64 - 95 is in the Taken state.	R	0
9	IU1	In-Use flag 0, covering lock registers 32 - 63. Read 0x0: All lock registers 32 - 63 are in the Not Taken state. Read 0x1: At least one of the lock registers 32 - 63 is in the Taken state.	R	0

Bits	Field Name	Description	Type	Reset
8	IU0	In-Use flag 0, covering lock registers 0 - 31. Read 0x0: All lock registers 0 - 31 are in the Not Taken state. Read 0x1: At least one of the lock registers 0 - 31 is in the Taken state.	R	0
7:1	RESERVED	Reserved. Reads return 0.	R	0x00
0	RESETDONE	Reset done status. Read 0x0: Reset in progress. Read 0x1: Reset is completed.	R	1

Table 16-15. Register Call Summary for Register SPINLOCK_SYSTATUS

Spinlock Functional Description • Spinlock Software Reset: [0]
Spinlock Programming Guide • Basic Spinlock Operations: [1][2]
Spinlock Register Manual • Spinlock Register Summary: [3]

Table 16-16. SPINLOCK_LOCK_REG_i

Address Offset	0x0000 0800	**index**	i = 0 to 255
Physical Address	0x4A0F 6800 + (0x4 * i)	**Instance**	Spinlock
Description	This register contains the state of one lock.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															TAKEN

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved. Reads return 0. Writes are ignored.	R	0x0000 0000
0	TAKEN	Lock State Read 0x0: Lock was previously Not Taken (free). The requester is granted the lock. Write 0x0: Set the lock to Not Taken (free). Read 0x1: Lock was previously Taken. The requester is not granted the lock and must retry. Write 0x1: No update to the lock value.	RW	0

Table 16-17. Register Call Summary for Register SPINLOCK_LOCK_REG_i

Spinlock Functional Description • Spinlock Functional Operation: [0][1][2][3][4][5]
Spinlock Programming Guide • Basic Spinlock Operations: [6][7][8][9]
Spinlock Register Manual • Spinlock Register Summary: [10]

Timers

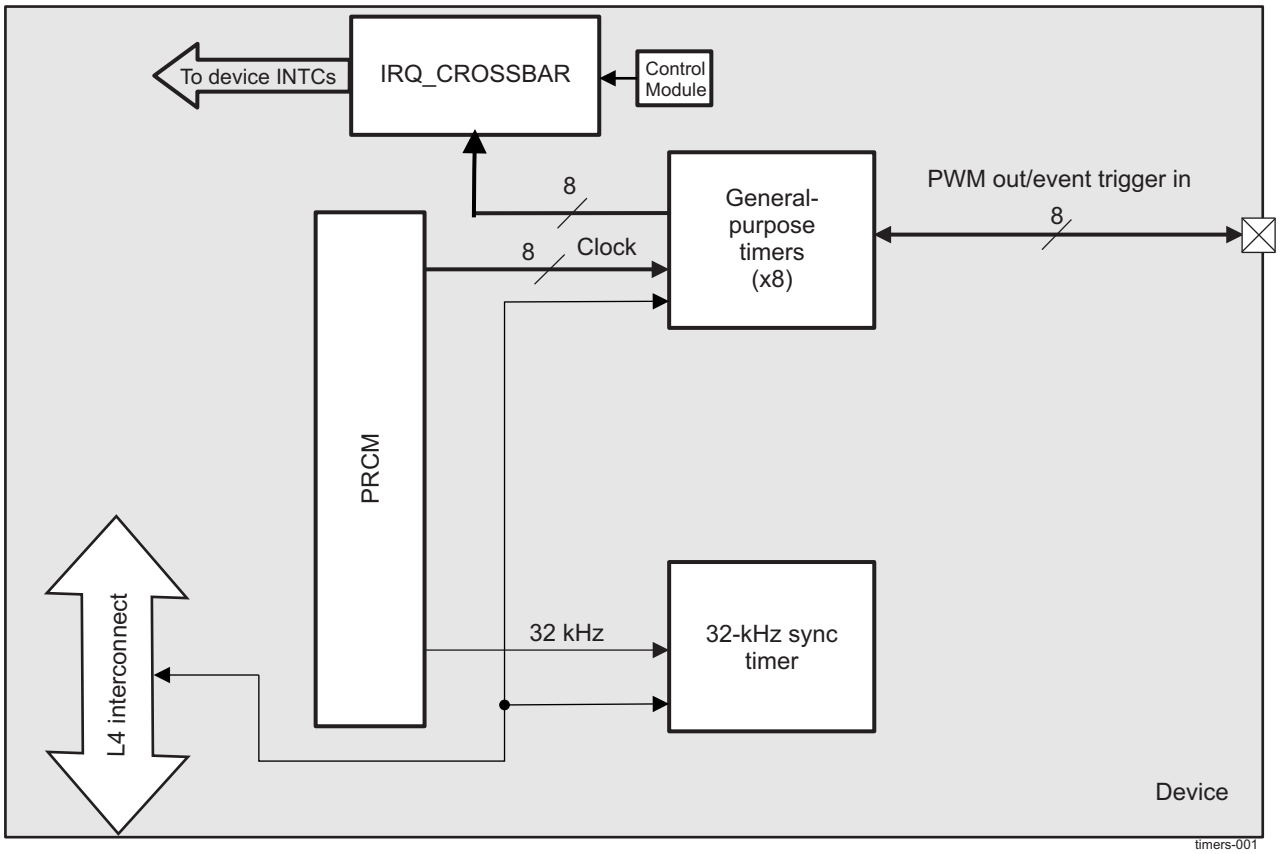
This chapter describes the timer modules for the device.

Topic **Page**

17.1 Timers Overview

The device includes several types of timers used by the system software, including eight general-purpose (GP) timers, and a 32-kHz synchronized timer (COUNTER_32K). Figure 17-1 shows a high-level block diagram of the device timers.

Figure 17-1. Timers Overview



The 32-kHz sync timer, which is reset only at power up, provides the operating system (OS) with a stable timing source that stores the relative time since the last power cycle of the product. The eight GP timers, which are useful as basic timers, are included to generate time-stamp-based interrupts to the system software or to use as a source of pulse-width modulation (PWM) signals.

17.2 General-Purpose Timers

17.2.1 General-Purpose Timers Overview

The device has eight GP timers: TIMER1 through TIMER8.

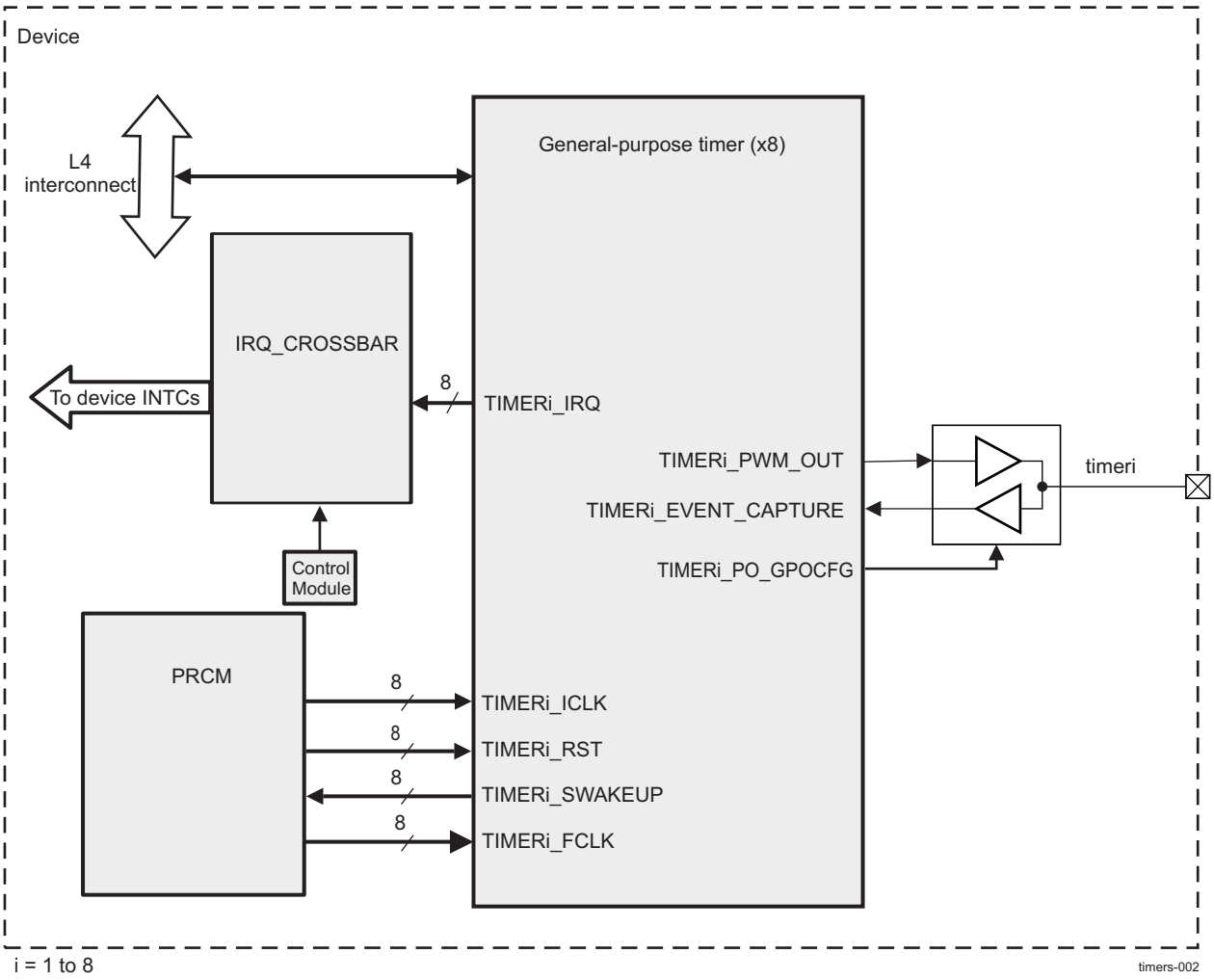
- TIMER1 (1-ms tick) includes a specific function to generate accurate tick interrupts to the operating system and it belongs to the PD_WKUPAON domain.
- TIMER2 through TIMER8 belong to the PD_COREAON module.

Each timer can be clocked from the system clock (19.2, 20, or 27 MHz) or the 32-kHz clock. Select the clock source at the power, reset, and clock management (PRCM) module level.

Each timer provides an interrupt through the device IRQ_CROSSBAR.

Each timer is connected to an external pin by their PWM output or their event capture input pin (for external timer triggering). Figure 17-2 shows an overview of the GP timers.

Figure 17-2. GP Timers Overview



17.2.1.1 GP Timer Features

The following are the main features of the GP timer controllers:

- Level 4 (L4) slave interface support:
  - 32-bit data bus width

  - 32- or 16-bit access supported
  - 8-bit access not supported
  - 10-bit address bus width
  - Burst mode not supported
  - Write nonposted transaction mode supported
- Interrupts generated on overflow, compare, and capture
- Free-running 32-bit upward counter
- Compare and capture modes
- Autoreload mode
- Start and stop mode
- Programmable divider clock source (2^n, where n = [0:8])
- Dedicated input trigger for capture mode and dedicated output trigger/PWM signal
- Dedicated GP output signal for using the TIMERi_GPO_CFG signal
- On-the-fly read/write register (while counting)
- 1-ms tick with 32.768-Hz functional clock generated (only TIMER1)

17.2.2 GP Timer Environment

17.2.2.1 GP Timer External System Interface

Each timer can send or receive stimulus to and from the external (off-chip) system. In the device all timers are configured to output a PWM pulse or receive an external event signal used as a trigger to capture the current timer count.

Figure 17-3 shows the external system interface for the GP timers, and Table 17-1 describes the GP timer inputs and outputs.

NOTE: Software control must ensure that MUX mode is configured to select the timeri (where i = 1 to 8) signal on only one pad. Other pads on which the same signal is multiplexed must be configured in safe mode or non-dmtimer mode to avoid two different pads driving the same signal.

For more information about the configuration of the timeri I/O pads, see Section 13.4.6.1, Pad Configuration Registers.

Figure 17-3. GP Timers External System Interface

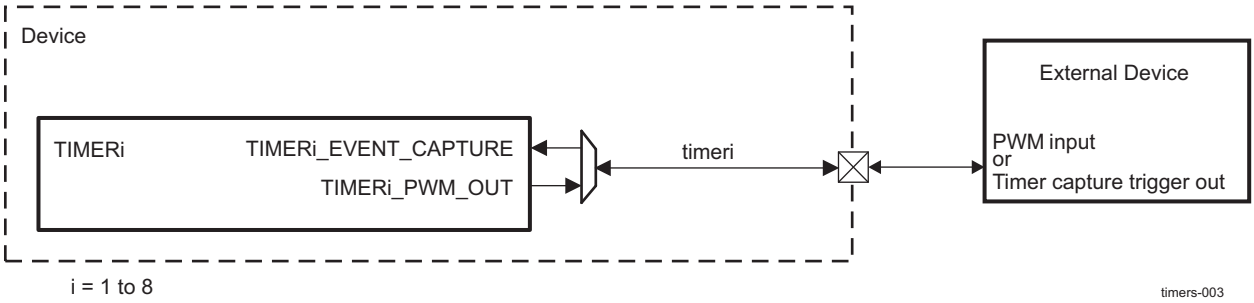


Table 17-1. Input/Output Description

Pin Name	Type (1)	Reset Value	Signal Name	Description
timer1	I/O	0	TIMER1_PI_EVENT_CAPTURE TIMER1_PWM_OUT	TIMER1 trigger input/ PWM output
timer2	I/O	0	TIMER2_PI_EVENT_CAPTURE TIMER2_PWM_OUT	TIMER2 trigger input/ PWM output
timer3	I/O	0	TIMER3_PI_EVENT_CAPTURE TIMER3_PWM_OUT	TIMER3 trigger input/ PWM output
timer4	I/O	0	TIMER4_PI_EVENT_CAPTURE TIMER4_PWM_OUT	TIMER4 trigger input/ PWM output
timer5	I/O	0	TIMER5_PI_EVENT_CAPTURE TIMER5_PWM_OUT	TIMER5 trigger input/ PWM output
timer6	I/O	0	TIMER6_PI_EVENT_CAPTURE TIMER6_PWM_OUT	TIMER6 trigger input/ PWM output
timer7	I/O	0	TIMER7_PI_EVENT_CAPTURE TIMER7_PWM_OUT	TIMER7 trigger input/ PWM output
timer8	I/O	0	TIMER8_PI_EVENT_CAPTURE TIMER8_PWM_OUT	TIMER8 trigger input/ PWM output

(1) When configured for that function; I = Input, O = Output

NOTE: Each TIMERi_PO_GPOCFG signal is used to as an output enable to control the function of the TIMERi pin (where i = 1 to 8) as the PWM output (PO_GPOCFG =0) or capture input (PO_GPOCFG = 1).

17.2.3 GP Timer Integration

Figure 17-4 shows the integration of the GP timer in the device.

Figure 17-4. GP Timer Integration

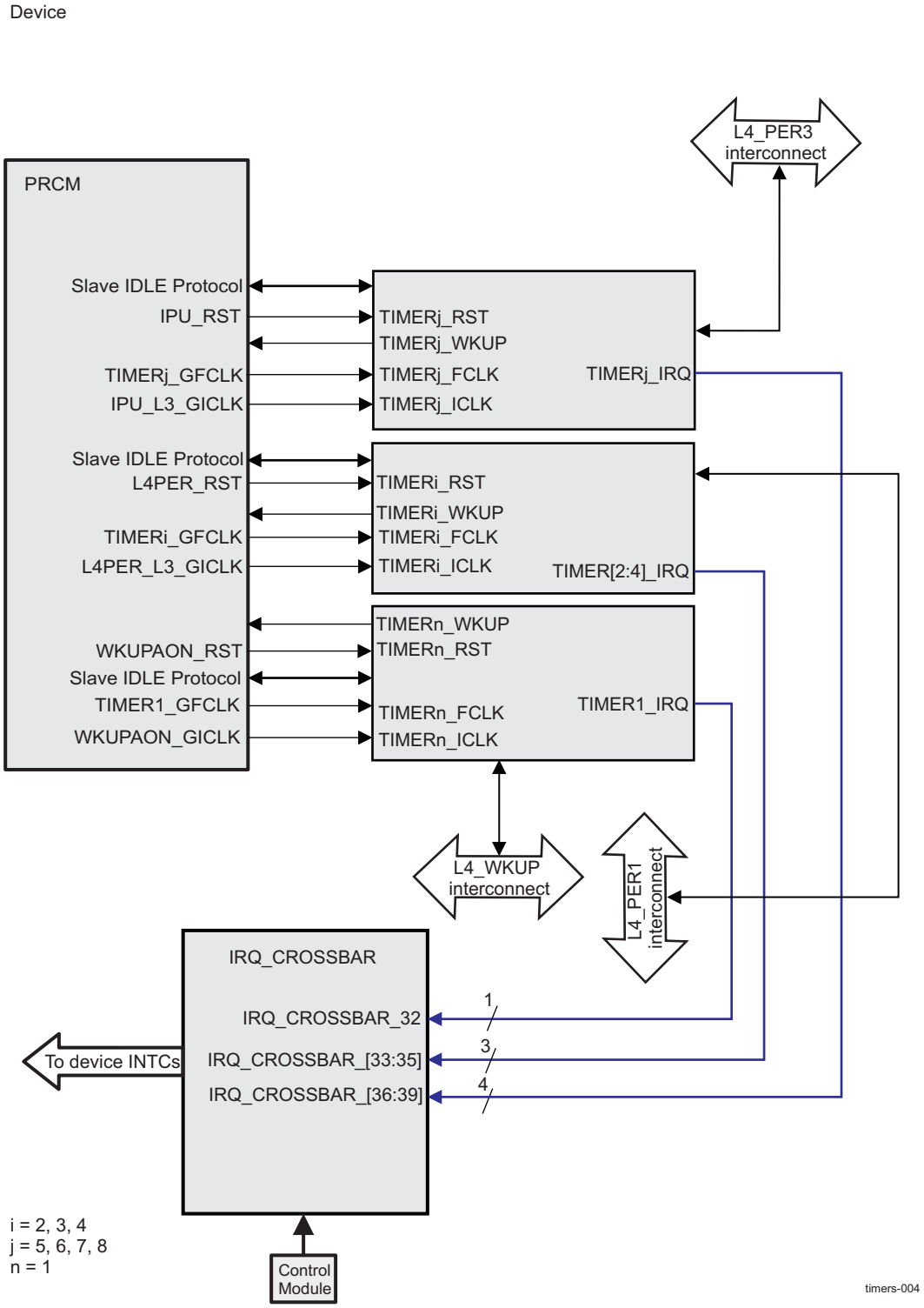


Table 17-2 through Table 17-4 summarize the integration of the module in the device.

Table 17-2. Integration Attributes

Module Instance	Attributes		
	Power Domain	Wake-Up Capability	Interconnect
TIMER1	PD_WKUPAON	Yes	L4_WKUP
TIMER2	PD_COREAON	Yes	L4_PER1
TIMER3	PD_COREAON	Yes	L4_PER1
TIMER4	PD_COREAON	Yes	L4_PER1
TIMER5	PD_COREAON	Yes	L4_PER3
TIMER6	PD_COREAON	Yes	L4_PER3
TIMER7	PD_COREAON	Yes	L4_PER3
TIMER8	PD_COREAON	Yes	L4_PER3

Table 17-3. Clocks and Resets

Clocks				
Module Instance	**Destination Signal Name**	**Source Signal Name**	**Source**	**Description**
TIMER1	TIMER1_FCLK	TIMER1_GFCLK	PRCM	TIMER1 functional clock
	TIMER1_ICLK	WKUPAON_GICLK	PRCM	TIMER1 interface clock
TIMER2	TIMER2_FCLK	TIMER2_GFCLK	PRCM	TIMER2 functional clock
	TIMER2_ICLK	L4PER_L3_GICLK	PRCM	TIMER2 interface clock
TIMER3	TIMER3_FCLK	TIMER3_GFCLK	PRCM	TIMER3 functional clock
	TIMER3_ICLK	L4PER_L3_GICLK	PRCM	TIMER3 interface clock
TIMER4	TIMER4_FCLK	TIMER4_GFCLK	PRCM	TIMER4 functional clock
	TIMER4_ICLK	L4PER_L3_GICLK	PRCM	TIMER4 interface clock
TIMER5	TIMER5_FCLK	TIMER5_GFCLK	PRCM	TIMER5 functional clock
	TIMER5_ICLK	IPU_L3_GICLK	PRCM	TIMER5 interface clock
TIMER6	TIMER6_FCLK	TIMER6_GFCLK	PRCM	TIMER6 functional clock
	TIMER6_ICLK	IPU_L3_GICLK	PRCM	TIMER6 interface clock
TIMER7	TIMER7_FCLK	TIMER7_GFCLK	PRCM	TIMER7 functional clock
	TIMER7_ICLK	IPU_L3_GICLK	PRCM	TIMER7 interface clock
TIMER8	TIMER8_FCLK	TIMER8_GFCLK	PRCM	TIMER8 functional clock
	TIMER8_ICLK	IPU_L3_GICLK	PRCM	TIMER8 interface clock
Resets				
TIMER1	TIMER1_RST	WKUPAON_RST	PRM	Reset to TIMER1
TIMER2	TIMER2_RST	L4PER_RST	PRM	Reset to TIMER2
TIMER3	TIMER3_RST	L4PER_RST	PRM	Reset to TIMER3
TIMER4	TIMER4_RST	L4PER_RST	PRM	Reset to TIMER4
TIMER5	TIMER5_RST	IPU_RST	PRM	Reset to TIMER5
TIMER6	TIMER6_RST	IPU_RST	PRM	Reset to TIMER6
TIMER7	TIMER7_RST	IPU_RST	PRM	Reset to TIMER7
TIMER8	TIMER8_RST	IPU_RST	PRM	Reset to TIMER8

Table 17-4. GP Timers Hardware Requests

Interrupt Requests				
Module Instance	**Source Signal Name**	**Destination IRQ_CROSSBAR Input**	**Default Mapping**	**Description**
TIMER1	TIMER1_IRQ	IRQ_CROSSBAR_32	DSP1_IRQ_63 DSP2_IRQ_63	TIMER1 interrupt

Table 17-4. GP Timers Hardware Requests (continued)

TIMER2	TIMER2_IRQ	IRQ_CROSSBAR_33	DSP1_IRQ_64	TIMER2 interrupt
			DSP2_IRQ_64	
TIMER3	TIMER3_IRQ	IRQ_CROSSBAR_34	DSP1_IRQ_65	TIMER3 interrupt
			DSP2_IRQ_65	
			IPU1_IRQ_53	
TIMER4	TIMER4_IRQ	IRQ_CROSSBAR_35	DSP1_IRQ_66	TIMER4 interrupt
			DSP2_IRQ_66	
			IPU1_IRQ_54	
TIMER5	TIMER5_IRQ	IRQ_CROSSBAR_36	DSP1_IRQ_67	TIMER5 interrupt
			DSP2_IRQ_67	
TIMER6	TIMER6_IRQ	IRQ_CROSSBAR_37	DSP1_IRQ_68	TIMER6 interrupt
			DSP2_IRQ_68	
TIMER7	TIMER7_IRQ	IRQ_CROSSBAR_38	DSP1_IRQ_69	TIMER7 interrupt
			DSP2_IRQ_69	
TIMER8	TIMER8_IRQ	IRQ_CROSSBAR_39	DSP1_IRQ_70	TIMER8 interrupt
			DSP2_IRQ_70	
No DMA Requests				

NOTE: The Default Mapping column in Table 17-4 shows the default mapping of module IRQ source signals. These IRQ source signals can also be mapped to other lines of each device interrupt controller through the IRQ_CROSSBAR module.
For more information about the IRQ_CROSSBAR module, see Section 13.4.6.4, *IRQ_CROSSBAR Module Functional Description*, in Chapter 13, *Control Module*.
For more information about the device interrupt controllers, see Chapter 12, *Interrupt Controllers*.

NOTE: For the description of the interrupt source, see Section 17.2.4.5, *GP Timer Interrupt*.

17.2.4 GP Timer Functional Description

Each GP timer contains a free-running upward counter with autoreload capability on overflow. The timer counter can be read and written on-the-fly (while counting). Each GP timer includes compare logic to allow an interrupt event on a programmable counter matching value. A dedicated output signal can be pulsed or toggled on either an overflow or a match event. This offers time-stamp trigger signaling or PWM signal sources. A dedicated input signal can be used to trigger an automatic timer counter capture or an interrupt event on a programmable input signal transition. A programmable clock divider (prescaler) allows reduction of the timer input clock frequency. All internal timer interrupt sources are merged into one module interrupt line and one wake-up line.

Each internal interrupt source can be independently enabled and disabled by a dedicated bit in the IRQSTATUS_SET and IRQSTATUS_CLR registers for the interrupt features, and a dedicated bit of the IRQWAKEEN register for the wake-up of TIMER1. In addition, a mechanism implemented in these timers generate an accurate tick interrupt.

For all other internal interrupts, the source can be independently enabled and disabled through the IRQENABLE_SET and IRQENABLE_CLR registers.

For each GP timer implemented in the device, there are two possible clock sources:

- 32-kHz clock
- System clock

The input clock source is selected in the registers in the PRCM configuration (see Section 17.2.1, *GP Timer Overview*).

Each GP timer supports three functional modes:

- Timer mode
- Capture mode
- Compare mode

The capture and compare modes are disabled by default after core reset.

17.2.4.1 GP Timer Block Diagram

Figure 17-5 is a block diagram of the common GP timers, and Figure 17-6 is a block diagram of the GP timers with a 1-ms tick generation module.

Figure 17-5. Block Diagram of TIMER2 Through TIMER8

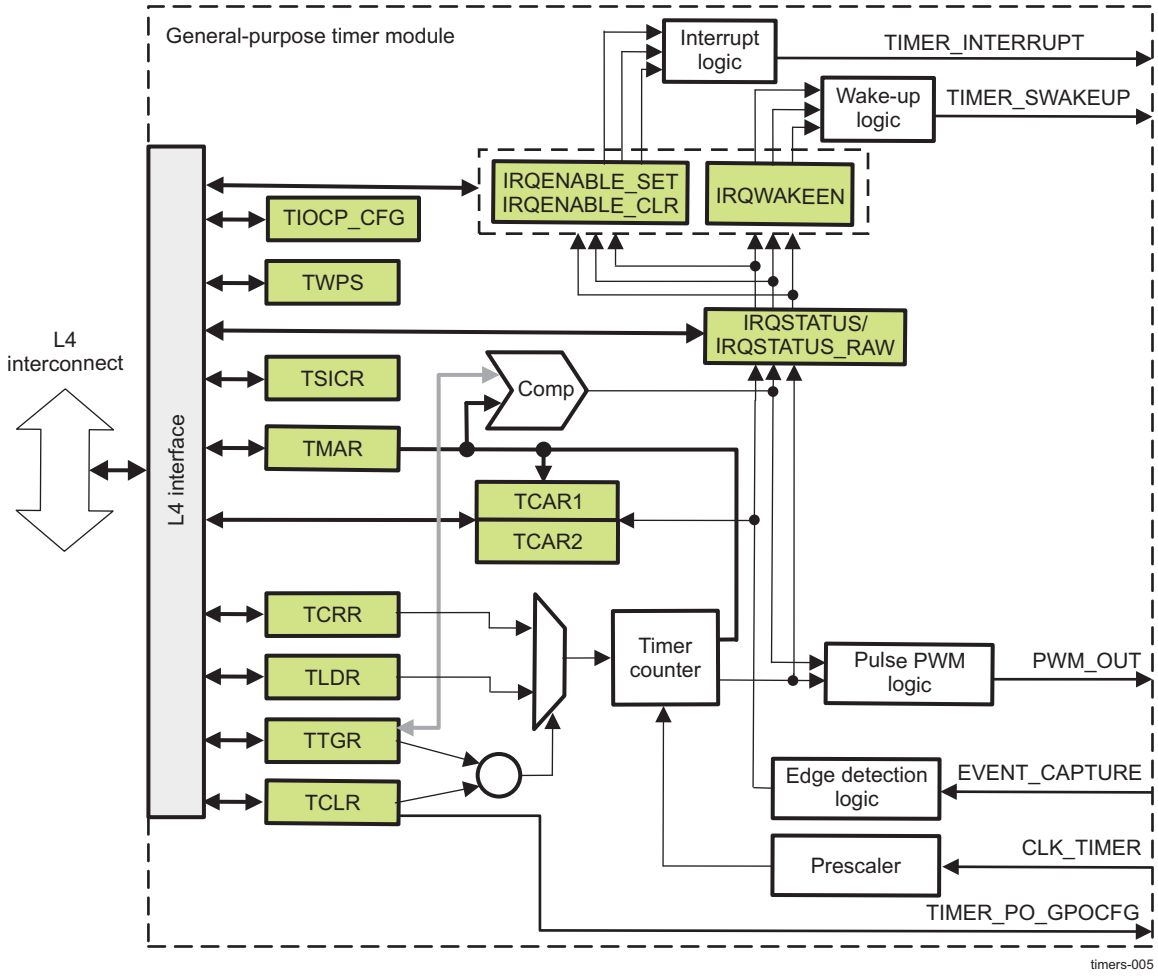
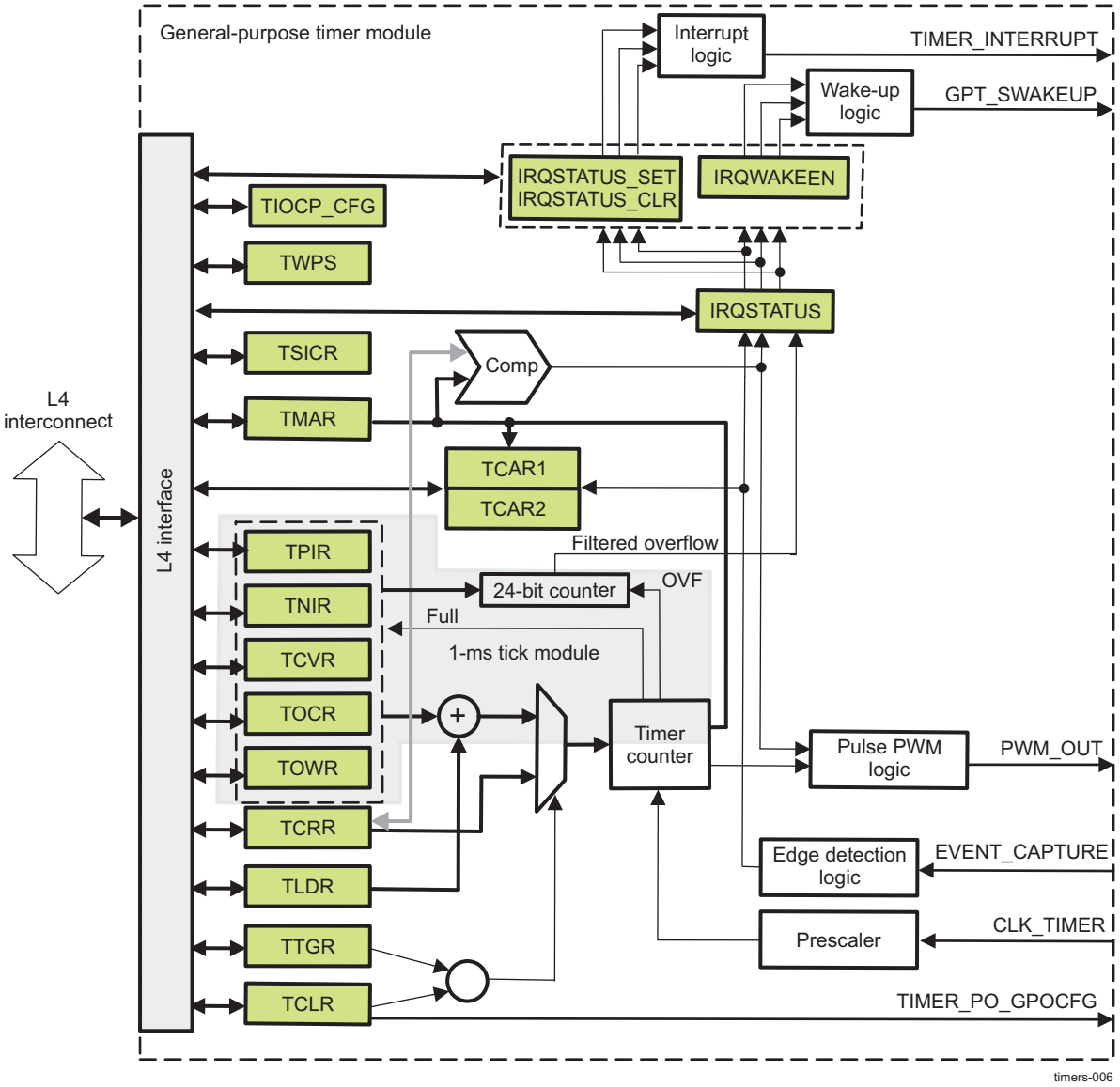


Figure 17-6. Block Diagram of TIMER1



17.2.4.2 TIMER1 Power Management

At the PRCM module level, when all conditions to shut off the functional or interface output clocks in the PRCM module are met (see Section 3.1.1.1.4, *Clock Domain-Level Clock Management*), the PRCM module automatically launches a hardware handshake protocol to ensure the GP timer is ready to have its clocks switched off. Namely, the PRCM module asserts an IDLE request to the GP timer.

Although this handshake is a hardware function and is out of software control, the way the GP timer acknowledges the PRCM IDLE request is configurable through the TIOCP_CFG[3:2] IDLEMODE bit field.

Table 17-5 lists the IDLEMODE settings and the related acknowledgment modes.

Table 17-5. IDLEMODE Settings

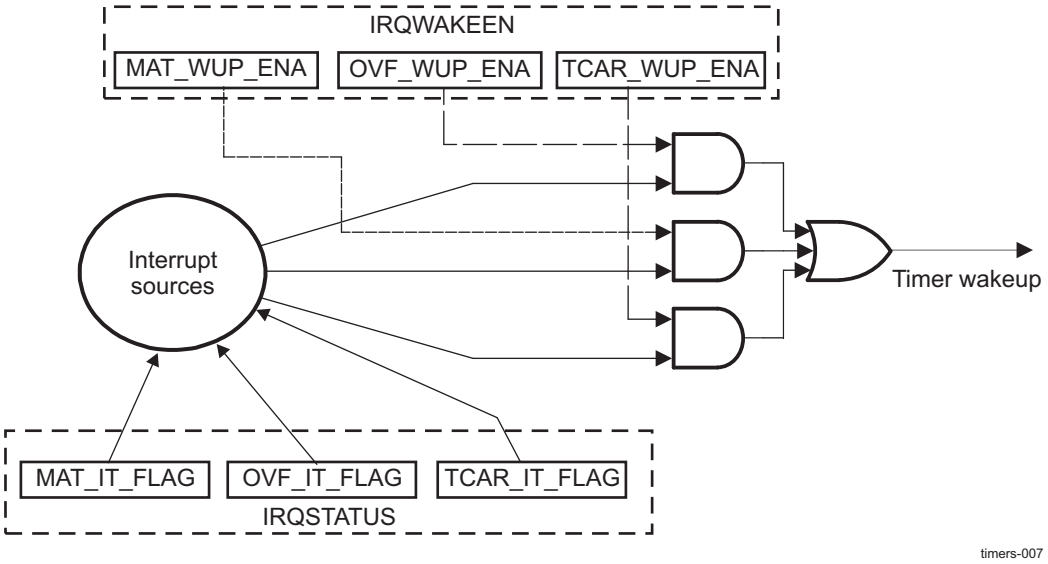
IDLEMODE Value	Selected Mode	Description
00	Force-idle	The GP timer unconditionally acknowledges the IDLE request from the PRCM module, regardless of its internal operations. This mode must be used carefully, because it does not prevent the loss of data when the clock is switched off.
01	No-idle	The GP timer never acknowledges an IDLE request from the PRCM module. This mode is safe from a module point of view, because it ensures that the clocks remain active. No-idle mode is not efficient from a power-saving perspective, because it does not allow the PRCM output clock to be shut off, and thus the power domain to be set to a lower power state.
10	Smart-idle	The GP timer acknowledges the IDLE request, basing its decision on its internal activity. The acknowledge signal is asserted only when all pending transactions and IRQ requests are treated. This is the best approach to efficient system power management.
11	Smart-idleWakeup	The module behaves like in smart-idle mode, with the exception, that it can issue a wake-up request in sleep mode, if the functional clock is not cut off.

17.2.4.2.1 Wake-Up Capability

If the TIOCP_CFG[3:2] IDLEMODE bit field sets the smart-idle mode or smart-idle with wakeup mode, the timer evaluates its internal capability to have the interface clock switched off. When there is no further internal activity (no pending interrupt sources: match, overflow, or timer capture events), the idle acknowledge signal is asserted and the timer enters sleep mode, ready to issue a wake-up request if is configured in smart-idle with wakeup mode.

Figure 17-7 shows the wake-up request generation. For more information about the GP timer clock control, see Section 3.6, *Clock Management Functional Description*, in Chapter 3, *Power, Reset and Clock Management*.

Figure 17-7. Wake-Up Request Generation



For TIMER1 the timer wake-up-enable register allows masking of the expected source of the wake-up event that generates a wake-up request. The register is synchronously programmed with the interface clock before the PRCM module sends an idle mode request. The expected source of the wake-up event is an overflow (TCRR), a timer match (the compare result of TCRR and TMAR matches the counter value), and a timer capture (detection of an external pulse transition of the correct polarity on the TIMER_EVENT_CAPTURE).

When the wake-up event is issued, the associated interrupt status bit is set in the timer status register (IRQSTATUS). The pending wake-up event is reset when the set status bit is overwritten with 1.

NOTE: The status bit must be reset to re-enter idle mode.

17.2.4.3 Power Management of Other GP Timers

At the PRCM module level, when all conditions to shut off the functional or interface output clocks of the PRCM module are met (see Section 3.1.1.1.4, *Clock Domain-Level Clock Management*), the PRCM module automatically launches a hardware handshake protocol to ensure the GP timer is ready to have its clocks switched off. Namely, the PRCM module asserts an IDLE request to the GP timer.

Although this handshake is a hardware function and is out of software control, the way the GP timer acknowledges the PRCM IDLE request is configurable through the TIOCP_CFG[3:2] IDLEMODE bit field.

Table 17-6 lists the IDLEMODE settings and the related acknowledgment modes.

Table 17-6. IDLEMODE Settings

IDLEMODE Value	Selected Mode	Description
00	Force-idle	The GP timer unconditionally acknowledges the IDLE request from the PRCM module, regardless of its internal operations. This mode must be used carefully, because it does not prevent the loss of data when the clock is switched off.
01	No-idle	The GP timer never acknowledges an IDLE request from the PRCM module. This mode is safe from a module point of view, because it ensures that the clocks remain active. INo-idle mode is not efficient from a power-saving perspective, however, because it does not allow the PRCM output clock to be shut off and thus the power domain to be set to a lower power state.

Table 17-6. IDLEMODE Settings (continued)

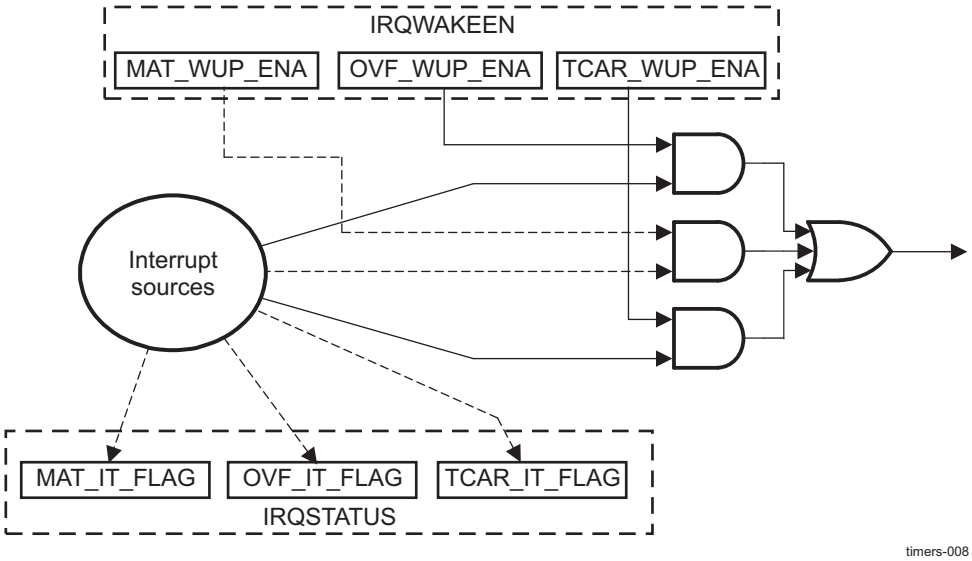
IDLEMODE Value	Selected Mode	Description
10	Smart-idle	The GP timer acknowledges the IDLE request, basing its decision on its internal activity. The acknowledge signal is asserted only when all pending transactions and IRQ requests are treated. This is the best approach to efficient system power management.
11	Smart-idleWakeup	The module behaves like in Smart-idle mode, with the exception, that it can issue a wake-up request in sleep mode, if the functional clock is not cut off.

17.2.4.3.1 Wake-Up Capability

If the TIOCP_CFG[3:2] IDLEMODE bit field sets the smart-idle mode or smart-idle with wakeup mode, the timer evaluates its internal capability to have the interface clock switched off. When there is no further internal activity (no pending interrupt sources: match, overflow, or timer capture events), the idle acknowledge signal is asserted and the timer enters sleep mode, ready to issue a wake-up request if is configured in smart-idle with wakeup mode. This wake-up request is sent only if the IRQWAKEEN[2:0] bit field enables the timer wake-up capability.

Figure 17-8 shows the wake-up request generation. For more information about the GP timer clock control, see Section 3.6, *Clock Management Functional Description*, in Chapter 3, *Power, Reset, and Clock Management*.

Figure 17-8. Wake-Up Request Generation



When the wake-up event is issued, the associated interrupt status bit is set in the timer status register (IRQSTATUS). The pending wake-up event is reset when the set status bit is overwritten with 1.

NOTE: The status bit must be reset to re-enter idle mode.

17.2.4.4 Software Reset

Two bits can generate a software reset of the GP timer:

- TIOCP_CFG[0] SOFTRESET
- TSICR[1] SFT

For both bits, all read accesses return 0.

The TIOCP_CFG[0] SOFTRESET bit allows resetting of the functional and interface domains. The TSICR[1] SFT bit allows resetting the functional part of the GP timer.

Before accessing or using the GP timer, the local host must ensure that both internal resets are released by reading the TIOCP_CFG[0] SOFTRESET bit. This bit monitors the internal reset status.

17.2.4.5 GP Timer Interrupts

The timer can issue an overflow interrupt, a timer match interrupt, and a timer capture interrupt. Each internal interrupt source can be independently enabled and disabled in the interrupt-enable register (IRQSTATUS_SET for TIMER1 and IRQENABLE_SET for other timers) and disabled in the interrupt-disable register (IRQSTATUS_CLR for TIMER1 and IRQENABLE_CLR for other timers). When the interrupt event is issued, the associated interrupt status bit is set in the timer status register (IRQSTATUS).

17.2.4.6 Timer Mode Functionality

The timer is an upward counter that can be started and stopped at any time through the timer control register (the TCLR[0] ST bit). The timer counter register (TCRR) can be loaded when stopped or on-the-fly (while counting). TCRR can be loaded directly by a TCRR write access with a new timer value. TCRR can also be loaded with the value held in the timer load register (TLDR) by a trigger register (TTGR) write access. The loading of TCRR is done regardless of the written value of TTGR. The value of TCRR can be read when stopped or captured on-the-fly by a TCRR read access. The timer is stopped and the counter value is set to 0 when the module reset is asserted. The timer is maintained at stop after the reset is released.

In one-shot mode (the TCLR[1] AR bit is set to 0), the counter is stopped after counting overflow occurs (the counter value remains at 0).

When the autoreload mode is enabled (the TCLR[1] AR bit is set to 1), TCRR is reloaded with the value of TLDR after a counting overflow occurs.

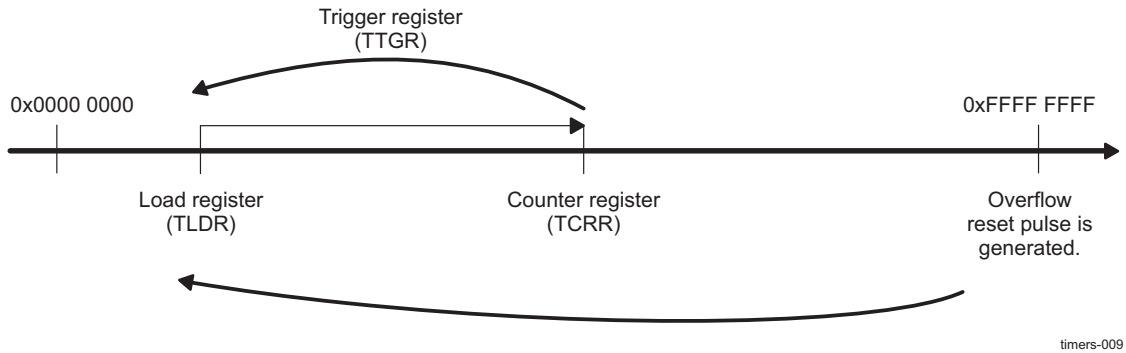
CAUTION

Do not put the overflow value (0xFFFF FFFF) in the TLDR register because it can lead to undesirable results.

An interrupt can be issued on overflow if the overflow interrupt-enable bit is set in the timer interrupt-enable register (the IRQSTATUS_SET[1] OVF_EN_FLAG bit is set to 1 for TIMER1/2/10 and the IRQENABLE_SET[1] OVF_EN_FLAG bit is set to 1 for other timers). A dedicated output pin (timer PWM) can be programmed in the TCLR[12] PT bit through the TCLR[11:10] (PT and TRG bits) to generate one positive pulse (prescaler duration) or to invert the current value (toggle mode) when an overflow occurs. The TCLR[12] PT bit selects pulse/toggle modulation (the TCLR[11:10] TRG bit field selects trigger mode).

Figure 17-9 shows the TCRR timing value.

Figure 17-9. TCRR Timing Value



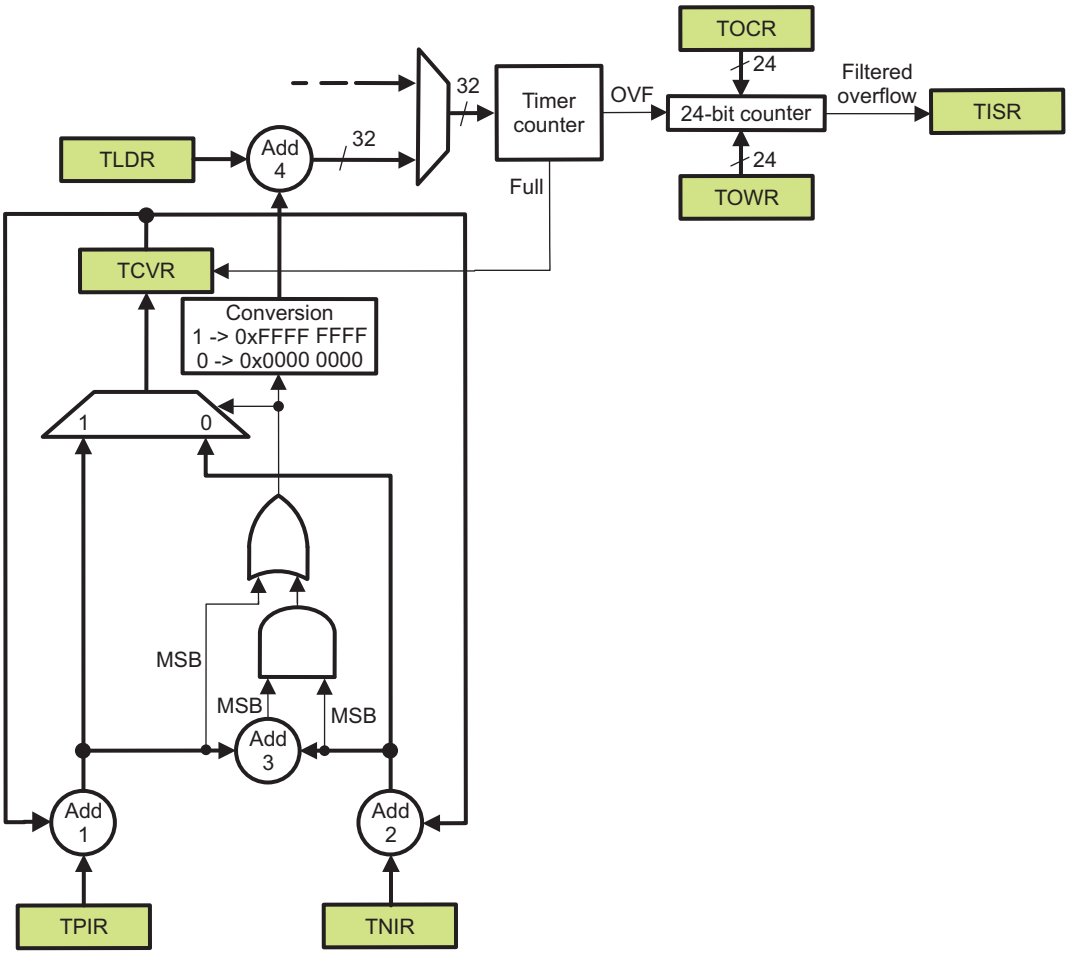
17.2.4.6.1 1-ms Tick Generation (Only TIMER1)

The interrupt period is not exactly 1 ms, because the timer input clock is 32.768 Hz. If the clock counts up to 32, it obtains a 0.977-ms period; if it counts up to 33, it obtains a 1.007-ms period. For large granularity, the error is cumulative and can generate important deviations from the standard value.

To minimize the error between a true 1-ms tick and the tick generated by the 32.768-Hz timer, the sequencing of periods less than 1 ms and periods greater than 1 ms must be shuffled. An additional 1-ms block is used to correct this error. See Figure 17-10.

In this implementation, the increment sequencing is automatically managed by the timer to minimize the error. The user must define only the value of the timer positive increment register (the TPIR[31:0] POSITIVE_INC_VALUE bit field) and the timer negative increment register (the TNIR[31:0] NEGATIVE_INC_VALUE bit field). An automatic adaptation mechanism is used to simplify the programming model.

Figure 17-10. Block Diagram of the 1-ms Tick Module



The TPIR, TNIR, and TCVR registers and adders Add1, Add2, and Add3 are used to define whether the next value loaded in the timer counter register (the TCRR[31:0] TIMER_COUNTER bit field) is the value of the TLDR[31:0] LOAD_VALUE bit field (period less than 1 ms) or the value of TLDR[31:0] LOAD_VALUE –1 (period greater than 1 ms).

Table 17-7 lists the value loaded in the TCRR according to the sign of the result of Add1, Add2, and Add3.

MSB = 0: Positive value; MSB = 1: Negative value

Table 17-7. Value Loaded in TCRR to Generate 1-ms Tick

Add1 MSB	Add2 MSB	Add3 MSB	Value of TCRR Register
0	0	0	TLDR[31:0] LOAD_VALUE bit field
0	0	1	TLDR[31:0] LOAD_VALUE bit field
0	1	0	TLDR[31:0] LOAD_VALUE bit field
0	1	1	TLDR[31:0] LOAD_VALUE –1
1	0	0	N/A
1	0	1	N/A
1	1	0	TLDR[31:0] LOAD_VALUE –1
1	1	1	TLDR[31:0] LOAD_VALUE –1

The values of the TPIR and TNIR registers are calculated using the following formulas:

- Positive increment value = ((INTEGER[F_{clk} × T_{tick}] + 1) × 1e6) – (F_{clk} × T_{tick} × 1e6)
- Negative increment value = (INTEGER[F_{clk} × T_{tick}] × 1e6) – (F_{clk} × T_{tick} × 1e6)

NOTE: F_{clk} clock frequency (kHz)

T_{tick} tick period (ms)

The timer overflow counter register (TOCR) and the timer overflow wrapping register (TOWR) are used to filter interrupts. When the timer overflows, it increments the 24-bit TOCR. When the values in the 24-bit TOCR match the values in the 24-bit TOWR and the timer overflow is asserted, the TOCR is reset and an interrupt is generated to the IRQSTATUS register.

NOTE: TOWR must to be set to requested value. For example, if no interrupt needs to be masked, TOWR must be set to 0; if one interrupt must be masked, TOWR must be set to 1; if two interrupts must be masked TOWR must be set to 2, and so on.

It is important to remember that it is not possible to mask the FFFFFF interrupt.

With the conversion block in reset state (the positive increment register, negative increment register, and counter value register are zeroed), the programming model and the behavior of TIMER1 remain unchanged.

For 1-ms tick with a 32.768-Hz clock:

- TPIR[31:0] POSITIVE_INC_VALUE = 232,000
- TNIR[31:0] NEGATIVE_INC_VALUE = –768,000
- TLDR[31:0] LOAD_VALUE = 0xFFFF FFE0

NOTE: Any value of the tick period can be generated with the appropriate value of the TPIR, TNIR, and TLDR registers.

By default, the TPIR, TNIR, TCVR, TOCR, and TOWR registers and the associated logic are in reset mode (all 0s) and have no effect on the programming model.

17.2.4.7 Capture Mode Functionality

When a transition is detected on the module input pin (EVENT_CAPTURE), the timer value in the TCRR can be captured and saved in the TCAR1 or TCAR2 register function of the mode selected in the TCLR[13] CAPT_MODE bit. The edge detection circuitry monitors transitions on the input pin (EVENT_CAPTURE).

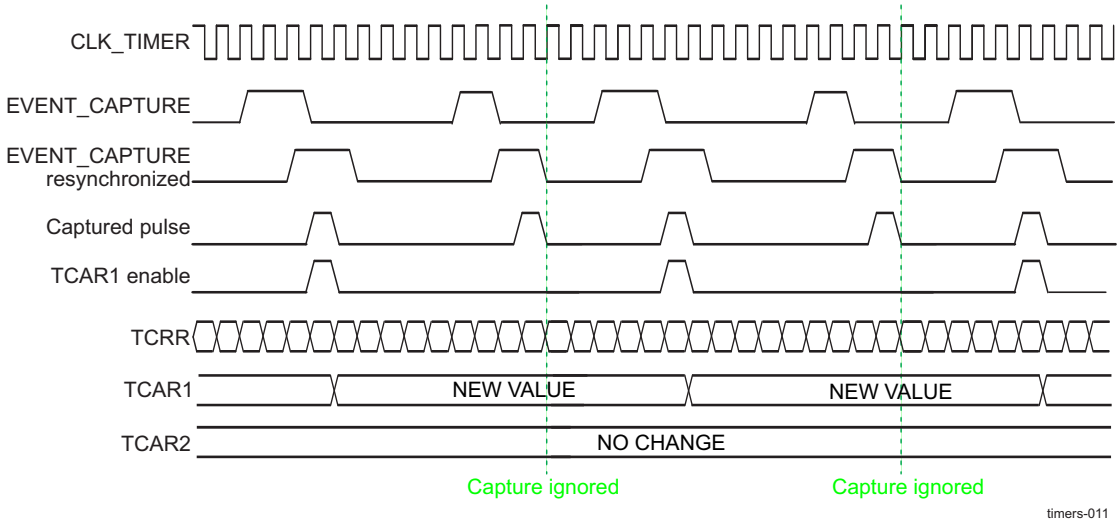
The rising edge, falling edge, or both, can be selected in the TCLR[9:8] TCM bit field to trigger the timer counter capture. The module sets the IRQSTATUS[2] TCAR_IT_FLAG bit when an active edge is detected, and at the same time, the counter value TCRR is stored in timer capture register TCAR1 or TCAR2, as follows:

- If the TCLR[13] CAPT_MODE bit is 0, on the first enabled capture event, the value of the counter register is saved in the TCAR1 register, and the next events are ignored (no update on the TCAR1 register and no interrupt triggering) until the detection logic is reset or the IRQSTATUS[2] TCAR_IT_FLAG is cleared by writing 1 to it.
- If the TCLR[13] CAPT_MODE bit is 1, on the first enabled capture event, the value of the counter register is saved in the TCAR1 register, and on the second enabled capture event, the value of the counter register is saved in the TCAR2 register. If a capture interrupt is enabled, the interrupt triggers on the second event capture. All other events are ignored (no update on TCAR1/TCAR2 and no interrupt triggering) until the detection logic is reset or the IRQSTATUS[2] TCAR_IT_FLAG bit is cleared by writing 1 to it. This mechanism is useful for period calculation of a clock, if that clock is connected to the EVENT_CAPTURE input pin.

The edge detection logic is reset (a new capture is enabled) when the active capture interrupt is served. The IRQSTATUS[2] TCAR_IT_FLAG bit is cleared by writing 1 to it or when the edge detection mode bits (the TCLR[9:8] TCM bit field) are changed from no-capture mode detection to any other mode. The timer functional clock (input to prescaler) is used to sample the input pin (EVENT_CAPTURE). A negative or positive pulse input can be detected when the pulse time is greater than the functional clock period. An interrupt is issued on edge detection if the capture interrupt-enable bit is set in the IRQSTATUS_SET[2] TCAR_EN_FLAG bit (for TIMER1) or in the IRQENABLE_SET[2] TCAR_EN_FLAG bit (for other timers). See the examples in Figure 17-11 and Figure 17-12.

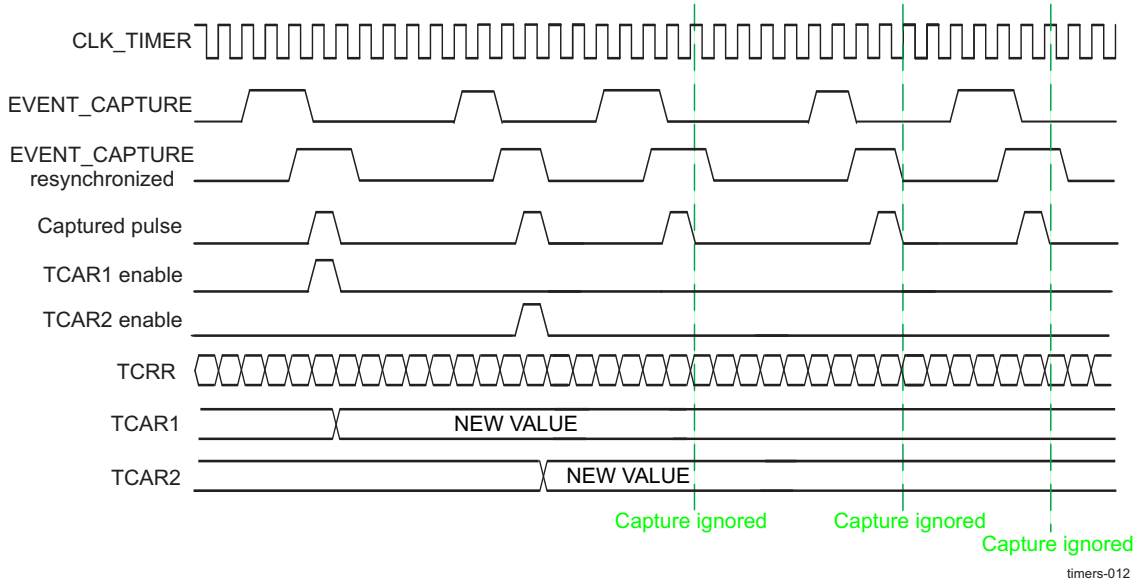
In Figure 17-11, the value of the TCLR[9:8] TCM bit field is 0b01, and the TCLR[13] CAPT_MODE bit is 0. Only the rising edge of EVENT_CAPTURE triggers a capture in the TCAR1 and TCAR2 registers, and only the TCAR1 register updates.

Figure 17-11. Capture Wave Example for TCLR[13] CAPT_MODE = 0



In Figure 17-12, the value of the TCLR[9:8] TCM bit field is 0b01, and the TCLR[13] CAPT_MODE bit is 1. Only the rising edge of EVENT_CAPTURE triggers a capture in the TCAR1 register on the first enabled event, and the TCAR2 register updates on the second enabled event.

Figure 17-12. Capture Wave Example for TCLR[13] CAPT_MODE = 1



17.2.4.8 Compare Mode Functionality

When the TCLR[6] CE bit of the compare-enable register is set to 1, the timer value (the TCRR[31:0] TIMER_COUNTER bit field) is continuously compared to the value held in the timer match register (TMAR). The value of the TMAR[31:0] COMPARE_VALUE bit field can be loaded at any time (timer counting or stopped). When the values of the TCRR and the TMAR registers match, an interrupt is issued, if the IRQSTATUS_SET[0] MAT_EN_FLAG bit (for TIMER1), or the IRQENABLE_SET[0] MAT_EN_FLAG bit (for other timers) is set.

To prevent any unwanted interrupts due to reset value matching effect, write a compare value to the TMAR before setting the TCLR[6] CE bit.

The dedicated output pin (timer PWM) can be programmed in the TCLR[12] PT bit through the TCLR[11:10] TRG bit field to generate one positive pulse (timer clock duration) or to invert the current value (toggle mode) when an overflow or a match occurs.

17.2.4.9 Prescaler Functionality

A prescaler can be used to divide the timer counter input clock frequency. The prescaler is enabled when the TCLR[5] PRE bit is set. The TCLR[4:2] PTV bit field sets the 2^n division ratio (the prescaler value is $2^{(PTV + 1)}$. The prescaler counter is reset when the timer counter is stopped or reloaded on-the-fly.

Table 17-8 lists the prescaler/timer reload values versus contexts.

Table 17-8. Prescaler/Timer Reload Values Versus Contexts

Context	Prescaler	Timer Counter
Overflow (when autoreload is on)	Reset	TLDR[31:0]
TCRR write	Reset	TCRR[31:0]
TTGR write	Reset	TLDR[31:0]
Stop	Reset	Frozen

17.2.4.10 Pulse-Width Modulation

The timer can be configured to provide a programmable PWM output. The timer PWM output pin can be configured to toggle on an event. The TCLR[11:10] TRG bit field determines on which register value the PWM pin toggles. Either overflow or both overflow and match can be selected to toggle the timer PWM pin when a compare condition occurs.

NOTE: In toggle mode, when TCLR[11:10] TRG = 0x2 (overflow and match), the first event that toggles the PWM line is an overflow event. If a match event occurs first, it does not toggle the PWM line (see Figure 17-14).

The TCLR[7] SCPWM bit can be programmed to set or clear the timer PWM output signal only while the counter is stopped or the trigger is off. This allows setting the output pin to a known state before modulation starts. Modulation synchronously stops when the TCLR[11:10] TRG bit field is cleared and overflow occurs. This allows fixing a deterministic state of the output pin when modulation stops.

In Figure 17-13, the internal overflow pulse is set each time the (0xFFFF FFFF – TLDR[31:0] LOAD_VALUE + 1) value is reached, and the internal match pulse is set when the counter reaches the value of TMAR. Depending on the value of the TCLR[12] PT bit and TCLR[11:10] TRG bit field, the timer provides pulse or PWM event on the output pin (timer PWM).

The TLDR and TMAR must keep values below the overflow value (0xFFFF FFFF) by at least two units. If the PWM trigger events are both overflow and match, the difference between the values kept in the TMAR and the value in the TLDR must be at least two units. When match event is used, the compare mode TCLR[6] CE bit must be set.

In Figure 17-13, the TCLR[7] SCPWM bit is set to 0. In Figure 17-14, the TCLR[7] SCPWM bit is set to 1. To obtain the desired wave form, start the counter at 0xFFFF FFFE value (to ensure an overflow first) or adjust the line polarity (TCLR[7] SCPWM bit).

Figure 17-13. Timing Diagram of PWM With TCLR[7] SCPWM Bit = 0

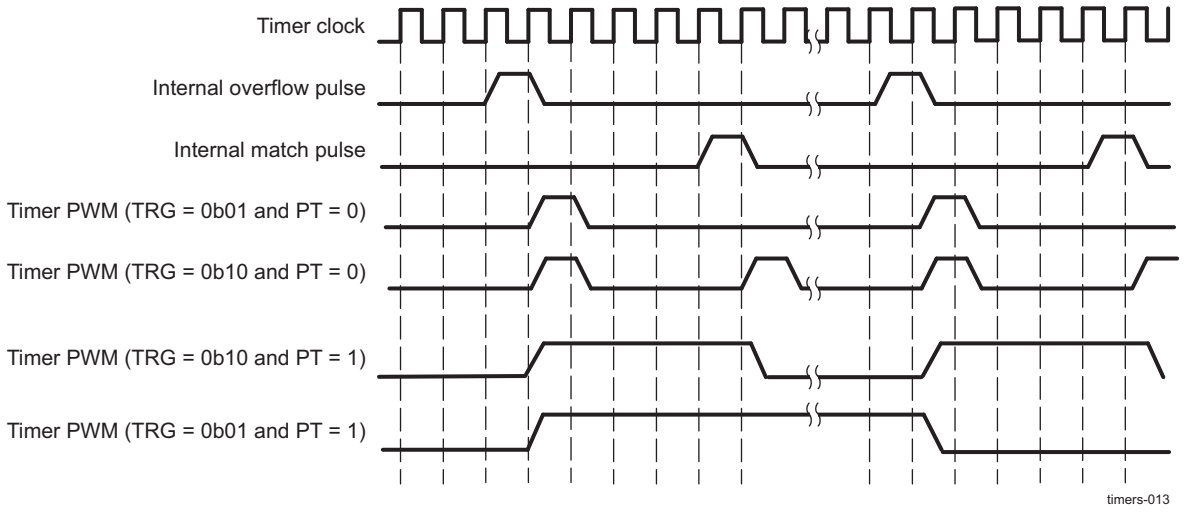
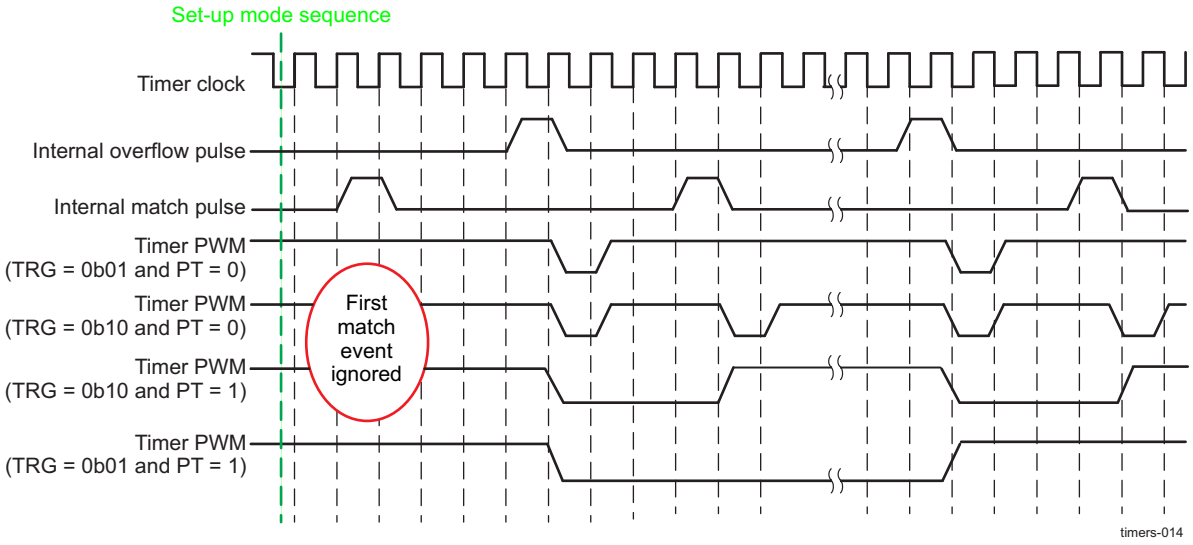


Figure 17-14. Timing Diagram of PWM With TCLR[7] SCPWM Bit = 1



17.2.4.11 Timer Counting Rate

The timer rate is defined by the following values:

- Value of the prescaler fields (the TCLR[5] PRE bit and the TCLR[4:2] PTV bit field)
- Value loaded into the TLDR

Table 17-9 lists the prescaler clock ratio values.

Table 17-9. Prescaler Clock Ratio Values

TCLR[5] PRE	TCLR[4:2] PTV	Divisor (PS)
0	X	1
1	0	2
1	1	4
1	2	8
1	3	16
1	4	32
1	5	64
1	6	128
1	7	256

Thus, the timer overflow rate is expressed as:

OVF_Rate = (0xFFFF FFFF – TLDR + 1) × (timer-functional clock period) × PS

With (timer-functional clock period) = 1/(timer-functional clock frequency) and PS = $2^{(PTV + 1)}$ if prescaler is enabled, or PS = 1 if prescaler is disabled.

NOTE: Internal resynchronization causes any write to the TCLR[1] ST bit to have some latency before the register is updated:

2.5 × functional clock cycles write_TIMER_TCLR_latency 3.5 × functional clock cycles

Remember to consider this latency whenever the timer must be started or stopped by a software change to the TCLR[1] ST bit.

NOTE:

- In non-PWM mode, TLDR must be maintained at less than or equal to 0xFFFF FFFE.
- In PWM mode, TLDR must be maintained at less than or equal to 0xFFFF FFFD.

For example, with a timer clock input of 32 kHz and the TCLR[5] PRE bit set to 0, the timer output period is as listed in Table 17-10.

Table 17-10. Value and Corresponding Interrupt Period

TLDR[31:0] LOAD_VALUE	Interrupt Period
0x0000 0000	37 h
0xFFFF 0000	2 s
0xFFFF FFF0	500 µs
0xFFFF FFFE	62.5 µs

17.2.4.12 Timer Under Emulation

During emulation mode, the timer continues to run according to the value of the TIOCP_ CFG[1] EMUFREE bit.

If the TIOCP_CFG[1] EMUFREE bit is set to 1, timer execution is not stopped in emulation mode and the interrupt is still generated when overflow or match is reached.

If the TIOCP_CFG[1] EMUFREE bit is set to 0, the prescaler and timer are frozen and both resume on exit from emulation mode. The asynchronous external input pin (timerx_pwm_evt, where x = [8:11]) is internally synchronized on two timer-clock rising edges.

17.2.4.13 Accessing GP Timer Registers

All accesses are nonposted until software reconfiguration. All registers are 32 bits wide, accessible through the OCP interface with 16- or 32-bit access (read/write).

Any 16-bit write access must be least significant bit (LSB) first, and the second write access must be most significant bit (MSB) first. Write operations to the following GP timer registers can skip the MSB access if it is not necessary to update the 16 MSBs of the register:

- TIDR (all GP timers)
- TIOCP_CFG (all GP timers)
- IRQSTATUS_SET (GP timer 1)
- IRQSTATUS_RAW (all GP timers)
- IRQSTATUS (all GP timers)
- IRQENABLE_SET (all GP timers except 1)
- IRQENABLE_CLR (all GP timers except 1)
- IRQSTATUS_SET (GP timer 1)
- IRQSTATUS_CLR (GP timer 1)
- IRQWAKEEN (all GP timers)
- TSICR (all GP timers)

Write operations to the following functional registers must be complete (the MSB must be written even if the MSB data is not used):

- TCLR (all GP timers)
- TCRR (all GP timers)
- TLDR (all GP timers)
- TTGR (all GP timers)
- TMAR (all GP timers)

- TPIR (GP timer 1)
- TNIR (GP timer 1)
- TCVR (GP timer 1)
- TOCR (GP timer 1)
- TOWR (GP timer 1)

The following L4 synchronous registers are not affected by the posted/nonposted mode selection; the write/read operation is effective and acknowledged (command accepted) after one L4 clock cycle from command assertion:

- TIDR
- TIOCP_CFG
- IRQSTATUS
- IRQSTATUS_RAW
- IRQENABLE_SET
- IRQENABLE_CLR
- IRQSTATUS_SET
- IRQSTATUS_CLR
- IRQWAKEEN
- TWPS
- TSICR

17.2.4.13.1 Writing to Timer Registers

The host uses the OCP interface to write to the following registers synchronously with the timer interface clock:

- TLDR
- TCRR
- TCLR
- TIOCP_CFG
- IRQSTATUS
- IRQENABLE_SET
- IRQENABLE_CLR
- IRQWAKEEN
- TTGR
- TSICR
- TMAR

TIMER1 also has the following registers:

- IRQSTATUS_SET
- IRQSTATUS_CLR
- TPIR
- TNIR
- TCVR
- TOCR
- TOWR

In 16-bit access mode, the 16 LSBs must be written before writing to the 16 MSBs.

17.2.4.13.1.1 Write Posting Synchronization Mode

This mode is used if the TSICR[2] POSTED bit is set to 1.

This mode uses a posted write scheme to update any internal register (TCLR, TCRR, TLDR, TTGR, TMAR, and TPIR, TNIR, TCVR, TOCR, and TOWR for TIMER1). Therefore, the write transaction is immediately acknowledged on the open-core protocol (OCP) interface, although the effective write operation occurs later because of a resynchronization in the timer clock domain. The advantage is that neither the interconnect nor the device that requested the write transaction is stalled.

For each register, the timer write-posted status (TWPS) register provides a status bit. In this mode, software must check this status bit before any write access. If a write is attempted to a register with a previous access pending, the previous access is discarded without notice.

The timer module updates the value of the timer counter register synchronously with the OCP clock. Consequently, any read access to TCRR does not add any resynchronization latency; the current value is always available.

NOTE: Because the overflow IRQ is generated when the value of TCRR reaches 0xFFFF FFFF, and not when it changes its value to the value after overflow, it is necessary to wait a delay of (1 × PS × timer-functional clock period) before any read access to TCRR to ensure a correct reading of its content.

NOTE: If the TTGR register is written during posted write to TCRR, the value written to TCRR will be discarded.

If a posted write to TCVR is started, the user must not write to TPIR or TNIR before the TCVR write is finished, because the value of TCVR is re-evaluated, so both the value to be written, and the recalculated value will be discarded.

If a write access is pending for a register, reading from this register does not yield a correct result. Software synchronization must be used to avoid incorrect results.

Functional frequency range: freq(timer clock) < freq(OCP interface clock) / 4.

17.2.4.13.1.2 Write Nonposting Synchronization Mode

This mode is used if the TSICR[2] POSTED bit is set to 0 (default value). It uses a nonposted write scheme to update any internal register. Therefore, the write transaction is not acknowledged on the L4 interface until the effective write operation occurs after the resynchronization in the timer functional clock domain. The drawback is that the interconnect and the device that requested the write transaction are stalled during this period.

The same full resynchronization scheme is used for a read transaction, and the same stall period applies. A register read following a write to the same register is always coherent.

This mode is functional regardless of the ratio between the OCP interface frequency and the timer clock frequency.

17.2.4.13.2 Reading From Timer Counter Registers

In 16-bit access mode, reading the 16 LSBs from the timer counter registers (TCRR, TCAR1, and TCAR2) captures the current timer counter value. This must be followed by reading the 16 MSBs. The synchronization schemes for read posted and read non-posted transactions are the same as the corresponded write transactions described before.

NOTE: LSB/MSB accesses cannot be interleaved (that is, the sequence LSB register 1, LSB register 2, MSB register 1, MSB register 2 is not supported).

The TCRR is a 32-bit “atomic datum” and its 16-bit capture is done on the 16-bit LSB first to allow atomic LSB16 + MSB16 capture. This capture scheme is also performed for the TCAR1 and TCAR2 registers as they can be changed due to internal processes too. DSP 16 bit accesses can be interleaved with MCU 32 bit accesses.

NOTE: The counter value of GPTimer5/6/7/8 should be read with a delay of 10 L4_PER clock cycles when the functional clock source is 32kHz and the IPU CD is in the hardware Auto state.

NOTE: The counter value of GPTimer5/6/7/8 can be read without any delay when the functional clock source is 32kHz and the IPU CD is in software Wakeup mode.

17.2.4.13.2.1 Read Posted

This mode is functional whatever the ratio between the OCP interface frequency and the functional clock frequency are. The recommended functional frequency range is freq(timer) < freq(OCP) / 4.

Read posted mode is used if TSICR[2] POSTED = 0x1 or TSICR[3] READ_MODE is set to 0. This mode uses a posted-read scheme for reading any internal timer register. The read transaction is immediately acknowledged on the OCP interface, and the value to be read has been resynchronized. With this method, neither the interconnect nor the device that requested the read transaction are stalled.

Read posted mode applies to TCRR, TCAR1, TCAR2, TCVR, and TOWR, which need resynchronization from functional to OCP clock domains.

17.2.4.13.2.2 Read Non-Posted

This mode is functional regardless of the ratio between the OCP interface frequency and the functional clock frequency. Recommended functional frequency range is freq(timer) >= freq(OCP) / 4.

Read non-posted mode is used if TSICR[2] POSTED = 0x0 and TSICR[3] READ_MODE = 0x1. This mode uses a non-posted read scheme for reading internal timer registers. The read transaction is not acknowledged on the OCP interface until the effective read operation occurs, after the resynchronization in the timer clock domain. The result is that both the interconnect and the device that requested the read transaction are stalled during this period.

This mode applies to TCRR, TCAR1, TCAR2, TCVR, and TOWR, which need resynchronization from functional to OCP clock domains.

17.2.4.14 Posted Mode Selection

A choice between two synchronization modes is made taking into account the frequency ratio and the stall periods that can be supported by the system, without impacting the global performance.

The posted mode selection applies only to registers that require synchronization on or from the timer clock domain. For write operation, the registers affected by posted and non-posted selection are TCLR, TLDR, TCRR, TTGR, TMAR, TPIR, TNIR, TCVR, TOCR, and TOWR. For read operation, the registers affected by this selection are: TCRR, TCAR1, TCAR2, TCVR, and TOWR.

The OCP clock domain synchronous registers TIDR, TIOCP_CFG, IRQSTATUS, IRQSTATUS_SET, IRQWAKEEN, TWPS, and TSICR are not affected by posted and non-posted mode selection. The operation (read or write) is effective and acknowledged after one OCP clock cycle from the command assertion.

The configuration of posted or non-posted mode can be changed (overwritten) by software by writing in TSICR[2] POSTED bit. The TSICR[3] READ_MODE defines how the read operation is performed when the module is configured in non-posted mode (see TSICR). The following cases are possible:

- TSICR[2] POSTED = 0x1 and TSICR[3] READ_MODE = x (don't care): read and write operations are expected in posted mode.
- TSICR[2] POSTED = 0x0 and TSICR[3] READ_MODE = 0x0: the write operation is executed in non-posted mode and read is executed in posted mode.
- TSICR[2]POSTED = 0x0 and TSICR[3] READ_MODE = 0x1: write is executed in non-posted mode and read is executed in non-posted mode.

17.2.5 GP Timer Low-Level Programming Models

This section describes the low-level hardware programming sequences for the configuration and use of the module.

17.2.5.1 Global Initialization

17.2.5.1.1 Global Initialization of Surrounding Modules

This section identifies the requirements for initializing the surrounding modules when the GP timer module is to be used for the first time after a device reset. This initialization of surrounding modules is based on the integration and environment of the GP timer. For more information, see Section 17.2.3, *GP Timers Integration*, and Section 17.2.2, *GP Timers Environment*. Table 17-11 summarizes the surrounding modules.

Table 17-11. Global Initialization of Surrounding Modules

Surrounding Modules	Comments
PRCM	The module interface and functional clocks must be enabled. For more information about the module configuration, see Section 3.1.1.1.2, *Module-Level Clock Management*, in Chapter 3, *Power, Reset, and Clock Management*.
Control module	The module-specific pad muxing must be set in the control module. For more information about the module configuration, see Section 13.4.6.1 *Pad Configuration Registers,* in Chapter 13, *Control Module*.
IRQ_CROSSBAR	The IRQ_CROSSBAR configuration must be done to enable the interrupts from the GP timer module. See Section 13.4.6.4, *IRQ_CROSSBAR Module Functional Description*.

17.2.5.1.2 GP Timer Module Global Initialization

17.2.5.1.2.1 Main Sequence – GP Timer Module Global Initialization

Table 17-12 identifies the main steps for initializing the GP timer module when the module is to be used for the first time.

Table 17-12. GP Timer Module Global Initialization

Step	Register/Bit Field/Programming Model	Value
Execute software reset.	TIOCP_CFG[0] SOFTRESET	0x1
Wait until reset release?	TIOCP_CFG[0] SOFTRESET	0x0
Configure idle mode.	TIOCP_CFG[3:2] IDLEMODE	0x-
Enable wake-up interrupt events.	IRQWAKEEN[2:0]	0x-
Select posted mode.	TSICR[2] POSTED	0x-

17.2.5.2 Operational Mode Configuration

17.2.5.2.1 GP Timer Mode

17.2.5.2.1.1 Main Sequence – GP Timer Mode Configuration

Table 17-13 lists the steps in the GP timer mode configuration.

Table 17-13. GP Timer Mode Configuration

Step	Register/Bit Field/Programming Model	Value
Select autoreload mode.	TCLR[1] AR	0x-
Set prescale timer value.	TCLR[4:2] PTV	0x-
Enable prescaler.	TCLR[5] PRE	0x1

Table 17-13. GP Timer Mode Configuration (continued)

Step	Register/Bit Field/Programming Model	Value
Enable overflow interrupt.	IRQSTATUS_SET[1] OVF_EN_FLAG(1) or IRQENABLE_SET[1] OVF_EN_FLAG(2)	0x1
Load timer counter value.	TCRR	0x-
Load timer load value.	TLDR	0x-
Start the timer.	TCLR[0] ST	0x1

(1) Applies only to TIMER1
(2) Applies to TIMER3 through TIMER8

17.2.5.2.2 GP Timer Compare Mode

17.2.5.2.2.1 Main Sequence – GP Timer Compare Mode Configuration

Table 17-14 lists the steps in the GP timer compare mode configuration.

Table 17-14. GP Timer Compare Mode Configuration

Step	Register/Bit Field/Programming Model	Value
Select autoreload mode.	TCLR[1] AR	0x-
Set prescale timer value.	TCLR[4:2] PTV	0x-
Enable prescaler.	TCLR[5] PRE	0x1
Enable match interrupt.	IRQSTATUS_SET[0] MAT_EN_FLAG(1) or IRQENABLE_SET[0]MAT_EN_FLAG(2)	0x1
Load timer counter value.	TCRR	0x-
Load timer compare value.	TMAR	0x-
Enable compare mode.	TCLR[6] CE	0x1
Start the timer.	TCLR[0] ST	0x1

(1) Applies only to TIMER1
(2) Applies to TIMER3 through TIMER8

17.2.5.2.3 GP Timer Capture Mode

17.2.5.2.3.1 Main Sequence – GP Timer Capture Mode Configuration

Table 17-15 lists the steps in the GP timer capture mode configuration.

Table 17-15. GP Timer Capture Mode Configuration

Step	Register/Bit Field/Programming Model	Value
Initialize capture mode.	See Section 17.2.5.2.3.2.	
Enable capture interrupt.	IRQENABLE_SET[2] TCAR_EN_FLAG (1) or IRQSTATUS_SET[2] TCAR_EN_FLAG (2)	0x1
Start the timer.	TCLR[0] ST	0x1
Detect event.	See Section 17.2.5.2.3.3.	

(1) Applies only to TIMER3 through TIMER8
(2) Applies only to TIMER1

17.2.5.2.3.2 Subsequence – Initialize Capture Mode

Table 17-16 lists the steps to initialize capture mode.

Table 17-16. Initialize Capture Mode

Step	Register/Bit Field/Programming Model	Value
Select autoreload mode.	TCLR[1] AR	0x-
Set prescale timer value.	TCLR[4:2] PTV	0x-
Enable prescaler.	TCLR[5] PRE	0x1
Select TIMERi (where i = 2 to 8). Capture input at device pin timeri.	TCLR[14] GPO_CFG	0x1
Select single or second event capture.	TCLR[13] CAPT_MODE	0x-
Select transition capture mode.	TCLR[9:8] TCM	0x-

17.2.5.2.3.3 Subsequence – Detect Event

Table 17-17 lists the steps in detecting an event.

Table 17-17. Detect Event

Step	Register/Bit Field/Programming Model	Value
Wait until event detected?	IRQSTATUS[2] TCAR_IT_FLAG	= 0x1
Read timer capture value.	TCAR1 and/or TCAR2	
Clear capture interrupt request.	IRQSTATUS[2] TCAR_IT_FLAG	0x1

17.2.5.2.4 GP Timer PWM Mode

17.2.5.2.4.1 Main Sequence – GP Timer PWM Mode Configuration

Table 17-18 lists the steps in the GP timer PWM mode configuration.

Table 17-18. GP Timer PWM Mode Configuration

Step	Register/Bit Field/Programming Model	Value
Select autoreload mode.	TCLR[1] AR	0x-
Set prescale timer value.	TCLR[4:2] PTV	0x-
Enable prescaler.	TCLR[5] PRE	0x1
Select trigger output mode.	TCLR[11:10] TRG	0x-
Select pulse or toggle modulation PWM mode.	TCLR[12] PT	0x-
Select TIMERi (where i = 2 to 11 and 13 to 16) PWM output at device pin timeri.	TCLR[14] GPO_CFG	0x0
Configure PWM output pin default value.	TCLR[7] SCPWM	0x-
Load timer load value.	TLDR	0x-
Load timer compare value.	TMAR	0x-
Enable compare.	TCLR[6] CE	0x1
Start the timer.	TCLR[0] ST	0x1

17.2.6 GP Timer Register Manual

17.2.6.1 GP Timer Instance Summary

Table 17-19 lists the base address and block size for the GP timer module instances.

Table 17-19. GP Timer Instance Summary

Module Name	Base Address L4_PER1 Interconnect	Base Address L4_PER3 Interconnect	Base Address L4_WKUP Interconnect	Size
TIMER2	0x4803 2000	–	–	92 Bytes
TIMER3	0x4803 4000	–	–	92 Bytes
TIMER4	0x4803 6000	–	–	92 Bytes
TIMER5	–	0x4882 0000	--	92 Bytes
TIMER6	–	0x4882 2000	--	92 Bytes
TIMER7	–	0x4882 4000	--	92 Bytes
TIMER8	–	0x4882 6000	--	92 Bytes
TIMER1	--	–	0x4AE1 8000	112 Bytes

17.2.6.2 GP Timer Registers

17.2.6.2.1 GP Timer Register Summary

CAUTION

The GP timer registers are limited to 32- and 16-bit data accesses; 8-bit access is not allowed and can corrupt the register content.

Table 17-20 through Table 17-22 provide the register summary and associated offset addresses for the 8 GP timer internal registers.

Table 17-20. TIMER1 Register Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	TIMER1 Physical Address L4_WKUP Interconnect
TIDR	RO	32	0x0000 0000	0x4AE1 8000
TIOCP_CFG	RW	32	0x0000 0010	0x4AE1 8010
IRQ_EOI	RW	32	0x0000 0020	0x4AE1 8020
IRQSTATUS_RAW	RW	32	0X0000 0024	0x4AE1 8024
IRQSTATUS	RW	32	0X0000 0028	0x4AE1 8028
IRQSTATUS_SET	RW	32	0X0000 002C	0x4AE1 802C
IRQSTATUS_CLR	RW	32	0X0000 0030	0x4AE1 8030
IRQWAKEEN	RW	32	0X0000 0034	0x4AE1 8034
TCLR	RW	32	0x0000 0038	0x4AE1 8038
TCRR	RW	32	0x0000 003C	0x4AE1 803C
TLDR	RW	32	0x0000 0040	0x4AE1 8040
TTGR	RW	32	0x0000 0044	0x4AE1 8044
TWPS	R0	32	0x0000 0048	0x4AE1 8048
TMAR	RW	32	0x0000 004C	0x4AE1 804C
TCAR1	RO	32	0x0000 0050	0x4AE1 8050
TSICR	RW	32	0x0000 0054	0x4AE1 8054
TCAR2	RO	32	0x0000 0058	0x4AE1 8058
TPIR	RW	32	0x0000 005C	0x4AE1 805C
TNIR	RW	32	0x0000 0060	0x4AE1 8060
TCVR	RW	32	0x0000 0064	0x4AE1 8064
TOCR	RW	32	0x0000 0068	0x4AE1 8068
TOWR	RW	32	0x0000 006C	0x4AE1 806C

Table 17-21. TIMER2, TIMER3 and TIMER4 Register Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	TIMER2 Physical Address L4_PER1 Interconnect	TIMER3 Physical Address L4_PER1 Interconnect	TIMER4 Physical Address L4_PER1 Interconnect
TIDR	R	32	0x0000 0000	0x4803 2000	0x4803 4000	0x4803 6000
TIOCP_CFG	RW	32	0x0000 0010	0x4803 2010	0x4803 4010	0x4803 6010
IRQ_EOI	RW	32	0x0000 0020	0x4803 2020	0x4803 4020	0x4803 6020
IRQSTATUS_RAW	RW	32	0x0000 0024	0x4803 2024	0x4803 4024	0x4803 6024
IRQSTATUS	RW	32	0x0000 0028	0x4803 2028	0x4803 4028	0x4803 6028
IRQENABLE_SET	RW	32	0x0000 002C	0x4803 202C	0x4803 402C	0x4803 602C
IRQENABLE_CLR	RW	32	0x0000 0030	0x4803 2030	0x4803 4030	0x4803 6030
IRQWAKEEN	RW	32	0x0000 0034	0x4803 2034	0x4803 4034	0x4803 6034
TCLR	RW	32	0x0000 0038	0x4803 2038	0x4803 4038	0x4803 6038
TCRR	RW	32	0x0000 003C	0x4803 203C	0x4803 403C	0x4803 603C
TLDR	RW	32	0x0000 0040	0x4803 2040	0x4803 4040	0x4803 6040
TTGR	RW	32	0x0000 0044	0x4803 2044	0x4803 4044	0x4803 6044
TWPS	R	32	0x0000 0048	0x4803 2048	0x4803 4048	0x4803 6048
TMAR	RW	32	0x0000 004C	0x4803 204C	0x4803 404C	0x4803 604C
TCAR1	R	32	0x0000 0050	0x4803 2050	0x4803 4050	0x4803 6050
TSICR	RW	32	0x0000 0054	0x4803 2054	0x4803 4054	0x4803 6054
TCAR2	R	32	0x0000 0058	0x4803 2058	0x4803 4058	0x4803 6058

Table 17-22. TIMER5, TIMER6, TIMER7 and TIMER8 Register Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	TIMER5 Physical Address L4_PER3 Interconnect	TIMER6 Physical Address L4_PER3 Interconnect	TIMER7 Physical Address L4_PER3 Interconnect	TIMER8 Physical Address L4_PER3 Interconnect
TIDR	R	32	0x0000 0000	0x4882 0000	0x4882 2000	0x4882 4000	0x4882 6000
TIOCP_CFG	RW	32	0x0000 0010	0x4882 0010	0x4882 2010	0x4882 4010	0x4882 6010
IRQ_EOI	RW	32	0x0000 0020	0x4882 0020	0x4882 2020	0x4882 4020	0x4882 6020
IRQSTATUS_RAW	RW	32	0x0000 0024	0x4882 0024	0x4882 2024	0x4882 4024	0x4882 6024
IRQSTATUS	RW	32	0x0000 0028	0x4882 0028	0x4882 2028	0x4882 4028	0x4882 6028
IRQENABLE_SET	RW	32	0x0000 002C	0x4882 002C	0x4882 202C	0x4882 402C	0x4882 602C
IRQENABLE_CLR	RW	32	0x0000 0030	0x4882 0030	0x4882 2030	0x4882 4030	0x4882 6030
IRQWAKEEN	RW	32	0x0000 0034	0x4882 0034	0x4882 2034	0x4882 4034	0x4882 6034
TCLR	RW	32	0x0000 0038	0x4882 0038	0x4882 2038	0x4882 4038	0x4882 6038
TCRR	RW	32	0x0000 003C	0x4882 003C	0x4882 203C	0x4882 403C	00x4882 603C
TLDR	RW	32	0x0000 0040	0x4882 0040	0x4882 2040	0x4882 4040	0x4882 6040
TTGR	RW	32	0x0000 0044	0x4882 0044	0x4882 2044	0x4882 4044	0x4882 6044
TWPS	R	32	0x0000 0048	0x4882 0048	0x4882 2048	0x4882 4048	0x4882 6048
TMAR	RW	32	0x0000 004C	0x4882 004C	0x4882 204C	0x4882 404C	0x4882 604C
TCAR1	R	32	0x0000 0050	0x4882 0050	0x4882 2050	0x4882 4050	0x4882 6050
TSICR	RW	32	0x0000 0054	0x4882 0054	0x4882 2054	0x4882 4054	0x4882 6054
TCAR2	R	32	0x0000 0058	0x4882 0058	0x4882 2058	0x4882 4058	0x4882 6058

17.2.6.2.2 GP Timer Register Description

Table 17-23 through Table 17-69 describe the individual GP timer registers.

Table 17-23. TIDR

Address Offset	0x0000 0000
Physical Address	See Table 17-20 Table 17-21 Table 17-22 **Instance** See Table 17-19
Description	This read-only register contains the revision number of the module. A write to this register has no effect. This register is used by software to track features, bugs, and compatibility.
Type	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REVISION																															

Bits	Field Name	Description	Type	Reset
31: 0	REVISION	IP Revision	R	0x– (1)

(1) TI internal data

Table 17-24. Register Call Summary for Register TIDR

General-Purpose Timers

- Accessing GP Timer Registers: [0][1]
- GP Timer Register Summary: [2][3][4]

Table 17-25. TIOCP_CFG

Address Offset	0x0000 0010		
Physical Address	See Table 17-20 Table 17-21 Table 17-22	**Instance**	See Table 17-19
Description	This register controls the various parameters of the L4 interface.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												IDLEMODE		EMUFREE	SOFTRESET

Bits	Field Name	Description	Type	Reset
31:4	RESERVED	Reserved	R	0x0000000
3:2	IDLEMODE	Power management, req/ack control 0x0: Force-idle mode: local target idle state follows (acknowledges) the system idle requests unconditionally, that is, regardless of the IP module internal requirements. Back-up mode, for debug only. 0x1: No-idle mode: local target never enters idle state. Back-up mode, for debug only. 0x2: Smart-idle mode: local target idle state eventually follows (acknowledges) the system idle requests, depending on the IP module internal requirements. IP module should not generate (IRQ- or DMA-request-related) wake-up events. 0x3: Smart-idle wake-up-capable mode: local target idle state eventually follows (acknowledges) the system idle requests, depending on the IP module internal requirements. IP module may generate (IRQ- or DMA-request-related) wake-up events when in IDLE state. Mode is relevant only if the appropriate IP module *swake-up* output(s) is (are) implemented.	RW	0x0
1	EMUFREE	Emulation mode 0x0: The timer is frozen in emulation mode (PINSUSPENDN signal active). 0x1: The timer runs free, regardless of PINSUSPENDN value.	RW	0
0	SOFTRESET	Software reset 0x0: Read 0: reset done, no pending action Write 0: No action 0x1: Read 1: initiate software reset Write 1: Reset ongoing	RW	0

Table 17-26. Register Call Summary for Register TIOCP_CFG

General-Purpose Timers

- TIMER1 Power Management: [0]
- Wake-Up Capability: [1]
- Power Management of Other GP Timers: [2]
- Wake-Up Capability: [3]
- Software Reset: [4][5][6]
- Timer Under Emulation: [7][8]
- Accessing GP Timer Registers: [9][10]
- Writing to Timer Registers: [11]
- GP Timer Module Global Initialization: [12][13][14]
- GP Timer Register Summary: [15][16][17]

Table 17-27. IRQ_EOI

Address Offset	0x0000 0020		
Physical Address	See Table 17-20 Table 17-21 Table 17-22	**Instance**	See Table 17-19
Description	Software End-Of-Interrupt: Allows the generation of further pulses on the interrupt line, if an new interrupt event is pending, when using the pulsed output. Unused when using the level interrupt line (depending on module integration).		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															LINE_NUMBER

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	LINE_NUMBER	Write the number of the interrupt line to apply a SW EOI to it. Note that there is only a single line (i.e. number 0). Read : Read always returns 0 Write 0 : SW EOI on interrupt line Write 1 : No action	RW	0x0

Table 17-28. Register Call Summary for Register IRQ_EOI

General-Purpose Timers

- GP Timer Register Summary: [0][1][2]

Table 17-29. IRQSTATUS_RAW

Address Offset	0x0000 0024		
Physical Address	See Table 17-20 Table 17-21 Table 17-22	**Instance**	See Table 17-19
Description	Component interrupt-request status. Check the corresponding secondary status register. Raw status is set even if event is not enabled. Write 1 to set the (raw) status, mostly for debug.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																													TCAR_IT_FLAG	OVF_IT_FLAG	MAT_IT_FLAG

Bits	Field Name	Description	Type	Reset
31:3	RESERVED	Reserved	R	0x0000 0000
2	TCAR_IT_FLAG	IRQ status for capture Read 0: No event pending Write 0: No action Read 1: IRQ event pending Write 1: Trigger IRQ event by software.	RW	0
1	OVF_IT_FLAG	IRQ status for overflow Read 0: No event pending Write 0: No action Read 1: IRQ event pending Write 1: Trigger IRQ event by software.	RW	0
0	MAT_IT_FLAG	IRQ status for match Read 0: No event pending Write 0: No action Read 1: IRQ event pending Write 1: Trigger IRQ event by software.	RW	0

Table 17-30. Register Call Summary for Register IRQSTATUS_RAW

General-Purpose Timers

- Accessing GP Timer Registers: [0][1]
- GP Timer Register Summary: [2][3][4]

Table 17-31. IRQSTATUS

Address Offset	0x0000 0028		
Physical Address	See Table 17-20 Table 17-21 Table 17-22	**Instance**	See Table 17-19
Description	Component interrupt-request status. Check the corresponding secondary status register. Enabled status is not set unless event is enabled. Write 1 to clear the status after interrupt has been serviced (raw status gets cleared, that is, even if not enabled).		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																													TCAR_IT_FLAG	OVF_IT_FLAG	MAT_IT_FLAG

Bits	Field Name	Description	Type	Reset
31:3	RESERVED	Reserved	R	0x0000 0000
2	TCAR_IT_FLAG	IRQ status for capture Read 0: No event pending Write 0: No action Read 1: IRQ event pending Write 1: Clear any pending event.	RW	0
1	OVF_IT_FLAG	IRQ status for overflow Read 0: No event pending Write 0: No action Read 1: IRQ event pending Write 1: Clear any pending event.	RW	0
0	MAT_IT_FLAG	IRQ status for match Read 0: No event pending Write 0: No action Read 1: IRQ event pending Write 1: Clear any pending event.	RW	0

Table 17-32. Register Call Summary for Register IRQSTATUS

General-Purpose Timers

- Wake-Up Capability: [0]
- Wake-Up Capability: [1]
- GP Timer Interrupts: [2]
- 1-ms Tick Generation (Only TIMER1): [3]
- Capture Mode Functionality: [4][5][6][7]
- Accessing GP Timer Registers: [8][9]
- Writing to Timer Registers: [10]
- GP Timer Capture Mode: [11][12]
- GP Timer Register Summary: [13][14][15]

Table 17-33. IRQENABLE_SET

Address Offset	0x0000 002C		
Physical Address	See Table 17-20 Table 17-21 Table 17-22	**Instance**	See Table 17-19
Description	Component interrupt-request enable. Write 1 to set (enable interrupt). Readout equal to corresponding _CLR register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																													TCAR_EN_FLAG	OVF_EN_FLAG	MAT_EN_FLAG

Bits	Field Name	Description	Type	Reset
31:3	RESERVED	Reserved	R	0x0000 0000
2	TCAR_EN_FLAG	IRQ enable for compare Read 0: IRQ event is disabled. Write 0: No action Read 1: IRQ event is enabled Write 1: Set IRQ enable.	RW	0
1	OVF_EN_FLAG	IRQ enable for overflow Read 0: IRQ event is disabled. Write 0: No action Read 1: IRQ event is enabled. Write 1: Set IRQ enable.	RW	0

Bits	Field Name	Description	Type	Reset
0	MAT_EN_FLAG	IRQ enable for match Read 0: IRQ event is disabled. Write 0: No action Read 1: IRQ event is enabled. Write 1: Set IRQ enable.	RW	0

Table 17-34. Register Call Summary for Register IRQENABLE_SET

General-Purpose Timers

- GP Timer Functional Description: [0]
- GP Timer Interrupts: [1]
- Timer Mode Functionality: [2]
- Capture Mode Functionality: [3]
- Compare Mode Functionality: [4]
- Accessing GP Timer Registers: [5][6]
- Writing to Timer Registers: [7]
- GP Timer Mode: [8]
- GP Timer Compare Mode: [9]
- GP Timer Capture Mode: [10]
- GP Timer Register Summary: [11][12]

Table 17-35. IRQENABLE_CLR

Address Offset	0x0000 0030		
Physical Address	See Table 17-20 Table 17-21 Table 17-22	**Instance**	See Table 17-19
Description	Component interrupt-request enable. Write 1 to clear (disable interrupt). Readout equal to corresponding _SET register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																													TCAR_EN_FLAG	OVF_EN_FLAG	MAT_EN_FLAG

Bits	Field Name	Description	Type	Reset
31:3	RESERVED	Reserved	R	0x0000 0000
2	TCAR_EN_FLAG	IRQ enable for compare Read 0: IRQ event is disabled. Write 0: No action Read 1: IRQ event is enabled. Write 1: Clear IRQ enable.	RW	0
1	OVF_EN_FLAG	IRQ enable for overflow Read 0: IRQ event is disabled. Write 0: No action Read 1: IRQ event is enabled. Write 1: Clear IRQ enable.	RW	0
0	MAT_EN_FLAG	IRQ enable for match Read 0: IRQ event is disabled. Write 0: No action Read 1: IRQ event is enabled. Write 1: Clear IRQ enable.	RW	0

Table 17-36. Register Call Summary for Register IRQENABLE_CLR

General-Purpose Timers

- GP Timer Functional Description: [0]
- GP Timer Interrupts: [1]
- Accessing GP Timer Registers: [2][3]
- Writing to Timer Registers: [4]
- GP Timer Register Summary: [5][6]

Table 17-37. IRQWAKEEN

Address Offset	0x0000 0034		
Physical Address	See Table 17-20 Table 17-21 Table 17-22	**Instance**	See Table 17-19
Description	Wake-up-enabled events taking place when module is idle should generate an asynchronous wake-up.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																													TCAR_WUP_ENA	OVF_WUP_ENA	MAT_WUP_ENA

Bits	Field Name	Description	Type	Reset
31:3	RESERVED	Reserved	R	0x0000 0000
2	TCAR_WUP_ENA	Wake-up generation for compare 0x0: Wake-up disabled 0x1: Wake-up enabled	RW	0
1	OVF_WUP_ENA	Wake-up generation for overflow 0x0: Wake-up disabled 0x1: Wake-up enabled	RW	0
0	MAT_WUP_ENA	Wake-up generation for match 0x0: Wake-up disabled 0x1: Wake-up enabled	RW	0

Table 17-38. Register Call Summary for Register IRQWAKEEN

General-Purpose Timers

- GP Timer Functional Description: [0]
- Wake-Up Capability: [1]
- Accessing GP Timer Registers: [2][3]
- Writing to Timer Registers: [4]
- GP Timer Module Global Initialization: [5]
- GP Timer Register Summary: [6][7][8]

Table 17-39. TCLR

Address Offset	0x0000 0038		
Physical Address	See Table 17-20 Table 17-21 Table 17-22	**Instance**	See Table 17-19
Description	This register controls optional features specific to the timer functionality.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																GPO_CFG	CAPT_MODE	PT	TRG		TCM		SCPWM	CE	PRE	PTV			AR	ST	

Bits	Field Name	Description	Type	Reset
31:15	RESERVED	Reserved	R	0x00000
14	GPO_CFG	General-purpose output - this register directly drives the PO_GPOCFG output. For specific use of the GPO_CFG bit, see Section 17.2.2.1, *GP Timer External System Interface*. 0x0: PO_GPOCFG drives 0. 0x1: PO_GPOCFG drives 1.	RW	0
13	CAPT_MODE	Capture mode select bit (first/second) 0x0: Single capture: Capture the first enabled capture event in TCAR1. 0x1: Capture on second event: Capture the second enabled capture event in TCAR1 and the second enabled capture event in TCAR2.	RW	0
12	PT	Pulse or toggle mode on TIMERi_PWM_out output pin 0x0: Pulse modulation 0x1: Toggle modulation	RW	0
11:10	TRG	Trigger output mode on TIMERi_PWM_out output pin 0x0: No trigger 0x1: Trigger on overflow. 0x2: Trigger on overflow and match. 0x3: Reserved	RW	0x0
9:8	TCM	Transition capture mode on TIMERi_EVENT_CAPTURE input pin (When the TCM field passed from (00) to any other combination, the TCAR_IT_FLAG and the edge detection logic are cleared.) 0x0: No capture 0x1: Capture on rising edges of TIMERi_EVENT_CAPTURE pin 0x2: Capture on falling edges of TIMERi_EVENT_CAPTURE pin 0x3: Capture on both edges of TIMERi_EVENT_CAPTURE pin	RW	0x0
7	SCPWM	Pulse width modulation output pin default setting This bit must be set or clear while the timer is stopped or the trigger is off. 0x0: Clear the TIMERi_PWM_out output pin and select positive pulse for pulse mode. 0x1: Set the TIMERi_PWM_out output pin and select negative pulse for pulse mode.	RW	0

Bits	Field Name	Description	Type	Reset
6	CE	Compare enable 0x0: Compare mode is disable. 0x1: Compare mode is enable.	RW	0
5	PRE	Prescaler enable 0x0: The TIMER clock input pin clocks the counter. 0x1: The divided input pin clocks the counter.	RW	0
4:2	PTV	Prescale clock timer value The timer counter is prescaled with the value $2^{(PTV+1)}$. Example: PTV = 3, counter increases value (if started) after 16 functional clock periods.	RW	0x0
1	AR	Autoreload mode 0x0: One shot timer 0x1: Autoreload timer	RW	0
0	ST	Start/stop timer control 0x0: Stop timer: Only the counter is frozen. If one-shot mode selected (AR =0), this bit is automatically reset by internal logic when the counter is overflowed. 0x1: Start timer	RW	0

Table 17-40. Register Call Summary for Register TCLR

General-Purpose Timers

- Timer Mode Functionality: [2][3][4][5][6][7][8]
- Capture Mode Functionality: [9][10][11][12][13][14][15][16][17]
- Compare Mode Functionality: [18][19][20][21]
- Prescaler Functionality: [22][23]
- Pulse-Width Modulation: [24][25][26][27][28][29][30][31][32][33]
- Timer Counting Rate: [34][35][36][37][38][39][40]
- Accessing GP Timer Registers: [41]
- Writing to Timer Registers: [42][43]
- GP Timer Mode: [44][45][46][47]
- GP Timer Compare Mode: [48][49][50][51][52]
- GP Timer Capture Mode: [53][54][55][56][57][58][59]
- GP Timer PWM Mode: [60][61][62][63][64][65][66][67][68]
- GP Timer Register Summary: [69][70][71]
- GP Timer Register Description: [72][73]

Table 17-41. TCRR

Address Offset	0x0000 003C		
Physical Address	See Table 17-20 Table 17-21 Table 17-22	**Instance**	See Table 17-19
Description	This register holds the value of the internal counter.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TIMER_COUNTER																															

Bits	Field Name	Description	Type	Reset
31:0	TIMER_COUNTER	Value of TIMER counter	RW	0x0000 0000

Table 17-42. Register Call Summary for Register TCRR

General-Purpose Timers

- Wake-Up Capability: [0][1]
- Timer Mode Functionality: [2][3][4][5][6][7][8][9][10]
- 1-ms Tick Generation (Only TIMER1): [11][12][13]
- Capture Mode Functionality: [14][15]
- Compare Mode Functionality: [16][17]
- Prescaler Functionality: [18][19]
- Accessing GP Timer Registers: [20]
- Writing to Timer Registers: [21][22][23][24][25][26][27]
- Reading From Timer Counter Registers: [28][29]
- GP Timer Mode: [30]
- GP Timer Compare Mode: [31]
- GP Timer Register Summary: [32][33][34]
- GP Timer Register Description: [35][36]
- TIMER1 Register Description: [37][38][39]

Table 17-43. TLDR

Address Offset	0x0000 0040		
Physical Address	See Table 17-20 Table 17-21 Table 17-22	**Instance**	See Table 17-19
Description	This register holds the timer load value.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LOAD_VALUE																															

Bits	Field Name	Description	Type	Reset
31:0	LOAD_VALUE	Timer counter value loaded on overflow in autoreload mode or on TTGR write access. LOAD_VALUE must be different than the timer overflow value (0xFFFF FFFF).	RW	0x0000 0000

Table 17-44. Register Call Summary for Register TLDR

General-Purpose Timers

- Timer Mode Functionality: [0][1][2]
- 1-ms Tick Generation (Only TIMER1): [3][4][5][6][7][8][9][10][11][12]
- Prescaler Functionality: [13][14]
- Pulse-Width Modulation: [15][16][17]
- Timer Counting Rate: [18][19][20][21][22]
- Accessing GP Timer Registers: [23]
- Writing to Timer Registers: [24][25]
- GP Timer Mode: [26]
- GP Timer PWM Mode: [27]
- GP Timer Register Summary: [28][29][30]
- GP Timer Register Description: [31][32]

Table 17-45. TTGR

Address Offset	0x0000 0044		
Physical Address	See Table 17-20 Table 17-21 Table 17-22	**Instance**	See Table 17-19
Description	The read value of this register is always 0xFFFF FFFF.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TTGR_VALUE																															

Bits	Field Name	Description	Type	Reset
31:0	TTGR_VALUE	Writing to the TTGR register causes the TCRR to be loaded from TLDR and the prescaler counter to be cleared. Reload is done regardless of the AR field value of the TCLR register.	RW Rreturns 1s	0xFFFF FFFF

Table 17-46. Register Call Summary for Register TTGR

General-Purpose Timers

- Timer Mode Functionality: [0][1]
- Prescaler Functionality: [2]
- Accessing GP Timer Registers: [3]
- Writing to Timer Registers: [4][5][6]
- GP Timer Register Summary: [7][8][9]
- GP Timer Register Description: [10][11][12]

Table 17-47. TWPS

Address Offset	0x0000 0048		
Physical Address	See Table 17-20 Table 17-21 Table 17-22	**Instance**	See Table 17-19
Description	This register contains the write posting bits for all writable functional registers.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						W_PEND_TOWR	W_PEND_TOCR	W_PEND_TCVR	W_PEND_TNIR	W_PEND_TPIR	W_PEND_TMAR	W_PEND_TTGR	W_PEND_TLDR	W_PEND_TCRR	W_PEND_TCLR

Bits	Field Name	Description	Type	Reset
31:10	RESERVED	Reserved	R	0x0000000
9	W_PEND_TOWR	Write pending for the TOWR register Read 1: Write pending Read 0: No write pending	R	0
8	W_PEND_TOCR	Write pending for the TOCR register Read 1: Write pending Read 0: No write pending	R	0
7	W_PEND_TCVR	Write pending for the TCVR register Read 1: Write pending Read 0: No write pending	R	0

Bits	Field Name	Description	Type	Reset
6	W_PEND_TNIR	Write pending for the TNIR register Read 1: Negative increment register write pending Read 0: No negative increment register write pending	R	0
5	W_PEND_TPIR	Write pending for the TPIR register Read 1: Positive increment register write pending Read 0: No positive increment register write pending	R	0
4	W_PEND_TMAR	When equal to 1, a write is pending to the TMAR register.	R	0
3	W_PEND_TTGR	When equal to 1, a write is pending to the TTGR register.	R	0
2	W_PEND_TLDR	When equal to 1, a write is pending to the TLDR register.	R	0
1	W_PEND_TCRR	When equal to 1, a write is pending to the TCRR register.	R	0
0	W_PEND_TCLR	When equal to 1, a write is pending to the TCLR register.	R	0

Table 17-48. Register Call Summary for Register TWPS

General-Purpose Timers
- Accessing GP Timer Registers: [0]
- Writing to Timer Registers: [1]
- GP Timer Register Summary: [2][3][4]

Table 17-49. TMAR

Address Offset	0x0000 004C		
Physical Address	See Table 17-20 Table 17-21 Table 17-22	**Instance**	See Table 17-19
Description	The compare logic consists of a 32-bit-wide, read/write data TMAR register and logic to compare counter.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
COMPARE_VALUE																															

Bits	Field Name	Description	Type	Reset
31:0	COMPARE_VALUE	Value to be compared to the timer counter	RW	0x0000 0000

Table 17-50. Register Call Summary for Register TMAR

General-Purpose Timers
- Wake-Up Capability: [0]
- Compare Mode Functionality: [1][2][3][4]
- Pulse-Width Modulation: [5][6][7]
- Accessing GP Timer Registers: [8]
- Writing to Timer Registers: [9][10]
- GP Timer Compare Mode: [11]
- GP Timer PWM Mode: [12]
- GP Timer Register Summary: [13][14][15]
- GP Timer Register Description: [16][17]

Table 17-51. TCAR1

Address Offset	0x0000 0050		
Physical Address	See Table 17-20 Table 17-21 Table 17-22	**Instance**	See Table 17-19
Description	This register holds the first captured value of the counter register.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CAPTURE_VALUE1																															

Bits	Field Name	Description	Type	Reset
31:0	CAPTURE_VALUE1	First timer counter value captured on an external event trigger	R	0x0000 0000

Table 17-52. Register Call Summary for Register TCAR1

General-Purpose Timers

- Capture Mode Functionality: [0][1][2][3][4][5][6][7][8]
- Reading From Timer Counter Registers: [9][10]
- GP Timer Capture Mode: [11]
- GP Timer Register Summary: [12][13][14]
- GP Timer Register Description: [15][16]

Table 17-53. TSICR

Address Offset	0x0000 0054		
Physical Address	See Table 17-20 Table 17-21 Table 17-22	**Instance**	See Table 17-19
Description	Timer synchronous interface control register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												READ_MODE	POSTED	SFT	RESERVED

Bits	Field Name	Description	Type	Reset
31:4	RESERVED	Reserved	R	0x000 0000
3	READ_MODE	Select posted/non-posted mode for read operation: 0x0: When the module is configured in non-posted mode(POSTED = '0'), the read operation is executed as read posted. 0x1: When the module is configured in non-posted mode(POSTED = '0'), the read operation is executed as read non-posted. NOTE: When the module is configured in posted mode(POSTED = '1'), this bit is not used. NOTE: For GP TIMER1 this bit is write only.	RW	0
2	POSTED	Posted mode selection 0x0: Posted mode inactive: Delay the command accept output signal. 0x1: Posted mode active	RW	0

Bits	Field Name	Description	Type	Reset
1	SFT	This bit resets all the functional part of the module. 0x0: Software reset is disabled. 0x1: Software reset is enabled.	RW	0
0	RESERVED	Reserved	R	0

Table 17-54. Register Call Summary for Register TSICR

General-Purpose Timers

- Software Reset: [0][1]
- Accessing GP Timer Registers: [2][3]
- Writing to Timer Registers: [4][5][6]
- Reading From Timer Counter Registers: [7][8][9][10]
- Posted Mode Selection: [11][12][13][14][15][16][17][18]
- GP Timer Module Global Initialization: [19]
- GP Timer Register Summary: [20][21][22]

Table 17-55. TCAR2

Address Offset	0x0000 0058		
Physical Address	See Table 17-20 Table 17-21 Table 17-22	**Instance**	See Table 17-19
Description	This register holds the second captured value of the counter register.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CAPTURE_VALUE2																															

Bits	Field Name	Description	Type	Reset
31:0	CAPTURE_VALUE2	Second timer counter value captured on an external event trigger	R	0x0000 0000

Table 17-56. Register Call Summary for Register TCAR2

General-Purpose Timers

- Capture Mode Functionality: [0][1][2][3][4][5]
- Reading From Timer Counter Registers: [6][7]
- GP Timer Capture Mode: [8]
- GP Timer Register Summary: [9][10][11]
- GP Timer Register Description: [12]

17.2.6.2.3 TIMER1 Register Description

Table 17-57. TPIR

Address Offset	0x0000 005C		
Physical Address	See Table 17-20	**Instance**	See Table 17-19
Description	This register is used for 1-ms tick generation. The TPIR register holds the value of the positive increment. The value of this register is added to the value of TCVR to determine whether next value loaded in TCRR is the subperiod value or the overperiod value.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
POSITIVE_INC_VALUE																															

Bits	Field Name	Description	Type	Reset
31:0	POSITIVE_INC_VALUE	Value of the positive increment	RW	0x0000 0000

Table 17-58. Register Call Summary for Register TPIR

General-Purpose Timers

- 1-ms Tick Generation (Only TIMER1): [0][1][2][3][4][5]
- Accessing GP Timer Registers: [6]
- Writing to Timer Registers: [7][8][9]
- GP Timer Register Summary: [10]
- GP Timer Register Description: [11]
- TIMER1 Register Description: [12]

Table 17-59. TNIR

Address Offset	0x0000 0060		
Physical Address	See Table 17-20	**Instance**	See Table 17-19
Description	This register is used for 1-ms tick generation. The TNIR register holds the value of the negative increment. The value of this register is added to the value of the TCVR to determine whether next value loaded in TCRR is the subperiod value or the overperiod value.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NEGATIVE_INV_VALUE																															

Bits	Field Name	Description	Type	Reset
31:0	NEGATIVE_INV_VALUE	Value of the negative increment	RW	0x0000 0000

Table 17-60. Register Call Summary for Register TNIR

General-Purpose Timers

- 1-ms Tick Generation (Only TIMER1): [0][1][2][3][4][5]
- Accessing GP Timer Registers: [6]
- Writing to Timer Registers: [7][8][9]
- GP Timer Register Summary: [10]
- GP Timer Register Description: [11]
- TIMER1 Register Description: [12]

Table 17-61. TCVR

Address Offset	0x0000 0064		
Physical Address	See Table 17-20	**Instance**	See Table 17-19
Description	This register is used for 1-ms tick generation. The TCVR register determines whether next value loaded in TCRR is the subperiod value or the overperiod value.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
COUNTER_VALUE																															

Bits	Field Name	Description	Type	Reset
31:0	COUNTER_VALUE	Value of CVR counter	RW	0x0000 0000

Table 17-62. Register Call Summary for Register TCVR

General-Purpose Timers

- 1-ms Tick Generation (Only TIMER1): [0][1]
- Accessing GP Timer Registers: [2]
- Writing to Timer Registers: [3][4][5][6][7]
- GP Timer Register Summary: [8]
- GP Timer Register Description: [9]
- TIMER1 Register Description: [10][11][12]

Table 17-63. TOCR

Address Offset	0x0000 0068		
Physical Address	See Table 17-20	**Instance**	See Table 17-19
Description	This register is used to mask the tick interrupt for a selected number of ticks.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								OVF_COUNTER_VALUE																							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED	Reads return 0.	R	0x00
23:0	OVF_COUNTER_VALUE	Number of overflow events	RW	0x000000

Table 17-64. Register Call Summary for Register TOCR

General-Purpose Timers

- 1-ms Tick Generation (Only TIMER1): [0][1][2][3][4]
- Accessing GP Timer Registers: [5]
- Writing to Timer Registers: [6][7]
- GP Timer Register Summary: [8]
- GP Timer Register Description: [9]

Table 17-65. TOWR

Address Offset	0x0000 006C		
Physical Address	See Table 17-20	**Instance**	See Table 17-19
Description	This register holds the number of masked overflow interrupts.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								OVF_WRAPPING_VALUE																							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED	Reads return 0.	R	0x00
23:0	OVF_WRAPPING_VALUE	Number of masked interrupts	RW	0x000000

Table 17-66. Register Call Summary for Register TOWR

General-Purpose Timers

- 1-ms Tick Generation (Only TIMER1): [0][1][2][3][4][5][6]
- Accessing GP Timer Registers: [7]
- Writing to Timer Registers: [8][9]
- GP Timer Register Summary: [10]
- GP Timer Register Description: [11]

Table 17-67. IRQSTATUS_SET

Address Offset	0x0000 002C		
Physical Address	See Table 17-20	**Instance**	See Table 17-19
Description	Component interrupt-request enable. Write 1 to set (enable interrupt). Readout equal to corresponding _CLR register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																													TCAR_EN_FLAG	OVF_EN_FLAG	MAT_EN_FLAG

Bits	Field Name	Description	Type	Reset
31:3	RESERVED	Reserved	R	0x0000 0000
2	TCAR_EN_FLAG	IRQ enable for compare Read 0: IRQ event is disabled. Write 0: No action Read 1: IRQ event is enabled Write 1: Set IRQ enable.	RW	0
1	OVF_EN_FLAG	IRQ enable for overflow Read 0: IRQ event is disabled. Write 0: No action Read 1: IRQ event is enabled. Write 1: Set IRQ enable.	RW	0
0	MAT_EN_FLAG	IRQ enable for match Read 0: IRQ event is disabled. Write 0: No action Read 1: IRQ event is enabled. Write 1: Set IRQ enable.	RW	0

Table 17-68. Register Call Summary for Register IRQSTATUS_SET

General-Purpose Timers

- GP Timer Functional Description: [0]
- GP Timer Interrupts: [1]
- Timer Mode Functionality: [2]
- Capture Mode Functionality: [3]
- Compare Mode Functionality: [4]
- Accessing GP Timer Registers: [5][6][7]
- Writing to Timer Registers: [8]
- GP Timer Mode: [9]
- GP Timer Compare Mode: [10]
- GP Timer Capture Mode: [11]
- GP Timer Register Summary: [12]

Table 17-69. IRQSTATUS_CLR

Address Offset	0x0000 0030		
Physical Address	See Table 17-20	**Instance**	See Table 17-19
Description	Component interrupt-request enable. Write 1 to clear (disable interrupt). Readout equal to corresponding _SET register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																													TCAR_EN_FLAG	OVF_EN_FLAG	MAT_EN_FLAG

Bits	Field Name	Description	Type	Reset
31:3	RESERVED	Reserved	R	0x0000 0000
2	TCAR_EN_FLAG	IRQ enable for compare Read 0: IRQ event is disabled. Write 0: No action Read 1: IRQ event is enabled. Write 1: Clear IRQ enable.	RW	0
1	OVF_EN_FLAG	IRQ enable for overflow Read 0: IRQ event is disabled. Write 0: No action Read 1: IRQ event is enabled. Write 1: Clear IRQ enable.	RW	0
0	MAT_EN_FLAG	IRQ enable for match Read 0: IRQ event is disabled. Write 0: No action Read 1: IRQ event is enabled. Write 1: Clear IRQ enable.	RW	0

Table 17-70. Register Call Summary for Register IRQSTATUS_CLR

General-Purpose Timers

- GP Timer Functional Description: [0]
- GP Timer Interrupts: [1]
- Accessing GP Timer Registers: [2][3]
- Writing to Timer Registers: [4]
- GP Timer Register Summary: [5]

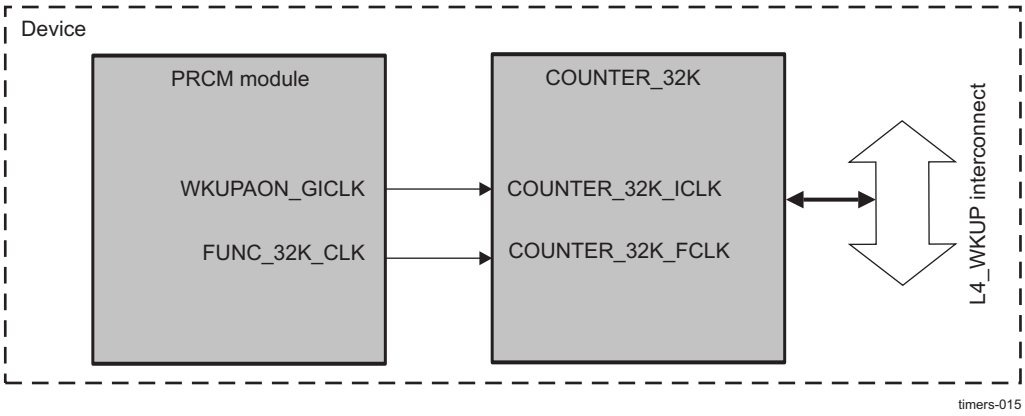
17.3 32-kHz Synchronized Timer (COUNTER_32K)

17.3.1 32-kHz Synchronized Timer Overview

The 32-kHz synchronized timer (COUNTER_32K) is a 32-bit counter clocked by the falling edge of the 32-kHz system clock.

Figure 17-15 is the block diagram of the 32-kHz synchronized timer.

Figure 17-15. 32-kHz Synchronized Timer Block Diagram



17.3.1.1 32-kHz Synchronized Timer Features

The main features of the 32-kHz synchronized timer controller are:

- L4 slave interface (OCP) support:
  - 32-bit data bus width
  - 32-/16-bit access supported
  - 8-bit access not supported
  - 16-bit address bus width
  - Burst mode not supported
  - Write nonposted transaction mode not supported
- Only read operations are supported on the module registers; no write operation is supported (no error/no action on write).
- Free-running 32-bit upward counter
- Start and keep counting after power-on reset
- Automatic roll over to 0; highest value reached: 0xFFFF FFFF
- On-the-fly read (while counting)

17.3.2 32-kHz Synchronized Timer Integration

The synchronized timer is accessible only through the L4_WKUP interface.

Table 17-71 through Table 17-73 summarize the integration of the module in the device.

Table 17-71. Integration Attributes

Module Instance	Attributes		
	Power Domain	Wake-Up Capability	Interconnect
COUNTER_32K	PD_WKUPAON	No	L4_WKUP

Table 17-72. Clocks and Resets

Clocks				
Module Instance	**Destination Signal Name**	**Source Signal Name**	**Source**	**Description**
COUNTER_32K	COUNTER_32K _FCLK	FUNC_32K_CLK	PRCM	COUNTER_32K functional 32KHz clock
COUNTER_32K	COUNTER_32K _ICLK	WKUPAON_GICLK	PRCM	COUNTER_32K interface clock
Resets				
COUNTER_32K	COUNTER_32K _NRESPWRON	WKUPAON_SYS_PWRON _RST	PRM	Reset to COUNTER_32K. Reset all internal logic, running on COUNTER_32K_FCLK
COUNTER_32K	COUNTER_32K _OCPRESETN	WKUPAON_RST	PRM	Reset to COUNTER_32K. Reset all L4 interface logic, running on COUNTER_32K_ICLK.

Table 17-73. Hardware Requests

No Interrupt Requests
No DMA Requests

17.3.3 32-kHz Synchronized Timer Functional Description

The synchronized timer is a counter that starts on the rising edge of an external asynchronous signal (COUNTER_32K_NRESPWRON). When COUNTER_32K_NRESPWRON is released (on the rising edge of COUNTER_32K_FCLK), the counter starts counting up from the reset value of the counter register on the falling edge of the 32-kHz COUNTER_32K_FCLK clock after three inverted 32-kHz clock periods. After reaching its highest value, the counter wraps back to 0 and starts counting again with no additional delay.

Figure 17-16 shows the reset synchronization timing diagram.

Figure 17-16. Reset Resynchronization Timing Diagram

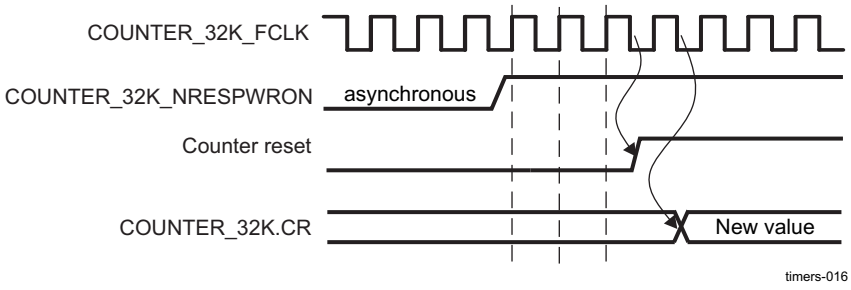
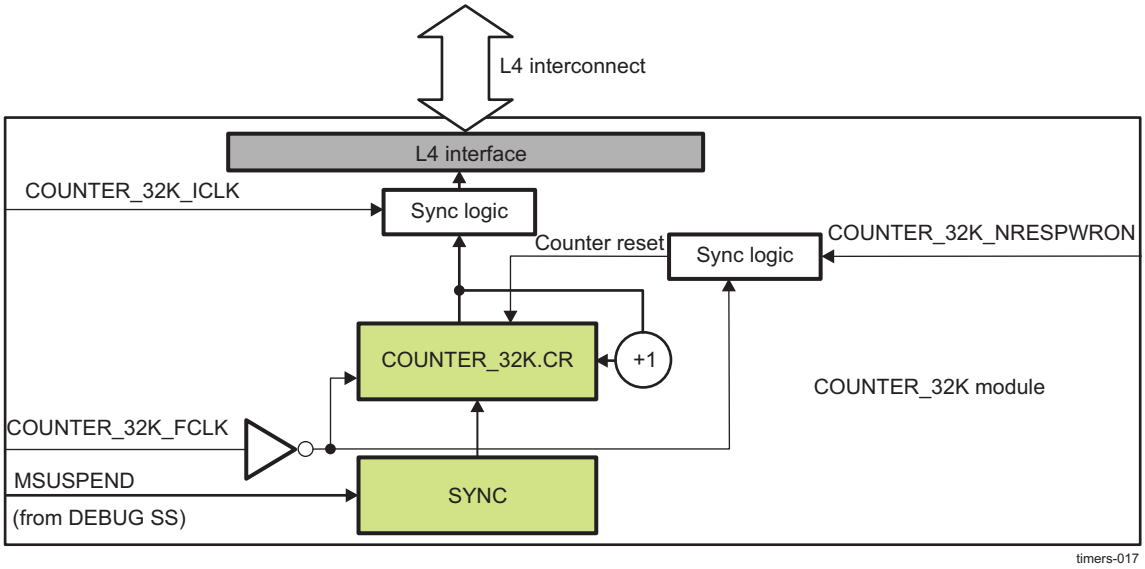


Figure 17-17 is the block diagram of the synchronized timer.

Figure 17-17. CONTER_32K Block Diagram



The sync logic ensures the correctness of the read transaction by synchronizing the counter register read access on COUNTER_32K_ICLK, because the COUNTER_32K_ICLK clock signal is completely asynchronous with COUNTER_32K_FCLK. The COUNTER_32K_NRESPWRON input resets the counter register (CR). The inverted COUNTER_32K_FCLK clocks the counter register CR.

The counting is temporally stopped when MSUSPEND control signal (coming from DEBUG SS) is asserted.

17.3.3.1 Reading the 32-kHz Synchronized Timer

The counter register (CR) is 32 bits wide. For correct count capture, it must be accessed as 16-bit LSB access first and 16-bit MSB access next. The value of the counter is read through the L4 interconnect slave interface. Internal synchronization logic allows the reading of the counter value with COUNTER_32K_ICLK while the counter is running. The time latency to read the synchronized counter register is one COUNTER_32K_ICLK clock period.

The user can select between two syncronization schemes by setting SYSCONFIG[0]SYNCMODE bit.

- SYSCONFIG[0]SYSCMODE = 0x0 - default. In this mode COUNTER_32K timer uses Gray encode/decode scheme. When the L4 interface sends a LSB16 read request command, the 32 bit coded value is registered directly to the interface domain. Due to the characteristics of this encoding if a read command arrives during a count up event either the old or the new value of CR is captured, not a transient value. The captured value will be decoded and send on the SDATA bus. The MSB16 read command reads upper 16 bits of the 32 bit value of counter register captured during the last LSB16 read access.
- SYSCONFIG[0]SYNCMODE = 0x1 - legacy synchronization scheme. In this mode the value of the CR is synchronized to the OCP domain at every count up event. This synchronization is possible because the COUNTER_32K_ICLK is much faster that the 32KHz clock (COUNTER_32K_FCLK). The drawback of this method is that if the interface clock is switched back after wake up from idle mode, the syncronized value will be updated only at the next count up event , until then it will be incorrect.

17.3.4 COUNTER_32K Timer Register Manual

Table 17-74 lists the base address and block size for the 32-kHz synchronized timer. It is memory-mapped to the L4 peripheral bus memory space.

Table 17-74. COUNTER_32KTimer Instance Summary

Module Name	Module Base Address	Size
L4_WKUP_COUNTER_32K	0x4AE0 4000	52 Bytes

17.3.4.1 COUNTER_32K Timer Register Mapping Summary

CAUTION

The 32-kHz synchronized timer registers are limited to 32- and 16-bit data accesses; 8-bit access is not allowed and can corrupt the register content.

Table 17-75 lists the 32-kHz synchronized timer registers. Table 17-76 through Table 17-80 describe the register bits.

Table 17-75. COUNTER_32KTimer Register Summary

Register Name	Type	Register Width (Bits)	Address Offset	L4_WKUP_COUNTER_32K Base Address
REVISION	R	32	0x0000 0000	0x4AE0 4000
SYSCONFIG	RW	32	0x0000 0010	0x4AE0 4010
CR	R	32	0x0000 0030	0x4AE0 4030

17.3.4.2 COUNTER_32K Timer Register Description

Table 17-76. REVISION

Address Offset	0x0 0000
Physical Address	0x4AE0 4000
Description	This register contains the sync counter IP revision code.
Type	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REVISION																															

Bits	Field Name	Description	Type	Reset
31:0	REVISION	IP revision	R	0x-(1)

(1) TI internal data

Table 17-77. Register Call Summary for Register REVISION

32-kHz Synchronized Timer (COUNTER_32K)

- COUNTER_32K Timer Register Mapping Summary: [0]
- COUNTER_32K Timer Register Description: [1]

Table 17-78. SYSCONFIG

Address Offset	0x0 0010
Physical Address	0x4AE0 4010
Description	This register is used for idle modes only.
Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Reserved																											IDLEMODE		Reserved		SYNCMODE

Bits	Field Name	Description		Type	Reset
31:5	Reserved	Reads return 0.		R	0x0
4:3	IDLEMODE	Power management REQ/ACK control		RW	0x0
		0x0:	Force-idle. An IDLE request is acknowledged unconditionally.		
		0x1:	No-idle. An IDLE request is never acknowledged.		
		0x2:	Reserved		
		0x3:	Reserved		
2:1	Reserved	Reads return 0.		R	0x0
0	SYNCMODE	Synchronization scheme		RW	0x0
		0x0	Gray synchronization scheme. Ensures that a stable value of the CR register is read.		
		0x1	Legacy synchronization scheme.		

Table 17-79. Register Call Summary for Register SYSCONFIG

32-kHz Synchronized Timer (COUNTER_32K)

- Reading the 32-kHz Synchronized Timer: [0][1][2]
- COUNTER_32K Timer Register Mapping Summary: [3]

Table 17-80. CR

Address Offset	0x0 0030
Physical Address	0x4AE0 4030
Description	This register contains the 32-kHz sync counter value.
Type	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
COUNTER_VALUE																															

Bits	Field Name	Description	Type	Reset
31:0	COUNTER_VALUE	Counter register value	R	0x00000003

Table 17-81. Register Call Summary for Register CR

32-kHz Synchronized Timer (COUNTER_32K)

- 32-kHz Synchronized Timer Functional Description: [0]
- Reading the 32-kHz Synchronized Timer: [1][2][3]
- COUNTER_32K Timer Register Mapping Summary: [4]
- COUNTER_32K Timer Register Description: [5]

Serial Communication Interfaces

This chapter describes the features and operation of the Serial Communication Interfaces (SCI) for the device.

Topic **Page**

18.1 Multimaster I²C Controller

This section describes the two inter-integrated circuit (I²C) controller modules in the device.

NOTE: The device I2C controllers do not support high-speed (HS) mode. See the Device Data Manual for details.

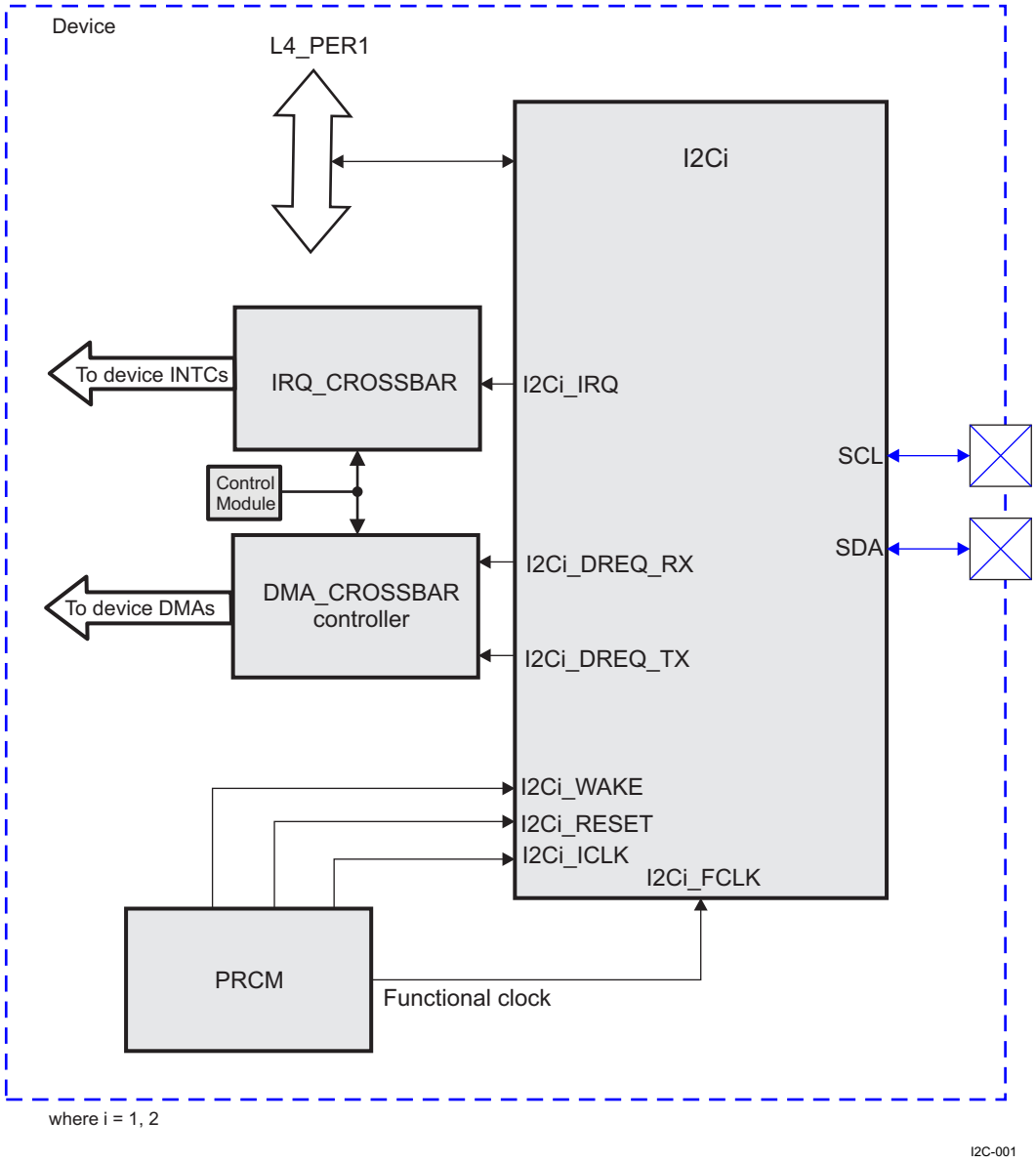
18.1.1 I²C Overview

The device contains two multimaster inter-integrated circuit (I²C) controllers (I2C*i* modules, where *i* = 1, 2) each of which provides an interface between a local host (LH), such as a digital signal processor (DSP), and any I²C-bus-compatible device that connects through the I²C serial bus. External components attached to the I²C bus can serially transmit and receive up to 8 bits of data to and from the LH device through the 2-wire I²C interface.

Each multimaster I²C controller can be configured to act like a slave or master I²C-compatible device.

Figure 18-1 shows the I²C.

Figure 18-1. I²C Controllers



The multimaster I²C controllers have the following features:

- Compliant with Philips I²C specification version 2.1
- Supports a standard mode (up to 100 kbps) and fast mode (up to 400 kbps)
- 7-bit and 10-bit device addressing modes
- General call
- Start/Restart/Stop
- Multimaster transmitter/slave receiver mode
- Multimaster receiver/slave transmitter mode
- Combined master transmit/receive and receive/transmit mode
- Built-in FIFOs (16 bytes) for buffered read or write
- Module enable/disable capability
- Programmable multislave channel (responds to four separate addresses)
- Programmable clock generation

- 8-bit-wide data access
- Designed for low power consumption
- Implement Auto Idle mechanism
- Implement Idle Request/Idle Acknowledge handshake mechanism
- Support for asynchronous wakeup mechanism
- Two direct memory access (DMA) channels
- Wide interrupt capability

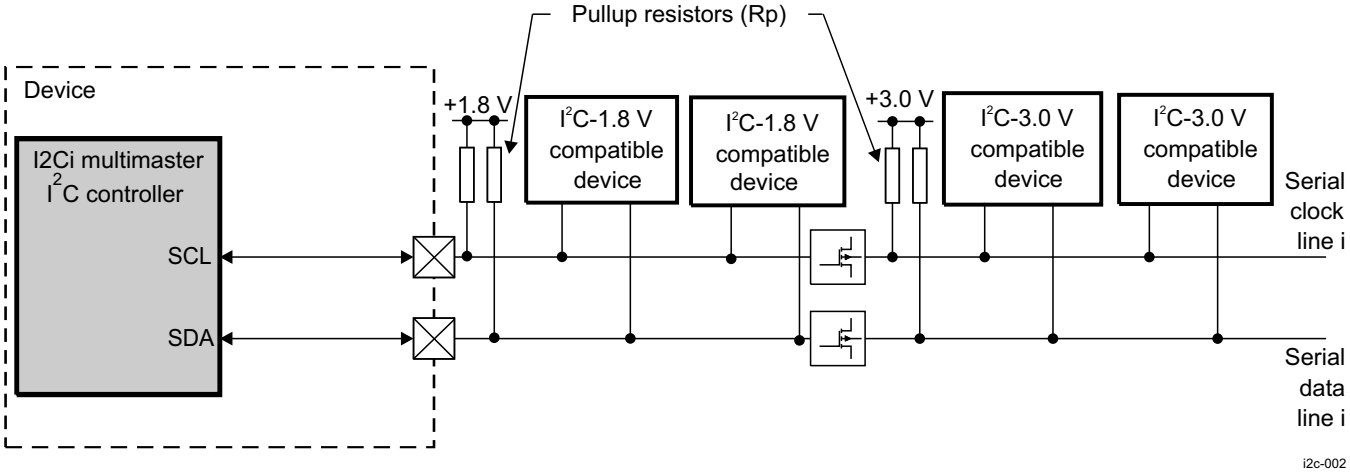
18.1.2 I²C Environment

This section describes the I²C application fields from an environment point of view (external connections). It describes I²C connectivity options, lists all possible interfaces, and describes the protocol and data format used in each case.

18.1.2.1 I²C Typical Application

Figure 18-2 shows the multimaster I²C controllers and their related connections with I²C-compliant devices.

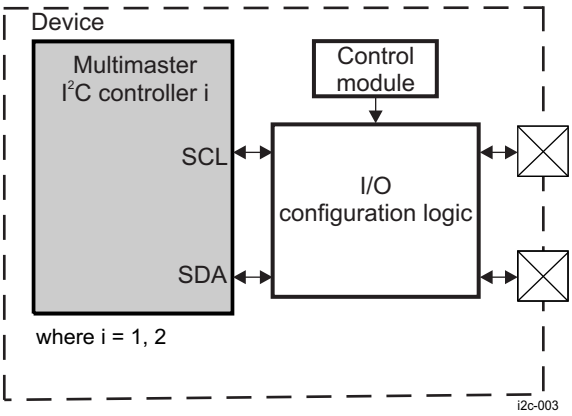
Figure 18-2. I²C and Typical Connections to I²C Devices



18.1.2.1.1 I²C Pins for Typical Connections in I²C Mode

Figure 18-3shows the multimaster I²C controller pins used for typical connections with I²C devices.

Figure 18-3. I²C Interface Signals



18.1.2.1.2 I²C Interface Typical Connections

Table 18-1 lists the pins associated with the I²C interface.

Table 18-1. I²C Input/Output

Signal	Device Level Signal	I/O[1]	Description	Reset Value
SCL	i2c*j*_scl[2]	I/O	I²C serial clock line.	1

(1) I = Input; O = Output
(2) j= 1 and 2

Table 18-1. I²C Input/Output (continued)

Signal	Device Level Signal	I/O[1]	Description	Reset Value
SDA	i2c*j*_sda[2]	I/O	I²C serial data line.	1

NOTE: For the i2c*j*_scl and i2c*i*_scl signals to work properly, the INPUTENABLE bit of the appropriate CTRL_CORE_PAD_x registers should be set to 0x1 because of retiming purposes.

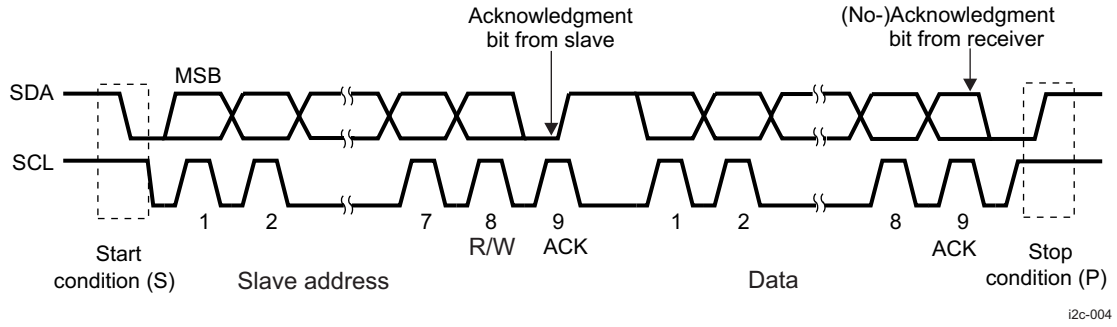
NOTE: The path from a module pin to device pad(s) is defined at the device I/O logic level. The I/O logic maps the module signals to the different pads of the device and is programmable in the Control Module registers. Refer to the sections Section 13.4.6.1 *Pad Functional Multiplexing* of the chapter *Control Module*, for more information.

18.1.2.2 I²C Typical Connection Protocol and Data Format

18.1.2.2.1 I²C Serial Data Format

The I²C controller operates in 8-bit word data format (byte write access supported for the last access). Each byte transmitted or received on the serial data line is 8 bits long. The number of bytes that can be transmitted or received is not restricted. The data is transferred with the most-significant bit (MSB) first. In receiver mode, each byte is followed by an acknowledge bit from the I²C. Figure 18-4 shows a typical I²C communication format.

Figure 18-4. I²C Data Transfer

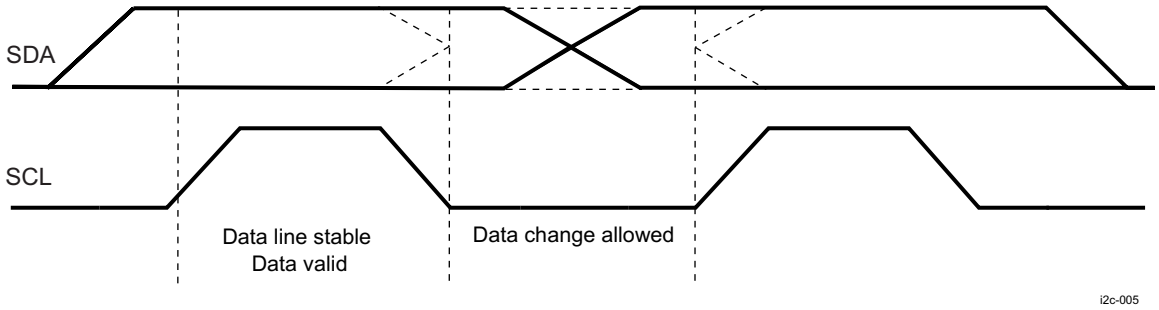


18.1.2.2.2 I²C Data Validity

The data on the serial data line (SDA) must be stable during the high period of the serial clock line. The high and low states of the data line can change only when the clock signal on the serial clock line (SCL) is low.

Figure 18-5 is an example of data validity requirements.

Figure 18-5. I²C Bit Transfer on the I²C Bus



18.1.2.2.3 I²C Start and Stop Conditions

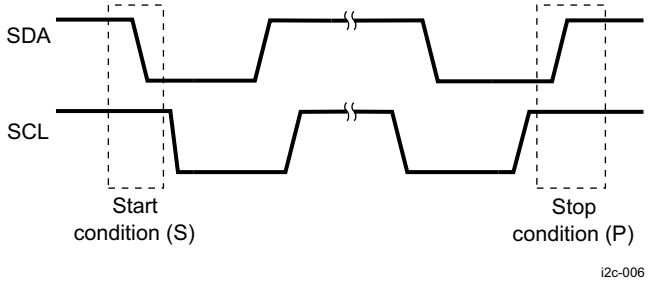
The I²C module generates start (S) and stop (P) conditions when it is configured as a master.

- An S condition is a high-to-low transition on the serial data line while serial clock line is high.
- A P condition is a low-to-high transition on the serial data line while serial clock line is high.

The bus is considered busy after the S condition (the I2C*i*. [12] BB bit is 1 to indicate that the bus is busy) and free after the P condition (the I2C*i*.I2C_IRQSTATUS_RAW [12] BB bit is 0 to indicate that the bus is free).

Figure 18-6 shows the waveforms that occur during an S and a P condition.

Figure 18-6. I²C S and P Condition Events



NOTE: I²C controller does not support messages non-compliant with I2C standard. Void messages are non-standard I²C messages and will lockup the controller. A void message is a START condition followed by a STOP condition, in other words, while the bus is free the SDA line is pulled low (START) and then released (STOP). This would result in a timeout (software) of the next master transfer which would never complete. A soft reset of the controller is recommended for recovery.

18.1.2.2.4 I²C Addressing

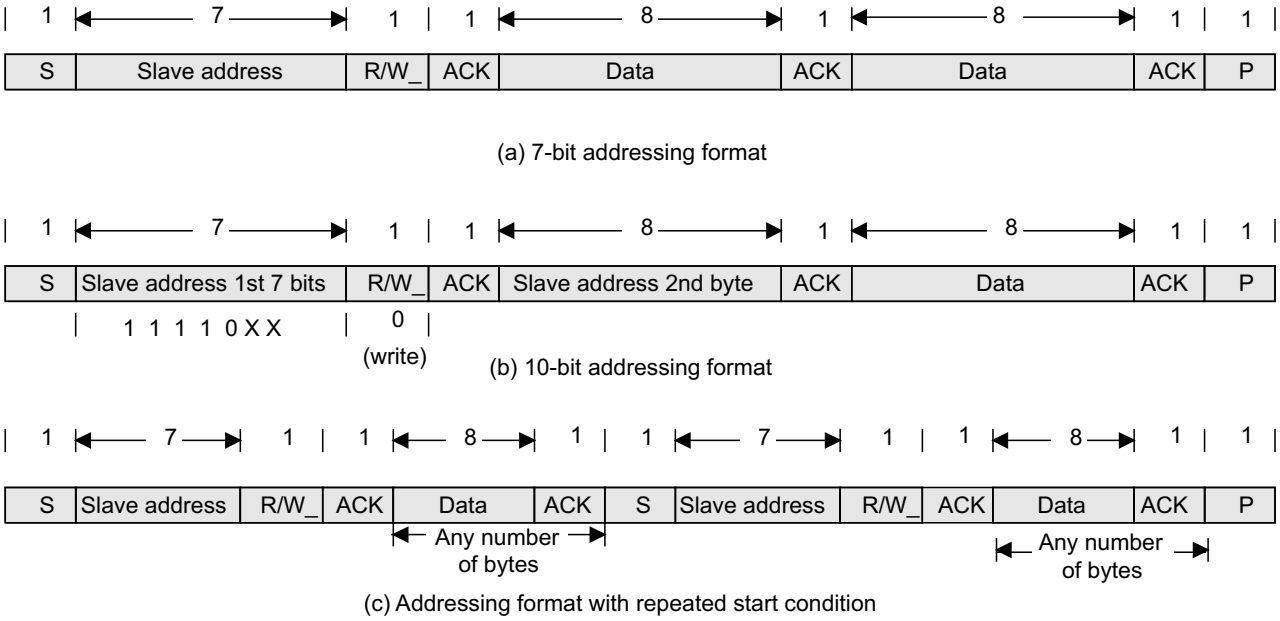
The I²C module supports two data formats in fast/standard (F/S) modes:

- 7-bit/10-bit addressing format
- 7-bit/10-bit addressing format with repeated start (Sr) condition

18.1.2.2.4.1 Data Transfer Formats in F/S Mode

Figure 18-7 shows the I²C data transfer formats in F/S mode.

Figure 18-7. I²C Data Transfer Formats in F/S Mode



The first word after an S condition consists of 8 bits. In acknowledge mode, an extra dedicated acknowledgment bit is inserted after each byte.

In addressing formats with 7-bit addresses, the first byte is composed of 7 MSB slave address bits and 1 least-significant bit (LSB) R/W_ bit.

The LSB R/W_ bit of the address byte indicates the transmission direction of the data bytes that follow it. If R/W_ is 0, the master writes data to the selected slave; if it is 1, the master reads data from the slave.

In addressing formats with 10-bit addresses, the structure of the first byte is 11110XXY, where XX is the two MSBs of the 10-bit addresses, and Y is the R/W_ bit. If the R/W_ bit is 0, the next byte contains the last 8 bits of the slave address. If the R/W_ bit is 1, the next byte contains data transmitted from the slave to the master.

18.1.2.2.5 I²C Master Transmitter

In master transmitter mode, data assembled in one of the previously described data formats is shifted out on the serial data line SDA in sync with the self-generated clock pulses on the serial clock line SCL. The clock pulses are inhibited and SCL is held low when the intervention of the processor is required (XUDF) after a byte is transmitted.

18.1.2.2.6 I²C Master Receiver

Master receiver mode can be entered only from master transmitter mode. With any of the address formats (a), (b), or (c) (see Figure 18-7), if R/W_ is high, the module enters master receiver mode after the slave address byte and bit R/W_ are transmitted. Serial data bits received on bus line SDAI are shifted in synchronization with the self-generated clock pulses on SCLI.

18.1.2.2.7 I²C Slave Transmitter

Slave transmitter mode can be entered only from slave receiver mode. With any of the address formats (a), (b), or (c) (see Figure 18-7), the slave transmitter is entered if the slave address byte is the same as its own address and bit R/W_ is transmitted, if R/W_ is high. The slave transmitter shifts the serial data out on the data line SDA in sync with the clock pulses that are generated by the master device. It does not generate the clock but it can hold clock line SCL low while intervention of the LH is required (XUDF).

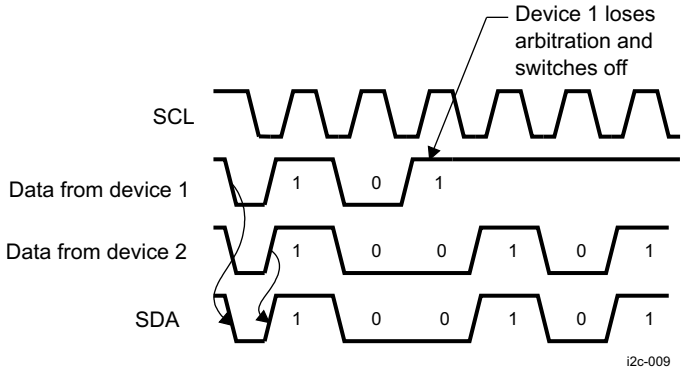
18.1.2.2.8 I²C Slave Receiver

In this mode, serial data bits received on the bus line SDA are shifted-in in sync with the clock pulses on SCL that are generated by the master device. It does not generate the clock but it can hold clock line SCL low while intervention of the LH is required (ROVR) after a byte is received.

18.1.2.2.9 I²C Bus Arbitration

If two or more master transmitters start a transmission on the same bus almost simultaneously, an arbitration procedure is invoked. The arbitration procedure uses the data presented on the serial bus by the competing transmitters. When a transmitter senses that a high signal it has presented on the bus has been overruled by a low signal, it switches to the slave receiver mode, sets the arbitration lost (AL) flag, and generates the arbitration lost interrupt. Figure 18-8 shows the arbitration procedure between two devices. The arbitration procedure gives priority to the device that transmits the serial data stream with the lowest binary value. If two or more devices send identical first bytes, arbitration continues on the subsequent bytes.

Figure 18-8. I²C Arbitration Between Master Transmitters



18.1.2.2.10 I²C Clock Generation and Synchronization

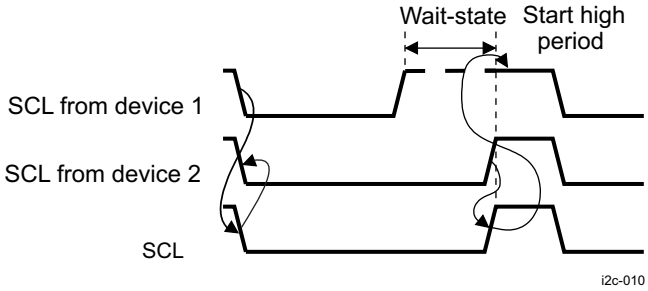
Under normal conditions, only one master device generates the clock signal, SCL. However, there are two or more master devices during the arbitration procedure, and the clock must be synchronized so that the data output can be compared. The wired-AND property of the clock line means that a device that first generates a low period of the clock line overrules the other devices. At this high/low transition, the clock generators of the other devices are forced to start generation of their own low period. The clock line is then held low by the device with the longest low period, while the other devices that finish their low periods must wait for the clock line to be released before starting their high periods. A synchronized signal on the clock line is thus obtained, where the slowest device determines the length of the low period and the fastest device determines the length of the high period. If a device pulls down the clock line for a longer time, the result is that all clock generators must enter the WAIT-state. In this way a slave can slow down a fast master and the slow device can create enough time to store a received byte or prepare a byte to be transmitted (clock stretching).

NOTE: In case the SCL or SDA lines are stuck low, a bus clearing operation is supported:

- If the clock line (SCL) is stuck low, the preferential procedure is to reset the bus using the hardware reset signal if your I²C devices have hardware reset inputs. If the I²C devices do not have hardware reset inputs, cycle power to the devices to activate the mandatory internal power-on reset (POR) circuit.
- If the data line (SDA) is stuck low, the master should send nine clock pulses. The device that held the bus low should release it sometime within those nine clocks. If not, then use the hardware reset or cycle power to clear the bus.

Figure 18-9 shows clock synchronization.

Figure 18-9. I²C Clock Generators Synchronization

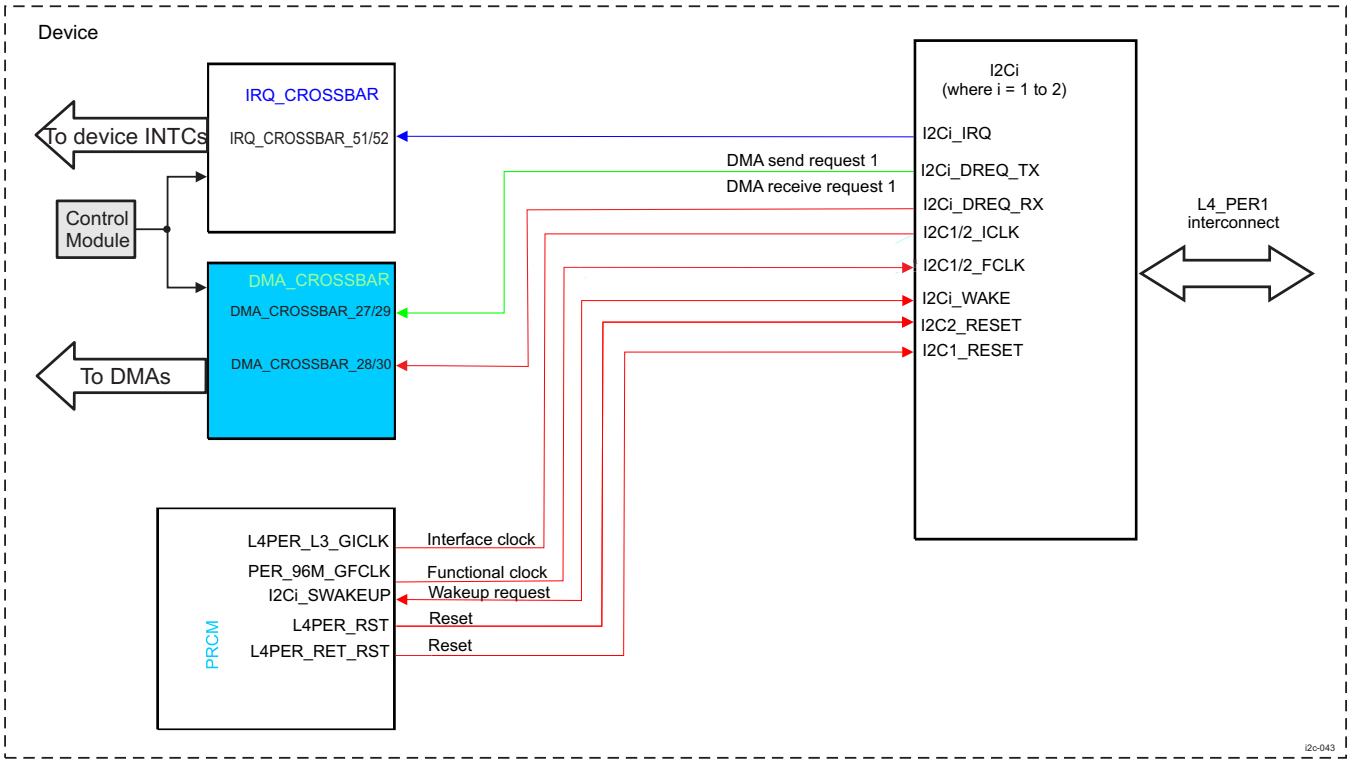


18.1.3 I²C Integration

This section describes module integration in the device, including information about clocks, resets, and hardware requests.

Figure 18-10shows the integration of the two I²C controllers in the device.

Figure 18-10. I²C Integration



NOTE: For more information about the slave idle protocol and the wake-up request, see Section 3.7, *Power Management Functional Description*, in Chapter 3, *Power, Reset, and Clock Management*.

Table 18-2 through Table 18-4 summarize the integration of the module in the device.

Table 18-2. I²C Integration Attributes

Module Instance	Attributes	
	Power Domain	Interconnect
I2C1	PD_COREAON	L4_PER1
I2C2	PD_COREAON	L4_PER1

Table 18-3. I²C Clocks and Resets

Clocks				
Module Instance	Destination Signal Name	Source Signal Name	Source	Description
I2C1	I2C1_ICLK	L4PER_L3_GICLK	PRCM	I2C1 interface clock
	I2C1_FCLK	PER_96M_GFCLK	PRCM	I2C1 functional clock
I2C2	I2C2_ICLK	L4PER_L3_GICLK	PRCM	I2C2 interface clock
	I2C2_FCLK	PER_96M_GFCLK	PRCM	I2C2 functional clock
Resets				

Table 18-3. I²C Clocks and Resets (continued)

Module Instance	Destination Signal Name	Source Signal Name	Source	Description
I2C1	I2C1_RESET	L4PER_RET_RST	PRCM	I2C1 reset
I2C2	I2C2_RESET	L4PER_RST	PRCM	I2C2 reset

Table 18-4. I²C Hardware Requests

Interrupt Requests

Module Instance	Source Signal Name	Destination IRQ_CROSSBAR Input	Default Mapping	Destination
I2C1	I2C1_IRQ	IRQ_CROSSBAR_51	-	I2C1 interrupt request
			DSP1_IRQ_82	
			DSP2_IRQ_82	
			IPU1_IRQ_41	
I2C2	I2C2_IRQ	IRQ_CROSSBAR_52	-	I2C2 interrupt request
			DSP1_IRQ_83	
			DSP2_IRQ_83	
			IPU1_IRQ_42	

DMA Requests

Module Instance	Source Signal Name	Destination DMA_CROSSBAR Input	Default Mapping	Description
I2C1	I2C1_DREQ_TX	DMA_CROSSBAR_27	DMA_EDMA_DREQ_26	I2C1 DMA transmit request
	I2C1_DREQ_RX	DMA_CROSSBAR_28	DMA_EDMA_DREQ_27	I2C1 DMA receive request
I2C2	I2C2_DREQ_TX	DMA_CROSSBAR_29	DMA_EDMA_DREQ_28	I2C2 DMA transmit request
	I2C2_DREQ_RX	DMA_CROSSBAR_30	DMA_EDMA_DREQ_29	I2C2 DMA receive request

NOTE: The "**Default Mapping**" column in Table 18-4 *I²C Hardware Requests* shows the default mapping of module IRQ and DREQ source signals. These module IRQ and DREQ source signals can also be mapped to other lines of each device Interrupt or DMA controller through the IRQ_CROSSBAR and DMA_CROSSBAR modules, respectively.
For more information about the IRQ_CROSSBAR module, see Section 13.4.6.4, *IRQ_CROSSBAR Module Functional Description*, in Chapter 13, *Control Module*.
For more information about the DMA_CROSSBAR module, see Section 13.4.6.5, *DMA_CROSSBAR Module Functional Description*, in Section 13.4.6.5Chapter 13, *Control Module*.
For more information about the device interrupt controllers, see Chapter 12, *Interrupt Controllers*.
For more information about the device EDMA module, see Chapter 11, *Enhanced DMA*.

NOTE:

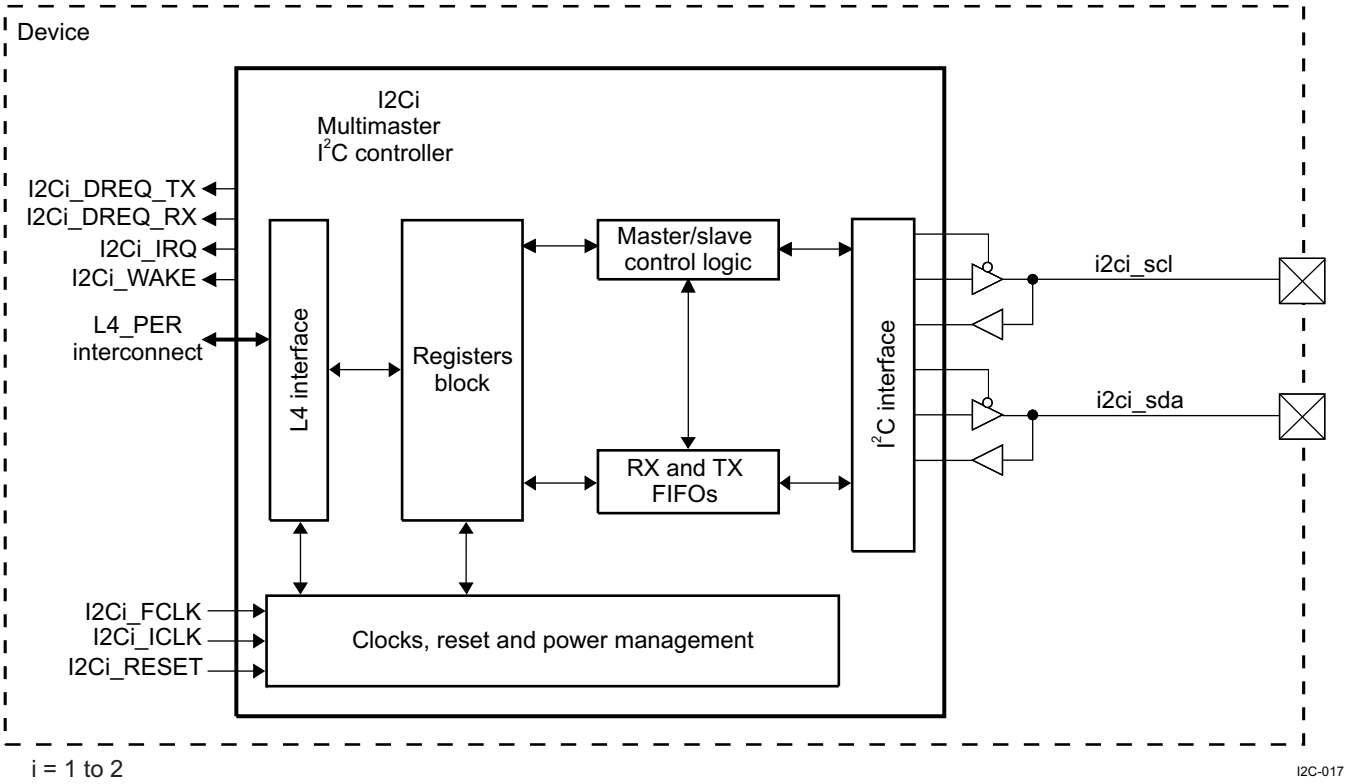
- For a description of interrupt source, see Table 18-10.
- For a description of DMA source, see Table 18-11.

18.1.4 I²C Functional Description

18.1.4.1 I²C Block Diagram

Figure 18-11 is the multimaster I²C controller block diagram.

Figure 18-11. I²C Block Diagram



The two multimaster I²C controllers can be configured in F/S I²C mode . The operation mode is selected by configuring the I2C*i*.I2C_CON[13:12] OPMODE bit field. Table 18-5 lists the available operation modes.

Table 18-5. I²C Operation Mode Selection

Operation Mode	Value of I2C*i*.I2C_CON[13:12] OPMODE Bit Field
F/S I²C	0x0
Reserved	0x1
Reserved	0x2
Reserved (not used)	0x3

18.1.4.2 I²C Clocks

18.1.4.2.1 I²C Clocking

Each multimaster I²C controller uses the I2C*i*_FCLK functional clock in the PRCM module. The internal sampling clock I2C*i*_INTERNAL_CLK is generated by dividing the functional clock by the I2C*i*.I2C_PSC[7:0] PSC bit field value + 1 in F/S mode.

The low time of the I2C*i*_SCLL signal is determined by the I2C*i*.I2C_SCLL[7:0] SCLL bit field in F/S mode.

The high time of the I2C*i*_SCLL signal is determined by the I2C*i*.I2C_SCLH[7:0] SCLH bit field in F/S mode.

Table 18-6 lists the t_{LOW} and t_{high} values in master mode only (in slave mode, the I²C controller does not generate the I²C clock).

Table 18-6. I²C t_{LOW} and t_{high} Values of the I²C Clock

Mode	I2C*i* Clock	t_{LOW}	t_{high}
F/S first phase	I2C*i*_INTERNAL_CLK = I2C*i*_FCLK / (I2C*i*.I2C_PSC[7:0] PSC bit field + 1)	(I2C*i*.I2C_SCLL[7:0] SCLL bit field value + 7) x I2C*i*_INTERNAL_CLK period	(I2Ci.I2C_SCLH [7:0] SCLH bit filed value + 5) x I2Ci_INTERNAL_CLK period

NOTE: The equations in Table 18-6 give the SCLL timing values for SCLL/SCLH/HSSCLL/HSSCLH at I²C controller outputs. Actual tlow and thigh periods may vary depending on the board (the load capacitance on the SCLL signal). If necessary, any adjustments to the SCLL/SCLH/HSSCLL/HSSCLH values must be determined by measurements of actual SCL signal on the board.

CAUTION

During active mode (the I2C*i*.I2C_CON[15] I2C_EN bit is set to 1), make no changes to the I2C*i*.I2C_SCLL and I2C*i*.I2C_SCLH registers. Changes may result in unpredictable behavior.

Table 18-7 lists the register values for obtaining the maximum I²C bit rates and the maximum period of the filtered spikes in F/S modes.

Table 18-7. I²C Register Values for Maximum I²C Bit Rates in I²C F/S Modes(1)

	I²C Mode for I2C*i*,		Description
	Standard Mode	**Fast Mode**	
I2C*i*_FCLK frequency (MHz)	96		
I2C *i* .I2C_PSC[7:0] PSC bit field value	**23**	**9**	Prescaler value for F/S modes
I2C*i*_INTERNAL_CLK frequency (MHz)	4	9.6	
I2C *i* .I2C_SCLL[7:0] SCLL bit field value	**13**	**5**	Value for F/S mode
I2C *i* .I2C_SCLH[7:0] SCLH bit field value	**15**	**7**	Value for F/S mode
Maximum bit rate (Mbps)	0.1	0.4	F/S mode maximum bit rate
Maximum filter period (ns)	250	104.2	
Maximum filter period (ns)			

(1) Programmable fields are in bold.

NOTE: This table presents informative values only for the configuration parameters and the I²C bus performance obtained according to these values. The delays added by the analog pads are not considered in these figures.

NOTE: For I2Ci (where i=1, 2)

I2Ci_INTERNAL_CLK freq = I2Ci_FCLK / (PSC +1)

F/S filter period = 1 / I2Ci_INTERNAL_CLK

FS bit rate = I2Ci_INTERNAL_CLK / (SCLL+ 7 + SCLH + 5)

18.1.4.2.2 I²C Automatic Blocking of the I²C Clock Feature

This feature offers the possibility for the LH to command the blocking of the I²C clock after the slave addressing phase, when the I²C controller is addressed by an external master device using a certain Own Address.

The release of the I²C clock can be performed independently for each Own Address (I2C*i*.I2C_OA, and I2C*i*.I2C_OAx registers, where *i* = 1 to 2, x = 1, 2, 3) by deasserting the corresponding bit in the I2C*i*.I2C_SBLOCK register.

18.1.4.3 I²C Software Reset

Each multimaster I²C controller supports the software reset by accessing the I2C*i*.I2C_SYSC[1] SRST bit (1: reset; 0: normal mode).

The software reset status can be checked by accessing the I2C*i*.I2C_SYSS[0] RDONE bit (1: reset is done; 0: reset is ongoing).

To do a software reset, the following steps must be done:

1. Ensure that the module is disabled (clear the I2C*i*.I2C_CON[15] I2C_EN bit to 0).
2. Set the I2C*i*.I2C_SYSC[1] SRST bit to 1.
3. Enable the module by setting I2C*i*.I2C_CON[15] I2C_EN bit to 1.
4. Check the I2C*i*.I2C_SYSS[0] RDONE bit until it is set to 1 to indicate the software reset is complete.

NOTE: The I2C*i*.I2C_CON[15] I2C_EN bit can hold the functional clock domain of the multimaster I²C controller in reset after the device reset has been released. When the system bus reset is removed, this bit remains cleared. The functional part of the I²C controller is held in reset state while this bit is 0, and all configuration registers can be accessed.

The I2C*i*.I2C_CON[15] I2C_EN bit must be set to 1 to enable the functional part of the I²C controller.

The I2C*i*.I2C_SYSS[0] RDONE bit is asserted only after the module is enabled by setting the I2C*i*.I2C_CON[15] I2C_EN bit to 1.

18.1.4.4 I²C Power Management

Table 18-8 describes power-management features available for the multimaster I²C controllers.

NOTE:

- For information about source clock gating and sleep/wake-up transitions description, see Section 3.1.1.1.2, *Module-Level Clock Management*, in Chapter 3, *Power, Reset, and Clock Management*.
- For descriptions of EnaWakeUp, IdleMode, ClockActivity, and StandbyMode features, see Section 3.1.1.1.2, *Module-Level Clock Management*, in Chapter 3, *Power, Reset, and Clock Management*.

Table 18-8. I²C Local Power-Management Features

Feature	Registers	Description
Clock auto gating	I2C*i*.I2C_SYSC[0] AUTOIDLE	This bit allows a local power optimization inside the module.
Slave idle modes	I2C*i*.I2C_SYSC[4:3] IDLEMODE	Force-idle, no-idle, smart-idle, and smart-idle wakeup-capable modes are available.
Clock activity	I2C*i*.I2C_SYSC[9:8] CLOCKACTIVITY	For configuration details, see Table 18-9.
Global wake-up enable	I2C*i*.I2C_SYSC[2] ENAWAKEUP	This bit enables the wake-up feature at module level.

NOTE: The voltage controllers, in which the I²C controller is implemented, have no idle request/acknowledge mechanism. The idle modes for the voltage controllers are directly managed by the PRCM module.

Table 18-9. I²C Clock Activity Settings

I2C*i*.I2C_SYSC[9:8] CLOCKACTIVITY	Clock State When Module is in IDLE State		Features Available/Unavailable When Module is in IDLE State
	I2C*i*_ICLK	I2C*i*_ FCLK	
00	OFF	OFF	Both clocks are disabled.
10	OFF	ON	Interface clock is disabled; functional clock is enabled
01	ON	OFF	Functional clock is disabled; interface clock is enabled
11	ON	ON	Both clocks are enabled.

CAUTION

The PRCM module has no hardware means of reading the settings of CLOCKACTIVITY. Thus, software must ensure consistent programming between the I²C CLOCKACTIVITY and I²C clock PRCM control bits. For a description of the ClockActivity feature, see Section 3.1.1.1.2, *Module-Level Clock Management* in Chapter 3, *Power, Reset, and Clock Management*.

18.1.4.5 I²C Interrupt Requests

Table 18-10 lists the event flags, and their mask, that can cause module interrupts.

Table 18-10. I²C Events

Event Flag	Event Unmask	Event Mask	Map to	Description
I2C*i*.I2C_IRQSTATUS[0] AL	I2C*i*.I2C_IRQENABLE_SET[0] AL_IE	I2C*i*.I2C_IRQENABLE_CLR[0] AL_IE	I2C*i*_IRQ	Arbitration lost IRQ enabled status. This bit is automatically set by the hardware when it loses the Arbitration in master transmit mode, an interrupt is signaled to the IRQ_Crossbar. During reads, it always returns 0.
I2C*i*.I2C_IRQSTATUS[1] NACK	I2C*i*.I2C_IRQENABLE_SET[1] NACK_IE	I2C*i*.I2C_IRQENABLE_CLR[1] NACK_IE	I2C*i*_IRQ	No acknowledgement IRQ enabled status. Bit is set when No Acknowledge is received, an interrupt is signaled to the IRQ_Crossbar. Write 1 to clear this bit.
I2C*i*.I2C_IRQSTATUS[2] ARDY	I2C*i*.I2C_IRQENABLE_SET[2] ARDY_IE	I2C*i*.I2C_IRQENABLE_CLR[2] ARDY_IE	I2C*i*_IRQ	Register access ready IRQ enabled status. When set to 1 it indicates that previous access has been performed and registers are ready to be accessed again. An interrupt is signaled to the IRQ_Crossbar. Write 1 to clear.
I2C*i*.I2C_IRQSTATUS[3] RRDY	I2C*i*.I2C_IRQENABLE_SET[3] RRDY_IE	I2C*i*.I2C_IRQENABLE_CLR[3] RRDY_IE	I2C*i*_IRQ	Receive data ready IRQ enabled status. Set to 1 by core when receiver mode, a new data can be read. When set to 1 by core, an interrupt is signaled to the IRQ_Crossbar. Write 1 to clear.

Table 18-10. I²C Events (continued)

Event Flag	Event Unmask	Event Mask	Map to	Description
I2C*i*.I2C_IRQSTATUS[4] XRDY	I2C*i*.I2C_IRQENABLE_SET[4] XRDY_IE	I2C*i*.I2C_IRQENABLE_CLR[4] XRDY_IE	I2C*i*_IRQ	Transmit data ready IRQ enabled status. Set to 1 by core when transmitter and when new data is requested. When set to 1 by core, an interrupt is signaled to the IRQ_Crossbar. Write 1 to clear.
I2C*i*.I2C_IRQSTATUS[5] GC	I2C*i*.I2C_IRQENABLE_SET[5] GC_IE	I2C*i*.I2C_IRQENABLE_CLR[5] GC_IE	I2C*i*_IRQ	General call IRQ enabled status. Set to 1 by core when General call address detected and interrupt signaled to the IRQ_Crossbar. Write 1 to clear.
I2C*i*.I2C_IRQSTATUS[6] STC	I2C*i*.I2C_IRQENABLE_SET[6] STC_IE	I2C*i*.I2C_IRQENABLE_CLR[6] STC_IE	I2C*i*_IRQ	Start condition IRQ enabled status.
I2C*i*.I2C_IRQSTATUS[7] AERR	I2C*i*.I2C_IRQENABLE_SET[7] AERR_IE	I2C*i*.I2C_IRQENABLE_CLR[7] AERR_IE	I2C*i*_IRQ	Access Error IRQ enabled status.
I2C*i*.I2C_IRQSTATUS[8] BF	I2C*i*.I2C_IRQENABLE_SET[8] BF_IE	I2C*i*.I2C_IRQENABLE_CLR[8] BF_IE	I2C*i*_IRQ	Access Error IRQ enabled status.
I2C*i*.I2C_IRQSTATUS[9] AAS	I2C*i*.I2C_IRQENABLE_SET[9] AAS_IE	I2C*i*.I2C_IRQENABLE_CLR[9] AAS_IE	I2C*i*_IRQ	Address recognized as slave IRQ enabled status.
I2C*i*.I2C_IRQSTATUS[10] XUDF	I2C*i*.I2C_IRQENABLE_SET [10] XUDF_IE	I2C*i*.I2C_IRQENABLE_CLR[10] XUDF_IE	I2C*i*_IRQ	Transmit underflow enabled status. Writing into this bit has no effect.
I2C*i*.I2C_IRQSTATUS[11] ROVR	I2C*i*.I2C_IRQENABLE_SET [11] ROVR_IE	I2C*i*.I2C_IRQENABLE_CLR[11] ROVR_IE	I2C*i*_IRQ	Receive overrun enabled status. Writing into this bit has no effect.
I2C*i*.I2C_IRQSTATUS[12] BB	Reserved	Reserved	I2C*i*_IRQ	Bus busy enabled status. Writing into this bit has no effect.
I2C*i*.I2C_IRQSTATUS[13] RDR	I2C*i*.I2C_IRQENABLE_SET [13] RDR_IE	I2C*i*.I2C_IRQENABLE_CLR[13] RDR_IE	I2C*i*_IRQ	Receive draining IRQ enabled status.
I2C*i*.I2C_IRQSTATUS[14] XDR	I2C*i*.I2C_IRQENABLE_SET [14] XDR_IE	I2C*i*.I2C_IRQENABLE_CLR[14] XDR_IE	I2C*i*_IRQ	Transmit draining IRQ enabled status.

18.1.4.6 I²C DMA Requests

Each multimaster I²C controller can generate two DMA requests to the device DMA controllers through the DMA_CROSSBAR module. Table 18-11 lists the DMA requests. For information about DMA generation, see Section 18.1.4.8.3, *I²C FIFO DMA Mode (I²C Mode Only)*.

Table 18-11. I²C DMA Requests

Name	Source	Description
I2C1_DREQ_TX	I2C1	I2C1 DMA write request to inform the DMAs to write new data in the I2C1.I2C_DATA[7:0] DATA bit field
I2C1_DREQ_RX	I2C1	I2C1 DMA read request to inform the DMAs to read the data in the I2C1.I2C_DATA[7:0] DATA bit field
I2C2_DREQ_TX	I2C2	I2C2 DMA write request to inform the DMAs to write new data in the I2C2.I2C_DATA[7:0] DATA bit field
I2C2_DREQ_RX	I2C2	I2C2 DMA read request to inform the DMAs to read the data in the I2C2.I2C_DATA[7:0] DATA bit field

NOTE: For more information about I2C*i*_DREQ_TX and I2C*i*_DREQ_RX(where *i* = 1 to 2) signals mapping to DMA_CROSSBAR, see Section 11.3.2, *Mapping of DMA Requests to DMA_CROSSBAR Inputs*, in Table 18-4, *I²C Hardware Requests*.

18.1.4.7 I²C Programmable Multislave Channel Feature

This feature allows each multimaster I²C controller to be addressed using four separate Own Addresses configured in the I2C*i*.I2C_OA and I2C*i*.I2C_OAx registers (where = 1 to 2 , x = 1, 2, 3). An additional register (I2C*i*.I2C_ACTOA) is used to indicate to the LH which address is used by the external master to communicate with the I²C controller.

Each Own Address can be independently configured in 7-bit or 10-bit mode by setting the corresponding bit (I2C*i*.I2C_CON[7] XOA0, I2C*i*.I2C_CON[6] XOA1, I2C*i*.I2C_CON[5] XOA2, or I2C*i*.I2C_CON[4] XOA3).

18.1.4.8 I²C FIFO Management

Each multimaster I²C controller implements two internal 8-bit FIFOs, the RX and TX FIFOs.

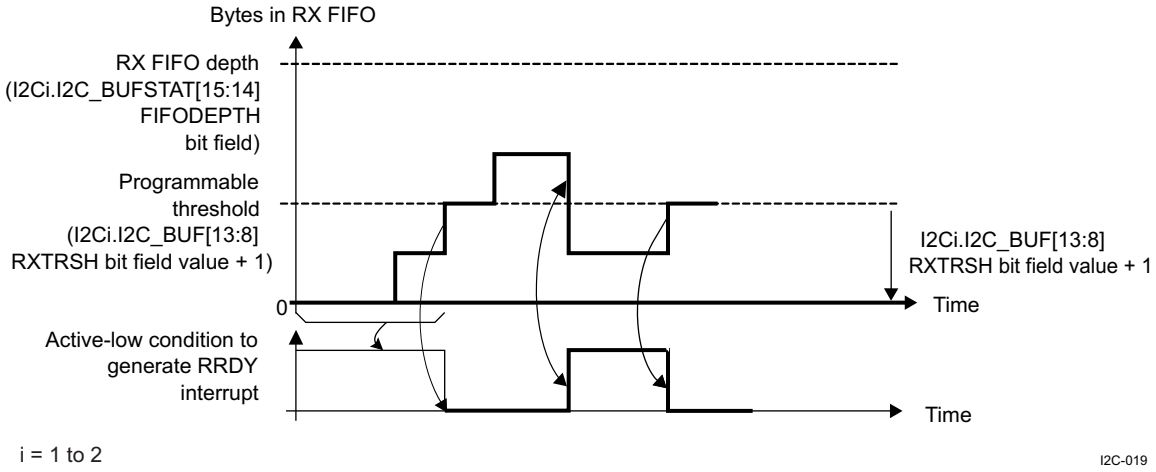
The depth of the RX and TX FIFOs can be checked by reading the I2C*i*.I2C_BUFSTAT[15:14] FIFODEPTH bit field (0x0: 8 bytes, 0x1: 16 bytes, 0x2: 32 bytes, and 0x3: 64 bytes).

18.1.4.8.1 I²C FIFO Interrupt Mode

In FIFO interrupt mode (relevant interrupts enabled by the I2C*i*.I2C_IRQENABLE_SET register), an interrupt signal informs the processor of the receiver and transmitter status. These interrupts are raised when the RX/TX FIFO thresholds (defined by the I2C*i*.I2C_BUF[13:8] RXTRSH bit field value + 1 for the RX FIFO or the I2C*i*.I2C_BUF[5:0] TXTRSH bit field value + 1 for the TX FIFO) are reached; the interrupt signals instruct the LH to transfer data to the destination (from the I²C controller in receive mode and/or from any source to the I²C controller FIFO in transmit mode).

Figure 18-12 and Figure 18-13 show receive and transmit operations, respectively, from a FIFO management point of view.

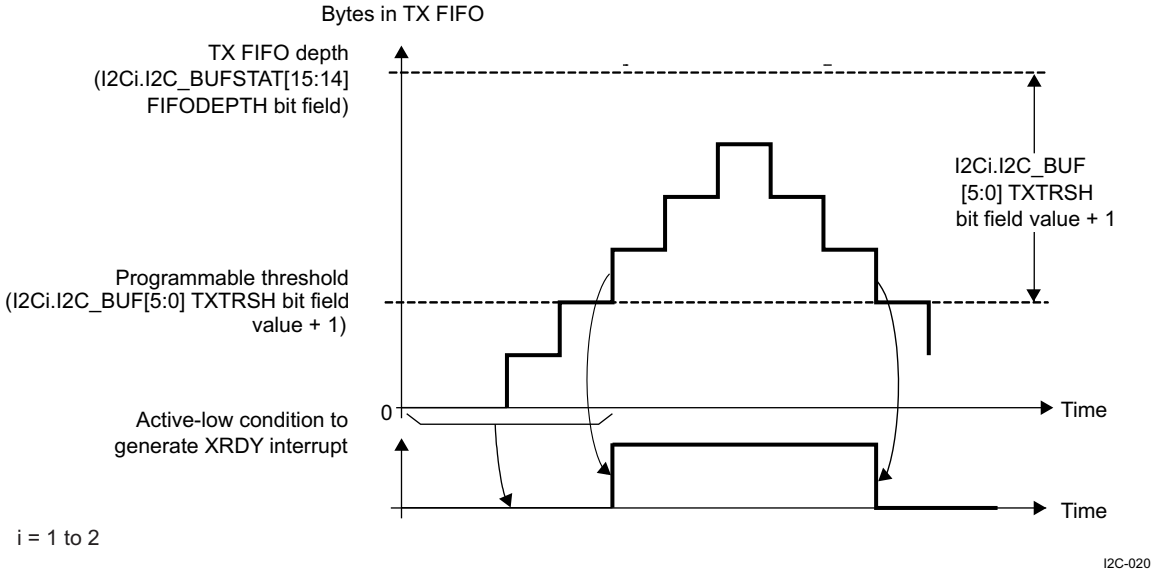
Figure 18-12. I²C Receive FIFO Interrupt Request Generation



In Figure 18-12, the RRDY interrupt condition shows that the condition for generating an RRDY interrupt is achieved. The interrupt request is generated when this signal is active, and it can be cleared only by the LH by writing 1 in the corresponding bit. If the condition is still present after clearing the previous interrupt, another interrupt request is generated.

In receive mode, an RRDY interrupt is generated as soon as the FIFO reaches its receive threshold (I2C*i*.I2C_BUF[13:8] RXTRSH bit field value + 1). The interrupt can be deasserted only when the LH has handled enough bytes to make the number of bytes in the RX FIFO lower than the programmed threshold. For each interrupt, the LH can be configured to read a number of bytes equal to the value of the RX FIFO threshold.

Figure 18-13. I²C Transmit FIFO Interrupt Request Generation



In Figure 18-13, the XRDY interrupt condition shows that the condition for generating an XRDY interrupt is achieved. The interrupt request is generated when TX FIFO is empty or when the TX FIFO threshold is not reached, and the LH can clear the XRDY status bit by setting the I2C*i*.I2C_IRQENABLE_CLR [4] XRDY_IE bit to 1 after transmitting the configured number of bytes. If the condition is still present after clearing the previous interrupt, another interrupt request is generated.

In interrupt mode, the module offers two options for the LH application to handle the interrupts:

- When detecting an interrupt request (XRDY or RRDY type), the LH can write/read 1 data byte to/from the TX/RX FIFO and then clear the interrupt. The module reasserts the interrupt until the interrupt condition is not met.
- When detecting an interrupt request (XRDY or RRDY type), the LH can be programmed to write/read the amount of data bytes specified by the corresponding FIFO threshold (I2C_BUF[5:0] TXTRSH + 1 or I2C_BUF[5:0] RXTRSH + 1). In this case, the interrupt condition is cleared and the next interrupt is asserted again when the XRDY or RRDY condition is met again.

If the second-interrupt-serving approach is used, an additional mechanism (draining feature) is implemented for cases where the transfer length is not a multiple of the FIFO threshold value (see Section 18.1.4.8.4, *Draining Feature [I²C Mode Only]*).

NOTE: In slave transmit mode (the I2C*i*.I2C_CON[10] MST bit is cleared and the I2C*i*.I2C_CON[9] TRX bit is set to 1), the draining feature must not be used, because the transfer length is not known at configuration time, and the external master can end the transfer at any point by not acknowledging 1 data byte. If the draining feature is used in slave transmit mode, data can remain in the TX FIFO without being transmitted over the I²C bus. In this case, the TX FIFO must be cleared by setting the I2C*i*.I2C_BUF[6] TXFIFO_CLR bit.

18.1.4.8.2 I²C FIFO Polling Mode

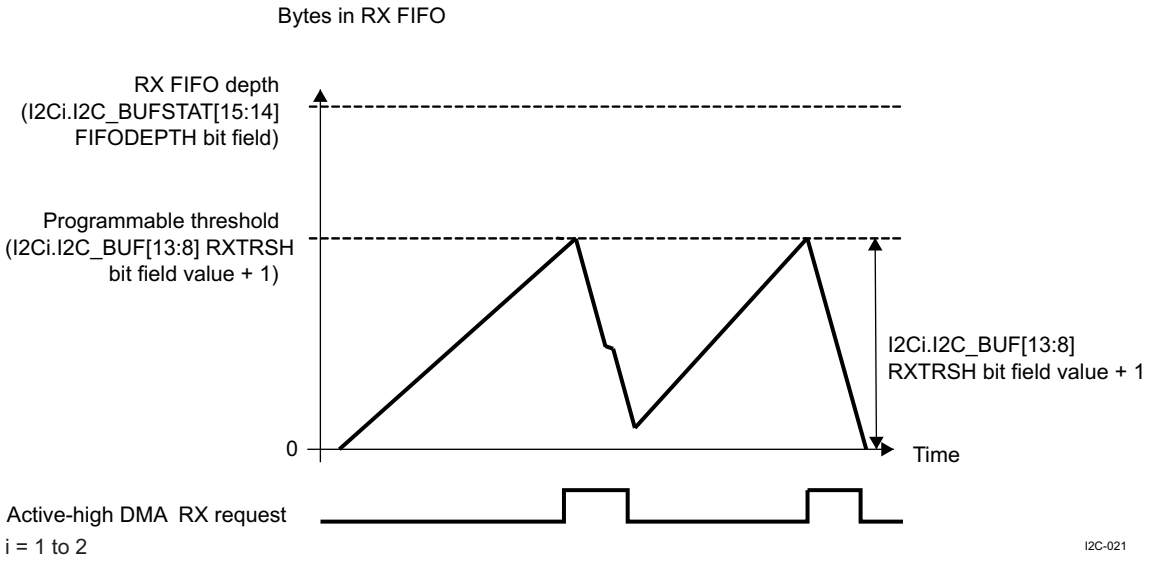
In FIFO polling mode (the I2C*i*.I2C_IRQENABLE_SET [4] XRDY_IE and I2C*i*.I2C_IRQENABLE_SET [3] RRDY_IE bits are disabled), the status of the module (receiver or transmitter) can be checked by polling the I2C*i*.I2C_IRQSTATUS_RAW [4] XRDY and the I2C*i*.I2C_IRQSTATUS_RAW [3] RRDY bits (the I2C*i*.I2C_IRQSTATUS_RAW [13] RDR and I2C*i*.I2C_IRQSTATUS_RAW [14] XDR bits can also be polled if the draining feature is enabled). The I2C*i*.I2C_IRQSTATUS_RAW [4] XRDY and I2C*i*.I2C_IRQSTATUS_RAW [3] RRDY bits accurately reflect the interrupt conditions described in the discussion of FIFO interrupt mode.

18.1.4.8.3 I²C FIFO DMA Mode

In receive mode, a DMA request is generated by the I2C*i*_DREQ_RX signal as soon as the RX FIFO exceeds its threshold level (the I2C*i*.I2C_BUF[13:8] RXTRSH bit field value + 1). This request is deasserted when the number of bytes defined by the threshold level is read by the DMA controller.

Figure 18-14 shows the DMA request generation in receive mode.

Figure 18-14. I²C Receive FIFO DMA Request Generation



In transmit mode, a DMA request is automatically asserted by the I2C*i*_DREQ_TX signal when the TX FIFO is empty. This request is deasserted when the number of bytes (the I2C*i*.I2C_BUF[5:0] TXTRSH bit field value + 1) is written in the FIFO by the DMA controller. If an insufficient number of bytes is written, the DMA request remains active. and show the DMA TX transfers with different values for the I2C*i*.I2C_BUF[5:0] TXTRSH bit field.

Figure 18-15. I²C Transmit FIFO Request Generation (High Threshold)

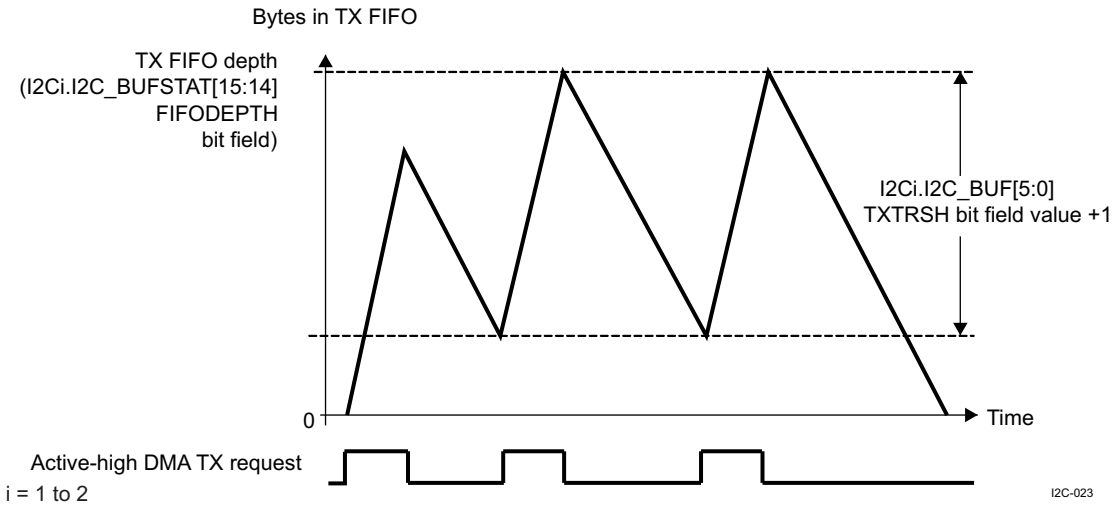
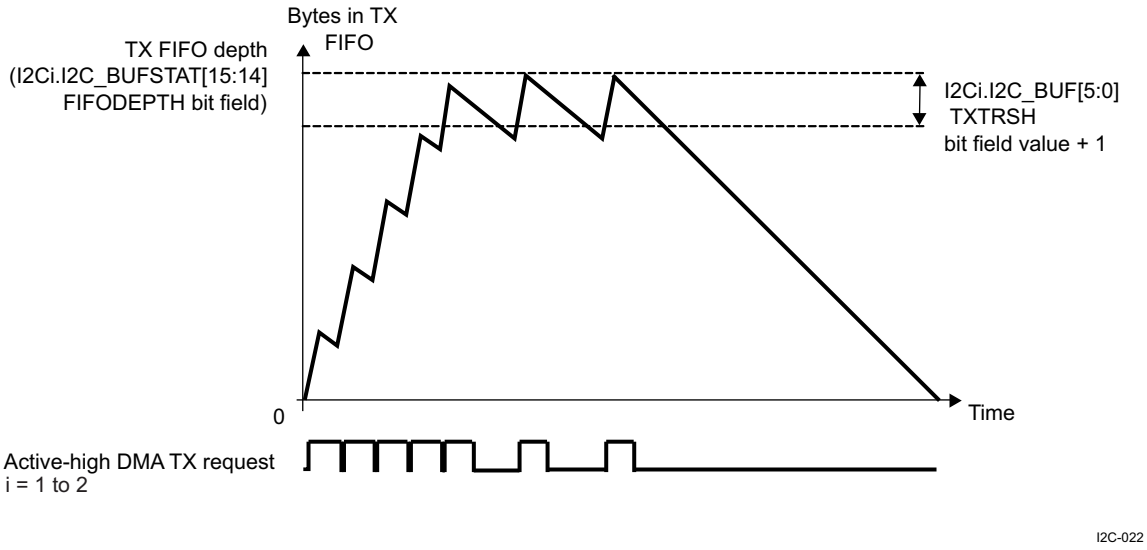


Figure 18-16. I²C Transmit FIFO Request Generation (Low Threshold)



The I2C module provides the possibility to the user to clear the RX or TX FIFO, by setting the I2Ci.I2C_BUF[14]RXFIFO_CLR and I2Ci.I2C_BUF[6]TXFIFO_CLR registers, which act like software reset for the FIFOs. In DMA mode, these bits will also reset the DMA state machines.

The FIFO clearing feature can be used when the following conditions are met:

1. The module is configured as a transmitter
2. The external receiver responds with a NACK in the middle of the transfer
3. There is still data in TX FIFO waiting to be transferred

18.1.4.8.4 I²C Draining Feature

The draining feature is implemented to handle the end of a transfer whose length is not a multiple of the FIFO threshold values (the I2C*i*.I2C_BUF[13:8] RXTRSH bit field value + 1 for the RX threshold and the I2C*i*.I2C_BUF[5:0] TXTRSH field value + 1 for the TX threshold). It can also transfer the remaining number of bytes (because the threshold is not reached).

This feature prevents the LH or the DMA controller from trying more FIFO accesses than necessary (for example, to generate at the end of a transfer a DMA RX request having fewer bytes in the FIFO than the configured DMA transfer length). Otherwise, an AERR interrupt is generated by the I2C*i*.I2C_IRQSTATUS_RAW [7] AERR bit.

The draining mechanism generates an interrupt using the I2C*i*.I2C_IRQSTATUS_RAW [13] RDR or I2C*i*.I2C_IRQSTATUS_RAW [14] XDR bit at the end of the transfer, informing the LH that it must check the amount of data left to be transferred (the I2C*i*.I2C_BUFSTAT[13:8] RXSTAT or I2C*i*.I2C_BUFSTAT[5:0] TXSTAT bit fields) and enable the draining feature of the DMA controller by reconfiguring the DMA transfer length according to this value (when the DMA mode is enabled) or perform only the required number of data accesses (when the DMA mode is disabled).

In receive mode (master or slave), if the RX FIFO threshold (the I2C*i*.I2C_BUF[13:8] RXTRSH bit field value + 1) is not reached, but the transfer ends on the I²C bus and data remains in the RX FIFO (less than the threshold), the receive draining interrupt (the I2C*i*.I2C_IRQSTATUS_RAW [13] RDR bit) is asserted to inform the LH that it can read the amount of data in the RX FIFO (the I2C*i*.I2C_BUFSTAT[13:8] RXSTAT bit field). The LH performs a number of data read accesses equal to the I2C*i*.I2C_BUFSTAT[13:8] RXSTAT bit field (interrupt or polling mode), or reconfigures the DMA controller with the required value to drain the FIFO.

In master transmit mode, if the TX FIFO threshold (the I2C*i*.I2C_BUF[5:0] TXTRSH bit field value + 1) is not reached, but the amount of data remaining to be written in the TX FIFO is less than the threshold, the transmit draining interrupt (the I2C*i*.I2C_IRQSTATUS_RAW [14] XDR bit) is asserted to inform the LH that it can read the amount of data remaining to be written in the TX FIFO (the I2C*i*.I2C_BUFSTAT[5:0] TXSTAT bit field). The LH must write the required number of data bytes specified by the I2C*i*.I2C_BUFSTAT[5:0] TXSTAT bit field value or reconfigure the DMA controller with the value required to transfer the last bytes to the FIFO.

In master mode, the LH can alternately not check the values of the I2C*i*.I2C_BUFSTAT[5:0] TXSTAT and I2C*i*.I2C_BUFSTAT[13:8] RXSTAT bit fields, because it can obtain this information internally (by computing the I2C*i*.I2C_CNT[15:0] DATACOUNT bit field value modulo I2C*i*.I2C_BUF[13:8] RXTRSH or I2C*i*.I2C_BUF[5:0] TXTRSH).

By default, the draining feature is disabled; it can be enabled using the I2C*i*.I2C_IRQENABLE_SET [14] XDR_IE or I2C*i*.I2C_IRQENABLE_SET [13] RDR_IE bits (default disabled) only for transfers with lengths not equal to the threshold values (I2C*i*.I2C_BUF[5:0] TXTRSH bit field value + 1 for the TX threshold or the I2C*i*.I2C_BUF[13:8] RXTRSH bit field value + 1 for the RX threshold).

18.1.4.9 I²C Noise Filter

The noise filter is used to suppress any noise that is 50 ns or less in case of F/S operation modes. The noise filter is always one period of the I2C*i*_INTERNAL_CLK clock.

For standard mode (for example, the I2C*i*.I2C_PSC[7:0] PSC bit field = 4), the maximum width of suppressed spikes is 46.1 ns.

To ensure correct filtering, the prescaler must be programmed accordingly by the 2C*i*.I2C_PSC[7:0] PSC bit field.

18.1.4.10 I²C System Test Mode

A system test mode is available for multimaster I²C controller module testing. This mode is enabled by setting the I2C*i*.I2C_SYSTEST[15] ST_EN bit to 1. When this bit is cleared to 0, the I²C controller is configured in normal operation mode.

In system test mode, the I2C*i*.I2C_SYSTEST [13:12] TMODE bit field selects the type of test. Table 18-12 lists the tests available for the multimaster I²C controllers.

Table 18-12. I²C List of Tests

I2C*i*.I2C_SYSTEST[13:12] TMODE	Test	Description
00	Functional mode	Normal operation mode
01	Reserved (not used)	
10	Test of i2c*i*_scl serial clock line	The i2c*i*_scl line is driven with a permanent clock as if mastered with the parameters set in the I2C*i*.I2C_PSC, I2C*i*.I2C_SCLL, and I2C*i*.I2C_SCLH registers.
11	Loop-back mode + i2c*i*_scl/ i2c*i*_sda I/O	In master transmit mode only, data transmitted out of the I2C*i*.I2C_DATA register (write action) is received in the same I2C*i*.I2C_DATA register through an internal path through the FIFO buffers. The DMA and interrupt requests are normally generated if they are enabled. Moreover, the i2c*i*_scl and i2c*i*_sda are controlled with the I2C*i*.I2C_SYSTEST[3:0] bits.

NOTE: When the I2C*i*.I2C_SYSTEST[13:12] TMODE bit field is set to 11, the I²C controller must be configured in I²C F/S (I2C*i*.I2C_CON[13:12] OPMODE set to 00).

NOTE: In normal operation mode (the I2C*i*.I2C_SYSTEST[15] ST_EN bit cleared to 0), the I2C*i*.I2C_SYSTEST[3:0] bits that control the i2c*i*_scl, i2c*i*_sda lines in system test mode are read-only bits.

In system test mode (the I2C*i*.I2C_SYSTEST[15] ST_EN bit set to 1), the I2C*i*.I2C_IRQSTATUS_RAW.XRDY, I2C_IRQSTATUS_RAW.RRDY, I2C_IRQSTATUS_RAW.XUDF, I2C_IRQSTATUS_RAW.ROVR, I2C_IRQSTATUS_RAW.ARDY and I2C_IRQSTATUS_RAW.NACK status bits can be set to 1 when the I2C*i*.I2C_SYSTEST[11] SSB bit is set to 1. Clearing the I2C*i*.I2C_SYSTEST[11] SSB bit to 0 does not clear the I2C*i*.I2C_IRQSTATUS_RAW bits to 0. The I2C*i*.I2C_IRQSTATUS_RAW bit field can be cleared to 0 only by writing 1 in the corresponding bits.

18.1.5 I²C Programming Guide

18.1.5.1 I²C Low-Level Programming Models

18.1.5.1.1 I²C Programming Model

This section describes the programming model of the multimaster I²C controllers configured in I²C mode.

18.1.5.1.1.1 Main Program

18.1.5.1.1.1.1 Configure the Module Before Enabling the I²C Controller

Before enabling the I²C controller, perform the following steps:

1. Enable the functional and interface clocks (see Table 18-3).
2. Program the prescaler to obtain an approximately 12-MHz internal sampling clock by programming the corresponding value in the I2C*i*.I2C_PSC[7:0] PSC bit field. This value depends on the frequency of the functional clock (I2C*i*_FCLK).
3. Program the I2C*i*.I2C_SCLL[7:0] SCLL and I2C*i*.I2C_SCLH[7:0] SCLH bit fields to obtain a bit rate of 100 kbps or 400 kbps. These values depend on the internal sampling clock frequency (see Table 18-6).
4. Configure the Own Address of the I²C controller by storing it in the I2C*i*.I2C_OA register. Up to four Own Addresses can be programmed in the I2C*i*.I2C_OA and I2C*i*.I2C_OAx registers (where x = 1, 2, 3) for each I²C controller.

NOTE: For a 10-bit address, set the corresponding expand Own Address bit in the I2C*i*.I2C_CON register.

5. Set the TX threshold (in transmitter mode) and the RX threshold (in receiver mode) by setting the I2C*i*.I2C_BUF[5:0] TXTRSH bit field to (TX threshold – 1) and the I2C*i*.I2C_BUF[13:8] RXTRSH bit field to (RX threshold – 1), where the TX and RX thresholds are greater than or equal to 1.
6. Take the I²C controller out of reset by setting the I2C*i*.I2C_CON[15] I2C_EN bit to 1.

18.1.5.1.1.1.2 Initialize the I²C Controller

To initialize the I²C controller, perform the following steps:

1. Configure the I2C*i*.I2C_CON register:
   - For master or slave mode, set the I2C*i*.I2C_CON[10] MST bit (0: slave; 1: master).
   - For transmitter or receiver mode, set the I2C*i*.I2C_CON[9] TRX bit (0: receiver; 1: transmitter).
2. If using an interrupt to transmit and receive data, set the corresponding bit in the I2C*i*.I2C_IRQENABLE_SET register to 1 (the I2C*i*.I2C_IRQENABLE_SET [4] XRDY_IE bit for the transmit interrupt, the I2C*i*.I2C_IRQENABLE_SET [3] RRDY bit for the receive interrupt).
3. If using DMA to receive and transmit data, set the corresponding bit in the I2C*i*.I2C_BUF register to 1 (the I2C*i*.I2C_BUF[15] RDMA_EN bit for the receive DMA channel, the I2C*i*.I2C_BUF[7] XDMA_EN bit for the transmit DMA channel).

18.1.5.1.1.1.3 Configure Slave Address and the Data Control Register

In master mode, configure the slave address register by programming the I2C*i*.I2C_SA[9:0] SA bit field and the number of data bytes (I^2C data payload) associated with the transfer by programming the I2C*i*.I2C_CNT[15:0] DCOUNT bit field.

NOTE: For a 10-bit address, set the I2C*i*.I2C_CON[8] XSA bit to 1.

18.1.5.1.1.1.4 Initiate a Transfer

Poll the I2C*i*.I2C_IRQSTATUS_RAW [12] BB bit. If it is cleared to 0 (bus not busy), configure the I2C*i*.I2C_CON[0] STT and I2C*i*.I2C_CON[1] STP bits. To initiate a transfer, the I2C*i*.I2C_CON[0] STT bit must be set to 1, and it is not mandatory to set the I2C*i*.I2C_CON[1] STP bit to 1.

18.1.5.1.1.1.5 Receive Data

Poll the I2C*i*.I2C_IRQSTATUS_RAW [3] RRDY bit, or use the RRDY interrupt (the I2C*i*.I2C_IRQENABLE_SET [3] RRDY_IE bit must be set to 1) or the DMA RX channel (the I2C*i*.I2C_BUF[15] RDMA_EN bit must be set to 1 together with I2C_DMARXENABLE_SET) to read the receive data in the I2C*i*.I2C_DATA register.

If the transfer length does not equal the RX FIFO threshold (the I2C*i*.I2C_BUF[13:8] RTRSH bit field + 1), use the draining feature (enable the RDR interrupt by setting the I2C*i*.I2C_IRQENABLE_SET [13] RDR_IE bit to 1).

NOTE: In receive mode only, the I2C*i*.I2C_IRQSTATUS_RAW [11] ROVR (receive overrun) bit indicates whether the receiver has experienced overrun. An overrun condition occurs when the shift register and the RX FIFO are full. An overrun condition does not result in data loss; the I^2C controller simply holds i2c*i*_scl to low to prevent other bytes from being received.

The I2C*i*.I2C_IRQSTATUS_RAW[7] AERR bit is set to 1 when a read access is performed in the I2C*i*.I2C_DATA register while the RX FIFO is empty. The corresponding interrupt can be enabled by setting the I2C*i*.I2C_IRQENABLE_SET [7] AERR_IE bit to 1.

18.1.5.1.1.1.6 Transmit Data

Poll the I2C*i*.I2C_IRQSTATUS_RAW [4] XRDY bit, or use the XRDY interrupt (the I2C*i*.I2C_IRQENABLE_SET [4] XRDY_IE bit must be set to 1) or the DMA TX channel (the I2C*i*.I2C_BUF[7] XDMA_EN bit must be set to 1 together with I2C_DMATXENABLE_SET) to write data to the I2C*i*.I2C_DATA register.

If the transfer length does not equal the TX FIFO threshold (the I2C*i*.I2C_BUF[5:0] TXTRSH bit field + 1), use the draining feature (enable the XDR interrupt by setting the I2C*i*.I2C_IRQENABLE_SET [14] XDR_IE bit to 1).

NOTE: In transmit mode only, the I2C*i*.I2C_IRQSTATUS_RAW [10] XUDF bit indicates whether the transmitter has experienced underflow.

In master transmit mode, underflow occurs when the shift register and the TX FIFO are empty and there are still some bytes to transmit (the value of the I2C*i*.I2C_CNT[15:0] DCOUNT bit field is not 0).

In slave transmit mode, underflow occurs when the shift register and the TX FIFO are empty and the external I^2C master device still requests data bytes to be read.

The I2C*i*.I2C_IRQSTATUS_RAW [7] AERR bit is set to 1 when a write access is performed in the I2C*i*.I2C_DATA register while the TX FIFO is full. The corresponding interrupt can be enabled by setting the I2C*i*.I2C_IRQENABLE_SET [7] AERR_IE bit to 1.

18.1.5.1.1.2 Interrupt Subroutine Sequence

1. Test for arbitration lost (the I2C*i*.I2C_IRQSTATUS_RAW [0] AL bit) and resolve accordingly.
2. Test for no acknowledgment (the I2C*i*.I2C_IRQSTATUS_RAW [1] NACK bit) and resolve accordingly.
3. Test for register access ready (the I2C*i*.I2C_IRQSTATUS_RAW [2] ARDY bit) and resolve accordingly.
4. Test for receive data ready (the I2C*i*.I2C_IRQSTATUS_RAW [3] RRDY bit) and resolve accordingly.
5. Test for transmit data ready (the I2C*i*.I2C_IRQSTATUS_RAW [4] XRDY bit) and resolve accordingly.
6. Test for general call (the I2C*i*.I2C_IRQSTATUS_RAW [5] GC bit) and resolve accordingly.
7. Test for start (S) condition (the I2C*i*.I2C_IRQSTATUS_RAW [6] STC bit) and resolve accordingly. For this test, the functional clock must be inactive.
8. Test for access error (the I2C*i*.I2C_IRQSTATUS_RAW [7] AERR bit) and resolve accordingly.
9. Test for bus free (the I2C*i*.I2C_IRQSTATUS_RAW [8] BF bit) and resolve accordingly.

18.1.5.1.1.3 Programming Flow Diagrams

Figure 18-17 through Figure 18-25 are procedure flow charts for programming the F/S I²C modes.

Figure 18-17. I²C Setup Procedure

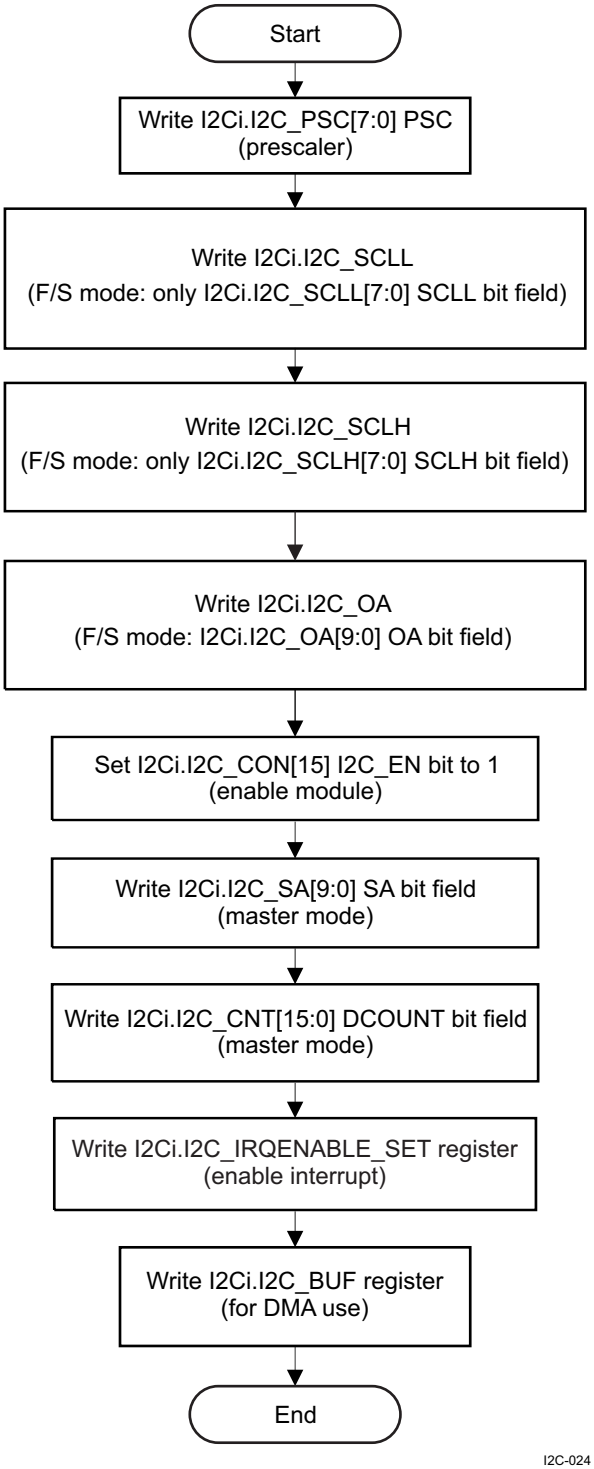


Table 18-13. Subprocess Call Summary for Sequence – I²C Setup Procedure

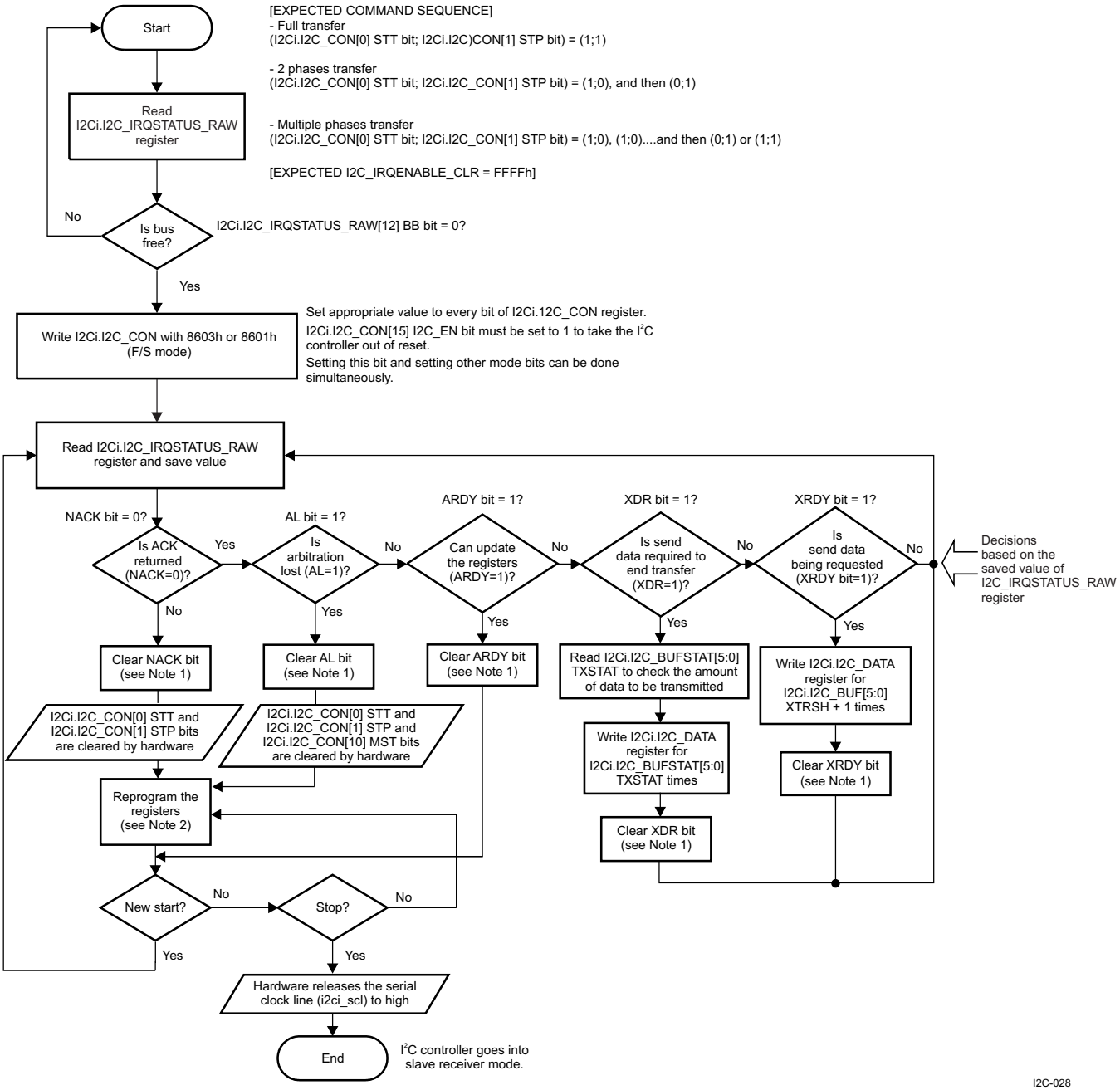
Subprocess Name	Cross-Reference
Pad configuration	See Section 13.4.6.1, *PAD Configuration Registers* in *Chapter 13Control Module*

Table 18-14. I²C Register Call Summary for Sequence – Setup Procedure

Register Name	Register Name	Register Name
I2C*i*.I2C_PSC (1)	I2C*i*.I2C_CON (1)	I2C*i*.I2C_BUF (1)
I2C*i*.I2C_SCLL (1)	I2C*i*.I2C_SA (1)	I2C*i*.I2C_IRQENABLE_SET (1)
I2C*i*.I2C_SCLH (1)	I2C*i*.I2C_CNT (1)	I2C*i*.I2C_OA (1)

(1) *i* = 1 to 2

Figure 18-18. I²C Master Transmitter Mode, Polling Method, in F/S Modes



(1) The NACK, AL, ARDY, XDR, and XRDY bits are cleared by writing 1 to each corresponding bit in the I2C*i*.I2C_IRQSTATUS register.

(2) Reprogram the registers means: I2C*i*.I2C_CON[11] STB and/or I2C*i*.I2C_CON[10] MST bit and/or I2C*i*.I2C_SA[9:0] SA register and/or I2C*i*.I2C_CNT[15:0] DCOUNT register and/or I2C*i*.I2C_CON[0] STT bit and/or I2C*i*.I2C_CON[1] STP bit.

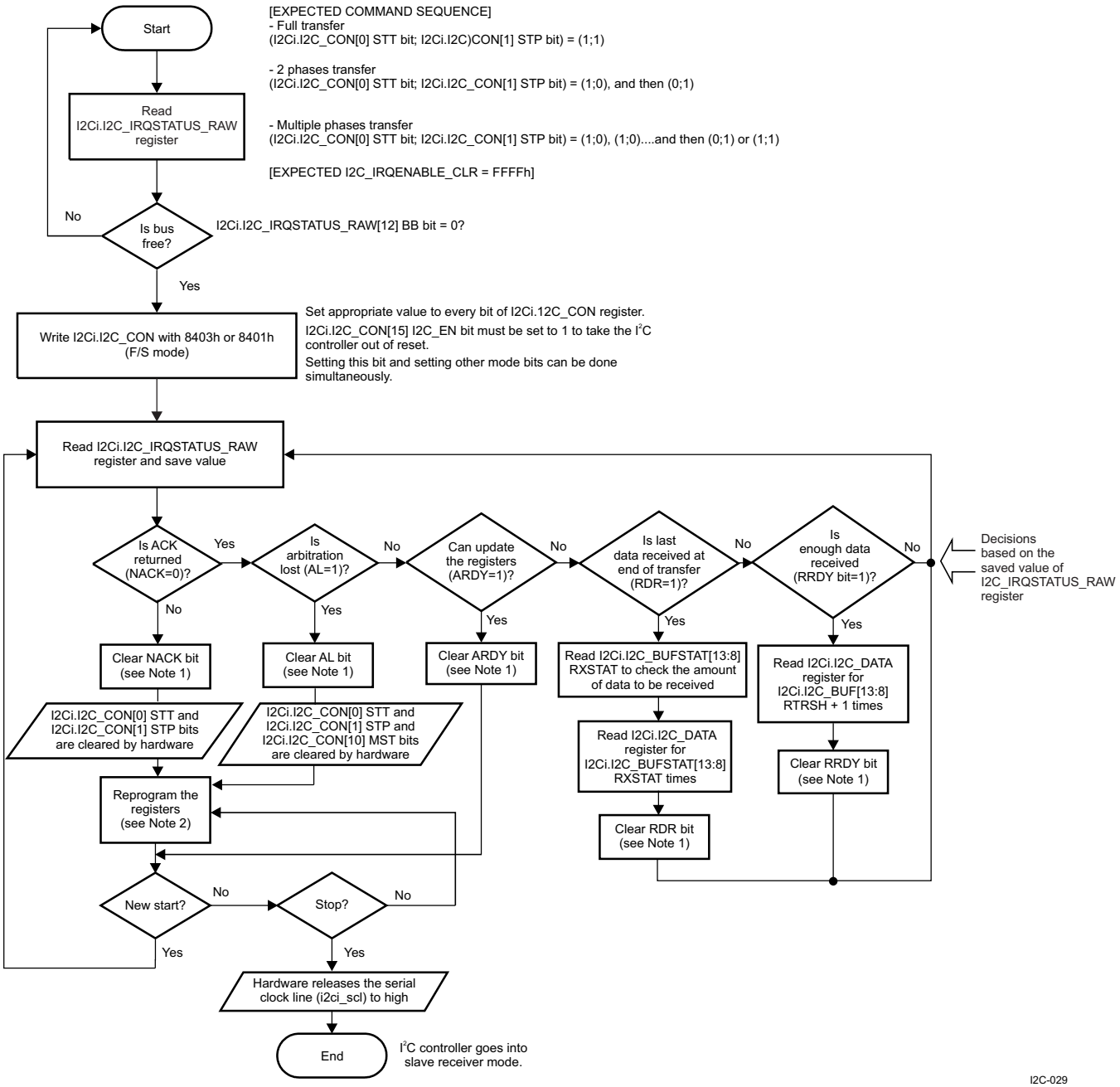
NOTE: The FIFO clearing can be made when the module is configured as transmitter, the receiver send a NACK in the middle of the transfer, and there is still data in the FIFO.

Table 18-15. I²C Register Call Summary for Sequence – Master Transmitter Mode, Polling Method, in F/S Modes

Register Name	Register Name	Register Name
I2C*i*.I2C_IRQSTATUS_RAW [(1)]	I2C*i*.I2C_DATA [(1)]	I2C*i*.I2C_CON [(1)]
I2C*i*.I2C_BUFSTAT [(1)]	I2C*i*.I2C_CON [(1)]	I2C*i*.I2C_BUF [(1)]

(1) *i* = 1 to 2

Figure 18-19. I²C Master Receiver Mode, Polling Method, in F/S Modes



(1) The NACK, AL, ARDY, RDR, and RRDY bits are cleared by writing 1 to each corresponding bit in the I2C*i*.I2C_IRQSTATUS register.

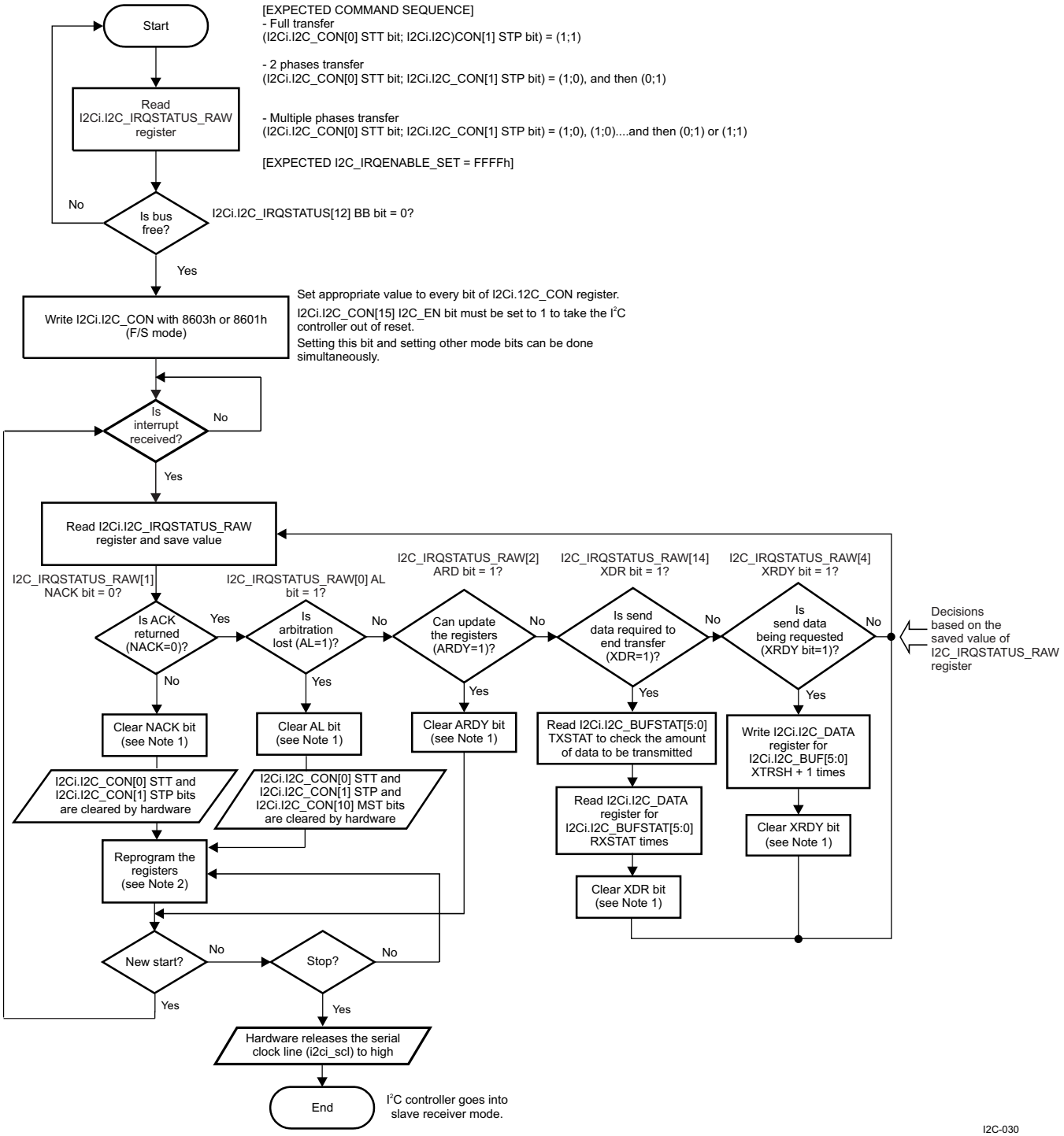
(2) Reprogram registers means: I2C*i*.I2C_CON[11] STB and/or I2C*i*.I2C_CON[10] MST bit and/or I2C*i*.I2C_SA[9:0] SA register and/or I2C*i*.I2C_CNT[15:0] DCOUNT register and/or I2C*i*.I2C_CON[0] STT bit and/or I2C*i*.I2C_CON[1] STP bit.

Table 18-16. I²C Register Call Summary for Sequence – Master Receiver Mode, Polling Method, in F/S Modes

Register Name	Register Name	Register Name
I2C*i*.I2C_IRQSTATUS_RAW (1)	I2C*i*.I2C_BUFSTAT (1)	I2C*i*.I2C_BUF (1)
I2C*i*.I2C_CON (1)	I2C*i*.I2C_DATA (1)	

(1) *i* = 1 to 2

Figure 18-20. I²C Master Transmitter Mode, Interrupt Method, in F/S Modes



(1) The NACK, AL, ARDY, XDR, and XRDY bits are cleared by writing 1 to each corresponding bit in the I2C*i*.I2C_IRQSTATUS register.

(2) Reprogram registers means: I2C*i*.I2C_CON[11] STB and/or I2C*i*.I2C_CON[10] MST bit and/or I2C*i*.I2C_SA[9:0] SA register and/or I2C*i*.I2C_CNT[15:0] DCOUNT register and/or I2C*i*.I2C_CON[0] STT bit and/or I2C*i*.I2C_CON[1] STP bit.

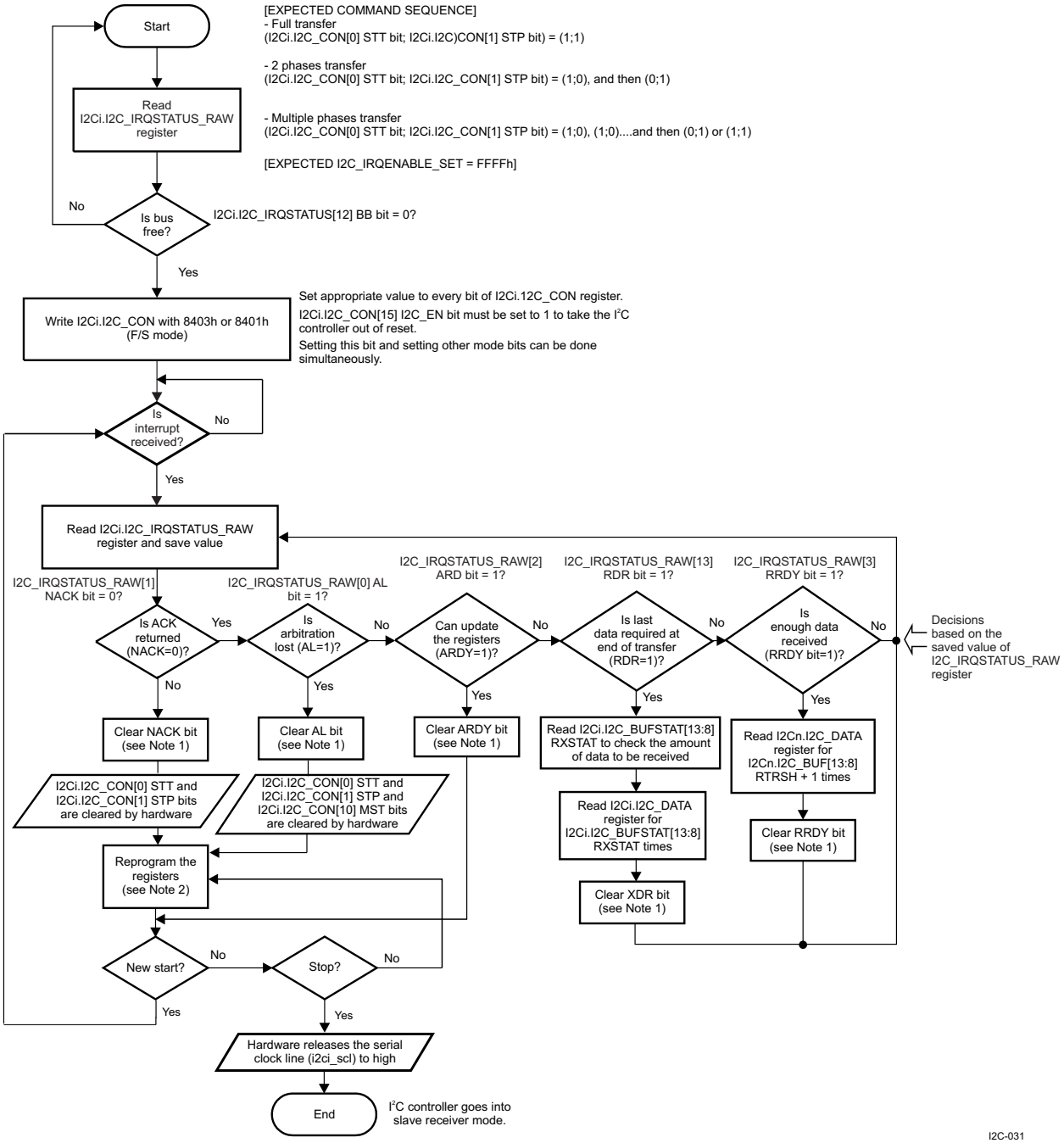
NOTE: The FIFO clearing can be made when the module is configured as transmitter, the receiver send a NACK in the middle of the transfer, and there is still data in the FIFO.

Table 18-17. I²C Register Call Summary for Sequence – Master Transmitter Mode, Interrupt Method, in F/S Modes

Register Name	Register Name	Register Name
I2C*i*.I2C_IRQSTATUS_RAW [(1)]	I2C*i*.I2C_BUFSTAT [(1)]	I2C*i*.I2C_BUF [(1)]
I2C*i*.I2C_CON [(1)]	I2C*i*.I2C_DATA [(1)]	

(1) *i* = 1 to 2

Figure 18-21. HS I²C Master Receiver Mode, Interrupt Method, in F/S Modes



(1) The NACK, AL, ARDY, RDR, and RRDY bits are cleared by writing 1 to each corresponding bit in the I2C*i*I2C_IRQSTATUS register.

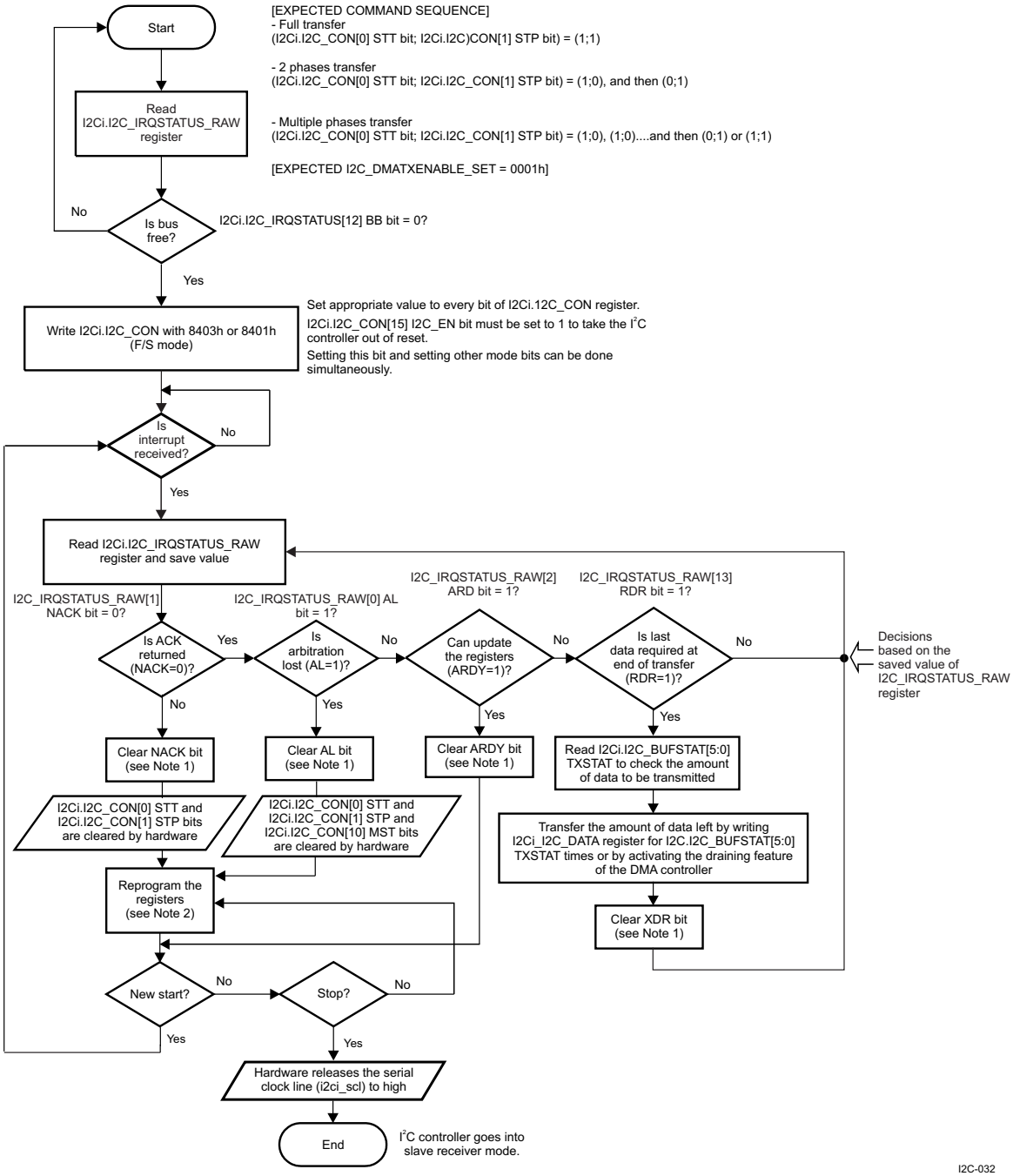
(2) Reprogram registers means: I2C*i*.I2C_CON[11] STB and/or I2C*i*.I2C_CON[10] MST bit and/or I2C*i*.I2C_SA[9:0] SA register and/or I2C*i*.I2C_CNT[15:0] DCOUNT register and/or I2C*i*.I2C_CON[0] STT bit and/or I2C*i*.I2C_CON[1] STP bit.

Table 18-18. I²C Register Call Summary for Sequence – Master Receiver Mode, Interrupt Method, in F/S Modes

Register Name	Register Name	Register Name
I2C*i*.I2C_IRQSTATUS_RAW [(1)]	I2C*i*.I2C_BUFSTAT [(1)]	I2C*i*.I2C_BUF [(1)]
I2C*i*.I2C_CON [(1)]	I2C*i*.I2C_DATA [(1)]	

(1) *i* = 1 to 2

Figure 18-22. I²C Master Transmitter Mode, DMA Method in F/S Modes



(1) The NACK, AL, ARDY, and XDR bits are cleared by writing 1 to each corresponding bit in the I2C*i*.I2C_IRQSTATUS register.

(2) Reprogram registers means: I2C*i*.I2C_CON[11] STB and/or I2Ci.I2C_CON[10] MST bit and/or I2C*i*.I2C_SA[9:0] SA register and/or I2C*i*.I2C_CNT[15:0] DCOUNT register and/or I2C*i*.I2C_CON[0] STT bit and/or I2C*i*.I2C_CON[1] STP bit.

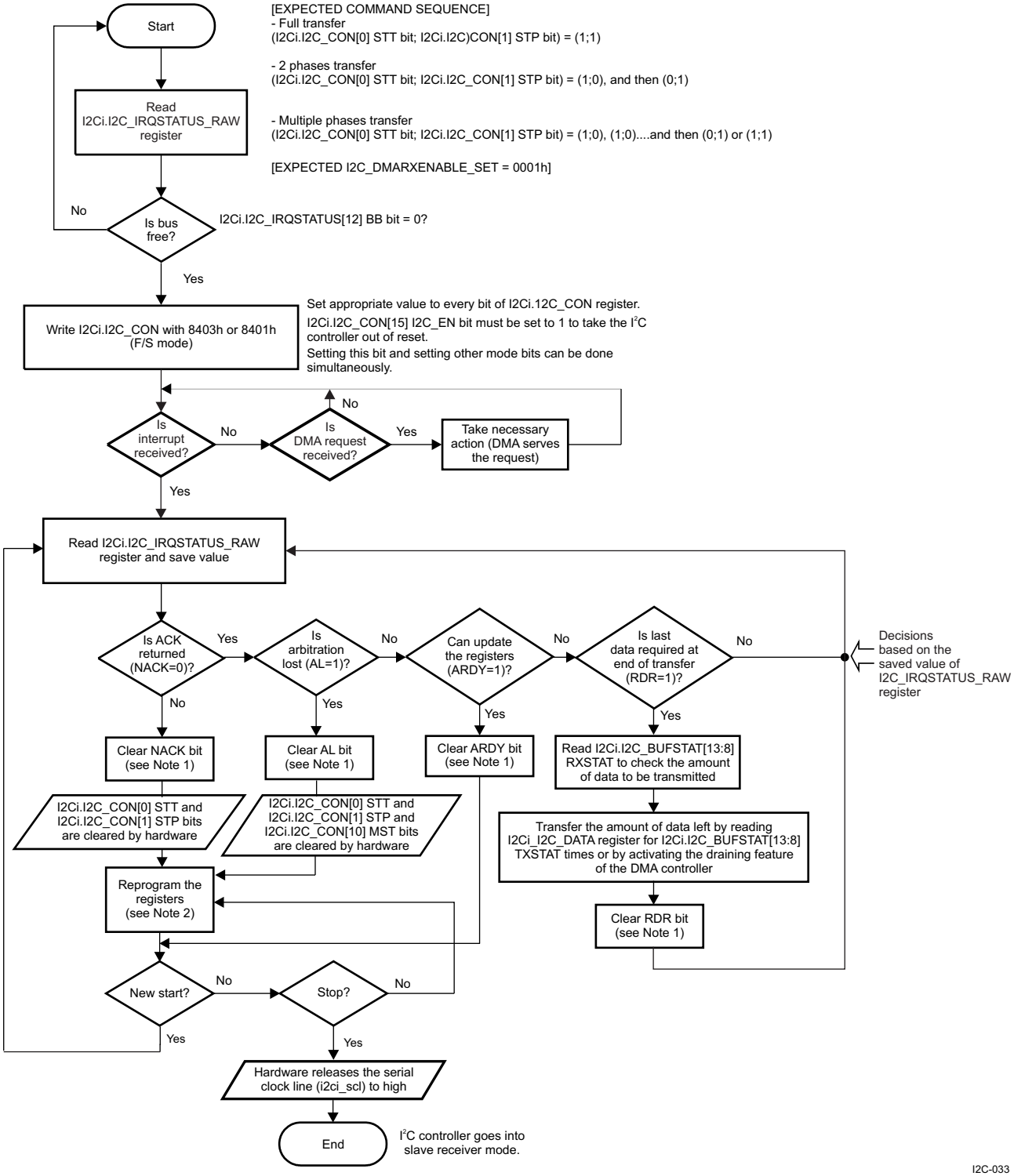
NOTE: The FIFO clearing can be made when the module is configured as transmitter, the receiver send a NACK in the middle of the transfer, and there is still data in the FIFO.

Table 18-19. I²C Register Call Summary for Sequence – Master Transmitter Mode, DMA Method in F/S Modes

Register Name	Register Name
I2C*i*.I2C_IRQSTATUS_RAW [1]	I2Ci.I2C_BUFSTAT [1]
I2C*i*.I2C_CON [1]	I2Ci.I2C_DATA [1]

[1] *i* = 1 to 2

Figure 18-23. I²C Master Receiver Mode, DMA Method in F/S Modes



(1) The NACK, AL, ARDY, and RDR bits are cleared by writing 1 to each corresponding bit in the I2C*i*.I2C_IRQSTATUS register.

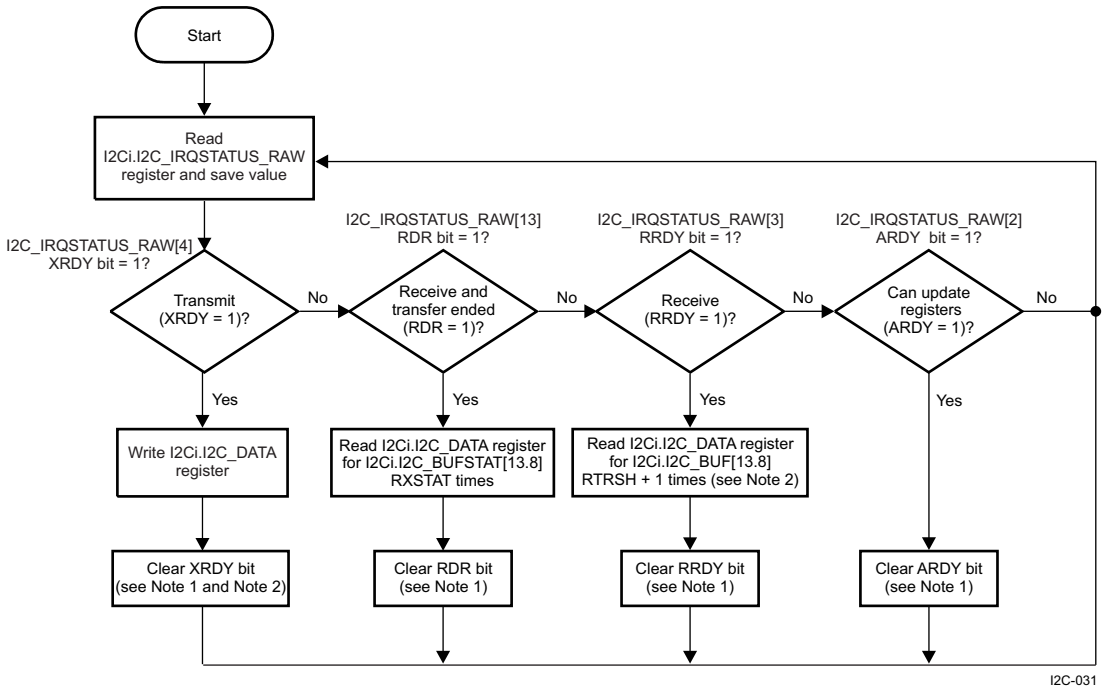
(2) Reprogram registers means: I2C*i*.I2C_CON[11] STB and/or I2C*i*.I2C_CON[10] MST bit and/or I2C*i*.I2C_SA[9:0] SA register and/or I2C*i*.I2C_CNT[15:0] DCOUNT register and/or I2C*i*.I2C_CON[0] STT bit and/or I2C*i*.I2C_CON[1] STP bit.

Table 18-20. I²C Register Call Summary for Sequence – Master Receiver Mode, DMA Method in F/S Modes

Register Name	Register Name
I2C*i*.I2C_IRQSTATUS_RAW (1)	I2C*i*.I2C_BUFSTAT (1)
I2Ci.I2C_CON (1)	I2C*i*.I2C_DATA (1)

(1) *i* = 1 to 2

Figure 18-24. I²C Slave Transmitter/Receiver Mode, Polling



(1) The XRDY, RDR, RRDY, and ARDY bits are cleared by writing 1 to each corresponding bit in the I2C*i*.I2C_IRQSTATUS register.

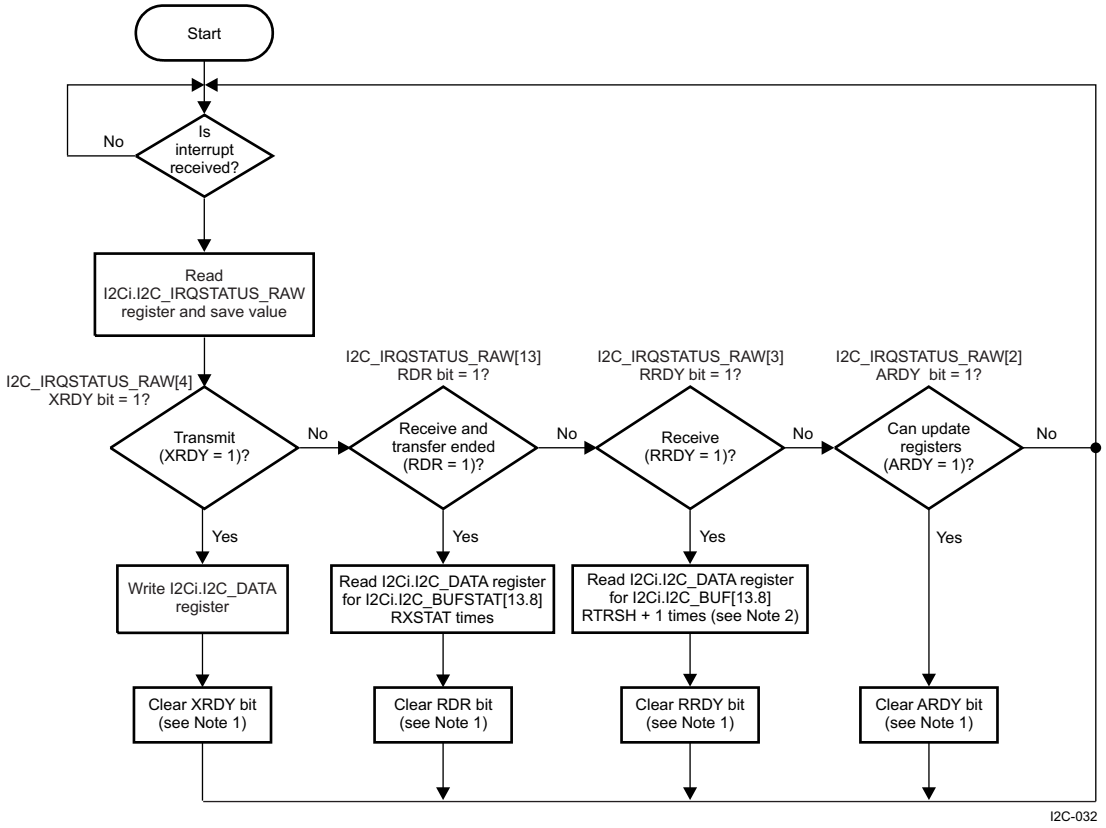
(2) In slave transmitter mode, the amount of data requested by the external master I²C device is unknown; thus, the I2C*i*.I2C_BUF[5:0] XTRSH bit field must be configured to 0x0 (TX threshold = 1).

Table 18-21. I²C Register Call Summary for Sequence – Slave Transmitter/Receiver Mode, Polling

Register Name	Register Name
I2C*i*.I2C_IRQSTATUS_RAW (1)	I2C*i*.I2C_BUFSTAT (1)
I2C*i*.I2C_DATA (1)	I2C*i*.I2C_BUF (1)

(1) *i* = 1 to 2

Figure 18-25. I²C Slave Transmitter/Receiver Mode, Interrupt



(1) The XRDY, RDR, RRDY, and ARDY bits are cleared by writing 1 to each corresponding bit in the I2C*i*.I2C_IRQSTATUS register.

(2) In slave transmitter mode, the amount of data requested by the external master I²C device is unknown; thus, the I2C*i*.I2C_BUF[5:0] XTRSH bit field must be configured to 0x0 (TX threshold = 1).

Table 18-22. I²C Register Call Summary for Sequence – Slave Transmitter/Receiver Mode, Interrupt

Register Name	Register Name
I2C*i*.I2C_IRQSTATUS_RAW (1)	I2C*i*.I2C_BUFSTAT (1)
I2C*i*.I2C_DATA (1)	I2C*i*.I2C_BUF (1)

(1) *i* = 1 to 2

18.1.6 I²C Register Manual

18.1.6.1 I²C Instance Summary

Table 18-23 lists the base address and block size for the I²C module instances.

Table 18-23. I²C Instance Summary

Module Name	Module Base Address	Size
I2C1	0x4807 0000	214 Bytes
I2C2	0x4807 2000	214 Bytes

18.1.6.2 I²C Registers

18.1.6.2.1 I²C Register Summary

Table 18-24 provides the register summary and associated offset addresses for the two I ²C registers.

Table 18-24. I²C Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	I2C1 Physical Address	I2C2 Physical Address
I2C_REVNB_LO	R	16	0x0000 0000	0x4807 0000	0x4807 2000
I2C_REVNB_HI	R	16	0x0000 0004	0x4807 0004	0x4807 2004
I2C_SYSC	RW	16	0x0000 0010	0x4807 0010	0x4807 2010
I2C_EOI	W	16	0x0000 0020	0x4807 0020	0x4807 2020
I2C_IRQSTATUS_RAW	RW	16	0x0000 0024	0x4807 0024	0x4807 2024
I2C_IRQSTATUS	RW	16	0x0000 0028	0x4807 0028	0x4807 2028
I2C_IRQENABLE_SET	RW	16	0x0000 002C	0x4807 002C	0x4807 202C
I2C_IRQENABLE_CLR	RW	16	0x0000 0030	0x4807 0030	0x4807 2030
I2C_WE	RW	16	0x0000 0034	0x4807 0034	0x4807 2034
I2C_DMARXENABLE_SET	RW	16	0x0000 0038	0x4807 0038	0x4807 2038
I2C_DMATXENABLE_SET	RW	16	0x0000 003C	0x4807 003C	0x4807 203C
I2C_DMARXENABLE_CLR	RW	16	0x0000 0040	0x4807 0040	0x4807 2040
I2C_DMATXENABLE_CLR	RW	16	0x0000 0044	0x4807 0044	0x4807 2044
I2C_DMARXWAKE_EN	RW	16	0x0000 0048	0x4807 0048	0x4807 2048
I2C_DMATXWAKE_EN	RW	16	0x0000 004C	0x4807 004C	0x4807 204C
RESERVED	RW	16	0x0000 0084	0x4807 0084	0x4807 2084
RESERVED	RW	16	0x0000 0088	0x4807 0088	0x4807 2088
I2C_SYSS	RW	16	0x0000 0090	0x4807 0090	0x4807 2090
I2C_BUF	RW	16	0x0000 0094	0x4807 0094	0x4807 2094
I2C_CNT	RW	16	0x0000 0098	0x4807 0098	0x4807 2098
I2C_DATA	RW	16	0x0000 009C	0x4807 009C	0x4807 209C
I2C_CON	RW	16	0x0000 00A4	0x4807 00A4	0x4807 20A4
I2C_OA	RW	16	0x0000 00A8	0x4807 00A8	0x4807 20A8
I2C_SA	RW	16	0x0000 00AC	0x4807 00AC	0x4807 20AC
I2C_PSC	RW	16	0x0000 00B0	0x4807 00B0	0x4807 20B0
I2C_SCLL	RW	16	0x0000 00B4	0x4807 00B4	0x4807 20B4
I2C_SCLH	RW	16	0x0000 00B8	0x4807 00B8	0x4807 20B8
I2C_SYSTEST	RW	16	0x0000 00BC	0x4807 00BC	0x4807 20BC
I2C_BUFSTAT	R	16	0x0000 00C0	0x4807 00C0	0x4807 20C0

Table 18-24. I²C Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	I2C1 Physical Address	I2C2 Physical Address
I2C_OA1	RW	16	0x0000 00C4	0x4807 00C4	0x4807 20C4
I2C_OA2	RW	16	0x0000 00C8	0x4807 00C8	0x4807 20C8
I2C_OA3	RW	16	0x0000 00CC	0x4807 00CC	0x4807 20CC
I2C_ACTOA	R	16	0x0000 00D0	0x4807 00D0	0x4807 20D0
I2C_SBLOCK	RW	16	0x0000 00D4	0x4807 00D4	0x4807 20D4

18.1.6.2.2 I²C Register Description

Table 18-25 through Table 18-87 describe the individual I²C registers.

Table 18-25. I2C_REVNB_LO

Address Offset	0x0000 0000		
Physical Address	0x4807 0000 0x4807 2000	**Instance**	I2C1 I2C2
Description	IP Revision Identifier (X.Y.R) Used by software to track features, bugs, and compatibility		
Type	R		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REVISION															

Bits	Field Name	Description	Type	Reset
15:0	REVISION	IP Revision	R	TI internal data

Table 18-26. Register Call Summary for Register I2C_REVNB_LO

Multimaster I2C Controller
- I2C Register Summary: [0]

Table 18-27. I2C_REVNB_HI

Address Offset	0x0000 0004		
Physical Address	0x4807 0004 0x4807 2004	**Instance**	I2C1 I2C2
Description	IP Revision Identifier (X.Y.R) Used by software to track features, bugs, and compatibility		
Type	R		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REVISION															

Bits	Field Name	Description	Type	Reset
15:0	REVISION	IP Revision	R	TI internal data

Table 18-28. Register Call Summary for Register I2C_REVNB_HI

Multimaster I2C Controller
- I2C Register Summary: [0]

Table 18-29. I2C_SYSC

Address Offset	0x0000 0010		
Physical Address	0x4807 0010 0x4807 2010	**Instance**	I2C1 I2C2
Description	System Configuration register		
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED						CLKACTIVITY		RESERVED			IDLEMODE		ENAWAKEUP	SRST	AUTOIDLE

Bits	Field Name	Description	Type	Reset
15:10	RESERVED	Reserved	R	0x00
9:8	CLKACTIVITY	Clock Activity selection bits	RW	0x0
		0x0: Both clocks can be cut off		
		0x1: Only OCP clock must be kept active; system clock can be cut off		
		0x2: Only system clock must be kept active; OCP clock can be cut off		
		0x3: Both clocks must be kept active		
7:5	RESERVED	Reads return 0.	R	0x0
4:3	IDLEMODE	Idle Mode selection bits	RW	0x0
		0x0: Force Idle mode		
		0x1: No Idle mode		
		0x2: Smart Idle mode		
		0x3: Smart-idle wakeup-capable mode		
2	ENAWAKEUP	Enable Wakeup control bit	RW	0
		0x0: Wakeup mechanism is disabled		
		0x1: Wakeup mechanism is enabled		
1	SRST	SoftReset bit	RW	0
		0x0: Normal mode		
		0x1: The module is reset		
0	AUTOIDLE	Autoidle bit	RW	1
		0x0: Auto Idle mechanism is disabled		
		0x1: Auto Idle mechanism is enabled		

Table 18-30. Register Call Summary for Register I2C_SYSC

Multimaster I2C Controller

- I2C Software Reset: [0][1]
- I2C Power Management: [2][3][4][5][6]
- I2C Register Summary: [7]

Table 18-31. I2C_EOI

Address Offset	0x0000 0020		
Physical Address	0x4807 0020 0x4807 2020	**Instance**	I2C1 I2C2
Description	End Of Interrupt number specification		
Type	W		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED															LINE_NUMBER

Bits	Field Name	Description	Type	Reset
15:1	RESERVED	Reserved	R	0x0
0	LINE_NUMBER	Software End Of Interrupt (EOI) control. Write number of interrupt output.	W	0x0

Table 18-32. Register Call Summary for Register I2C_EOI

Multimaster I2C Controller
- I2C Register Summary: [0]

Table 18-33. I2C_IRQSTATUS_RAW

Address Offset	0x0000 0024		
Physical Address	0x4807 0024 0x4807 2024	**Instance**	I2C1 I2C2
Description	Per-event raw interrupt status vector. The raw status is set even if event is not enabled. Write 1 to set the (raw) status, mostly for debug		
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	XDR	RDR	BB	ROVR	XUDF	AAS	BF	AERR	STC	GC	XRDY	RRDY	ARDY	NACK	AL

Bits	Field Name	Description	Type	Reset
15	RESERVED	Write 0s for future compatibility. Read returns 0.	RW	0
14	XDR	Transmit draining IRQ status. 0x0: Transmit draining inactive. 0x1: Transmit draining enabled.	RW	0
13	RDR	Receive draining IRQ status. 0x0: Receive draining inactive. 0x1: Receive draining enabled.	RW	0
12	BB	Bus busy status. Writing into this bit has no effect. Read 0x1: Bus is occupied. Read 0x0: Bus is free.	R	0

Bits	Field Name	Description	Type	Reset
11	ROVR	Receive overrun status. Writing into this bit has no effect. Read 0x1: Receiver overrun. Read 0x0: Normal operation.	RW	0
10	XUDF	Transmit underflow status. Writing into this bit has no effect. Read 0x1: Transmit underflow. Read 0x0: Normal operation.	RW	0
9	AAS	Address recognized as slave IRQ status. 0x0: No action. 0x1: Address recognized.	RW	0
8	BF	Bus Free IRQ status. 0x0: No action. 0x1: Bus Free.	RW	0
7	AERR	Access Error IRQ status. 0x0: No action. 0x1: Access Error.	RW	0
6	STC	Start Condition IRQ status. 0x0: No action. 0x1: Start Condition detected.	RW	0

Bits	Field Name	Description	Type	Reset
5	GC	General call IRQ status. Set to 1 by core when General call address detected and interrupt signaled to IRQ_Crossbar. (1) 0x0: No general call detected. 0x1: General call address detected.	RW	0
4	XRDY	Transmit data ready IRQ status. Set to 1 by core when transmitter and when new data is requested. When set to 1 by core, an interrupt is signaled to IRQ_Crossbar. (1) 0x0: Transmision ongoing. 0x1: Transmit data ready.	RW	0
3	RRDY	Receive data ready IRQ status. Set to 1 by core when receiver mode, a new data is able to be read. When set to 1 by core, an interrupt is signaled to IRQ_Crossbar. (1) 0x0: No data available. 0x1: Receive data available.	RW	0
2	ARDY	Register access ready IRQ status. When set to 1 it indicates that previous access has been performed and registers are ready to be accessed again. An interrupt is signaled to IRQ_Crossbar. (1) 0x0: Module busy. 0x1: Access ready.	RW	0
1	NACK	No acknowledgement IRQ status. Bit is set when No Acknowledge has been received, an interrupt is signaled to IRQ_Crossbar. (1) 0x0: Normal operation. 0x1: Not Acknowledge detected.	RW	0
0	AL	Arbitration lost IRQ status. This bit is automatically set by the hardware when it loses the Arbitration in master transmit mode, an interrupt is signaled to IRQ_Crossbar. During reads, it always returns 0. 0x0: Normal operation. 0x1: Arbitration lost detected.	RW	0

(1) Writing 1 in the bit field will only set the respective field to 1, used mainly for debug.

Table 18-34. Register Call Summary for Register I2C_IRQSTATUS_RAW

Multimaster I2C Controller

- I2C Start and Stop Conditions: [0]
- I2C FIFO Polling Mode: [1][2][3][4][5][6]
- I2C Draining Feature: [7][8][9][10][11]
- I2C System Test Mode: [12][13][14][15][16][17][18][19]
- I2C Programming Model: [20][21][22][23][24][25][26][27][28][29][30][31][32][33][34][35][36][37][38][39][40][41][42][43]
- I2C Register Summary: [44]
- I2C Register Description: [45][46][47][48][49][50][51][52][53][54][55][56][57][58][59][60][61][62][63][64][65][66]

Table 18-35. I2C_IRQSTATUS

Address Offset	0x0000 0028		
Physical Address	0x4807 0028 0x4807 2028	**Instance**	I2C1 I2C2
Description	Per-event enabled interrupt status vector		
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	XDR	RDR	BB	ROVR	XUDF	AAS	BF	AERR	STC	GC	XRDY	RRDY	ARDY	NACK	AL

Bits	Field Name	Description	Type	Reset
15	RESERVED	Write 0s for future compatibility. Read returns 0.	RW	0
14	XDR	Transmit draining IRQ enabled status. 0x0: Transmit draining inactive. 0x1: Transmit draining enabled.	RW W1toClr	0
13	RDR	Receive draining IRQ enabled status. 0x0: Receive draining inactive. 0x1: Receive draining enabled.	RW W1toClr	0
12	BB	Bus busy enabled status. Writing into this bit has no effect. Read 0x1: Bus is occupied. Read 0x0: Bus is free.	R	0
11	ROVR	Receive overrun enabled status. Writing into this bit has no effect. Read 0x1: Receiver overrun. Read 0x0: Normal operation.	RW W1toClr	0
10	XUDF	Transmit underflow enabled status. Writing into this bit has no effect. Read 0x1: Transmit underflow. Read 0x0: Normal operation.	RW W1toClr	0
9	AAS	Address recognized as slave IRQ enabled status. 0x0: No action. 0x1: Address recognized.	RW W1toClr	0
8	BF	Bus Free IRQ enabled status. 0x0: No action. 0x1: Bus Free.	RW W1toClr	0
7	AERR	Access Error IRQ enabled status. 0x0: No action. 0x1: Access Error.	RW W1toClr	0
6	STC	Start Condition IRQ enabled status. 0x0: No action. 0x1: Start Condition detected.	RW W1toClr	0
5	GC	General call IRQ enabled status. Set to 1 by core when General call address detected and interrupt signaled to IRQ_Crossbar. Write 1 to clear. 0x0: No general call detected. 0x1: General call address detected.	RW W1toClr	0

Bits	Field Name	Description	Type	Reset
4	XRDY	Transmit data ready IRQ enabled status. Set to 1 by core when transmitter and when new data is requested. When set to 1 by core, an interrupt is signaled to IRQ_Crossbar. Write 1 to clear. 0x0: Transmision ongoing. 0x1: Transmit data ready.	RW W1toClr	0
3	RRDY	Receive data ready IRQ enabled status. Set to 1 by core when receiver mode, a new data is able to be read. When set to 1 by core, an interrupt is signaled to IRQ_Crossbar. Write 1 to clear. 0x0: No data available. 0x1: Receive data available.	RW W1toClr	0
2	ARDY	Register access ready IRQ enabled status. When set to 1 it indicates that previous access has been performed and registers are ready to be accessed again. An interrupt is signaled to IRQ_Crossbar. Write 1 to clear. 0x0: Module busy. 0x1: Access ready.	RW W1toClr	0
1	NACK	No acknowledgement IRQ enabled status. Bit is set when No Acknowledge has been received, an interrupt is signaled to IRQ_Crossbar. Write 1 to clear this bit. 0x0: Normal operation. 0x1: Not Acknowledge detected.	RW W1toClr	0
0	AL	Arbitration lost IRQ enabled status. This bit is automatically set by the hardware when it loses the Arbitration in master transmit mode, an interrupt is signaled to IRQ_Crossbar. During reads, it always returns 0. 0x0: Normal operation. 0x1: Arbitration lost detected.	RW W1toClr	0

Table 18-36. Register Call Summary for Register I2C_IRQSTATUS

Multimaster I2C Controller

- I2C Interrupt Requests: [0][1][2][3][4][5][6][7][8][9][10][11][12][13][14]
- I2C Programming Model: [15][16][17][18][19][20][21][22]
- I2C Register Summary: [23]

Table 18-37. I2C_IRQENABLE_SET

Address Offset	0x0000 002C		
Physical Address	0x4807 002C 0x4807 202C	**Instance**	I2C1 I2C2
Description	Per-event interrupt enable bit vector.		
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	XDR_IE	RDR_IE	RESERVED	ROVR	XUDF	AAS_IE	BF_IE	AERR_IE	STC_IE	GC_IE	XRDY_IE	RRDY_IE	ARDY_IE	NACK_IE	AL_IE

Bits	Field Name	Description	Type	Reset
15	RESERVED	Write 0s for future compatibility. Read returns 0.	R	0
14	XDR_IE	Transmit Draining interrupt enable set. Unmask the interrupt signaled by bit in I2C_IRQSTATUS_RAW[XDR]. Read: 0x0: Transmit Draining interrupt disabled 0x1: Transmit Draining interrupt enabled Write: 0x0: Has no effect 0x1: Enables the Transmit Draining interrupt	RW W1toSet	0
13	RDR_IE	Receive Draining interrupt enable set. Unmask the interrupt signaled by bit in I2C_IRQSTATUS_RAW [RDR]. Read: 0x0: Receive Draining interrupt disabled 0x1: Receive Draining interrupt enabled Write: 0x0: Has no effect 0x1: Enables the Receive Draining interrupt	RW W1toSet	0
12	RESERVED	Reserved	R	0
11	ROVR	Receive overrun enable set. Read: 0x0: Receive overrun interrupt disabled 0x1: Receive overrun interrupt enabled Write: 0x0: Has no effect 0x1: Enables the Receive overrun interrupt	RW W1toSet	0
10	XUDF	Transmit underflow enable set. Read: 0x0: Transmit underflow interrupt disabled 0x1: Transmit underflow interrupt enabled Write: 0x0: Has no effect 0x1: Enables the Transmit underflow interrupt	RW W1toSet	0
9	AAS_IE	Addressed as Slave interrupt enable set. Unmask the interrupt signaled by bit in I2C_IRQSTATUS_RAW [AAS]. Read: 0x0: Addressed as Slave interrupt disabled 0x1: Addressed as Slave interrupt enabled Write: 0x0: Has no effect 0x1: Enables the Addressed as Slave interrupt	RW W1toSet	0
8	BF_IE	Bus Free interrupt enable set. Unmask the interrupt signaled by bit in I2C_IRQSTATUS_RAW [BF]. Read: 0x0: Bus Free interrupt disabled 0x1: Bus Free interrupt enabled Write: 0x0: Has no effect 0x1: Enables the Bus Free interrupt	RW W1toSet	0

Bits	Field Name	Description	Type	Reset
7	AERR_IE	Access Error interrupt enable set. Unmask the interrupt signaled by bit in I2C_IRQSTATUS_RAW [AERR]. Read: 0x0: Access Error interrupt disabled 0x1: Access Error interrupt enabled Write: 0x0: Has no effect 0x1: Enables the Access Error interrupt	RW W1toSet	0
6	STC_IE	Start Condition interrupt enable set. Unmask the interrupt signaled by bit in I2C_IRQSTATUS_RAW [STC]. Read: 0x0: Start Condition interrupt disabled 0x1: Start Condition interrupt enabled Write: 0x0: Has no effect 0x1: Enables the Start Condition interrupt	RW W1toSet	0
5	GC_IE	General call Interrupt enable set. Unmask the interrupt signaled by bit in I2C_IRQSTATUS_RAW [GC] Read: 0x0: General call interrupt disabled 0x1: General call interrupt enabled Write: 0x0: Has no effect 0x1: Enables the General call interrupt	RW W1toSet	0
4	XRDY_IE	Transmit data ready interrupt enable set. Unmask the interrupt signaled by bit in I2C_IRQSTATUS_RAW [XRDY] Read: 0x0: Transmit data ready interrupt disabled 0x1: Transmit data ready interrupt enabled Write: 0x0: Has no effect 0x1: Enables the Transmit data ready interrupt	RW W1toSet	0
3	RRDY_IE	Receive data ready interrupt enable set. Unmask the interrupt signaled by bit in I2C_IRQSTTUS_RAW [RRDY] Read: 0x0: Receive data ready interrupt disabled 0x1: Receive data ready interrupt enabled Write: 0x0: Has no effect 0x1: Enables the Receive data ready interrupt	RW W1toSet	0
2	ARDY_IE	Register access ready interrupt enable set. Unmask the interrupt signaled by bit in I2C_IRQSTATUS_RAW [ARDY] Read: 0x0: Register access ready interrupt disabled 0x1: Register access ready interrupt enabled Write: 0x0: Has no effect 0x1: Enables the Register access ready interrupt	RW W1toSet	0

Bits	Field Name	Description	Type	Reset
1	NACK_IE	No acknowledgement interrupt enable set. Unmask the interrupt signaled by bit in I2C_IRQSTATUS_RAW [NACK] Read: 0x0: Not Acknowledge interrupt disabled 0x1: Not Acknowledge interrupt enabled Write: 0x0: Has no effect 0x1: Enables the Not Acknowledge interrupt	RW W1toSet	0
0	AL_IE	Arbitration lost interrupt enable set. Unmask the interrupt signaled by bit in I2C_IRQSTATUS_RAW [AL] Read: 0x0: Arbitration lost interrupt disabled 0x1: Arbitration lost interrupt enabled Write: 0x0: Has no effect 0x1: Enables the Arbitration lost interrupt	RW W1toSet	0

Table 18-38. Register Call Summary for Register I2C_IRQENABLE_SET

Multimaster I2C Controller

- I2C Interrupt Requests: [0][1][2][3][4][5][6][7][8][9][10][11][12][13]
- I2C FIFO Interrupt Mode: [14]
- I2C FIFO Polling Mode: [15][16]
- I2C Draining Feature: [17][18]
- I2C Programming Model: [19][20][21][22][23][24][25][26][27][28]
- I2C Register Summary: [29]

Table 18-39. I2C_IRQENABLE_CLR

Address Offset	0x0000 0030		
Physical Address	0x4807 0030 0x4807 2030	**Instance**	I2C1 I2C2
Description	Per-event interrupt clear bit vector.		
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	XDR_IE	RDR_IE	RESERVED	ROVR	XUDF	AAS_IE	BF_IE	AERR_IE	STC_IE	GC_IE	XRDY_IE	RRDY_IE	ARDY_IE	NACK_IE	AL_IE

Bits	Field Name	Description	Type	Reset
15	RESERVED	Write 0s for future compatibility. Read returns 0.	R	0
14	XDR_IE	Transmit Draining interrupt enable clear. Mask the interrupt signaled by bit in I2C_IRQSTATUS_RAW [XDR]. Read: 0x0: Transmit Draining interrupt disabled 0x1: Transmit Draining interrupt enabled Write: 0x0: Has no effect 0x1: Disables the Transmit Draining interrupt	RW W1toClr	0

Bits	Field Name	Description	Type	Reset
13	RDR_IE	Receive Draining interrupt enable clear. Mask the interrupt signaled by bit in I2C_IRQSTAUS_RAW [RDR]. Read: 0x0: Receive Draining interrupt disabled 0x1: Receive Draining interrupt enabled Write: 0x0: Has no effect 0x1: Disables the Receive Draining interrupt	RW W1toClr	0
12	RESERVED	Reserved	R	0
11	ROVR	Receive overrun enable clear. Read: 0x0: Receive overrun interrupt disabled 0x1: Receive overrun interrupt enabled Write: 0x0: Has no effect 0x1: Disables the Receive overrun interrupt	RW W1toClr	0
10	XUDF	Transmit underflow enable clear. Read: 0x0: Transmit underflow interrupt disabled 0x1: Transmit underflow interrupt enabled Write: 0x0: Has no effect 0x1: Disables the Transmit underflow interrupt	RW W1toClr	0
9	AAS_IE	Addressed as Slave interrupt enable clear. Mask the interrupt signaled by bit in I2C_IRQSTATUS_RAW [AAS]. Read: 0x0: Addressed as Slave interrupt disabled 0x1: Addressed as Slave interrupt enabled Write: 0x0: Has no effect 0x1: Disables the Addressed as Slave interrupt	RW W1toClr	0
8	BF_IE	Bus Free interrupt enable clear. Mask the interrupt signaled by bit in I2C_IRQSTATUS_RAW [BF]. Read: 0x0: Bus Free interrupt disabled 0x1: Bus Free interrupt enabled Write: 0x0: Has no effect 0x1: Disables the Bus Free interrupt	RW W1toClr	0
7	AERR_IE	Access Error interrupt enable clear. Mask the interrupt signaled by bit in I2C_IRQSTATUS_RAW [AERR]. Read: 0x0: Access Error interrupt disabled 0x1: Access Error interrupt enabled Write: 0x0: Has no effect 0x1: Disables the Access Error interrupt	RW W1toClr	0

Bits	Field Name	Description	Type	Reset
6	STC_IE	Start Condition interrupt enable clear. Mask the interrupt signaled by bit in I2C_IRQSTATUS_RAW [STC]. Read: 0x0: Start Condition interrupt disabled 0x1: Start Condition interrupt enabled Write: 0x0: Has no effect 0x1: Disables the Start Condition interrupt	RW W1toClr	0
5	GC_IE	General call Interrupt enable clear. Mask the interrupt signaled by bit in I2C_IRQSTATUS_RAW [GC] Read: 0x0: General call interrupt disabled 0x1: General call interrupt enabled Write: 0x0: Has no effect 0x1: Disables the General call interrupt	RW W1toClr	0
4	XRDY_IE	Transmit data ready interrupt enable clear. Mask the interrupt signaled by bit in I2C_IRQSTATUS_RAW [XRDY] Read: 0x0: Transmit data ready interrupt disabled 0x1: Transmit data ready interrupt enabled Write: 0x0: Has no effect 0x1: Disables the Transmit data ready interrupt	RW W1toClr	0
3	RRDY_IE	Receive data ready interrupt enable clear. Mask the interrupt signaled by bit in I2C_IRQSTAUS_RAW [RRDY] Read: 0x0: Receive data ready interrupt disabled 0x1: Receive data ready interrupt enabled Write: 0x0: Has no effect 0x1: Disables the Receive data ready interrupt	RW W1toClr	0
2	ARDY_IE	Register access ready interrupt enable clear. Mask the interrupt signaled by bit in I2C_IRQSTATUS_RAW [ARDY] Read: 0x0: Register access ready interrupt disabled 0x1: Register access ready interrupt enabled Write: 0x0: Has no effect 0x1: Disables the Register access ready interrupt	RW W1toClr	0
1	NACK_IE	No acknowledgement interrupt enable clear. Mask the interrupt signaled by bit in I2C_IRQSTATUS_RAW [NACK] Read: 0x0: Not Acknowledge interrupt disabled 0x1: Not Acknowledge interrupt enabled Write: 0x0: Has no effect 0x1: Disables the Not Acknowledge interrupt	RW W1toClr	0

Bits	Field Name	Description	Type	Reset
0	AL_IE	Arbitration lost interrupt enable clear. Mask the interrupt signaled by bit in I2C_IRQSTATUS_RAW[AL] Read: 0x0: Arbitration lost interrupt disabled 0x1: Arbitration lost interrupt enabled Write: 0x0: Has no effect 0x1: Disables the Arbitration lost interrupt	RW W1toClr	0

Table 18-40. Register Call Summary for Register I2C_IRQENABLE_CLR

Multimaster I2C Controller

- I2C Interrupt Requests: [0][1][2][3][4][5][6][7][8][9][10][11][12][13]
- I2C FIFO Interrupt Mode: [14]
- I2C Register Summary: [15]

Table 18-41. I2C_WE

Address Offset	0x0000 0034		
Physical Address	0x4807 0034 0x4807 2034	**Instance**	I2C1 I2C2
Description	I2C wakeup enable vector.		
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	XDR	RDR	RESERVED	ROVR	XUDF	AAS	BF	RESERVED	STC	GC	RESERVED	DRDY	ARDY	NACK	AL

Bits	Field Name	Description	Type	Reset
15	RESERVED	Reserved	R	0
14	XDR	Transmit Draining wakeup set. 0x0: Transmit draining wakeup disabled 0x1: Transmit draining wakeup enabled	RW	0
13	RDR	Receive Draining wakeup set. 0x0: Receive draining wakeup disabled 0x1: Receive draining wakeup enabled	RW	0
12	RESERVED	Reserved	R	0
11	ROVR	Receive overrun wakeup set. 0x0: Receive overrun wakeup disabled 0x1: Receive overrun wakeup enabled	RW	0
10	XUDF	Transmit underflow wakeup set. 0x0: Transmit underflow wakeup disabled 0x1: Transmit underflow wakeup enabled	RW	0
9	AAS	Address as slave IRQ wakeup set. 0x0: Addressed as slave wakeup disabled 0x1: Addressed as slave wakeup enabled	RW	0
8	BF	Bus Free IRQ wakeup set. 0x0: Bus Free wakeup disabled 0x1: Bus Free wakeup enabled	RW	0

Bits	Field Name	Description	Type	Reset
7	RESERVED	Reserved	R	0
6	STC	Start Condition IRQ wakeup set. 0x0: Start condition wakeup disabled 0x1: Start condition wakeup enabled	RW	0
5	GC	General call IRQ wakeup set. 0x0: General call wakeup disabled 0x1: General call wakeup enabled	RW	0
4	RESERVED	Reserved	R	0
3	DRDY	Receive/Transmit data ready IRQ wakeup set. 0x0: Transmit/receive data ready wakeup disabled 0x1: Transmit/receive data ready wakeup enabled	RW	0
2	ARDY	Register access ready IRQ wakeup set. 0x0: Register access ready wakeup disabled 0x1: Register access ready wakeup enabled	RW	0
1	NACK	No acknowledgment IRQ wakeup set. 0x0: Not Acknowledge wakeup disabled 0x1: Not Acknowledge wakeup enabled	RW	0
0	AL	Arbitration lost IRQ wakeup set. 0x0: Arbitration lost wakeup disabled 0x1: Arbitration lost wakeup enabled	RW	0

Table 18-42. Register Call Summary for Register I2C_WE

Multimaster I2C Controller

- I2C Register Summary: [0]

Table 18-43. I2C_DMARXENABLE_SET

Address Offset	0x0000 0038		
Physical Address	0x4807 0038 0x4807 2038	**Instance**	I2C1 I2C2
Description	Per-event DMA RX enable set.		
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED															DMARX_ENABLE_SET

Bits	Field Name	Description	Type	Reset
15:1	RESERVED	Reserved	R	0x0000
0	DMARX_ENABLE_SET	Receive DMA channel enable set.	RW	0

Table 18-44. Register Call Summary for Register I2C_DMARXENABLE_SET

Multimaster I2C Controller

- I2C Programming Model: [0]
- I2C Register Summary: [1]

Table 18-45. I2C_DMATXENABLE_SET

Address Offset	0x0000 003C		
Physical Address	0x4807 003C 0x4807 203C	**Instance**	I2C1 I2C2
Description	Per-event DMA TX enable set.		
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED															DMATX_ENABLE_SET

Bits	Field Name	Description	Type	Reset
15:1	RESERVED	Reserved	R	0x0000
0	DMATX_ENABLE_SET	Transmit DMA channel enable set.	RW	0

Table 18-46. Register Call Summary for Register I2C_DMATXENABLE_SET

Multimaster I2C Controller

- I2C Programming Model: [0]
- I2C Register Summary: [1]

Table 18-47. I2C_DMARXENABLE_CLR

Address Offset	0x0000 0040		
Physical Address	0x4807 0040 0x4807 2040	**Instance**	I2C1 I2C2
Description	Per-event DMA RX enable clear.		
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED															DMARX_ENABLE_CLEAR

Bits	Field Name	Description	Type	Reset
15:1	RESERVED	Reserved	R	0x0000
0	DMARX_ENABLE_CLEAR	Receive DMA channel enable clear.	RW	0

Table 18-48. Register Call Summary for Register I2C_DMARXENABLE_CLR

Multimaster I2C Controller
- I2C Register Summary: [0]

Table 18-49. I2C_DMATXENABLE_CLR

Address Offset	0x0000 0044		
Physical Address	0x4807 0044 0x4807 2044	**Instance**	I2C1 I2C2
Description	Per-event DMA TX enable clear.		
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED															DMATX_ENABLE_CLEAR

Bits	Field Name	Description	Type	Reset
15:1	RESERVED	Reserved	R	0x0000
0	DMATX_ENABLE_CLEAR	Transmit DMA channel enable clear.	RW	0

Table 18-50. Register Call Summary for Register I2C_DMATXENABLE_CLR

Multimaster I2C Controller
- I2C Register Summary: [0]

Table 18-51. I2C_DMARXWAKE_EN

Address Offset	0x0000 0048		
Physical Address	0x4807 0048 0x4807 2048	**Instance**	I2C1 I2C2
Description	Per-event DMA RX wakeup enable.		
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	XDR	RDR	RESERVED	ROVR	XUDF	AAS	BF	RESERVED	STC	GC	RESERVED	DRDY	ARDY	NACK	AL

Bits	Field Name	Description	Type	Reset
15	RESERVED	Reserved	R	0
14	XDR	Transmit Draining wakeup set. 0x0: Transmit draining wakeup disabled 0x1: Transmit draining wakeup enabled	RW	0
13	RDR	Receive Draining wakeup set. 0x0: Receive draining wakeup disabled 0x1: Receive draining wakeup enabled	RW	0
12	RESERVED	Reserved	R	0
11	ROVR	Receive overrun wakeup set. 0x0: Receive overrun wakeup disabled 0x1: Receive overrun wakeup enabled	RW	0
10	XUDF	Transmit underflow wakeup set. 0x0: Transmit underflow wakeup disabled 0x1: Transmit underflow wakeup enabled	RW	0
9	AAS	Address as slave IRQ wakeup set. 0x0: Addressed as slave wakeup disabled 0x1: Addressed as slave wakeup enabled	RW	0
8	BF	Bus Free IRQ wakeup set. 0x0: Bus Free wakeup disabled 0x1: Bus Free wakeup enabled	RW	0
7	RESERVED	Reserved	R	0
6	STC	Start Condition IRQ wakeup set. 0x0: Start condition wakeup disabled 0x1: Start condition wakeup enabled	RW	0
5	GC	General call IRQ wakeup set. 0x0: General call wakeup disabled 0x1: General call wakeup enabled	RW	0
4	RESERVED	Reserved	R	0
3	DRDY	Receive/Transmit data ready IRQ wakeup set. 0x0: Transmit/receive data ready wakeup disabled 0x1: Transmit/receive data ready wakeup enabled	RW	0
2	ARDY	Register access ready IRQ wakeup set. 0x0: Register access ready wakeup disabled 0x1: Register access ready wakeup enabled	RW	0
1	NACK	No acknowledgment IRQ wakeup set. 0x0: Not Acknowledge wakeup disabled 0x1: Not Acknowledge wakeup enabled	RW	0
0	AL	Arbitration lost IRQ wakeup set. 0x0: Arbitration lost wakeup disabled 0x1: Arbitration lost wakeup enabled	RW	0

Table 18-52. Register Call Summary for Register I2C_DMARXWAKE_EN

Multimaster I2C Controller

- I2C Register Summary: [0]

Table 18-53. I2C_DMATXWAKE_EN

Address Offset	0x0000 004C		
Physical Address	0x4807 004C 0x4807 204C	**Instance**	I2C1 I2C2
Description	Per-event DMA TX wakeup enable.		
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	XDR	RDR	RESERVED	ROVR	XUDF	AAS	BF	RESERVED	STC	GC	RESERVED	DRDY	ARDY	NACK	AL

Bits	Field Name	Description	Type	Reset
15	RESERVED	Reserved	R	0
14	XDR	Transmit Draining wakeup set. 0x0: Transmit draining wakeup disabled 0x1: Transmit draining wakeup enabled	RW	0
13	RDR	Receive Draining wakeup set. 0x0: Receive draining wakeup disabled 0x1: Receive draining wakeup enabled	RW	0
12	RESERVED	Reserved	R	0
11	ROVR	Receive overrun wakeup set. 0x0: Receive overrun wakeup disabled 0x1: Receive overrun wakeup enabled	RW	0
10	XUDF	Transmit underflow wakeup set. 0x0: Transmit underflow wakeup disabled 0x1: Transmit underflow wakeup enabled	RW	0
9	AAS	Address as slave IRQ wakeup set. 0x0: Addressed as slave wakeup disabled 0x1: Addressed as slave wakeup enabled	RW	0
8	BF	Bus Free IRQ wakeup set. 0x0: Bus Free wakeup disabled 0x1: Bus Free wakeup enabled	RW	0
7	RESERVED	Reserved	R	0
6	STC	Start Condition IRQ wakeup set. 0x0: Start condition wakeup disabled 0x1: Start condition wakeup enabled	RW	0
5	GC	General call IRQ wakeup set. 0x0: General call wakeup disabled 0x1: General call wakeup enabled	RW	0
4	RESERVED	Reserved	R	0
3	DRDY	Receive/Transmit data ready IRQ wakeup set. 0x0: Transmit/receive data ready wakeup disabled 0x1: Transmit/receive data ready wakeup enabled	RW	0
2	ARDY	Register access ready IRQ wakeup set. 0x0: Register access ready wakeup disabled 0x1: Register access ready wakeup enabled	RW	0

Bits	Field Name	Description	Type	Reset
1	NACK	No acknowledgment IRQ wakeup set. 0x0: Not Acknowledge wakeup disabled 0x1: Not Acknowledge wakeup enabled	RW	0
0	AL	Arbitration lost IRQ wakeup set. 0x0: Arbitration lost wakeup disabled 0x1: Arbitration lost wakeup enabled	RW	0

Table 18-54. Register Call Summary for Register I2C_DMATXWAKE_EN

Multimaster I2C Controller
- I2C Register Summary: [0]

Table 18-55. I2C_SYSS

Address Offset	0x0000 0090		
Physical Address	0x4807 0090 0x4807 2090	**Instance**	I2C1 I2C2
Description	System Status register		
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED															RDONE

Bits	Field Name	Description	Type	Reset
15:1	RESERVED	Reserved	R	0x0000
0	RDONE	Reset done bit Read 0x1: Reset completed Read 0x0: Internal module reset in on-going	RW	1

Table 18-56. Register Call Summary for Register I2C_SYSS

Multimaster I2C Controller
- I2C Software Reset: [0][1][2]
- I2C Register Summary: [3]

Table 18-57. I2C_BUF

Address Offset	0x0000 0094		
Physical Address	0x4807 0094 0x4807 2094	**Instance**	I2C1 I2C2
Description	Buffer Configuration register		
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RDMA_EN	RXFIFO_CLR	RXTRSH						XDMA_EN	TXFIFO_CLR	TXTRSH					

Bits	Field Name	Description	Type	Reset
15	RDMA_EN	Receive DMA channel enable 0x0: Receive DMA channel disabled 0x1: Receive DMA channel enabled	RW	0
14	RXFIFO_CLR	Receive FIFO clear 0x0: Normal mode 0x1: Rx FIFO is reset	RW	0
13:8	RXTRSH	Threshold value for FIFO buffer in RX mode	RW	0x00
7	XDMA_EN	Transmit DMA channel enable 0x0: Transmit DMA channel disabled 0x1: Transmit DMA channel enabled	RW	0
6	TXFIFO_CLR	Transmit FIFO clear 0x0: Normal mode 0x1: Tx FIFO is reset	RW	0
5:0	TXTRSH	Threshold value for FIFO buffer in TX mode	RW	0x00

Table 18-58. Register Call Summary for Register I2C_BUF

Multimaster I2C Controller
- I2C FIFO Interrupt Mode: [0][1][2][3][4][5]
- I2C FIFO DMA Mode: [6][7][8][9][10]
- I2C Draining Feature: [11][12][13][14][15][16][17][18]
- I2C Programming Model: [19][20][21][22][23][24][25][26][27][28][29][30][31][32][33][34][35][36]
- I2C Register Summary: [37]

Table 18-59. I2C_CNT

Address Offset	0x0000 0098		
Physical Address	0x4807 0098 0x4807 2098	**Instance**	I2C1 I2C2
Description	Data counter register		
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DCOUNT															

Bits	Field Name	Description	Type	Reset
15:0	DCOUNT	Data count	RW	0x0000

Table 18-60. Register Call Summary for Register I2C_CNT

Multimaster I2C Controller
- I2C Draining Feature: [0]
- I2C Programming Model: [1][2][3][4][5][6][7][8][9]
- I2C Register Summary: [10]

Table 18-61. I2C_DATA

Address Offset	0x0000 009C		
Physical Address	0x4807 009C 0x4807 209C	**Instance**	I2C1 I2C2
Description	Data access register		
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DATA							

Bits	Field Name	Description	Type	Reset
15:8	RESERVED	Reserved	R	0x00
7:0	DATA	Transmit/Receive data FIFO endpoint	RW	0x--

Table 18-62. Register Call Summary for Register I2C_DATA

Multimaster I2C Controller

- I2C DMA Requests: [0][1][2][3]
- I2C System Test Mode: [4][5]
- I2C Programming Model: [6][7][8][9][10][11][12][13][14][15][16][17]
- I2C Register Summary: [18]

Table 18-63. I2C_CON

Address Offset	0x0000 00A4		
Physical Address	0x4807 00A4 0x4807 20A4	**Instance**	I2C1 I2C2
Description	I2C configuration register.		
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
I2C_EN	RESERVED	OPMODE		STB	MST	TRX	XSA	XOA0	XOA1	XOA2	XOA3	RESERVED		STP	STT

Bits	Field Name	Description	Type	Reset
15	I2C_EN	I2C module enable. 0x0: Controller in reset. FIFO are cleared and status bits are set to their default value 0x1: Module enabled	RW	0
14	RESERVED	Reserved	R	0
13:12	OPMODE	Operation mode selection. 0x0: I2C Fast/Standard mode. 0x1: Reserved. 0x3: Reserved. 0x2: Reserved	RW	0x0
11	STB	Start byte mode (master mode only). 0x0: Normal mode 0x1: Start byte mode	RW	0

Bits	Field Name	Description	Type	Reset
10	MST	Master/slave mode. 0x0: Slave mode 0x1: Master mode	RW	0
9	TRX	Transmitter/Receiver mode (master mode only). 0x0: Receiver mode 0x1: Transmitter mode	RW	0
8	XSA	Expand Slave address. 0x0: 7-bit address mode 0x1: 10-bit address mode	RW	0
7	XOA0	Expand Own address 0. 0x0: 7-bit address mode 0x1: 10-bit address mode	RW	0
6	XOA1	Expand Own address 1. 0x0: 7-bit address mode 0x1: 10-bit address mode	RW	0
5	XOA2	Expand Own address 2. 0x0: 7-bit address mode 0x1: 10-bit address mode	RW	0
4	XOA3	Expand Own address 3. 0x0: 7-bit address mode 0x1: 10-bit address mode	RW	0
3:2	RESERVED	Reserved	R	0x0
1	STP	Stop condition (master mode only). 0x0: No action or stop condition detected 0x1: Stop condition queried	RW	0
0	STT	Start condition (master mode only). 0x0: No action or start condition detected 0x1: Start condition queried	RW	0

Table 18-64. Register Call Summary for Register I2C_CON

Multimaster I2C Controller

- I2C Block Diagram: [0][1]
- I2C Clocking: [2]
- I2C Software Reset: [3][4][5][6][7]
- I2C Programmable Multislave Channel Feature: [8][9][10][11]
- I2C FIFO Interrupt Mode: [12][13]
- I2C System Test Mode: [14]
- I2C Programming Model: [16][17][18][19][20][21][22][23][24][25][26][27][28][29][30][31][32][33][34][35][36][37][38][39][40][41][42][43][44][45][46][47][48][49][50][51][52][53][54][55][56][57]
- I2C Register Summary: [58]

Table 18-65. I2C_OA

Address Offset	0x0000 00A8		
Physical Address	0x4807 00A8 0x4807 20A8	**Instance**	I2C1 I2C2
Description	Own address register		
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MCODE			RESERVED					OA							

Bits	Field Name	Description	Type	Reset
15:13	MCODE	Master Code	RW	0x0
12:10	RESERVED	Reserved	R	0x0
9:0	OA	Own address	RW	0x000

Table 18-66. Register Call Summary for Register I2C_OA

Multimaster I2C Controller

- I2C Automatic Blocking of the I2C Clock Feature: [0][1]
- I2C Programmable Multislave Channel Feature: [2][3]
- I2C Programming Model: [4][5][6][7]
- I2C Register Summary: [8]

Table 18-67. I2C_SA

Address Offset	0x0000 00AC		
Physical Address	0x4807 00AC 0x4807 20AC	**Instance**	I2C1 I2C2
Description	Slave address register		
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								SA							

Bits	Field Name	Description	Type	Reset
15:10	RESERVED	Reserved	R	0x00
9:0	SA	Slave address	RW	0x3FF

Table 18-68. Register Call Summary for Register I2C_SA

Multimaster I2C Controller

- I2C Programming Model: [0][1][2][3][4][5][6][7]
- I2C Register Summary: [8]

Table 18-69. I2C_PSC

Address Offset	0x0000 00B0		
Physical Address	0x4807 00B0 0x4807 20B0	**Instance**	I2C1 I2C2
Description	I2C Clock Prescaler Register		
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								PSC							

Bits	Field Name	Description	Type	Reset
15:8	RESERVED	Reserved	R	0x00
7:0	PSC	Fast/Standard mode prescale sampling clock divider value 0x0: Divide by 1 0x1: Divide by 2 ..... 0xFF: Divide by 256	RW	0x00

Table 18-70. Register Call Summary for Register I2C_PSC

Multimaster I2C Controller

- I2C Clocking: [0][1][2]
- I2C Noise Filter: [4][5]
- I2C System Test Mode: [6]
- I2C Programming Model: [7][8]
- I2C Register Summary: [9]

Table 18-71. I2C_SCLL

Address Offset	0x0000 00B4		
Physical Address	0x4807 00B4 0x4807 20B4	**Instance**	I2C1 I2C2
Description	I2C SCL Low Time Register.		
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								SCLL							

Bits	Field Name	Description	Type	Reset
15:8	RESERVED	Reserved	R	0x00
7:0	SCLL	Fast/standard mode SCL low time The value of the bit field is automatically increased by 7.	RW	0x00

Table 18-72. Register Call Summary for Register I2C_SCLL

Multimaster I2C Controller

- I2C Clocking: [0][2][6][7]
- I2C System Test Mode: [10]
- I2C Programming Model: [11][13]
- I2C Register Summary: [14]

Table 18-73. I2C_SCLH

Address Offset	0x0000 00B8		
Physical Address	0x4807 00B8 0x4807 20B8	**Instance**	I2C1 I2C2
Description	I2C SCL High Time Register.		
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								SCLH							

Bits	Field Name	Description	Type	Reset
15:8	RESERVED	Reserved	R	0x00
7:0	SCLH	Fast/standard mode SCL high time The value of the bit field is automatically increased by 5.	RW	0x00

Table 18-74. Register Call Summary for Register I2C_SCLH

Multimaster I2C Controller

- I2C Clocking: [0][2][5][6]
- I2C System Test Mode: [9]
- I2C Programming Model: [10][12]
- I2C Register Summary: [13]

Table 18-75. I2C_SYSTEST

Address Offset	0x0000 00BC		
Physical Address	0x4807 00BC 0x4807 20BC	**Instance**	I2C1 I2C2
Description	I2C System Test Register.		
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ST_EN	FREE	TMODE		SSB	RESERVED		SCL_I_FUNC	SCL_O_FUNC	SDA_I_FUNC	SDA_O_FUNC	RESERVED	SCL_I	SCL_O	SDA_I	SDA_O

Bits	Field Name	Description	Type	Reset
15	ST_EN	System test enable. 0x0: Normal mode. All others bits in register are read only 0x1: System test enabled. Permit other system test registers bits to be set	RW	0
14	FREE	Free running mode (on breakpoint) 0x0: Stop mode (on breakpoint condition). If Master mode, it stops after completion of the ongoing bit transfer. In slave mode, it stops during the phase transfer when 1 byte is completely transmitted/received. 0x1: Free running mode	RW	0

Bits	Field Name	Description	Type	Reset
13:12	TMODE	Test mode select. 0x0: Functional mode (default) 0x1: Reserved 0x3: Loop back mode select + SDA/SCL IO mode select 0x2: Test of SCL counters (SCLL, SCLH, PSC). SCL provides a permanent clock with master mode.	RW	0x0
11	SSB	Set all status bits in I2C_IRQSTATUS_RAW [14:0]. 0x0: No action 0x1: Set interrupt status bits to 1.	RW	0
10:9	RESERVED	Reserved	R	0x0
8	SCL_I_FUNC	SCL line input value (functional mode). Read 0x1: Read 1 from SCL line Read 0x0: Read 0 from SCL line	R	1
7	SCL_O_FUNC	SCL line output value (functional mode). Read 0x1: Driven 1 on SCL line Read 0x0: Driven 0 on SCL line	R	1
6	SDA_I_FUNC	SDA line input value (functional mode). Read 0x1: Read 1 from SDA line Read 0x0: Read 0 from SDA line	R	1
5	SDA_O_FUNC	SDA line output value (functional mode). Read 0x1: Driven 1 to SDA line Read 0x0: Driven 0 to SDA line	R	1
4	RESERVED	Reserved	R	0
3	SCL_I	SCL line sense input value Read 0x1: Read 1 from SCL line Read 0x0: Read 0 from SCL line	R	0
2	SCL_O	SCL line drive output value. 0x0: Write 0 to SCL line 0x1: Write 1 to SCL line	RW	0
1	SDA_I	SDA line sense input value. Read 0x1: Read 1 from SDA line Read 0x0: Read 0 from SDA line	R	0
0	SDA_O	SDA line drive output value. 0x0: Write 0 to SDA line 0x1: Write 1 to SDA line	RW	0

Table 18-76. Register Call Summary for Register I2C_SYSTEST

Multimaster I2C Controller

- I2C System Test Mode: [0][1][2][3][4][5][6][7][8][9]
- I2C Register Summary: [10]

Table 18-77. I2C_BUFSTAT

Address Offset	0x0000 00C0		
Physical Address	0x4807 00C0 0x4807 20C0	**Instance**	I2C1 I2C2
Description	I2C Buffer Status Register.		
Type	R		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FIFODEPTH		RXSTAT						RESERVED		TXSTAT					

Bits	Field Name	Description	Type	Reset
15:14	FIFODEPTH	Internal FIFO buffers depth. Read 0x0: 8-bytes FIFO. Read 0x1: 16-bytes FIFO. Read 0x2: 32-bytes FIFO. Read 0x3: 64-bytes FIFO.	R	0x1
13:8	RXSTAT	RX Buffer Status	R	0x00
7:6	RESERVED	Reserved	R	0x0
5:0	TXSTAT	TX Buffer Status.	R	0x00

Table 18-78. Register Call Summary for Register I2C_BUFSTAT

Multimaster I2C Controller

- I2C FIFO Management: [0]
- I2C Draining Feature: [1][2][3][4][5][6][7][8]
- I2C Programming Model: [9][10][11][12][13][14][15][16]
- I2C Register Summary: [17]

Table 18-79. I2C_OA1

Address Offset	0x0000 00C4		
Physical Address	0x4807 00C4 0x4807 20C4	**Instance**	I2C1 I2C2
Description	I2C Own Address 1 Register		
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								OA1							

Bits	Field Name	Description	Type	Reset
15:10	RESERVED	Reserved	R	0x00
9:0	OA1	Own address 1	RW	0x000

Table 18-80. Register Call Summary for Register I2C_OA1

Multimaster I2C Controller

- I2C Register Summary: [0]

Table 18-81. I2C_OA2

Address Offset	0x0000 00C8		
Physical Address	0x4807 00C8 0x4807 20C8	**Instance**	I2C1 I2C2
Description	I2C Own Address 2 Register		
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								OA2							

Bits	Field Name	Description	Type	Reset
15:10	RESERVED	Reserved	R	0x00
9:0	OA2	Own address 2	RW	0x000

Table 18-82. Register Call Summary for Register I2C_OA2

Multimaster I2C Controller
- I2C Register Summary: [0]

Table 18-83. I2C_OA3

Address Offset	0x0000 00CC		
Physical Address	0x4807 00CC 0x4807 20CC	**Instance**	I2C1 I2C2
Description	I2C Own Address 3 Register		
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								OA3							

Bits	Field Name	Description	Type	Reset
15:10	RESERVED	Reserved	R	0x00
9:0	OA3	Own address 3	RW	0x000

Table 18-84. Register Call Summary for Register I2C_OA3

Multimaster I2C Controller
- I2C Register Summary: [0]

Table 18-85. I2C_ACTOA

Address Offset	0x0000 00D0		
Physical Address	0x4807 00D0 0x4807 20D0	**Instance**	I2C1 I2C2
Description	I2C Active Own Address Register.		
Type	R		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												OA3_ACT	OA2_ACT	OA1_ACT	OA0_ACT

Bits	Field Name	Description	Type	Reset
15:4	RESERVED	Reserved	R	0x000
3	OA3_ACT	Own Address 3 active. Read 0x1: Own Address active. Read 0x0: Own Address inactive.	R	0
2	OA2_ACT	Own Address 2 active. Read 0x1: Own Address active. Read 0x0: Own Address inactive.	R	0
1	OA1_ACT	Own Address 1 active. Read 0x1: Own Address active. Read 0x0: Own Address inactive.	R	0
0	OA0_ACT	Own Address 0 active. Read 0x1: Own Address active. Read 0x0: Own Address inactive.	R	0

Table 18-86. Register Call Summary for Register I2C_ACTOA

Multimaster I2C Controller
- I2C Programmable Multislave Channel Feature: [0]
- I2C Register Summary: [1]

Table 18-87. I2C_SBLOCK

Address Offset	0x0000 00D4		
Physical Address	0x4807 00D4 0x4807 20D4	**Instance**	I2C1 I2C2
Description	I2C Clock Blocking Enable Register.		
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												OA3_EN	OA2_EN	OA1_EN	OA0_EN

Bits	Field Name	Description	Type	Reset
15:4	RESERVED	Reserved	R	0x000
3	OA3_EN	Enable I2C Clock Blocking for Own Address 3. 0x0: I2C Clock Released. 0x1: I2C Clock Blocked.	RW	0
2	OA2_EN	Enable I2C Clock Blocking for Own Address 2. 0x0: I2C Clock Released. 0x1: I2C Clock Blocked.	RW	0
1	OA1_EN	Enable I2C Clock Blocking for Own Address 1. 0x0: I2C Clock Released. 0x1: I2C Clock Blocked.	RW	0
0	OA0_EN	Enable I2C Clock Blocking for Own Address 0. 0x0: I2C Clock Released. 0x1: I2C Clock Blocked.	RW	0

Table 18-88. Register Call Summary for Register I2C_SBLOCK

Multimaster I2C Controller • I2C Automatic Blocking of the I2C Clock Feature: [0] • I2C Register Summary: [1]

18.2 UART

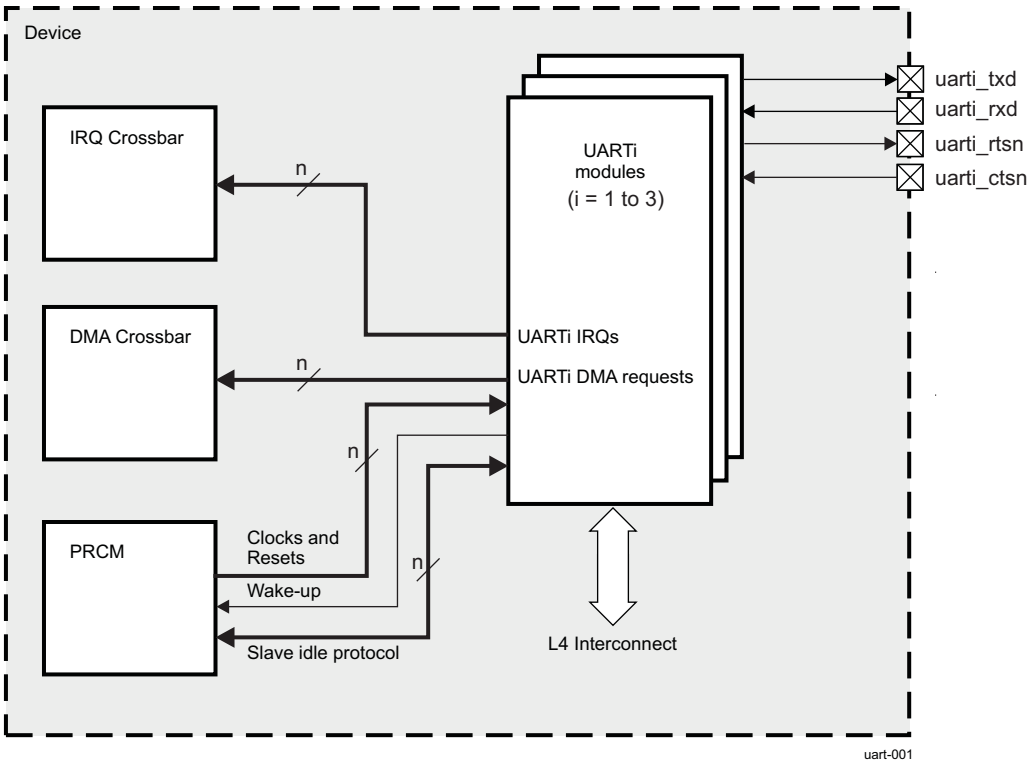
This chapter describes the function, operation, and configuration of the universal asynchronous receiver/transmitter (UART) module in the device.

18.2.1 UART Overview

The UART is a simple L4 slave peripheral that utilizes the EDMA for data transfer or IRQ polling via CPU. There are 3 UART modules in the device. Each UART can be used for configuration and data exchange with a number of external peripheral devices or interprocessor communication between devices.

Figure 18-26 shows the UART module overview.

Figure 18-26. UART Overview



18.2.1.1 UART Features

The UARTi (where i = 1 to 3) include the following features:

- 16C750 compatibility
- 64-byte FIFO buffer for receiver and 64-byte FIFO for transmitter
- Programmable interrupt trigger levels for FIFOs
- Baud generation based on programmable divisors N (where N = 1…16,384) operating from a fixed functional clock of 48 MHz or 192 MHz

Oversampling is programmed by software as 16 or 13. Thus, the baud rate computation is one of two options:

- Baud rate = (functional clock / 16) / N
- Baud rate = (functional clock / 13) / N
- This software programming mode enables higher baud rates with the same error amount without changing the clock source
- Break character detection and generation
- Configurable data format:

  - Data bit: 5, 6, 7, or 8 bits
  - Parity bit: Even, odd, none
  - Stop-bit: 1, 1.5, 2 bit(s)
- Flow control: Hardware (RTS/CTS) or software (XON/XOFF)
- The 48 MHz functional clock option allows baud rates up to 3.6Mbps
- The 192 MHz functional clock option allows baud rates up to 12Mbps

18.2.2 UART Environment

This section describes the UART connection with an external device.

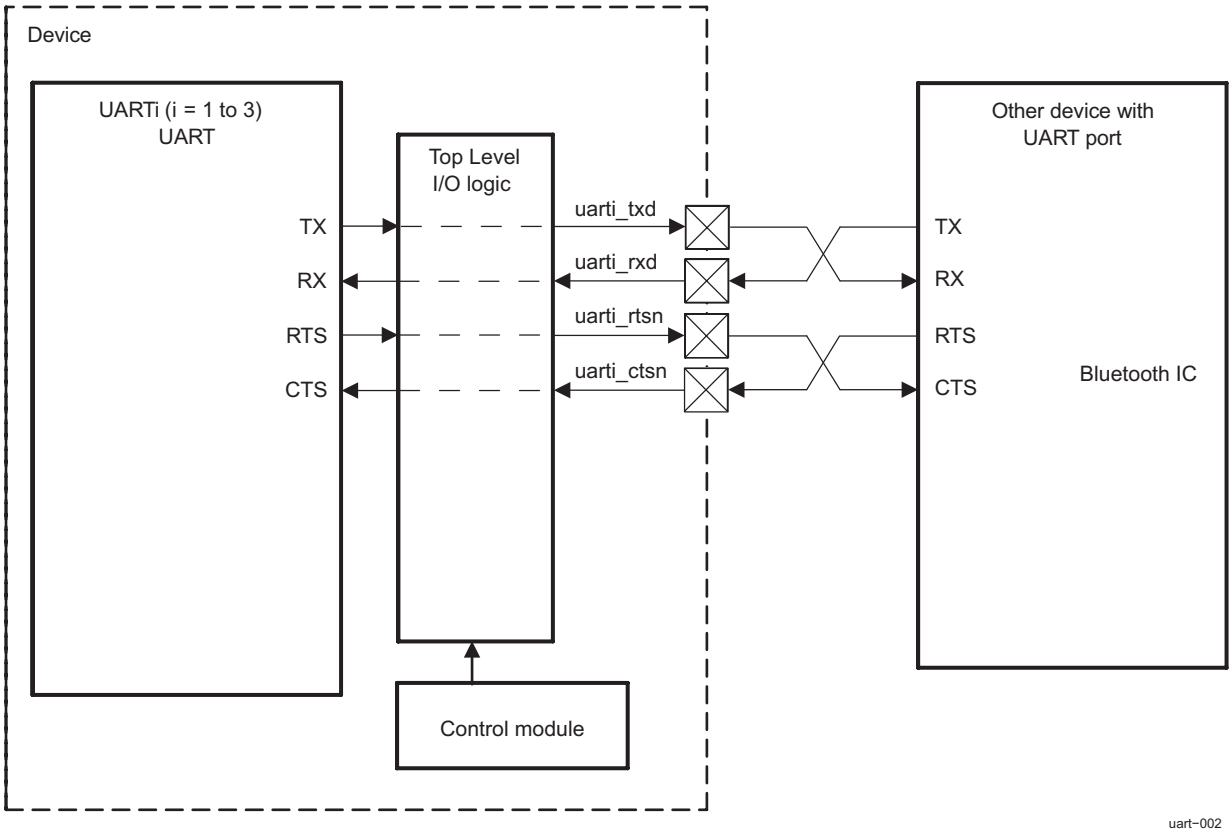
- The UART interface is described in Section 18.2.2.1.

18.2.2.1 UART Interface

18.2.2.1.1 System Using UART Communication With Hardware Handshake

Each UART instance can be easily connected to the UART port of an external IC (seeFigure 18-27).

Figure 18-27. UART Mode Bus System Overview



18.2.2.1.2 UART Interface Description

Table 18-89 lists the UART interface input/output (I/O) signals.

Table 18-89. UART Interface Signals

Signal	I/O[1]	Description	Module Level Reset Value
UARTi Interface Signals[2]			
uarti_rxd	I	Serial data input.	Unknown
uarti_txd	O	Serial data output This pin is set to low on reset (when the UARTi.UART_MDR1[2:0] bit field is set to 0x7) and takes the inactive level when the UARTi.UART_MDR1 register is programmed; that is, the output is 1 (inactive for UART modem modes).	0
uarti_ctsn	I	Clear to send	Unknown

(1) I = Input; O = Output; I/O = Bidirectional
(2) i = 1 to 3

Table 18-89. UART Interface Signals (continued)

Signal	I/O[1]	Description	Module Level Reset Value
		Active-low modem status signal. Reading the UARTi.UART_MSR[4] NCTS_STS bit checks the condition of uarti_ctsn. Reading the UARTi.UART_MSR[0] CTS_STS bit checks a change of state of uarti_ctsn since the last read of the modem status register. The auto-CTS mode uses uarti_ctsn to control the transmitter.	
uarti_rtsn	O	Request to send When active (low), the module is ready to receive data. Setting the UARTi.UART_MCR[1] RTS bit activates uarti_rtsn, which becomes inactive as the result of a module reset, loopback mode, or clearing the UARTi.UART_MCR[1] RTS bit. In auto-RTS mode, uarti_rtsn becomes inactive as a result of the receiver threshold logic.	1

NOTE: The path from a module pin to device pad(s) is defined at the device I/O logic level. The I/O logic maps the module signals to the different pads of the device and is programmable in the Control Module registers. For more information on control module settings, see Section 13.4.6.1, *Pad Configuration Registers*, in Chapter 13, *Control Module*.

18.2.2.1.3 UART Protocol and Data Format

The UART device operates in three modes:

- UART 16x (<= 230.4 kbps)
- UART 16x with autobauding (>= 1200 bps and >= 115.2 kbps)
- UART 13x (>= 460.8 kbps)

CAUTION

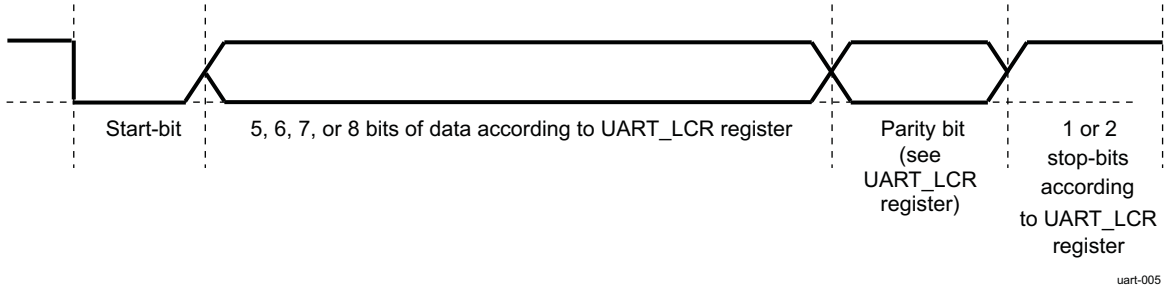
To be used as a UART, the operating mode must be programmed appropriately in the UARTi.UART_MDR1[2:0] MODE_SELECT bit field to select UART.

The UART uses a wired interface for serial communication with a remote device.

The UART is functionally compatible with the TL16C750 UART and earlier designs such as the TL16C550.

Figure 18-28 shows the UART frame data format.

Figure 18-28. UART Frame Data Format

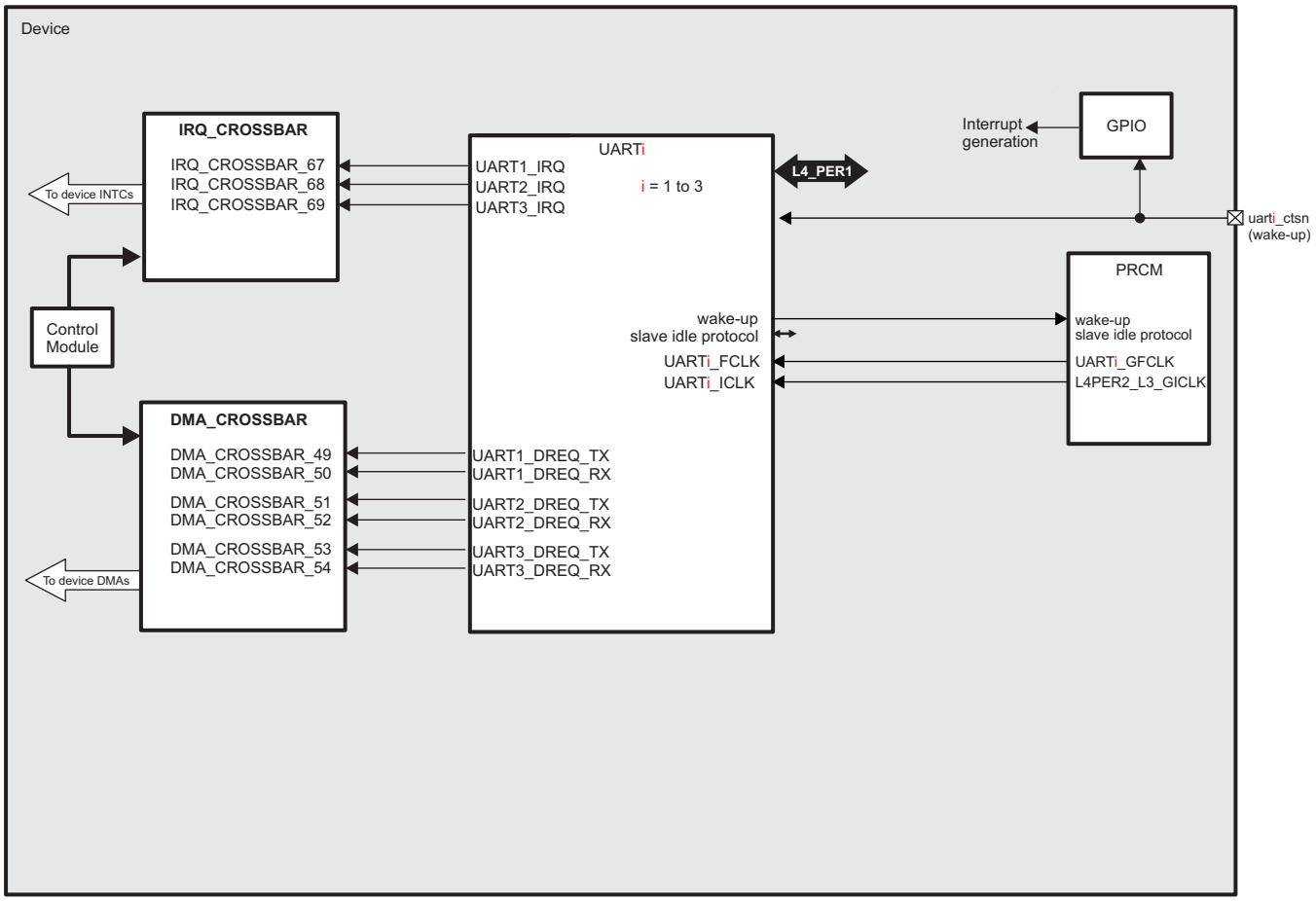


18.2.3 UART Integration

This section describes module integration in the device, including information about clocks, resets, and hardware requests.

Figure 18-29 shows the device internal connections with related modules for UART functions.

Figure 18-29. UART Integration



NOTE: For more information about the slave idle and master standby protocols and the wake-up request, see Section 3.1.1.1.4, *Clock Domain-Level Clock Management*, in Chapter 3, *Power, Reset, and Clock Management*

Table 18-90 through Table 18-92 summarize the integration of the module in the device.

Table 18-90. UART Integration Attributes

Module Instance	Attributes		
	Power Domain	Wake-Up Capability	Interconnect
UART1	PD_COREAON	Yes	L4_PER1
UART2	PD_COREAON	Yes	L4_PER1
UART3	PD_COREAON	Yes	L4_PER1

Table 18-91. UART Clocks and Resets

Clocks				
Module Instance	**Destination Signal**	**Source Signal**	**Source**	**Description**
UART1	UART1_ICLK	L4PER_L3_GICLK	PRCM	UART1 interface clock
	UART1_FCLK	UART1_GFCLK	PRCM	UART1 functional clock
UART2	UART2_ICLK	L4PER_L3_GICLK	PRCM	UART2 interface clock
	UART2_FCLK	UART2_GFCLK	PRCM	UART2 functional clock
UART3	UART3_ICLK	L4PER_L3_GICLK	PRCM	UART3 interface clock
	UART3_FCLK	UART3_GFCLK	PRCM	UART3 functional clock
Resets				
Module Instance	**Destination Signal**	**Source Signal**	**Source**	**Description**
UART1	UART1_RST	L4PER_RET_RST	PRCM	UART1 reset
UART2	UART2_RST	L4PER_RET_RST	PRCM	UART2 reset
UART3	UART3_RST	L4PER_RET_RST	PRCM	UART3 reset

Table 18-92. UART Hardware Requests

Interrupt Requests				
Module Instance	**Source Signal Name**	**Destination IRQ_CROSSBAR input**	**Default Mapping**	**Description**
UART1	UART1_IRQ	IRQ_CROSSBAR_67	-	UART module 1 IRQ line
UART2	UART2_IRQ	IRQ_CROSSBAR_68	-	UART module 2 IRQ line
UART3	UART3_IRQ	IRQ_CROSSBAR_69	IPU_IRQ_45	UART module 3 IRQ line
Direct Memory Access (DMA) Requests				
Module Instance	**Source Signal Name**	**Destination DMA_CROSSBAR input**	**Default Mapping**	**Description**
UART1	UART1_DREQ_TX	DMA_CROSSBAR_49	DMA_EDMA_DREQ_48	UART module 1 – transmit request
	UART1_DREQ_RX	DMA_CROSSBAR_50	DMA_EDMA_DREQ_49	UART module 1 – receive request
UART2	UART2_DREQ_TX	DMA_CROSSBAR_51	DMA_EDMA_DREQ_50	UART module 2 – transmit request
	UART2_DREQ_RX	DMA_CROSSBAR_52	DMA_EDMA_DREQ_51	UART module 2 – receive request
UART3	UART3_DREQ_TX	DMA_CROSSBAR_53	DMA_EDMA_DREQ_52	UART module 3 – transmit request
	UART3_DREQ_RX	DMA_CROSSBAR_54	DMA_EDMA_DREQ_53	UART module 3 – receive request

NOTE: The "**Default Mapping**" column in Table 18-92, *UART Hardware Requests*, shows the default mapping of module IRQ and DREQ source signals. These module IRQ and DREQ source signals can also be mapped to other lines of each device Interrupt or DMA controller through the IRQ_CROSSBAR and DMA_CROSSBAR modules, respectively.
For more information about the IRQ_CROSSBAR module, see Section 13.4.6.4, *IRQ_CROSSBAR Module Functional Description*, in Chapter 13, *Control Module*.
For more information about the DMA_CROSSBAR module, see Section 13.4.6.5, *DMA_CROSSBAR Module Functional Description*, in Chapter 13, *Control Module*.
For more information about the device interrupt controllers, see Chapter 12, *Interrupt Controllers*.
For more information about the device EDMA module, see Chapter 11, *Enhanced DMA*.

18.2.4 UART Functional Description

18.2.4.1 Block Diagram

The UART module can be divided into three main blocks:

- FIFO management
- Mode selection
- Protocol formatting

FIFO management is common to all functions and enables the transmission and reception of data from the host processor point of view.

There are two modes:

- Function mode: Routes the data to UART and enables UART function
- Register mode: Enables conditional access to registers

For more information about mode configuration, see Section 18.2.4.7, *Mode Selection*.

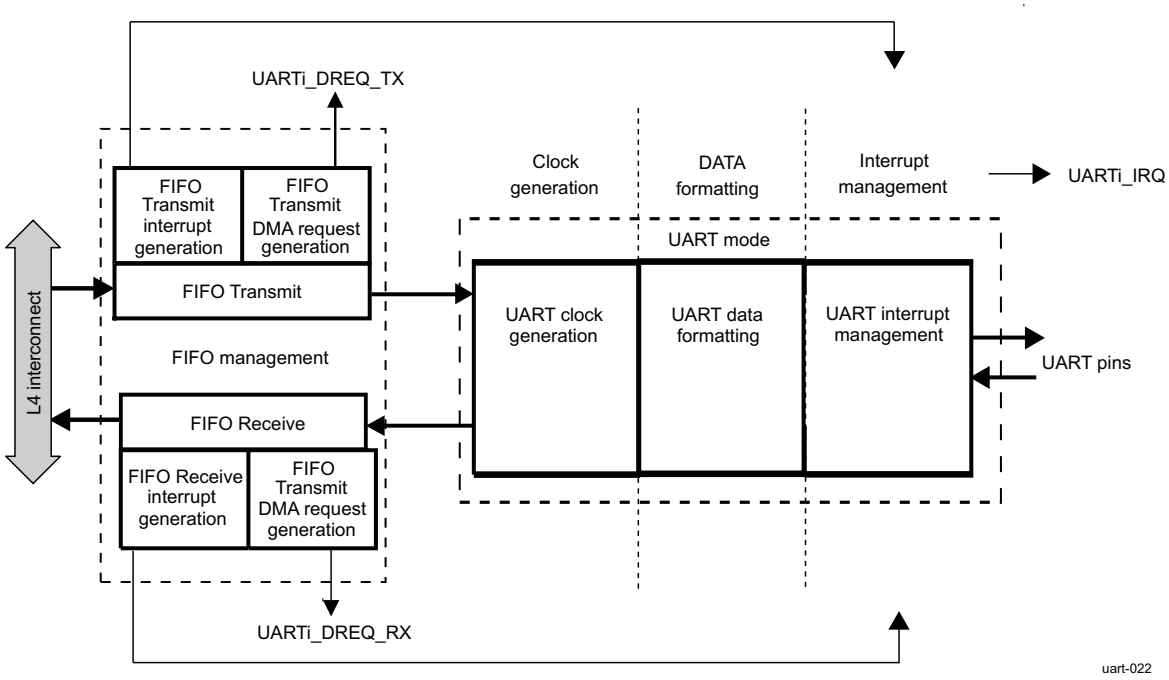
Protocol formatting has three subcategories:

- Clock generation: The 48-MHz input clock generates all necessary clocks.
- Data formatting: Each function uses its own state-machine that is responsible for the transition between FIFO data and frame data associated with it.
- Interrupt management: When an interrupt is generated, the UART_IIR register indicates the interrupt type.
  - Seven interrupts prioritized in six different levels

In parallel with these functional blocks, a power-saving strategy exists for each function.

Figure 18-30 is the UART block diagram.

Figure 18-30. UART Functional Block Diagram



18.2.4.2 Clock Configuration

Each UART uses a 48-MHz functional clock for its logic and to generate external interface signals. Each UART uses an interface clock for register accesses. The PRCM module generates and controls all these clocks (for more information, see Section 3.1.1.1.4, *Clock Domain-Level Clock Management*, in Chapter 3, *Power, Reset, and Clock Management*).

The idle and wake-up processes use a handshake protocol between the PRCM and the UART (for a description of the protocol, see Section 3.1.1.1.4, *Clock Domain Level Clock Management*, in Chapter 3, *Power, Reset, and Clock Management*). The UARTi.UART_SYSC[4:3] IDLEMODE bit field controls UART idle mode.

18.2.4.3 Software Reset

The UARTi.UART_SYSC[1] SOFTRESET bit controls the software reset; setting this bit to 1 triggers a software reset functionally equivalent to hardware reset.

18.2.4.4 Power Management

18.2.4.4.1 UART Mode Power Management

18.2.4.4.1.1 Module Power Saving

In UART modes, sleep mode is enabled by setting the UARTi.UART_IER[4] SLEEP_MODE bit to 1 (when the UARTi.UART_EFR[4] ENHANCED_EN bit is set to 1).

Sleep mode is entered when all of the following conditions exist:

- The serial data input line, uarti_rxd, is idle.
- The TX FIFO and TX shift register are empty.
- The RX FIFO is empty.
- The only pending interrupts are THR interrupts.

Sleep mode is a good way to lower UART power consumption, but this state can be achieved only when the UART is set to modem mode. Therefore, even if the UART has no key role functionally, it must be initialized in a functional mode to take advantage of sleep mode.

In sleep mode, the module clock and baud rate clock are stopped internally. Because most registers are clocked by these clocks, this greatly reduces power consumption. The module wakes up when a change is detected on the uarti_rxd line, when data is written to the TX FIFO, and when there is a change in the state of the modem input pins.

An interrupt can be generated on a wake-up event by setting the UARTi.UART_SCR[4] RX_CTS_WU_EN bit to 1. To understand how to manage the interrupt, see Section 18.2.4.5.1.2, *Wake-Up Interrupt*.

NOTE: There must be no writing to the divisor latches, UARTi.UART_DLL and UARTi.UART_DLH, to set the baud clock (BCLK) while in sleep mode. It is advisable to disable sleep mode using the UARTi.UART_IER[4] SLEEP_MODE bit before writing to the UARTi.UART_DLL or UARTi.UART_DLH register.

18.2.4.4.1.2 System Power Saving

Sleep and auto-idle modes are embedded power-saving features. Power-reduction techniques can be applied at the system level by shutting down certain internal clock and power domains of the device.

The UART supports an idle req/idle ack handshaking protocol used at the system level to shut down the UART clocks in a clean and controlled manner and to switch the UART from interrupt-generation mode to wake-up generation mode for unmasked events (see the UARTi.UART_SYSC[2] ENAWAKEUP bit and the UARTi.UART_WER register).

For more information, see Section 3.1.1.1.2, *Module Level Clock Management*, Chapter 3, *Power, Reset, and Clock Management*.

18.2.4.4.2 Local Power Management

Table 18-93 describes power-management features available for the UART.

NOTE: For information about source clock gating and the sleep/wake-up transitions description, see Section 3.1.1.1.2, *Module Level Clock Management*, Chapter 3, *Power, Reset, and Clock Management*.

Table 18-93. Local Power-Management Features

Feature	Registers	Description
Clock autogating	UART_SYSC[0] AUTOIDLE	This bit allows local power optimization in the module by gating the UARTi_ICLK clock on interface activity or gating the UARTi_FCLK clock on internal activity.
Slave idle modes	UART_SYSC[4:3] IDLEMODE	Force-idle, no-idle, smart-idle, and smart-idle wakeup-capable modes are available.
Clock activity	N/A	Feature not available
Master standby modes	N/A	Feature not available
Global wake-up enable	UART_SYSC[2] ENAWAKEUP	This bit enables the wake-up feature at module level.
Wake-Up sources enable	N/A	Feature not available

18.2.4.5 Interrupt Requests

18.2.4.5.1 UART Interrupt Management

18.2.4.5.1.1 UART Interrupts

UART includes seven possible interrupts prioritized to six levels.

When an interrupt is generated, the interrupt identification register (UARTi.UART_IIR) sets the UARTi.UART_IIR[0] IT_PENDING bit to 0 to indicate that an interrupt is pending, and indicates the type of interrupt through the UARTi.UART_IIR[5:1] bit field. Table 18-94 summarizes the interrupt control functions.

Table 18-94. UART Interrupts

IIR[5:0]	Priority Level	Interrupt Type	Interrupt Source	Interrupt Reset Method
000001	None	None	None	None
000110	1	Receiver line status	OE, FE, PE, or BI errors occur in characters in the RX FIFO.	FE, PE, BI: Read the UART_RHR register. OE: Read the UART_LSR register.
001100	2	RX time-out	Stale data in RX FIFO	Read the UART_RHR register.
000100	2	RHR interrupt	DRDY (data ready) (FIFO disable)	Read the UART_RHR register until the interrupt condition disappears.
			RX FIFO above trigger level (FIFO enable)	
000010	3	THR interrupt	TFE (THR empty) (FIFO disable)	Write to the UART_THR until the interrupt condition disappears.
			TX FIFO below trigger level (FIFO enable)	
000000	4	Modem status	See the UART_MSR register.	Read the MSR register.
010000	5	XOFF interrupt/special character interrupt	Receive XOFF characters/special character	Receive XON character(s), if XOFF interrupt/read of the UART_IIR register, if special character interrupt.
100000	6	CTS, RTS	RTS pin or CTS pin change state from active (low) to inactive (high).	Read the UART_IIR register.

For the receiver-line status interrupt, the RX_FIFO_STS bit (UARTi.UART_LSR[7]) generates the interrupt.

For the XOFF interrupt, if an XOFF flow character detection caused the interrupt, the interrupt is cleared by an XON flow character detection. If special character detection caused the interrupt, the interrupt is cleared by a read of the UARTi.UART_IIR register.

18.2.4.5.1.2 Wake-Up Interrupt

Wake-up interrupt is a special interrupt that works differently from other interrupts. This interrupt is enabled when the UARTi.UART_SCR[4] RX_CTS_WU_EN bit is set to 1. The UARTi.UART_IIR register is not modified when this occurs; the UART3.UART_SSR[1] RX_CTS_WU_STS bit must be checked to detect a wake-up event.

When a wake-up interrupt occurs, it can be cleared only by resetting the UARTi.UART_SCR[4] RX_CTS_WU_EN bit. This bit must be reenabled (set to 1) after the current wake-up interrupt event is processed to detect the next incoming wake-up event.

18.2.4.6 FIFO Management

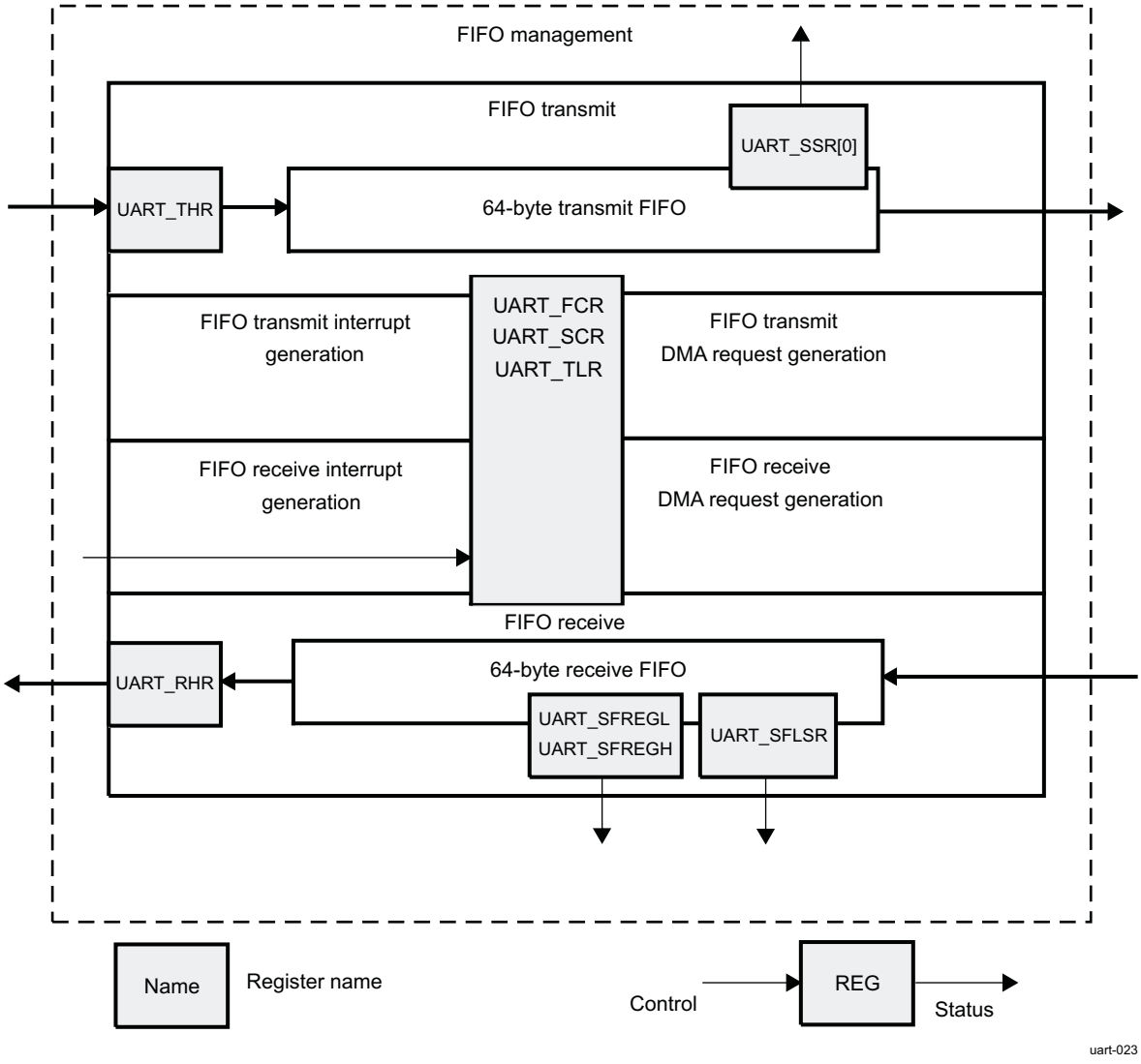
The FIFO is accessed by reading and writing the UARTi.UART_RHR and UARTi.UART_THR registers. Parameters are controlled using the FIFO control register (UARTi.UART_FCR) and supplementary control register (UARTi.UART_SCR). Reading the UARTi.UART_SSR[0] TX_FIFO_FULL bit at 1 means the FIFO is full.

The UARTi.UART_TLR register controls the FIFO trigger level, which enables DMA and interrupt generation. After reset, transmit (TX) and receive (RX) FIFOs are disabled; thus, the trigger level is the default value of 1 byte. Figure 18-31 shows the FIFO management registers.

NOTE: Data in the UARTi.UART_RHR register is not overwritten when an overflow occurs.

NOTE: Bits UARTI.UART_FCR[2] TX_FIFO_CLEAR and UARTI.UART_FCR[1] RX_FIFO_CLEAR are automatically cleared by hardware after 4* UARTi_ICLK + 5* UARTi_FCLK clock cycles. This delay is needed to finish the resetting of the corresponding FIFO and DMA control registers.

Figure 18-31. FIFO Management Registers



18.2.4.6.1 FIFO Trigger

18.2.4.6.1.1 Transmit FIFO Trigger

Table 18-95 lists the TX FIFO trigger level settings.

Table 18-95. TX FIFO Trigger Level Setting Summary

SCR[6]	TLR[3:0]	TX FIFO Trigger Level
0	= 0x0	Defined by the UARTi.UART_FCR[5:4] TX_FIFO_TRIG bit field (8,16, 32, or 56 spaces)
0	!= 0x0	Defined by the UARTi.UART_TLR[3:0] TX_FIFO_TRIG_DMA bit field (from 4 to 60 spaces with a granularity of 4 spaces)
1	Value	Defined by the concatenated value of TX_FIFO_TRIG_DMA and TX_FIFO_TRIG (from 1 to 63 spaces with a granularity of 1 space) **Note:** The combination of TX_FIFO_TRIG_DMA = 0x0 and TX_FIFO_TRIG = 0x0 (all zeros) is not supported (minimum of 1 space required). All zeros result in unpredictable behavior.

18.2.4.6.1.2 Receive FIFO Trigger

Table 18-96 lists the RX FIFO trigger-level settings.

Table 18-96. RX FIFO Trigger-Level Setting Summary

SCR[7]	TLR[7:4]	RX FIFO Trigger Level
0	= 0x0	Defined by the UARTi.UART_FCR[7:6] RX_FIFO_TRIG bit field (8,16, 56, or 60 characters)
0	!= 0x0	Defined by the UARTi.UART_TLR[7:4] RX_FIFO_TRIG_DMA bit field (from 4 to 60 characters with a granularity of 4 characters)
1	Value	Defined by the concatenated value of RX_FIFO_TRIG_DMA and RX_FIFO_TRIG (from 1 to 63 characters with a granularity of 1 character) **Note:** The combination of RX_FIFO_TRIG_DMA = 0x0 and RX_FIFO_TRIG = 0x0 (all zeros) is not supported (minimum of 1 character required). All zeros result in unpredictable behavior.

The receive threshold is programmed using the UARTi.UART_TCR[7:4] RX_FIFO_TRIG_START and UARTi.UART_TCR[3:0] RX_FIFO_TRIG_HALT bit fields:

- Trigger levels from 0 to 60 bytes are available with a granularity of 4 (trigger level = 4 × [4-bit register value]).
- To ensure correct device operation, ensure that RX_FIFO_TRIG_HALT > RX_FIFO_TRIG when auto-RTS is enabled.

  Delay = [4 + 16 × (1 + CHAR_LENGTH + Parity + Stop – 0.5)] × Baud_rate + 4 × FCLK

  NOTE: The RTS signal is deasserted after the UART module receives the data over RX_FIFO_TRIG_HALT. Delay means how long the UART module takes to deassert the RTS signal after reaching RX_FIFO_TRIG_HALT.

- In FIFO interrupt mode with flow control, ensure that the trigger level to HALT transmission is greater than or equal to the RX FIFO trigger level (the UARTi.UART_TCR[7:4] RX_FIFO_TRIG_START bit field or the UARTi.UART_FCR[7:6] RX_FIFO_TRIG bit field); otherwise, FIFO operation stalls. In FIFO DMA mode with flow control, this concept does not exist, because a DMA request is sent when a byte is received.

18.2.4.6.2 FIFO Interrupt Mode

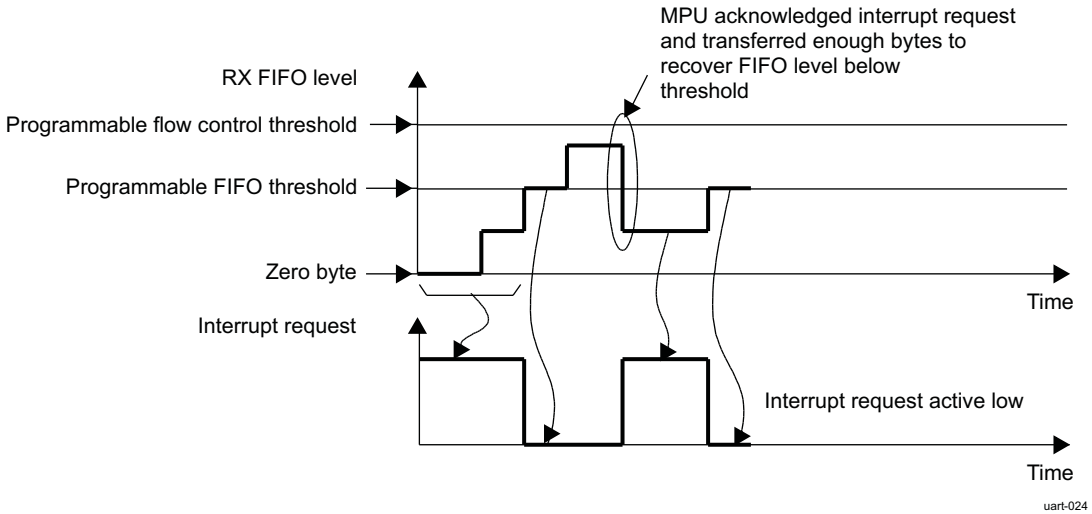
In FIFO interrupt mode (the FIFO control register UARTi.UART_FCR[0] FIFO_EN bit is set to 1 and relevant interrupts are enabled by the UARTi.UART_IER register), an interrupt signal informs the processor of the status of the receiver and transmitter. These interrupts are raised when the RX/TX FIFO threshold (the UARTi.UART_TLR[7:4] RX_FIFO_TRIG_DMA and UARTi.UART_TLR[3:0] TX_FIFO_TRIG_DMA bit fields or the UARTi.UART_FCR[7:6] RX_FIFO_TRIG and UARTi.UART_FCR[5:4] TX_FIFO_TRIG bit fields, respectively) is reached.

The interrupt signals instruct the CPU to transfer data to the destination (from the UART in receive mode and/or from any source to the UART FIFO in transmit mode).

When UART flow control is enabled with interrupt capabilities, the UART flow control FIFO threshold (the UARTi.UART_TCR[3:0] RX_FIFO_TRIG_HALT bit field) must be greater than or equal to the RX FIFO threshold.

Figure 18-32 shows the generation of the RX FIFO interrupt request.

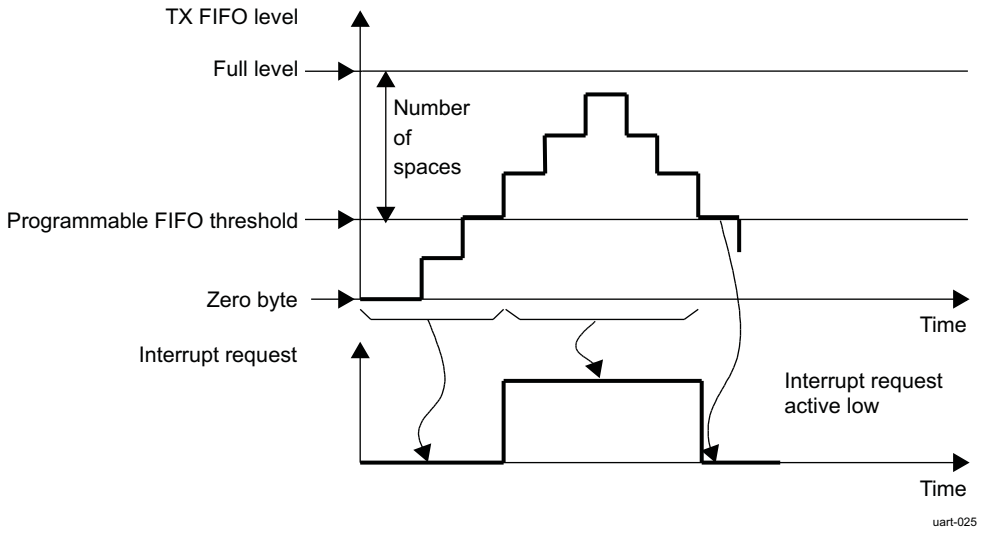
Figure 18-32. RX FIFO Interrupt Request Generation



In receive mode, no interrupt is generated until the RX FIFO reaches its threshold. Once low, the interrupt can be deasserted only when the CPU has handled enough bytes to put the FIFO level below threshold. The flow control threshold is set at a higher value than the FIFO threshold.

Figure 18-33 shows the generation of the TX FIFO interrupt request.

Figure 18-33. TX FIFO Interrupt Request Generation



In transmit mode, an interrupt request is automatically asserted when the TX FIFO is empty. This request is deasserted when the TX FIFO crosses the threshold level. The interrupt line is deasserted until a sufficient number of elements is transmitted to go below the TX FIFO threshold.

18.2.4.6.3 FIFO Polled Mode Operation

In FIFO polled mode (the UARTi.UART_FCR[0] FIFO_EN bit is set to 0 and the relevant interrupts are disabled by the UARTi.UART_IER register), the status of the receiver and transmitter can be checked by polling the line status register (UARTi.UART_LSR).

This mode is an alternative to the FIFO interrupt mode of operation in which the status of the receiver and transmitter is automatically determined by sending interrupts to the CPU.

18.2.4.6.4 FIFO DMA Mode Operation

Although the DMA operation includes four modes (DMA modes 0 through 3), the information in Table 18-92, *Hardware Requests*, assumes that mode 1 is used. (Mode 2 and mode 3 are legacy modes that use only one DMA request for each module.)

In mode 2, the remaining DMA request is used for RX. In mode 3, the remaining DMA request is used for TX.

DMA requests in mode 2 and mode 3 use the UARTi_DREQ_TX signals (where i = 1 to 10).

The UARTi_DREQ_RX signals are not used by the module in mode 2 and mode 3:

The DMA mode and signals usage can be selected as follows:

- When the UARTi.UART_SCR[0] DMA_MODE_CTL bit is set to 0, setting the UARTi.UART_FCR[3] DMA_MODE bit to 0 enables DMA mode 0. Setting the UARTi.UART_FCR[3]DMA_MODE bit to 1 enables DMA mode 1.
- When the UARTi.UART_SCR[0]DMA_MODE_CTL bit is set to 1, the UARTi.UART_SCR[2:1] DMA_MODE_2 bit field determines DMA mode 0 to mode 3 based on the supplementary control register (SCR) description.

For example:

- If no DMA operation is desired, set theUARTi.UART_SCR[0]DMA_MODE_CTL bit to 1 and the UARTi.UART_SCR[2:1]DMA_MODE_2 bit field to 0x0. (The DMA_MODE bit is discarded.)
- If DMA mode 1 is desired, set the UARTi.UART_SCR[0]DMA_MODE_CTL bit to 0 and the UARTi.UART_FCR[3]DMA_MODE bit to 1, or set the UARTi.UART_SCR[0]DMA_MODE_CTL bit to 1 and the SCR[2:1]DMA_MODE_2 bit field to 01. (The UARTi.UART_FCR[3]DMA_MODE bit is discarded.)

If the FIFOs are disabled (the UARTi.UART_FCR[0] FIFO_EN bit is set to 0), the DMA occurs in single-character transfers.

When DMA mode 0 is programmed, the signals associated with DMA operation are not active.

Depending on UARTi.UART_MDR3[2] SET_DMA_TX_THRESHOLD, the threshold can be programmed different ways:

- SET_TX_DMA_THRESHOLD = 1:

  The threshold value will be the value of the TX_DMA_THRESHOLD register. If SET_TX_DMA_THRESHOLD + TX trigger spaces 64, then the default method of threshold is used: threshold value = TX FIFO size.
- SET_TX_DMA_THRESHOLD = 0:

  The threshold value = TX FIFO size TX trigger space. The TX DMA line is asserted if the TX FIFO level is lower then the threshold. It remains asserted until TX trigger spaces number of bytes are written into the FIFO. The DMA line is then deasserted and the FIFO level is compared with the threshold value.

18.2.4.6.4.1 DMA sequence to disable TX DMA

In order to disable TX DMA if it is not needed anymore (e.g. all transfers are done and UART idle mode is desired), the following sequence must be use

1. DMA mode 1 is set (both TX/RX DMA) by registers UARTI.UART_SCR[0] = 0 and UARTI.UART_FCR[3] = 1:
   A. Set the UARTI.UART_SCR[2:1]DMA_MODE_2 bit fields to 01 (DMA mode 1)
   B. Set the UARTI.UART_SCR[0]DMA_MODE_CTL bit to1 (this setting of UARTI.UART_SCR[0]DMA_MODE-CTL will ignores UARTI.UART_FCR[3]DMA_MODE_CTL bit)

      NOTE: It is strongly suggested to do steps ‘a’ and ‘b’ in two separate write in order to avoid malfunction of the device.

   C. Set the UARTI.UART_FCR[3]DMA_MODE bit to 0. It is not necessary but suggested to avoid restore of DMA mode 1 during accidental reset of UARTI.UART_SCR[0]DMA_MODE_CTL bit. Be sure that all data was read out from RX FIFO and if it possible disable the RX side. In UART mode the RTS/CTS or XOFF/XON protocol can be used.

      NOTE: There can be RX DATA loss during the next steps if all DATA was not read out or there was an ongoing reception!

   D. Set the UARTI.UART_FCR[2:1]DMA_MODE bit field to 11 (clear TX and RX FIFO and resets its counter logic to 0. Returns to 0 after clearing FIFO).
   E. Set the UARTI.UART_SCR[2:1]DMA_MODE_2 bit field to 10 (DMA mode 2, RX only).
   F. Set the UARTI.UART_FCR[2:1]DMA_MODE bit field to 11 (clear TX and RX FIFO and the DMA request again).
   G. Set the UARTI.UART_SCR[2:1] DMA_MODE_2 bit field to 00 (no DMA) or keep 10 if RX DMA is needed.
2. DMA mode 1 is set (both TX/RX DMA) by registers UARTI.UART_FCR[3] = 0 and UARTI.UART_SCR[0] = 1, UARTI.UART_SCR[2:1] = 01. It is almost the same as above, but steps ‘a’, and ‘b’ can be skipped:
   A. Set the UARTI.UART_FCR[3]DMA_MODE bit to 0. It is not necessary but suggested to avoid restore of DMA mode 1 during accidental reset of UARTI.UART_SCR[0]DMA_MODE_CTL bit. Be sure that all data was read out from RX FIFO and if it possible disable the RX side. In UART mode the RTS/CTS or XOFF/XON protocol can be used.

      NOTE: There can be RX DATA loss during the next steps if all DATA was not read out or there was an ongoing reception!

   B. Set the UARTI.UART_FCR[2:1]DMA_MODE bit field to 11 (clear TX and RX FIFO and resets its

counter logic to 0. Returns to 0 after clearing FIFO).

C. Set the UARTI.UART_SCR[2:1]DMA_MODE_2 bit field to 10 (DMA mode 2, RX only).

D. Set the UARTI.UART_FCR[2:1]DMA_MODE bit field to 11 (clear TX and RX FIFO and the DMA request again).

E. Set the UARTI.UART_SCR[2:1] DMA_MODE_2 bit field to 00 (no DMA) or keep 10 if RX DMA is needed.

3. DMA mode 3 is set (TX DMA only) by registers UARTI.UART_FCR[3] = 0 and UARTI.UART_SCR[0] = 1, UARTI.UART_SCR[2:1] = 11. It is the same as above:

A. Set the UARTI.UART_FCR[3]DMA_MODE bit to 0. It is not necessary but suggested to avoid restore of DMA mode 1 during accidental reset of UARTI.UART_SCR[0]DMA_MODE_CTL bit. Be sure that all data was read out from RX FIFO and if it possible disable the RX side. In UART mode the RTS/CTS or XOFF/XON protocol can be used.

NOTE: There can be RX DATA loss during the next steps if all DATA was not read out or there was an ongoing reception!

B. Set the UARTI.UART_FCR[2:1]DMA_MODE bit field to 11 (clear TX and RX FIFO and resets its counter logic to 0. Returns to 0 after clearing FIFO).

C. Set the UARTI.UART_SCR[2:1]DMA_MODE_2 bit field to 10 (DMA mode 2, RX only).

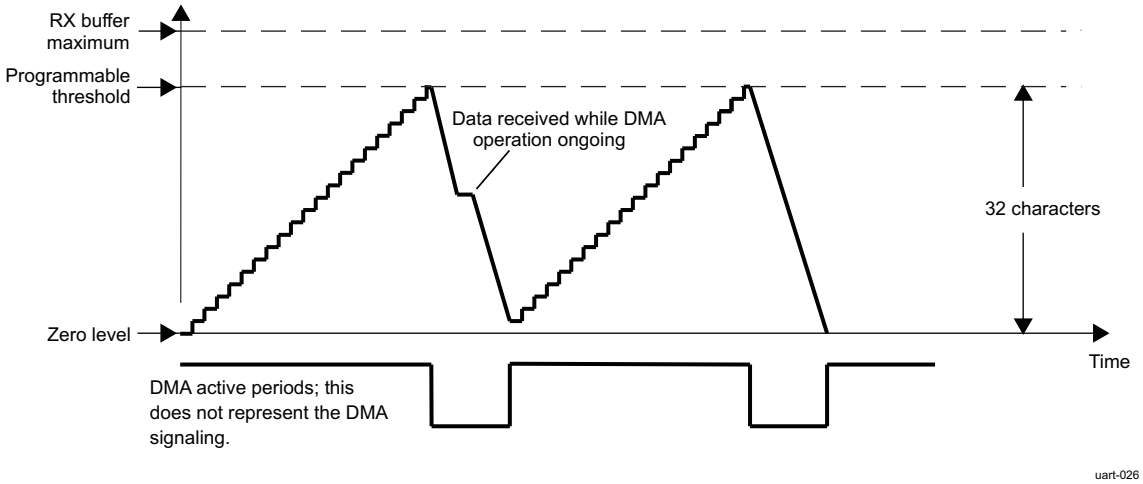
D. Set the UARTI.UART_FCR[2:1]DMA_MODE bit field to 11 (clear TX and RX FIFO and the DMA request again).

E. Set the UARTI.UART_SCR[2:1] DMA_MODE_2 bit field to 00 (no DMA) or keep 10 if RX DMA is needed.

18.2.4.6.4.2 DMA Transfers (DMA Mode 1, 2, or 3)

Figure 18-34 through Figure 18-37 show the supported DMA operations.

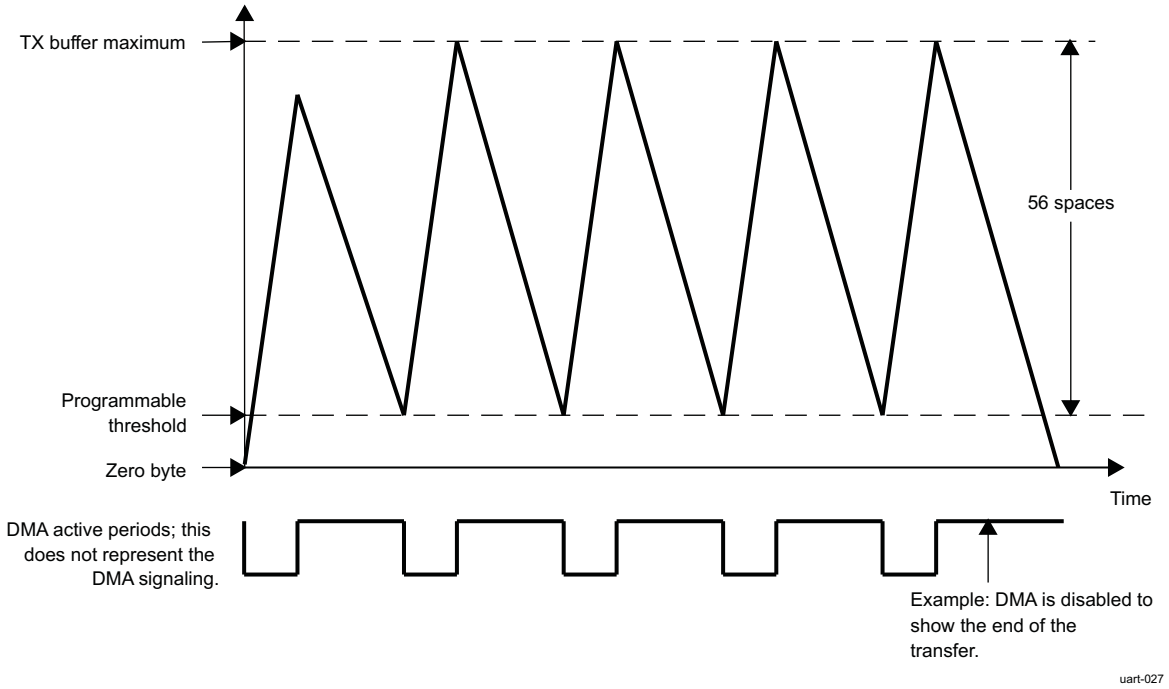
Figure 18-34. Receive FIFO DMA Request Generation (32 Characters)



In receive mode, a DMA request is generated when the RX FIFO reaches its threshold level defined in the trigger level register (UARTi.UART_TLR). This request is deasserted when the number of bytes defined by the threshold level is read by the device DMA controllers.

In transmit mode, a DMA request is automatically asserted when the TX FIFO is empty. This request is deasserted when the number of bytes defined by the number of spaces in the UARTi.UART_TLR register is written by the device DMA controllers. If an insufficient number of characters is written, the DMA request stays active.

Figure 18-35. Transmit FIFO DMA Request Generation (56 Spaces)



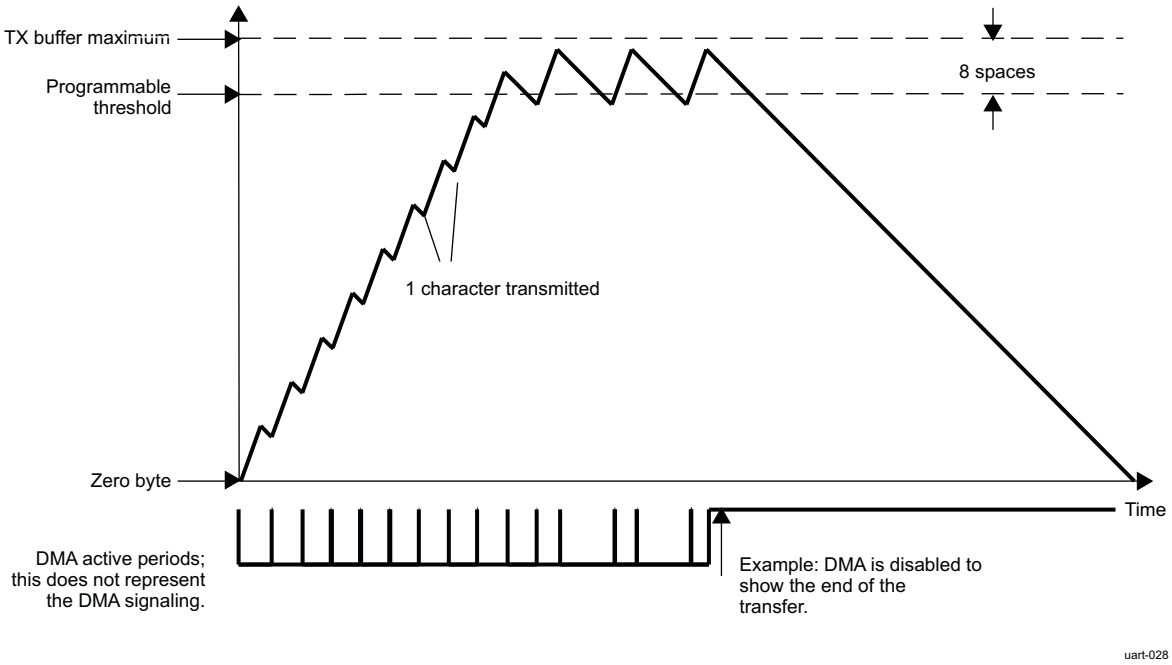
The DMA request is again asserted if the FIFO can receive the number of bytes defined by the UARTi.UART_TLR register.

The threshold can be programmed in a number of ways. Figure 18-35 shows a DMA transfer operating with a space setting of 56 that can arise from using the auto settings in the UARTi.UART_FCR[5:4] TX_FIFO_TRIG bit field or the UARTi.UART_TLR[3:0] TX_FIFO_TRIG_DMA bit field concatenated with the TX_FIFO_TRIG bit field.

The setting of 56 spaces in the UART module must correlate with the settings of the device DMA controllers, so that the buffer does not overflow (program the DMA request size of the LH controller to equal the number of spaces in the UART module).

Figure 18-36 shows an example with eight spaces to show the buffer level crossing the space threshold. The LH DMA controller settings must correspond to those of the UART module.

Figure 18-36. Transmit FIFO DMA Request Generation (8 Spaces)

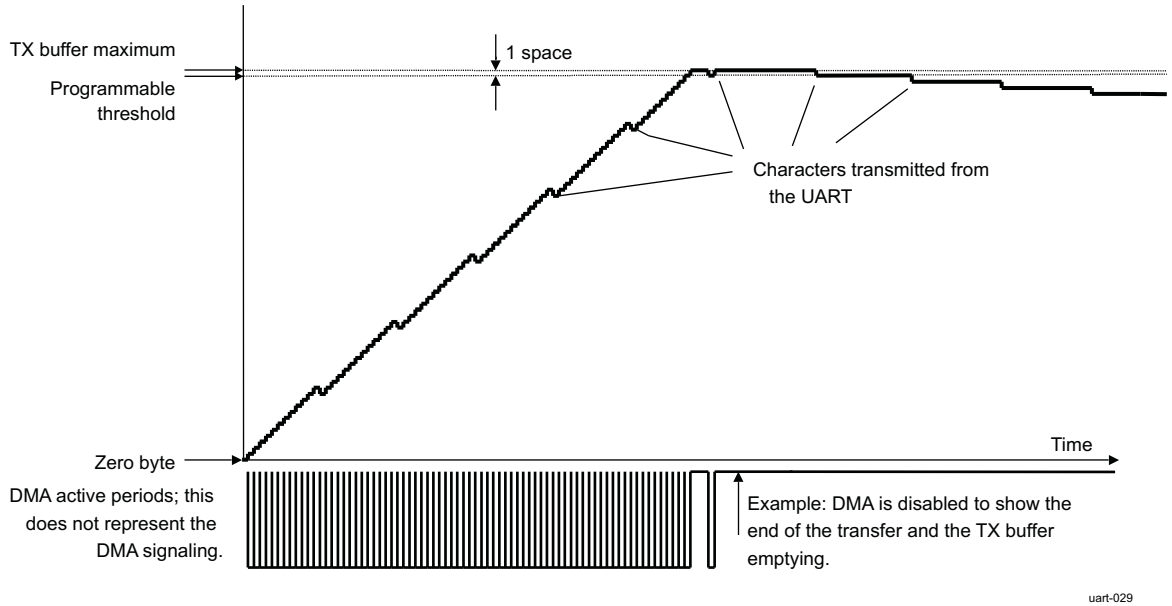


The next example shows the setting of one space that uses the DMA for each transfer of one character to the transmit buffer (see Figure 18-37). The buffer is filled faster than the baud rate at which data is transmitted to the TX pin. Eventually, the buffer is completely full and the DMA operations stop transferring data to the transmit buffer.

On two occasions, the buffer holds the maximum amount of data words; shortly after this, the DMA is disabled to show the slower transmission of the data words to the TX pin. Eventually, the buffer is emptied at the rate specified by the baud rate settings of the UARTi.UART_DLL and UARTi.UART_DLH registers.

The DMA settings must correspond to the system LH DMA controller settings to ensure correct operation of this logic.

Figure 18-37. Transmit FIFO DMA Request Generation (1 Space)



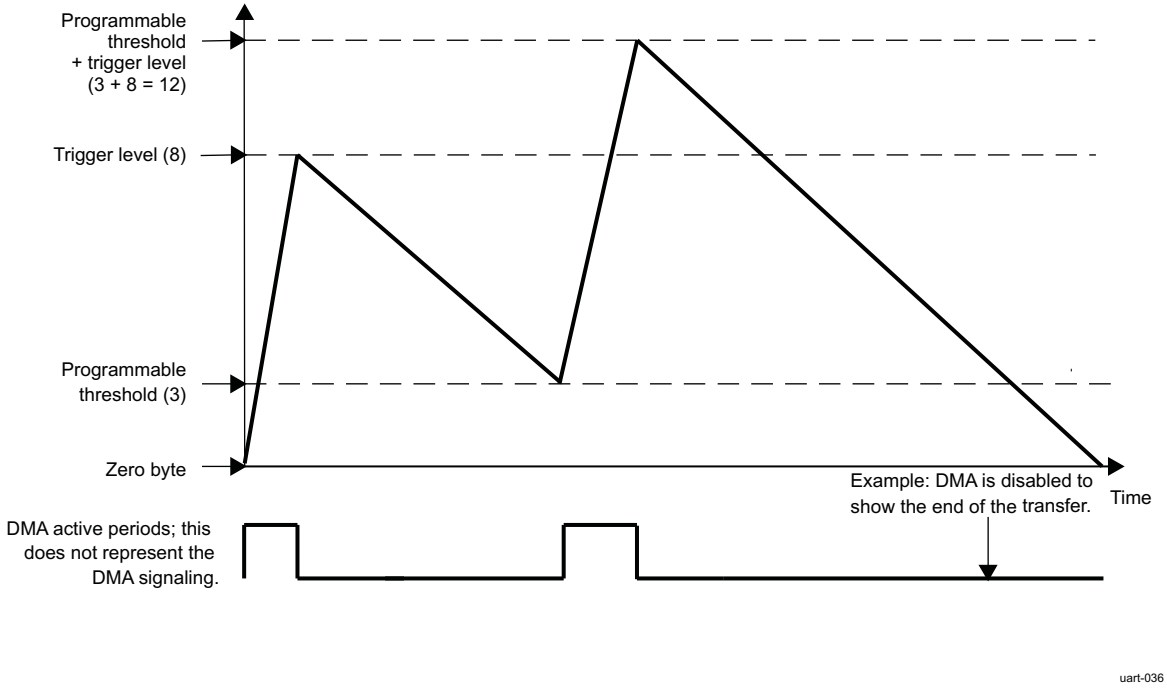
The final example illustrates the setting of eight spaces but setting the TX DMA threshold directly by setting UART_MDR3[1] NONDEFAULT_FREQ bit and UART_TX_DMA_THRESHOLD register (see Figure 18-38). In the example, UART_TX_DMA_THRESHOLD[5:0] TX_DMA_THRESHOLD = 3 and the trigger level is 8. The buffer is filled at a faster rate than the BAUD rate transmits data to the TX pin. The buffer is filled with 8 bytes and the DMA operations stop transferring data to the transmit buffer. When the buffer is emptied to the threshold level by transmission, the DMA operation activates again to fill the buffer with 8 bytes.

Eventually, the buffer will be emptied at the rate specified by the BAUD Rate settings of the UART_DLL and UART_DLH registers.

If the selected threshold level + trigger level exceeds max buffer size, then the original TX DMA threshold method is used to prevent TX overrun, regardless of the UART_MDR3[1] NONDEFAULT_FREQ value.

The DMA settings should correspond to the system Local Host DMA controller settings in order to ensure the correct operation of this logic.

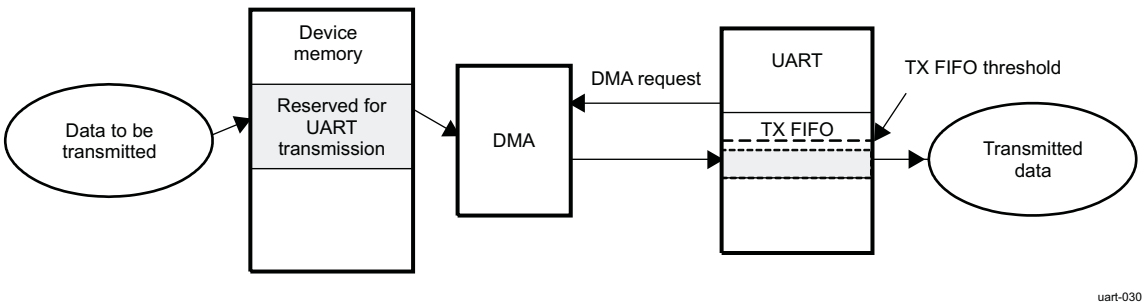
Figure 18-38. Transmit FIFO DMA Request Generation Using Direct TX DMA Threshold Programming. (Threshold = 3; Spaces = 8)



18.2.4.6.4.3 DMA Transmission

Figure 18-39 shows DMA transmission.

Figure 18-39. DMA Transmission



1. Data to be transmitted are put in the device memory reserved for UART transmission by the DMA:

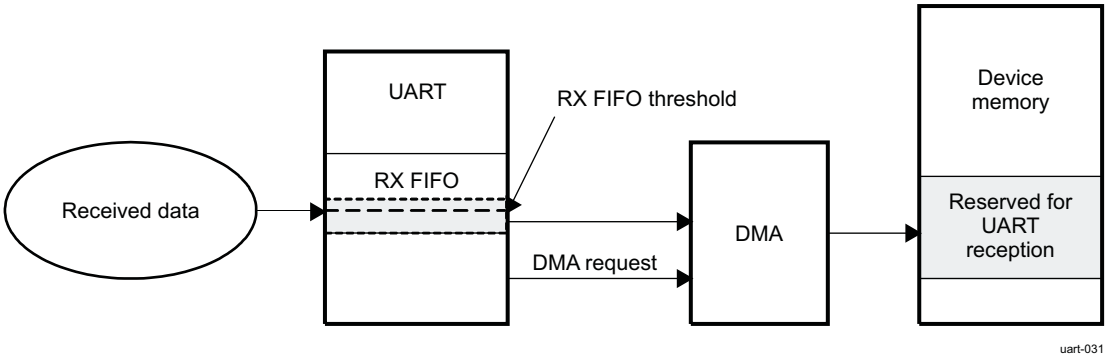
a. Until the TX FIFO trigger level is not reached, a DMA request is generated
b. An element (1 byte) is transferred from the SDRAM to the TX FIFO at each DMA request (DMA element synchronization).

2. Data in the TX FIFO are automatically transmitted.
3. The end of the transmission is signaled by the UARTi.UART_THR empty (TX FIFO empty).

18.2.4.6.4.4 DMA Reception

Figure 18-40 shows DMA reception.

Figure 18-40. DMA Reception



1. Enable the reception.
2. Received data are put in the RX FIFO.
3. Data are transferred from the RX FIFO to the device memory by the DMA:
   a. At each received byte, the RX FIFO trigger level (one character) is reached and a DMA request is generated.
   b. An element (1 byte) is transferred from the RX FIFO to the SDRAM at each DMA request (DMA element synchronization).
4. The end of the reception is signaled by the EOF interrupt.

18.2.4.7 Mode Selection

18.2.4.7.1 Register Access Modes

18.2.4.7.1.1 Operational Mode and Configuration Modes

Register access depends on the register access mode, although register access modes are not correlated to functional mode selection. Three different modes are available:

- Operational mode
- Configuration mode A
- Configuration mode B

Operational mode is the selected mode when the function is active; serial data transfer can be performed in this mode.

Configuration mode A and configuration mode B are used during module initialization steps. These modes enable access to configuration registers, which are hidden in the operational mode. The modes are used when the module is inactive (no serial data transfer processed) and only for initialization or reconfiguration of the module.

The value of the UARTi.UART_LCR register determines the register access mode (see Table 18-97).

Table 18-97. UART Register Access Mode Programming (Using UART_LCR)

Mode	Condition
Configuration mode A	UART_LCR[7] = 0x1 and UART_LCR[7:0] != 0xBF
Configuration mode B	UART_LCR[7] = 0x1 and UART_LCR[7:0] = 0xBF
Operational mode	UART_LCR[7] = 0x0

18.2.4.7.1.2 Register Access Submode

In each access register mode (operational mode or configuration mode A/B), some register accesses are conditional on the programming of a submode (MSR_SPR, TCR_TLR, and XOFF). These registers are identified in Table 18-116, *UART Register Manual*.

Table 18-98 through Table 18-100 summarize the register access submodes.

Table 18-98. Subconfiguration Mode A Summary

Mode	Condition
MSR_SPR	(UART_EFR[4] = 0x0 or UART_MCR[6] = 0x0)
TCR_TLR	UART_EFR[4] = 0x1 and UART_MCR[6] = 0x1

Table 18-99. Subconfiguration Mode B Summary

Mode	Condition
TCR_TLR	UART_EFR[4] = 0x1 and UART_MCR[6] = 0x1
XOFF	(UART_EFR[4] = 0x0 or UART_MCR[6] = 0x0)

Table 18-100. Suboperational Mode Summary

Mode	Condition
MSR_SPR	UART_EFR[4] = 0x0 or UART_MCR[6] = 0x0
TCR_TLR	UART_EFR[4] = 0x1 and UART_MCR[6] = 0x1

18.2.4.7.2 UART Mode Selection

To select a mode, set the UARTi.UART_MDR1[2:0] MODE_SELECT bit field (see Table 18-101).

Table 18-101. UART Mode Selection

Value	Mode
0x0:	UART 16x mode
0x1:	Reserved
0x2:	UART 16x auto-baud
0x3:	UART 13x mode
0x4:	Reserved
0x5:	Reserved
0x6:	Reserved
0x7:	Disable (default state)

MODE_SELECT is effective when the module is in operational mode (see Section 18.2.4.7.1, *Register Access Modes*).

18.2.4.7.2.1 Registers Available for the UART Function

Only the registers listed in Table 18-102 are used for the UART function.

Table 18-102. UART Mode Register Overview

Address Offset	Registers					
	Configuration Mode A		Configuration Mode B		Operational Mode	
	Read	Write	Read	Write	Read	Write
0x00	UART_DLL	UART_DLL	UART_DLL	UART_DLL	UART_RHR	UART_THR
0x04	UART_DLH	UART_DLH	UART_DLH	UART_DLH	UART_IER	UART_IER
0x08	UART_IIR	UART_FCR	UART_EFR	UART_EFR	UART_IIR	UART_FCR
0x0C	UART_LCR	UART_LCR	UART_LCR	UART_LCR	UART_LCR	UART_LCR
0x10	UART_MCR	UART_MCR	UART_XON1_ADDR1	UART_XON1_ADDR1	UART_MCR	UART_MCR
0x14	UART_LSR	–	UART_XON2_ADDR2	UART_XON2_ADDR2	UART_LSR	–
0x18	UART_MSR UART_TCR	UART_TCR	UART_XOFF1 UART_TCR	UART_XOFF1 UART_TCR	UART_MSR UART_TCR	UART_TCR
0x1C	UART_TLR UART_SPR	UART_TLR UART_SPR	UART_TLR UART_XOFF2	UART_TLR UART_XOFF2	UART_TLR UART_SPR	UART_TLR UART_SPR
0x20	UART_MDR1	UART_MDR1	UART_MDR1	UART_MDR1	UART_MDR1	UART_MDR1
0x24	–	–	–	–	–	–
0x28	–	–	–	–	–	–
0x2C	–	–	–	–	–	–
0x30	–	–	–	–	–	–
0x34	–	–	–	–	–	–
0x38	UART_UASR	–	UART_UASR	–	–	–
0x3C	–	–	–	–	–	–
0x40	UART_SCR	UART_SCR	UART_SCR	UART_SCR	UART_SCR	UART_SCR
0x44	UART_SSR	–	UART_SSR	–	UART_SSR	–
0x48	–	–	–	–	–	–
0x50	UART_MVR	–	UART_MVR	–	UART_MVR	–
0x54	UART_SYSC	UART_SYSC	UART_SYSC	UART_SYSC	UART_SYSC	UART_SYSC
0x58	UART_SYSS	–	UART_SYSS	–	UART_SYSS	–
0x5C	UART_WER	UART_WER	UART_WER	UART_WER	UART_WER	UART_WER
0x60	–	–	–	–	–	–
0x64	UART_RXFIFO_LVL	UART_RXFIFO_LVL	UART_RXFIFO_LVL	UART_RXFIFO_LVL	UART_RXFIFO_LVL	UART_RXFIFO_LVL
0x68	UART_TXFIFO_LVL	UART_TXFIFO_LVL	UART_TXFIFO_LVL	UART_TXFIFO_LVL	UART_TXFIFO_LVL	UART_TXFIFO_LVL
0x6C	UART_IER2	UART_IER2	UART_IER2	UART_IER2	UART_IER2	UART_IER2
0x70	UART_ISR2	UART_ISR2	UART_ISR2	UART_ISR2	UART_ISR2	UART_ISR2
0x74	UART_FREQ_SEL	UART_FREQ_SEL	UART_FREQ_SEL	UART_FREQ_SEL	UART_FREQ_SEL	UART_FREQ_SEL
0x80	UART_MDR3	UART_MDR3	UART_MDR3	UART_MDR3	UART_MDR3	UART_MDR3
0x84	UART_TX_DMA_THRESHOLD	UART_TX_DMA_THRESHOLD	UART_TX_DMA_THRESHOLD	UART_TX_DMA_THRESHOLD	UART_TX_DMA_THRESHOLD	UART_TX_DMA_THRESHOLD

18.2.4.8 Protocol Formatting

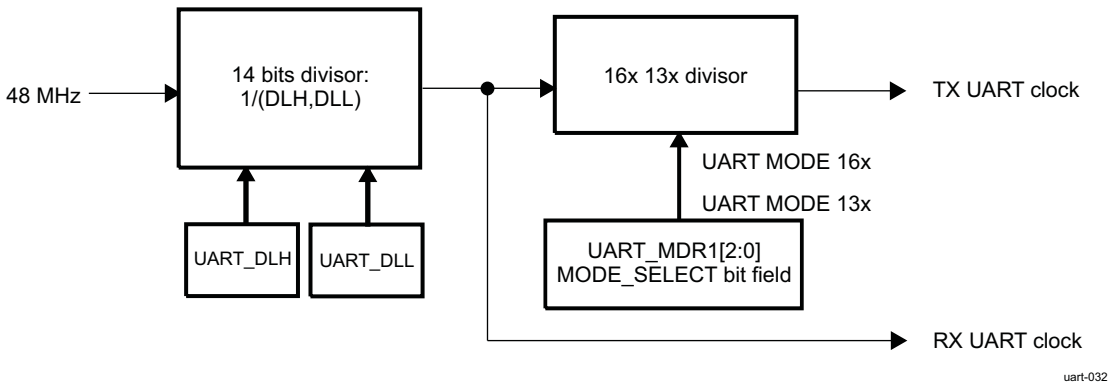
18.2.4.8.1 UART Mode

18.2.4.8.1.1 UART Clock Generation: Baud Rate Generation

The UART function contains a programmable baud generator and a set of fixed dividers that divide the 48-MHz clock input down to the expected baud rate.

Figure 18-41 shows the baud rate generator and associated controls.

Figure 18-41. Baud Rate Generation



CAUTION

Before initializing or modifying clock parameter controls (UARTi.UART_DLH, UARTi.UART_DLL), MODE_SELECT = DISABLE (UARTi.UART_MDR1[2:0]) must be set to 0x7. Failure to observe this rule can result in unpredictable module behavior.

18.2.4.8.1.2 Choosing the Appropriate Divisor Value

Two divisor values are:

- UART 16x mode: Divisor value = Operating frequency / (16x baud rate)
- UART 13x mode: Divisor value = Operating frequency / (13x baud rate)

Table 18-103 and Table 18-104 describe the UART baud rate settings.

Table 18-103. UART Baud Rate Settings (48-MHz Clock)

Baud Rate	Baud Multiple	DLH, DLL (Decimal)	DLH, DLL (Hex)	Actual Baud Rate	Error (%)
0.3 kbps	16x	10000	0x27, 0x10	0.3 kbps	0
0.6 kbps	16x	5000	0x13, 0x88	0.6 kbps	0
1.2 kbps	16x	2500	0x09, 0xC4	1.2 kbps	0
2.4 kbps	16x	1250	0x04, 0xE2	2.4 kbps	0
4.8 kbps	16x	625	0x02, 0x71	4.8 kbps	0
9.6 kbps	16x	312	0x01, 0x38	9.6153 kbps	+0.16
14.4 kbps	16x	208	0x00, 0xD0	14.423 kbps	+0.16
19.2 kbps	16x	156	0x00, 0x9C	19.231 kbps	+0.16
28.8 kbps	16x	104	0x00, 0x68	28.846 kbps	+0.16
38.4 kbps	16x	78	0x00, 0x4E	38.462 kbps	+0.16
57.6 kbps	16x	52	0x00, 0x34	57.692 kbps	+0.16
115.2 kbps	16x	26	0x00, 0x1A	115.38 kbps	+0.16
230.4 kbps	16x	13	0x00, 0x0D	230.77 kbps	+0.16
460.8 kbps	13x	8	0x00, 0x08	461.54 kbps	+0.16
921.6 kbps	13x	4	0x00, 0x04	923.08 kbps	+0.16
1.843 Mbps	13x	2	0x00, 0x02	1.846 Mbps	+0.16
3.0 Mbps	16x	1	0x00, 0x01	3.0 Mbps	0
3.6884 Mbps	13x	1	0x00, 0x01	3.6923 Mbps	+0.16

Table 18-104. UART Baud Rate Settings (192-MHz Clock)

Baud Rate	Baud Multiple	DLH, DLL (Decimal)	DLH, DLL (Hex)	Actual Baud Rate	Error (%)
1.2 kbps	16x	10000	0x27, 0x10	1.2 kbps	0
2.4 kbps	16x	5000	0x13, 0x88	2.4 kbps	0
4.8 kbps	16x	2500	0x09, 0xC4	4.8 kbps	0
9.6 kbps	16x	1250	0x04, 0xE2	9.6 kbps	0
14.4 kbps	16x	834	0x03, 0x42	14.388 kbps	-0.08
19.2 kbps	16x	625	0x02, 0x71	19.2 kbps	0
28.8 kbps	16x	417	0x01, 0xA1	28.777 kbps	-0.08
38.4 kbps	16x	312	0x01, 0x38	38.462 kbps	+0.16
57.6 kbps	16x	208	0x00, 0xD0	57.692 kbps	+0.16
115.2 kbps	16x	104	0x00, 0x68	115.385 kbps	+0.16
230.4 kbps	16x	52	0x00, 0x34	230.769 kbps	+0.16
460.8 kbps	16x	26	0x00, 0x1A	461.538 kbps	+0.16
921.6 kbps	16x	13	0x00, 0x0D	923.077 kbps	+0.16
1.8432 Mbps	13x	8	0x00, 0x08	1.846154 Mbps	+0.16
3.0 Mbps	16x	4	0x00, 0x04	3.0 Mbps	0
3.6864 Mbps	13x	4	0x00, 0x04	3.692308 Mbps	+0.16
7.3728 Mbps	13x	2	0x00, 0x02	7.384615 Mbps	+0.16
12.0 Mbps	16x	1	0x00, 0x01	12.0 Mbps	0

18.2.4.8.1.3 UART Data Formatting

The UART can use hardware flow control to manage transmission and reception. Hardware flow control significantly reduces software overhead and increases system efficiency by automatically controlling serial data flow using the RTS output and CTS input signals.

The UART is enhanced with the autobauding function. In control mode, autobauding lets the speed, the number of bits per character, and the parity selected be set automatically.

18.2.4.8.1.3.1 Frame Formatting

When autobauding is not used, frame format attributes must be defined in the UARTi.UART_LCR register.

Character length is specified using the UARTi.UART_LCR[1:0] CHAR_LENGTH bit field.

The number of stop-bits is specified using the UARTi.UART_LCR[2] NB_STOP bit.

The parity bit is programmed using the UARTi.UART_LCR[5:3] PARITY_EN, PARITY_TYPE_1, and PARITY_TYPE_2 bit fields (see Table 18-105).

Table 18-105. UART Parity Bit Encoding

PARITY_EN	PARITY_TYPE_1	PARITY_TYPE_2	Parity
0	N/A	N/A	No parity
1	0	0	Odd parity
1	1	0	Even parity
1	0	1	Forced 1
1	1	1	Forced 0

18.2.4.8.1.3.2 Hardware Flow Control

Hardware flow control is composed of auto-CTS and auto-RTS. Auto-CTS and auto-RTS can be enabled and disabled independently by programming the UARTi.UART_EFR[7:6] AUTO_CTS_EN and AUTO_RTS_EN bit fields, respectively.

With auto-CTS, uarti_ctsn must be active before the module can transmit data.

Auto-RTS activates the uarti_rtsn output only when there is enough room in the RX FIFO to receive data. It deactivates the uarti_rtsn output when the RX FIFO is sufficiently full. The HALT and RESTORE trigger levels in the UARTi.UART_TCR register determine the levels at which uarti_rtsn is activated and deactivated.

If auto-CTS and auto-RTS are enabled, data transmission does not occur unless the RX FIFO has empty space. Thus, overrun errors are eliminated during hardware flow control. If auto-CTS and auto-RTS are not enabled, overrun errors occur if the transmit data rate exceeds the RX FIFO latency.

- Auto-RTS:

  Auto-RTS data flow control originates in the receiver block. The RX FIFO trigger levels used in auto-RTS are stored in the UARTi.UART_TCR register. uarti_rtsn is active if the RX FIFO level is below the HALT trigger level in the UARTi.UART_TCR[3:0] RX_FIFO_TRIG_HALT bit field. When the RX FIFO HALT trigger level is reached, uarti_rtsn is deasserted. The sending device (for example, another UART) can send an additional byte after the trigger level is reached because it may not recognize the deassertion of RTS until it begins sending the additional byte.

  uarti_rtsn is automatically reasserted when the RX FIFO reaches the RESUME trigger level programmed by the UARTi.UART_TCR[7:4] RX_FIFO_TRIG_START bit field. This reassertion requests the sending device to resume transmission.

  In this case, uarti_rtsn is an active-low signal.

- Auto-CTS:

  The transmitter circuitry checks uarti_ctsn before sending the next data byte. When uarti_ctsn is active, the transmitter sends the next byte. To stop the transmitter from sending the next byte, uarti_ctsn must be deasserted before the middle of the last stop-bit currently sent.

  The auto-CTS function reduces interrupts to the host system. When auto-CTS flow control is enabled, the uarti_ctsn state changes do not have to trigger host interrupts because the device automatically controls its own transmitter. Without auto-CTS, the transmitter sends any data present in the transmit FIFO, and a receiver overrun error can result.

  In this case, uarti_ctsn is an active-low signal.

18.2.4.8.1.3.3 Software Flow Control

Software flow control is enabled through the enhanced feature register (UARTi.UART_EFR) and the modem control register (UARTi.UART_MCR). Different combinations of software flow control can be enabled by setting different combinations of the UARTi.UART_EFR[3:0] bit field (see Table 18-106).

Two other enhanced features relate to software flow control:

- XON-any function (UARTi.UART_MCR[5]): Operation resumes after receiving any character after the XOFF character is recognized. If special character detect is enabled and special character is received after XOFF1, it does not resume transmission. The special character is stored in the RX FIFO.

  NOTE: The XON-any character is written into the RX FIFO even if it is a software flow character.

- Special character (UARTi.UART_EFR[5]): Incoming data is compared to XOFF2. When the special character is detected, the XOFF interrupt (UARTi.UART_IIR[4]) is set, but it does not halt transmission. The XOFF interrupt is cleared by a read of UARTi.UART_IIR. The special character is transferred to the RX FIFO. Special character does not work with XON2, XOFF2, or sequential XOFFs.

Table 18-106. UART_EFR[3:0] Software Flow Control Options

Bit 3	Bit 2	Bit 1	Bit 0	TX, RX Software Flow Controls
0	0	X	X	No transmit flow control
1	0	X	X	Transmit XON1, XOFF1
0	1	X	X	Transmit XON2, XOFF2

Table 18-106. UART_EFR[3:0] Software Flow Control Options (continued)

Bit 3	Bit 2	Bit 1	Bit 0	TX, RX Software Flow Controls
1	1	X	X	Transmit XON1, XON2: XOFF1, XOFF2(1)
X	X	0	0	No receive flow control
X	X	1	0	Receiver compares XON1, XOFF1
X	X	0	1	Receiver compares XON2, XOFF2
X	X	1	1	Receiver compares XON1, XON2: XOFF1, XOFF2(1)

(1) In these cases, the XON1 and XON2 characters or the XOFF1 and XOFF2 characters must be transmitted/received sequentially with XON1/XOFF1 followed by XON2/XOFF2.
XON1 is defined in the UARTi.UART_XON1_ADDR1[7:0] XON_WORD1 bit field. XON2 is defined in the UARTi.UART_XON2_ADDR2[7:0] XON_WORD2 bit field.
XOFF1 is defined in the UARTi.UART_XOFF1[7:0] XOFF_WORD1 bit field. XOFF2 is defined in the UARTi.UART_XOFF2[7:0] XOFF_WORD2 bit field.

18.2.4.8.1.3.3.1 Receive (RX)

When software flow control operation is enabled, the UART compares incoming data with XOFF1/2 programmed characters (in certain cases, XOFF1 and XOFF2 must be received sequentially). When the correct XOFF characters are received, transmission stops after transmission of the current character completes. Detection of XOFF also sets the UARTi.UART_IIR[4] bit (if enabled by UARTi.UART_IER[5]) and causes the interrupt line to go low.

To resume transmission, an XON1/2 character must be received (in certain cases, XON1 and XON2 must be received sequentially). When the correct XON characters are received, the UARTi.UART_IIR[4] bit is cleared and the XOFF interrupt disappears.

NOTE: When a parity, framing, or break error occurs while receiving a software flow control character, this character is treated as normal data and is written to the RX FIFO.

When XON-any and special character detect are disabled and software flow control is enabled, no valid XON or XOFF characters are written to the RX FIFO. For example, when UARTi.UART_EFR[1:0] = 0x2, if XON1 and XOFF1 characters are received, they are not written to the RX FIFO.

When pairs of software flow characters are programmed to be received sequentially (UARTi.UART_EFR[1:0] = 0x3), the software flow characters are not written to the RX FIFO if they are received sequentially. However, received XON1/XOFF1 characters must be written to the RX FIFO if the subsequent character is not XON2/XOFF2.

18.2.4.8.1.3.3.2 Transmit (TX)

Two XOFF1 characters are transmitted when the RX FIFO passes the trigger level programmed by UARTi.UART_TCR[3:0]. As soon as the RX FIFO reaches the trigger level programmed by UARTi.UART_TCR[7:4], two XON1 characters are sent, so the data transfer recovers.

NOTE: If software flow control is disabled after an XOFF character is sent, the module transmits XON characters automatically to enable normal transmission.

The transmission of XOFF(s)/XON(s) follows the same protocol as transmission of an ordinary byte from the TX FIFO. This means that even if the word length is 5, 6, or 7 characters, the 5, 6, or 7 LSBs of XOFF1/2 and XON1/2 are transmitted. The 5, 6, or 7 bits of a character are seldom transmitted, but this function is included to maintain compatibility with earlier designs.

It is assumed that software flow control and hardware flow control are never enabled simultaneously.

18.2.4.8.1.3.4 Autobauding Modes

In autobauding mode, the UART can extract transfer characteristics (speed, length, and parity) from an "at" (AT) command (ASCII code). These characteristics are used to receive data after an AT and to send data.

The following AT commands are valid:

AT	DATA	<CR>
at	DATA	<CR>
A/		
a/		

A line break during the acquisition of the sequence AT is not recognized, and an echo function is not implemented in hardware.

A/ and a/ are not used to extract characteristics, but they must be recognized because of their special meaning. A/ or a/ is used to instruct the software to repeat the last received AT command; therefore, an a/ always follows an AT, and transfer characteristics are not expected to change between an AT and an a/.

When a valid AT is received, AT and all subsequent data, including the final <CR> (0x0D), are saved to the RX FIFO. The autobaud state-machine waits for the next valid AT command. If an a/ (A/) is received, the a/ (A/) is saved in the RX FIFO and the state-machine waits for the next valid AT command.

On the first successful detection of the baud rate, the UART activates an interrupt to signify that the AT (upper or lower case) sequence is detected. The UARTi.UART_UASR register reflects the correct settings for the baud rate detected. Interrupt activity can continue in this fashion when a subsequent character is received. Therefore, it is recommended that the software enable the RHR interrupt when using the autobaud mode.

The following settings are detected in autobaud mode with a module clock of 48 MHz:

- Speed:
  - 115.2K baud
  - 57.6K baud
  - 38.4K baud
  - 28.8K baud
  - 19.2K baud
  - 14.4K baud
  - 9.6K baud
  - 4.8K baud
  - 2.4K baud
  - 1.2K baud
- Length: 7 or 8 bits
- Parity: Odd, even, or space

NOTE: The combination of 7-bit character plus space parity is not supported.

Autobauding mode is selected when the UARTi.UART_MDR1[2:0] MODE_SELECT bit field is set to 0x2. In UART autobauding mode, UARTi.UART_DLL, UARTi.UART_DLH, and UARTi.UART_LCR[5:0] bit field settings are not used; instead, UASR is updated with the configuration detected by the autobauding logic.

UASR Autobauding Status Register Use

This register is used to set up transmission according to the characteristics of the previous reception instead of the UARTi.UART_LCR, UARTi.UART_DLL, and UARTi.UART_DLH registers when the UART is in autobauding mode.

To reset the autobauding hardware (to start a new AT detection) or to set the UART in standard mode (no autobaud), the UARTi.UART_MDR1[2:0] MODE_ SELECT bit field must be set to reset state (0x7) and then to the UART in autobauding mode (0x2) or to the UART in standard mode (0x0).

Use limitation:

- Only 7- and 8-bit characters (5- and 6-bit not supported)
- 7-bit character with space parity not supported
- Baud rate between 1200 and 115,200 bps (10 possibilities)

18.2.4.8.1.3.5 Error Detection

When the UARTi.UART_LSR register is read, the UARTi.UART_LSR[4:2] bit field reflects the error bits (BI: break condition, FE: framing error, PE: parity error) of the character at the top of the RX FIFO (the next character to be read). Therefore, reading the UARTi.UART_LSR register and then reading the UARTi.UART_RHR register identifies errors in a character.

Reading the UARTi.UART_RHR register updates the BI, FE, and PE bits (see Table 18-94 for the UART mode interrupts).

The UARTi.UART_LSR[7] RX_FIFO_STS bit is set when there is an error in the RX FIFO and is cleared only when no errors remain in the RX FIFO.

NOTE: Reading the UARTi.UART_LSR register does not cause an increment of the RX FIFO read pointer. The RX FIFO read pointer is incremented by reading the UARTi.UART_RHR register.

Reading the UARTi.UART_LSR register clears the OE bit if it is set (see Table 18-94 for the UART mode interrupts).

18.2.4.8.1.3.6 Overrun During Receive

Overrun during receive occurs if the RX state-machine tries to write data into the RX FIFO when it is already full. When overrun occurs, the device interrupts the CPU with the UARTi.UART_IIR[5:1] IT_TYPE bit field set to 0x3 (receiver line status error) and discards the remaining portion of the frame.

Overrun also causes an internal flag to be set, which disables further reception. Before the next frame can be received, the CPU must:

- Reset the RX FIFO.
- Read the UARTi.UART_RESUME register, which clears the internal flag.

18.2.4.8.1.3.7 Time-Out and Break Conditions

18.2.4.8.1.3.7.1 Time-Out Counter

An RX idle condition is detected when the receiver line (uarti_rxd) is high for a time that equals 4x the programmed word length + 12 bits. uarti_rxd is sampled midway through each bit.

For sleep mode, the counter is reset when there is activity on uarti_rxd.

For the time-out interrupt, the counter counts only when there is data in the RX FIFO, and the count is reset when there is activity on uarti_rxd or when the UARTi.UART_RHR register is read.

18.2.4.8.1.3.7.2 Break Condition

When a break condition occurs, uarti_txd is pulled low. A break condition is activated by setting the UARTi.UART_LCR[6] BREAK_EN bit. The break condition is not aligned on word stream (a break condition can occur in the middle of a character). The only way to send a break condition on a full character is:

1. Reset the TX FIFO (if enabled).
2. Wait for the transmit shift register to empty (the UARTi.UART_LSR[6] TX_SR_E bit is set to 1).

3. Take a guard time according to stop-bit definition.
4. Set the BREAK_EN bit to 1.

The break condition is asserted while the BREAK_EN bit is set to 1.

18.2.5 UART Basic Programming Model

This section describes the procedure for operating the UART with FIFO and DMA or interrupts. This three-part procedure ensures the quick start of the UART. It does not cover every UART feature.

The first programming model covers software reset of the UART. The second programming model describes FIFO and DMA configuration. The last programming model describes protocol, baud rate, and interrupt configuration.

NOTE: Each programming model can be used independently of the other two; for instance, reconfiguring the FIFOs and DMA settings only.

Each programming model can be executed starting from any UART register access mode (register modes, submodes, and other register dependencies). However, if the UART register access mode is known before executing the programming model, some steps that enable or restore register access are optional. For more information, see Section 18.2.4.7.1, *Register Access Modes*.

18.2.5.1 Global Initialization

18.2.5.1.1 Surrounding Modules Global Initialization

This section identifies the requirements for initializing the surrounding modules when the UART module is to be used for the first time after a device reset. This initialization of surrounding modules is based on the integration of the UART.

For more information, see Section 18.2.3, *UART Integration*.

Table 18-107. Global Initialization of Surrounding Modules for UART

Surrounding Modules	Comments
PRCM	UART functional and interface clocks must be enabled. For more information, see Chapter 3, *Power, Reset, and Clock Management*.
Control module	Module-specific pad muxing must be set in the control module. For more information, see Chapter 13, *Control Module*.
IRQ_CROSSBAR	IRQ_CROSSBAR configuration must be done to allow module IRQs to be mapped to certain device INTC line. For more information see Section 13.4.6.4, *IRQ_CROSSBAR Module Functional Description*, in Chapter 13, *Control Module*
Interrupt controllers	Device INTCs must be configured to enable the interrupt request generation. For information about enabling interrupts, see *Chapter 12, Interrupt Controllers*.
DMA_CROSSBAR	DMA_CROSSBAR configuration must be done to allow module DREQs to be mapped to certain device DMA line. For more information see Section 13.4.6.4, *DMA_CROSSBAR Module Functional Description*, in Chapter 13, *Control Module*.
DMA controllers	DMA controllers configuration must be done to enable the module DMA channel request.
Interconnects	For information about the L4 interconnect configuration, see Chapter 9, *Interconnect*.

18.2.5.1.2 UART Module Global Initialization

The procedure in Table 18-108 can be used to initialize UART when performing software reset.

Table 18-108. UART Global Initialization

Step	Register/Bit Field/Programming Model	Value
Perform a software reset.	UART_SYSC[1] SOFTRESET	0x1
Wait until reset is finished.	UART_SYSS[0] RESETDONE	= 0x1

18.2.5.2 Mode selection

The Table 18-109 describes how to set different register access mode.

Table 18-109. Configure register access mode

Step	Register/Bit Field/Programming Model	Value
Set the regiser access mode A	UART_LCR[7]DIV_EN	0x1
	UART_LCR[7:0]	!=0xBF
Set the regiser access mode B	UART_LCR[7:0]	0xBF
Set the operational mode	UART_LCR[7]DIV_EN	0x1

18.2.5.3 Submode selection

This section describes how to set different register access submode.

Table 18-110. Configure register access submode TCR_TLR

Step	Register/Bit Field/Programming Model	Value
Configure the submode TCR_TLR		
Configure mode B	see Table 18-109	
Enable writing to register bits UART_MCR[7:5]	UART_EFR[4]ENHANCED_EN	0x1
Configure mode A	see Table 18-109	0x1
Set the submode TCR_TLR	UART_MCR[6]TCR_TLR	0x1

Table 18-111. Configure register access submode MSR_SPR

Step	Register/Bit Field/Programming Model	Value
First option configure the submode MSR_SPR		
Configure mode B	see Table 18-109	
Set the submode MSR_SPR	UART_EFR[4]ENHANCED_EN	0x0
Second option configure the submode MSR_SPR		
Configure mode B	see Table 18-109	
Enable writing to register bits UART_MCR[7:5]	UART_EFR[4]ENHANCED_EN	0x1
Set the submode MSR_SPR	UART_MCR[6]TCR_TLR	0x0

Table 18-112. Configure register access submode XOFF

Step	Register/Bit Field/Programming Model	Value
Configuration of the XOFF		
Configure B	see Table 18-109	
Set the submode XOFF	UART_EFR[4] ENHANCED_EN	0x0

18.2.5.4 Load FIFO trigger and DMA mode settings

18.2.5.4.1 DMA mode Settings

To enable and configure the DMA mode, perform the following steps:

Table 18-113. DMA mode settings

Step	Register/Bit Field/Programming Model	Value
Set the option of DMA mode configuration.	UART_SCR[0] DMA_MODE_CTL	0x-
IF Configure DMA mode 0 and 1	UART_SCR[0] DMA_MODE_CTL	0x0
Select the DMA mode, for more information see Section 18.2.4.6.4	UART_FCR[3] DMA_MODE	0x-
IF Configure DMA mode from 0 to 3	UART_SCR[0] DMA_MODE_CTL	0x1
Select the DMA mode, for more information see Section 18.2.4.6.4	UART_SCR[2:1] DMA_MODE_2	0x-

18.2.5.4.2 FIFO Trigger Settings

In this section is described configuration and settings of FIFO trigger level, which enable DMA and interrupt generation.

Table 18-114. Load FIFO triggers defined by the FCR

Step	Register/Bit Field/Programming Model	Value
Configure register submode TCR_TLR	see Table 18-110	0x-
Set the desire RX FIFO trigger level	UART_FCR[5:4]TX_FIFO_TRIG	0x-
Set the desire TX FIFO trigger level	UART_FCR[7:6]RX_FIFO_TRIG	0x-

Table 18-115. Load FIFO triggers defined by the TLR

Step	Register/Bit Field/Programming Model	Value
Configure register submode TCR_TLR	see Table 18-110	0x-
Set the desire RX FIFO trigger level	UART_TLR[7:4]RX_FIFO_TRIG_DMA	0x-
Set the desire TX FIFO trigger level	UART_TLR[3:0]TX_FIFO_TRIG_DMA	0x-

Table 18-116. Load FIFO triggers defined by the concatenated value

Step	Register/Bit Field/Programming Model	Value
Configure register submode TCR_TLR	see Table 18-110	0x-
Set the register bit	UART_SCR[7]RX_TRIG_GRANU1	0x1
Set the desire RX FIFO trigger level	UART_TLR[7:4]RX_FIFO_TRIG_DMA UART_FCR[7:6]RX_FIFO_TRIG	0x-
Set the register bit	UART_SCR[6]TX_TRIG_GRANU1	0x1
Set the desire TX FIFO trigger level	UART_TLR[3:0]TX_FIFO_TRIG_DMA UART_FCR[5:4]TX_FIFO_TRIG	0x-

18.2.5.5 Protocol, Baud rate and interrupt settings

18.2.5.5.1 Baud rate settings

Table 18-117. Baud rate settings

Step	Register/Bit Field/Programming Model	Value
Disable UART mode	UART_MDR1[2:0] MODE_SELECT	0x7
Switch to register configuration mode B	see Table 18-109	
Enable access to UART_IER[7:4]	UART_EFR[4] ENHANCED_EN	0x1
Switch register operational mode	see Table 18-109	
Disable sleep mode	UART_IER[4] SLEEP_MODE	0x0
Switch to register configuration mode A or B	see Table 18-109	
Set the appropriate divisor value	UART_DLL[7:0] CLOCK_LSB UART_DLH[5:0] CLOCK_MSB	0x-

18.2.5.5.2 Interrupt settings

Table 18-118. Interrupt settings

Step	Register/Bit Field/Programming Model	Value
Switch to register configuration mode B	see Table 18-109	0x7
Enable access to UART_IER[7:4]	UART_EFR[4] ENHANCED_EN	
Switch register operational mode	see Table 18-109	
Set the desired interrupt configuration (0: Disable the interrupt; 1: Enable the interrupt)	UART_IER[7] CTS_IT UART_IER[6] RTS_IT UART_IER[5] XOFF_IT UART_IER[4] SLEEP_MODE UART_IER[3] MODEM_STS_IT UART_IER[2] LINE_STS_IT UART_IER[1] THR_IT UART_IER[0] RHR_IT	0x-

18.2.5.5.3 Protocol settings

Load the desired protocol formatting (parity, stop-bit, character length) and switch to register operational mode.

Table 18-119. Protocol settings

Step	Register/Bit Field/Programming Model	Value
Load desired protocol formatting, see Section 18.2.4.8.1.3.1, *Frame Formatting*	UART_LCR[5] PARITY_TYPE_2 UART_LCR[4] PARITY_TYPE_1 UART_LCR[3] PARITY_EN UART_LCR[2] NB_STOP UART_LCR[1:0] PARITY_LENGTH	0x-
Swtich to register operational mode	UART_LCR[7] DIV_EN UART_LCR[6] BREAK_EN	0x0

18.2.5.5.4 UART Mode Selection

Table 18-120. UART mode selection

Step	Register/Bit Field/Programming Model	Value
Load the desired UART mode, see Section 18.2.4.7.2, *UART Mode Selection*	UART_MDR1[2:0] MODE_SELECT	0x-

18.2.5.6 Hardware and Software Flow Control Configuration

This section describes the programming steps to enable and configure hardware and software flow control. Hardware and software flow control cannot be used at the same time.

18.2.5.6.1 Hardware Flow Control Configuration

Table 18-121. Hardware Flow Control Configuration

Step	Register/Bit Field/Programming Model	Value
Configure register submode TCR_TLR	see Table 18-110	0x7
Load the start and halt trigger value.	UART_TCR[7:4] AUTO_RTS_START UART_TCR[3:0] AUTO_RTS_HALT	0x-
Enable or disable receive and transmit hardware flow control mode.	UART_EFR[7] AUTO_CTS_EN UART_EFR[6] AUTO_RTS_EN	0x-

18.2.5.6.2 Software Flow Control Configuration

Table 18-122. Software Flow Control Configuration

Step	Register/Bit Field/Programming Model	Value
Set the register access submode XOFF	see Table 18-112	
Load the software control characters	UART_XON1_ADDR1[7:0] XON_WORD1 UART_XON2_ADDR2[7:0] XON_WORD2 UART_XOFF1[7:0] XOFF_WORD1 UART_XOFF2[7:0] XOFF_WORD2	0x-
Set the register access submode TCR_TLR	see Table 18-110	
Enable or disable XON any function (0: Disable; 1: Enable).	UART_MCR[5] XON_EN	0x-
Load start and halt trigger value for software flow control	UART_TCR[7:4] AUTO_RTS_START UART_TCR[3:0] AUTO_RTS_HALT	0x-
Enable or disable special character function(0: Disable; 1: Enable)	UART_EFR[5] SPEC_CHAR	0x-
Set the software flow control mode	UART_EFR[3:0] SW_FLOW_CONTROL	0x-

18.2.6 UART Register Manual

18.2.6.1 UART Instance Summary

Table 18-123. UART Instance Summary

Module Name	Module Base Address	Size
UART1	0x4806 A000	136 Bytes
UART2	0x4806 C000	136 Bytes
UART3	0x4802 0000	136 Bytes

18.2.6.2 UART Registers

18.2.6.2.1 UART Register Summary

Table 18-124. UART Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	UART1 Physical Address	UART2 Physical Address	UART3 Physical Address
UART_THR	W	32	0x0000 0000	0x4806 A000	0x4806 C000	0x4802 0000
UART_DLL	RW	32	0x0000 0000	0x4806 A000	0x4806 C000	0x4802 0000
UART_RHR	R	32	0x0000 0000	0x4806 A000	0x4806 C000	0x4802 0000
UART_IER	RW	32	0x0000 0004	0x4806 A004	0x4806 C004	0x4802 0004
UART_DLH	RW	32	0x0000 0004	0x4806 A004	0x4806 C004	0x4802 0004
UART_FCR	W	32	0x0000 0008	0x4806 A008	0x4806 C008	0x4802 0008
UART_IIR	R	32	0x0000 0008	0x4806 A008	0x4806 C008	0x4802 0008
UART_EFR	RW	32	0x0000 0008	0x4806 A008	0x4806 C008	0x4802 0008
UART_LCR	RW	32	0x0000 000C	0x4806 A00C	0x4806 C00C	0x4802 000C
UART_XON1_ADDR1	RW	32	0x0000 0010	0x4806 A010	0x4806 C010	0x4802 0010
UART_MCR	RW	32	0x0000 0010	0x4806 A010	0x4806 C010	0x4802 0010
UART_LSR	R	32	0x0000 0014	0x4806 A014	0x4806 C014	0x4802 0014
UART_XON2_ADDR2	RW	32	0x0000 0014	0x4806 A014	0x4806 C014	0x4802 0014
UART_XOFF1	RW	32	0x0000 0018	0x4806 A018	0x4806 C018	0x4802 0018
UART_MSR	R	32	0x0000 0018	0x4806 A018	0x4806 C018	0x4802 0018
UART_TCR	RW	32	0x0000 0018	0x4806 A018	0x4806 C018	0x4802 0018
UART_XOFF2	RW	32	0x0000 001C	0x4806 A01C	0x4806 C01C	0x4802 001C
UART_TLR	RW	32	0x0000 001C	0x4806 A01C	0x4806 C01C	0x4802 001C
UART_SPR	RW	32	0x0000 001C	0x4806 A01C	0x4806 C01C	0x4802 001C
UART_MDR1	RW	32	0x0000 0020	0x4806 A020	0x4806 C020	0x4802 0020
RESERVED	RW	32	0x0000 0024	0x4806 A024	0x4806 C024	0x4802 0024
RESERVED	RW	32	0x0000 0028	0x4806 A028	0x4806 C028	0x4802 0028
RESERVED	RW	32	0x0000 002C	0x4806 A02C	0x4806 C02C	0x4802 002C
RESERVED	RW	32	0x0000 0030	0x4806 A030	0x4806 C030	0x4802 0030
RESERVED	RW	32	0x0000 0034	0x4806 A034	0x4806 C034	0x4802 0034
UART_UASR	R	32	0x0000 0038	0x4806 A038	0x4806 C038	0x4802 0038
RESERVED	RW	32	0x0000 003C	0x4806 A03C	0x4806 C03C	0x4802 003C
UART_SCR	RW	32	0x0000 0040	0x4806 A040	0x4806 C040	0x4802 0040
UART_SSR	RW	32	0x0000 0044	0x4806 A044	0x4806 C044	0x4802 0044
RESERVED	RW	32	0x0000 0048	0x4806 A048	0x4806 C048	0x4802 0048
UART_MVR	R	32	0x0000 0050	0x4806 A050	0x4806 C050	0x4802 0050

Table 18-124. UART Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	UART1 Physical Address	UART2 Physical Address	UART3 Physical Address
UART_SYSC	RW	32	0x0000 0054	0x4806 A054	0x4806 C054	0x4802 0054
UART_SYSS	R	32	0x0000 0058	0x4806 A058	0x4806 C058	0x4802 0058
UART_WER	RW	32	0x0000 005C	0x4806 A05C	0x4806 C05C	0x4802 005C
RESERVED	RW	32	0x0000 0060	0x4806 A060	0x4806 C060	0x4802 0060
UART_RXFIFO_LVL	R	32	0x0000 0064	0x4806 A064	0x4806 C064	0x4802 0064
UART_TXFIFO_LVL	R	32	0x0000 0068	0x4806 A068	0x4806 C068	0x4802 0068
UART_IER2	RW	32	0x0000 006C	0x4806 A06C	0x4806 C06C	0x4802 006C
UART_ISR2	RW	32	0x0000 0070	0x4806 A070	0x4806 C070	0x4802 0070
UART_FREQ_SEL	RW	32	0x0000 0074	0x4806 A074	0x4806 C074	0x4802 0074
RESERVED	R	32	0x0000 0078	0x4806 A078	0x4806 C078	0x4802 0078
RESERVED	R	32	0x0000 007C	0x4806 A07C	0x4806 C07C	0x4802 007C
UART_MDR3	RW	32	0x0000 0080	0x4806 A080	0x4806 C080	0x4802 0080
UART_TX_DMA_THRESHOLD	RW	32	0x0000 0084	0x4806 A084	0x4806 C084	0x4802 0084

18.2.6.2.2 UART Register Description

Table 18-125. UART_THR

Address Offset	0x0000 0000		
Physical Address	0x4806 A000 0x4806 C000 0x4802 0000	**Instance**	UART1 UART2 UART3
Description	The transmitter section consists of the transmit holding register (THR) and the transmit shift register. The THR is a 64-byte FIFO. The local host (LH) writes data to the THR. The data is placed in the transmit shift register where it is shifted out serially on the TX output. If the FIFO is disabled, location 0 of the FIFO stores the data.		
Type	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								THR							

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Write has no effect.	W	0x000000
7:0	THR	Transmit holding register	W	0x-

Table 18-126. Register Call Summary for Register UART_THR

UART
- UART Interrupt Management: [0]
- FIFO Management: [1]
- FIFO DMA Mode Operation: [2]
- UART Mode Selection: [3]
- UART Register Summary: [4]

Table 18-127. UART_RHR

Address Offset	0x0000 0000		
Physical Address	0x4806 A000 0x4806 C000 0x4802 0000	**Instance**	UART1 UART2 UART3
Description	The receiver section consists of the receiver holding register (RHR) and the receiver shift register. The RHR is a 64-byte FIFO. The receiver shift register receives serial data from RX input. The data is converted to parallel data and moved to the RHR. If the FIFO is disabled, location 0 of the FIFO stores the single data character. **Note:** If an overflow occurs, the data in the RHR is not overwritten.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								RHR							

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Read returns 0	R	0x000000
7:0	RHR	Receive holding register	R	0x-

Table 18-128. Register Call Summary for Register UART_RHR

UART

- UART Interrupt Management: [0][1][2]
- FIFO Management: [3][4]
- UART Mode Selection: [5]
- UART Mode: [6][7][8][9]
- UART Register Summary: [10]

Table 18-129. UART_DLL

Address Offset	0x0000 0000		
Physical Address	0x4806 A000 0x4806 C000 0x4802 0000	**Instance**	UART1 UART2 UART3
Description	This register, with UART_DLH, stores the 14-bit divisor for generation of the baud clock in the baud rate generator. DLH stores the most-significant part of the divisor. DLL stores the least-significant part of the divisor.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								CLOCK_LSB							

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Read returns 0. Write has no effect.	RW	0x000000
7:0	CLOCK_LSB	Stores the 8-bit LSB divisor value	RW	0x00

Table 18-130. Register Call Summary for Register UART_DLL

UART

- UART Mode Power Management: [0][1]
- FIFO DMA Mode Operation: [2][3]
- UART Mode Selection: [4][5][6][7]
- UART Mode: [8][9][10]
- Baud rate settings: [11]
- UART Register Summary: [12]
- UART Register Description: [13][14][15][16][17]

Table 18-131. UART_IER

Address Offset	0x0000 0004		
Physical Address	0x4806 A004 0x4806 C004 0x4802 0004	**Instance**	UART1 UART2 UART3
Description	Interrupt enable register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								CTS_IT	RTS_IT	XOFF_IT	SLEEP_MODE	MODEM_STS_IT	LINE_STS_IT	THR_IT	RHR_IT

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Read returns 0. Write has no effect.	RW	0x000000
7	CTS_IT	0x0: Disables the CTS* interrupt 0x1: Enables the CTS* interrupt	RW	0
6	RTS_IT	0x0: Disables the RTS* interrupt 0x1: Enables the RTS* interrupt	RW	0
5	XOFF_IT	0x0: Disables the XOFF interrupt 0x1: Enables the XOFF interrupt	RW	0
4	SLEEP_MODE	0x0: Disables sleep mode 0x1: Enables sleep mode (stop baud rate clock when the module is inactive)	RW	0
3	MODEM_STS_IT	0x0: Disables the modem status register interrupt 0x1: Enables the modem status register interrupt	RW	0
2	LINE_STS_IT	0x0: Disables the receiver line status interrupt 0x1: Enables the receiver line status interrupt	RW	0
1	THR_IT	0x0: Disables the THR interrupt 0x1: Enables the THR interrupt	RW	0
0	RHR_IT	0x0: Disables the RHR interrupt and time-out interrupt 0x1: Enables the RHR interrupt and time-out interrupt	RW	0

Table 18-132. Register Call Summary for Register UART_IER

UART

- UART Mode Power Management: [0][1]
- FIFO Interrupt Mode: [2]
- FIFO Polled Mode Operation: [3]
- UART Mode Selection: [4][5]
- UART Mode: [6]
- Baud rate settings: [7][8]
- Interrupt settings: [9][10][11][12][13][14][15][16][17]
- UART Register Summary: [18]
- UART Register Description: [19]

Table 18-133. UART_DLH

Address Offset	0x0000 0004		
Physical Address	0x4806 A004 0x4806 C004 0x4802 0004	**Instance**	UART1 UART2 UART3
Description	This register, with UART_DLL, stores the 14-bit divisor for generating the baud clock in the baud rate generator. DLH stores the most-significant part of the divisor. DLL stores the least-significant part of the divisor.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								RESERVED		CLOCK_MSB					

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Read returns 0. Write has no effect.	RW	0x000000
7:6	RESERVED	Read returns 0. Write has no effect.	RW	0x0
5:0	CLOCK_MSB	Stores the 6-bit MSB divisor value	RW	0x00

Table 18-134. Register Call Summary for Register UART_DLH

UART

- UART Mode Power Management: [0][1]
- FIFO DMA Mode Operation: [2][3]
- UART Mode Selection: [4][5][6][7]
- UART Mode: [8][9][10]
- Baud rate settings: [11]
- UART Register Summary: [12]
- UART Register Description: [13][14][15][16][17]

Table 18-135. UART_IIR

Address Offset	0x0000 0008		
Physical Address	0x4806 A008 0x4806 C008 0x4802 0008	**Instance**	UART1 UART2 UART3
Description	Interrupt identification register. The IIR is a read-only register that provides the source of the interrupt in a prioritized manner.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								FCR_MIRROR		IT_TYPE				IT_PENDING	

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Read returns 0. Write has no effect.	R	0x000000
7:6	FCR_MIRROR	Mirror the contents of UART_FCR[0] on both bits.	R	0x0
5:1	IT_TYPE	Read 0x0: Modem interrupt. Priority = 4	R	0x00
		Read 0x1: THR interrupt. Priority = 3		

Bits	Field Name	Description	Type	Reset
		Read 0x2: RHR interrupt. Priority = 2		
		Read 0x3: Receiver line status error. Priority = 3		
		Read 0x6: Rx time-out. Priority = 2		
		Read 0x8: XOFF/special character. Priority = 5		
		Read 0x10: CTS, RTS, DSR change state from active (low) to inactive (high) Priority = 6		
0	IT_PENDING	Read 0x0: An interrupt is pending.	R	1
		Read 0x1: No interrupt is pending.		

Table 18-136. Register Call Summary for Register UART_IIR

UART

- Block Diagram: [0]
- UART Interrupt Management: [1][2][3][4][5][6][7]
- UART Mode Selection: [8][9]
- UART Mode: [10][11][12][13][14]
- UART Register Summary: [15]
- UART Register Description: [16][17][18][19]

Table 18-137. UART_FCR

Address Offset	0x0000 0008		
Physical Address	0x4806 A008 0x4806 C008 0x4802 0008	**Instance**	UART1 UART2 UART3
Description	FIFO control register **Notes:** Bits 4 and 5 can only be written to when UART_EFR[4] = 1. Bits 0 and 3 can be changed only when the baud clock is not running (DLL and DLH set to 0). See Table 18-95 for UART_FCR[5:4] setting restriction when UART_SCR[6] = 1. See Table 18-96 for UART_FCR[7:6] setting restriction when UART_SCR[7] = 1.		
Type	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								RX_FIFO_TRIG		TX_FIFO_TRIG		DMA_MODE	TX_FIFO_CLEAR	RX_FIFO_CLEAR	FIFO_EN

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Write has no effect.	W	0x000000
7:6	RX_FIFO_TRIG	Sets the trigger level for the RX FIFO: If UART_SCR[7] = 0 and UART_TLR[7:4] = 0000: 00: 8 characters 01: 16 characters 10: 56 characters 11: 60 characters If UART_SCR[7] = 0 and UART_TLR[7:4] != 0000, RX_FIFO_TRIG is not considered. If UART_SCR[7] = 1, RX_FIFO_TRIG is 2 LSBs of the trigger level (1-63 on 6 bits) with the granularity 1.	W	0x0

Bits	Field Name	Description	Type	Reset
5:4	TX_FIFO_TRIG	Sets the trigger level for the TX FIFO: If UART_SCR[6] = 0 and UART_TLR[3:0] = 0000: 00: 8 spaces 01: 16 spaces 10: 32 spaces 11: 56 spaces If UART_SCR[6] = 0 and UART_TLR[3:0] != 0000, TX_FIFO_TRIG is not considered. If UART_SCR[6] = 1, TX_FIFO_TRIG is 2 LSBs of the trigger level (1-63 on 6 bits) with the granularity 1	W	0x0
3	DMA_MODE	This register is considered if UART_SCR[0] = 0. Write 0x0: DMA_MODE 0 (No DMA) Write 0x1: DMA_MODE 1 (UART_nDMA_REQ[0] in TX (UARTi_DREQ_TX), UART_nDMA_REQ[1] in RX (UARTi_DREQ_RX))	W	0
2	TX_FIFO_CLEAR	Write 0x0: No change Write 0x1: Clears the TX FIFO and resets its counter logic to 0. Returns to 0 after clearing FIFO.	W	0
1	RX_FIFO_CLEAR	Write 0x0: No change Write 0x1: Clears the RX FIFO and resets its counter logic to 0. Returns to 0 after clearing FIFO.	W	0
0	FIFO_EN	Write 0x0: Disables the transmit and RX FIFOs. The transmit and receive holding registers are 1-byte FIFOs. Write 0x1: Enables the transmit and RX FIFOs. The transmit and receive holding registers are 64-byte FIFOs.	W	0

Table 18-138. Register Call Summary for Register UART_FCR

UART

- FIFO Management: [0][1][2]
- FIFO Trigger: [3][4][5]
- FIFO Interrupt Mode: [6][7][8]
- FIFO Polled Mode Operation: [9]
- FIFO DMA Mode Operation: [10][11][12][13][14][15][16][17][18][19][20][21][22][23][24][25][26][27][28]
- UART Mode Selection: [29][30]
- DMA mode Settings: [31]
- FIFO Trigger Settings: [32][33][34][35]
- UART Register Summary: [36]
- UART Register Description: [37][38][39][40][41][42][43][44][45][46][47]

Table 18-139. UART_EFR

Address Offset	0x0000 0008		
Physical Address	0x4806 A008 0x4806 C008 0x4802 0008	**Instance**	UART1 UART2 UART3
Description	Enhanced feature register This register enables or disables enhanced features.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								AUTO_CTS_EN	AUTO_RTS_EN	SPECIAL_CHAR_DETECT	ENHANCED_EN	SW_FLOW_CONTROL			

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Read returns 0. Write has no effect.	RW	0x000000
7	AUTO_CTS_EN	Auto-CTS enable bit 0x0: Normal operation 0x1: Auto-CTS flow control is enabled. Transmission is halted when the CTS* pin is high (inactive).	RW	0
6	AUTO_RTS_EN	Auto-RTS enable bit 0x0: Normal operation 0x1: Auto-RTS flow control is enabled. RTS* pin goes high (inactive) when the RX FIFO HALT trigger level, UART_TCR[3:0], is reached, and goes low (active) when the RX FIFO RESTORE transmission trigger level is reached.	RW	0
5	SPECIAL_CHAR_DETECT	0x0: Normal operation 0x1: Special character detect enable. Received data is compared with XOFF2 data. If a match occurs, the received data is transferred to the RX FIFO and the UART_IIR[4] bit is set to 1 to indicate that a special character was detected.	RW	0
4	ENHANCED_EN	Enhanced functions write enable bit 0x0: Disables writing to IER bits 4-7, UART_FCR bits 4-5, and UART_MCR bits 5-7. 0x1: Enables writing to IER bits 4-7, UART_FCR bits 4-5, and UART_MCR bits 5-7.	RW	0
3:0	SW_FLOW_CONTROL	Combinations of software flow control can be selected by programming bit 3 - bit 0. See Table 18-106.	RW	0x0

Table 18-140. Register Call Summary for Register UART_EFR

UART

- UART Mode Power Management: [0]
- Register Access Modes: [1][2][3][4][5][6]
- UART Mode Selection: [7][8]
- UART Mode: [9][10][11][12][13][14]
- Submode selection: [15][16][17][18]
- Baud rate settings: [19]
- Interrupt settings: [20]
- Hardware Flow Control Configuration: [21][22]
- Software Flow Control Configuration: [23][24]
- UART Register Summary: [25]
- UART Register Description: [26]

Table 18-141. UART_LCR

Address Offset	0x0000 000C		
Physical Address	0x4806 A00C 0x4806 C00C 0x4802 000C	**Instance**	UART1 UART2 UART3
Description	Line control register LCR[6:0] define transmission and reception parameters. **Note:** When LCR[6] is set to 1, the TX line is forced to 0 and remains in this state as long as LCR[6] = 1.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								DIV_EN	BREAK_EN	PARITY_TYPE2	PARITY_TYPE1	PARITY_EN	NB_STOP	CHAR_LENGTH	

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Read returns 0. Write has no effect.	RW	0x000000
7	DIV_EN	0x0: Normal operating condition 0x1: Divisor latch enable. Allows access to UART_DLL, UART_DLH, and other registers (see Table 18-97).	RW	0
6	BREAK_EN	Break control bit 0x0: Normal operating condition 0x1: Forces the transmitter output to go low to alert the communication terminal	RW	0
5	PARITY_TYPE2	Selects the forced parity format (if UART_LCR[3] = 1). If UART_LCR[5] = 1 and UART_LCR[4] = 0, the parity bit is forced to 1 in the transmitted and received data. If UART_LCR[5] = 1 and UART_LCR[4] = 1, the parity bit is forced to 0 in the transmitted and received data.	RW	0
4	PARITY_TYPE1	0x0: Odd parity is generated (if UART_LCR[3] = 1). 0x1: Even parity is generated (if UART_LCR[3] = 1).	RW	0

UART_LCR[3]	UART_LCR[4]	UART_LCR[5]	Parity
0	N/A	N/A	No parity
1	0	0	Odd parity
1	1	0	Even parity
1	0	1	Forced 1
1	1	1	Forced 0

Bits	Field Name	Description	Type	Reset
3	PARITY_EN	0x0: No parity 0x1: A parity bit is generated during transmission and the receiver checks for received parity.	RW	0
2	NB_STOP	Specifies the number of stop-bits 0x0: 1 stop-bit (word length = 5, 6, 7, 8) 0x1: 1.5 stop-bits (word length = 5) 2 stop-bits (word length = 6, 7, 8)	RW	0
1:0	CHAR_LENGTH	Specifies the word length to be transmitted or received 0x0: 5 bits 0x1: 6 bits 0x2: 7 bits 0x3: 8 bits	RW	0x0

Table 18-142. Register Call Summary for Register UART_LCR

UART

- Register Access Modes: [0][1][2][3][4][5]
- UART Mode Selection: [6][7][8][9][10][11]
- UART Mode: [12][13][14][15][16][17][18]
- Mode selection: [19][20][21][22]
- Protocol settings: [23][24][25][26][27][28][29]
- UART Register Summary: [30]
- UART Register Description: [31][32][33][34][35][36][37][38][39][40][41][42]

Table 18-143. UART_XON1_ADDR1

Address Offset	0x0000 0010		
Physical Address	0x4806 A010 0x4806 C010 0x4802 0010	**Instance**	UART1 UART2 UART3
Description	UART mode: XON1 character		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								XON_WORD1							

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Read returns 0. Write has no effect.	RW	0x000000
7:0	XON_WORD1	Stores the 8-bit XON1 character in UART modes	RW	0x00

Table 18-144. Register Call Summary for Register UART_XON1_ADDR1

UART

- UART Mode Selection: [0][1]
- UART Mode: [2]
- Software Flow Control Configuration: [3]
- UART Register Summary: [4]

Table 18-145. UART_MCR

Address Offset	0x0000 0010		
Physical Address	0x4806 A010 0x4806 C010 0x4802 0010	**Instance**	UART1 UART2 UART3
Description	Modem control register MCR[3:0] controls the interface with the modem, data set, or peripheral device that emulates the modem.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								RESERVED	TCR_TLR	XON_EN	LOOPBACK_EN	CD_STS_CH	RI_STS_CH	RTS	DTR

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Read returns 0. Write has no effect.	RW	0x000000
7	RESERVED	Read returns 0. Write has no effect.	RW	0
6	TCR_TLR	0x0: No action 0x1: Enables access to the UART_TCR and UART_TLR registers	RW	0
5	XON_EN	0x0: Disable XON any function. 0x1: Enable XON any function.	RW	0
4	LOOPBACK_EN	0x0: Normal operating mode 0x1: Enable local loopback mode (internal). In this mode, the MCR[3:0] signals are looped back into the UART_MSR[7:4] bit field. The transmit output is looped back to the receive input internally.	RW	0
3	CD_STS_CH	0x0: In loopback, forces DCD* input high and IRQ outputs to inactive state 0x1: In loopback, forces DCD* input low and IRQ outputs to inactive state	RW	0
2	RI_STS_CH	0x0: In loopback, forces RI* input high 0x1: In loopback, forces RI* input low	RW	0
1	RTS	In loopback, controls the UART_MSR[4] bit. If auto-RTS is enabled, the RTS* output is controlled by hardware flow control. 0x0: Force RTS* output to inactive (high). 0x1: Force RTS* output to active (low).	RW	0
0	DTR	0x0: Force DTR* output to inactive (high). 0x1: Force DTR* output to active (low).	RW	0

Table 18-146. Register Call Summary for Register UART_MCR

UART

- UART Interface Description: [0][1]
- Register Access Modes: [2][3][4][5][6][7]
- UART Mode Selection: [8][9][10][11]
- UART Mode: [12][13]
- Submode selection: [14][15][16][17]
- Software Flow Control Configuration: [18]
- UART Register Summary: [19]
- UART Register Description: [20][21][22][23][24][25][26][27][28][29]

Table 18-147. UART_LSR

Address Offset	0x0000 0014		
Physical Address	0x4806 A014 0x4806 C014 0x4802 0014	**Instance**	UART1 UART2 UART3
Description	Line status register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								RX_FIFO_STS	TX_SR_E	TX_FIFO_E	RX_BI	RX_FE	RX_PE	RX_OE	RX_FIFO_E

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Read returns 0.	R	0x000000
7	RX_FIFO_STS	Read 0x0: Normal operation Read 0x1: At least one parity error, framing error, or break indication in the RX FIFO. Bit 7 is cleared when no more errors are present in the RX FIFO.	R	0
6	TX_SR_E	Read 0x0: Transmitter hold (TX FIFO) and shift registers are not empty. Read 0x1: Transmitter hold (TX FIFO) and shift registers are empty.	R	1
5	TX_FIFO_E	Read 0x0: Transmit hold register (TX FIFO) is not empty. Read 0x1: Transmit hold register (TX FIFO) is empty. The transmission is not necessarily complete.	R	1
4	RX_BI	Read 0x0: No break condition Read 0x1: A break was detected while the data from the RX FIFO was received (for example, RX input was low for one character + 1 bit time frame).	R	0
3	RX_FE	Read 0x0: No framing error in data RX FIFO Read 0x1: Framing error occurred in data from RX FIFO (received data did not have a valid stop-bit).	R	0
2	RX_PE	Read 0x0: No parity error in data from RX FIFO Read 0x1: Parity error in data from RX FIFO	R	0
1	RX_OE	Read 0x0: No overrun error Read 0x1: Overrun error occurred. Set when the character in the receive shift register is not transferred to the RX FIFO. This occurs only when the RX FIFO is full.	R	0
0	RX_FIFO_E	Read 0x0: No data in the RX FIFO Read 0x1: At least one data character in the RX FIFO	R	0

Table 18-148. Register Call Summary for Register UART_LSR

UART

- UART Interrupt Management: [0][1]
- FIFO Polled Mode Operation: [2]
- UART Mode Selection: [3][4]
- UART Mode: [5][6][7][8][9][10][11]
- UART Register Summary: [12]

Table 18-149. UART_XON2_ADDR2

Address Offset	0x0000 0014		
Physical Address	0x4806 A014 0x4806 C014 0x4802 0014	**Instance**	UART1 UART2 UART3
Description	Stores the 8-bit XON2 character in UART moldes		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								XON_WORD2							

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Read returns 0. Write has no effect.	RW	0x000000
7:0	XON_WORD2	Stores the 8-bit XON2 character in UART modes	RW	0x00

Table 18-150. Register Call Summary for Register UART_XON2_ADDR2

UART

- UART Mode Selection: [0][1]
- UART Mode: [2]
- Software Flow Control Configuration: [3]
- UART Register Summary: [4]

Table 18-151. UART_TCR

Address Offset	0x0000 0018		
Physical Address	0x4806 A018 0x4806 C018 0x4802 0018	**Instance**	UART1 UART2 UART3
Description	Transmission control register This register stores the RX FIFO threshold levels to start/stop transmission during hardware/software flow control. Notes: Trigger levels from 0 to 60 bytes are available with a granularity of 4. (Trigger level = 4 x [4-bit register value]) The programmer must ensure that UART_TCR[3:0] > UART_TCR[7:4] when auto-RTS or software flow control is enabled to avoid a mis-operation of the device. In FIFO interrupt mode with flow control, the programmer must ensure that the trigger level to halt transmission is greater than or equal to the RX FIFO trigger level (UART_TLR[7:4] or UART_FCR[7:6]); otherwise, FIFO operation stalls. In FIFO DMA mode with flow control, this concept does not exist because a DMA request is sent each time a byte is received.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								RX_FIFO_TRIG_START				RX_FIFO_TRIG_HALT			

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Read returns 0. Write has no effect.	RW	0x000000
7:4	RX_FIFO_TRIG_START	RX FIFO trigger level to RESTORE transmission (0 - 60)	RW	0x0
3:0	RX_FIFO_TRIG_HALT	RX FIFO trigger level to HALT transmission (0 - 60)	RW	0xF

Table 18-152. Register Call Summary for Register UART_TCR

UART

- FIFO Trigger: [0][1][2]
- FIFO Interrupt Mode: [3]
- UART Mode Selection: [4][5][6][7][8][9]
- UART Mode: [10][11][12][13][14][15]
- Hardware Flow Control Configuration: [16][17]
- Software Flow Control Configuration: [18][19]
- UART Register Summary: [20]
- UART Register Description: [21][22][23][24]

Table 18-153. UART_XOFF1

Address Offset	0x0000 0018		
Physical Address	0x4806 A018 0x4806 C018 0x4802 0018	**Instance**	UART1 UART2 UART3
Description	UART mode XOFF1 character		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								XOFF_WORD1							

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Read returns 0. Write has no effect.	RW	0x000000
7:0	XOFF_WORD1	Stores the 8-bit XOFF1 character used in UART modes	RW	0x00

Table 18-154. Register Call Summary for Register UART_XOFF1

UART

- UART Mode Selection: [0][1]
- UART Mode: [2]
- Software Flow Control Configuration: [3]
- UART Register Summary: [4]

Table 18-155. UART_MSR

Address Offset	0x0000 0018		
Physical Address	0x4806 A018 0x4806 C018 0x4802 0018	**Instance**	UART1 UART2 UART3
Description	Modem status register. This register provides information about the current state of the control lines from the modem, data set, or peripheral device to the LH. It also indicates when a control input from the modem changes state.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								NCD_STS	NRI_STS	NDSR_STS	NCTS_STS	DCD_STS	RI_STS	DSR_STS	CTS_STS

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Read returns 0.	R	0x000000
7	NCD_STS	This bit is the complement of the DCD* input. In loopback mode, it is equivalent to UART_MCR[3].	R	-
6	NRI_STS	This bit is the complement of the RI* input. In loopback mode, it is equivalent to UART_MCR[2].	R	-
5	NDSR_STS	This bit is the complement of the DSR* input. In loopback mode, it is equivalent to UART_MCR[0].	R	-
4	NCTS_STS	This bit is the complement of the CTS* input. In loopback mode, it is equivalent to UART_MCR[1].	R	-
3	DCD_STS	Indicates that DCD* input (or UART_MCR[3] in loopback) changed. Cleared on a read.	R	0
2	RI_STS	Indicates that RI* input (or UART_MCR[2] in loopback) changed state from low to high. Cleared on a read.	R	0
1	DSR_STS	Read 0x1: Indicates that DSR* input (or UART_MCR[0] in loopback) changed state. Cleared on a read.	R	0
0	CTS_STS	Read 0x1: Indicates that CTS* input (or UART_MCR[1] in loopback) changed state. Cleared on a read.	R	0

Table 18-156. Register Call Summary for Register UART_MSR

UART
- UART Interface Description: [0][1]
- UART Interrupt Management: [2]
- UART Mode Selection: [3][4]
- UART Register Summary: [5]
- UART Register Description: [6][7]

Table 18-157. UART_SPR

Address Offset	0x0000 001C		
Physical Address	0x4806 A01C 0x4806 C01C 0x4802 001C	**Instance**	UART1 UART2 UART3
Description	Scratchpad register This read/write register does not control the module. It is a scratchpad register to be used by the programmer to hold temporary data.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								SPR_WORD							

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Read returns 0. Write has no effect.	RW	0x000000
7:0	SPR_WORD	Scratchpad register	RW	0x00

Table 18-158. Register Call Summary for Register UART_SPR

UART
- UART Mode Selection: [0][1][2][3]
- UART Register Summary: [4]

Table 18-159. UART_TLR

Address Offset	0x0000 001C		
Physical Address	0x4806 A01C 0x4806 C01C 0x4802 001C	**Instance**	UART1 UART2 UART3
Description	Trigger level register This register stores the programmable transmit and RX FIFO trigger levels for DMA and IRQ generation.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								RX_FIFO_TRIG_DMA				TX_FIFO_TRIG_DMA			

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Read returns 0. Write has no effect.	RW	0x000000
7:4	RX_FIFO_TRIG_DMA	Receive FIFO trigger level	RW	0x0
3:0	TX_FIFO_TRIG_DMA	Transmit FIFO trigger level	RW	0x0

Table 18-160. Register Call Summary for Register UART_TLR

UART

- FIFO Management: [0]
- FIFO Trigger: [1][2]
- FIFO Interrupt Mode: [3][4]
- FIFO DMA Mode Operation: [5][6][7][8]
- UART Mode Selection: [9][10][11][12][13][14]
- FIFO Trigger Settings: [15][16][17][18]
- UART Register Summary: [19]
- UART Register Description: [20][21][22][23][24][25]

Table 18-161. UART_XOFF2

Address Offset	0x0000 001C		
Physical Address	0x4806 A01C 0x4806 C01C 0x4802 001C	**Instance**	UART1 UART2 UART3
Description	UART mode XOFF2 character		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								XOFF_WORD2							

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Read returns 0. Write has no effect.	RW	0x000000
7:0	XOFF_WORD2	Stores the 8-bit XOFF2 character used in UART modes.	RW	0x00

Table 18-162. Register Call Summary for Register UART_XOFF2

UART

- UART Mode Selection: [0][1]
- UART Mode: [2]
- Software Flow Control Configuration: [3]
- UART Register Summary: [4]

Table 18-163. UART_MDR1

Address Offset	0x0000 0020		
Physical Address	0x4806 A020 0x4806 C020 0x4802 0020	**Instance**	UART1 UART2 UART3
Description	Mode definition register 1 The mode of operation can be programmed by writing to MDR1[2:0] and therefore the MDR1 must be programmed on startup after configuration of the configuration registers (UART_DLL, UART_DLH, and UART_LCR). The value of MDR1[2:0] must not be changed again during normal operation. **Note:** If the module is disabled by setting the MODE_SELECT field to 111, interrupt requests can still be generated unless disabled through the interrupt enable register (UART_IER). In this case, UART mode interrupts are visible. Reading the interrupt identification register (UART_IIR) shows UART mode interrupt flags.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								FRAME_END_MODE	SIP_MODE	SCT	SET_TXIR	IR_SLEEP	MODE_SELECT		

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Read returns 0. Write has no effect.	RW	0x000000
7	FRAME_END_MODE	Reserved	RW	0
6	SIP_MODE	Reserved	RW	0
5	SCT	Reserved	RW	0
4	SET_TXIR	Reserved	RW	0
3	IR_SLEEP	Reserved	RW	0
2:0	MODE_SELECT	0x0: UART 16x mode 0x1: Reserved 0x2: UART 16x auto-baud 0x3: UART 13x mode 0x4: Reserved 0x5: Reserved 0x6: Reserved 0x7: Disable (default state)	RW	0x7

Table 18-164. Register Call Summary for Register UART_MDR1

UART

- UART Interface Description: [0][1]
- UART Protocol and Data Format: [2]
- UART Mode Selection: [3][4][5][6][7][8][9]
- UART Mode: [10][11][12]
- Baud rate settings: [13]
- UART Mode Selection: [14]
- UART Register Summary: [15]
- UART Register Description: [16][17][18]

Table 18-165. UART_UASR

Address Offset	0x0000 0038		
Physical Address	0x4806 A038 0x4806 C038 0x4802 0038	**Instance**	UART1 UART2 UART3
Description	UART autobauding status register This status register returns the speed, the number of bits by characters, and the type of the parity in UART autobauding mode. In autobauding mode, the input frequency of the UART modem must be fixed to 48 MHz. Any other module clock frequency results in incorrect baud rate recognition. **Note:** When the UART is in autobauding mode, this register, instead of the UART_LCR, UART_DLL, and UART_DLH registers, is used to set up transmission according to the characteristics of the previous reception. To reset the autobauding hardware (to start a new AT detection), set UART_MDR1[2:0] to 111 (reset value), then set UART_MDR1[2:1] to 010 (UART in autobaud mode). To set the UART to standard mode (no autobaud), set UART_MDR1[2:1] to 000.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								PARITY_TYPE		BIT_BY_CHAR	SPEED				

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Read returns 0.	R	0x000000
7:6	PARITY_TYPE	Read 0x0: No parity identified	R	0x0
		Read 0x1: Parity space		
		Read 0x2: Even parity		
		Read 0x3: Odd parity		
5	BIT_BY_CHAR	Read 0x0: 7-bit character identified	R	0
		Read 0x1: 8-bit character identified		
4:0	SPEED	Used to report the speed identified	R	0x00
		Read 0x0: No speed identified		
		Read 0x1: 115,200 baud		
		Read 0x2: 57,600 baud		
		Read 0x3: 38,400 baud		
		Read 0x4: 28,800 baud		
		Read 0x5: 19,200 baud		
		Read 0x6: 14,400 baud		
		Read 0x7: 9,600 baud		
		Read 0x8: 4,800 baud		
		Read 0x9: 2,400 baud		

Bits	Field Name	Description	Type	Reset
		Read 0xA: 1,200 baud		

Table 18-166. Register Call Summary for Register UART_UASR

UART

- UART Mode Selection: [0][1]
- UART Mode: [2]
- UART Register Summary: [3]

Table 18-167. UART_SCR

Address Offset	0x0000 0040		
Physical Address	0x4806 A040 0x4806 C040 0x4802 0040	**Instance**	UART1 UART2 UART3
Description	Supplementary control register **Note:** Bit 4 enables the wake-up interrupt, but this interrupt is not mapped into the UART_IIR register. Therefore, when an interrupt occurs and there is no interrupt pending in the UART_IIR register, the UART_SSR[1] bit must be checked. To clear the wake-up interrupt, bit UART_SCR[4] must be reset to 0.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								RX_TRIG_GRANU1	TX_TRIG_GRANU1	DSR_IT	RX_CTS_DSR_WAKE_UP_ENABLE	TX_EMPTY_CTL_IT	DMA_MODE_2		DMA_MODE_CTL

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Read returns 0. Write has no effect.	RW	0x000000
7	RX_TRIG_GRANU1	0x0: Disables the granularity of 1 for trigger RX level 0x1: Enables the granularity of 1 for trigger RX level	RW	0
6	TX_TRIG_GRANU1	0x0: Disables the granularity of 1 for trigger TX level 0x1: Enables the granularity of 1 for trigger TX level	RW	0
5	DSR_IT	NOT USED IN THIS DEVICE	RW	0
4	RX_CTS_DSR_WAKE_UP_ENABLE	0x0: Disables the wake-up interrupt and clears SSR[1] 0x1: Waits for a falling edge of pins RX or CTS* to generate an interrupt	RW	0
3	TX_EMPTY_CTL_IT	0x0: Normal mode for THR interrupt (see UART mode interrupts table) 0x1: The THR interrupt is generated when TX FIFO and TX shift register are empty.	RW	0

Bits	Field Name	Description	Type	Reset
2:1	DMA_MODE_2	Used to specify the DMA mode valid if the UART_SCR[0] bit = 1 0x0: DMA mode 0 (no DMA) 0x1: DMA mode 1 (UART_nDMA_REQ[0] in TX, UART_nDMA_REQ[1] in RX) 0x2: DMA mode 2 (UART_nDMA_REQ[0] in RX) 0x3: DMA mode 3 (UART_nDMA_REQ[0] in TX)	RW	0x0
0	DMA_MODE_CTL	0x0: The DMA_MODE is set with UART_FCR[3]. 0x1: The DMA_MODE is set with UART_SCR[2:1].	RW	0

Table 18-168. Register Call Summary for Register UART_SCR

UART

- UART Mode Power Management: [0]
- UART Interrupt Management: [1][2]
- FIFO Management: [3]
- FIFO DMA Mode Operation: [4][5][6][7][8][9][10][11][12][13][14][15][16][17][18][19][20][21][22][23][24][25][26][27]
- UART Mode Selection: [28][29][30][31][32][33]
- DMA mode Settings: [34][35][36][37]
- FIFO Trigger Settings: [38][39]
- UART Register Summary: [40]
- UART Register Description: [41][42][43][44][45][46][47][48][49][50][51][52][53]

Table 18-169. UART_SSR

Address Offset	0x0000 0044		
Physical Address	0x4806 A044 0x4806 C044 0x4802 0044	**Instance**	UART1 UART2 UART3
Description	Supplementary status register **Note:** Bit 1 is reset only when UART_SCR[4] is reset to 0.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								RESERVED					DMA_COUNTER_RST	RX_CTS_DSR_WAKE_UP_STS	TX_FIFO_FULL

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Read returns 0.	R	0x000000
7:3	RESERVED	Read returns 0.	R	0x00
2	DMA_COUNTER_RST	0x0: The DMA counter will not be reset if the corresponding FIFO is reset (through UART_FCR[1] or UART_FCR[2]). 0x1: The DMA counter will be reset if corresponding FIFO is reset (through UART_FCR[1] or UART_FCR[2]).	RW	1
1	RX_CTS_DSR_WAKE_UP_STS	Read 0x0: No falling edge event on RX and CTS*.	R	0

Bits	Field Name	Description	Type	Reset
		Read 0x1: A falling edge occurred on RX or CTS*.		
0	TX_FIFO_FULL	Read 0x0: TX FIFO is not full.	R	0
		Read 0x1: TX FIFO is full.		

Table 18-170. Register Call Summary for Register UART_SSR

UART
- UART Interrupt Management: [0]
- FIFO Management: [1]
- UART Mode Selection: [2][3][4]
- UART Register Summary: [5]
- UART Register Description: [6]

Table 18-171. UART_MVR

Address Offset	0x0000 0050		
Physical Address	0x4806 A050 0x4806 C050 0x4802 0050	**Instance**	UART1 UART2 UART3
Description	Module version register The reset value is fixed by hardware and corresponds to the RTL revision of this module. A reset has no effect on the value returned.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REV																															

Bits	Field Name	Description	Type	Reset
31:0	REV	Revision number	R	0x-- TI internal data

Table 18-172. Register Call Summary for Register UART_MVR

UART
- UART Mode Selection: [0][1][2]
- UART Register Summary: [3]

Table 18-173. UART_SYSC

Address Offset	0x0000 0054		
Physical Address	0x4806 A054 0x4806 C054 0x4802 0054	**Instance**	UART1 UART2 UART3
Description	System configuration register The AUTOIDLE bit controls a power-saving technique to reduce the logic power consumption of the open-core protocol (OCP) interface. When the feature is enabled, the clock is gated off until an OCP command for this device is detected. When the software reset bit is set high, it causes a full device reset.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								RESERVED			IDLEMODE		ENAWAKEUP	SOFTRESET	AUTOIDLE

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Read returns 0. Write has no effect.	RW	0x000000
7:5	RESERVED	Read returns 0.	R	0x0
4:3	IDLEMODE	Power management req/ack control 0x0: Force-idle: Idle request is acknowledged unconditionally. 0x1: No-idle: Idle request is never acknowledged. 0x2: Smart-idle: Idle request is acknowledged based in module internal activity. 0x3: Smart-idle Wake-up: Acknowledgement to an idle request is given based in the internal activity of the module. The module is allowed to generate wake-up request.	RW	0x0
2	ENAWAKEUP	Wake-up feature control 0x0: Wakeup is disabled. 0x1: Wake-up capability is enabled.	RW	0
1	SOFTRESET	Software reset. Set this bit to 1 to trigger a module reset. This bit is automatically reset by the hardware. Read returns 0. 0x0: Normal mode 0x1: The module is reset.	RW	0
0	AUTOIDLE	Internal OCP clock gating strategy 0x0: Clock is running. 0x1: Automatic OCP clock gating strategy is applied, based on OCP interface activity	RW	0

Table 18-174. Register Call Summary for Register UART_SYSC

UART

- Clock Configuration: [0]
- Software Reset: [1]
- UART Mode Power Management: [2]
- Local Power Management: [3][4][5]
- UART Mode Selection: [6][7][8][9][10][11]
- UART Module Global Initialization: [12]
- UART Register Summary: [13]

Table 18-175. UART_SYSS

Address Offset	0x0000 0058		
Physical Address	0x4806 A058 0x4806 C058 0x4802 0058	**Instance**	UART1 UART2 UART3
Description	System status register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								RESERVED							RESETDONE

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Read returns 0.	R	0x000000
7:1	RESERVED	Read returns 0.	R	0x00
0	RESETDONE	Internal reset monitoring	R	0
		Read 0x0: Internal module reset is ongoing.		
		Read 0x1: Reset complete		

Table 18-176. Register Call Summary for Register UART_SYSS

UART

- UART Mode Selection: [0][1][2]
- UART Module Global Initialization: [3]
- UART Register Summary: [4]

Table 18-177. UART_WER

Address Offset	0x0000 005C		
Physical Address	0x4806 A05C 0x4806 C05C 0x4802 005C	**Instance**	UART1 UART2 UART3
Description	Wake-up enable register The UART wake-up enable register is used to mask and unmask a UART event that would subsequently notify the system. An event is any activity in the logic that could cause an interrupt and/or an activity that would require the system to wake up. Even if the wakeup is disabled for certain events, if these events are also an interrupt to the UART, the UART registers the interrupt.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								TX_WAKEUP_EN	EVENT_6_RECEIVER_LINE_STATUS_INTERRUPT	EVENT_5_RHR_INTERRUPT	EVENT_4_RX_ACTIVITY	EVENT_3_DCD_CD_ACTIVITY	EVENT_2_RI_ACTIVITY	EVENT_1_DSR_ACTIVITY	EVENT_0_CTS_ACTIVITY

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Read returns 0. Write has no effect.	RW	0x000000
7	TX_WAKEUP_EN	0x0: Event is not allowed to wake up the system. 0x1: Event can wake up the system: it can be THR_IT or TX_DMA request and/or TX_SATUS_IT.	RW	0
6	EVENT_6_RECEIVER_LINE_STATUS_INTERRUPT	0x0: Event is not allowed to wake up the system. 0x1: Event can wake up the system.	RW	1
5	EVENT_5_RHR_INTERRUPT	0x0: Event is not allowed to wake up the system. 0x1: Event can wake up the system.	RW	1
4	EVENT_4_RX_ACTIVITY	0x0: Event is not allowed to wake up the system. 0x1: Event can wake up the system.	RW	1
3	EVENT_3_DCD_CD_ACTIVITY	0x0: Event is not allowed to wake up the system 0x1: Event can wake up the system	RW	1
2	EVENT_2_RI_ACTIVITY	0x0: Event is not allowed to wake up the system. 0x1: Event can wake up the system.	RW	1
1	EVENT_1_DSR_ACTIVITY	0x0: Event is not allowed to wake up the system. 0x1: Event can wake up the system.	RW	1
0	EVENT_0_CTS_ACTIVITY	0x0: Event is not allowed to wake up the system. 0x1: Event can wake up the system.	RW	1

Table 18-178. Register Call Summary for Register UART_WER

UART
- UART Mode Power Management: [0]
- UART Mode Selection: [1][2][3][4][5][6]
- UART Register Summary: [7]

Table 18-179. UART_RXFIFO_LVL

Address Offset	0x0000 0064		
Physical Address	0x4806 A064 0x4806 C064 0x4802 0064	**Instance**	UART1 UART2 UART3
Description	Level of the RX FIFO		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								RXFIFO_LVL							

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Read returns 0.	R	0x000000
7:0	RXFIFO_LVL	Shows the number of received bytes in the RX FIFO	R	0x00

Table 18-180. Register Call Summary for Register UART_RXFIFO_LVL

UART
- UART Mode Selection: [0][1][2][3][4][5]
- UART Register Summary: [6]

Table 18-181. UART_TXFIFO_LVL

Address Offset	0x0000 0068		
Physical Address	0x4806 A068 0x4806 C068 0x4802 0068	**Instance**	UART1 UART2 UART3
Description	Level of the TX FIFO		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								TXFIFO_LVL							

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Read returns 0.	R	0x000000
7:0	TXFIFO_LVL	Shows the number of written bytes in the TX FIFO	R	0x00

Table 18-182. Register Call Summary for Register UART_TXFIFO_LVL

UART
- UART Mode Selection: [0][1][2][3][4][5]
- UART Register Summary: [6]

Table 18-183. UART_IER2

Address Offset	0x0000 006C		
Physical Address	0x4806 A06C 0x4806 C06C 0x4802 006C	**Instance**	UART1 UART2 UART3
Description	Enables RX/TX FIFOs empty corresponding interrupts		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														EN_TXFIFO_EMPTY	EN_RXFIFO_EMPTY

Bits	Field Name	Description	Type	Reset
31:2	RESERVED	Read returns 0. Write has no effect.	RW	0x0000 0000
1	EN_TXFIFO_EMPTY	Enables TX FIFO empty corresponding interrupt 0x0: Enables EN_TXFIFO_EMPTY interrupt 0x1: Disables EN_TXFIFO_EMPTY interrupt	RW	0
0	EN_RXFIFO_EMPTY	Enables RX FIFO empty corresponding interrupt 0x0: Enables EN_RXFIFO_EMPTY interrupt 0x1: Disables EN_RXFIFO_EMPTY interrupt	RW	0

Table 18-184. Register Call Summary for Register UART_IER2

UART
- UART Mode Selection: [0][1][2][3][4][5]
- UART Register Summary: [6]

Table 18-185. UART_ISR2

Address Offset	0x0000 0070		
Physical Address	0x4806 A070 0x4806 C070 0x4802 0070	**Instance**	UART1 UART2 UART3
Description	Status of RX/TX FIFOs empty corresponding interrupts		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														TXFIFO_EMPTY_STS	RXFIFO_EMPTY_STS

Bits	Field Name	Description	Type	Reset
31:2	RESERVED	Read returns 0. Write has no effect.	RW	0x0000 0000
1	TXFIFO_EMPTY_STS	Used to generate interrupt if the TX_FIFO is empty (software flow control) 0x0: TXFIFO_EMPTY interrupt not pending. 0x1: TXFIFO_EMPTY interrupt pending.	RW	1
0	RXFIFO_EMPTY_STS	Used to generate interrupt if the RX_FIFO is empty (software flow control) 0x0: RXFIFO_EMPTY interrupt not pending. 0x1: RXFIFO_EMPTY interrupt pending.	RW	1

Table 18-186. Register Call Summary for Register UART_ISR2

UART
- UART Mode Selection: [0][1][2][3][4][5]
- UART Register Summary: [6]

Table 18-187. UART_FREQ_SEL

Address Offset	0x0000 0074		
Physical Address	0x4806 A074 0x4806 C074 0x4802 0074	**Instance**	UART1 UART2 UART3
Description	Sample per bit selector		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								FREQ_SEL							

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Read returns 0. Write has no effect.	RW	0x0000 0000
7:0	FREQ_SEL	Sets the sample per bit if nondefault frequency is used. UART_MDR3[1] must be set to 1 after this value is set. Must be equal to or higher then 6.	RW	0x1A

Table 18-188. Register Call Summary for Register UART_FREQ_SEL

UART
- UART Mode Selection: [0][1][2][3][4][5]
- UART Register Summary: [6]
- UART Register Description: [7]

Table 18-189. UART_MDR3

Address Offset	0x0000 0080		
Physical Address	0x4806 A080 0x4806 C080 0x4802 0080	**Instance**	UART1 UART2 UART3
Description	Mode definition register 3		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																													SET_DMA_TX_THRESHOLD	NONDEFAULT_FREQ	DISABLE_CIR_RX_DEMOD

Bits	Field Name	Description	Type	Reset
31:3	RESERVED	Read returns 0. Write has no effect.	RW	0x0000 0000
2	SET_DMA_TX_THRESHOLD	Enable to set different TXDMA threshold in UART_TX_DMA_THRESHOLD register.	RW	0
1	NONDEFAULT_FREQ	Used to enable the NONDEFAULT fclk frequencies. 0x0: Disables using NONDEFAULT fclk frequencies. 0x1: Enables using NONDEFAULT fclk frequencies (set UART_FREQ_SEL and UART_DLH / UART_DLL).	RW	0
0	DISABLE_CIR_RX_DEMOD	Used to enable CIR RX demodulation. CIR IS NOT SUPPORTED 0x0: Enables CIR RX demodulation. 0x1: Disables CIR RX demodulation.	RW	0

Table 18-190. Register Call Summary for Register UART_MDR3

UART

- FIFO DMA Mode Operation: [0][1][2]
- UART Mode Selection: [3][4][5][6][7][8]
- UART Register Summary: [9]
- UART Register Description: [10][11]

Table 18-191. UART_TX_DMA_THRESHOLD

Address Offset	0x0000 0084		
Physical Address	0x4806 A084 0x4806 C084 0x4802 0084	**Instance**	UART1 UART2 UART3
Description	Use to manually set the TX DMA threshold level. UART_MDR3[2] SET_TX_DMA_THRESHOLD must be 1 and must be value + tx_trigger_level = 64 (TX FIFO size). If not, 64-tx_trigger_level will be used without modifying the value of this register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								TX_DMA_THRESHOLD							

Bits	Field Name	Description	Type	Reset
31:6	RESERVED	Reserved	RW	0x0000000
5:0	TX_DMA_THRESHOLD	Used to manually set the TX DMA threshold level	RW	0x00

Table 18-192. Register Call Summary for Register UART_TX_DMA_THRESHOLD

UART

- FIFO DMA Mode Operation: [0][1]
- UART Mode Selection: [2][3][4][5][6][7]
- UART Register Summary: [8]
- UART Register Description: [9]

18.3 Multichannel Serial Peripheral Interface

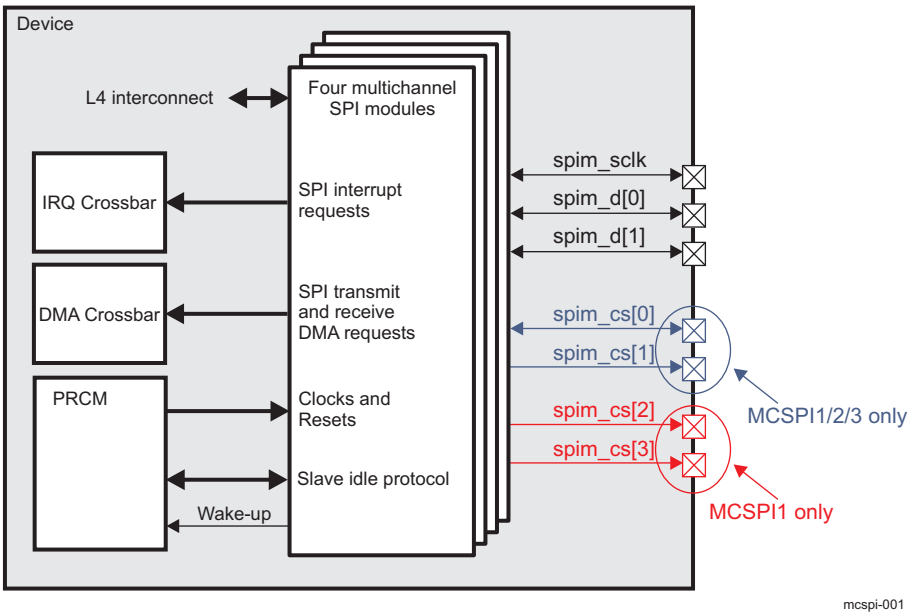
This section describes the four Multichannel Serial Peripheral Interface (McSPI) modules for the device.

18.3.1 McSPI Overview

The SPI is a master/slave synchronous serial bus. There are four separate McSPI modules (McSPI1, McSPI2, McSPI3, and McSPI4) in the device (see Figure 18-42). All these four modules are able to work as both master and slave and support the following chip selects:

- McSPI1: spi1_cs[0], spi1_cs[1], spi1_cs[2], spi1_cs[3]
- McSPI2: spi2_cs[0], spi2_cs[1]
- McSPI3: spi3_cs[0], spi3_cs[1]
- McSPI4: spi4_cs[0]

Figure 18-42. Multichannel SPI Modules



The McSPI modules include the following main features:

- Serial clock with programmable frequency, polarity, and phase for each channel
- Wide selection of SPI word lengths, ranging from 4 to 32 bits
- Up to four master channels, or single channel in slave mode
- Master multichannel mode:
  - Full duplex/half duplex
  - Transmit-only/receive-only/transmit-and-receive modes
  - Flexible input/output (I/O) port controls per channel
  - Programmable clock granularity
  - SPI configuration per channel. This means, clock definition, polarity enabling and word width
- Single interrupt line for multiple interrupt source events
- Power management through wake-up capabilities
- Enable the addition of a programmable start-bit for SPI transfer per channel (start-bit mode)
- Supports start-bit write command
- Supports start-bit pause and break sequence
- Programmable timing control between chip select and external clock generation
- Built-in FIFO available for a single channel

18.3.2 McSPI Environment

18.3.2.1 Basic McSPI Pins for Master Mode

Figure 18-43 shows all of the McSPI interface signals in master mode.

Figure 18-43. McSPI Interface Signals in Master Mode

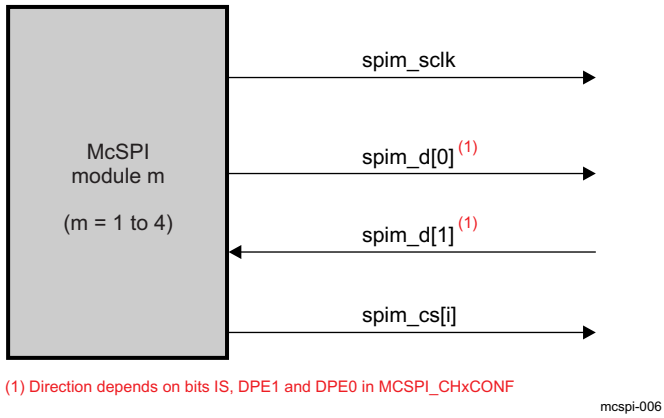


Table 18-193 describes the McSPI I/O in master mode.

Table 18-193. McSPI I/O Description (Master Mode)

Device-Level Signal Name	Module Signal Name	I/O(1)	Description
spim_sclk	SPICLK	O	SPIm module serial clock output
spim_d[0]	SPIDAT[0]	O(2)	SPI Data I/O. Can be configured either as input or as output depending on MCSPI_CHxCONF[18] IS and MCSPI_CHxCONF[16] DPE0.
spim_d[1]	SPIDAT[1]	I(2)	SPI Data I/O. Can be configured either as input or as output depending on MCSPI_CHxCONF[18] IS and MCSPI_CHxCONF[17] DPE1.
spim_cs[i]	SPIEN[x]	O	SPIm module chip-select i output

(1) I = Input; O = Output
(2) Example configuration only.

NOTE: For the spim_sclk signals to work properly, the INPUTENABLE bit of the appropriate CTRL_CORE_PAD_x registers should be set to 0x1 because of retiming purposes.

18.3.2.2 Basic McSPI Pins for Slave Mode

Figure 18-44 shows all of the McSPI interface signals in slave mode.

Figure 18-44. McSPI Interface Signals in Slave Mode

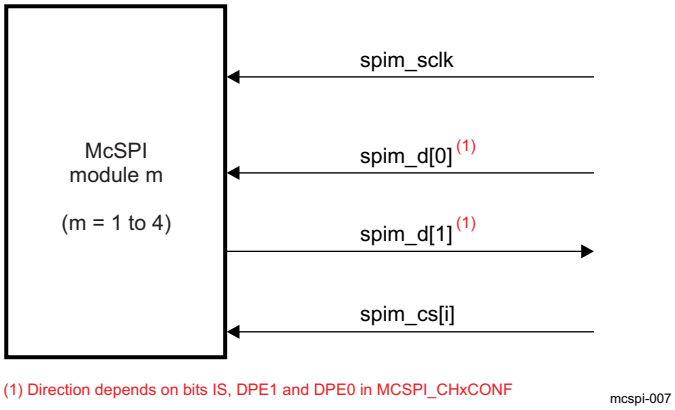


Table 18-194 describes the McSPI I/O in slave mode.

Table 18-194. McSPI I/O Description (Slave Mode)

Device-Level Signal Name	Module Signal Name	I/O(1)	Description
spim_sclk	SPICLK	I	McSPIm module serial clock input
spim_d[0]	SPIDAT[0]	I(2)	SPI Data I/O. Can be configured either as input or as output depending on MCSPI_CHxCONF[18] IS and MCSPI_CHxCONF[16] DPE0.
spim_d[1]	SPIDAT[1]	O(2)	SPI Data I/O. Can be configured either as input or as output depending on MCSPI_CHxCONF[18] IS and MCSPI_CHxCONF[17] DPE1.
spim_cs[i]	SPIEN[x]	I	McSPIm module chip-select i input. Can be selected through MCSPI_CH0CONF[22:21] SPIENSLV bit field

(1) I = Input; O = Output
(2) Example configuration only.

18.3.2.3 Multichannel SPI Protocol and Data Format

The synchronous SPI protocol allows a master device to initiate serial data transfers to a slave device. A slave select line (SPIEN[x]) allows selection of an individual slave SPI device. Slave devices that are not selected do not interfere with SPI bus activities.

McSPI offers the flexibility to modify the following parameters to adapt to the device features:

- Word length

  McSPI supports any SPI word ranging from 4 bits to 32 bits long (the MCSPI_CHxCONF[11:7] WL bit field).

  SPI word length can be changed between transmissions to allow the master device to communicate with peripheral slaves that have different requirements.

- SPI enable (SPIEN[x], for channel x of instance m)

  The polarity of the SPI enable signals is programmable (the MCSPI_CHxCONF[6] EPOL bit). SPIEN[x] signals can be active high or low.

  Assertion of the SPIEN[x] signals is programmable and can be done manually or automatically. The manual assertion mode is available in single master mode only. SPIEN[x] can be kept active between words with the MCSPI_CHxCONF[20] FORCE bit.

  Two consecutive words for two different slave devices can go along with active SPIEN[x] signals with different polarity.

- Programmable start-bit

  In start-bit mode a start-bit is added before the SPI word length to indicate how the next SPI word must be handled. The start-bit is enabled by setting the MCSPI_CHxCONF[23] SBE bit to 1. The MCSPI_CHxCONF[24] SBPOL bit defines the polarity of the start-bit.

- Programmable SPI clock
  - Bit rate

    In master mode, the baud rate of the SPI serial clock is programmable using the 48-MHz reference clock (from the power, reset, and clock managment [PRCM] module). Table 18-195 lists the SPICLK bit rates obtained for data transfer when programming the clock divider (the MCSPI_CHxCONF[5:2] CLKD bit field).

Table 18-195. SPI Master Clock Rates

Divider	Clock Rate
1	48 MHz(1)
2	24 MHz(1)
4	12 MHz
8	6 MHz
16	3 MHz
32	1.5 MHz
64	750 kHz
128	375 kHz
256	~187 kHz
512	~93.7 kHz
1024	~46.8 kHz
2048	~23.4 kHz
4096	~11.7 kHz

(1) These frequencies are not necessarily supported by all SPI modules. For more information, see the *Timing Requirements and Switching Characteristics* chapter in the device data manual.

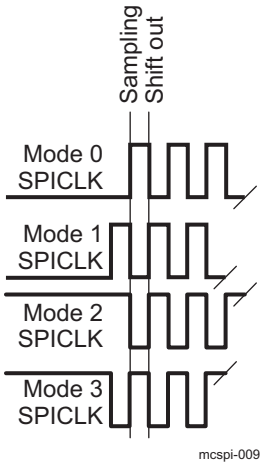
  - Polarity and phase

    The polarity (the MCSPI_CHxCONF[1] POL bit) and the phase (the MCSPI_CHxCONF[0] PHA bit) of the SPI serial clock (SPICLK) are configurable to offer four combinations. Software selects the right combination, depending on the device. See Table 18-196 and Figure 18-45.

Table 18-196. Phase and Polarity Combinations

Polarity (POL)	Phase (PHA)	SPI Mode	Description
0	0	Mode 0	SPICLK is inactive low and sampling occurs at the rising edge.
0	1	Mode 1	SPICLK is inactive low and sampling occurs at the falling edge.
1	0	Mode 2	SPICLK is inactive high and sampling occurs at the falling edge.
1	1	Mode 3	SPICLK is inactive high and sampling occurs at the rising edge.

Figure 18-45. Phase and Polarity Combinations



18.3.2.3.1 Transfer Format

In master and slave modes, the McSPI drives the data lines when SPIEN[x] is asserted.

Each word is transmitted starting with the most-significant bit (MSB).

This section explains the two cases of data transmission determined by the clock phase (PHA) and the type of data transmission using a start-bit (SBE) called the start-bit mode:

- Transmission in mode 0 and mode 2 (PHA = 0)

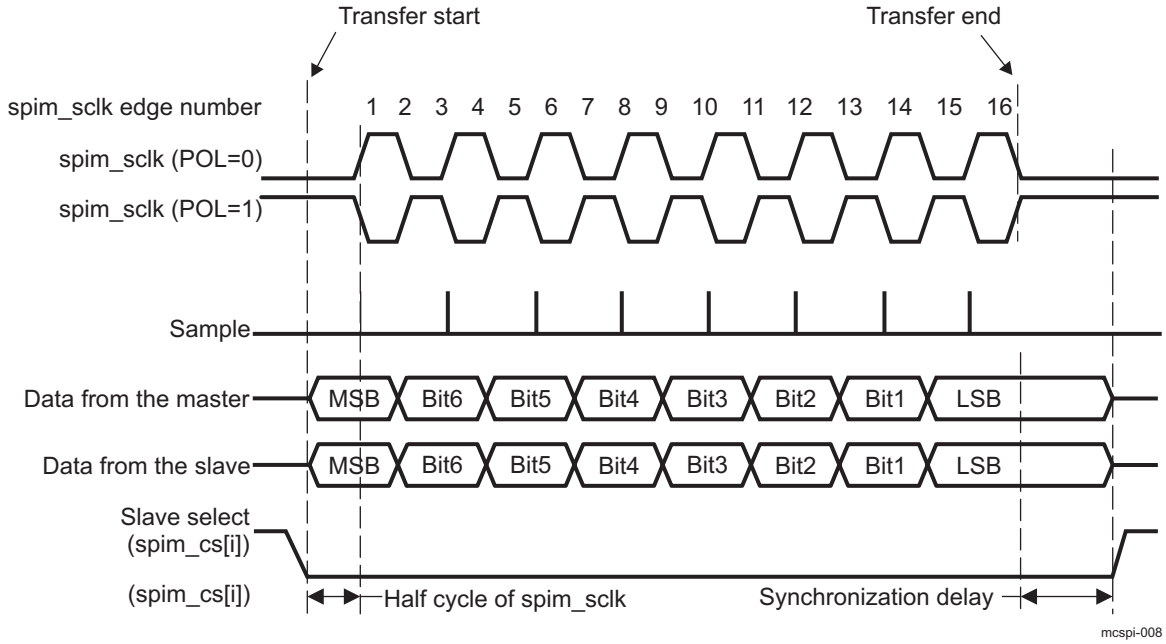
  When PHA = 0, the first bit of the SPI word to transmit (on the master or the slave data output pin) is valid one-half cycle of SPICLK after the assertion of SPIEN.

  Therefore, the first edge of the SPICLK line is used by the master to sample the first data bit sent by the slave. On the same edge, the first data bit sent by the master is sampled by the slave.

  On the next SPICLK edge, the received data bit is shifted into the receive shift register and a new data bit is transmitted on the serial data line.

  This process continues for a number of pulses on the SPICLK line defined by the SPI word length programmed in the master device, with data being latched on odd-numbered edges and shifted on even-numbered edges. See Figure 18-46.

Figure 18-46. Full-Duplex Transfer Format With PHA = 0



- Transmission in mode 1 and mode 3 (PHA = 1)

  When PHA = 1, the first bit of the SPI word to transmit (on the master or the slave data output pin) is valid on the following SPICLK edge (one-half cycle later). This is the sampling edge for the master and slave. A synchronization delay is added between the activation of SPIEN[x] and the first SPICLK edge.

  The received data bit is shifted into the shift register on the third SPICLK edge.

  This process continues for a number of pulses on the SPICLK line defined by the SPI word length programmed in the master device, with data being latched on even-numbered edges and shifted on odd-numbered edges.

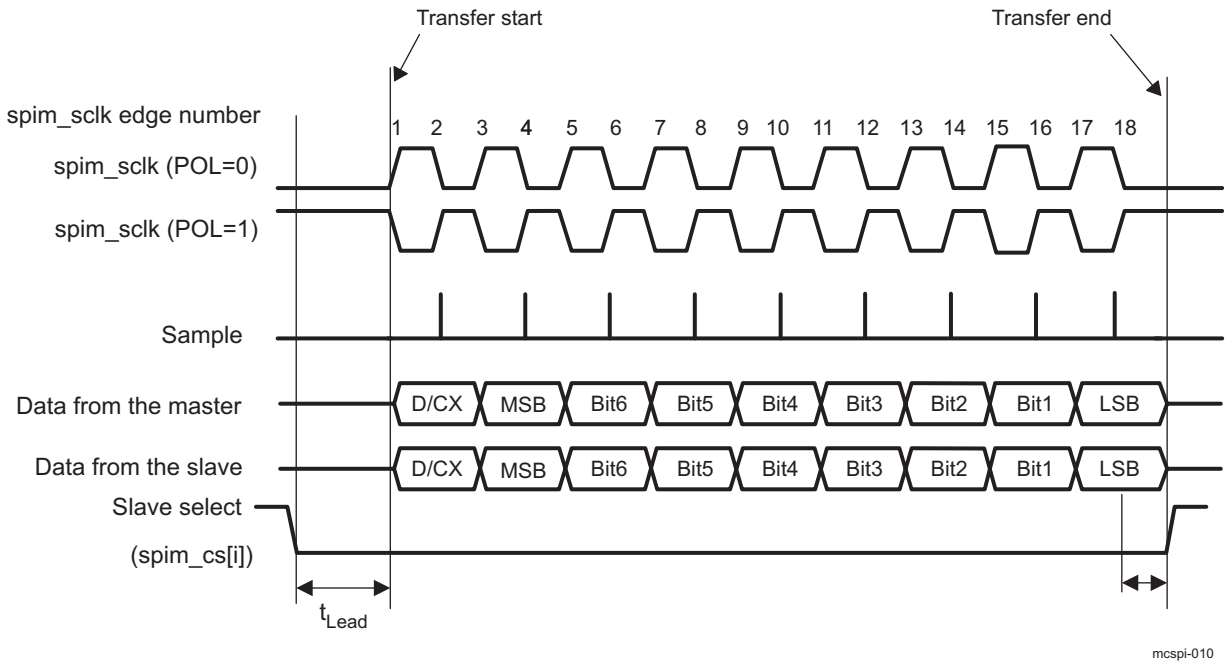
  NOTE: The minimum synchronization delay is one cycle of SPICLK, if the frequency of SPICLK equals the frequency of SPIm_FCLK (McSPIm functional clock) in master mode. The minimum synchronization delay is one-half cycle of SPICLK, if the frequency of SPICLK is lower than the frequency of SPIm_FCLK in the master and slave modes.

- Transmission with a start-bit (SBE = 1)

  When the MCSPI_CHxCONF[23] SBE bit is set to 1, a start-bit is added before the MSB to indicate whether the next SPI word must be handled as a command or as data.

  Figure 18-47 shows an example of a data transfer with an extra start-bit.

Figure 18-47. Extended SPI Transfer With a Start-Bit (SBE = 1)



18.3.2.4 SPI in Master Mode

Figure 18-48 shows a case in master mode (full-duplex) where the McSPI module is connected with two slave devices.

Figure 18-48. McSPI Master Mode (Full Duplex)

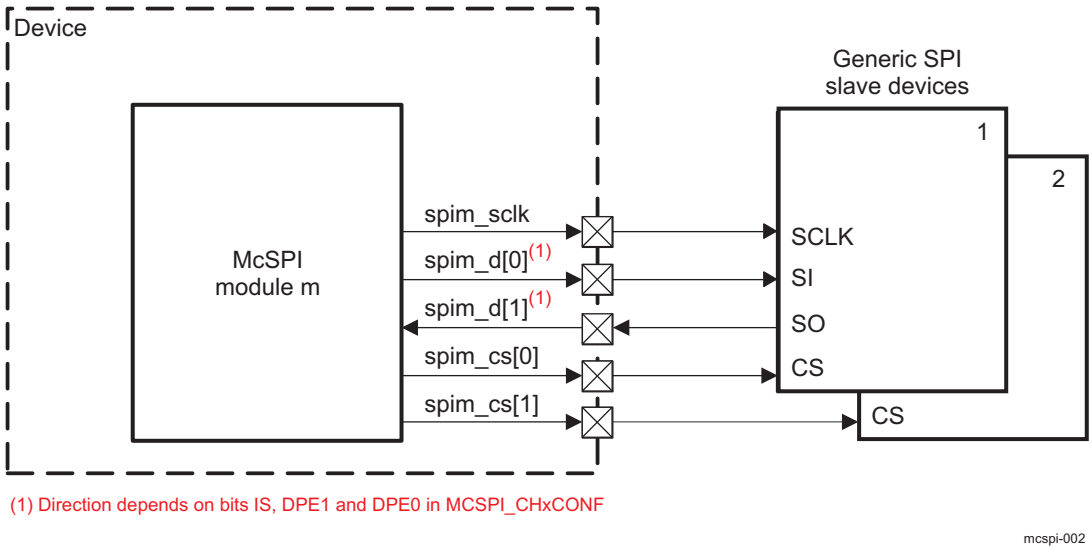
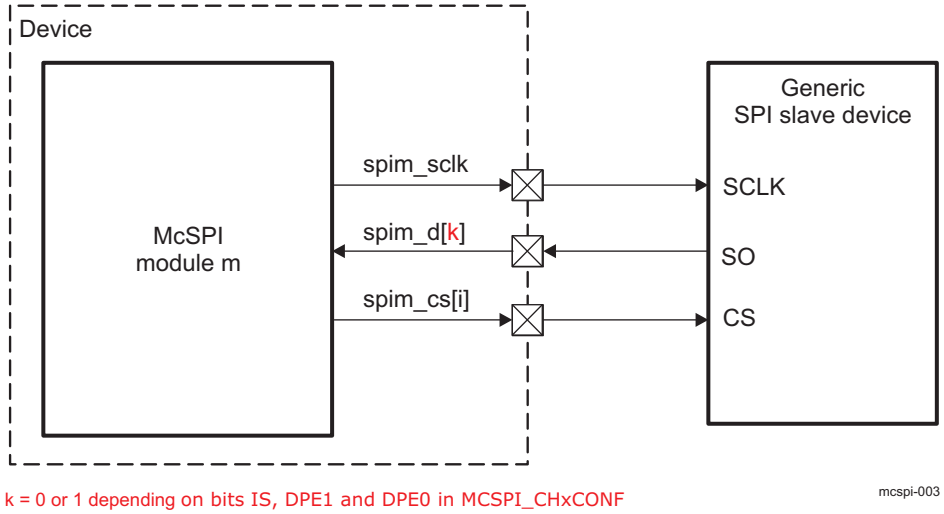


Figure 18-49 shows the master single mode, which can also be configured in receive-only mode.

Figure 18-49. McSPI Master Single Mode (Receive Only)



18.3.2.5 SPI in Slave Mode

Figure 18-50 shows a case in slave mode (full-duplex).

NOTE: Only channel 0 can be configured as slave, but the chip-enable signal can be connected to any SPIEN[x] pin and then rerouted internally to channel 0 (the MCSPI_CHxCONF[22:21] SPIENSLV bit field [where x = 0]). For more information, see Section 18.3.4.4, *Slave Mode*.

Figure 18-50. McSPI Slave Mode (Full Duplex)

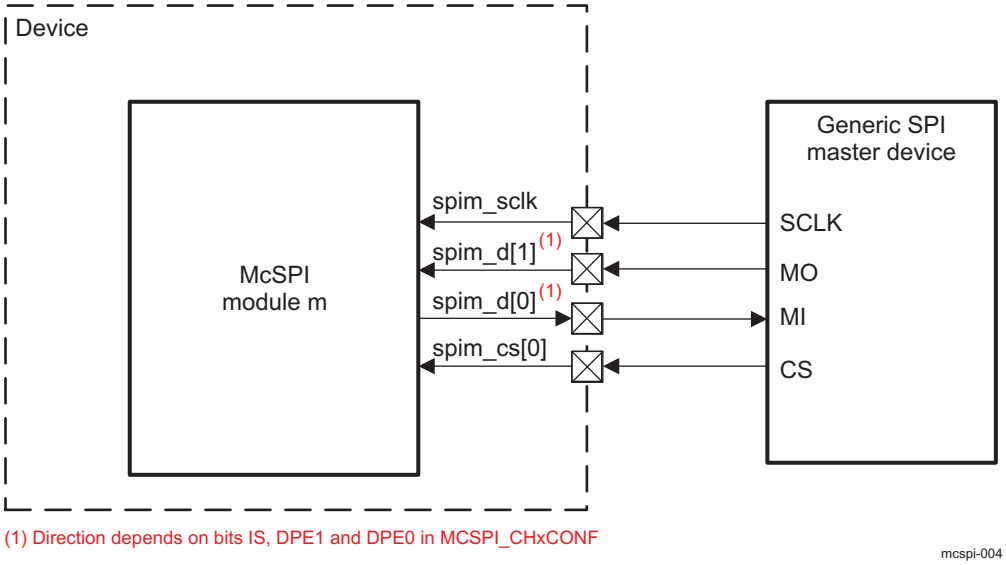
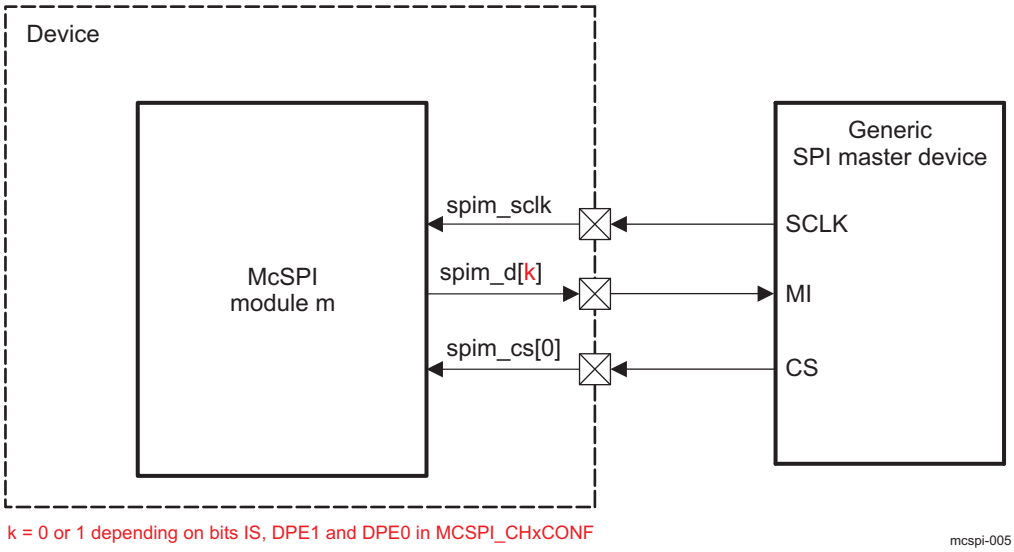


Figure 18-51 shows the slave single mode, which can also be configured in transmit-only mode.

Figure 18-51. McSPI Slave Single Mode (Transmit Only)



18.3.3 McSPI Integration

This section describes module integration in the device, including information about clocks, resets, and hardware requests.

Figure 18-52 shows McSPI integration.

Figure 18-52. McSPI Integration

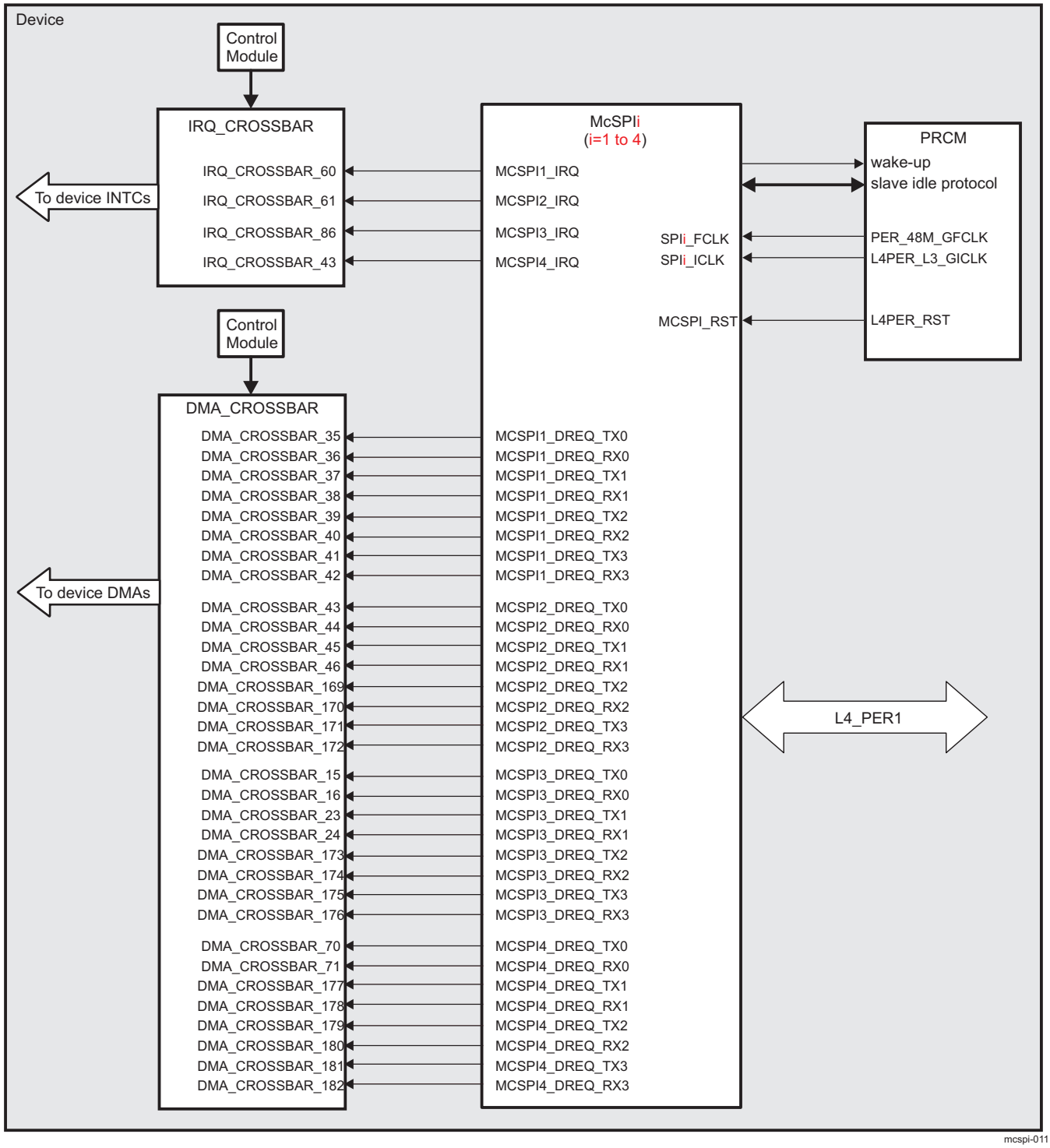


Table 18-197 through Table 18-199 summarize the integration of the module in the device.

Table 18-197. McSPI Integration Attributes

Module Instance	Attributes	
	Power Domain	Interconnect
McSPI1	PD_COREAON	L4_PER1
McSPI2	PD_COREAON	L4_PER1
McSPI3	PD_COREAON	L4_PER1
McSPI4	PD_COREAON	L4_PER1

Table 18-198. McSPI Clocks and Resets

Clocks				
Module Instance	**Destination Signal Name**	**Source Signal Name**	**Source**	**Description**
McSPI1	SPI1_ICLK	L4PER_L3_GICLK	PRCM	Interface clock
	SPI1_FCLK	PER_48M_GFCLK	PRCM	Functional clock
McSPI2	SPI2_ICLK	L4PER_L3_GICLK	PRCM	Interface clock
	SPI2_FCLK	PER_48M_GFCLK	PRCM	Functional clock
McSPI3	SPI3_ICLK	L4PER_L3_GICLK	PRCM	Interface clock
	SPI3_FCLK	PER_48M_GFCLK	PRCM	Functional clock
McSPI4	SPI4_ICLK	L4PER_L3_GICLK	PRCM	Interface clock
	SPI4_FCLK	PER_48M_GFCLK	PRCM	Functional clock
Resets				
Module Instance	**Destination Signal Name**	**Source Signal Name**	**Source**	**Description**
McSPI1	MCSPI_RST	L4PER_RST	PRCM	McSPI1 reset signal
McSPI2	MCSPI_RST	L4PER_RST	PRCM	McSPI2 reset signal
McSPI3	MCSPI_RST	L4PER_RST	PRCM	McSPI3 reset signal
McSPI4	MCSPI_RST	L4PER_RST	PRCM	McSPI4 reset signal

Table 18-199. McSPI Hardware Requests

Interrupt Requests				
Module Instance	**Source Signal Name**	**Destination IRQ_CROSSBAR Input**	**Default Mapping**	**Description**
McSPI1	MCSPI1_IRQ	IRQ_CROSSBAR_60	DSP1_IRQ_91 DSP2_IRQ_91 IPU_IRQ_57	McSPI module 1 interrupt request
McSPI2	MCSPI2_IRQ	IRQ_CROSSBAR_61	DSP1_IRQ_92 DSP2_IRQ_92 IPU_IRQ_58	McSPI module 2 interrupt request
McSPI3	MCSPI3_IRQ	IRQ_CROSSBAR_86	-	McSPI module 3 interrupt request. This IRQ source signal is not mapped by default to any device INTC.
McSPI4	MCSPI4_IRQ	IRQ_CROSSBAR_43	DSP1_IRQ_74 DSP2_IRQ_74	McSPI module 4 interrupt request
DMA Requests				
Module Instance	**Source Signal Name**	**Destination DMA_CROSSBAR Input**	**Default Mapping**	**Description**

Table 18-199. McSPI Hardware Requests (continued)

McSPI1	MCSPI1_DREQ_TX0	DMA_CROSSBAR_35	DMA_EDMA_DREQ_34	McSPI module 1 - transmit request channel 0
	MCSPI1_DREQ_RX0	DMA_CROSSBAR_36	DMA_EDMA_DREQ_35	McSPI module 1 - receive request channel 0
	MCSPI1_DREQ_TX1	DMA_CROSSBAR_37	DMA_EDMA_DREQ_36	McSPI module 1 - transmit request channel 1
	MCSPI1_DREQ_RX1	DMA_CROSSBAR_38	DMA_EDMA_DREQ_37	McSPI module 1 - receive request channel 1
	MCSPI1_DREQ_TX2	DMA_CROSSBAR_39	DMA_EDMA_DREQ_38	McSPI module 1 - transmit request channel 2
	MCSPI1_DREQ_RX2	DMA_CROSSBAR_40	DMA_EDMA_DREQ_39	McSPI module 1 - receive request channel 2
	MCSPI1_DREQ_TX3	DMA_CROSSBAR_41	DMA_EDMA_DREQ_40	McSPI module 1 - transmit request channel 3
	MCSPI1_DREQ_RX3	DMA_CROSSBAR_42	DMA_EDMA_DREQ_41	McSPI module 1 - receive request channel 3
McSPI2	MCSPI2_DREQ_TX0	DMA_CROSSBAR_43	DMA_EDMA_DREQ_42	McSPI module 2 - transmit request channel 0
	MCSPI2_DREQ_RX0	DMA_CROSSBAR_44	DMA_EDMA_DREQ_43	McSPI module 2 - receive request channel 0
	MCSPI2_DREQ_TX1	DMA_CROSSBAR_45	DMA_EDMA_DREQ_44	McSPI module 2 - transmit request channel 1
	MCSPI2_DREQ_RX1	DMA_CROSSBAR_46	DMA_EDMA_DREQ_45	McSPI module 2 - receive request channel 1
	MCSPI2_DREQ_TX2	DMA_CROSSBAR_169	-	McSPI module 2 - transmit request channel 2
	MCSPI2_DREQ_RX2	DMA_CROSSBAR_170	-	McSPI module 2 - receive request channel 2
	MCSPI2_DREQ_TX3	DMA_CROSSBAR_171	-	McSPI module 2 - transmit request channel 3
	MCSPI2_DREQ_RX3	DMA_CROSSBAR_172	-	McSPI module 2 - receive request channel 3
McSPI3	MCSPI3_DREQ_TX0	DMA_CROSSBAR_15	DMA_EDMA_DREQ_14	McSPI module 3 - transmit request channel 0
	MCSPI3_DREQ_RX0	DMA_CROSSBAR_16	DMA_EDMA_DREQ_15	McSPI module 3 - receive request channel 0
	MCSPI3_DREQ_TX1	DMA_CROSSBAR_23	DMA_EDMA_DREQ_22	McSPI module 3 - transmit request channel 1
	MCSPI3_DREQ_RX1	DMA_CROSSBAR_24	DMA_EDMA_DREQ_23	McSPI module 3 - receive request channel 1
	MCSPI3_DREQ_TX2	DMA_CROSSBAR_173	-	McSPI module 3 - transmit request channel 2
	MCSPI3_DREQ_RX2	DMA_CROSSBAR_174	-	McSPI module 3 - receive request channel 2
	MCSPI3_DREQ_TX3	DMA_CROSSBAR_175	-	McSPI module 3 - transmit request channel 3
	MCSPI3_DREQ_RX3	DMA_CROSSBAR_176	-	McSPI module 3 - receive request channel 3

Table 18-199. McSPI Hardware Requests (continued)

McSPI4	MCSPI4_DREQ_TX0	DMA_CROSSBAR_70	-	McSPI module 4 - transmit request channel 0
	MCSPI4_DREQ_RX0	DMA_CROSSBAR_71	-	McSPI module 4 - receive request channel 0
	MCSPI4_DREQ_TX1	DMA_CROSSBAR_177	-	McSPI module 4 - transmit request channel 1
	MCSPI4_DREQ_RX1	DMA_CROSSBAR_178	-	McSPI module 4 - receive request channel 1
	MCSPI4_DREQ_TX2	DMA_CROSSBAR_179	-	McSPI module 4 - transmit request channel 2
	MCSPI4_DREQ_RX2	DMA_CROSSBAR_180	-	McSPI module 4 - receive request channel 2
	MCSPI4_DREQ_TX3	DMA_CROSSBAR_181	-	McSPI module 4 - transmit request channel 3
	MCSPI4_DREQ_RX3	DMA_CROSSBAR_182	-	McSPI module 4 - receive request channel 3

NOTE: The Default Mapping column in Table 18-199 *McSPI Hardware Requests* shows the default mapping of module IRQ and DREQ source signals. These module IRQ and DREQ source signals can also be mapped to other lines of each device Interrupt or DMA controller through the IRQ_CROSSBAR and DMA_CROSSBAR modules, respectively.
For more information about the IRQ_CROSSBAR module, see Section 13.4.6.4, *IRQ_CROSSBAR Module Functional Description*, in Chapter 13, *Control Module*.
For more information about the DMA_CROSSBAR module, see Section 13.4.6.5, *DMA_CROSSBAR Module Functional Description*, in Chapter 13, *Control Module*.

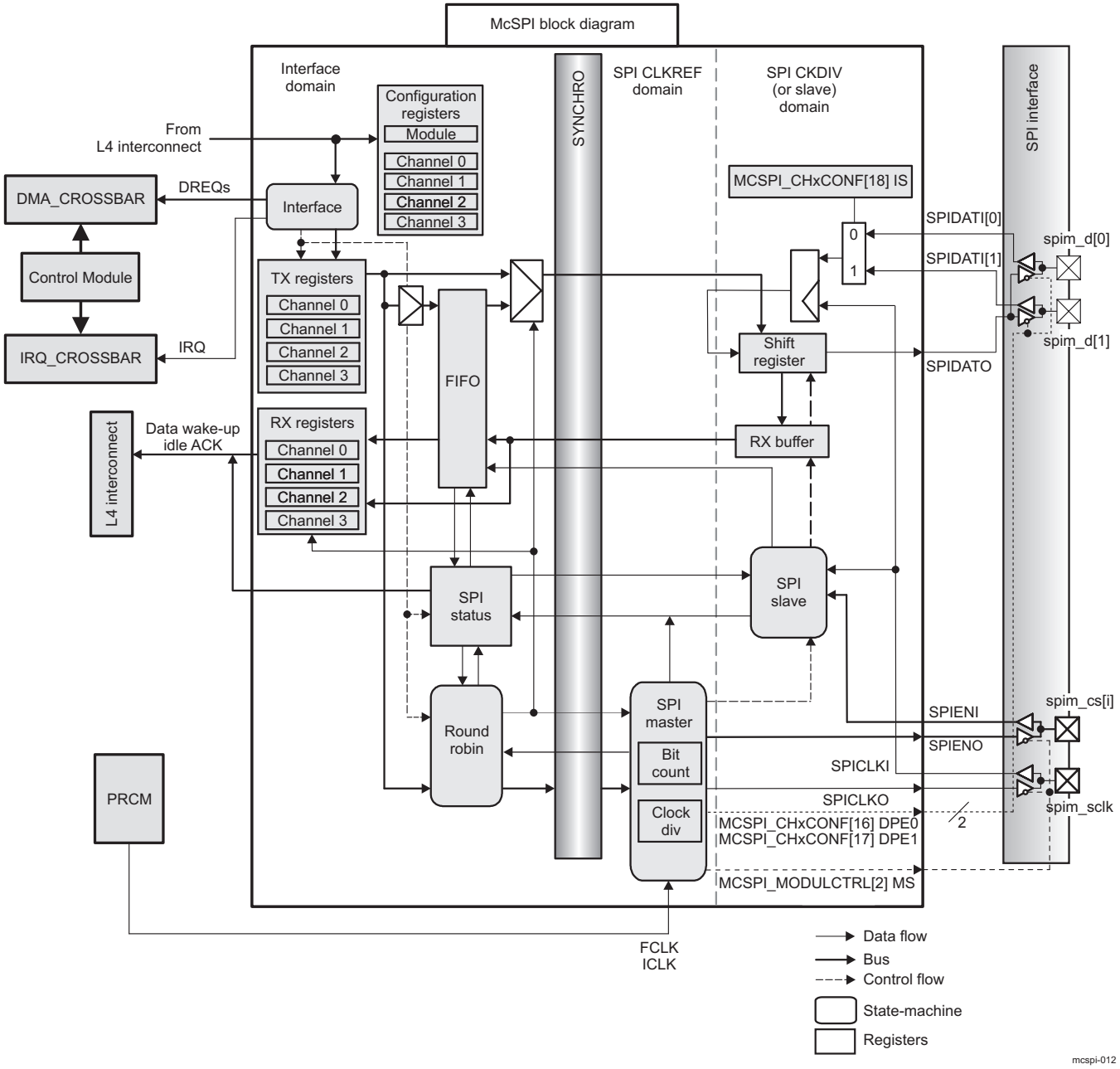
NOTE: For more information about the device interrupt controllers, see Chapter 12, *Interrupt Controllers*.
For more information about the device EDMA module, see Chapter 11, *Enhanced DMA*.

18.3.4 McSPI Functional Description

18.3.4.1 McSPI Block Diagram

Figure 18-53 shows the McSPI module.

Figure 18-53. McSPI Block Diagram



18.3.4.2 Reset

The McSPI module can be reset either by hardware or by software reset. All configuration registers and all state machines are reset by the hardware reset signal (MCSPI_RST). McSPI can be reset by software through the MCSPI_SYSCONFIG[1] SOFTRESET bit. This bit has the same impact on the module as the hardware reset signal. The only exception is that the MCSPI_SYSCONFIG register is not affected by that software reset.

18.3.4.3 Master Mode

18.3.4.3.1 Master Mode Features

The McSPI master mode supports multichannel communication with up to four independent SPI communication channel contexts. The McSPI initiates a data transfer on the data lines (SPIDAT[0] and SPIDAT[1]) and generates clock (SPICLK) and control (SPIEN) signals.

Connected to multiple external devices, the McSPI exchanges data with one SPI device at a time through two main modes (available in slave mode):

- Two-data-pins interface mode (transmit-and-receive mode for full-duplex transmission)
- Single-data-pin interface mode (recommended for half-duplex transmission)

NOTE: There is a fixed chip select line allocation in multichannel master mode. Channel x SPIEN[x] is mapped to spim_cs[i] pin.

Two DMA request events (read and write) allow synchronized accesses of the DMA controller with the activity of McSPI.

Three interrupt events can be used for data transmission and reception in master mode (for more information about interrupts, see Section 18.3.4.7.1, *Interrupt Events in Master Mode*).

18.3.4.3.2 Master Transmit-and-Receive Mode (Full Duplex)

In full-duplex transmission, data is transmitted (shifted out serially on SPIDAT[0]) and received (shifted in serially on SPIDAT[1]) simultaneously on separate data lines.

The master transmit-and-receive mode is programmable per channel (the MCSPI_CHxCONF[13:12] TRM bit field).

Channel access to the shift registers for transmission/reception is based on the MCSPI_TXx transmitter register state, the MCSPI_RXx receiver register state, and round-robin arbitration.

Channels that meet the following rules are included in the round-robin list of active channels scheduled for transmission and/or reception. The arbiter skips channels that do not meet the rules and searches in the rotation for the next enabled channel.

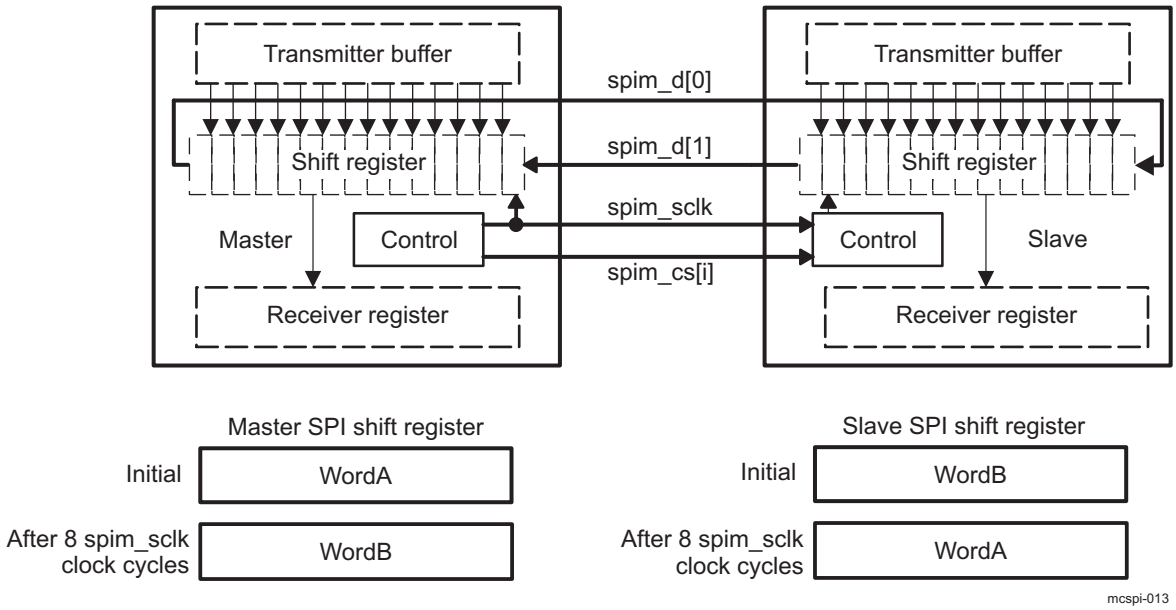
- **Rule 1:** Only enabled channels (the MCSPI_CHxCTRL[0] EN bit) can be scheduled for transmission and/or reception.
- **Rule 2:** If its MCSPI_TXx transmitter register is not empty (the MCSPI_CHxSTAT[1] TXS bit), an enabled channel can be scheduled when the shift register is assigned. If the MCSPI_TXx register is empty when the shift register is assigned, the TXx_UNDERFLOW event is activated, and the next enabled channel with new data to transmit is scheduled (see also transmit-only mode).
- **Rule 3:** An enabled channel can be scheduled if its receive register is not full (the MCSPI_CHxSTAT[0] RXS bit) when the shift register is assigned (see also receive-only mode). Therefore, the MCSPI_RXx register cannot be overwritten. The SPI1.MCSPI_IRQSTATUS[3] RX0_OVERFLOW bit is never set to this mode.

When SPI word transfer completes (the MCSPI_CHxSTAT[2] EOT bit is set), the updated MCSPI_TXx register of the next scheduled channel is loaded into the shift register. The serialization (transmit-and-receive) starts depending on the channel communication configuration. When serialization completes, the received data transfers to the channel receive register.

The serial clock (SPICLK) synchronizes shifting and sampling of the information on the two serial data lines (SPIDAT[0] and SPIDAT[1]). Each time a bit transfers out from the master, 1 bit transfers in from the slave.

Figure 18-54 shows an example of a full-duplex system with a master device (McSPI module *m*) on the left and a slave device on the right. After eight cycles of the serial clock SPICLK, WordA transfers from the master to the slave. At the same time, WordB transfers from the slave to the master.

Figure 18-54. SPI Full-Duplex Transmission (Example)



18.3.4.3.3 Master Transmit-Only Mode (Half Duplex)

The master transmit-only mode prevents a device initiator from reading the MCSPI_RXx register (minimizing data movement) when only transmission is meaningful.

The master transmit-only mode is programmable per channel (the MCSPI_CHxCONF[13:12] TRM bit field). Transmission starts only after data is loaded into the MCSPI_TXx register.

Rule 1 and Rule 2, defined in Section 18.3.4.3.2, apply in this mode.

Rule 3, defined in Section 18.3.4.3.2, does not apply.

In master transmit-only mode, the MCSPI_RXx register state FULL does not prevent transmission and the MCSPI_RXx register is always overwritten with the new SPI word. This event is not significant when only transmission is meaningful. Thus, the RX0_OVERFLOW bit in the MCSPI_IRQSTATUS register is never set in this mode.

The hardware automatically disables the RX_FULL interrupt and the DMA read requests.

The transfer status is given by the MCSPI_CHxSTAT[2] EOT bit.

18.3.4.3.4 Master Receive-Only Mode (Half Duplex)

The master receive mode prevents a device initiator from refilling the MCSPI_TXx register (minimizing data movement) when only reception is meaningful.

The master receive mode is programmable per channel (the MCSPI_CHxCONF[13:12] TRM bit field).

The master receive-only mode enables channel scheduling only on the empty state of the MCSPI_RXx register.

Rule 1 and Rule 3, defined in Section 18.3.4.3.2, apply in this mode.

Rule 2, defined in Section 18.3.4.3.2, does not apply.

In the master receive-only mode, software must write dummy data to the MCSPI_TXx register. Only one dummy write is enough to receive any number of words from the slave. Software must ensure that the MCSPI_TXx register is always full (the TXx_EMPTY bits of MCSPI_IRQSTATUS) when receiving. The content of the MCSPI_TXx register is always loaded into the shift register when the shift register is assigned. After writing the dummy data to the MCSPI_TXx register, the TXx_EMPTY and TXx_UNDERFLOW bits in the MCSPI_IRQSTATUS register are never set in receive-only mode.

The MCSPI_CHxSTAT[2] EOT bit gives the status of serialization. The RXx_FULL bits of the MCSPI_IRQSTATUS register are set when received data is loaded from the shift register to the corresponding MCSPI_RXx register. The MCSPI_IRQSTATUS[3] RX0_OVERFLOW bit is never set in this mode.

18.3.4.3.5 Single-Channel Master Mode

When the McSPI is configured as a master device with a single enabled channel (MCSPI_MODULCTRL[2] MS = 0 and MCSPI_MODULCTRL[0] SINGLE = 1), the assertion of the SPIEN[x] signal is optional depending on device connected to the controller. In 3-pin mode (MCSPI_MODULCTRL[1] PIN34 = 1) the controller starts transmitting data when a write to the MCSPI_TXx register or the FIFO is performed. In 4-pin mode (MCSPI_MODULCTRL[1] PIN34 = 0) the assertion and de-assertion of SPIEN[x] is controlled by software using the MCSPI_CHxCONF[20] FORCE bit.

18.3.4.3.5.1 Programming Tips When Switching to Another Channel

When a single channel is enabled and data transfer is ongoing:

- Wait for the SPI word transfer to complete (wait until the MCSPI_CHxSTAT[2] EOT bit is set to 1) before disabling the current channel and enabling a different channel.
- Disable the current channel, and then enable the other channel.

18.3.4.3.5.2 Force SPIEN[x] Mode

Continuous transfers are allowed manually by keeping the SPIEN[x] signal active for successive SPI words transfer. Several sequences (configuration/enable/disable of the channel) can be run without deactivating the SPIEN[x] line. This mode is supported by all channels and any master sequence can be used (transmit-receive, transmit-only, receive-only).

Keeping the SPIEN[x] active mode is supported when:

- A single channel is used (with the MCSPI_MODULCTRL[0] SINGLE bit set to 1).
- Transfer parameters are loaded in the configuration register of the appropriate channel (MCSPI_CHxCONF).

  The state of the SPIEN[x] signal is programmable:

  – Writing 1 to the MCSPI_CHxCONF[20] FORCE bit drives the SPIEN[x] line high when the MCSPI_CHxCONF[6] EPOL bit is set to 0. SPIEN[x] is driven low when the MCSPI_CHxCONF[6] EPOL bit is set to 1.

  – Writing 0 to the MCSPI_CHxCONF[20] FORCE bit drives the SPIEN[x] line low when the MCSPI_CHxCONF[6] EPOL bit is set to 0. SPIEN[x] is driven high when the MCSPI_CHxCONF[6] EPOL bit is set to 1.
- A single channel is enabled (the MCSPI_CHxCTRL[0] EN bit is set to 1). The first enabled channel activates the SPIEN[x] line.

When the channel is enabled, the SPIEN[x] signal activates with the programmed polarity. As in the multichannel master mode, the transfer start depends on the status of the MCSPI_TXx register (the MCSPI_CHxSTAT[1] TXS bit), the status of the MCSPI_RXx register (the MCSPI_CHxSTAT[1] RXS bit), and the defined mode (the MCSPI_CHxCONF[13:12] TRM bit field) of the channel enabled.

The MCSPI_CHxSTAT[2] EOT bit gives the transfer status of each SPI word. The RXx_FULL bit in the MCSPI_IRQSTATUS register is set when received data is loaded from the shift register to the MCSPI_RXx register.

A change in the configuration parameters is propagated directly on the SPI interface. If the SPIEN[x] signal is activated, ensure that the configuration is changed only between SPI words to avoid corrupting the current transfer.

NOTE: To avoid data corruption, SPIEN polarity and SPICLK phase and SPICLK polarity must not be modified when the SPIEN[x] signal is activated.

A delay between SPI words that requires the connected SPI slave device to switch from one configuration to another (for instance, from transmit-only to receive-only) must be handled by software.

At the end of the last SPI word, the channel must be deactivated (the MCSPI_CHxCTRL[0] EN bit set to 0) and SPIEN[x] can be forced to its INACTIVE state using the MCSPI_CHxCONF[20] FORCE bit.

Figure 18-55 and Figure 18-56 show successive transfers with SPIEN[x] maintained active low with a different configuration for each SPI word in single-data-pin and dual-data-pin interface modes, respectively.

Figure 18-55. Continuous Transfers With SPIEN[x] Maintained Active (Single-Data-Pin Interface Mode)

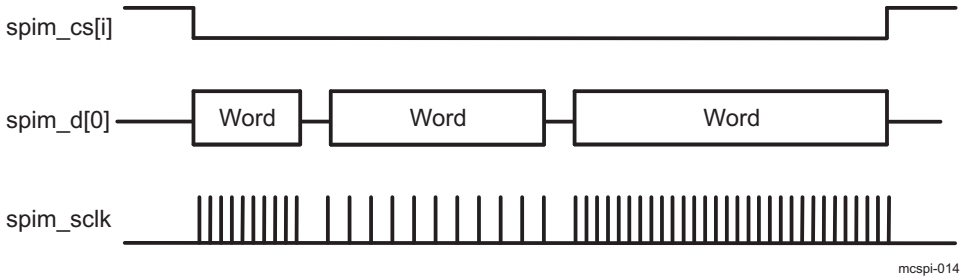
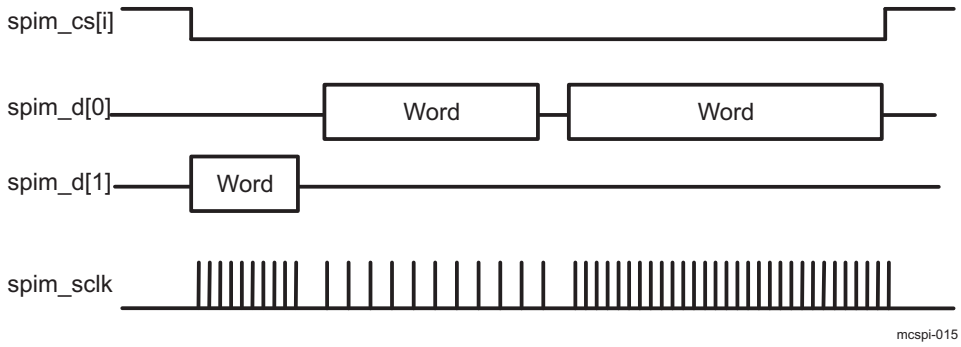


Figure 18-56. Continuous Transfers With SPIEN[x] Maintained Active (Dual-Data-Pin Interface Mode)



NOTE: The SPIEN[x] signal can be maintained active via software using the MCSPI_CHxCONF[20] FORCE bit only when the MCSPI_MODULCTRL[0] SINGLE bit is set to 0x1.

18.3.4.3.5.3 Turbo Mode

Turbo mode improves the throughput of the SPI interface when a single channel is enabled by allowing transfers until the shift register and the MCSPI_RXx register are full. Turbo mode is time saving when a transfer exceeds two words. This mode is programmable per channel (through the SPI1.MCSPI_CHxCONF[9] TURBO bit).

When several channels are enabled, the TURBO bit has no effect and the channel access to the shift registers remains as previously described.

In turbo mode, Rule 1 and Rule 2 apply, but Rule 3 does not (see Section 18.3.4.3.2, *Master Transmit-and-Receive Mode (Full Duplex)*). An enabled channel can be scheduled if its receive register is full (the MCSPI_CHxSTAT[0] RXS bit) when the shift-register is assigned until the shift register is full.

The MCSPI_RXx register cannot be overwritten in turbo mode. Consequently, the MCSPI_IRQSTATUS[3] RX0_OVERFLOW bit is never set in this mode.

18.3.4.3.6 Start-Bit Mode

In start-bit mode, an extended bit is added before the SPI word to indicate whether the next SPI word must be handled as a command or as data. This feature is available only in master mode. Start-bit mode cannot be used at the same time as turbo mode and/or force SPIEN[x] mode. In this case, only one channel can be used; round-robin arbitration is not possible.

This mode is programmable per channel by setting the MCSPI_CHxCONF[23] SBE bit to 1. The polarity of the extended bit is programmable per channel. When the MCSPI_CHxCONF[24] SBPOL bit is set to 0, the SPI word must be handled as a command. When the MCSPI_CHxCONF[24] SBPOL bit is set to 1, the SPI word must be handled as data. Moreover, start-bit polarity can be changed dynamically during start-bit transfer without disabling the channel for reconfiguration; in this case, users must configure the MCSPI_CHxCONF[24] SBPOL bit before writing the SPI word to be transmitted to the TX register.

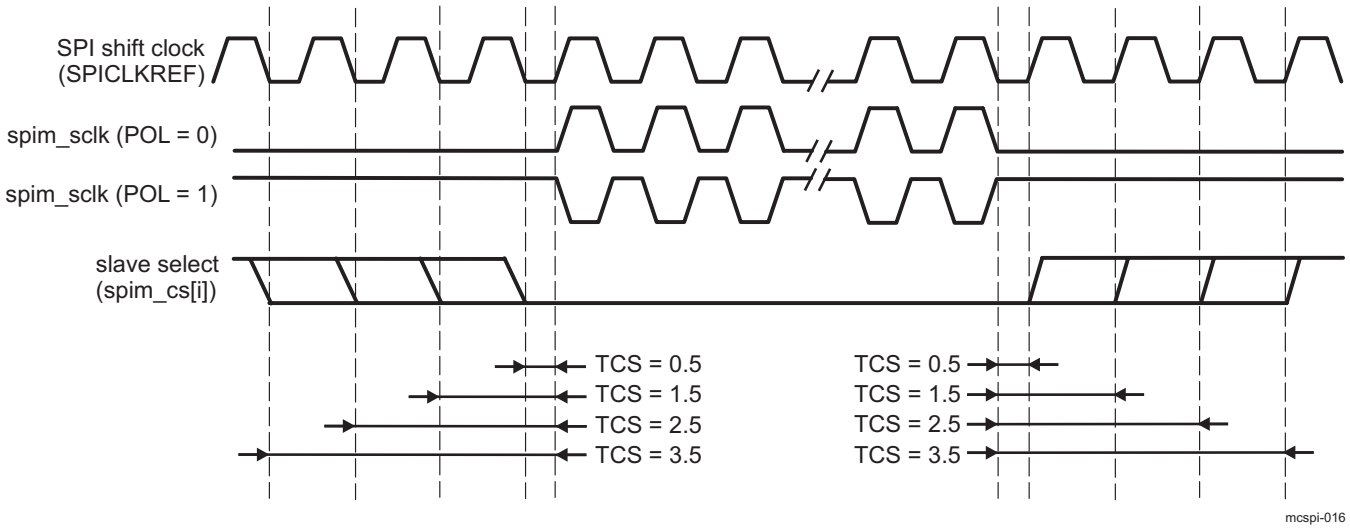
18.3.4.3.7 Chip-Select Timing Control

The chip-select (CS) timing control is available only in master mode with automatic CS generation (the MCSPI_MODULCTRL[0] SINGLE bit set to 0) to add a programmable delay between CS assertion and first clock edge, or CS removal and last clock edge. This option is available only in 4-pin mode when MCSPI_MODULCTRL[1] PIN34 set to 0.

This mode is programmable per channel through the MCSPI_CHxCONF[26:25] TCS0 bit field.

Figure 18-57 shows the CS SPIEN timing controls.

Figure 18-57. CS (SPIEN) Timing Controls



NOTE: Because of the design implementation for transfers using a clock divider ratio set to 1 (clock bypassed), a half cycle must be added to the value between CS assertion and the first clock edge with PHA = 1 or between CS removal and the last clock edge with PHA = 0.

18.3.4.3.8 Programmable SPI Clock

In master mode, the baud rate of the SPI serial clock is programmable.

An internal reference clock, SPIm_FCLK, is used as input of a programmable divider (the MCSPI_CHxCONF[5:2] CLKD bit field) to generate the bit rate of the serial output clock SPICLK. Table 18-200 summarizes the supported divisor values.

Table 18-200. SPI Master Clock Rates

Divider	Clock Rate
1	48 MHz(1)
2	24 MHz(1)
4	12 MHz
8	6 MHz
16	3 MHz
32	1.5 MHz
64	750 kHz
128	375 kHz
256	~187 kHz
512	~93.7 kHz
1024	~46.8 kHz
2048	~23.4 kHz
4096	~11.7 kHz
8192 and higher: Division not supported	–

(1) These frequencies are not necessarily supported by all SPI modules. For more information, see the *Timing Requirements and Switching Characteristics* chapter in the device data manual.

18.3.4.3.8.1 Clock Ratio Granularity

By default, the clock division ratio is defined by the MCSPI_CHxCONF[5:2] CLKD bit field with power-of-2 granularity leading to a clock division in the range 1 to 4096; in this case, the duty cycle is always 50 percent. With the MCSPI_CHxCONF[29] CLKG bit, clock division granularity can be changed to one clock cycle; in that case the MCSPI_CHxCTRL[15:8] EXTCLK bit field is concatenated with the MCSPI_CHxCONF[5:2] CLKD bit field to give a 12-bit-wide division ratio in the range 1 to 4096.

When granularity is one clock cycle (the CLKG bit set to 1), for the odd value of the clock ratio, the clock high level lasts one clock cycle more than the low level, depending on the MCSPI_CHxCONF[1] POL and MCSPI_CHxCONF[0] PHA bits (see Table 18-201).

Table 18-201. CLKSPIO High/Low Time Computation

Clock Ratio F_{RATIO}	CLKSPIO High Time	CLKSPIO Low Time
1	T_{HIGH_REF}	T_{LOW_REF}
Even ≥ 2	T_ref × (F_{RATIO}/2)	T_ref × (F_{RATIO}/2)
Odd ≥ (POL = PHA)	T_ref × (F_{RATIO}– 1)/2	T_ref × (F_{RATIO} + 1)/2
Odd ≥ (POL ≠ PHA)	T_ref × (F_{RATIO}+ 1)/2	T_ref × (F_{RATIO} – 1)/2

NOTE: F_{RATIO} = SPICLK frequency (F_{OUT}) division ratio
T_{HIGH} = SPICLK high time period
T_{LOW} = SPICLK low time period
T_ref = FCLK period
T_{HIGH_REF} = FCLK high time period
T_{LOW_REF} = FCLK low time period

If the CLKG bit is set to 1; F_{RATIO} = EXTCLK concatenated with CLKD + 1.

For odd ratio values, the duty cycle is calculated as follows:

Duty_cycle = (1 – 1/F_{RATIO})/2

Table 18-202 shows examples of clock granularity with a clock source frequency of 48 MHz.

Table 18-202. Clock Granularity Examples

EXTCLK	CLKD	CLKG	F_{RATIO}	PHA	POL	T_{HIGH} (ns)	T_{LOW} (ns)	T_{PERIOD} (ns)	Duty Cycle	F_{OUT} (MHz)
X	0	0	1	X	X	10.4	10.4	20.8	50–50	48
X	1	0	2	X	X	20.8	20.8	41.6	50–50	24
X	2	0	4	X	X	41.6	41.6	83.2	50–50	12
X	3	0	8	X	X	83.2	83.2	166.4	50–50	6
0	0	1	1	X	X	10.4	10.4	20.8	50–50	48
0	1	1	2	X	X	20.8	20.8	41.6	50–50	24
0	2	1	3	1	0	41.6	20.8	62.4	66–33	16
0	2	1	3	1	1	20.8	41.6	62.4	33–66	16
0	3	1	4	X	X	41.6	41.6	83.2	50–50	12
5	0	1	81	1	0	852.8	832	1684.8	50.6–49.4	0.592
5	7	1	88	X	X	915.2	915.2	1830.4	50–50	0.545

18.3.4.4 Slave Mode

To select the McSPI slave mode, set the MCSPI_MODULCTRL[2] MS bit.

A McSPI slave device can be connected to up to four external SPI master devices but handles transactions with one SPI master device at a time.

In slave mode, the McSPI initiates data transfer on the data lines (SPIDAT[0] and SPIDAT[1]) when it is selected by an active control signal (SPIEN[x]) and receives an SPI clock (SPICLK) from the external SPI master device. Only channel 0 can be configured as a slave but through the MCSPI_CH0CONF[22:21] SPIENSLV bit field any of the SPIEN[x] signals can be used to select the McSPI module. In slave mode and when the MCSPI_MODULCTRL[1] PIN34 is set to 0x0 (default behavior), the McSPI uses the edge of SPIEN[x] to detect word length. For this reason, SPIEN[x] must become inactive between each word.

When the MCSPI_MODULCTRL[1] PIN34 is set to 0x0, the McSPI does not support SPIEN[x] active between SPI words. In this case, the McSPI uses the edge to detect word length.

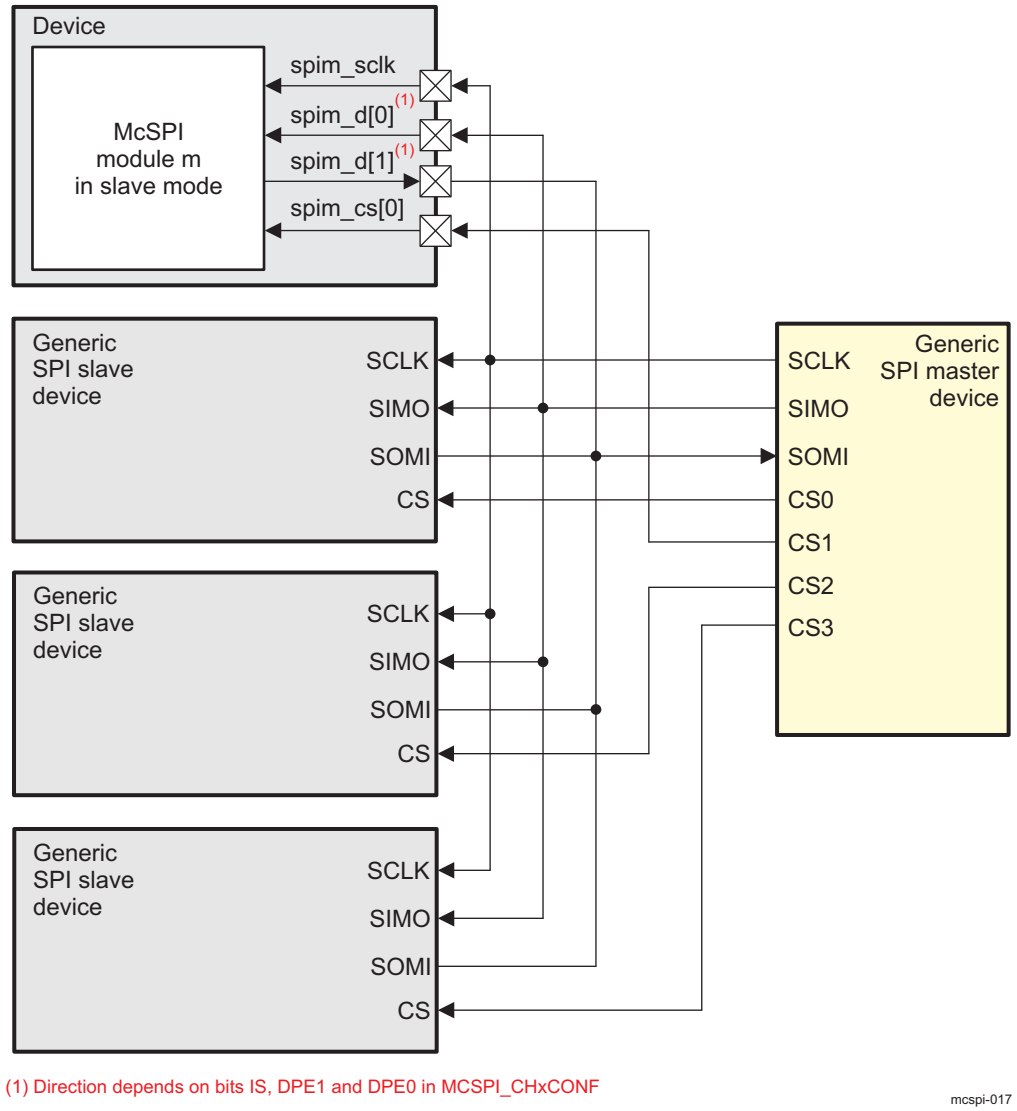
When the MCSPI_MODULCTRL[1] PIN34 is set to 0x1, a multiword transfer can be performed without needing the external SPI master to deactivate SPIEN[x] between each word as in this case the McSPI module works in 3-pin slave mode and SPIEN[x] is not needed.

18.3.4.4.1 Dedicated Resources

Only channel 0 can be enabled in slave mode. In this section, register names such as SPI1.MCSPI_CHxCTRL stand for SPI1.MCSPI_CH0CTRL, where x = 0 (channel 0 control register).

Figure 18-58 shows an example of four slaves wired on a single master device.

Figure 18-58. Example of McSPI Slave With One Master and Multiple Slave Devices on Channel 0



Channel 0 in slave mode has the following resources:

- Its own channel enable, programmable with the MCSPI_CHxCTRL[0] EN bit (where x = 0). This channel must be enabled before transmission and reception.
- For this mode, the slave-select signal can be detected on any of the SPIEN[x] ports. This is programmable with the MCSPI_CHxCONF[22:21] SPIENSLV bit field (where x = 0).
- Its own transmitter register, MCSPI_TXx (where x = 0), on top of the common transmit shift register. If the MCSPI_TXx register is empty, the MCSPI_CHxSTAT[1] TXS bit (where x = 0) is set. If McSPI is selected by an external master (the active signal on the SPIEN[x] port assigned to channel 0), the MCSPI_TXx register content of channel 0 is always loaded into the shift register, whether its content is updated or not. The MCSPI_TXx register must be loaded before McSPI is selected by a master.
- Its own receiver register, MCSPI_RXx (where x = 0), on top of the common receive shift register. If the MCSPI_RXx register is full, the MCSPI_CHxSTAT[0] RXS bit (where x = 0) is set.

NOTE: The MCSPI_TXx and MCSPI_RXx registers of the other channels are not used. Reading from or writing to a channel register other than channel 0 has no effect.

- Its own communication configuration with the following parameters through the MCSPI_CHxCONF register (where x = 0):

- Transmit and receive modes, programmable with the TRM field
- Interface mode (two data pins or single data pin) and data pins assignment, both programmable with the IS and DPE bits. (The SPIm modules are in slave mode after reset and must be properly configured for the modules to act in master mode.)
- SPI word length, programmable with the WL bits
- SPIEN[x] polarity, programmable with the EPOL bit
- SPICLK polarity, programmable with the POL bit
- SPICLK phase, programmable with the PHA bit

The SPICLK frequency of a transfer is controlled by the external SPI master connected to the McSPI slave device. The MCSPI_CHxCONF[5:2] CLKD bit field (where x = 0) is not used in slave mode.

NOTE: The configuration of the channel can be loaded in the MCSPI_CHxCONF register (where x = 0) only when the channel is disabled.

- Two DMA request events, read and write, synchronize read/write accesses of the DMA controller with the activity of McSPI. DMA requests are asserted using the MCSPI_CHxCONF[15] DMAR bit (where x = 0) for reading and the MCSPI_CHxCONF[14] DMAW bit (where x = 0) for writing.
- Four interrupt events (see Section 18.3.4.7.2, *Interrupt Events in Slave Mode*)

18.3.4.4.2 Slave Transmit-and-Receive Mode

The slave receive mode is programmable (set the MCSPI_CHxCONF[13:12] TRM bit field [where x = 0] to 0x0).

In slave transmit-and-receive mode, the MCSPI_TXx register must be loaded before McSPI is selected by an external SPI master device.

After a channel is enabled, transmission and reception proceed with interrupt and DMA request events.

The MCSPI_TXx register content is always loaded in the shift register whether it is updated or not. The event TXx_UNDERFLOW is activated accordingly and does not prevent transmission.

When the SPI word transfer completes (the MCSPI_CHxSTAT0[2] EOT bit [where x = 0] is set to 1), the received data is transferred to the channel receive register.

To use McSPI as a slave transmit-only device, the RXx_FULL and RX0_OVERFLOW interrupts and DMA read requests must be disabled due to the state of the MCSPI_RXx register (see Section 18.3.4.7.2, *Interrupt Events in Slave Mode*).

18.3.4.4.3 Slave Transmit-Only Mode

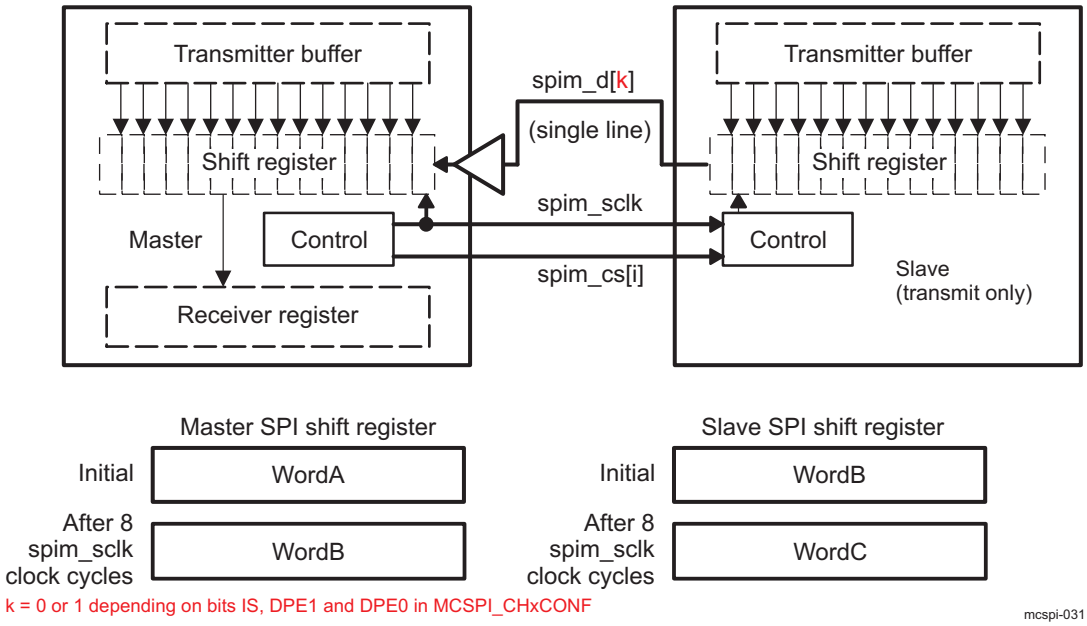
The slave transmit-only mode is programmable (set the MCSPI_CHxCONF[13:12] TRM bit field [where x = 0] to 0x2) and avoids the requirement for a device initiator to read the MCSPI_RXx register (minimizing data movement) only when transmission is meaningful.

To use the McSPI as a slave transmit-only device, the RXx_FULL and RX0_OVERFLOW interrupts and DMA read requests must be disabled due to the state of the MCSPI_RXx register.

When the SPI word transfer completes, the MCSPI_CHxSTAT[2] EOT bit is set (where x = 0).

Figure 18-59 shows a half-duplex system with a master device on the left and a transmit-only slave device on the right. Each time a bit transfers out from the slave, 1 bit transfers in the master. After eight cycles of the serial clock SPICLK, WordB transfers from the slave to the master.

Figure 18-59. SPI Half-Duplex Transmission (Transmit-Only Slave)



18.3.4.4.4 Slave Receive-Only Mode

The slave receive mode is programmable (set the MCSPI_CHxCONF[13:12] TRM bit field [where x = 0] to 0x1).

In receive-only mode, the MCSPI_TXx register must be loaded before the McSPI is selected by an external SPI master device. The MCSPI_TXx register content is always loaded into the shift register whether it is updated or not. The TXx_UNDERFLOW event is activated accordingly and does not prevent transmission.

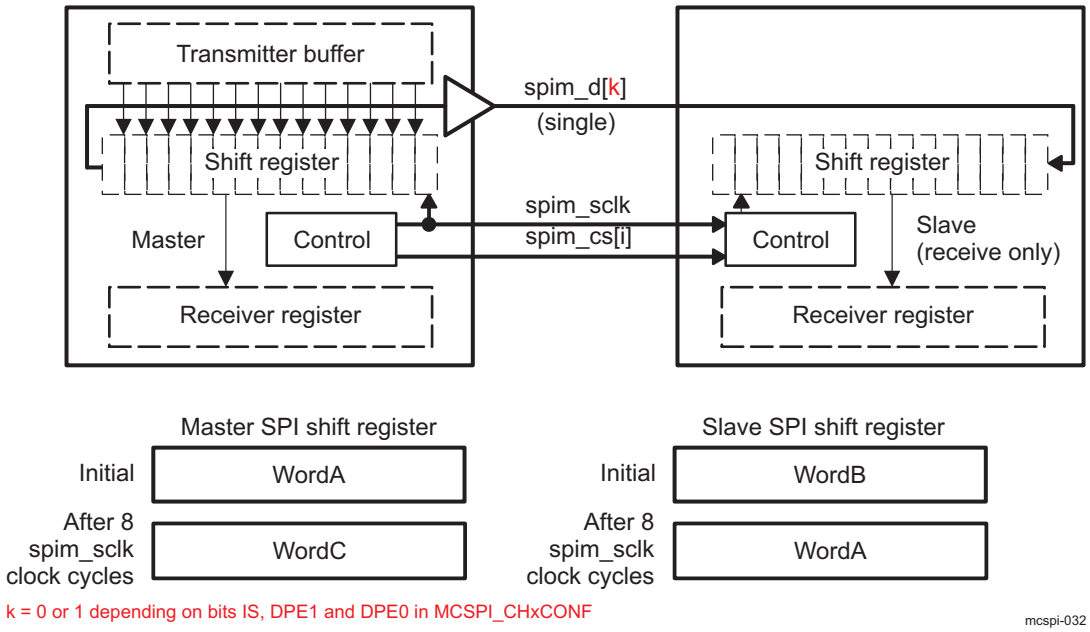
When the SPI word transfer completes (the MCSPI_CHxSTAT0[2] EOT bit [where x = 0] is set to 1), the received data is transferred to the channel receive register.

To use the McSPI as a slave receive-only device, the TXx_EMPTY and TXx_UNDERFLOW interrupts and the DMA write requests must be disabled due to the state of the MCSPI_TXx register.

For a full-duplex transmission, the serial clock (SPICLK) synchronizes shifting and sampling of the information on the single serial data line. For full duplex, two data lines are required. If SPICLK synchronizes on a single serial data line, the data line should be half-duplex.

Figure 18-60 shows a half-duplex system with a master device on the left and a receive-only slave device on the right. Each time a bit transfers out from the master, 1 bit transfers in from the slave. After eight cycles of the serial clock SPICLK, WordA transfers from the master to the slave.

Figure 18-60. SPI Half-Duplex Transmission (Receive-Only Slave)



18.3.4.5 3-Pin or 4-Pin Mode

Depending on targeted application the SPI interface can be configured to use 3 or 4 pins through the MCSPI_MODULCTRL[1] PIN34 bit. If this bit is set to 0, McSPI is in 4-pin mode using the SPICLK, SPIDAT[0], SPIDAT[1] and SPIEN[x] signals. If PIN34 is set to 1 the controller is in 3-pin mode and SPIEN[x] is not used. In this mode all options related to chip select management are useless (EPOL, FORCE and TCS0 bits of MCSPI_CHxCONF). 3-pin and 4-pin operation applies to both master and slave modes.

18.3.4.6 FIFO Buffer Management

The McSPI controller has a built-in 128-byte buffer to unload the DMA or interrupt handler and improve data throughput.

This buffer can be used by only one channel at a time and is selected by setting the MCSPI_CHxCONF[28] FFER or MCSPI_CHxCONF[27] FFEW bit to 1. If several channels are selected and several FIFO enable bit fields are set to 1, the controller forces the buffer not to be used; the driver must set only one FIFO enable bit field.

The buffer can be used in the following modes:

- Master or slave mode
- Transmit-only, receive-only, or transmit-and-receive mode
- Single channel or turbo mode, or normal round-robin mode. In round-robin mode the buffer is used by only one channel.

Every word length (MCSPI_CHxCONF[11:7] WL) is supported.

In transmit-and-receive mode, the buffer can be used in transmit (see Figure 18-61) or receive (see Figure 18-62) directions, or in both directions. If only one direction is chosen in transmit-and-receive mode, the full buffer is used for this direction. In both directions, the buffer is split into two halves, one for each direction (see Figure 18-63).

Figure 18-61. Buffer Used in Transmit Direction Only

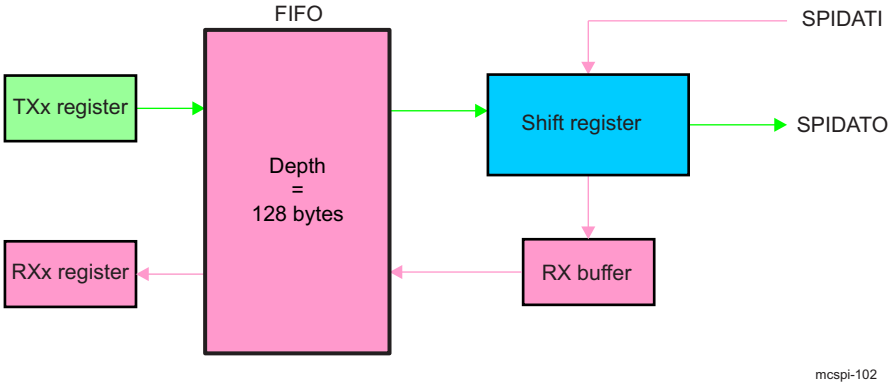


Figure 18-62. Buffer Used in Receive Direction Only

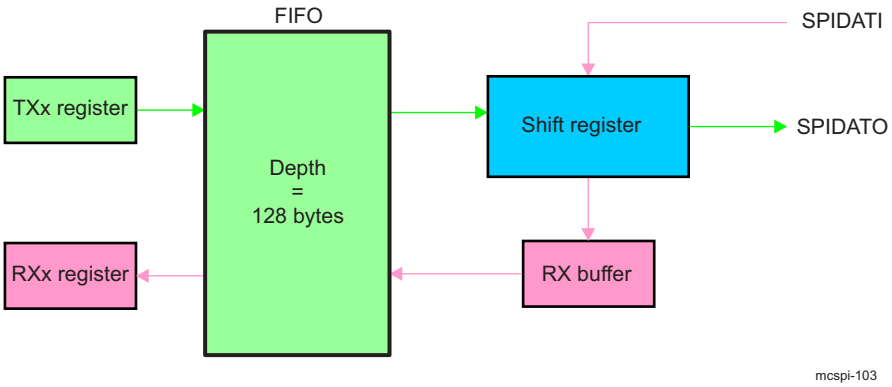
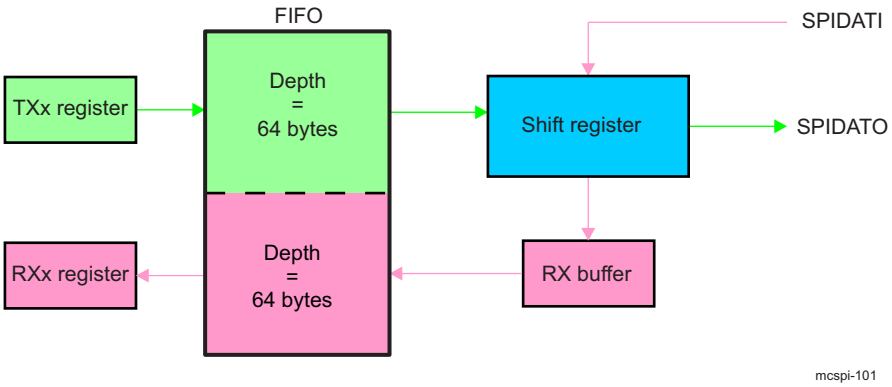


Figure 18-63. Buffer Used for Transmit and Receive Directions



Two levels (MCSPI_XFERLEVEL[5:0] AEL and MCSPI_XFERLEVEL[13:8] AFL) rule the buffer management. The granularity of these levels is 1 byte; it is not aligned with the SPI word length. The driver must set these values as a multiple of the SPI word length defined in MCSPI_CHxCONF[11:7] WL. Table 18-203 lists the number of bytes written in the FIFO, depending on the word length.

Table 18-203. FIFO Writes, Word Length Relationship

	SPI Word Length (WL)		
	3 ≤ WL≤ 7	**8 ≤ WL ≤ 15**	**16 ≤ WL ≤ 31**
Number of bytes written in the FIFO	1 byte	2 bytes	4 bytes

The FIFO buffer pointers are reset when the corresponding channel is enabled or the FIFO configuration changes.

18.3.4.6.1 Buffer Almost Full

The MCSPI_XFERLEVEL[15:8] AFL bit field is needed when the buffer is used to receive an SPI word from a slave (the MCSPI_CHxCONF[28] FFER bit must be set to 1). It defines the almost-full buffer status. See Figure 18-64.

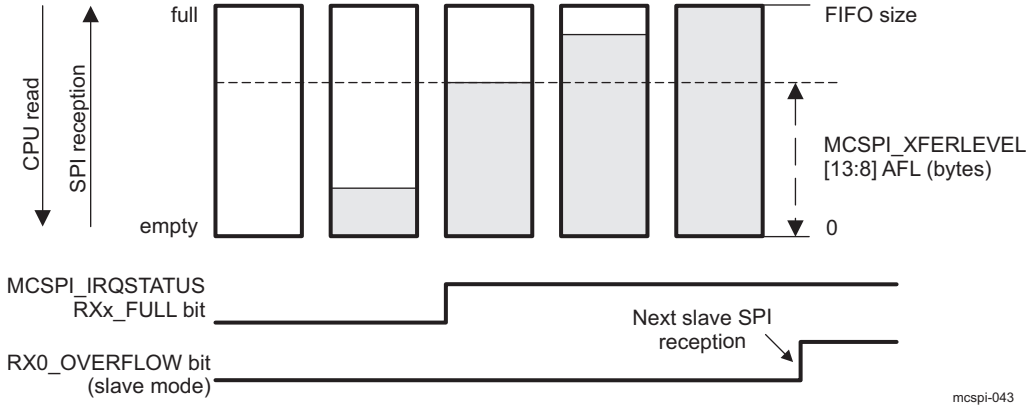
When the FIFO pointer reaches this level, an interrupt or a DMA request is sent to the IRQ_CROSSBAR and then to the desired INTC to enable the system to read AFL + 1 bytes from the receive register.

NOTE: AFL + 1 must correspond to a multiple value of the MCSPI_CHxCONF[11:7] WL bit field.

When DMA is used, the request is deasserted after the first receive register read.

No new request is asserted again as long as the system has not performed the correct number of read accesses.

Figure 18-64. Buffer Almost Full Level (AFL)



NOTE: The MCSPI_IRQSTATUS register bits are not available in DMA mode. In DMA mode, the SPIm_DMA_RXx request is asserted on the same conditions as the MCSPI_IRQSTATUS RXx_FULL flag.

18.3.4.6.2 Buffer Almost Empty

The MCSPI_XFERLEVEL[7:0] AEL bit field is needed when the buffer is used to transmit an SPI word to a slave (the MCSPI_CHxCONF[27] FFEW bit must be set to 1). It defines the almost-empty buffer status. See Figure 18-65.

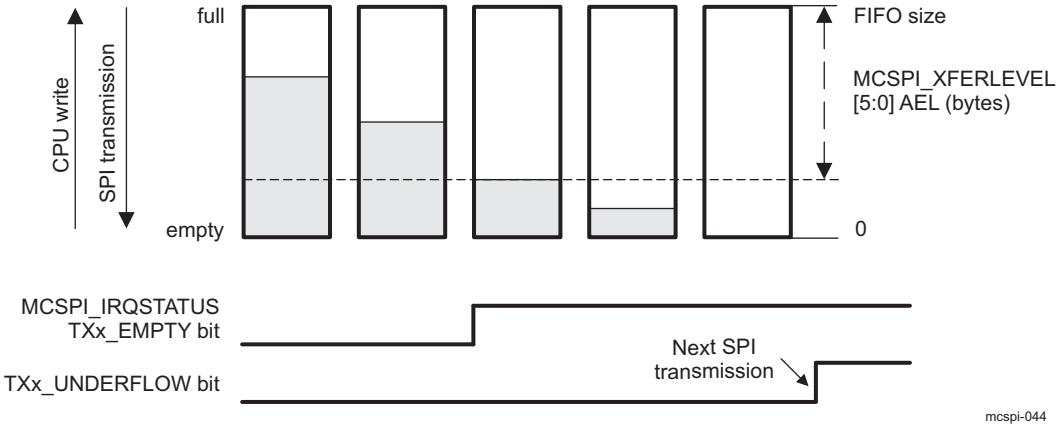
When the FIFO pointer does not reach this level, an interrupt or a DMA request is sent to the IRQ_CROSSBAR and then to the desired INTC to enable the system to write AEL + 1 bytes to the transmit register.

NOTE: AEL + 1 must correspond to a multiple value of the MCSPI_CHxCONF[11:7] WL bit field.

When DMA is used, the request is deasserted after the first transmit register write.

No new request is asserted again as long as the system has not performed the correct number of write accesses.

Figure 18-65. Buffer Almost Empty Level (AEL)



NOTE: The MCSPI_IRQSTATUS register bits are not available in DMA mode. In DMA mode, the SPIm_DMA_TXx request is asserted on the same conditions as the MCSPI_IRQSTATUS TXx_EMPTY flag.

18.3.4.6.3 End of Transfer Management

When the FIFO buffer is enabled for a channel, the user must previously configure in the MCSPI_XFERLEVEL register the AEL and AFL levels and especially the MCSPI_XFERLEVEL[31:16] WCNT bit field to define the number of SPI words to be transferred using the FIFO before enabling the channel.

This counter lets the controller stop the transfer correctly after a defined number of SPI word transfers. If WNCT is set to 0x0000, the counter is not used and the user must stop the transfer manually by disabling the channel; in this case, the user does not know how many SPI transfers have been done. For received words, software must poll the CHxSTAT[5] RXFFE bit and read the MCSPI_RXx receive register to empty the FIFO buffer.

When the end-of-word count interrupt is generated (the MCSPI_IRQSTATUS[17] EOW bit is set), the user can disable the channel and poll the MCSPI_CHxSTAT[5] RXFFE bit to know the last SPI words in the FIFO buffer and read them.

No new request is asserted as long as the system has not performed the correct number of write accesses.

18.3.4.7 Interrupts

Each channel can issue interrupt events.

Each interrupt event has status bits in the MCSPI_IRQSTATUS register (RXx_FULL, TXx_UNDERFLOW, TXx_EMPTY, etc.) (where x = 0, 3) that indicate whether service is required. Each status bit has an interrupt enable bit (a mask) in the MCSPI_IRQENABLE register (RXx_FULL_ENABLE, TXx_UNDERFLOW_ENABLE, TXx_EMPTY_ENABLE, etc.).

When an interrupt occurs and a mask is later applied on it, the interrupt line is not asserted again, even if the interrupt source is not serviced.

The McSPI supports interrupt-driven and polling operations.

18.3.4.7.1 Interrupt Events in Master Mode

In master mode, the interrupt events related to the state of the MCSPI_TXx register are TXx_EMPTY and TXx_UNDERFLOW. The interrupt event related to the state of the MCSPI_RXx register is RXx_FULL.

18.3.4.7.1.1 TXx_EMPTY

The TXx_EMPTY event is activated when a channel is enabled and its MCSPI_TXx register is empty (transient event). Enabling a channel automatically triggers this event, except in master receive-only mode (see Section 18.3.4.3.4, *Master Receive-Only Mode*). When the FIFO buffer is enabled (the MCSPI_CHxCONF[27] FFEW bit is set to 1), the MCSPI_IRQSTATUS TXx_EMPTY bit is set as soon as there is enough space in the buffer to write a number of bytes defined by the MCSPI_XFERLEVEL[5:0] AEL bit field.

The MCSPI_TXx register must be loaded with data to remove the source of the interrupt; the MCSPI_IRQSTATUS TXx_EMPTY interrupt status bit must be cleared for interrupt line deassertion (if the event is enabled as the interrupt source).

When FIFO is enabled, no new TXx_EMPTY event is asserted as long as a device initiator has not performed the number of writes into the MCSPI_TXx register defined by the MCSPI_XFERLEVEL[5:0] AEL bit field. This device initiator must perform the correct number of writes.

18.3.4.7.1.2 TXx_UNDERFLOW

The event TXx_UNDERFLOW is activated when the channel is enabled and if the MCSPI_TXx register or the FIFO is empty (not updated with new data) when an external master device starts a data transfer with the McSPI (transmit and receive).

The TXx_UNDERFLOW is a harmless warning in master mode.

To avoid having a TXx_UNDERFLOW event at the beginning of a transmission, the TXx_UNDERFLOW event is not activated when no data has been loaded into the MCSPI_TXx register, because the channel is enabled. To avoid having a TXx_UNDERFLOW event, the MCSPI_TXx register must seldom be loaded.

The MCSPI_IRQSTATUS TXx_UNDERFLOW interrupt status bit must be cleared for interrupt line deassertion (if the event is enabled as the interrupt source).

18.3.4.7.1.3 RXx_ FULL

The RXx_FULL event is activated when a channel is enabled and the MCSPI_RXx register becomes filled (transient event). When the FIFO buffer is enabled (the MCSPI_CHxCONF[28] FFER bit is set to 1), RXx_ FULL is asserted as soon as the number of bytes held in the FIFO to be read reaches the MCSPI_XFERLEVEL[13:8] AFL threshold.

The MCSPI_RXx register must be read to remove the source of the interrupt; the MCSPI_IRQSTATUS RXx_FULL interrupt status bit must be cleared for interrupt line deassertion (if the event is enabled as the interrupt source).

When FIFO is enabled, no new RXx_FULL event is asserted as long as a device initiator has not performed AFL + 1 reads into MCSPI_RXx. This device initiator must perform the correct number of reads.

18.3.4.7.1.4 End Of Word Count

The MCSPI_IRQSTATUS[17] EOW event (end of word count) is activated when the channel is enabled and configured to use the built-in FIFO. This interrupt is raised when the controller performs the number of transfers defined in the MCSPI_XFERLEVEL[31:16] WCNT bit field. If WCNT is set to 0x0000, the counter is not enabled and this interrupt is not generated.

The end of word count interrupt also indicates that the SPI transfer is halted on the channel using the FIFO buffer as soon as MCSPI_XFERLEVEL[31:16] WCNT is not reloaded and the channel is not re-enabled.

The MCSPI_IRQSTATUS[17] EOW interrupt status bit must be cleared for interrupt line deassertion (if the event is enabled as the interrupt source).

18.3.4.7.2 Interrupt Events in Slave Mode

In slave mode, the interrupt events related to the state of the MCSPI_TXx register are TXx_EMPTY and TXx_UNDERFLOW. The interrupt events related to the state of the MCSPI_RXx are RXx_FULL and RX0_OVERFLOW (channels 1, 2, and 3 do not have a receiver overflow status bit). See the MCSPI_IRQSTATUS register.

18.3.4.7.2.1 TXx_EMPTY

The TXx_EMPTY event is activated when a channel is enabled and its MCSPI_TXx register is empty. Enabling the channel automatically raises this event. If the FIFO buffer is enabled (the MCSPI_CHxCONF[27] FFEW bit is set to 1), the TXx_EMPTY event is asserted as soon as there is enough space in buffer to write a number of bytes defined by the MCSPI_XFERLEVEL[5:0] AEL bit field.

The MCSPI_TXx register must be loaded with data to remove the source of the interrupt; the MCSPI_IRQSTATUS TXx_EMPTY interrupt status bit must be cleared for interrupt line deassertion (if the event is enabled as the interrupt source).

When FIFO is enabled, no new TXx_EMPTY event is asserted as long as a device initiator has not performed the number of writes into the MCSPI_TXx register defined by MCSPI_XFERLEVEL[5:0] AEL bit field. This device initiator must perform the correct number of writes.

18.3.4.7.2.2 TXx_UNDERFLOW

The TXx_UNDERFLOW event is activated when a channel is enabled and if the MCSPI_TXx register is empty (not updated with new data) when an external master device starts a data transfer with the McSPI (transmit and receive).

When FIFO is enabled, the data emitted while the underflow event is raised is not the last data written in the FIFO but the next data in the FIFO (an old transmitted value or a dummy data in the FIFO has been reset).

TXx_UNDERFLOW indicates an error (data loss) in slave mode.

To avoid having a TXx_UNDERFLOW event at the beginning of a transmission, the TXx_UNDERFLOW event is not activated when no data has been loaded into the MCSPI_TXx register because the channel is enabled.

The MCSPI_IRQSTATUS TXx_UNDERFLOW interrupt status bit must be cleared for interrupt line deassertion (if the event is enabled as the interrupt source).

18.3.4.7.2.3 RXx_FULL

The RXx_FULL event is activated when a channel is enabled and the MCSPI_RXx register is being filled (transient event). When the FIFO buffer is enabled (the MCSPI_CHxCONF[28] FFER bit is set to 1), RXx_FULL is asserted as soon as the number of bytes held in the buffer to read defined by the MCSPI_XFERLEVEL[13:8] AFL bit field.

The MCSPI_RXx register must be read to remove the source of the interrupt; the MCSPI_IRQSTATUS RXx_FULL interrupt status bit must be cleared for interrupt line deassertion (if the event is enabled as the interrupt source).

When FIFO is enabled, no new RXx_FULL event is asserted as long as a device initiator has not performed AFL + 1 reads into MCSPI_RXx. This device initiator must perform the correct number of reads.

18.3.4.7.2.4 RX0_OVERFLOW

The RX0_OVERFLOW event is activated in slave mode in transmit-and-receive mode or receive-only mode when a channel is enabled and the MCSPI_RXx register or FIFO is full when a new SPI word is received. The MCSPI_RXx register is always overwritten with the new SPI word. If the FIFO is enabled, data within the FIFO are overwritten; it must be considered as corrupted. The RX0_OVERFLOW event should not appear in slave mode using the FIFO.

The RX0_OVERFLOW event indicates an error (data loss) in slave mode.

The MCSPI_IRQSTATUS[3] RX0_OVERFLOW interrupt status bit must be cleared for interrupt line deassertion (if the event is enabled as the interrupt source).

18.3.4.7.2.5 End Of Word Count

The MCSPI_IRQSTATUS[17] EOW event (end of word count) is activated when the channel is enabled and configured to use the built-in FIFO. This interrupt is raised when the controller performs the number of transfers defined in the MCSPI_XFERLEVEL[31:16] WCNT bit field. If WCNT is set to 0x0000, the counter is not enabled and this interrupt is not generated.

The end of word count interrupt also indicates that the SPI transfer is halted on the channel using the FIFO buffer as soon as WCNT is not reloaded and the channel is not re-enabled.

The MCSPI_IRQSTATUS[17] EOW interrupt status bit must be cleared for interrupt line deassertion (if the event is enabled as the interrupt source).

18.3.4.7.3 Interrupt-Driven Operation

An interrupt enable bit in the MCSPI_IRQENABLE register can be set to enable each event to generate interrupt requests when the corresponding event occurs. Status bits are automatically set by hardware logic conditions.

When an event occurs (the single interrupt line is asserted), the following must be performed:

1. Read the MCSPI_IRQSTATUS register to identify which event occurred.
2. Read the MCSPI_RXx register that corresponds to the event to remove the source of an RXx_FULL event or write into the MCSPI_TXx register that corresponds to the event to remove the source of a TXx_EMPTY event. No action is required to remove the source of the WKS (wake-up), TXx_UNDERFLOW, and RX0_OVERFLOW events.
3. Set the corresponding bit of the MCSPI_IRQSTATUS register to 1 to clear an interrupt status and then release the interrupt line.

The interrupt status bit must always be reset after channel enabling and before events are enabled as interrupt sources.

18.3.4.7.4 Polling

When the interrupt capability of an event is disabled in the MCSPI_IRQENABLE register, the interrupt line is not asserted, but the status bits in the MCSPI_IRQSTATUS register can be polled by software to detect when the corresponding event occurs.

Once the expected event occurs:

- RXx_FULL: To remove the source of the event, the corresponding MCSPI_RXx register must be read.
- TXx_EMPTY: To remove the source of the event, it must be written into the corresponding MCSPI_TXx register.
- WKS (wake-up), TXx_UNDERFLOW, and RX0_OVERFLOW: No action is required to remove the source of the event.

To clear an interrupt, set the corresponding status bit of the MCSPI_IRQSTATUS register to 1. This does not affect the interrupt line state.

18.3.4.8 DMA Requests

Each McSPI channel, if enabled, can issue DMA requests. There are two DMA request lines per McSPI channel (one for read and one for write).

The DMA read request line is asserted when the McSPI channel is enabled and new data is available in the receive register of the McSPI channel. A DMA read request can be individually masked with the SPI1.MCSPI_CHxCONF[15] DMAR bit. The DMA read request line is deasserted when reading of the MCSPI_RXx register of the McSPI channel completes.

The DMA write request line is asserted when the McSPI channel is enabled and the MCSPI_TXx register of the McSPI channel is empty. A DMA write request can be individually masked with the SPI1.MCSPI_CHxCONF[14] DMAW bit. The DMA write request line is deasserted when loading of the MCSPI_TXx register of the channel completes.

18.3.4.9 Power Saving Management

Power consumption can be optimized by switching off internal clocks (interface and functional clock) when there is no activity. The McSPI is compliant with the idle and wake-up system handshake protocol.

18.3.4.9.1 Normal Mode

In normal mode, internal SPI module clocks are automatically switched off (autogated) when there is no activity in slave or master mode.

Autogating of the module interface clock and functional clock occurs when the following conditions are met:

- The MCSPI_SYSCONFIG[0] AUTOIDLE bit is set.
- In master mode, there is no data to transmit or receive in all channels.
- In slave mode, the McSPI is not selected by the external master and there are no register accesses.

Autogating of the module interface clock and functional clock stops when the following conditions are met:

- In master mode, an internal access occurs.
- In slave mode, an internal access occurs or the McSPI is selected by the external master.

18.3.4.9.2 Idle Mode

At the PRCM module level, when all conditions are met to shut off the PER_48M_GFCLK or L4PER_L3_GICLK output clock, the PRCM module automatically launches a hardware handshake protocol to ensure that the McSPI is ready to have its clocks switched off. Namely, the PRCM module asserts an IDLE request to the McSPI.

Although this handshake is completely hardware-oriented and out of software control, the method in which the McSPI module acknowledges the PRCM IDLE request is configurable through the MCSPI_SYSCONFIG[4:3] SIDLEMODE bit field.

The settings of the SIDLEMODE bit field and the related acknowledgment modes are:

- Force-idle mode (the MCSPI_SYSCONFIG[4:3] SIDLEMODE bit field is set to 0x0): The McSPI module acknowledges unconditionally the IDLE request from the PRCM module, regardless of its internal operations. This mode must be used carefully in this case because it does not prevent the loss of data when the clock is switched off.
- No-idle mode (the SIDLEMODE bit field is set to 0x1): The McSPI never acknowledges an IDLE request from the PRCM module and is safe from a module point of view because it ensures that the clocks remain active. However, it is not efficient to save power because it does not allow the PRCM output clock to be shut off and thus the power domain to be set to a lower power state.
- Smart-idle mode (the SIDLEMODE bit field is set to 0x2): The McSPI acknowledges the IDLE request, basing its decision on its internal activity. Namely, the acknowledge signal is asserted only when all pending transactions, IRQs, or DMA requests are treated. This is the best approach for efficient system power management.

When configured in smart-idle mode, the McSPI also offers an additional granularity on the PER_48M_GFCLK and L4PER_L3_GICLK gating. The MCSPI_SYSCONFIG[9:8] CLOCKACTIVITY bit field determines which clock shuts down (the PER_48M_GFCLK, L4PER_L3_GICLK, neither clock, or both clocks).

The setting of the CLOCKACTIVITY bit field is used internally to the McSPI to determine on which part of the module the conditions to acknowledge the PRCM IDLE request are tested. For example, if PER_48M_GFCLK is not shut down on a PRCM IDLE request, the McSPI considers only L4PER_L3_GICLK and the associated pending activities before acknowledging the request.

Some McSPI features are associated with L4PER_L3_GICLK and others with PER_48M_GFCLK. Using the CLOCKACTIVITY bit field with the smart-idle mode ensures that the features associated with the clock that remains active are always enabled, even if the McSPI acknowledges an IDLE request.

The settings of the CLOCKACTIVITY bit field and the associated features are:

- CLOCKACTIVITY set to 00: ICLK off and FCLK off, ICLK and FCLK are considered for generating the acknowledge. This setting also means that FCLK and ICLK are likely to be shut down on a PRCM IDLE request.
- CLOCKACTIVITY set to 01: ICLK on and FCLK off, ICLK is not shut down on a PRCM IDLE request; only FCLK is concerned.
- CLOCKACTIVITY set to 10: ICLK off and FCLK on, FCLK is not shut down on a PRCM IDLE request; only ICLK is concerned.
- CLOCKACTIVITY set to 11: ICLK on and FCLK on, none of the clocks are shut down. This means the McSPI can potentially acknowledge the IDLE request without checking the internal functions linked to its clocks.

CAUTION

The PRCM module does not have a hardware means of reading the CLOCKACTIVITY settings. Therefore, software must ensure consistent programming between CLOCKACTIVITY and the PER_48M_GFCLK and L4PER_L3_GICLK control bits in the PRCM module. If the McSPI is disabled in the CM_FCLKEN and CM_ICLKEN PRCM registers while CLOCKACTIVITY is set to 11, nothing prevents the PRCM module from asserting its IDLE request, which is acknowledged regardless of the features associated with the McSPI clocks. This can lead to unpredictable behavior.

18.3.4.9.2.1 Wake-Up Event in Smart-Idle Mode

The module wake-up feature is enabled when the MCSPI_SYSCONFIG[2] ENAWAKEUP and MCSPI_WAKEUPENABLE[0] WKEN bits are set. Wake-up capability is relevant only when the module is configured in slave mode.

The module generates an asynchronous wake-up request to the system power manager to switch back the interface clock and the functional clock. A wakeup is requested when channel 0 is enabled and an asynchronous selection occurs on the mcspim.csx port associated with channel 0 (see the definition for the MCSPI_CHxCONF[22:21] SPIENSLV bit field [where x = 0] in the register table description).

After the McSPI wake-up request, the system power manager must reactivate the interface clock:

- Before the beginning of the second SPI word serialization when the McSPI is in slave transmit-only mode or in slave transmit-and-receive mode
- Before the end of the second received SPI word in slave receive-only mode. To avoid data loss, the first received SPI word must be read from the MCSPI_RXx register (where x = 0) before the completion of the second SPI word serialization.

Table 18-204 lists the supported cases in smart-idle mode.

Table 18-204. Smart-Idle Mode and Wake-Up Capabilities

Mode	Interface Clock	SPI Clock Ref	Functionality	Wake-Up Event
Master	Must be maintained	Must be maintained	Full functionality, but the module does not generate a new interrupt or DMA request until the system exits wake-up mode	No wake-up event
Slave	Can be switched off	Can be switched off	An SPI word can be transmitted and/or received, but the module does not generate any new interrupts or DMA requests until the system exits wake-up mode.	The module asynchronously sends a wake-up request if an event on the SPIEN[x] port associated with channel 0 is detected.

In wake-up mode, the interrupt and DMA request lines are no longer asserted.

Any access to the module in wake-up mode generates an error as long as the interface clock is alive.

18.3.4.9.2.2 Transitions From Smart-Idle Mode to Normal Mode

The McSPI detects the end of the wake period through the idle and wake-up hardware handshake protocol.

The interrupt status register (the MCSPI_IRQSTATUS[16] WKS bit) is updated with the event causing the wakeup; the wake-up event at the origin of the transition to the normal mode is converted to its corresponding interrupt when enabled by the MCSPI_IRQENABLE[16] WKE bit or the DMA request.

Interrupts and wake-up events have independent enable and disable controls, accessible through the MCSPI_IRQENABLE and MCSPI_WAKEUPENABLE registers. Software must ensure the overall consistency.

The interrupt status register MCSPI_IRQSTATUS is updated with the event causing the wakeup; the wake-up event at the origin of the transition to normal mode is converted to its corresponding interrupt request or DMA request. The module is fully operational.

18.3.4.9.2.3 Force-Idle Mode

Force-idle mode is enabled and exited as follows:

- Force-idle mode is enabled when the MCSPI_SYSCONFIG[4:3] SIDLEMODE bit field is set to 0x0.
  - In force-idle mode, the McSPI responds unconditionally to the IDLE request by deasserting unconditionally the interrupt and DMA request lines, if asserted. In addition, the wake-up capability is totally inhibited even if the MCSPI_SYSCONFIG[2] ENAWAKEUP and MCSPI_WAKEUPENABLE[0] WKEN bits are set.
  - The transition from normal mode to idle mode does not affect the interrupt event bits of the MCSPI_IRQSTATUS register.
  - In force-idle mode, because the module must be disabled, the interrupt and DMA request lines are likely deasserted. The interface clock and SPI clock provided to the McSPI can be switched off.
  - An IDLE request during an SPI data transfer can lead to an unexpected and unpredictable result. Software must avoid such a request.
- The module exits force-idle mode through the idle and wake-up hardware handshake protocol.

  The module is fully operational. The interrupt and DMA request lines are optionally asserted one clock cycle later.

18.3.5 McSPI Programming Guide

This section describes the low-level hardware programming sequences for the configuration and use of the McSPI module.

18.3.5.1 Global Initialization

18.3.5.1.1 Surrounding Modules Global Initialization

This section identifies the requirements for initializing the surrounding modules when the McSPI module is to be used for the first time after a device reset. This initialization of surrounding modules is based on the integration and environment of the McSPI. For further information, see Section 18.3.3, *McSPI Integration* and Section 18.3.2, *McSPI Environment*.

Table 18-205 lists the information on the global initialization of the surrounding modules.

Table 18-205. Global Initialization of Surrounding Modules

Surrounding Modules	Comments
PRCM	MCSPI_FCLK functional clock must be enabled. See Section 3.1.1.1.2, *Module-Level Clock Management*, in Chapter 3, *Power, Reset, and Clock Management*.
L4 Interconnect	For information about L4_PER1 interconnect configuration, see Section 9.3, *L4 Interconnects*, in Chapter 9, *Interconnect*.
DMA_CROSSBAR	DMA_CROSSBAR configuration must be done to allow module DREQs to be mapped to certain device DMA line. For more information see Section 13.4.6.5, *DMA_CROSSBAR Module Functional Description*, in Chapter 13, *Control Module*.
EDMA	EDMA configuration must be done to enable the module DMA channel requests.
IRQ_CROSSBAR	IRQ_CROSSBAR configuration must be done to allow module IRQs to be mapped to certain device INTC line. For more information see Section 13.4.6.4, *IRQ_CROSSBAR Module Functional Description*, in Chapter 13, *Control Module*
Device INTCs	Device INTCs must be configured to enable the interrupt request generation. For more information see Chapter 12 *Interrupt Controllers*

18.3.5.1.2 McSPI Global Initialization

18.3.5.1.2.1 Main Sequence – McSPI Global Initialization

The procedure in Table 18-206 can be used to initialize McSPI when performing software reset.

Table 18-206. McSPI Global Initialization

Step	Register/Bit Field/Programming Model	Value
Perform a software reset.	MCSPI_SYSCONFIG[1] SOFTRESET	1
Wait until reset is finished?	MCSPI_SYSSTATUS[0] RESETDONE	=1
Configure static settings (such as SPI master or slave) as required.	MCSPI_MODULCTRL[8:0]	0x-
Write MCSPI_SYSCONFIG	MCSPI_SYSCONFIG	0x-

18.3.5.2 Operational Mode Configuration

18.3.5.2.1 McSPI Operational Modes

The selection of the working mode is done with the MCSPI_CHxCONF register.

Table 18-207. McSPI Receive Mode Initialization

Step	Register/Bit Field/Programming Model	Value
Set receive mode for the channel.	MCSPI_CHxCONF[13:12] TRM	0x1
Configure SPI clock polarity/phase, clock divider, word length, and others for the channel.	MCSPI_CHxCONF	0x-
Reset the status bits.	MCSPI_IRQSTATUS	0x0

Table 18-208. McSPI Transmit Mode Initialization

Step	Register/Bit Field/Programming Model	Value
Set transmit mode for the channel.	MCSPI_CHxCONF[13:12] TRM	0x2
Configure SPI clock polarity/phase, clock divider, word length, and others for the channel.	MCSPI_CHxCONF	0x-
Reset the status bits.	MCSPI_IRQSTATUS	0x0

Table 18-209. McSPI Transmit-and-Receive Mode Initialization

Step	Register/Bit Field/Programming Model	Value
Set transmit and receive mode for the channel.	MCSPI_CHxCONF[13:12] TRM	0x0
Configure SPI clock polarity/phase, clock divider, word length, and others for the channel.	MCSPI_CHxCONF	0x-
Reset the status bits.	MCSPI_IRQSTATUS	0x0

18.3.5.2.1.1 Common Transfer Sequence

McSPI module allows the transfer of one or several words, according to different modes:

- MASTER Normal, MASTER Turbo, SLAVE
- TRANSMIT–RECEIVE, TRANSMIT-ONLY, RECEIVE-ONLY
- Write and Read requests: Interrupts, DMA
- SPIEN[x] lines assertion/deassertion: automatic, manual

For all these sequences, the host process contains the main process and the interrupt routines.

The interrupt routines are called on the interrupt signals or by an internal call if the module is used in polling mode.

Table 18-210 represents the main sequence which is common to all transfers.

In multi-channel master mode, the sequences of different channels can be run simultaneously.

Table 18-210. Common Transfer Sequence (Main Process)

Step	Register/Bit Field/Programming Model	Value
Write MCSPI_IRQSTATUS to reset channel status bits	MCSPI_IRQSTATUS[channel x bits]	0b1111
Write MCSPI_IRQENABLE to enable interrupts	MCSPI_IRQENABLE	0x-
Write MCSPI_CHxCONF to configure the channel	MCSPI_CHxCONF	0x-
Start the channel	MCSPI_CHxCTRL[0] EN	1
Wait for the first write request (TX empty or DMA write)		
Write the transmitter register with data	MCSPI_TXx	0x-
Wait for the host event for end of transfer		
Stop the channel	MCSPI_CHxCTRL[0] EN	0

18.3.5.2.1.2 End of Transfer Sequences

The end of transfer depends on the transfer mode. Table 18-211 summarizes the type of end of transfer per transfer mode and gives a reference to the appropriate section for details.

Table 18-211. End of Transfer Sequences

		TRANSMIT-AND-RECEIVE		TRANSMIT-ONLY		RECEIVE-ONLY	
		INTERRUPT	DMA	INTERRUPT	DMA	INTERRUPT	DMA
MASTER Normal	End of transfer sequence	See Section 18.3.5.2.1.3		See Section 18.3.5.2.1.4.1	See Section 18.3.5.2.1.4.2	See Section 18.3.5.2.1.5.1	See Section 18.3.5.2.1.5.2
	Minimum number of word	1	1	1	1	1	2
	DMA transfer size		N		N		N-1
MASTER Turbo	End of transfer sequence	See Section 18.3.5.2.1.3		See Section 18.3.5.2.1.4.1	See Section 18.3.5.2.1.4.2	See Section 18.3.5.2.1.6.1	See Section 18.3.5.2.1.6.2
	Minimum number of word	1	1	1	1	2	3
	DMA transfer size		N		N		N-2
SLAVE	End of transfer sequence	See Section 18.3.5.2.1.3		See Section 18.3.5.2.1.4.1	See Section 18.3.5.2.1.4.2	See Section 18.3.5.2.1.7	
	Minimum number of word	1	1	1	1	1	1
	DMA transfer size		N		N	N	N

The transfer to execute has a size of N words.

The different sequences can be merged in one process to manage transfers of several types. The end of transfer sequences are described from the start of the channel.

In these sequences, some soft variables are used:

- write_count = 0
- read_count = 0
- channel_enable = FALSE
- last_transfer = FALSE
- last_request = FALSE

They are initialized before starting the channel.

18.3.5.2.1.3 Transmit-and-Receive (Master and Slave)

If the requests are configured in DMA, write_count and read_count are assigned with ‘N’ when the DMA handlers have completed their ‘N’ OCP accesses.

Table 18-212. Transmit-and-Receive (Master and Slave) (Main Process)

Step	Register/Bit Field/Programming Model	Value
Start the channel	MCSPI_CHxCTRL[0] EN	1
Wait for write_count = N **AND** read_count = N		
Stop the channel	MCSPI_CHxCTRL[0] EN	0

Table 18-213. Transmit-and-Receive (Master and Slave) (Interrupt Routine)

Step	Register/Bit Field/Programming Model	Value
Read MCSPI_IRQSTATUS	MCSPI_IRQSTATUS	0x-
Write MCSPI_IRQSTATUS to reset channel status bits	MCSPI_IRQSTATUS[channel x bits]	0b1111
IF: TXx_EMPTY		
Write the transmitter register with data	MCSPI_TXx	0x-
Increment write_count +1		
IF: RXx_FULL		
Read the receiver register	MCSPI_RXx	0x-
Increment read_count +1		
ENDIF		

18.3.5.2.1.4 Transmit-Only (Master and Slave)

18.3.5.2.1.4.1 Based on Interrupt Requests

Table 18-214. Transmit-Only With Interrupts (Master and Slave) (Main Process)

Step	Register/Bit Field/Programming Model	Value
Start the channel	MCSPI_CHxCTRL[0] EN	1
Wait until last transfer = TRUE		
Wait for end of transfer	MCSPI_CHxSTAT[2] EOT	=1
Stop the channel	MCSPI_CHxCTRL[0] EN	0

Table 18-215. Transmit-Only With Interrupts (Master and Slave) (Interrupt Routine)

Step	Register/Bit Field/Programming Model	Value
Read MCSPI_IRQSTATUS	MCSPI_IRQSTATUS	0x-
Write MCSPI_IRQSTATUS to reset channel status bits	MCSPI_IRQSTATUS[channel x bits]	0b1111
IF: TXx_EMPTY **AND** write_count < N		
Write the transmitter register with data	MCSPI_TXx	0x-
Increment write_count +1		
ELSEIF: write_count ≥ N		
last_transfer = TRUE		
ENDIF		

18.3.5.2.1.4.2 Based on DMA Write Requests

When the DMA handler has completed its ‘N’ OCP accesses, write_count is assigned with ‘N’.

Table 18-216. Transmit-Only With DMA (Master and Slave) (Main Process)

Step	Register/Bit Field/Programming Model	Value
Start the channel	MCSPI_CHxCTRL[0] EN	1
Wait until write_count = N		
Disable DMA write request	MCSPI_CHxCONF[14] DMAW	0
Wait until last_transfer = TRUE		
Wait for end of transfer	MCSPI_CHxSTAT[2] EOT	=1
Stop the channel	MCSPI_CHxCTRL[0] EN	0

Table 18-217. Transmit-Only With DMA (Master and Slave) (Interrupt Routine)

Step	Register/Bit Field/Programming Model	Value
Read MCSPI_IRQSTATUS	MCSPI_IRQSTATUS	0x-
Write MCSPI_IRQSTATUS to reset channel status bits	MCSPI_IRQSTATUS[channel x bits]	0b1111
IF: TXx_EMPTY **AND** write_count = N		
last_transfer = TRUE		
ENDIF		

18.3.5.2.1.5 Master Normal Receive-Only

18.3.5.2.1.5.1 Based on Interrupt Requests

Table 18-218. Receive-Only With Interrupt (Master Normal) (Main Process)

Step	Register/Bit Field/Programming Model	Value
Start the channel	MCSPI_CHxCTRL[0] EN	1
Wait until last_request = TRUE		
Stop the channel	MCSPI_CHxCTRL[0] EN	0
Read the receiver register	MCSPI_RXx	0x-
Increment read_count +1		

Table 18-219. Receive-Only With Interrupt (Master Normal) (Interrupt Routine)

Step	Register/Bit Field/Programming Model	Value
Read MCSPI_IRQSTATUS	MCSPI_IRQSTATUS	0x-
Write MCSPI_IRQSTATUS to reset channel status bits	MCSPI_IRQSTATUS[channel x bits]	0b1111
IF: RXx_FULL **AND** read_count = N - 1		
last_request = TRUE		
ELSEIF: read_count ≠ N - 1		
Read the receiver register	MCSPI_RXx	0x-
Increment read_count +1		
ENDIF		

18.3.5.2.1.5.2 Based on DMA Read Requests

When the DMA handler has completed its ‘N-1’ OCP accesses, read_count is assigned with ‘N-1’.

Table 18-220. Receive-Only With DMA (Master Normal) (Main Process)

Step	Register/Bit Field/Programming Model	Value
Start the channel	MCSPI_CHxCTRL[0] EN	1
Wait until read_count = N - 1		
Disable DMA read request	MCSPI_CHxCONF[15] DMAR	0
Wait until last_transfer = TRUE		
Stop the channel	MCSPI_CHxCTRL[0] EN	0
Read the receiver register	MCSPI_RXx	0x-
Increment read_count +1		

Table 18-221. Receive-Only With DMA (Master Normal) (Interrupt Routine)

Step	Register/Bit Field/Programming Model	Value
Read MCSPI_IRQSTATUS	MCSPI_IRQSTATUS	0x-
Write MCSPI_IRQSTATUS to reset channel status bits	MCSPI_IRQSTATUS[channel x bits]	0b1111
IF: RXx_FULL **AND** read_count = N		
last_transfer = TRUE		
ENDIF		

18.3.5.2.1.6 Master Turbo Receive-Only

18.3.5.2.1.6.1 Based on Interrupt Requests

Table 18-222. Receive-Only With Interrupt (Master Turbo) (Main Process)

Step	Register/Bit Field/Programming Model	Value
Start the channel	MCSPI_CHxCTRL[0] EN	1
Wait until channel_enable = TRUE		
Wait until last_transfer = TRUE		
Wait for end of transfer	MCSPI_CHxSTAT[2] EOT	=1
Stop the channel	MCSPI_CHxCTRL[0] EN	0
Wait until channel_enable = FALSE		

Table 18-223. Receive-Only With Interrupt (Master Turbo) (Interrupt Routine)

Step	Register/Bit Field/Programming Model	Value
Read MCSPI_IRQSTATUS	MCSPI_IRQSTATUS	0x-
Write MCSPI_IRQSTATUS to reset channel status bits	MCSPI_IRQSTATUS[channel x bits]	0b1111
IF: RXx_FULL		
IF: read_count = N - 2		
last_transfer = TRUE		
Wait until channel_enable = FALSE		
ENDIF		
IF: read_count < N		
Read the receiver register	MCSPI_RXx	0x-
Increment read_count +1		
ENDIF		
ENDIF		

18.3.5.2.1.6.2 Based on DMA Read Requests

When the DMA handler has completed its ‘N-2’ OCP accesses read_count is assigned with ‘N-2‘.

Table 18-224. Receive-Only With DMA (Master Turbo) (Main Process)

Step	Register/Bit Field/Programming Model	Value
Start the channel	MCSPI_CHxCTRL[0] EN	1
Wait until channel_enable = TRUE		
Wait until read_count = TRUE		
Disable DMA read request	MCSPI_CHxCONF[15] DMAR	0

Table 18-224. Receive-Only With DMA (Master Turbo) (Main Process) (continued)

Step	Register/Bit Field/Programming Model	Value
Wait until last_transfer = TRUE		
Wait for end of transfer	MCSPI_CHxSTAT[2] EOT	=1
Stop the channel	MCSPI_CHxCTRL[0] EN	0
Wait until channel_enable = FALSE		

Table 18-225. Receive-Only With DMA (Master Turbo) (Interrupt Routine)

Step	Register/Bit Field/Programming Model	Value
Read MCSPI_IRQSTATUS	MCSPI_IRQSTATUS	0x-
Write MCSPI_IRQSTATUS to reset channel status bits	MCSPI_IRQSTATUS[channel x bits]	0b1111
IF: RXx_FULL		
IF: read_count = N - 2		
last_transfer = TRUE		
Wait until channel_enable = FALSE		
ENDIF		
IF: read_count < N		
Read the receiver register	MCSPI_RXx	0x-
Increment read_count +1		
ENDIF		
ENDIF		

18.3.5.2.1.7 Slave Receive-Only

If the requests are configured in DMA, read_count is assigned with ‘N’ when the DMA handler has completed its ‘N‘ OCP accesses.

Table 18-226. Receive-Only (Slave) (Main Process)

Step	Register/Bit Field/Programming Model	Value
Start the channel	MCSPI_CHxCTRL[0] EN	1
Wait until read_count = N		
Stop the channel	MCSPI_CHxCTRL[0] EN	0

Table 18-227. Receive-Only (Slave) (Interrupt Routine)

Step	Register/Bit Field/Programming Model	Value
Read MCSPI_IRQSTATUS	MCSPI_IRQSTATUS	0x-
Write MCSPI_IRQSTATUS to reset channel status bits	MCSPI_IRQSTATUS[channel x bits]	0b1111
IF: RXx_FULL		
Read the receiver register	MCSPI_RXx	0x-
Increment read_count +1		
ENDIF		

18.3.5.2.1.8 Transfer Procedures With FIFO

These flows describe the transfer with FIFO.

The McSPI module allows the transfer of one or several words, according to different modes:

- MASTER Normal, MASTER Turbo, SLAVE
- TRANSMIT–RECEIVE, TRANSMIT-ONLY, RECEIVE-ONLY
- Write and Read requests: IRQ, DMA

For all these flows, the host process contains the main process and the interrupt routine. This routine is called on the IRQ signals or by an internal call if the module is used in polling mode.

For more information, see Section 18.3.4.6, *FIFO Buffer Management*.

Table 18-228. FIFO Mode Common Sequence (Master) (Main Process)

Step	Register/Bit Field/Programming Model	Value
Write MCSPI_IRQSTATUS to reset channel status bits	MCSPI_IRQSTATUS	1
Write MCSPI_IRQENABLE to enable interrupts	MCSPI_IRQENABLE	1
Write MCSPI_CHxCONF to configure the channel	MCSPI_CHxCONF	0x-
Write MCSPI_XFERLEVEL	MCSPI_XFERLEVEL	0x-
Start the channel	MCSPI_CHxCTRL[0] EN	1
IF: Receive only		
Wait for the write request (TX empty or DMA write)		
Write for the transmitter register with data	MCSPI_TXx	0x-
ENDIF		
Wait for the host event for end of transfer		
Stop the channel	MCSPI_CHxCTRL[0] EN	0

18.3.5.2.1.8.1 Common Transfer Sequence in FIFO Mode

This flow describes the host sequence for a transfer of any type defined in Section 18.3.5.2.1.8, *Transfer Procedures With FIFO*.

In multi-channel, only one channel can use the FIFO.

Before enabling the FIFO for a channel (MCSPI_CHxCONF[28] FFER and MCSPI_CHxCONF[27] FFEW bits), the host must check that the FIFO is not enabled for another channel, even if these channels are not used.

In transmit-and-receive mode, the FIFO can be enabled for write or read request only, without FIFO for the other request.

In Slave mode, the channel 0 only can be activated. The correct SPIEN line is chosen in MCSPI_CHxCONF[22:21] SPIENSLV bits (where x = 0).

The McSPI module can start the transfer only when the first write request has been released by writing the MCSPI_TXx register, even in receive-only mode (only one write request occurs in this case).

18.3.5.2.1.8.2 End of Transfer Sequences in FIFO Mode

Table 18-229 summarizes the type of end of transfer per transfer mode and gives a reference to the appropriate section for details.

Table 18-229. End of Transfer Sequences in FIFO Mode

Word count	TRANSMIT AND RECEIVE	TRANSMIT-ONLY	RECEIVE-ONLY
Yes	See Figure 18-66	See Figure 18-68	See Figure 18-69
No	See Figure 18-67	See Figure 18-68	See Figure 18-70

The end of transfer sequences are described from the start of the channel.

In these sequences, some soft variables are used:

- write_count = N
- read_count = N
- last_request = FALSE

They are initialized before starting the channel.

18.3.5.2.1.8.3 Transmit-and-Receive With Word Count

Figure 18-66 shows the flow of a transfer in transmit-and-receive mode, with word count.

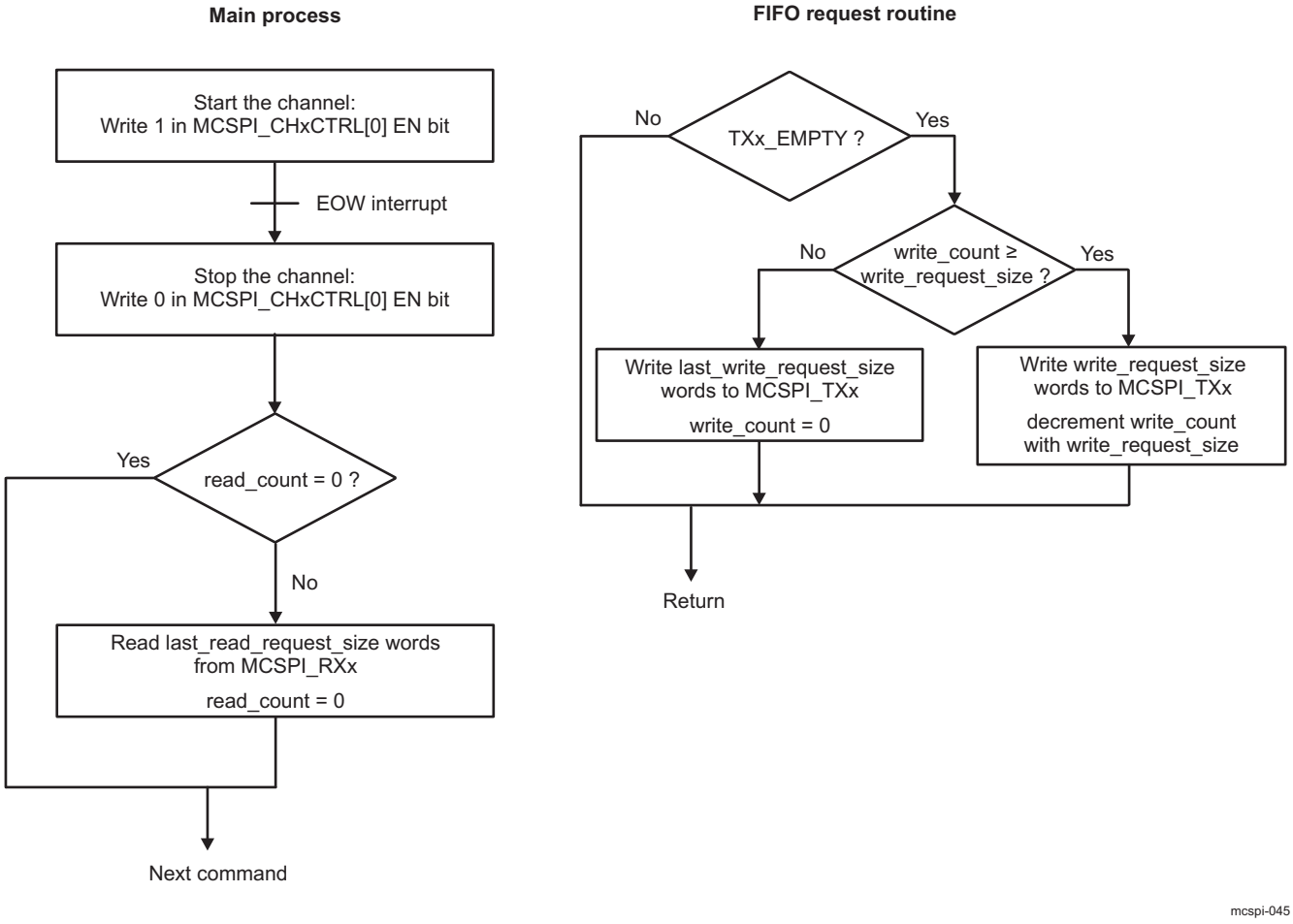


Figure 18-66. FIFO Mode Transmit-and-Receive With Word Count (Master)

18.3.5.2.1.8.4 Transmit-and-Receive Without Word Count

Figure 18-67 shows the flow of a transfer in transmit-and-receive mode, without word count.

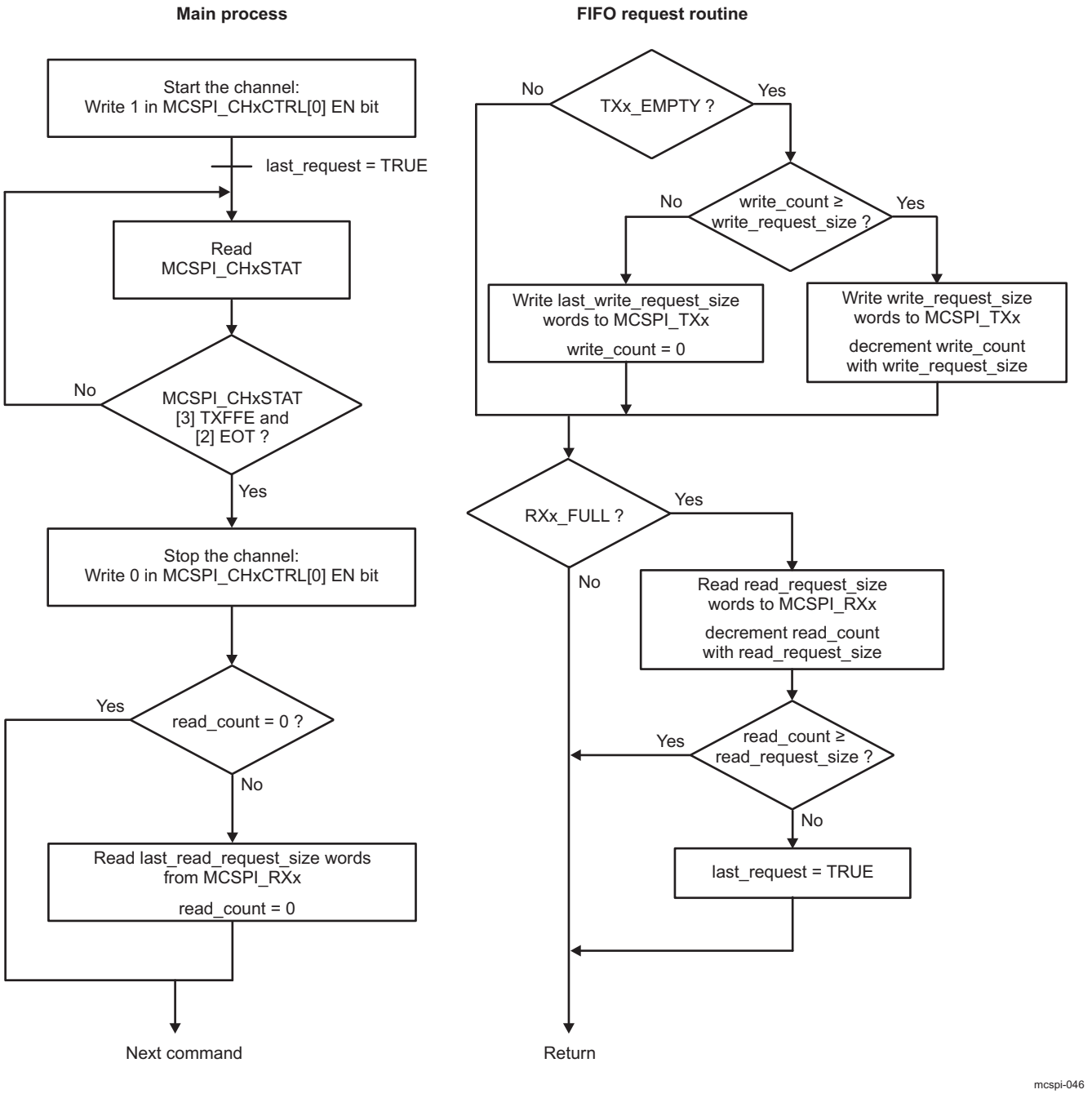


Figure 18-67. FIFO Mode Transmit-and-Receive Without Word Count (Master)

18.3.5.2.1.8.5 Transmit-Only

Figure 18-68 shows the flow of a transfer in transmit-only mode, with or without word count. The difference between word count enabled or not is just on the condition after starting the channel:

- word count enable: wait for EOW interrupt
- word count disable: wait for write_count = 0

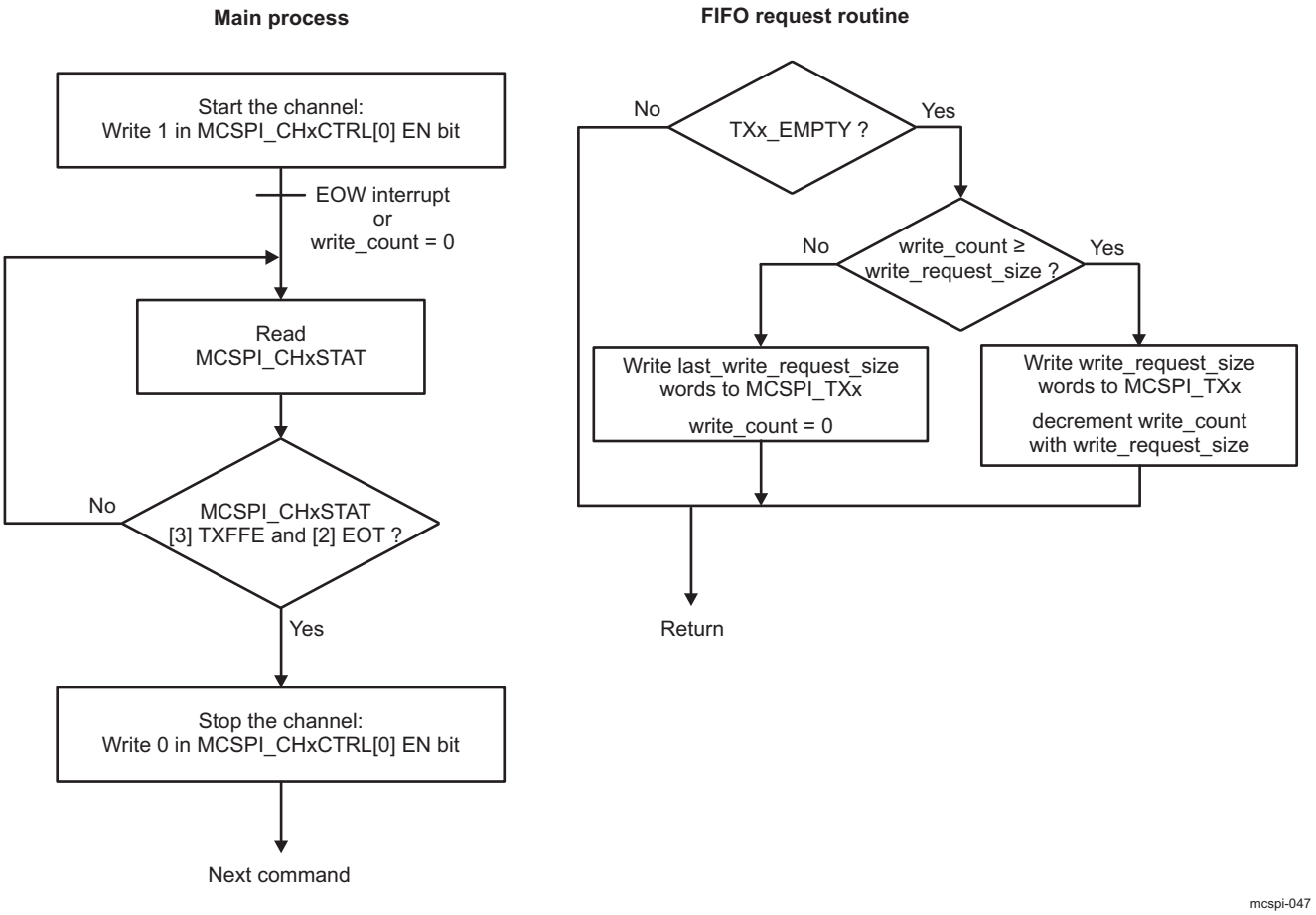


Figure 18-68. FIFO Mode Transmit-Only (Master)

18.3.5.2.1.8.6 Receive-Only With Word Count

Figure 18-69 shows the flow of a transfer in receive-only mode, with word count.

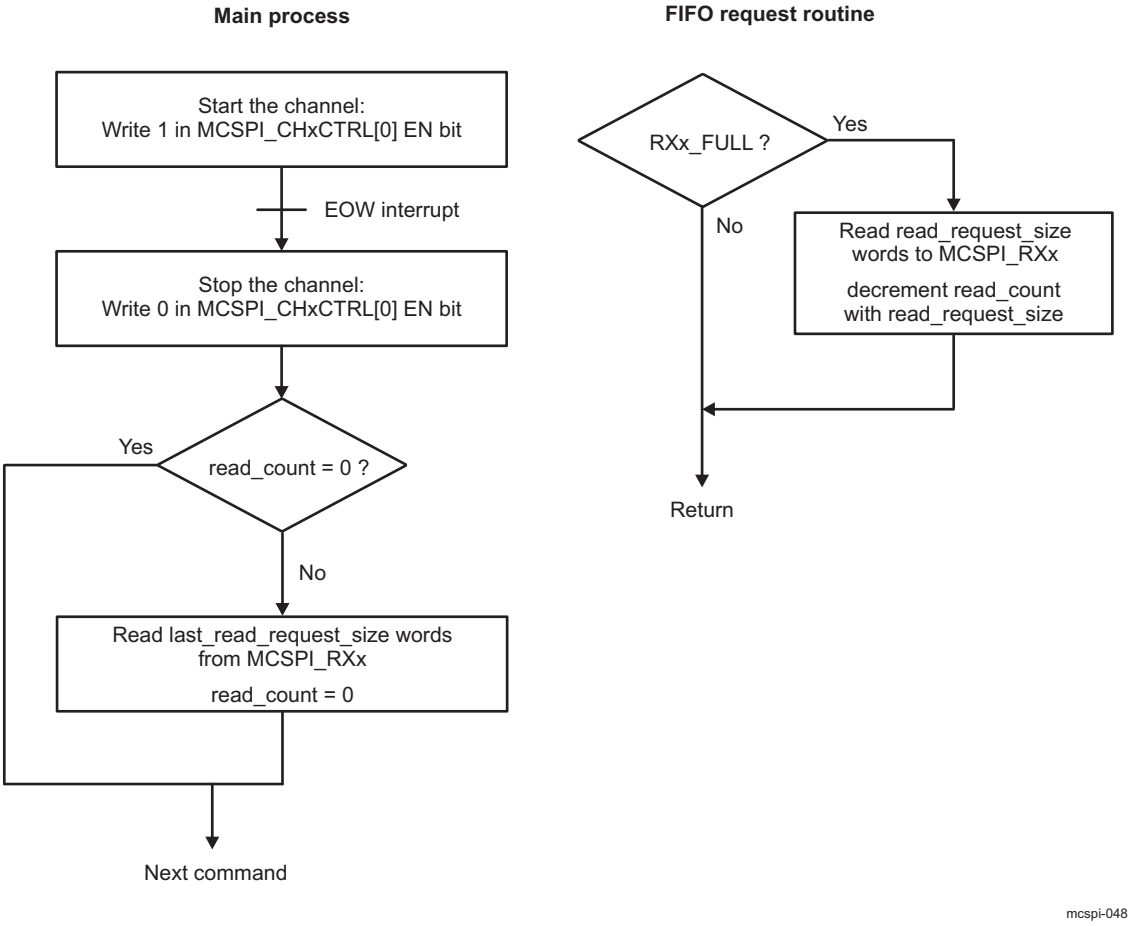


Figure 18-69. FIFO Mode Receive-Only With Word Count (Master)

18.3.5.2.1.8.7 Receive-Only Without Word Count

Figure 18-70 shows the flow of a transfer in receive-only mode, without word count.

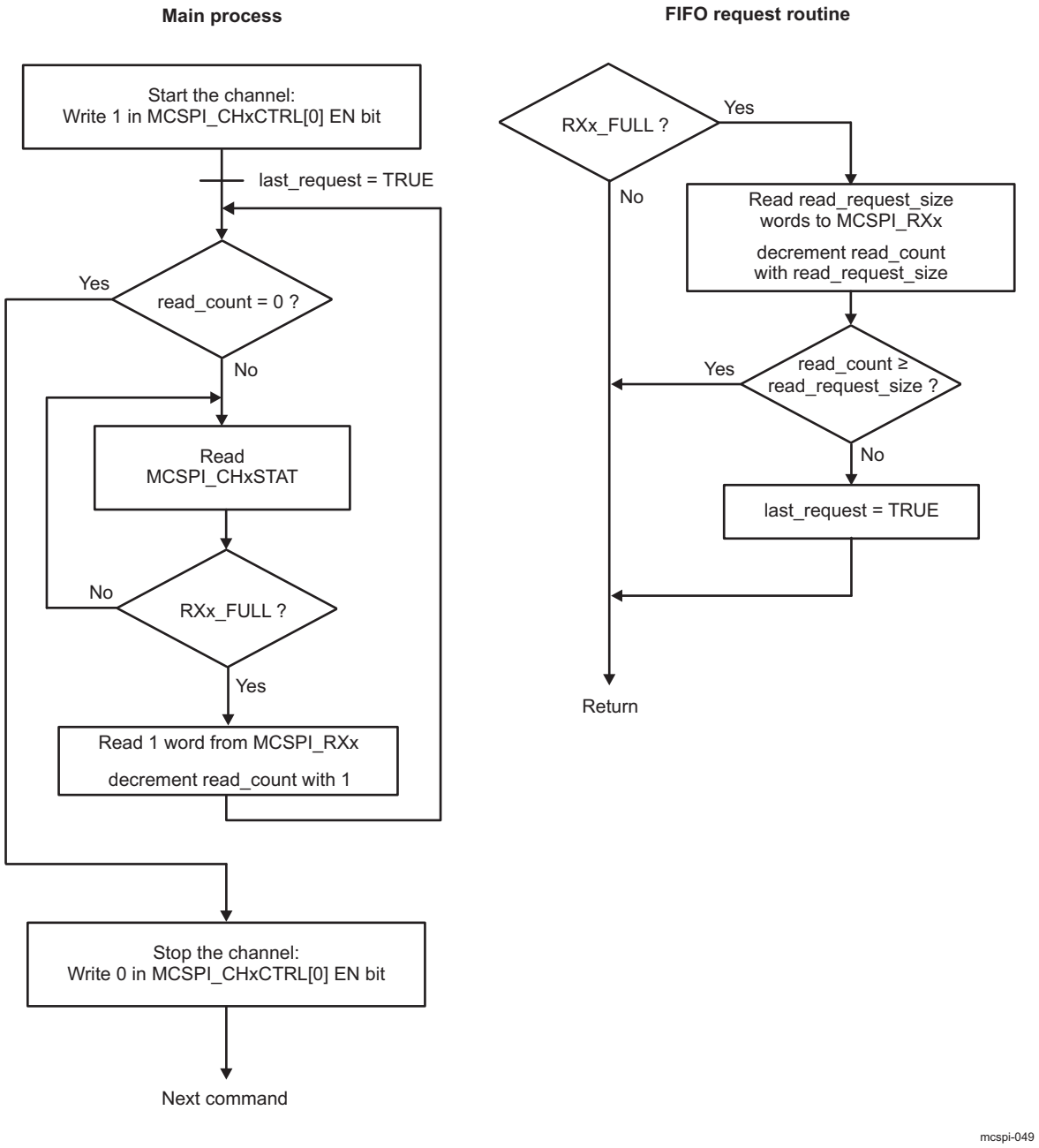


Figure 18-70. FIFO Mode Receive-Only Without Word Count (Master)

18.3.5.3 Common Transfer Procedures Without FIFO – Polling Method

18.3.5.3.1 Receive-Only Procedure – Polling Method

Table 18-230 lists the receive-only procedure using the polling method.

Table 18-230. Receive-Only Procedure – Polling Method

Step	Register/Bit Field/Programming Model	Value
Configure the channel according to the mode	See Table 18-207.	
Start the channel.	MCSPI_CHxCTRL[0] EN	1

Table 18-230. Receive-Only Procedure – Polling Method (continued)

Step	Register/Bit Field/Programming Model	Value
Wait for end-of-transfer.	MCSPI_CHxSTAT[2] EOT	=1
Read the receiver register.	MCSPI_RXx	0x-
Stop the channel if no more data is expected.	MCSPI_CHxCTRL[0] EN	0

18.3.5.3.2 Receive-Only Procedure – Interrupt Method

Table 18-231 lists the receive-only procedure using the interrupt method.

Table 18-231. Receive-Only Procedure – Interrupt Method

Step	Register/Bit Field/Programming Model	Value
Configure the channel according to the mode.	See Table 18-207.	
Start the channel.	MCSPI_CHxCTRL[0] EN	1
Enable the interrupt for the receiver register.	MCSPI_IRQENABLE[2] RX_FULL_ENABLE	1
Wait for interrupt.		
Read the status register.	MCSPI_IRQSTATUS[2] RX_FULL	1
Disable the interrupt if no more data is expected.	MCSPI_IRQENABLE[2] RX_FULL_ENABLE	0
Stop the channel if no more data is expected.	MCSPI_CHxCTRL[0] EN	0
Read the receiver register.	MCSPI_RXx	0x-

18.3.5.3.3 Transmit-Only Procedure – Polling Method

Table 18-232 lists the transmit-only procedure using the polling method.

Table 18-232. Transmit-Only Procedure – Polling Method

Step	Register/Bit Field/Programming Model	Value
Configure the channel according to the mode.	See Table 18-208	
Start the channel.	MCSPI_CHxCTRL[0] EN	1
Write the transmitter register with data.	MCSPI_TXx	0x-
Wait until end of transfer?	MCSPI_CHxSTAT[2] EOT	=1
Stop the channel.	MCSPI_CHxCTRL[0] EN	0

18.3.5.3.4 Transmit-and-Receive Procedure – Polling Method

Table 18-232 lists the transmit-and-receive procedure using the polling method.

Table 18-233. Transmit-and-Receive Procedure – Polling Method

Step	Register/Bit Field/Programming Model	Value
Configure the channel according to the mode.	See Table 18-209.	
Start the channel.	MCSPI_CHxCTRL[0] EN	1
Write the transmitter register with data.	MCSPI_TXx	0x-
Wait until transmit/receive word?	MCSPI_CHxSTAT[2] EOT	=1
Stop the channel.	MCSPI_CHxCTRL[0] EN	0
Read the receiver register.	MCSPI_TXx	0x-

18.3.6 McSPI Register Manual

18.3.6.1 McSPI Instance Summary

Table 18-234. McSPI Instance Summary

Module Name	Base Address	Size
McSPI1	0x4809 8000	4 KiB
McSPI2	0x4809 A000	4 KiB
McSPI3	0x480B 8000	4 KiB
McSPI4	0x480B A000	4 KiB

18.3.6.2 McSPI Registers

18.3.6.2.1 McSPI Register Summary

Table 18-235 lists the McSPI registers. Each register is 32 bits wide.

Table 18-235. McSPI Register Summary

Register	Type	Offset Address	McSPI1 Physical Address	McSPI2 Physical Address	McSPI3 Physical Address	McSPI4 Physical Address
MCSPI_HL_REV	RW	0x00	0x4809 8000	0x4809 A000	0x480B 8000	0x480B A000
MCSPI_HL_HWINFO	RW	0x04	0x4809 8004	0x4809 A004	0x480B 8004	0x480B A004
MCSPI_HL_SYSCONFIG	RW	0x10	0x4809 8010	0x4809 A010	0x480B 8010	0x480B A010
MCSPI_REVISION	R	0x100	0x4809 8100	0x4809 A100	0x480B 8100	0x480B A100
MCSPI_SYSCONFIG	RW	0x110	0x4809 8110	0x4809 A110	0x480B 8110	0x480B A110
MCSPI_SYSSTATUS	R	0x114	0x4809 8114	0x4809 A114	0x480B 8114	0x480B A114
MCSPI_IRQSTATUS	RW	0x118	0x4809 8118	0x4809 A118	0x480B 8118	0x480B A118
MCSPI_IRQENABLE	RW	0x11C	0x4809 811C	0x4809 A11C	0x480B 811C	0x480B A11C
MCSPI_WAKEUPENABLE	RW	0x120	0x4809 8120	0x4809 A120	0x480B 8120	0x480B A120
MCSPI_SYST	RW	0x124	0x4809 8124	0x4809 A124	0x480B 8124	0x480B A124
MCSPI_MODULCTRL	RW	0x128	0x4809 8128	0x4809 A128	0x480B 8128	0x480B A128
MCSPI_CHxCONF (1)	RW	0x12C + (0x14 * x)	0x4809 812C + (0x14 * x)	0x4809 A12C + (0x14 * x)	0x480B 812C + (0x14 * x)	0x480B A12C + (0x14 * x)
MCSPI_CHxSTAT (1)	R	0x130 + (0x14 * x)	0x4809 8130 + (0x14 * x)	0x4809 A130 + (0x14 * x)	0x480B 8130 + (0x14 * x)	0x480B A130 + (0x14 * x)
MCSPI_CHxCTRL (1)	RW	0x134 + (0x14 * x)	0x4809 8134 + (0x14 * x)	0x4809 A134 + (0x14 * x)	0x480B 8134 + (0x14 * x)	0x480B A134 + (0x14 * x)
MCSPI_TXx (1)	RW	0x138 + (0x14 * x)	0x4809 8138 + (0x14 * x)	0x4809 A138 + (0x14 * x)	0x480B 8138 + (0x14 * x)	0x480B A138 + (0x14 * x)
MCSPI_RXx (1)	R	0x13C + (0x14 * x)	0x4809 813C + (0x14 * x)	0x4809 A13C + (0x14 * x)	0x480B 813C + (0x14 * x)	0x480B A13C + (0x14 * x)
MCSPI_XFERLEVEL	RW	0x17C	0x4809 817C	0x4809 A17C	0x480B 817C	0x480B A17C
MCSPI_DAFTX	RW	0x0000 0180	0x4809 8180	0x4809 A180	0x480B 8180	0x480B A180
MCSPI_DAFRX	RW	0x0000 01A0	0x4809 81A0	0x4809 A1A0	0x480B 81A0	0x480B A1A0

(1) x = 0 to 3 for McSPI1
x = 0 to 1 for McSPI2 and McSPI3
x = 0 for McSPI4

18.3.6.2.2 McSPI Register Description

Table 18-236 through Table 18-272 describe the individual McSPI register bits.

Table 18-236. MCSPI_HL_REV

Address Offset	0x00		
Physical Address	0x4809 8000 0x4809 A000 0x480B 8000 0x480B A000	**Instance**	McSPI1 McSPI2 McSPI3 McSPI4
Description	McSPI module revision identifier Used by software to track features, bugs, and compatibility		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REVISION																															

Bits	Field Name	Description	Type	Reset
31:0	REVISION	McSPI Module Revision	R	TI internal data

Table 18-237. Register Call Summary for Register MCSPI_HL_REV

Multichannel Serial Peripheral Interface
- McSPI Register Summary: [0]

Table 18-238. MCSPI_HL_HWINFO

Address Offset	0x04		
Physical Address	0x4809 8004 0x4809 A004 0x480B 8004 0x480B A004	**Instance**	McSPI1 McSPI2 McSPI3 McSPI4
Description	Information about the module's hardware configuration.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																									RETMODE	FFNBYTE					USEFIFO

Bits	Field Name	Description	Type	Reset
31:7	RESERVED	Reserved These bits are initialized to 0, and writes to them are ignored.	R	0x0000000
6	RETMODE	Retention Mode. This bit field indicates whether the retention mode is supported. 0x0: Retention mode disabled 0x1: Retention mode enabled	R	0
5:1	FFNBYTE	FIFO number of bytes parameter Read 0x1: FIFO 16 bytes depth Read 0x2: FIFO 32 bytes depth Read 0x4: FIFO 64 bytes depth Read 0x8: FIFO 128 bytes depth Read 0x10: FIFO 256 bytes depth	R	0x8

Bits	Field Name	Description	Type	Reset
0	USEFIFO	Use of a FIFO enable. This bit indicates if a FIFO is integrated within controller design with its management. Read 0x0: FIFO not implemented in design Read 0x1: FIFO and its management implemented in design	R	1

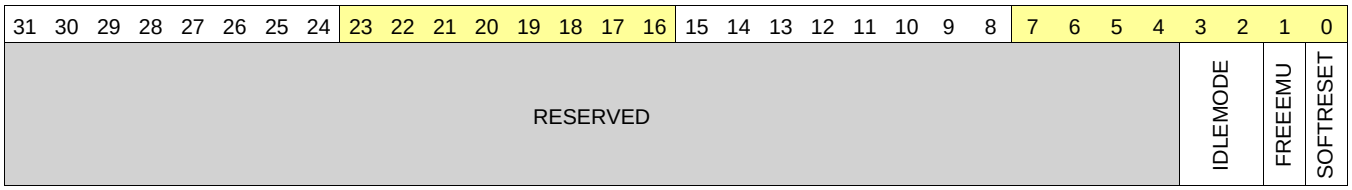
Table 18-239. Register Call Summary for Register MCSPI_HL_HWINFO

Multichannel Serial Peripheral Interface

- McSPI Register Summary: [0]

Table 18-240. MCSPI_HL_SYSCONFIG

Address Offset	0x10		
Physical Address	0x4809 8010 0x4809 A010 0x480B 8010 0x480B A010	**Instance**	McSPI1 McSPI2 McSPI3 McSPI4
Description	Clock management configuration		
Type	RW		



Bits	Field Name	Description	Type	Reset
31:4	RESERVED		R	0x0000000
3:2	IDLEMODE	Configuration of the local target state management mode. By definition, target can handle read/write transaction as long as it is out of IDLE state. 0x0: Force-idle mode: local target's IDLE state follows (acknowledges) the system's IDLE requests unconditionally, that is, regardless of the module's internal requirements. Backup mode, for debug only. 0x1: No-idle mode: local target never enters IDLE state. Backup mode, for debug only. 0x2: Smart-idle mode: local target's IDLE state eventually follows (acknowledges) the system's IDLE requests, depending on the IP module's internal requirements. Module shall not generate (IRQ- or DMA-request-related) wake-up events. 0x3: Smart-idle wake-up-capable mode: local target's IDLE state eventually follows (acknowledges) the system's IDLE requests, depending on the module's internal requirements. Module may generate (IRQ- or DMA-request-related) wake-up events when in IDLE state. Mode is relevant only if the appropriate IP module "swake-up" output(s) is (are) implemented.	RW	0x2
1	FREEEMU	Sensitivity to emulation (debug) suspend input signal. 0x0: Module is sensitive to emulation suspend. 0x1: Module is not sensitive to emulation suspend.	RW	0

Bits	Field Name	Description	Type	Reset
0	SOFTRESET	Software reset. (Optional) Write 0x0: No action Read 0x0: Reset done, no pending action Read 0x1: Reset (software or other) ongoing Write 0x1: Initiate software reset	RW	0

Table 18-241. Register Call Summary for Register MCSPI_HL_SYSCONFIG

Multichannel Serial Peripheral Interface
- McSPI Register Summary: [0]

Table 18-242. MCSPI_REVISION

Address Offset	0x100		
Physical Address	0x4809 8100 0x4809 A100 0x480B 8100 0x480B A100	**Instance**	McSPI1 McSPI2 McSPI3 McSPI4
Description	This register contains the McSPI revision number.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								REVISION							

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Reads return 0	R	0x0
7:0	REVISION	McSPI core revision [7:4] Major revision [3:0] Minor revision Examples: 0x10 for 1.0, 0x21 for 2.1	R	TI Internal data

Table 18-243. Register Call Summary for Register MCSPI_REVISION

Multichannel Serial Peripheral Interface
- McSPI Register Summary: [0]

Table 18-244. MCSPI_SYSCONFIG

Address Offset	0x110		
Physical Address	0x4809 8110 0x4809 A110 0x480B 8110 0x480B A110	**Instance**	McSPI1 McSPI2 McSPI3 McSPI4
Description	This register allows controlling various parameters of the configuration interface and is not affected by software reset.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						CLOCKACTIVITY		RESERVED			SIDLEMODE		ENAWAKEUP	SOFTRESET	AUTOIDLE

Bits	Field Name	Description	Type	Reset
31:10	RESERVED	Reads return 0	RW	0x000000
9:8	CLOCKACTIVITY	Clocks activity during wake-up mode period 0x0: Interface and functional clocks may be switched off. 0x1: Interface clock is maintained. Functional clock may be switched off. 0x2: Functional clock is maintained. Interface clock may be switched off. 0x3: Interface and functional clocks are maintained.	RW	0x0
7:5	RESERVED	Reads returns 0	RW	0x0
4:3	SIDLEMODE	Power management 0x0: If an IDLE request is detected, the McSPI acknowledges it unconditionally and goes in inactive mode. Interrupt, DMA requests and wake-up lines are unconditionally deasserted and the module wake-up capability is deactivated even if the [2] ENAWAKEUP bit is set. 0x1: If an IDLE request is detected, the request is ignored and the module does not switch to wake-up mode, and keeps on behaving normally. 0x2: If an IDLE request is detected, the module will switch to wake-up mode based on its internal activity, and the wake-up capability can be used if the bit [2] ENAWAKEUP is set. 0x3: Reserved - do not use.	RW	0x2
2	ENAWAKEUP	Wake-up feature control 0x0: Wake-up capability is disabled. 0x1: Wake-up capability is enabled.	RW	1
1	SOFTRESET	Software reset. During reads it always returns 0. 0x0: (write) Normal mode 0x1: (write) Set this bit to 1 to trigger a module reset. The bit is automatically reset by the hardware.	RW	0
0	AUTOIDLE	Internal interface clock-gating strategy 0x0: Interface clock is free-running. 0x1: Automatic interface clock gating strategy is applied, based on the interface activity.	RW	1

Table 18-245. Register Call Summary for Register MCSPI_SYSCONFIG

Multichannel Serial Peripheral Interface

- Reset: [0][1]
- Normal Mode: [2]
- Idle Mode: [3][4][5][6][7][8]
- McSPI Global Initialization: [9][10][11]
- McSPI Register Summary: [12]
- McSPI Register Description: [13][14]

Table 18-246. MCSPI_SYSSTATUS

Address Offset	0x114		
Physical Address	0x4809 8114 0x4809 A114 0x480B 8114 0x480B A114	**Instance**	McSPI1 McSPI2 McSPI3 McSPI4
Description	This register provides status information about the module excluding the interrupt status information.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															RESETDONE

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved for future module specific status information. Read returns 0.	R	0x0000 0000
0	RESETDONE	Internal reset monitoring Read 0x0: Internal module reset is ongoing Read 0x1: Reset completed	R	0

Table 18-247. Register Call Summary for Register MCSPI_SYSSTATUS

Multichannel Serial Peripheral Interface

- McSPI Global Initialization: [0]
- McSPI Register Summary: [1]

Table 18-248. MCSPI_IRQSTATUS

Address Offset	0x118		
Physical Address	0x4809 8118 0x4809 A118 0x480B 8118 0x480B A118	**Instance**	McSPI1 McSPI2 McSPI3 McSPI4
Description	The interrupt status regroups all the status of the module internal events that can generate an interrupt.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														EOW	WKS	RESERVED	RX3_FULL	TX3_UNDERFLOW	TX3_EMPTY	RESERVED	RX2_FULL	TX2_UNDERFLOW	TX2_EMPTY	RESERVED	RX1_FULL	TX1_UNDERFLOW	TX1_EMPTY	RX0_OVERFLOW	RX0_FULL	TX0_UNDERFLOW	TX0_EMPTY

Bits	Field Name	Description	Type	Reset
31:18	RESERVED	Reads return 0	RW	0x0000
17	EOW	End of word count event when a channel is enabled using the FIFO buffer and the channel had sent the number of SPI word defined by MCSPI_XFERLEVEL[31:16] WCNT. Write 0x0: Event status bit unchanged Read 0x0: Event false Read 0x1: Event is pending Write 0x1: Event status bit is reset	RW W1toClr	0
16	WKS	Wake-up event in slave mode when an active control signal is detected on the SPIEN line programmed in the field MCSPI_CHxCONF[22:21] SPIENSLV Read 0x0: Event false Write 0x0: Event status bit unchanged Write 0x1: Event status bit is reset Read 0x1: Event is pending	RW W1toClr	0
15	RESERVED	Reads returns 0	RW	0
14	RX3_FULL	Receiver register is full or almost full. Only when Channel 3 is enabled Read 0x0: Event false Write 0x0: Event status bit unchanged Write 0x1: Event status bit is reset Read 0x1: Event is pending	RW W1toClr	0
13	TX3_UNDERFLOW	Transmitter register underflow. Only when Channel 3 is enabled. The transmitter register is empty (not updated by host or DMA with new data) before its time slot assignment. Exception: No TX_underflow event when no data has been loaded into the transmitter register since channel has been enabled. Read 0x0: Event false Write 0x0: Event status bit unchanged Write 0x1: Event status bit is reset Read 0x1: Event is pending	RW W1toClr	0
12	TX3_EMPTY	Transmitter register is empty or almost empty. Note: Enabling the channel automatically rises this event. Read 0x0: Event false Write 0x0: Event status bit unchanged Write 0x1: Event status bit is reset Read 0x1: Event is pending	RW W1toClr	0
11	RESERVED	Reads returns 0.	RW	0
10	RX2_FULL	Receiver register full or almost full. Channel 2 Read 0x0: Event false Write 0x0: Event status bit unchanged Write 0x1: Event status bit is reset Read 0x1: Event is pending	RW W1toClr	0
9	TX2_UNDERFLOW	Transmitter register underflow. Channel 2 Read 0x0: Event false Write 0x0: Event status bit unchanged Write 0x1: Event status bit is reset Read 0x1: Event is pending	RW W1toClr	0

Bits	Field Name	Description	Type	Reset
8	TX2_EMPTY	Transmitter register empty or almost empty. Channel 2 Read 0x0: Event false Write 0x0: Event status bit unchanged Write 0x1: Event status bit is reset Read 0x1: Event is pending	RW W1toClr	0
7	RESERVED	Reads returns 0	RW	0
6	RX1_FULL	Receiver register full or almost full. Channel 1 Read 0x0: Event false Write 0x0: Event status bit unchanged Write 0x1: Event status bit is reset Read 0x1: Event is pending	RW W1toClr	0
5	TX1_UNDERFLOW	Transmitter register underflow. Channel 1 Read 0x0: Event false Write 0x0: Event status bit unchanged Write 0x1: Event status bit is reset Read 0x1: Event is pending	RW W1toClr	0
4	TX1_EMPTY	Transmitter register empty or almost empty. Channel 1 Read 0x0: Event false Write 0x0: Event status bit unchanged Write 0x1: Event status bit is reset Read 0x1: Event is pending	RW W1toClr	0
3	RX0_OVERFLOW	Receiver register overflow (slave mode only). Channel 0 Read 0x0: Event false Write 0x0: Event status bit unchanged Write 0x1: Event status bit is reset Read 0x1: Event is pending	RW W1toClr	0
2	RX0_FULL	Receiver register full or almost full. Channel 0 Read 0x0: Event false Write 0x0: Event status bit unchanged Write 0x1: Event status bit is reset Read 0x1: Event is pending	RW W1toClr	0
1	TX0_UNDERFLOW	Transmitter register underflow. Channel 0 Read 0x0: Event false Write 0x0: Event status bit unchanged Write 0x1: Event status bit is reset Read 0x1: Event is pending	RW W1toClr	0
0	TX0_EMPTY	Transmitter register empty or almost empty. Channel 0 Read 0x0: Event false Write 0x0: Event status bit unchanged Write 0x1: Event status bit is reset Read 0x1: Event is pending	RW W1toClr	0

Table 18-249. Register Call Summary for Register MCSPI_IRQSTATUS

Multichannel Serial Peripheral Interface

- Master Transmit-and-Receive Mode (Full Duplex): [0]
- Master Transmit-Only Mode (Half Duplex): [1]
- Master Receive-Only Mode (Half Duplex): [2][3][4][5]
- Single-Channel Master Mode: [6][7]
- Buffer Almost Full: [8][9]
- Buffer Almost Empty: [10][11]
- End of Transfer Management: [12]
- Interrupts: [13]
- Interrupt Events in Master Mode: [14][15][16][17][18][19]
- Interrupt Events in Slave Mode: [20][21][22][23][24][25][26]
- Interrupt-Driven Operation: [27][28]
- Polling: [29][30]
- Idle Mode: [31][32][33]
- McSPI Operational Modes: [34][35][36][37][38][39][40][41][42][43][44][45][46][47][48][49][50][51][52][53][54][55][56][57][58][59][60][61][62][63][64][65][66][67][68][69][70][71][72]
- Receive-Only Procedure – Interrupt Method: [73]
- McSPI Register Summary: [74]
- McSPI Register Description: [75][76][77]

Table 18-250. MCSPI_IRQENABLE

Address Offset	0x11C		
Physical Address	0x4809 811C 0x4809 A11C 0x480B 811C 0x480B A11C	**Instance**	McSPI1 McSPI2 McSPI3 McSPI4
Description	This register allows enabling/disabling of the module internal sources of interrupt, on an event-by-event basis.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														EOW_ENABLE	WKE	RESERVED	RX3_FULL_ENABLE	TX3_UNDERFLOW_ENABLE	TX3_EMPTY_ENABLE	RESERVED	RX2_FULL_ENABLE	TX2_UNDERFLOW_ENABLE	TX2_EMPTY_ENABLE	RESERVED	RX1_FULL_ENABLE	TX1_UNDERFLOW_ENABLE	TX1_EMPTY_ENABLE	RX0_OVERFLOW_ENABLE	RX0_FULL_ENABLE	TX0_UNDERFLOW_ENABLE	TX0_EMPTY_ENABLE

Bits	Field Name	Description	Type	Reset
31:18	RESERVED	Reads return 0.	RW	0x0000
17	EOW_ENABLE	End of Word count Interrupt Enable. 0x0: Interrupt disabled 0x1: Interrupt enabled	RW	0
16	WKE	Wake-up event interrupt enable in slave mode when an active control signal is detected on the SPIEN line programmed in the MCSPI_CHxCONF[22:21] SPIENSLV bits 0x0: Interrupt disabled 0x1: Interrupt enabled	RW	0
15	RESERVED	Reads returns 0.	RW	0

Bits	Field Name	Description	Type	Reset
14	RX3_FULL_ENABLE	Receiver register Full Interrupt Enable. Channel 3 0x0: Interrupt disabled 0x1: Interrupt enabled	RW	0
13	TX3_UNDERFLOW_ENABLE	Transmitter register Underflow Interrupt Enable. Channel 3 0x0: Interrupt disabled 0x1: Interrupt enabled	RW	0
12	TX3_EMPTY_ENABLE	Transmitter register Empty Interrupt Enable. Channel 3 0x0: Interrupt disabled 0x1: Interrupt enabled	RW	0
11	RESERVED	Reads return 0.	RW	0
10	RX2_FULL_ENABLE	Receiver register Full Interrupt Enable. Channel 2 0x0: Interrupt disabled 0x1: Interrupt enabled	RW	0
9	TX2_UNDERFLOW_ENABLE	Transmitter register Underflow Interrupt Enable. Channel 2 0x0: Interrupt disabled 0x1: Interrupt enabled	RW	0
8	TX2_EMPTY_ENABLE	Transmitter register Empty Interrupt Enable. Channel 2 0x0: Interrupt disabled 0x1: Interrupt enabled	RW	0
7	RESERVED	Reads return 0.	RW	0
6	RX1_FULL_ENABLE	Receiver register Full Interrupt Enable. Channel 1 0x0: Interrupt disabled 0x1: Interrupt enabled	RW	0
5	TX1_UNDERFLOW_ENABLE	Transmitter register Underflow Interrupt Enable. Channel 1 0x0: Interrupt disabled 0x1: Interrupt enabled	RW	0
4	TX1_EMPTY_ENABLE	Transmitter register Empty Interrupt Enable. Channel 1 0x0: Interrupt disabled 0x1: Interrupt enabled	RW	0
3	RX0_OVERFLOW_ENABLE	Receiver register Overflow Interrupt Enable. Channel 0 0x0: Interrupt disabled 0x1: Interrupt enabled	RW	0
2	RX0_FULL_ENABLE	Receiver register Full Interrupt Enable. Channel 0 0x0: Interrupt disabled 0x1: Interrupt enabled	RW	0
1	TX0_UNDERFLOW_ENABLE	Transmitter register Underflow Interrupt Enable. Channel 0 0x0: Interrupt disabled 0x1: Interrupt enabled	RW	0
0	TX0_EMPTY_ENABLE	Transmitter register Empty Interrupt Enable. Channel 0 0x0: Interrupt disabled 0x1: Interrupt enabled	RW	0

Table 18-251. Register Call Summary for Register MCSPI_IRQENABLE

Multichannel Serial Peripheral Interface

- Interrupts: [0]
- Interrupt-Driven Operation: [1]
- Polling: [2]
- Idle Mode: [3][4]
- McSPI Operational Modes: [5][6][7][8]
- Receive-Only Procedure – Interrupt Method: [9][10]
- McSPI Register Summary: [11]

Table 18-252. MCSPI_WAKEUPENABLE

Address Offset	0x120		
Physical Address	0x4809 8120 0x4809 A120 0x480B 8120 0x480B A120	**Instance**	McSPI1 McSPI2 McSPI3 McSPI4
Description	The wake-up enable register allows enabling and disabling of the module internal sources of wakeup on event-by-event basis.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															WKEN

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reads returns 0.	RW	0x0000 0000
0	WKEN	Wake-up functionality in slave mode when an active control signal is detected on the SPIEN line programmed in the MCSPI_CHxCONF[22:21] SPIENSLV bits 0x0: The event is not allowed to wake-up the system, even if the global control bit MCSPI_SYSCONFIG[2] ENAWAKEUP is set. 0x1: The event is allowed to wake-up the system if the global control bit MCSPI_SYSCONFIG[2] ENAWAKEUP is set.	RW	0

Table 18-253. Register Call Summary for Register MCSPI_WAKEUPENABLE

Multichannel Serial Peripheral Interface

- Idle Mode: [0][1][2]
- McSPI Register Summary: [3]

Table 18-254. MCSPI_SYST

Address Offset	0x124		
Physical Address	0x4809 8124 0x4809 A124 0x480B 8124 0x480B A124	**Instance**	McSPI1 McSPI2 McSPI3 McSPI4
Description	This register is used to check the correctness of the system interconnect either internally to peripheral bus, or externally to device I/O pads, when the module is configured in system test (SYSTEST) mode.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																				SSB	SPIENDIR	SPIDATDIR1	SPIDATDIR0	WAKD	SPICLK	SPIDAT_1	SPIDAT_0	SPIEN_3	SPIEN_2	SPIEN_1	SPIEN_0

Bits	Field Name	Description	Type	Reset
31:12	RESERVED	Reads returns 0.	RW	0x00000
11	SSB	Set status bit 0x0: No action. Writing 0 does not clear already set status bits. This bit must be cleared before trying to clear a status bit of the MCSPI_IRQSTATUS register. 0x1: Force to 1 all status bits of MCSPI_IRQSTATUS register. Writing 1 into this bit sets to 1 all status bits in the MCSPI_IRQSTATUS register.	RW	0
10	SPIENDIR	Set the direction of the SPIEN[3:0] lines and SPICLK line. 0x0: Output (as in master mode) 0x1: Input (as in slave mode)	RW	0
9	SPIDATDIR1	Set the direction of the SPIDAT[1]. 0x0: Output 0x1: Input	RW	0
8	SPIDATDIR0	Set the direction of the SPIDAT[0]. 0x0: Output 0x1: Input	RW	0
7	WAKD	SWAKEUP output (signal data value of internal signal to system). The signal is driven high or low according to the value written into this bit. 0x0: The pin is driven low. 0x1: The pin is driven high.	RW	0
6	SPICLK	SPICLK line (signal data value) If [10] SPIENDIR = 1 (input mode direction), this bit returns the value on the CLKSPI line (high or low), and a write into this bit has no effect. If [10] SPIENDIR = 0 (output mode direction), the CLKSPI line is driven high or low according to the value written into this bit.	RW	0
5	SPIDAT_1	SPIDAT[1] line (signal data value) If [9] SPIDATDIR1 = 0 (output mode direction), the SPIDAT[1] line is driven high or low according to the value written into this bit. If [9] SPIDATDIR1 = 1 (input mode direction), this bit returns the value on the SPIDAT[1] line (high or low), and a write into this bit has no effect.	RW	0

Bits	Field Name	Description	Type	Reset
4	SPIDAT_0	SPIDAT[0] line (signal data value) If [8] SPIDATDIR0 = 0 (output mode direction), the SPIDAT[0] line is driven high or low according to the value written into this bit. If [8] SPIDATDIR0 = 1 (input mode direction), this bit returns the value on the SPIDAT[0] line (high or low), and a write into this bit has no effect.	RW	0
3	SPIEN_3	SPIEN[3] line (signal data value) If [10] SPIENDIR = 0 (output mode direction), the SPIENT[3] line is driven high or low according to the value written into this bit. If [10] SPIENDIR = 1 (input mode direction), this bit returns the value on the SPIEN[3] line (high or low), and a write into this bit has no effect.	RW	0
2	SPIEN_2	SPIEN[2] line (signal data value) If [10] SPIENDIR = 0 (output mode direction), the SPIENT[2] line is driven high or low according to the value written into this bit. If [10] SPIENDIR = 1 (input mode direction), this bit returns the value on the SPIEN[2] line (high or low), and a write into this bit has no effect.	RW	0
1	SPIEN_1	SPIEN[1] line (signal data value) If [10] SPIENDIR = 0 (output mode direction), the SPIENT[1] line is driven high or low according to the value written into this bit. If [10] SPIENDIR = 1 (input mode direction), this bit returns the value on the SPIEN[1] line (high or low), and a write into this bit has no effect.	RW	0
0	SPIEN_0	SPIEN[0] line (signal data value) If [10] SPIENDIR = 0 (output mode direction), the SPIENT[0] line is driven high or low according to the value written into this bit. If [10] SPIENDIR = 1 (input mode direction), this bit returns the value on the SPIEN[0] line (high or low), and a write into this bit has no effect.	RW	0

Table 18-255. Register Call Summary for Register MCSPI_SYST

Multichannel Serial Peripheral Interface

- McSPI Register Summary: [0]

Table 18-256. MCSPI_MODULCTRL

Address Offset	0x128		
Physical Address	0x4809 8128 0x4809 A128 0x480B 8128 0x480B A128	**Instance**	McSPI1 McSPI2 McSPI3 McSPI4
Description	This register is dedicated to the configuration of the serial peripheral interface.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							FDAA	MOA	INITDLY			SYSTEM_TEST	MS	PIN34	SINGLE

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reads returns 0.	RW	0x000000
8	FDAA	FIFO DMA address 256-bit aligned This bit is used when a FIFO is managed by the module and DMA connected to the controller provides only 256-bit aligned address. If this bit is set the enabled channel which uses the FIFO has its data managed through MCSPI_DAFTX and MCSPI_DAFRX registers instead of MCSPI_TXx and MCSPI_RXx registers. 0x0: FIFO data managed by MCSPI_TXx and MCSPI_RXx registers. 0x1: FIFO data managed by MCSPI_DAFTX and MCSPI_DAFRX registers.	RW	0
7	MOA	Multiple word interface access: this bit can only be used when a channel is enabled using a FIFO. It allows the system to perform multiple SPI word access for a single 32-bit interface word access. This is possible for WL < 16. 0x0: Multiple word access disabled 0x1: Multiple word access enabled with FIFO	RW	0
6:4	INITDLY	Initial SPI delay for first transfer: this field is an option only available in SINGLE master mode. The controller waits for a delay to transmit the first SPI word after channel enabled and corresponding TX register filled. This delay is based on SPI output frequency clock. No clock output provided to the boundary and chip select is not active in 4-pin mode within this period. 0x0: No delay for first spi transfer. 0x1: The controller wait 4 SPI bus clock 0x2: The controller wait 8 SPI bus clock 0x3: The controller wait 16 SPI bus clock 0x4: The controller wait 32 SPI bus clock	RW	0x0
3	SYSTEM_TEST	Enables the system test mode 0x0: Functional mode 0x1: System test mode (SYSTEST)	RW	0
2	MS	Master/slave 0x0: Master - The module generates the SPICLK and SPIEN[3:0]. 0x1: Slave - The module receives the SPICLK and SPIEN[3:0].	RW	1
1	PIN34	Pin mode selection: This bit is used in master or slave mode to configure the SPI pin mode (3-pin or 4-pin). If asserted the controller only uses SIMO, SOMI, and SPICLK clock pin for SPI transfers. 0x0: SPIEN is used as a chip-select. 0x1: SPIEN is not used. In this mode all related options to chip-select have no meaning.	RW	0
0	SINGLE	Single channel/Multi Channel (master mode only) 0x0: More than one channel will be used in master mode. 0x1: Only one channel will be used in master mode. This bit must be set in Force SPIEN[x] mode.	RW	0

Table 18-257. Register Call Summary for Register MCSPI_MODULCTRL

Multichannel Serial Peripheral Interface

- Single-Channel Master Mode: [0][1][2][3][4][5]
- Chip-Select Timing Control: [6][7]
- Slave Mode: [8][9][10][11]
- 3-Pin or 4-Pin Mode: [12]
- McSPI Global Initialization: [13]
- McSPI Register Summary: [14]
- McSPI Register Description: [15][16]

Table 18-258. MCSPI_CHxCONF

Address Offset	0x12C + (0x14 * x)	**Index**	x = 0 to 3 for McSPI1 x = 0 to 1 for McSPI2 and McSPI3 x = 0 for McSPI4.
Physical Address	0x4809 812C + (0x14 * x) 0x4809 A12C + (0x14 * x) 0x480B 812C + (0x14 * x) 0x480B A12C + (0x14 * x)	**Instance**	McSPI1 McSPI2 McSPI3 McSPI4
Description	This register is dedicated to the configuration of the channel x		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED		CLKG	FFER	FFEW	TCS0		SBPOL	SBE	SPIENSLV		FORCE	TURBO	IS	DPE1	DPE0	DMAR	DMAW	TRM		WL				EPOL	CLKD					POL	PHA

Bits	Field Name	Description	Type	Reset
31:30	RESERVED	Read returns 0.	R	0x0
29	CLKG	Clock divider granularity this bit defines the granularity of channel clock divider: power of 2 or one clock cycle granularity. When this bit is set the register MCSPI_CHxCTRL[15:8] EXTCLK must be configured to reach a maximum of 4096 clock divider ratio. Then the clock divider ratio is a concatenation of [5:2] CLKD and MCSPI_CHxCTRL[15:8] EXTCLK values 0x0: Clock granularity of power of 2 0x1: One clock cycle granularity	RW	0
28	FFER	FIFO enabled for receive: Only one channel can have this bit field set. 0x0: The buffer is not used to receive data. 0x1: The buffer is used to receive data.	RW	0
27	FFEW	FIFO enabled for transmit: Only one channel can have this bit field set. 0x0: The buffer is not used to transmit data. 0x1: The buffer is used to transmit data.	RW	0
26:25	TCS0	Chip-select time control This 2-bit field defines the number of interface clock cycles between CS toggling and first or last edge of SPI clock. 0x0: 0.5 clock cycle 0x1: 1.5 clock cycles 0x2: 2.5 clock cycles 0x3: 3.5 clock cycles	RW	0x0

Bits	Field Name	Description	Type	Reset
24	SBPOL	Start-bit polarity 0x0: Start-bit polarity is held to 0 during SPI transfer. 0x1: Start-bit polarity is held to 1 during SPI transfer.	RW	0
23	SBE	Start-bit enable for SPI transfer 0x0: Default SPI transfer length as specified by WL bit field 0x1: Start bit D/CX added before SPI transfer. Polarity is defined by bit [24] SBPOL	RW	0
22:21	SPIENSLV	Channel 0 only and slave mode only: SPI slave select signal detection. Reserved bits for other cases. 0x0: Detection enabled only on SPIEN[0] 0x1: Detection enabled only on SPIEN[1] 0x2: Detection enabled only on SPIEN[2] 0x3: Detection enabled only on SPIEN[3]	RW	0x0
20	FORCE	Manual SPIEN assertion to keep SPIEN active between SPI words (single channel master mode only). 0x0: Writing 0 into this bit drives low the SPIEN line when [6] EPOL=0, and drives it high when [6] EPOL=1. 0x1: Writing 1 into this bit drives high the SPIEN line when [6] EPOL=0, and drives it low when [6] EPOL=1.	RW	0
19	TURBO	Turbo mode 0x0: Turbo is deactivated (recommended for single SPI word transfer). 0x1: Turbo is activated to maximize the throughput for multiple SPI words transfer.	RW	0
18	IS	Input Select 0x0: Data line 0 (SPIDAT[0]) selected for reception 0x1: Data line 1 (SPIDAT[1]) selected for reception	RW	1
17	DPE1	Transmission enable for data line 1 0x0: Data line 1 (SPIDAT[1]) selected for transmission 0x1: No transmission on Data Line1 (SPIDAT[1])	RW	1
16	DPE0	Transmission Enable for data line 0 0x0: Data Line0 (SPIDAT[0]) selected for transmission 0x1: No transmission on data line 0 (SPIDAT[0])	RW	0
15	DMAR	DMA read request The DMA read request line is asserted when the channel is enabled and a new data is available in the receive register of the channel. The DMA read request line is deasserted on read completion of the receive register of the channel. 0x0: DMA read request disabled 0x1: DMA read request enabled	RW	0
14	DMAW	DMA write request. The DMA write request line is asserted when The channel is enabled and the transmitter register of the channel is empty. The DMA write request line is deasserted on load completion of the transmitter register of the channel. 0x0: DMA write request disabled 0x1: DMA write request enabled	RW	0

Bits	Field Name	Description	Type	Reset
13:12	TRM	Transmit/receive modes	RW	0x0
		0x0: Transmit-and-receive mode		
		0x1: Receive-only mode		
		0x2: Transmit-only mode		
		0x3: Reserved		
11:7	WL	SPI word length	RW	0x00
		0x0: Reserved		
		0x1: Reserved		
		0x2: Reserved		
		0x3: The SPI word is 4 bits long		
		0x4: The SPI word is 5 bits long		
		0x5: The SPI word is 6 bits long		
		0x6: The SPI word is 7 bits long		
		0x7: The SPI word is 8 bits long		
		0x8: The SPI word is 9 bits long		
		0x9: The SPI word is 10 bits long		
		0xA: The SPI word is 11 bits long		
		0xB: The SPI word is 12 bits long		
		0xC: The SPI word is 13 bits long		
		0xD: The SPI word is 14 bits long		
		0xE: The SPI word is 15 bits long		
		0xF: The SPI word is 16 bits long		
		0x10: The SPI word is 17 bits long		
		0x11: The SPI word is 18 bits long		
		0x12: The SPI word is 19 bits long		
		0x13: The SPI word is 20 bits long		
		0x14: The SPI word is 21 bits long		
		0x15: The SPI word is 22 bits long		
		0x16: The SPI word is 23 bits long		
		0x17: The SPI word is 24 bits long		
		0x18: The SPI word is 25 bits long		
		0x19: The SPI word is 26 bits long		
		0x1A: The SPI word is 27 bits long		
		0x1B: The SPI word is 28 bits long		
		0x1C: The SPI word is 29 bits long		
		0x1D: The SPI word is 30 bits long		
		0x1E: The SPI word is 31 bits long		
		0x1F: The SPI word is 32 bits long		
6	EPOL	SPIEN polarity	RW	0
		0x0: SPIEN is held high during the ACTIVE state.		
		0x1: SPIEN is held low during the ACTIVE state.		

Bits	Field Name	Description	Type	Reset
5:2	CLKD	Frequency divider for SPICLK (only when the module is a Master SPI device). A programmable clock divider divides the SPI reference clock (FCLK) with a 4-bit value, and results in a new clock SPICLK available to shift-in and shift-out data. By default, the clock divider ratio has a power of 2 granularity when [29] CLKG is cleared. Otherwise, this field is the 4-LSB bit of a 12-bit register concatenated with clock divider extension MCSPI_CHxCTRL[15:8] EXTCLK register. The value description below defines the clock ratio when [29] CLKG is set to 0. 0x0: 1 0x1: 2 0x2: 4 0x3: 8 0x4: 16 0x5: 32 0x6: 64 0x7: 128 0x8: 256 0x9: 512 0xA: 1024 0xB: 2048 0xC: 4096 0xD: 8192 0xE: 16384 0xF: 32768	RW	0x0
1	POL	SPICLK polarity (see Section 18.3.2.3.1, *Transfer Format*) 0x0: SPICLK is held low during the INACTIVE state 0x1: SPICLK is held high during the INACTIVE state	RW	0
0	PHA	SPICLK phase (see Section 18.3.2.3.1, *Transfer Format*) 0x0: Data are latched on odd-numbered edges of SPICLK. 0x1: Data are latched on even-numbered edges of SPICLK.	RW	0

Table 18-259. Register Call Summary for Register MCSPI_CHxCONF

Multichannel Serial Peripheral Interface

- Basic McSPI Pins for Master Mode: [0][1][2][3]
- Basic McSPI Pins for Slave Mode: [4][5][6][7]
- Multichannel SPI Protocol and Data Format: [8][9][10][11][12][13][14][15]
- Transfer Format: [16]
- SPI in Slave Mode: [17]
- Master Transmit-and-Receive Mode (Full Duplex): [18]
- Master Transmit-Only Mode (Half Duplex): [19]
- Master Receive-Only Mode (Half Duplex): [20]
- Single-Channel Master Mode: [21][22][23][24][25][26][27][28][29][30][31][32]
- Start-Bit Mode: [33][34][35][36]
- Chip-Select Timing Control: [37]
- Programmable SPI Clock: [38][39][40][41][42][43]
- Dedicated Resources: [44][45][46][47][48][49]
- Slave Transmit-and-Receive Mode: [50]
- Slave Transmit-Only Mode: [51]
- Slave Receive-Only Mode: [52]
- 3-Pin or 4-Pin Mode: [53]
- FIFO Buffer Management: [54][55][56][57]
- Buffer Almost Full: [58][59]
- Buffer Almost Empty: [60][61]
- Interrupt Events in Master Mode: [62][63]
- Interrupt Events in Slave Mode: [64][65]
- DMA Requests: [66][67]
- Idle Mode: [68]
- McSPI Operational Modes: [69][70][71][72][73][74][75][76][77][78][79][80][81][82][83][84][85]
- McSPI Register Summary: [86]
- McSPI Register Description: [87][88][89][90][91][92][93]

Table 18-260. MCSPI_CHxSTAT

Address Offset	0x130 + (0x14 * x)	**Index**	x = 0 to 3 for McSPI1 x = 0 to 1 for McSPI2 and McSPI3 x = 0 for McSPI4.
Physical Address	0x4809 8130 + (0x14 * x) 0x4809 A130 + (0x14 * x) 0x480B 8130 + (0x14 * x) 0x480B A130 + (0x14 * x)	**Instance**	McSPI1 McSPI2 McSPI3 McSPI4
Description	This register provides status information about transmitter and receiver registers of channel x.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								RXFFF	RXFFE	TXFFF	TXFFE	EOT	TXS	RXS	

Bits	Field Name	Description	Type	Reset
31:7	RESERVED	Read returns 0.	R	0x0000000
6	RXFFF	Channel x FIFO receive buffer full status Read 0x0: FIFO receive buffer is not full Read 0x1: FIFO receive buffer is full	R	0
5	RXFFE	Channel x FIFO receive buffer empty status Read 0x0: FIFO receive buffer is not empty Read 0x1: FIFO receive buffer is empty	R	0

Bits	Field Name	Description	Type	Reset
4	TXFFF	Channel x FIFO transmit buffer full status Read 0x0: FIFO transmit buffer is not full Read 0x1: FIFO transmit buffer is full	R	0
3	TXFFE	Channel x FIFO transmit buffer empty status Read 0x0: FIFO transmit buffer is not empty Read 0x1: FIFO transmit buffer is empty	R	0
2	EOT	Channel x end of transfer status. The definitions of beginning and end of transfer vary with master versus slave and the transfer format (transmit/receive modes, turbo mode). See dedicated chapters for details. Read 0x0: This flag is automatically cleared when the shift register is loaded with the data from the transmitter register (beginning of transfer). Read 0x1: This flag is automatically set to one at the end of an SPI transfer.	R	0
1	TXS	Channel x transmitter register status Read 0x0: Register is full. Read 0x1: Register is empty.	R	0
0	RXS	Channel x receiver register status Read 0x0: Register is empty. Read 0x1: Register is full.	R	0

Table 18-261. Register Call Summary for Register MCSPI_CHxSTAT

Multichannel Serial Peripheral Interface

- Master Transmit-and-Receive Mode (Full Duplex): [0][1][2]
- Master Transmit-Only Mode (Half Duplex): [3]
- Master Receive-Only Mode (Half Duplex): [4]
- Single-Channel Master Mode: [5][6][7][8][9]
- Dedicated Resources: [10][11]
- Slave Transmit-and-Receive Mode: [12]
- Slave Transmit-Only Mode: [13]
- Slave Receive-Only Mode: [14]
- End of Transfer Management: [15]
- McSPI Operational Modes: [16][17][18][19]
- Receive-Only Procedure – Polling Method: [20]
- Transmit-Only Procedure – Polling Method: [21]
- Transmit-and-Receive Procedure – Polling Method: [22]
- McSPI Register Summary: [23]

Table 18-262. MCSPI_CHxCTRL

Address Offset	0x134 + (0x14 * x)	**Index**	x = 0 to 3 for McSPI1 x = 0 to 1 for McSPI2 and McSPI3 x = 0 for McSPI4.
Physical Address	0x4809 8134 + (0x14 * x) 0x4809 A134 + (0x14 * x) 0x480B 8134 + (0x14 * x) 0x480B A134 + (0x14 * x)	**Instance**	McSPI1 McSPI2 McSPI3 McSPI4
Description	This register is dedicated to enable channel x.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																EXTCLK								RESERVED							EN

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Read returns 0.	RW	0x0000
15:8	EXTCLK	Clock ratio extension: this field is used to concatenate with MCSPI_CHxCONF[5:2] CLKD register for clock ratio only when granularity is one clock cycle (MCSPI_CHxCONF[29] CLKG set to 1). Then the maximum value reached is 4096 clock divider ratio. 0x0: Clock ratio is CLKD + 1. 0x1: Clock ratio is CLKD + 1 + 16. ... 0xFF: Clock ratio is CLKD + 1 + 4080.	RW	0x00
7:1	RESERVED	Read returns 0.	RW	0x00
0	EN	Channel enable 0x0: Channel x is not active. 0x1: Channel x is active.	RW	0

Table 18-263. Register Call Summary for Register MCSPI_CHxCTRL

Multichannel Serial Peripheral Interface

- Master Transmit-and-Receive Mode (Full Duplex): [0]
- Single-Channel Master Mode: [1][2]
- Programmable SPI Clock: [3]
- Dedicated Resources: [4][5]
- McSPI Operational Modes: [6][7][8][9][10][11][12][13][14][15][16][17][18][19][20][21][22][23][24][25]
- Receive-Only Procedure – Polling Method: [26][27]
- Receive-Only Procedure – Interrupt Method: [28][29]
- Transmit-Only Procedure – Polling Method: [30][31]
- Transmit-and-Receive Procedure – Polling Method: [32][33]
- McSPI Register Summary: [34]
- McSPI Register Description: [35][36][37]

Table 18-264. MCSPI_TXx

Address Offset	0x138 + (0x14 * x)	**Index**	x = 0 to 3 for McSPI1 x = 0 to 1 for McSPI2 and McSPI3 x = 0 for McSPI4.
Physical Address	0x4809 8138 + (0x14 * x) 0x4809 A138 + (0x14 * x) 0x480B 8138 + (0x14 * x) 0x480B A138 + (0x14 * x)	**Instance**	McSPI1 McSPI2 McSPI3 McSPI4
Description	This register contains a single SPI word for channel x to transmit on the serial link, whatever SPI word length is.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TDATA																															

Bits	Field Name	Description	Type	Reset
31:0	TDATA	Channel x data to transmit	RW	0x0000 0000

Table 18-265. Register Call Summary for Register MCSPI_TXx

Multichannel Serial Peripheral Interface

- Master Transmit-and-Receive Mode (Full Duplex): [0][1][2][3]
- Master Transmit-Only Mode (Half Duplex): [4]
- Master Receive-Only Mode (Half Duplex): [5][6][7][8][9]
- Single-Channel Master Mode: [10][11]
- Dedicated Resources: [12][13][14][15][16]
- Slave Transmit-and-Receive Mode: [17][18]
- Slave Receive-Only Mode: [19][20][21]
- Interrupt Events in Master Mode: [22][23][24][25][26][27][28]
- Interrupt Events in Slave Mode: [29][30][31][32][33][34]
- Interrupt-Driven Operation: [35]
- Polling: [36]
- DMA Requests: [37][38]
- McSPI Operational Modes: [39][40][41][42][43]
- Transmit-Only Procedure – Polling Method: [44]
- Transmit-and-Receive Procedure – Polling Method: [45][46]
- McSPI Register Summary: [47]
- McSPI Register Description: [48][49][50]

Table 18-266. MCSPI_RXx

Address Offset	0x13C + (0x14 * x)	**Index**	x = 0 to 3 for McSPI1 x = 0 to 1 for McSPI2 and McSPI3 x = 0 for McSPI4.
Physical Address	0x4809 813C + (0x14 * x) 0x4809 A13C + (0x14 * x) 0x480B 813C + (0x14 * x) 0x480B A13C + (0x14 * x)	**Instance**	McSPI1 McSPI2 McSPI3 McSPI4
Description	This register contains a single SPI word for channel x received through the serial link, whatever SPI word length is.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RDATA																															

Bits	Field Name	Description	Type	Reset
31:0	RDATA	Channel x received data	R	0x0000 0000

Table 18-267. Register Call Summary for Register MCSPI_RXx

Multichannel Serial Peripheral Interface

- Master Transmit-and-Receive Mode (Full Duplex): [0][1]
- Master Transmit-Only Mode (Half Duplex): [2][3][4]
- Master Receive-Only Mode (Half Duplex): [5][6]
- Single-Channel Master Mode: [7][8][9][10]
- Dedicated Resources: [11][12][13]
- Slave Transmit-and-Receive Mode: [14]
- Slave Transmit-Only Mode: [15][16]
- End of Transfer Management: [17]
- Interrupt Events in Master Mode: [18][19][20][21]
- Interrupt Events in Slave Mode: [22][23][24][25][26][27]
- Interrupt-Driven Operation: [28]
- Polling: [29]
- DMA Requests: [30]
- Idle Mode: [31]
- McSPI Operational Modes: [32][33][34][35][36][37][38]
- Receive-Only Procedure – Polling Method: [39]
- Receive-Only Procedure – Interrupt Method: [40]
- McSPI Register Summary: [41]
- McSPI Register Description: [42][43][44]

Table 18-268. MCSPI_XFERLEVEL

Address Offset	0x17C		
Physical Address	0x4809 817C 0x4809 A17C 0x480B 817C 0x480B A17C	**Instance**	McSPI1 McSPI2 McSPI3 McSPI4
Description	This register provides transfer levels needed while using FIFO buffer during transfer.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WCNT																AFL								AEL							

Bits	Field Name	Description	Type	Reset
31:16	WCNT	SPI word counter. This field holds the programmable value of number of SPI word to be transferred on channel which is using the FIFO buffer. When transfer had started, a read back in this field returns the current SPI word transfer index. 0x0: Counter not used 0x1: One word ... 0xFFFE: 65534 SPI words 0xFFFF: 65535 SPI words	RW	0x0000
15:8	AFL	Buffer almost full This field holds the programmable almost-full level value used to determine almost full buffer condition. If the user wants an interrupt or a DMA read request to be issued during a receive operation when the data buffer holds at least n bytes, then the buffer AFL must be set with n-1. 0x0: 1 byte 0x1: 2 bytes ... 0xFE: 255 bytes 0xFF: 256 bytes	RW	0x00
7:0	AEL	Buffer almost empty. this field holds the programmable almost-empty level value used to determine almost empty buffer condition. If the user wants an interrupt or a DMA write request to be issued during a transmit operation when the data buffer is able to receive n bytes, then the buffer AEL must be set with n-1. 0x0: 1 byte 0x1: 2 bytes ... 0xFE: 255 bytes 0xFF: 256 bytes	RW	0x00

Table 18-269. Register Call Summary for Register MCSPI_XFERLEVEL

Multichannel Serial Peripheral Interface

- FIFO Buffer Management: [0][1]
- Buffer Almost Full: [2]
- Buffer Almost Empty: [3]
- End of Transfer Management: [4][5]
- Interrupt Events in Master Mode: [6][7][8][9][10]
- Interrupt Events in Slave Mode: [11][12][13][14]
- McSPI Operational Modes: [15][16]
- McSPI Register Summary: [17]
- McSPI Register Description: [18]

Table 18-270. MCSPI_DAFTX

Address Offset	0x0000 0180		
Physical Address	0x4809 8180 0x4809 A180 0x480B 8180 0x480B A180	**Instance**	McSPI1 McSPI2 McSPI3 McSPI4
Description	This register contains the SPI words to be transmitted on the SPI bus when FIFO is used and DMA address is aligned on 256 bit. This register is an image of one of the MCSPI_TXx registers corresponding to the channel which has its FIFO enabled.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DAFTDATA																															

Bits	Field Name	Description	Type	Reset
31:0	DAFTDATA	FIFO data to transmit with DMA 256 bit aligned address. This field is only used when MCSPI_MODULCTRL[8] FDAA is set to 0x1 and only one of the enabled channels has the MCSPI_CHxCONF[27] FFEW bit set to 0x1. If these conditions are not met any access to this field returns a null value.	RW	0x00000000

Table 18-271. Register Call Summary for Register MCSPI_DAFTX

Multichannel Serial Peripheral Interface

- McSPI Register Summary: [0]
- McSPI Register Description: [1][2]

Table 18-272. MCSPI_DAFRX

Address Offset	0x0000 01A0		
Physical Address	0x4809 81A0 0x4809 A1A0 0x480B 81A0 0x480B A1A0	**Instance**	McSPI1 McSPI2 McSPI3 McSPI4
Description	This register contains the SPI words received from the SPI bus when FIFO is used and DMA address is aligned on 256 bit. This register is an image of one of the MCSPI_RXx registers corresponding to the channel which has its FIFO enabled.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DAFRDATA																															

Bits	Field Name	Description	Type	Reset
31:0	DAFRDATA	FIFO data received with DMA 256 bit aligned address. This field is only used when MCSPI_MODULCTRL[8] FDAA is set to 0x1 and only one of the enabled channels has the MCSPI_CHxCONF[28] FFER bit set to 0x1. If these conditions are not met any access to this field returns a null value.	R	0x00000000

Table 18-273. Register Call Summary for Register MCSPI_DAFRX

Multichannel Serial Peripheral Interface

- McSPI Register Summary: [0]
- McSPI Register Description: [1][2]

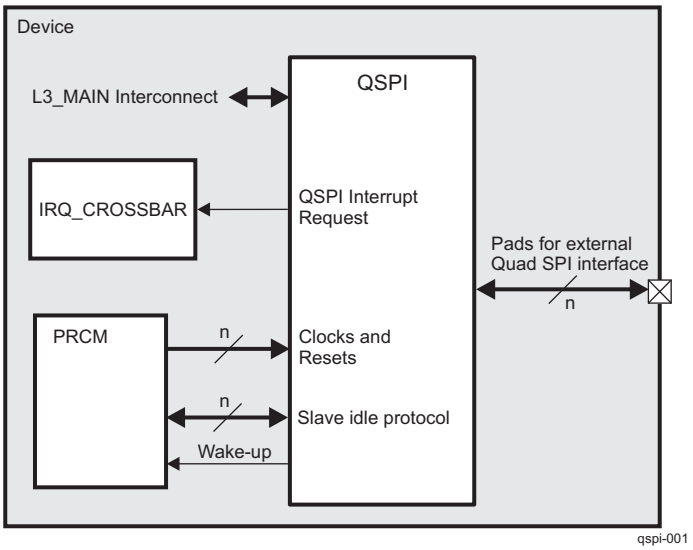
18.4 Quad Serial Peripheral Interface

18.4.1 Quad Serial Peripheral Interface Overview

The quad serial peripheral interface (QSPI) module is a kind of SPI module that allows single, dual, or quad read access to external SPI devices. This module has a memory mapped register interface, which provides a direct interface for accessing data from external SPI devices and thus simplifying software requirements. The QSPI works as a master only.

The one QSPI in the device is primarily intended for fast booting from quad-SPI flash memories. Figure 18-71 shows the QSPI module overview.

Figure 18-71. QSPI Overview



The QSPI supports the following features:

- General SPI features:
  - Programmable clock divider
  - Six pin interface
  - Programmable length (from 1 to 128 bits) of the words transferred
  - Programmable number (from 1 to 4096) of the words transferred
  - 4 external chip-select signals
  - Support for 3-, 4-, or 6-pin SPI interface
  - Optional interrupt generation on word or frame (number of words) completion
  - Programmable delay between chip select activation and output data from 0 to 3 QSPI clock cycles
  - Programmable signal polarities
  - Programmable active clock edge
  - Software-controllable interface allowing for any type of SPI transfer
  - Control through L3_MAIN configuration port
- Serial flash interface (SFI) features:
  - Serial flash read/write interface
  - Additional registers for defining read and write commands to the external serial flash device
  - 1 to 4 address bytes
  - Fast read support, where fast read requires dummy bytes after address bytes; 0 to 3 dummy bytes can be configured.
  - Dual read support

- Quad read support
- Little-endian support (only for memory mapped registers used to configure QSPI controller and not SPI content accesses)
- Linear increment addressing mode only

The QSPI supports only dual and quad reads. Dual or quad writes are not supported. In addition, there is no "pass through" mode supported where the data present on the QSPI input is sent to its output.

NOTE: The QSPI module does not support cache line wrap mode.

18.4.2 QSPI Environment

Figure 18-72 shows a typical connection of the QSPI module to the external quad-SPI flash memory.

Figure 18-72. QSPI Connected to an External Quad-SPI Flash Memory

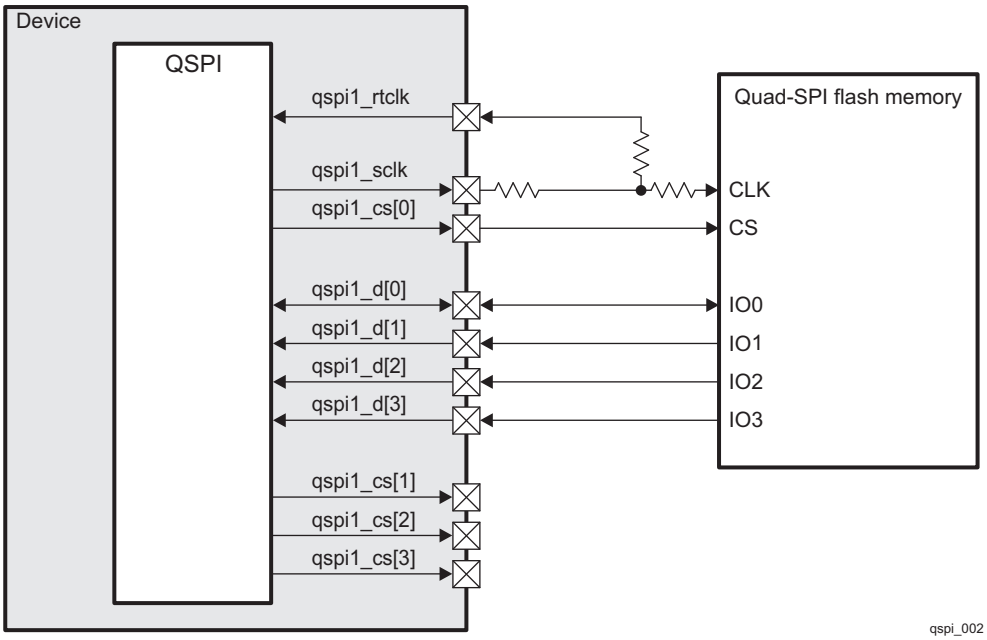


Table 18-274 lists and describes the QSPI I/O signals.

Table 18-274. QSPI I/O Signals

QSPI Signal/Pad name	I/O[1]	Description					
		3-pin[2] SPI Read (Single Read)	3-pin[2] SPI Write (Single Write)	4-pin[2] SPI Read (Single Read)	4-pin[2] SPI Write (Single Write)	4-pin[2] SPI Read (Dual Read)	6-pin[2] SPI Read (Quad Read)
qspi1_d[0]	IO	Used as SPI data input	Used as SPI data output	Not used	Used as SPI data output	Used as SPI data input 0	Used as SPI data input 0
qspi1_d[1]	I	Not used	Not used	Used as SPI data input	Not used	Used as SPI data input 1	Used as SPI data input 1
qspi1_d[2]	I	Not used	Not used	Not used	Not used	Not used	Used as SPI data input 2
qspi1_d[3]	I	Not used	Not used	Not used	Not used	Not used	Used as SPI data input 3
qspi1_sclk	O	Clock for the external SPI device					
qspi1_cs[0]	O	External SPI device chip-select 0					
qspi1_cs[1]	O	External SPI device chip-select 1					
qspi1_cs[2]	O	External SPI device chip-select 2					
qspi1_cs[3]	O	External SPI device chip-select 3					
qspi1_rtclk	I	The qspi1_sclk output must be connected to the qspi1_rtclk input, and is used for controlling the timing of the read return data when the QSPI module operates in Mode 0. In case Mode 3 is used, there is no need to connect the qspi1_sclk to the qspi1_rtclk.					

(1) I = Input; O = Output

(2) This is the pin count at the SPI flash memory side. The pin count at the device side is increased by one because of the qspi1_rtclk signal. References to the pin count throughout this chapter consider the pin count at the SPI flash memory side.

NOTE: In order to ensure proper timing, precise layout and routing requirements must be followed. For layout and routing requirements for all QSPI signals, see section “PCB Guidelines” of the device Data Manual.

18.4.3 QSPI Integration

Figure 18-73 shows the integration of the QSPI module in the device.

Figure 18-73. QSPI Integration

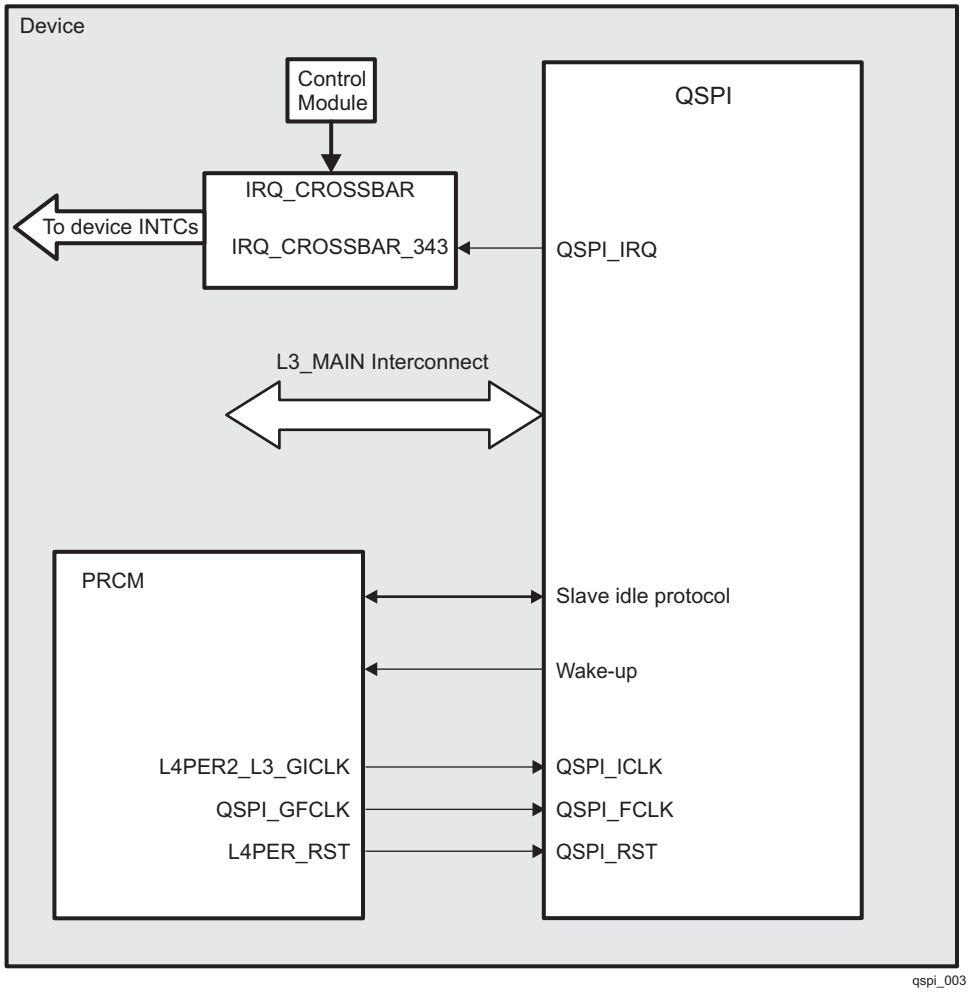


Table 18-275 through Table 18-277 summarize the integration of the QSPI in the device.

Table 18-275. QSPI Integration Attributes

Module Instance	Attributes		
	Power Domain	Wake-Up Capability	Interconnect
QSPI	PD_COREAON	Yes	L3_MAIN

Table 18-276. QSPI Clocks and Resets

Clocks				
Module Instance	**Destination Signal Name**	**Source Signal Name**	**Source**	**Description**
QSPI	QSPI_ICLK	L4PER2_L3_GICLK	PRCM	Interface clock for the QSPI
	QSPI_FCLK	QSPI_GFCLK	PRCM	Functional clock for the QSPI
Resets				
Module Instance	**Destination Signal Name**	**Source Signal Name**	**Source**	**Description**
QSPI	QSPI_RST	L4PER_RST	PRCM	Asynchronous reset signal for the QSPI

Table 18-277. QSPI Hardware Requests

Interrupt Requests				
Module Instance	**Source Signal Name**	**Destination IRQ_CROSSBAR Input**	**Default Mapping**	**Description**
QSPI	QSPI_IRQ	IRQ_CROSSBAR_343	–	QSPI interrupt request

NOTE: The Default Mapping column in Table 18-277 shows the default mapping of module IRQ source signals. These IRQ source signals can also be mapped to other lines of each device interrupt controller (INTC) through the IRQ_CROSSBAR module.
For more information about the IRQ_CROSSBAR module, see Section 13.4.6.4, *IRQ_CROSSBAR Module Functional Description*, in Chapter 13, *Control Module*.
For more information about the device INTCs, see Chapter 12, *Interrupt Controllers*.

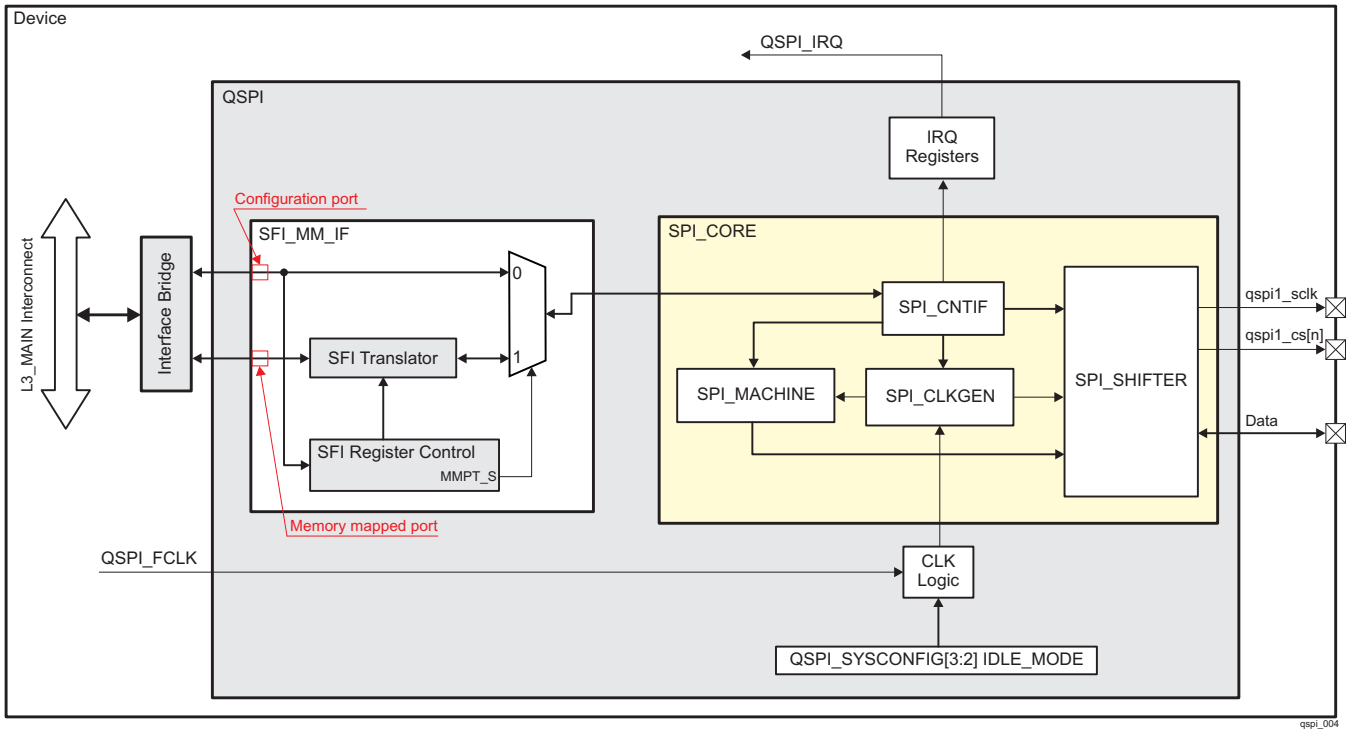
NOTE: For the description of the interrupt source, see Section 18.4.4.3, *QSPI Interrupt Requests*.

18.4.4 QSPI Functional Description

18.4.4.1 QSPI Block Diagram

Iinitial device boot from external SPI flash memory can be accomplished through the QSPI module. The interface is a simple 4-wire SPI used for control or data transfers. The QSPI also supports a 3-wire SPI protocol where the qspi1_d[0] signal is used as a bidirectional for reads and writes. In addition, a 6-wire mode can be used to support quad read devices. Figure 18-74 shows the QSPI block diagram.

Figure 18-74. QSPI Block Diagram



The QSPI is composed of two blocks. The first one is the SFI memory-mapped interface (SFI_MM_IF) and the second one is the SPI core (SPI_CORE). The SFI_MM_IF block is associated only with SPI flash memories and is used for specifying typical for the SPI flash memories settings (read or write command, number of address and dummy bytes, and so on) unlike the SPI_CORE block, which is associated with the SPI interface itself and is used to configure typical SPI settings (chip-select polarity, serial clock inactive state, SPI clock mode, length of the words transferred, and so on).

The SFI_MM_IF comprises the following two subblocks:

- SFI register control
- SFI translator

The SPI_CORE comprises the following four subblocks:

- SPI control interface (SPI_CNTIF)
- SPI clock generator (SPI_CLKGEN)
- SPI control state machine (SPI_MACHINE)
- SPI data shifter (SPI_SHIFTER)

In addition, an interface bridge connects the two ports (configuration port and memory-mapped port) of the SFI_MM_IF block to the L3_MAIN interconnect. There are no software controls associated with this interface bridge.

The QSPI supports long transfers through a frame-style sequence. In its generic SPI use mode, a word can be defined up to 128 bits and multiple words can be transferred during a single access. For each word, a device initiator must read or write the new data and then tell the QSPI to continue the current operation. Using this sequence, a maximum of 4096 128-bit words can be transferred in a single SPI read or write operation. This allows great flexibility when connecting the QSPI to various types of devices.

As opposed to the generic SPI use mode, the communication with serial flash-type devices requires sending a byte command, followed by sending bytes of data. Commands can be sent through the SPI_CORE block to communicate with a serial flash device; however, it is easier to do this using the SFI_MM_IF block because it is intended to ease the communication with serial flash devices. If the SPI_CORE is used to communicate with a serial flash device, software must load the command into the SPI data transfer register with additional configuration fields, perform the byte transfer, then place the data to be sent (or configure for receive) along with additional configuration fields, and perform that transfer. Reads and writes to serial flash devices are more specific. First, the read or write command byte is sent, followed by 1 to 4 bytes of address (corresponding to the address to read/write), then followed by the data write/receive phase. Data is always sent byte oriented. When the address is loaded, data can be continuously read or written, and the address will automatically increment to each byte address internally to the serial flash device.

NOTE: The SFI_MM_IF block only allows reading and writing to an externaly connected SPI flash device. The SFI_MM_IF block does not allow reads or writes to internal configuration and status registers of the SPI flash device. These registers must be accessed through the features of the SPI_CORE block.

18.4.4.1.1 SFI Register Control

The SFI register control block consists of the following five configuration registers:

- QSPI_SPI_SETUP0_REG
- QSPI_SPI_SETUP1_REG
- QSPI_SPI_SETUP2_REG
- QSPI_SPI_SETUP3_REG
- QSPI_SPI_SWITCH_REG

The first four registers let the user define the following:

- Byte command for a serial flash read specified by the QSPI_SPI_SETUPi_REG[7:0] RCMD bit field
- Byte command for a serial flash write specified by the QSPI_SPI_SETUPi_REG[23:16] WCMD bit field
- Number of address bytes required for the particular type of serial flash specified by the

QSPI_SPI_SETUPi_REG[9:8] NUM_A_BYTES bit field

- Number of “dummy bytes” that may be needed to support the fast read mode function of some serial flash devices. The QSPI_SPI_SETUPi_REG[11:10] NUM_D_BYTES bit field specifies the number of "dummy bits." In addition, the QSPI_SPI_SETUPi_REG[28:24] NUM_D_BITS bit field can also specify the number of "dummy bits."
- Whether the read command is single (normal), dual, or quad read mode command. This is specified by the QSPI_SPI_SETUPi_REG[13:12] READ_TYPE bit field. (i is equal to 0, 1, 2 and 3 and means that the QSPI_SPI_SETUPi_REG registers are associated with each of the four supported chip-selects [that is, four supported output SPI flash devices])

The QSPI_SPI_SWITCH_REG register acts as a static switch which allows the configuration port (shown in Figure 18-74) to connect directly to the SPI_CORE block, or allows the memory-mapped port (also shown in Figure 18-74) to connect to the SPI_CORE block. This is done using the QSPI_SPI_SWITCH_REG[0] MMPT_S bit.

In addition, the QSPI_SPI_SWITCH_REG[1] MM_INT_EN bit is used to enable or disable the word complete interrupt during operations using the memory-mapped port.

18.4.4.1.2 SFI Translator

The SFI translator block represents an FSM which, based on the configuration information loaded into the SFI register control block, converts each input read/write sequence into an SPI_CORE configuration sequence for access to the external serial flash memory.

A read sequence is converted into the following actions:

1. SPI chip-select goes active.
2. Read command byte is issued.
3. 1 to 4 address bytes, which correspond to the first address supplied, are issued.
4. 0 to 3 dummy bytes are issued, if “fast read” is supported.
5. Data bytes are read from the external SPI flash memory.
6. SPI chip-select goes inactive.

For linear addressing mode, action 5 is repeated until the byte count to be transferred reaches zero.

A write sequence is identical to a read sequence, except that a write sequence does not use dummy bytes.

Another important aspect with regard to writes is that a serial flash memory location can only be written to if the bits are erased in advance. Erased means the bits are set to 1. This means that writing only changes 1 contents to 0. It is not possible with this write to change the contents of a bit from 0 to 1. An erase command must be performed to do this operation. Erase commands cannot be executed on single byte locations. Depending on device types, there are page, block, and chip erase commands. To perform an erase command, the particular command must be sent over the SPI bus, and an internal register of the serial flash device must then be polled to determine when the erase completes. The erases must be done through the configuration port by software before performing any writes through the memory-mapped port. This means that writes are passed through to the serial flash device, but if the memory locations being modified are not properly erased before the write, the contents may not result in what was sent.

18.4.4.1.3 SPI Control Interface

The SPI control interface contains configuration registers used to configure the SPI core functionality of the QSPI. This block maintains all configuration settings for the SPI core (that is, settings specific for the SPI interface itself but not for the SPI flash memories).

The registers defined for this block are:

- The QSPI_PID register, which is read only and contains QSPI revision associated information
- The QSPI_SPI_CLOCK_CNTRL_REG register, which is used to control external SPI clock (qspi1_sclk)
- The QSPI_SPI_DC_REG register used to define the SPI clock mode and chip-select polarity for the four external SPI devices
- The QSPI_SPI_CMD_REG register used to control the operation of the SPI command. This register is

also used to configure and transfer data.

- Four data registers used for reading the data received and for writing the data to be transferred. These registers are:
  - QSPI_SPI_DATA_REG
  - QSPI_SPI_DATA_REG_1
  - QSPI_SPI_DATA_REG_2
  - QSPI_SPI_DATA_REG_3

  These four registers compose a 128-bit shift register.
- The QSPI_SPI_STATUS_REG register, which contains status information

All of these registers can only be written if the QSPI is not busy. This means that they can be written if the QSPI_SPI_STATUS_REG[0] BUSY bit is 0x0. The QSPI becomes busy when a write to the QSPI_SPI_CMD_REG[18:16] CMD bit field is performed. Writing to this bit field starts an SPI transaction and sets the QSPI_SPI_STATUS_REG[0] BUSY bit to 0x1. The CMD bit field can be written again when the BUSY bit is 0x0. In addition, the start of the SPI transaction is synchronized to the qspi1_sclk clock and clearing of the BUSY bit is synchronized to the QSPI_FCLK clock.

The register group QSPI_SPI_DATA_REG_3, QSPI_SPI_DATA_REG_2, QSPI_SPI_DATA_REG_1 and QSPI_SPI_DATA_REG is treated as a single 128-bit word for shifting data in and out. The QSPI_SPI_DATA_REG_3 register is used for the most significant bits and the QSPI_SPI_DATA_REG is used for the least significant bits.This applies for both reads and writes. For example, after reading a 128-bit word (WLEN = 0x7F) the most significant bit of the data read, that is bit 127, will be located at QSPI_SPI_DATA_REG_3[31] position and the least significant bit, that is bit 0 of the data read, will be located at the QSPI_SPI_DATA_REG[0] position.

The data written to this register group should be right justified so that a data pre-shifting is not required. The QSPI_SPI_CMD_REG[25:19] WLEN bit field determines the location of the most significant bit and the bit position that will be shifted out first during a write. In order to shift out byte data the WLEN bit field should be set to 0x7 and the data byte should be written to the lower byte of the QSPI_SPI_DATA_REG register. By setting the word length to 0x7 the QSPI_SPI_DATA_REG register will look like a pseudo 8-bit shift register. When the user wants to write 40-bit long word the WLEN bit field should be set to 0x27, the 32 least significant bits of data should be written to the QSPI_SPI_DATA_REG and the rest 8 most significant bits of data should be written to the lower byte of the QSPI_SPI_DATA_REG_1 register. By setting WLEN to 0x27 these two registers will look like a pseudo 40-bit shift register. When the word length is greater than 64 bits the QSPI_SPI_DATA_REG_2 register is also used and the previosly described logic applies. The QSPI_SPI_DATA_REG_3 register is used together with the other three data registers when the word length is greater than 96 bits.

When dual or quad read mode is used the number of the words transferred must be even. This number is configured through the QSPI_SPI_CMD_REG[11:0] FLEN bit field.

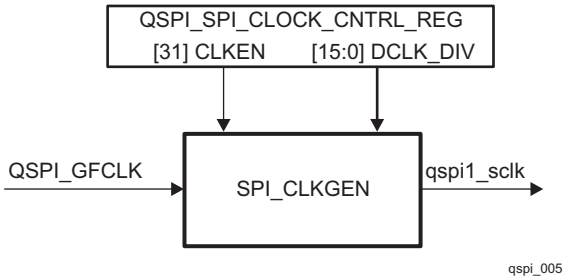
NOTE: The QSPI module does not support a "pass through" mode where the data present on qspi1_d[1] is sent to qspi1_d[0], when 4-pin non-dual read mode is used. This means that setting the QSPI_SPI_CMD_REG[18:16] CMD bit field to 0x1 causes the QSPI only to read from an external device using the qspi1_d[1] pad as an input and if a write to the same external device is desired, the CMD bit field should be set to 0x2, which causes the qspi1_d[0] pad to be used as an output.

18.4.4.1.4 SPI Clock Generator

The SPI clock generator uses the QSPI_FCLK clock as an input, and generates the qspi1_sclk, which is a divided version of the QSPI_FCLK clock. The divide ratio is a 16-bit value configured through the QSPI_SPI_CLOCK_CNTRL_REG[15:0] DCLK_DIV bit field and thus provides a division factor in a range from 1 to 65536. The QSPI_FCLK clock is divided by the DCLK_DIV value + 1 to provide the qspi1_sclk clock. When DCLK_DIV = 0x0 the QSPI_FCLK clock equals the DCLK clock. The value in the DCLK_DIV bit field applies only when the QSPI_SPI_CLOCK_CNTRL_REG[31] CLKEN bit is set to 0x1. Figure 18-75 shows the SPI_CLKGEN block.

If the CLKEN bit is 0x0 the command specified in the QSPI_SPI_CMD_REG[18:16] CMD bit field is not executed and the QSPI_SPI_STATUS_REG[0] BUSY bit is not set. The command is executed only if the CLKEN bit is 0x1 before write to the CMD bit field.

Figure 18-75. SPI_CLKGEN Block



18.4.4.1.5 SPI Control State-Machine

The SPI control state-machine (SPI_MACHINE) manages the operation of the SPI_CORE block. SPI_MACHINE takes control and configuration information from the registers in the SPI_CNTIF block as input and provides control information to the SPI data shifter. This information is used to control the SPI data port. The SPI_MACHINE also generates status information, which is sent back to the SPI_CNTIF block.

Writing a valid value to the QSPI_SPI_CMD_REG[18:16] CMD bit field sets immediately the QSPI_SPI_STATUS_REG[0] BUSY bit to 0x1, activates the corresponding qspi1_cs[n] (n = 0 to 3) and starts the SPI data transaction. The BUSY bit is cleared automatically when QSPI_SPI_CMD_REG[25:19] WLEN number of bits are shifted in or out. If the value of the QSPI_SPI_STATUS_REG[27:16] WDCNT bit field is different than 0x0 and WLEN number of bits are shifted already, the SPI_MACHINE waits until another write to the CMD bit field is performed. If the command written to the CMD bit field is valid, then this increments the value of the WDCNT bit field from 0x0 and starts shifting data in or out again. This is repeated until the WDCNT bit field reaches the frame length (QSPI_SPI_CMD_REG[11:0] FLEN), that is, all words of the frame are shifted or till earlier frame termination occurs. While the SPI_MACHINE is waiting for write to the CMD bit field the corresponding qspi1_cs[n] (n = 0 to 3) remains active and the BUSY flag is set to 0x0. In addition, the bit length for each word can be changed during a frame from 1 to 128 bits using the QSPI_SPI_CMD_REG[25:19] WLEN bit field.

The SPI_MACHINE also provides a mechanism to terminate the frame earlier. This is done by writing an invalid command to the CMD bit field. An invalid command corresponds to the 0x0 and 0x4 (reserved) values of the CMD bit field. Writing one of these values when the the WDCNT bit field is not equal to 0x0 and when the BUSY flag is 0x0 terminates the frame earlier.

The corresponding qspi1_cs[n] (n = 0 to 3) becomes inactive when all words are shifted or when the frame terminates earlier.

18.4.4.1.6 SPI Data Shifter

The SPI data shifter handles the capture and generation of the SPI interface signals. Based on control signals from the SPI_MACHINE and SPI_CNTIF blocks, data is shifted in or out on falling or rising edge of qspi1_sclk clock depending on the SPI clock mode selected. Table 18-278 lists the four defined clock modes of operation for the QSPI.

Table 18-278. SPI Clock Modes Definition

Mode	Settings in the QSPI_SPI_DC_REG Register		Description
	Value of the CKP bits	Value of the CKPH bits	
0	0	0	Data input captured on falling edge of qspi1_sclk clock. Data output generated on falling edge of qspi1_sclk clock
1	0	1	Data input captured on rising edge of qspi1_sclk clock. Data output generated on rising edge of qspi1_sclk clock

Table 18-278. SPI Clock Modes Definition (continued)

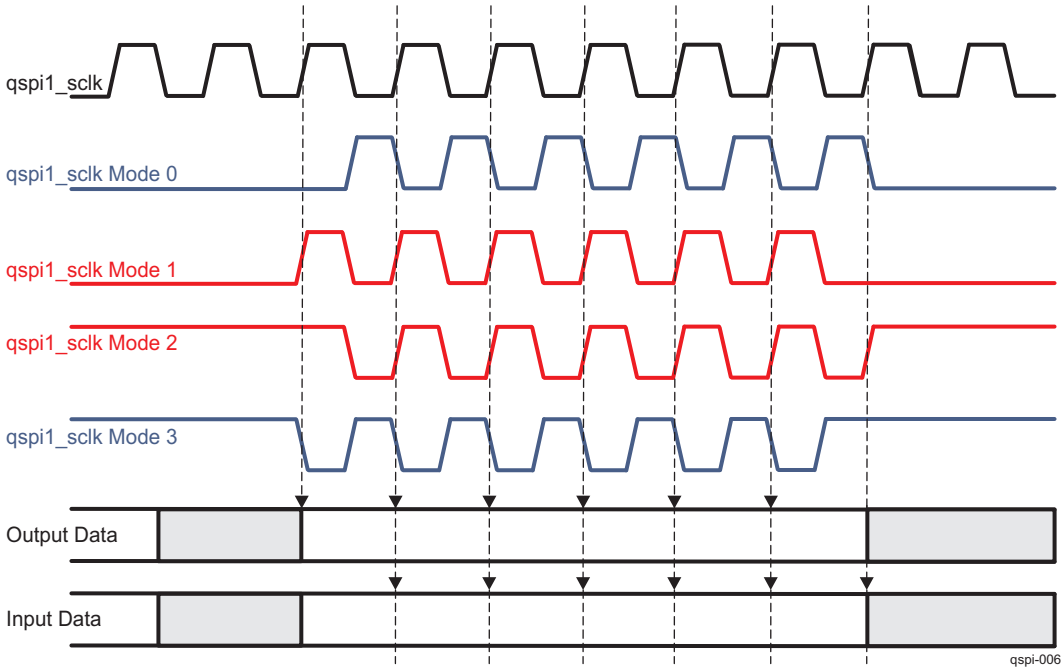
Mode	Settings in the QSPI_SPI_DC_REG Register		Description
	Value of the CKP bits	Value of the CKPH bits	
2	1	0	Data input captured on rising edge of qspi1_sclk clock. Data output generated on rising edge of qspi1_sclk clock
3	1	1	Data input captured on falling edge of qspi1_sclk clock. Data output generated on falling edge of qspi1_sclk clock

NOTE: Mode 1 and Mode 2 are not supported and should not be used.

The CKPi and CKPHi (i = 0 to 3) bits of the QSPI_SPI_DC_REG register control the clock modes. Each of these 4 bits corresponds to an output chip select.

Figure 18-76 shows all four clock modes. In addition, through the DDi (i = 0 to 3) bits of the QSPI_SPI_DC_REG register the data can be delayed from one to three qspi1_sclk clock cycles after the corresponding qspi1_cs[n] (n = 0 to 3) goes active. The active state of each chip-select can also be controlled through the CSPi (i = 0 to 3) bits of the QSPI_SPI_DC_REG register.

Figure 18-76. SPI Clock Modes



18.4.4.2 QSPI Clock Configuration

The QSPI complies with the PRCM slave-idle protocol. The QSPI_FCLK clock is gated based on the values loaded in the QSPI_SYSCONFIG[3:2] IDLE_MODE bit field. Three modes are supported:

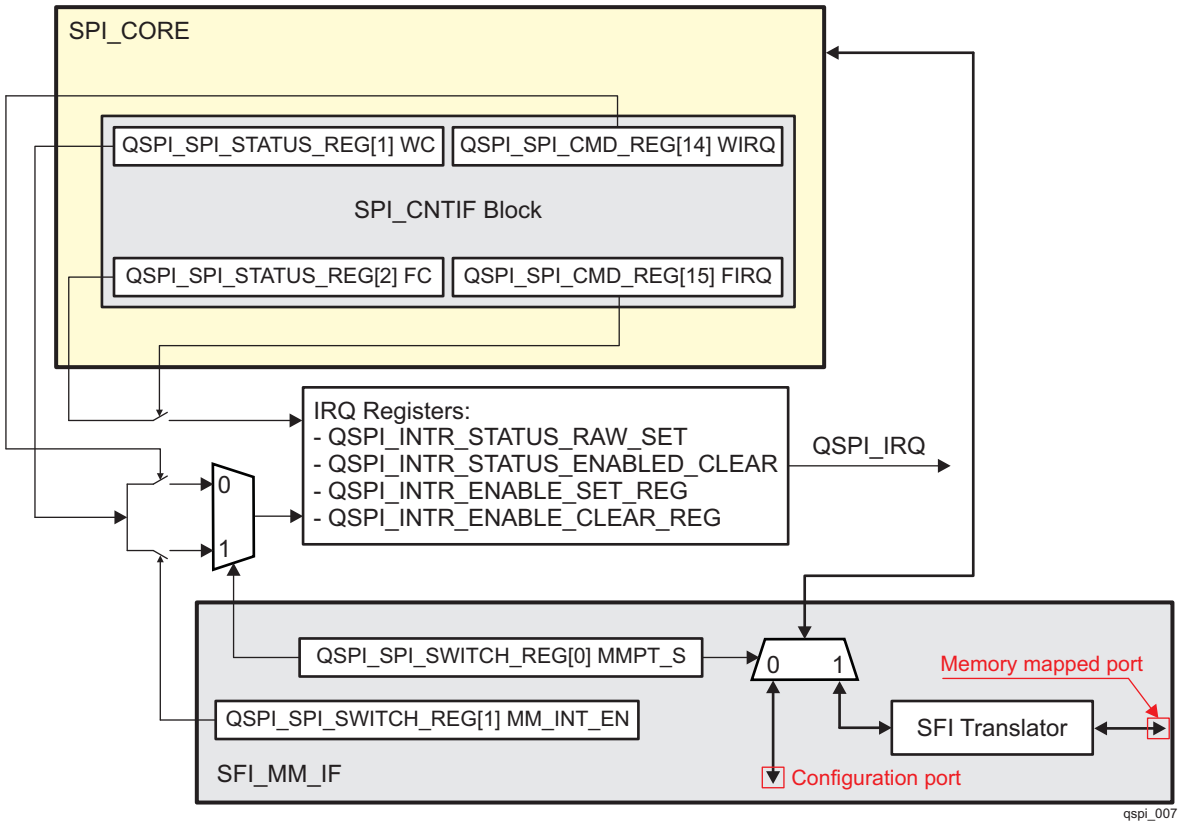
- Force-idle: The QSPI_FCLK clock is gated unconditionally by the QSPI.
- No-idle: The QSPI_FCLK clock is never gated by the QSPI.
- Smart-idle: The QSPI_FCLK clock is gated by the QSPI, depending on its internal requirements.

18.4.4.3 QSPI Interrupt Requests

The QSPI generates one interrupt request which is connected to the IRQ_CROSSBAR module. This interrupt request, QSPI_IRQ, is connected to the IRQ_CROSSBAR_343 input. The QSPI_IRQ interrupt line can be activated by one of the interrupt events listed in Table 18-279.

Figure 18-77 shows a logical representation of the QSPI interrupt generation scheme.

Figure 18-77. Logical Representation of the QSPI Interrupt Generation Scheme



QSPI_SPI_STATUS_REG[1] WC and QSPI_SPI_STATUS_REG[2] FC are status bits indicating whether word or frame transfer is complete. Setting the corresponding interrupt enable bit (WIRQ or FIRQ) in the QSPI_SPI_CMD_REG register allows these events (WC and FC) to generate an interrupt. The WC and FC bits are reset every time the user writes to the QSPI_SPI_CMD_REG register or reads the QSPI_SPI_STATUS_REG register. This is done to keep control parameters from changing the interface protocol signals while a transfer is in progress. Additionally, the QSPI_SPI_SWITCH_REG[1] MM_INT_EN bit is used to enable or disable the word complete interrupt during operations using the memory-mapped port.

When the QSPI_SPI_CMD_REG[14] WIRQ and QSPI_SPI_CMD_REG[15] FIRQ bits are set to 0x1 the following applies:

- The QSPI activates its interrupt line only if the interrupts are enabled by setting to 0x1 the corresponding bits in the QSPI_INTR_ENABLE_SET_REG register. These interrupts can be disabled by setting the corresponding bits in the QSPI_INTR_ENABLE_CLEAR_REG register to 0x1.
- After an interrupt has been serviced, software must clear the corresponding status flag. This is done by setting the corresponding bit in the QSPI_INTR_STATUS_ENABLED_CLEAR register to 0x1, which also clears the corresponding bit in the QSPI_INTR_STATUS_RAW_SET register. The status flags in the QSPI_INTR_STATUS_RAW_SET register are set even if the corresponding interrupt is disabled unlike those in the QSPI_INTR_STATUS_ENABLED_CLEAR register, which are set only if the corresponding interrupt is enabled.
- The QSPI also generates an interrupt if a certain bit in the QSPI_INTR_STATUS_RAW_SET register is set to 0x1 and the corresponding interrupt is enabled through the QSPI_INTR_ENABLE_SET_REG register. This feature is useful during user software debugging. In addition, even if interrupts are not enabled a corresponding raw flag in the QSPI_INTR_STATUS_RAW_SET register is set to 0x1 when an IRQ condition occurs.
- Even if interrupts are not enabled, a certain status bit in the QSPI_INTR_STATUS_RAW_SET register can also be cleared by setting to 0x1 the corresponding bit in the

QSPI_INTR_STATUS_ENABLED_CLEAR register.

It must be considered that the previously described scenario applies if the QSPI_SPI_CMD_REG[14] WIRQ and QSPI_SPI_CMD_REG[15] FIRQ bits are set to 0x1.

NOTE: The QSPI_IRQ interrupt line is activated only if at least one of the following conditions is met:

- The word complete interrupt is enabled:
  - during operations using the memory-mapped port by setting to 0x1 both the QSPI_SPI_SWITCH_REG[1] MM_INT_EN and QSPI_INTR_ENABLE_SET_REG[1] WIRQ_ENA_SET bits.
  - during operations using the configuration port by setting to 0x1 both the QSPI_SPI_CMD_REG[14] WIRQ and QSPI_INTR_ENABLE_SET_REG[1] WIRQ_ENA_SET bits.
- The frame complete interrupt is enabled setting to 0x1 both the QSPI_SPI_CMD_REG[15] FIRQ and QSPI_INTR_ENABLE_SET_REG[0] FIRQ_ENA_SET bits.

The QSPI_IRQ interrupt line is also activated when both the conditions are met.

Table 18-279 lists the event flags and the corresponding mask bits of the sources which can cause interrupts.

Table 18-279. QSPI Events

Event Flag	Event Mask	Description
QSPI_INTR_STATUS_RAW_SET[1] WIRQ_RAW QSPI_INTR_STATUS_ENABLED_CLEAR[1] WIRQ_ENA QSPI_SPI_STATUS_REG[1] WC	QSPI_INTR_ENABLE_SET_REG[1] WIRQ_ENA_SET QSPI_INTR_ENABLE_CLEAR_REG[1] WIRQ_ENA_CLR QSPI_SPI_CMD_REG[14] WIRQ	Word complete interrupt event. Asserted each time after a word is transferred or received.
QSPI_INTR_STATUS_RAW_SET[0] FIRQ_RAW QSPI_INTR_STATUS_ENABLED_CLEAR[0] FIRQ_ENA QSPI_SPI_STATUS_REG[2] FC	QSPI_INTR_ENABLE_SET_REG[0] FIRQ_ENA_SET QSPI_INTR_ENABLE_CLEAR_REG[0] FIRQ_ENA_CLR QSPI_SPI_CMD_REG[15] FIRQ	Frame complete interrupt event. Asserted each time after a frame is transferred or received.

18.4.4.4 QSPI Memory Regions

Two memory regions are associated with the QSPI. The first memory region is dedicated to the configuration port. Using this memory region, all internal registers can be programmed and serial transfers made from the four supported external SPI devices. The L3_MAIN start address at which the configuration port is available is 0x4B30 0000. The second memory region is associated mainly with the memory-mapped port and is used for communication directly with one of the four supported external SPI devices. This memory region starts at 0x5C00 0000 and ends at 0x5FFF FFFF L3_MAIN address.

The CTRL_CORE_CONTROL_IO_2[10:8] QSPI_MEMMAPPED_CS bit field provides a functionality for remapping the previously described address space which starts at 0x5C00 0000 L3_MAIN address to one of the four supported chip selects or to the configuration registers. The CTRL_CORE_CONTROL_IO_2 register resides in the CTRL_MODULE_CORE.

It is important to keep in mind that the configuration port provides an access to all the QSPI registers listed in Table 18-281. These are configuration registers and also four data registers. The configuration registers are used to configure typical SPI and serial flash memory settings and the four data registers are used for read and write operations. When communicating with an external SPI device (but not an SPI flash memory) the SPI_CORE module should be used and the data exchanged is available thorough these four data registers, which can be accessed only through the configuration port. When a communication with an external SPI flash memory is desired, the memory-mapped port should be used.

In other words, to read from an external SPI flash memory, first configure the QSPI through the configuration port and then perform a read through the memory-mapped port.

18.4.5 QSPI Register Manual

18.4.5.1 QSPI Instance Summary

Table 18-280. QSPI Instance Summary

Module Name	Module Base Address	Size
QSPI	0x4B30 0000	1MiB

18.4.5.2 QSPI registers

18.4.5.2.1 QSPI Register Summary

Table 18-281. QSPI Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	QSPI Base Address
QSPI_PID	R	32	0x0000 0000	0x4B30 0000
QSPI_SYSCONFIG	RW	32	0x0000 0010	0x4B30 0010
QSPI_INTR_STATUS_RAW_SET	RW	32	0x0000 0020	0x4B30 0020
QSPI_INTR_STATUS_ENABLED_CLEAR	RW	32	0x0000 0024	0x4B30 0024
QSPI_INTR_ENABLE_SET_REG	RW	32	0x0000 0028	0x4B30 0028
QSPI_INTR_ENABLE_CLEAR_REG	RW	32	0x0000 002C	0x4B30 002C
QSPI_INTC_EOI_REG	RW	32	0x0000 0030	0x4B30 0030
QSPI_SPI_CLOCK_CNTRL_REG	RW	32	0x0000 0040	0x4B30 0040
QSPI_SPI_DC_REG	RW	32	0x0000 0044	0x4B30 0044
QSPI_SPI_CMD_REG	RW	32	0x0000 0048	0x4B30 0048
QSPI_SPI_STATUS_REG	R	32	0x0000 004C	0x4B30 004C
QSPI_SPI_DATA_REG	RW	32	0x0000 0050	0x4B30 0050
QSPI_SPI_SETUP0_REG	RW	32	0x0000 0054	0x4B30 0054
QSPI_SPI_SETUP1_REG	RW	32	0x0000 0058	0x4B30 0058
QSPI_SPI_SETUP2_REG	RW	32	0x0000 005C	0x4B30 005C
QSPI_SPI_SETUP3_REG	RW	32	0x0000 0060	0x4B30 0060
QSPI_SPI_SWITCH_REG	RW	32	0x0000 0064	0x4B30 0064
QSPI_SPI_DATA_REG_1	RW	32	0x0000 0068	0x4B30 0068
QSPI_SPI_DATA_REG_2	RW	32	0x0000 006C	0x4B30 006C
QSPI_SPI_DATA_REG_3	RW	32	0x0000 0070	0x4B30 0070

18.4.5.2.2 QSPI Register Description

Table 18-282. QSPI_PID

Address Offset	0x0000 0000		
Physical Address	0x4B30 0000	**Instance**	QSPI
Description	Revision register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REVISION																															

Bits	Field Name	Description	Type	Reset
31:0	REVISION	IP Revision	R	TI Internal data

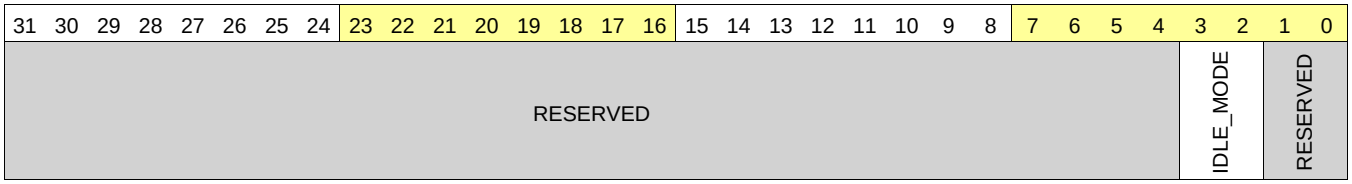
Table 18-283. Register Call Summary for Register QSPI_PID

Quad Serial Peripheral Interface

- SPI Control Interface: [0]
- QSPI Register Summary: [1]

Table 18-284. QSPI_SYSCONFIG

Address Offset	0x0000 0010		
Physical Address	0x4B30 0010	**Instance**	QSPI
Description			
Type	RW		



Bits	Field Name	Description	Type	Reset
31:4	RESERVED		R	0x2
3:2	IDLE_MODE	Configuration of the local target state management mode. By definition, target can handle read/write transaction as long as it is out of IDLE state. 0x0: Force-idle mode 0x1: No-idle mode 0x2: Smart-idle mode 0x3: Reserved.	RW	0x2
1:0	RESERVED		R	0x0

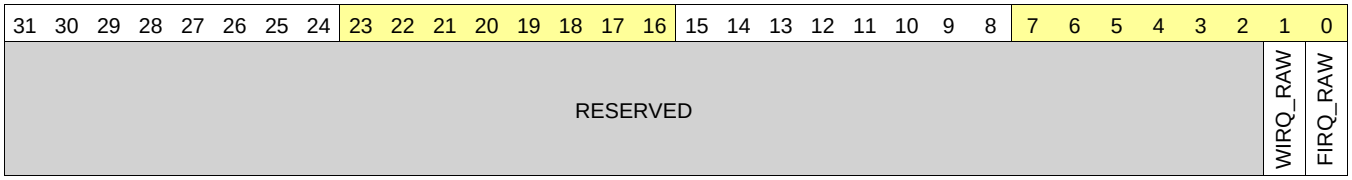
Table 18-285. Register Call Summary for Register QSPI_SYSCONFIG

Quad Serial Peripheral Interface

- QSPI Clock Configuration: [0]
- QSPI Register Summary: [1]

Table 18-286. QSPI_INTR_STATUS_RAW_SET

Address Offset	0x0000 0020		
Physical Address	0x4B30 0020	**Instance**	QSPI
Description	This register contains raw interrupt status flags.		
Type	RW		



Bits	Field Name	Description	Type	Reset
31:2	RESERVED		RW	0x0
1	WIRQ_RAW	Word Interrupt Status. Read indicates the raw status. Read: 0x0: No interrupt 0x1: Interrupt Write: 0x0: Has no effect 0x1: Sets this raw status bit	RW	0x0
0	FIRQ_RAW	Frame Interrupt Status. Read indicates the raw status. Read: 0x0: No interrupt 0x1: Interrupt Write: 0x0: Has no effect 0x1: Sets this raw status bit	RW	0x0

Table 18-287. Register Call Summary for Register QSPI_INTR_STATUS_RAW_SET

Quad Serial Peripheral Interface
- QSPI Interrupt Requests: [0][1][2][3][4][5][6]
- QSPI Register Summary: [7]

Table 18-288. QSPI_INTR_STATUS_ENABLED_CLEAR

Address Offset	0x0000 0024		
Physical Address	0x4B30 0024	**Instance**	QSPI
Description	This register contains status flags of the enabled interrupts.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														WIRQ_ENA	FIRQ_ENA

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1	WIRQ_ENA	Word Interrupt Enabled Status. Read indicates enabled status. Read: 0x0: No interrupt 0x1: Interrupt Write: 0x0: Has no effect 0x1: Clears the word interrupt status flag. The corresponding raw status flag is also cleared.	RW	0x0

Bits	Field Name	Description	Type	Reset
0	FIRQ_ENA	Frame Interrupt Enabled Status. Read indicates enabled status. Read: 0x0: No interrupt 0x1: Interrupt Write: 0x0: Has no effect 0x1: Clears the frame interrupt status flag. The corresponding raw status flag is also cleared.	RW	0x0

Table 18-289. Register Call Summary for Register QSPI_INTR_STATUS_ENABLED_CLEAR

Quad Serial Peripheral Interface
- QSPI Interrupt Requests: [0][1][2][3][4]
- QSPI Register Summary: [5]

Table 18-290. QSPI_INTR_ENABLE_SET_REG

Address Offset	0x0000 0028		
Physical Address	0x4B30 0028	**Instance**	QSPI
Description	This register enables the interrupts.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														WIRQ_ENA_SET	FIRQ_ENA_SET

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1	WIRQ_ENA_SET	Word interrupt enable. Read: 0x0: Word interrupt is disabled 0x1: Word interrupt enabled Write: 0x0: Has no effect 0x1: Enables the word interrupt	RW	0x0
0	FIRQ_ENA_SET	Frame interrupt enable. Read: 0x0: Frame interrupt is disabled 0x1: Frame interrupt is enabled Write: 0x0: Has no effect 0x1: Enables the frame interrupt	RW	0x0

Table 18-291. Register Call Summary for Register QSPI_INTR_ENABLE_SET_REG

Quad Serial Peripheral Interface

- QSPI Interrupt Requests: [0][1][2][3][4][5][6]
- QSPI Register Summary: [7]

Table 18-292. QSPI_INTR_ENABLE_CLEAR_REG

Address Offset	0x0000 002C		
Physical Address	0x4B30 002C	**Instance**	QSPI
Description	This register disables the interrupts.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														WIRQ_ENA_CLR	FIRQ_ENA_CLR
R-0x0																														RW-0x0	RW-0x0

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1	WIRQ_ENA_CLR	Word interrupt disable. Read: 0x0: Word interrupt is disabled 0x1: Word interrupt is enabled Write: 0x0: Has no effect 0x1: Clears the word interrupt	RW	0x0
0	FIRQ_ENA_CLR	Frame interrupt disable. Read: 0x0: Frame interrupt is disabled 0x1: Frame interrupt is enabled Write: 0x0: Has no effect 0x1: Clears the frame interrupt	RW	0x0

Table 18-293. Register Call Summary for Register QSPI_INTR_ENABLE_CLEAR_REG

Quad Serial Peripheral Interface

- QSPI Interrupt Requests: [0][1][2]
- QSPI Register Summary: [3]

Table 18-294. QSPI_INTC_EOI_REG

Address Offset	0x0000 0030		
Physical Address	0x4B30 0030	**Instance**	QSPI
Description	Software End-Of-Interrupt: Allows the generation of further pulses on the interrupt line, if a new interrupt event is pending, when using the pulsed output. Unused when using the level interrupt line (depending on module integration).		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
EOI_VECTOR																															

Bits	Field Name	Description	Type	Reset
31:0	EOI_VECTOR	Number associated with the interrupt outputs. There is one interrupt output. Write 0x0 after servicing the interrupt to be able to generate another interrupt if pulse interrupts are used. Any other write value is ignored.	RW	0x0

Table 18-295. Register Call Summary for Register QSPI_INTC_EOI_REG

Quad Serial Peripheral Interface
- QSPI Register Summary: [0]

Table 18-296. QSPI_SPI_CLOCK_CNTRL_REG

Address Offset	0x0000 0040		
Physical Address	0x4B30 0040	**Instance**	QSPI
Description	This register controls the external SPI clock generation. This register can only be written when the QSPI module is not busy, as identified by the QSPI_SPI_STATUS_REG[0] BUSY bit. RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CLKEN	RESERVED															DCLK_DIV															

Bits	Field Name	Description	Type	Reset
31	CLKEN	External SPI clock (qspi1_sclk) enable. 0x0: The qspi1_sclk clock is turned off 0x1: The qspi1_sclk clock is enabled	RW	0x0
30:16	RESERVED		R	0x0
15:0	DCLK_DIV	Divide ratio for the external SPI clock (qspi1_sclk)	RW	0x0

Table 18-297. Register Call Summary for Register QSPI_SPI_CLOCK_CNTRL_REG

Quad Serial Peripheral Interface
- SPI Control Interface: [0]
- SPI Clock Generator: [1][2]
- QSPI Register Summary: [3]

Table 18-298. QSPI_SPI_DC_REG

Address Offset	0x0000 0044		
Physical Address	0x4B30 0044	**Instance**	QSPI
Description	This register controls the different modes for each output chip select. This register can only be written when the QSPI module is not busy, as identified by the QSPI_SPI_STATUS_REG[0] BUSY bit.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED			DD3		CKPH3	CSP3	CKP3	RESERVED			DD2		CKPH2	CSP2	CKP2	RESERVED			DD1		CKPH1	CSP1	CKP1	RESERVED			DD0		CKPH0	CSP0	CKP0

Bits	Field Name	Description	Type	Reset
31:29	RESERVED		R	0x0
28:27	DD3	Data delay for chip select 3 0x0: Data is output on the same cycle as the qspi1_cs[3] goes active 0x1: Data is output 1 qspi1_sclk cycle after the qspi1_cs[3] goes active 0x2: Data is output 2 qspi1_sclk cycles after the qspi1_cs[3] goes active 0x3: Data is output 3 qspi1_sclk cycles after the qspi1_cs[3] goes active	RW	0x0
26	CKPH3	Clock phase for chip select 3. If CKP3 = 0: 0x0: Data shifted out on falling edge; input on falling edge 0x1: Data shifted out on rising edge; input on rising edge If CKP3 = 1: 0x0: Data shifted out on rising edge; input on rising edge 0x1: Data shifted out on falling edge; input on falling edge	RW	0x0
25	CSP3	Chip select polarity for chip select 3. 0x0: Active low 0x1: Active high	RW	0x0
24	CKP3	Clock polarity for chip select 3. 0x0: When there are no data transfers the qspi1_sclk is '0' 0x1: When there are no data transfers the qspi1_sclk is '1'	RW	0x0
23:21	RESERVED		R	0x0
20:19	DD2	Data delay for chip select 2 0x0: Data is output on the same cycle as the qspi1_cs[2] goes active 0x1: Data is output 1 qspi1_sclk cycle after the qspi1_cs[2] goes active 0x2: Data is output 2 qspi1_sclk cycles after the qspi1_cs[2] goes active 0x3: Data is output 3 qspi1_sclk cycles after the qspi1_cs[2] goes active	RW	0x0
18	CKPH2	Clock phase for chip select 2. If CKP2 = 0: 0x0: Data shifted out on falling edge; input on falling edge 0x1: Data shifted out on rising edge; input on rising edge If CKP2 = 1: 0x0: Data shifted out on rising edge; input on rising edge 0x1: Data shifted out on falling edge; input on falling edge	RW	0x0
17	CSP2	Chip select polarity for chip select 2. 0x0: Active low 0x1: Active high	RW	0x0

Bits	Field Name	Description	Type	Reset
16	CKP2	Clock polarity for chip select 2. 0x0: When there are no data transfers the qspi1_sclk is '0' 0x1: When there are no data transfers the qspi1_sclk is '1'	RW	0x0
15:13	RESERVED		R	0x0
12:11	DD1	Data delay for chip select 1 0x0: Data is output on the same cycle as the qspi1_cs[1] goes active 0x1: Data is output 1 qspi1_sclk cycle after the qspi1_cs[1] goes active 0x2: Data is output 2 qspi1_sclk cycles after the qspi1_cs[1] goes active 0x3: Data is output 3 qspi1_sclk cycles after the qspi1_cs[1] goes active	RW	0x0
10	CKPH1	Clock phase for chip select 1. If CKP1 = 0: 0x0: Data shifted out on falling edge; input on falling edge 0x1: Data shifted out on rising edge; input on rising edge If CKP1 = 1: 0x0: Data shifted out on rising edge; input on rising edge 0x1: Data shifted out on falling edge; input on falling edge	RW	0x0
9	CSP1	Chip select polarity for chip select 1. 0x0: Active low 0x1: Active high	RW	0x0
8	CKP1	Clock polarity for chip select 1. 0x0: When there are no data transfers the qspi1_sclk is '0' 0x1: When there are no data transfers the qspi1_sclk is '1'	RW	0x0
7:5	RESERVED		R	0x0
4:3	DD0	Data delay for chip select 0 0x0: Data is output on the same cycle as the qspi1_cs[0] goes active 0x1: Data is output 1 qspi1_sclk cycle after the qspi1_cs[0] goes active 0x2: Data is output 2 qspi1_sclk cycles after the qspi1_cs[0] goes active 0x3: Data is output 3 qspi1_sclk cycles after the qspi1_cs[0] goes active	RW	0x0
2	CKPH0	Clock phase for chip select 0. If CKP0 = 0: 0x0: Data shifted out on falling edge; input on falling edge 0x1: Data shifted out on rising edge; input on rising edge If CKP0 = 1: 0x0: Data shifted out on rising edge; input on rising edge 0x1: Data shifted out on falling edge; input on falling edge	RW	0x0
1	CSP0	Chip select polarity for chip select 0. 0x0: Active low 0x1: Active high	RW	0x0
0	CKP0	Clock polarity for chip select 0. 0x0: When there are no data transfers the qspi1_sclk is '0' 0x1: When there are no data transfers the qspi1_sclk is '1'	RW	0x0

Table 18-299. Register Call Summary for Register QSPI_SPI_DC_REG

Quad Serial Peripheral Interface

- SPI Control Interface: [0]
- SPI Data Shifter: [1][2][3][4]
- QSPI Register Summary: [5]

Table 18-300. QSPI_SPI_CMD_REG

Address Offset	0x0000 0048		
Physical Address	0x4B30 0048	**Instance**	QSPI
Description	This register sets up the SPI command. This register can only be written when the QSPI module is not busy, as identified by the QSPI_SPI_STATUS_REG[0] BUSY bit.		
Type	RW		

31 30	29 28	27 26	25 24	23 22 21 20 19 18 17 16	15 14	13 12	11 10 9 8	7 6 5 4 3 2 1 0
RESERVED	CSNUM	RESERVED	WLEN	CMD	FIRQ WIRQ	RESERVED	FLEN	

Bits	Field Name	Description	Type	Reset
31:30	RESERVED		R	0x0
29:28	CSNUM	Device select. Sets the active chip select for the current transfer. 0x0: Chip Select 0 active 0x1: Chip Select 1 active 0x2: Chip Select 2 active 0x3: Chip Select 3 active	RW	0x0
27:26	RESERVED		R	0x0
25:19	WLEN	Word length. Sets the size of the individual transfers from 1 to 128 bits. When a word length greater than 32 bits is configured, not only the QSPI_SPI_DATA_REG register, but also the QSPI_SPI_DATA_REG_1, QSPI_SPI_DATA_REG_2, QSPI_SPI_DATA_REG_3 are used. One or all of these registers are used depending on the length of words transferred. 0x0: 1 bit 0x1: 2 bits ... 0x7F: 128 bits	RW	0x0
18:16	CMD	Transfer command. 0x0: Reserved 0x1: 4-pin Read Single 0x2: 4-pin Write Single 0x3: 4-pin Read Dual 0x4: Reserved 0x5: 3-pin Read Single 0x6: 3-pin Write Single 0x7: 6-pin Read Quad	RW	0x0
15	FIRQ	Frame complete interrupt enable. 0x0: The interrupt is disabled 0x1: The interrupt is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
14	WIRQ	Word complete interrupt enable 0x0: The interrupt is disabled 0x1: The interrupt is enabled	RW	0x0
13:12	RESERVED		R	0x0
11:0	FLEN	Frame Length. 0x0: 1 word 0x1: 2 words ... 0xFFF: 4096 words	RW	0x0

Table 18-301. Register Call Summary for Register QSPI_SPI_CMD_REG

Quad Serial Peripheral Interface
- SPI Control Interface: [0][1][2][3][4]
- SPI Clock Generator: [5]
- SPI Control State-Machine: [6][7][8][9]
- QSPI Interrupt Requests: [10][11][12][13][14][15][16][17][18][19]
- QSPI Register Summary: [20]

Table 18-302. QSPI_SPI_STATUS_REG

Address Offset	0x0000 004C		
Physical Address	0x4B30 004C	**Instance**	QSPI
Description	This register contains indicators to allow the user to monitor the progression of a frame transfer. This register can only be written when the QSPI module is not busy, as identified by the QSPI_SPI_STATUS_REG[0] BUSY bit.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								WDCNT								RESERVED													FC	WC	BUSY

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:16	WDCNT	Word count. This field will reflect the 1-4096 words transferred	R	0x0
15:3	RESERVED		R	0x0
2	FC	Frame complete. This bit is set after the transmision of all the requested words completes. This bit is reset when QSPI_SPI_STATUS_REG register is read. 0x0: Transfer is not complete 0x1: Transfer is complete	R	0x0
1	WC	Word complete. This bit is set after each word transfer completes. This bit is reset when QSPI_SPI_STATUS_REG register is read. 0x0: Word transfer is not complete 0x1: Word transfer is complete	R	0x0
0	BUSY	Busy bit. Active transfer in progress. This bit is only set during an active word transfer. Between words it is cleared. 0x0: Idle 0x1: Busy	R	0x0

Table 18-303. Register Call Summary for Register QSPI_SPI_STATUS_REG

Quad Serial Peripheral Interface

- SPI Control Interface: [0][1][2]
- SPI Clock Generator: [3]
- SPI Control State-Machine: [4][5]
- QSPI Interrupt Requests: [6][7][8][9][10]
- QSPI Register Summary: [11]
- QSPI Register Description: [12][13][14][15][16][17][18][19][20][21]

Table 18-304. QSPI_SPI_DATA_REG

Address Offset	0x0000 0050		
Physical Address	0x4B30 0050	**Instance**	QSPI
Description	The data received in this register is shifted to the LSB position and the content of the register is shifted to the left. This register acts as the first 32-bit register of the 128-bit shift in/out register. This register is cleared between reads or writes and can only be written when the QSPI module is not busy, as identified by the QSPI_SPI_STATUS_REG[0] BUSY bit.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA																															

Bits	Field Name	Description	Type	Reset
31:0	DATA	Data register for read and write operations	RW	0x0

Table 18-305. Register Call Summary for Register QSPI_SPI_DATA_REG

Quad Serial Peripheral Interface

- SPI Control Interface: [0][1][2][3][4][5][6]
- QSPI Register Summary: [7]
- QSPI Register Description: [8]

Table 18-306. QSPI_SPI_SETUP0_REG

Address Offset	0x0000 0054		
Physical Address	0x4B30 0054	**Instance**	QSPI
Description	This register contains the read/write command setup for the memory mapped protocol translator (effecting chip select 0 output). By default (reset), the device uses a write command of 2, read command of 3 and address bytes number of 3. This default covers most of the serial flash devices, but can be changed.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED			NUM_D_BITS					WCMD								RESERVED		READ_TYPE		NUM_D_BYTES		NUM_A_BYTES		RCMD							

Bits	Field Name	Description	Type	Reset
31:29	RESERVED		R	0x0
28:24	NUM_D_BITS	Number of dummy bits to use if NUM_D_BYTES = 0x0	RW	0x0
23:16	WCMD	Write command	RW	0x2

Bits	Field Name	Description	Type	Reset
15:14	RESERVED		R	0x0
13:12	READ_TYPE	Determines if the read command is a single, dual or quad read mode command. 0x0: Normal read (all data input on qspi1_d[1]) 0x1: Dual read (odd bytes input on qspi1_d[1]; even bytes on qspi1_d[0]) 0x2: Normal read (all data input on qspi1_d[1]) 0x3: Quad read (uses also qspi1_d[2] and qspi1_d[3])	RW	0x0
11:10	NUM_D_BYTES	Number of dummy bytes to be used for fast read. 0x0: No dummy bytes required. Use the value in NUM_D_BITS 0x1: Use 8 bits 0x2: Use 16 bits 0x3: Use 24 bits	RW	0x0
9:8	NUM_A_BYTES	Number of address bytes to be sent. 0x0: 1 byte 0x1: 2 bytes 0x2: 3 bytes 0x3: 4 bytes	RW	0x2
7:0	RCMD	Read Command	RW	0x3

Table 18-307. Register Call Summary for Register QSPI_SPI_SETUP0_REG

Quad Serial Peripheral Interface
- SFI Register Control: [0]
- QSPI Register Summary: [1]

Table 18-308. QSPI_SPI_SETUP1_REG

Address Offset	0x0000 0058		
Physical Address	0x4B30 0058	**Instance**	QSPI
Description	This register contains the read/write command setup for the memory mapped protocol translator (effecting chip select 1 output). By default (reset), the device uses a write command of 2, read command of 3 and address bytes number of 3. This default covers most of the serial flash devices, but can be changed.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED			NUM_D_BITS					WCMD								RESERVED		READ_TYPE		NUM_D_BYTES		NUM_A_BYTES		RCMD							

Bits	Field Name	Description	Type	Reset
31:29	RESERVED		R	0x0
28:24	NUM_D_BITS	Number of dummy bits to use if NUM_D_BYTES = 0x0	RW	0x0
23:16	WCMD	Write command	RW	0x2
15:14	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
13:12	READ_TYPE	Determines if the read command is a single, dual or quad read mode command. 0x0: Normal read (all data input on qspi1_d[1]) 0x1: Dual read (odd bytes input on qspi1_d[1]; even bytes on qspi1_d[0]) 0x2: Normal read (all data input on qspi1_d[1]) 0x3: Quad read (uses also qspi1_d[2] and qspi1_d[3])	RW	0x0
11:10	NUM_D_BYTES	Number of dummy bytes to be used for fast read. 0x0: No dummy bytes required. Use the value in NUM_D_BITS 0x1: Use 8 bits 0x2: Use 16 bits 0x3: Use 24 bits	RW	0x0
9:8	NUM_A_BYTES	Number of address bytes to be sent. 0x0: 1 byte 0x1: 2 bytes 0x2: 3 bytes 0x3: 4 bytes	RW	0x2
7:0	RCMD	Read Command	RW	0x3

Table 18-309. Register Call Summary for Register QSPI_SPI_SETUP1_REG

Quad Serial Peripheral Interface
- SFI Register Control: [0]
- QSPI Register Summary: [1]

Table 18-310. QSPI_SPI_SETUP2_REG

Address Offset	0x0000 005C		
Physical Address	0x4B30 005C	**Instance**	QSPI
Description	This register contains the read/write command setup for the memory mapped protocol translator (effecting chip select 2 output). By default (reset), the device uses a write command of 2, read command of 3 and address bytes number of 3. This default covers most of the serial flash devices, but can be changed.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED			NUM_D_BITS					WCMD								RESERVED		READ_TYPE		NUM_D_BYTES		NUM_A_BYTES		RCMD							

Bits	Field Name	Description	Type	Reset
31:29	RESERVED		R	0x0
28:24	NUM_D_BITS	Number of dummy bits to use if NUM_D_BYTES = 0x0	RW	0x0
23:16	WCMD	Write command	RW	0x2
15:14	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
13:12	READ_TYPE	Determines if the read command is a single, dual or quad read mode command. 0x0: Normal read (all data input on qspi1_d[1]) 0x1: Dual read (odd bytes input on qspi1_d[1]; even bytes on qspi1_d[0]) 0x2: Normal read (all data input on qspi1_d[1]) 0x3: Quad read (uses also qspi1_d[2] and qspi1_d[3])	RW	0x0
11:10	NUM_D_BYTES	Number of dummy bytes to be used for fast read. 0x0: No dummy bytes required. Use the value in NUM_D_BITS 0x1: Use 8 bits 0x2: Use 16 bits 0x3: Use 24 bits	RW	0x0
9:8	NUM_A_BYTES	Number of address bytes to be sent. 0x0: 1 byte 0x1: 2 bytes 0x2: 3 bytes 0x3: 4 bytes	RW	0x2
7:0	RCMD	Read Command	RW	0x3

Table 18-311. Register Call Summary for Register QSPI_SPI_SETUP2_REG

Quad Serial Peripheral Interface
- SFI Register Control: [0]
- QSPI Register Summary: [1]

Table 18-312. QSPI_SPI_SETUP3_REG

Address Offset	0x0000 0060		
Physical Address	0x4B30 0060	**Instance**	QSPI
Description	This register contains the read/write command setup for the memory mapped protocol translator (effecting chip select 3 output). By default (reset), the device uses a write command of 2, read command of 3 and address bytes number of 3. This default covers most of the serial flash devices, but can be changed.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED			NUM_D_BITS					WCMD								RESERVED		READ_TYPE		NUM_D_BYTES		NUM_A_BYTES		RCMD							

Bits	Field Name	Description	Type	Reset
31:29	RESERVED		R	0x0
28:24	NUM_D_BITS	Number of dummy bits to use if NUM_D_BYTES = 0x0	RW	0x0
23:16	WCMD	Write command	RW	0x2
15:14	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
13:12	READ_TYPE	Determines if the read command is a single, dual or quad read mode command. 0x0: Normal read (all data input on qspi1_d[1]) 0x1: Dual read (odd bytes input on qspi1_d[1]; even bytes on qspi1_d[0]) 0x2: Normal read (all data input on qspi1_d[1]) 0x3: Quad read (uses also qspi1_d[2] and qspi1_d[3])	RW	0x0
11:10	NUM_D_BYTES	Number of dummy bytes to be used for fast read. 0x0: No dummy bytes required. Use the value in NUM_D_BITS 0x1: Use 8 bits 0x2: Use 16 bits 0x3: Use 24 bits	RW	0x0
9:8	NUM_A_BYTES	Number of address bytes to be sent. 0x0: 1 byte 0x1: 2 bytes 0x2: 3 bytes 0x3: 4 bytes	RW	0x2
7:0	RCMD	Read Command	RW	0x3

Table 18-313. Register Call Summary for Register QSPI_SPI_SETUP3_REG

Quad Serial Peripheral Interface
- SFI Register Control: [0]
- QSPI Register Summary: [1]

Table 18-314. QSPI_SPI_SWITCH_REG

Address Offset	0x0000 0064		
Physical Address	0x4B30 0064	**Instance**	QSPI
Description	This register allows initiators to switch control of the SPI core port between the configuration port and the SFI translator. In addition, an interrupt enable field is defined which is used to enable or disable word complete interrupt generation in memory mapped mode.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														MM_INT_EN	MMPT_S

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1	MM_INT_EN	Memory mapped mode interrupt enable. 0x0: Word complete interrupt is disabled during memory mapped operations 0x1: Word complete interrupt is enabled for memory mapped operations	RW	0x0
0	MMPT_S	MPT select. 0x0: Configuration port is selected to control the SPI_CORE. 0x1: SFI translator is selected to control the SPI_CORE.	RW	0x0

Table 18-315. Register Call Summary for Register QSPI_SPI_SWITCH_REG

Quad Serial Peripheral Interface

- SFI Register Control: [0][1][2][3]
- QSPI Interrupt Requests: [4][5]
- QSPI Register Summary: [6]

Table 18-316. QSPI_SPI_DATA_REG_1

Address Offset	0x0000 0068		
Physical Address	0x4B30 0068	**Instance**	QSPI
Description	The data received in this register is shifted to the LSB position and the content of the register is shifted to the left. This register acts as the second 32-bit register of the 128-bit shift in/out register. This register is cleared between reads or writes and can only be written when the QSPI module is not busy, as identified by the QSPI_SPI_STATUS_REG[0] BUSY bit.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA																															

Bits	Field Name	Description	Type	Reset
31:0	DATA	Data register for read and write operations	RW	0x0

Table 18-317. Register Call Summary for Register QSPI_SPI_DATA_REG_1

Quad Serial Peripheral Interface

- SPI Control Interface: [0][1][2]
- QSPI Register Summary: [3]
- QSPI Register Description: [4]

Table 18-318. QSPI_SPI_DATA_REG_2

Address Offset	0x0000 006C		
Physical Address	0x4B30 006C	**Instance**	QSPI
Description	The data received in this register is shifted to the LSB position and the content of the register is shifted to the left. This register acts as the third 32-bit register of the 128-bit shift in/out register. This register is cleared between reads or writes and can only be written when the QSPI module is not busy, as identified by the QSPI_SPI_STATUS_REG[0] BUSY bit.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA																															

Bits	Field Name	Description	Type	Reset
31:0	DATA	Data register for read and write operations	RW	0x0

Table 18-319. Register Call Summary for Register QSPI_SPI_DATA_REG_2

Quad Serial Peripheral Interface

- SPI Control Interface: [0][1][2]
- QSPI Register Summary: [3]
- QSPI Register Description: [4]

Table 18-320. QSPI_SPI_DATA_REG_3

Address Offset	0x0000 0070		
Physical Address	0x4B30 0070	**Instance**	QSPI
Description	The data received in this register is shifted to the LSB position and the content of the register is shifted to the left. This register acts as the fourth 32-bit register of the 128-bit shift in/out register. This register is cleared between reads or writes and can only be written when the QSPI module is not busy, as identified by the QSPI_SPI_STATUS_REG[0] BUSY bit.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA																															

Bits	Field Name	Description	Type	Reset
31:0	DATA	Data register for read and write operations	RW	0x0

Table 18-321. Register Call Summary for Register QSPI_SPI_DATA_REG_3

Quad Serial Peripheral Interface

- SPI Control Interface: [0][1][2][3][4]
- QSPI Register Summary: [5]
- QSPI Register Description: [6]

18.5 Multichannel Audio Serial Port

This section describes the multichannel audio serial port (McASP).

18.5.1 McASP Overview

This section introduces the multichannel audio serial port (McASP) module and describes its main functions and connections in the device.

The McASP functions as a general-purpose audio serial port optimized to the requirements of various audio applications. The McASP module can operate in both transmit and receive modes. The McASP is useful for time-division multiplexed (TDM) stream, Inter-IC Sound (I2S) protocols reception and transmission as well as for an intercomponent digital audio interface transmission (DIT). The McASP has the flexibility to gluelessly connect to a Sony/Philips digital interface (S/PDIF) transmit physical layer component.

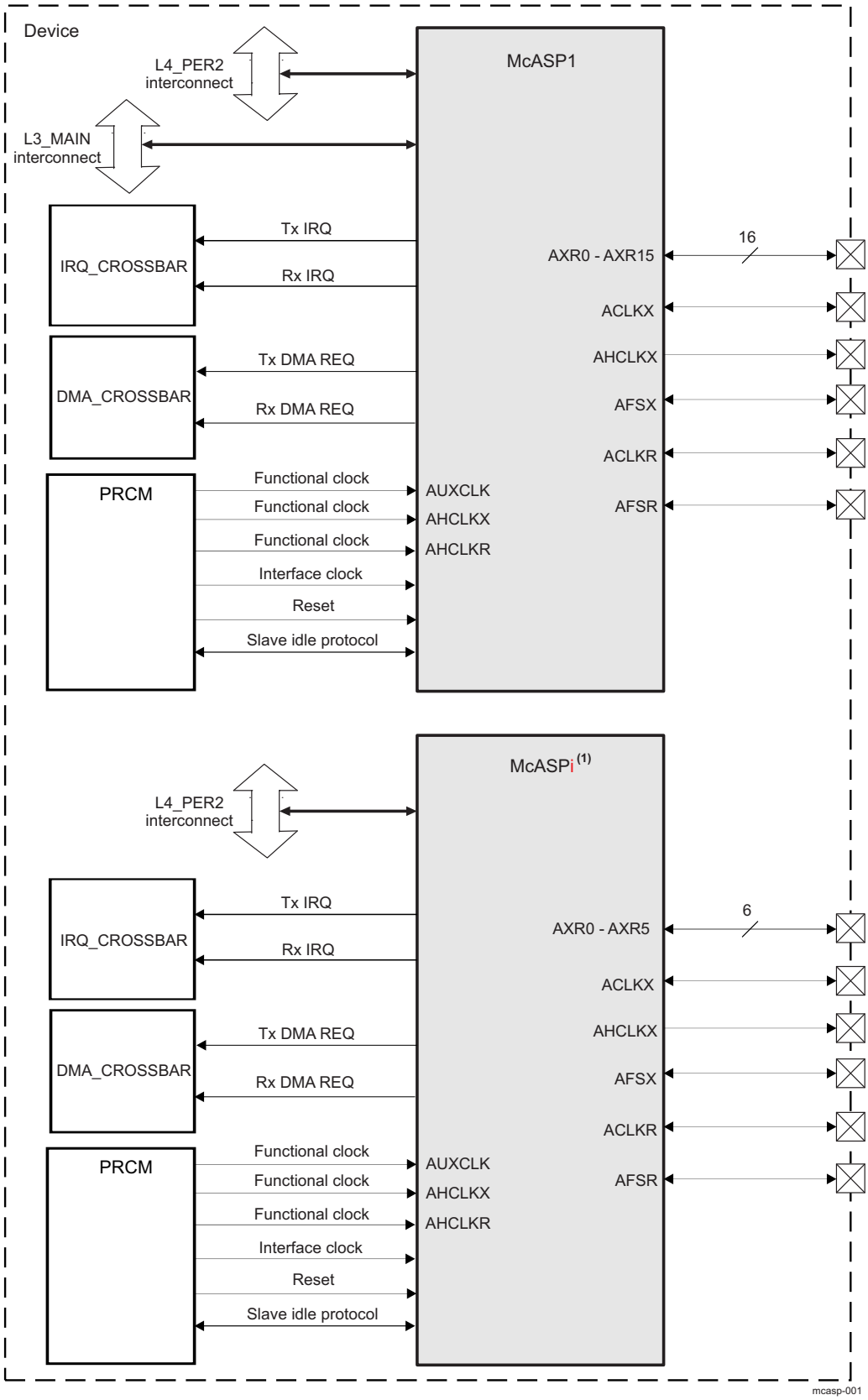
Although intercomponent digital audio interface reception (DIR) mode (i.e. S/PDIF stream receiving) is not natively supported by the McASP module, a specific TDM mode implementation for the McASP receivers allows an easy connection to external DIR components (for example, S/PDIF to I2S format converters).

The device have integrated three McASP modules with:

- McASP1 supporting up to 16 channels with independent TX/RX clock/sync domain
- McASP2 and McASP3 support up to 6 channels with independent TX/RX clock/sync domain.

Figure 18-78 shows the McASP modules in the device.

Figure 18-78. McASP Modules Overview



(1) i = 2 and 3

McASP module includes the following main features:

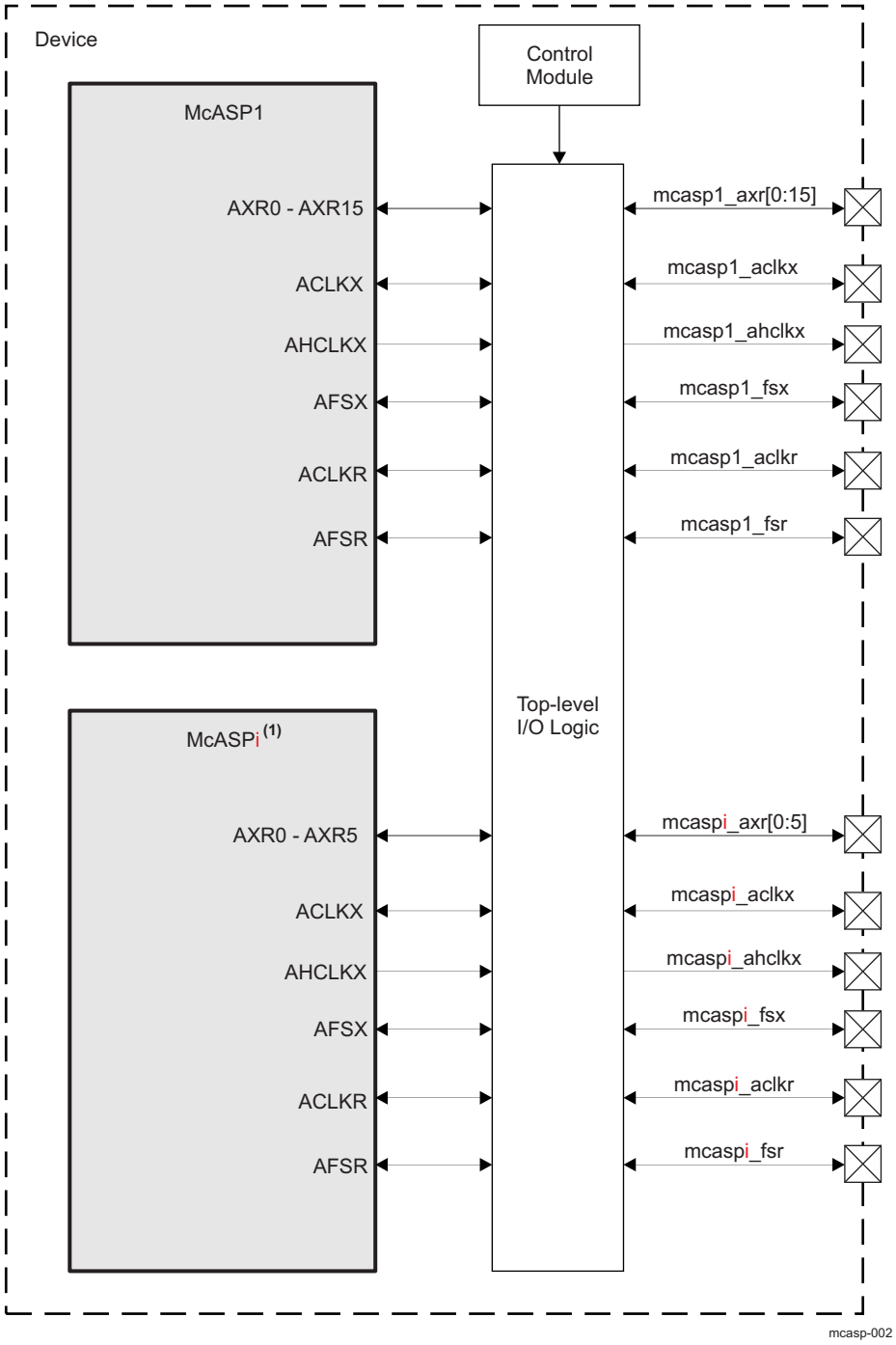
- McASP1 supports up to 16 channels and independent TX/RX clock/sync domains.
- Two modules (McASP2 and McASP3) supporting up to 6 channels each and independent TX/RX clock/sync domains.
- Independent serializer for each AXRx channel of each McASPx module.
- Idle request/acknowledge protocol
- A single 32-bit buffer per serializer for transmit and receive operations
- 2 x interconnect slave interface ports:
  - A configuration (CFG) port supplied with an internal L4-interconnect interface clock
  - A slave DMA data port synchronized with functional clock
- Two independent clock generator modules for transmit and receive.
  - Clocking flexibility allows the McASP to receive and transmit at different rates. For example, the McASP can receive data at 48 kHz but output up-sampled data at 96 kHz or 192 kHz.
- McASP module functional clock can be generated:
  - internally (master mode)
  - supplied over McASP serial interface (slave mode)
  - has a controllable functional clock divide ratio
- Independent transmit and receive modules, each includes:
  - Programmable clock and frame sync generator.
  - TDM streams from 2 to 32, and 384 time slots.
  - Support for time slot sizes of 8, 12, 16, 20, 24, 28, and 32 bits.
  - Data formatter for bit manipulation.
- Glueless connection to audio analog-to-digital converters (ADC), digital-to-analog converters (DAC), codec, digital audio interface receiver (DIR), and S/PDIF transmit physical layer components.
- Wide variety of I2S and similar bit-stream format.
- Integrated digital audio interface transmitter (DIT):
  - S/PDIF, IEC60958-1, AES-3 formats.
  - Enhanced channel status/user data RAM.
- 384-slot TDM with external digital audio interface receiver (DIR) device.
  - For DIR reception, an external DIR receiver integrated circuit should be used with I2S output format and connected to the McASP receive section.
- Support for 2x DMA requests (one per direction):
  - 1 level-sensitive transmit direct memory access (DMA) request common for all of the McASP serializers
  - 1 level-sensitive receive direct memory access (DMA) request common for all of the McASP serializers
  - All transmit DMA requests are mapped to the device DMA crossbar
- One transmit interrupt request common for all serializers
- One receive interrupt request common for all serializers
- Each of the Rx and Tx interrupts is propagated to different host processors via the device Interrupt Crossbar

NOTE: Because a serializer receive and transmit channels data is shared on the same McASP data pin, user can choose to have either Tx or Rx function from a serializer, not both at the same time.

18.5.2 McASP Environment

This section describes the McASP application fields from an environment point of view (external connections), along with the McASP connectivity options. This section also lists all of the possible interfaces and describes the protocol and data format used in each case. Figure 18-79 shows the McASP modules in their environment in the device.

Figure 18-79. McASP Environment



(1) i = 2 and 3

18.5.2.1 McASP Signals

Table 18-322 describes the McASP pins, their corresponding signal names at device level and specifies their links to functions.

Table 18-322. McASP I/O Signals

Module Pin Name	Device Level Signal Name	I/O[1]	Description	Module Pin Reset Value
		McASP1 module		
AXR0	mcasp1_axr[0]	I/O	Audio transmit/receive data - channel 0	HiZ
AXR1	mcasp1_axr[1]	I/O	Audio transmit/receive data - channel 1	HiZ
AXR2	mcasp1_axr[2]	I/O	Audio transmit/receive data - channel 2	HiZ
AXR3	mcasp1_axr[3]	I/O	Audio transmit/receive data - channel 3	HiZ
AXR4	mcasp1_axr[4]	I/O	Audio transmit/receive data - channel 4	HiZ
AXR5	mcasp1_axr[5]	I/O	Audio transmit/receive data - channel 5	HiZ
AXR6	mcasp1_axr[6]	I/O	Audio transmit/receive data - channel 6	HiZ
AXR7	mcasp1_axr[7]	I/O	Audio transmit/receive data - channel 7	HiZ
AXR8	mcasp1_axr[8]	I/O	Audio transmit/receive data - channel 8	HiZ
AXR9	mcasp1_axr[9]	I/O	Audio transmit/receive data - channel 9	HiZ
AXR10	mcasp1_axr[10]	I/O	Audio transmit/receive data - channel 10	HiZ
AXR11	mcasp1_axr[11]	I/O	Audio transmit/receive data - channel 11	HiZ
AXR12	mcasp1_axr[12]	I/O	Audio transmit/receive data - channel 12	HiZ
AXR13	mcasp1_axr[13]	I/O	Audio transmit/receive data - channel 13	HiZ
AXR14	mcasp1_axr[14]	I/O	Audio transmit/receive data - channel 14	HiZ
AXR15	mcasp1_axr[15]	I/O	Audio transmit/receive data - channel 15	HiZ
ACLKX	mcasp1_aclkx	I/O	Transmit bit clock	HiZ
AHCLKX	mcasp1_ahclkx	O	Transmit high-frequency master clock	HiZ
AFSX	mcasp1_fsx	I/O	Transmit frame synchronization	HiZ
ACLKR	mcasp1_aclkr	I/O	Receive bit clock	HiZ
AFSR	mcasp1_fsr	I/O	Receive frame synchronization	HiZ
		McASP2 module		
AXR0	mcasp2_axr[0]	I/O	Audio transmit/receive data - channel 0	HiZ
AXR1	mcasp2_axr[1]	I/O	Audio transmit/receive data - channel 1	HiZ
AXR2	mcasp2_axr[2]	I/O	Audio transmit/receive data - channel 2	HiZ
AXR3	mcasp2_axr[3]	I/O	Audio transmit/receive data - channel 3	HiZ
AXR4	mcasp2_axr[4]	I/O	Audio transmit/receive data - channel 4	HiZ
AXR5	mcasp2_axr[5]	I/O	Audio transmit/receive data - channel 5	HiZ
ACLKX	mcasp2_aclkx	I/O	Transmit bit clock	HiZ
AHCLKX	mcasp2_ahclkx	O	Transmit high-frequency master clock	HiZ
AFSX	mcasp2_fsx	I/O	Transmit frame synchronization	HiZ
ACLKR	mcasp2_aclkr	I/O	Receive bit clock	HiZ
AFSR	mcasp2_fsr	I/O	Receive frame synchronization	HiZ
		McASP3 module		
AXR0	mcasp3_axr[0]	I/O	Audio transmit/receive data - channel 0	HiZ
AXR1	mcasp3_axr[1]	I/O	Audio transmit/receive data - channel 1	HiZ
AXR2	mcasp3_axr[2]	I/O	Audio transmit/receive data - channel 2	HiZ
AXR3	mcasp3_axr[3]	I/O	Audio transmit/receive data - channel 3	HiZ
AXR4	mcasp3_axr[4]	I/O	Audio transmit/receive data - channel 4	HiZ
AXR5	mcasp3_axr[5]	I/O	Audio transmit/receive data - channel 5	HiZ
ACLKX	mcasp3_aclkx	I/O	Transmit bit clock	HiZ

(1) I = Input; O = Output; I/O = Bidirectional

Table 18-322. McASP I/O Signals (continued)

Module Pin Name	Device Level Signal Name	I/O[1]	Description	Module Pin Reset Value
AHCLKX	mcasp3_ahclkx	O	Transmit high-frequency master clock	HiZ
AFSX	mcasp3_fsx	I/O	Transmit frame synchronization	HiZ
ACLKR	mcasp3_aclkr	I/O	Receive bit clock	HiZ
AFSR	mcasp3_fsr	I/O	Receive frame synchronization	HiZ

NOTE: For the mcaspx_aclkx, mcaspx_ahclkx and mcaspx_aclkr signals to work properly, the INPUTENABLE bit of the appropriate CTRL_CORE_PAD_x registers should be set to 0x1 because of retiming purposes.

NOTE: The path from module pin to device pad(s) is defined at the device I/O logic level. The I/O logic maps the module signals to the different pads of the device and is programmable in the Control Module registers. For more information, refer to the Section 13.4.6.1, *Pad Configuration Registers*, and Section 13.5, *Control Module Register Manual*, in Chapter 13, *Control Module*.

NOTE: Except the pad configuration registers, additional Control Module registers have to be configured to map McASP2 and McASP3 signals to the device pads. The MUXMODE field of the corresponding pad configuration register has to be set to 0xF and the corresponding bit in CTRL_CORE_SMA_SW_14 and CTRL_CORE_SMA_SW_15 has to be set to 0x1.

NOTE: A serializer AXR data pin is shared between the transmit and receive logic of that serializer. The direction of data is determined in the MCASP_PDIR and the function (Tx or Rx) is selected in the corresponding serializer control register MCASP_XRSRCTLn.

18.5.2.2 Protocols and Data Formats

18.5.2.2.1 Protocols Supported

The McASP supports a wide variety of protocols:

- Transmit section supports:
  - Wide variety of I2S and similar bit-stream formats.
  - TDM streams from 2 to 32 time slots.
  - S/PDIF, IEC60958-1, AES-3 formats.
- Receive section supports:
  - Wide variety of I2S and similar bit-stream formats.
  - TDM streams from 2 to 32 time slots.
  - TDM stream of 384 time slots specifically designed for easy interface to external digital interface receiver (DIR) device transmitting DIR frames to McASP using the I2S protocol (one time slot for each DIR subframe).

The transmit and receive sections of the module may be individually programmed to support the following options on the basic serial protocol:

- Programmable clock and frame sync polarity (rising or falling edge): ACLKR/X, AHCLKR/X, and AFSR/X.
- Slot length (number of bits per time slot): 8, 12, 16, 20, 24, 28, 32 bits supported.
- Word length (bits per word): 8, 12, 16, 20, 24, 28, 32 bits; always less than or equal to the time slot

length.

- First-bit data delay: 0, 1, 2 bit clocks.
- Left/right alignment of word inside slot.
- Bit order: MSB first or LSB first.
- Bit mask/pad/rotate function.
  - Automatically aligns data internally in either Q31 or integer formats.
  - Automatically masks nonsignificant bits (sets to 0, 1, or extends value of another bit).

NOTE: In I2S mode, the transmit and receive sections can support simultaneous transfers on up to all serial data pins operating as 192 kHz stereo channels.

In DIT mode for McASP, additional features of the transmitters are:

- Transmit-only mode 384 time slots (subframe) per frame.
- Biphase encoded LVCMOS output
- Channel status RAM (384 bits).
- User data RAM (384 bits).
- Separate valid bit (V) for subframe A, B.
- Stereo Support Only (Mono means send data 2x via software)

In DIT mode, the transmitter can support a 192 kHz frame rate (stereo) on up to all serial data pins simultaneously (note that the internal bit clock for DIT runs two times faster than the equivalent bit clock for I2S mode, due to the need to generate Biphase Mark Encoded Data).

NOTE: The McASP does NOT natively support DIR-mode reception (i.e. receiving in the S/PDIF format). To allow this, the McASP can use a DIR-input to I2S-output converter implemented by an external device (i.e. external DIR component). To facilitate reception in this case, the TDM mode of McASP receivers logic is extended to support a non-standard 384-slot TDM stream.

NOTE: An external transceiver must be connected to the McASP port in the device to translate the electrical signals delivered by the McASP (1.2 V or 1.8 V LVCMOS levels) to the electrical levels of the S/PDIF standard.

18.5.2.2.2 Definition of Terms

The serial bitstream transmitted or received by a McASP serializer is a long sequence of 1s and 0s on an audio transmit/receive pins AXRn. However, the sequence has a hierarchical organization that can be described in terms of frames of data, slots, words, and bits.

A basic synchronous serial interface consists of three important components: clock, frame sync, and data. Figure 18-80 shows two of the three basic components: the clock signal (ACLKX/ACLKR) and the data signals AXRn. Figure 18-80 does not specify whether the clock is for transmit (ACLKX) or receive (ACLKR) because the definitions of terms apply to both receive and transmit interfaces. In operation, each transmitter and receiver uses the signals ACLKX and ACLKR as serial clock, respectively. Optionally, a receiver can use ACLKX as the serial clock when a transmitter and receiver (not from the same serializer) of the McASP are configured to operate synchronously.

- Bit:

  A bit is the smallest entity in the serial data stream. The beginning and end of each bit is marked by an edge of the serial clock. The duration of a bit is a serial clock period. A '1' is represented by a logic high on AXRn pins for the entire duration of the bit. A 0 is represented by a logic low on an AXRn pin for the entire duration of the bit.

- Word:

  A word is a group of bits that make up the data being transferred between the McASP and the external

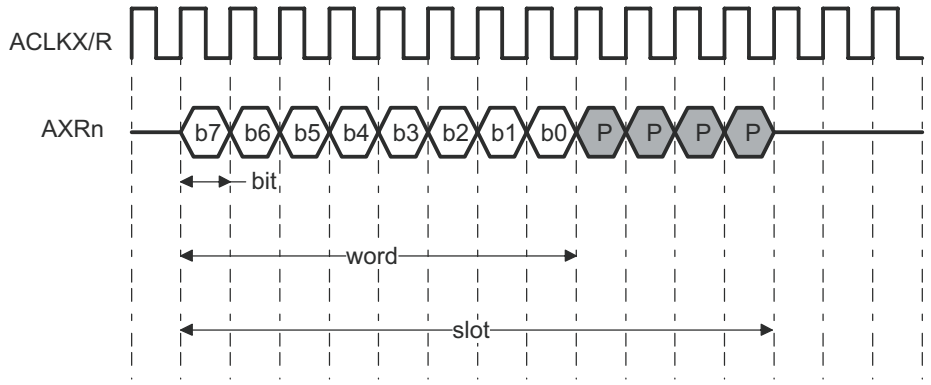
device. Figure 18-80 shows an 8-bit word.

- Slot:

  A slot consists of the bits that make up the word and can consist of additional bits used to pad the word to a convenient number of bits for the interface between the McASP and the external device. In Figure 18-80, the audio data consists of only 8 bits of useful data (8-bit word), but it is padded with four 0s (12-bit slot) to satisfy the desired protocol in interfacing to an external device. Within a slot, the bits can be shifted out of the McASP on an AXRn pin with either MSB or LSB first.

  When the word size is smaller than the slot size, the word can be aligned to the left of the slot (beginning) or to the right of the slot (end). The additional bits in the slot not belonging to the word can be padded with 0, 1, or with one of the bits (typically, the MSB or LSB) from the data word, i.e. left-aligned words within a slot are terminated with padding bits and right-aligned words within a slot are preceded by padding bits to fit in the slot size. Figure 18-81 shows these options.

Figure 18-80. Definition of Bit, Word, and Slot



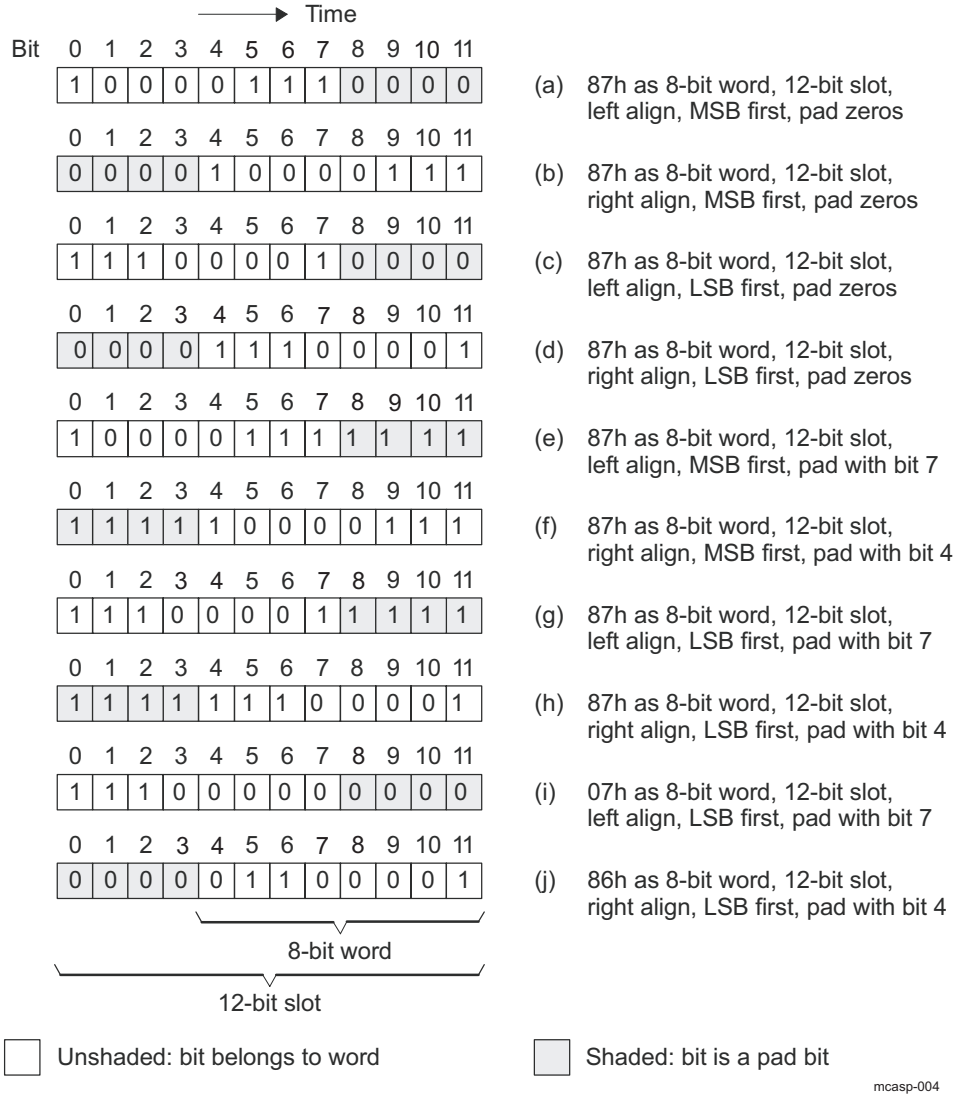
(1) b7:b0 - bits. Bits b7 to b0 form a word.

(2) P - pad bits. Bits b7 to b0, together with the 4 pad bits, form a slot.

(3) In this example, the data is transmitted MSB first, left-aligned.

mcasp-003

Figure 18-81. Bit Order and Word Alignment Within a Slot Examples



- Frame

  The third basic element of a synchronous serial interface is the frame synchronization signal, also referred to as frame sync in this chapter. A frame contains one or multiple slots, as determined by the desired protocol. Figure 18-82 shows an example frame of data and the frame definitions. In operation, the transmitter uses AFSX, and the receiver - AFSR signal. Figure 18-82 does NOT specify whether the frame sync (FS) is for transmit (AFSX) or receive (AFSR) because the definitions of terms apply to both receive and transmit interfaces. In operation, each transmitter/receiver uses AFSX/AFSR as a frame synchronization signal, respectively. Optionally, the receiver can use AFSX as the frame sync when the transmitter and receiver of the McASP are configured to operate synchronously. This example shows two slots in a frame (I2S format) and a frame-sync (FS) duration of a slot length.

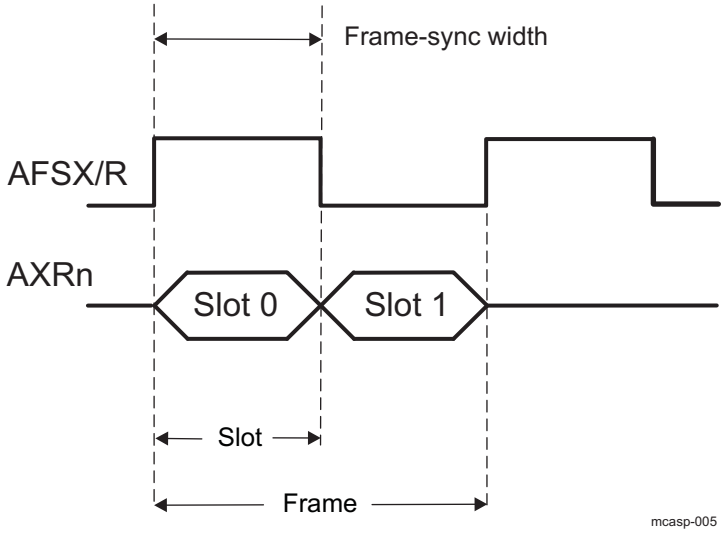
  This section shows only the generic definition of the frame sync. For more information about the frame-sync formats required for the transfer modes and protocols (TDM-mode and DIT-mode supported formats), see Section 18.5.2.2.3, *TDM Format* and Section 18.5.2.2.5, *S/PDIF-Coding Format*.

NOTE: All of the McASP serializers share the same, device pad accessible, clock and frame signals, as follows:

– AHCLKX, ACLKX, and AFSX for the transmitting section

– ACLKR and AFSR for the receiving section

Figure 18-82. Definition of Frame and Frame-Sync Width



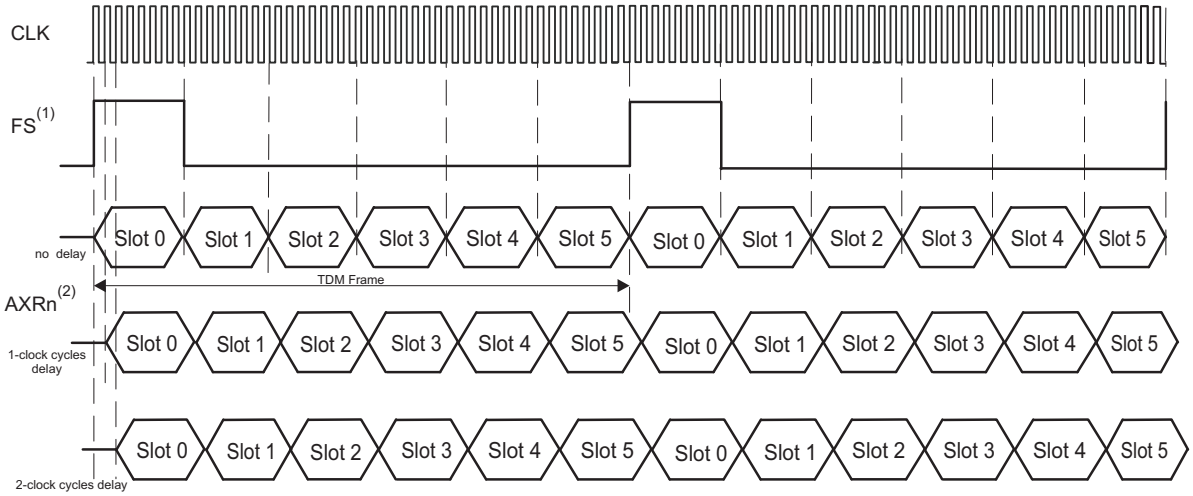
The following terms are used throughout this chapter:

- TDM: Time-division multiplexed. See Section 18.5.2.2.3 for details on the TDM protocol
- I2S: Inter-Integrated Sound protocol, commonly used on audio interfaces. The McASP supports the I2S protocol as part of the TDM mode (when configured as a 2-slot frame).
- DIT: Digital audio interface transmit. The McASP supports transmitting in S/PDIF format on each AXRn data pin.
- DIR: Digital audio interface receive. The McASP does NOT natively support receiving in S/PDIF format on AXRn data pins and requires an external DIR-to-TDM or DIR-to-I2S converter chip for a DIR-frame reception.
- Slot or time slot: For DIT/DIR format, a McASP time slot corresponds to a DIT/DIR subframe.

18.5.2.2.3 TDM Format

The TDM format is used to transfer data between the host CPU and one or more analog-to-digital converter (ADC), digital-to-analog converter (DAC), or S/PDIF receiver (DIR) devices. An example for a 6-slot (channel) TDM transmission on one McASP data pin - AXRn is illustrated on Figure 18-83.

Figure 18-83. TDM Format - 6 channel example



(1) - Frame sync duration of 1 slot - length is shown. A single bit - duration of FS is also supported

(2) - Slot 0 of AXRn stream is being offset with 0-, 1-, and 2- clock cycle delay from the frame sync, respectively.

mcasp-006

The TDM format uses three signals In a basic synchronous serial interface: data (AXRn), clock (CLK) and frame sync (FS). The data signal present on AXRn pin is fully synchronous to the serial clock (ACLKX or ACLKR). The data bits are grouped into words and slots (see also Section 18.5.2.2.2), the latter being also refered to as the "time-slots" or "channels" in TDM terminology. A frame consists of multiple time-slots. Each TDM frame is marked by the frame sync signal (AFSX or AFSR). The TDM transfer is continuous and periodic, with no delays between slots.

Within a certain frame, the last bit of slot N is followed immediately on the next serial clock with the first bit of the next slot N+1. On the boundary between two adjacent TDM-frames, the last bit of the last slot from the frame M, is followed immediately on the next clock cycle with the first bit of the first slot from the next frame M+1. For McASP, there is an option to offset the first bit of the first slot with a 0-,1- or 2-cycle delay from the frame sync signal.

The frame sync - AFSX/AFSR only marks the beginning of slot 0 and start of a new frame. Since it does not determine the boundaries of a slot, there is a requirement for a connected transmitter and receiver to agree on the number of transferred bits per slot.

In a typical audio system involving McASP module, a single TDM data frame is transferred during each sample period Ts of a data converter. The user has following choices to implement multiple channels:

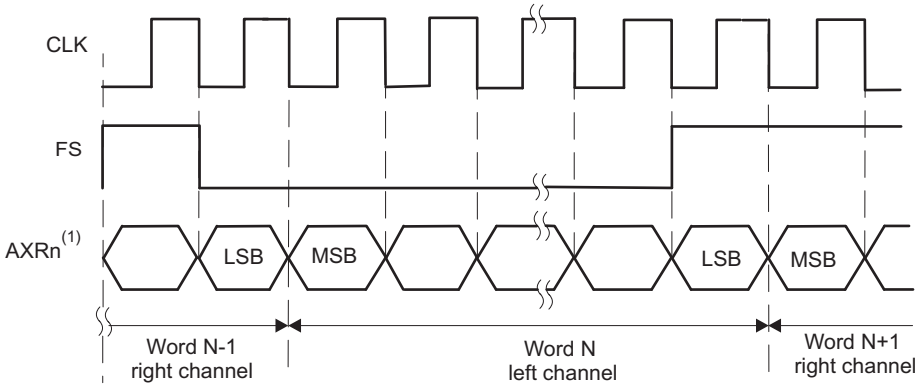
- Use more data slots (on a price of higher speed serial clock) per frame transmitted/received on just one of the available McASP data pins AXRn.
- Use less number of slots per TDM frame (requires a slower serial clock), making them available on several of the McASP pins AXRn.

18.5.2.2.4 I2S Format

The TDM transfer mode of the McASP supports the I2S format when frame is configured to have 2 slots. I2S format is specifically designed to transfer a stereo channel (left and right) over a single data pin AXRn. "Slots" are also commonly referred to as "channels". The frame width duration in the I2S format equals size of a slot. The frame signal is also referred to as "word select" in the I2S format.

The I2S protocol is illustrated on Figure 18-84.

Figure 18-84. I2S Format Overview



(1) - The example shows I2S data MSB-first transmission with 1-clock cycle delay between FS and data MSB

mcasp-036

18.5.2.2.5 S/PDIF Coding Format

The McASP transmitter supports the S/PDIF format with 3.3V biphase-mark encoded output. The S/PDIF format is supported by the DIT- transfer mode of the McASP. This section briefly discusses the S/PDIF coding format.

NOTE: The DIR- reception of S/PDIF format frames is NOT natively supported from the device McASP. For this purpose, an external DIR-to-TDM transfer mode adapter can be used between the remote device S/PDIF transmitter output and the McASP TDM-only compatible receiver input.

18.5.2.2.5.1 Biphase-Mark Code

In S/PDIF format, the digital signal is coded using the biphase-mark code (BMC). For each serializer transmitter n, the clock, frame, and data are embedded in only one signal - the data signal AXRn. In the BMC system, each data bit is encoded into two logical states (00, 01, 10, or 11) at the pin. These two logical states form a cell. The duration of the cell, which equals the duration of the data bit, is called a time interval. A logical 1 is represented by two transitions of the signal within a time interval, which corresponds to a cell with logical states 01 or 10. A logical 0 is represented by one transition within a time interval, which corresponds to a cell with logical states 00 or 11. In addition, the logical level at the start of a cell is inverted from the level at the end of the previous cell. Figure 18-85 and Table 18-323 show how data is encoded to the BMC format.

As shown in Figure 18-85, the clock frequency is twice the unencoded data bit rate. In addition, the clock is always programmed to 128 × f_s, where f_s is the sample rate (see Section 18.5.2.2.5.3, *Frame Format*, for details on how this clock rate is derived based on the S/PDIF format). The device receiving in S/PDIF format can recover the clock and frame information from the BMC signal.

Figure 18-85. Biphase-Mark Code

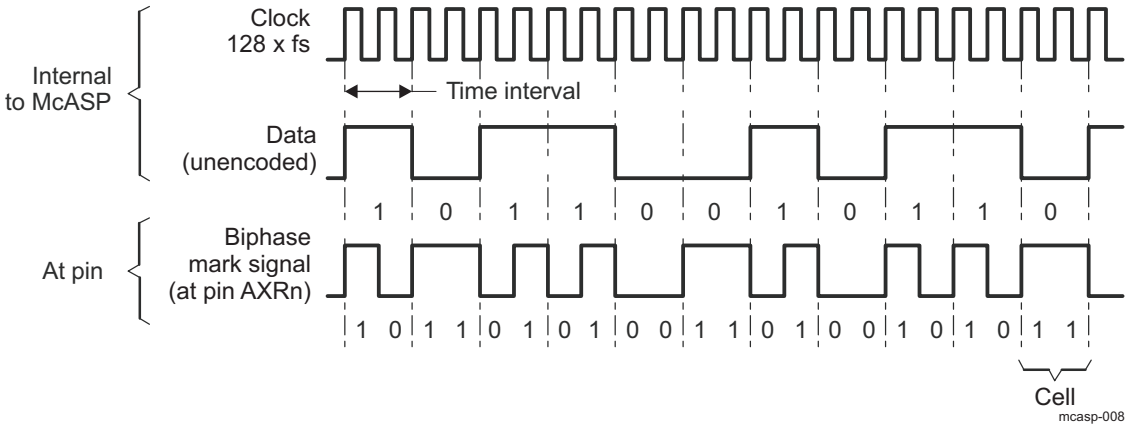


Table 18-323. Biphase-Mark Encoder

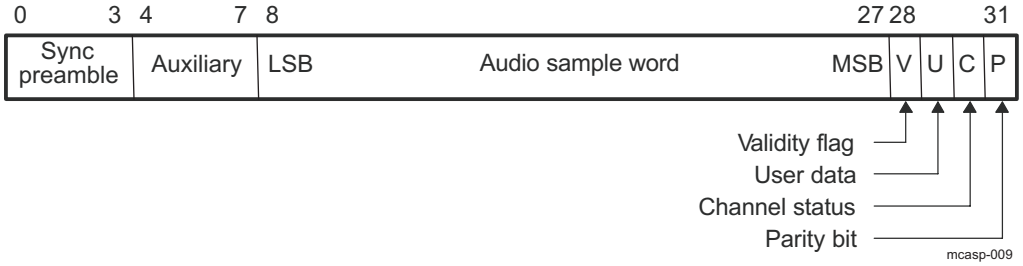
Data (Unencoded)	Previous State at Pin AXRn	BMC-Encoded Cell Output at Pin AXRn
0	0	11
0	1	00
1	0	10
1	1	01

18.5.2.2.5.2 S/PDIF Subframe Format

Every audio sample transmitted in a subframe consists of 32 S/PDIF time intervals (or cells), numbered 0 to 31. Figure 18-86 shows a subframe.

- Time intervals 0–3 carry one of the three permitted preambles to signify the type of audio sample in the current subframe. The preamble is not encoded in BMC format, and therefore the preamble code can contain more than two consecutive 0 or 1 logical states in a row. See Table 18-324.
- Time intervals 4–27 carry the audio sample word in linear 2s-complement representation. The MSB is carried by time interval 27. When a 24-bit coding range is used, the LSB is in time interval 4. When a 20-bit coding range is used, time intervals 8-27 carry the audio sample word with the LSB in time interval 8. Time intervals 4–7 may be used for other applications and are designated auxiliary sample bits.
- If the source provides fewer bits than the interface allows (20 or 24), the unused LSBs are set to logical 0. For a nonlinear PCM audio application or a data application, the main data field can carry any other information.
- Time interval 28 carries the validity bit (V) associated with the main data field in the subframe.
- Time interval 29 carries the user data channel (U) associated with the main data field in the subframe.
- Time interval 30 carries the channel status information (C) associated with the main data field in the subframe. The channel status indicates if the data in the subframe is digital audio or some other type of data.
- Time interval 31 carries a parity bit (P) such that time intervals 4–31 carry an even number of 1s and an even number of 0s (even parity). As listed in Table 18-324, the preambles (time intervals 0–3) are also defined with even parity.

Figure 18-86. S/PDIF Subframe Format



As listed in Table 18-324, the McASP DIT generates only one polarity of preambles, and it assumes the previous logical state is 0. This is because the McASP assures an even-polarity encoding scheme when transmitting in DIT mode. If an underrun condition occurs, the DIT resynchronizes to the correct logic level on the AXRn pin before continuing with the next transmission.

Table 18-324. Preamble Codes

Preamble Code[1]	Previous Logical State	Logical States on pin AXRn[2]	Description
B (or Z)	0	1110 1000	Start of a block and subframe 1
M (or X)	0	1110 0010	Subframe 1
W (or Y)	0	1110 0100	Subframe 2

(1) Historically, preamble codes are referred to as B, M, and W. For use in professional applications, preambles are referred to as Z, X, and Y, respectively.

(2) The preamble is not BMC-encoded. Each logical state is synchronized to the serial clock. These eight logical states make up time slots (cells) 0 to 3 in the S/PDIF stream.

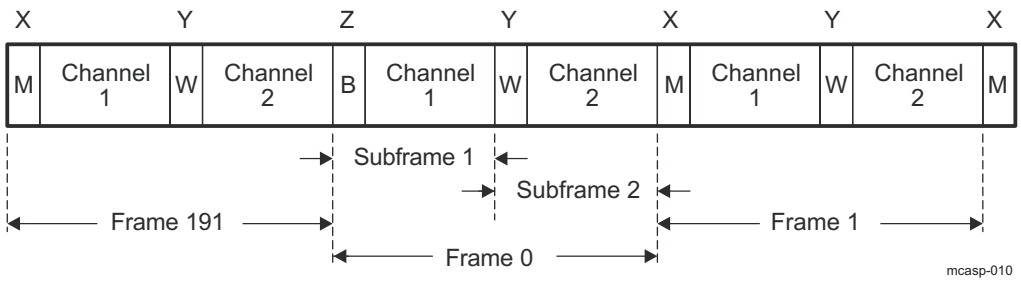
18.5.2.2.5.3 Frame Format

An S/PDIF frame is composed of two subframes (see Figure 18-87). For linear coded audio applications, the rate of frame transmission normally corresponds exactly to the source sampling frequency f_s. The S/PDIF format clock rate is therefore 128 × f_s (128 = 32 cells per subframe × 2 clocks per cell × 2 subframes per sample). For example, for an S/PDIF stream at a 192-kHz sampling frequency, the serial clock is 128 × 192 kHz = 24.58 MHZ.

In 2-channel operation mode, the samples taken from both channels are transmitted by time multiplexing in consecutive subframes. Both subframes contain valid data (cell 28 validity bits for A- and B- channels, both set to '0'). The first subframe (left or A channel in stereophonic operation and primary channel in monophonic operation) normally starts with preamble M. However, the preamble of the first subframe changes to preamble B once every 192 frames to identify the start of the block structure used to organize the channel status information. The second subframe (right or B channel in stereophonic operation and secondary channel in monophonic operation) always starts with preamble W.

In single-channel operation mode in a professional application, the frame format is the same as in the 2-channel mode. Data is carried in the first subframe and may be duplicated in the second subframe. If the second subframe is not carrying duplicate data, cell 28 (validity bit) is set to logical 1.

Figure 18-87. S/PDIF Frame Format



18.5.3 McASP Integration

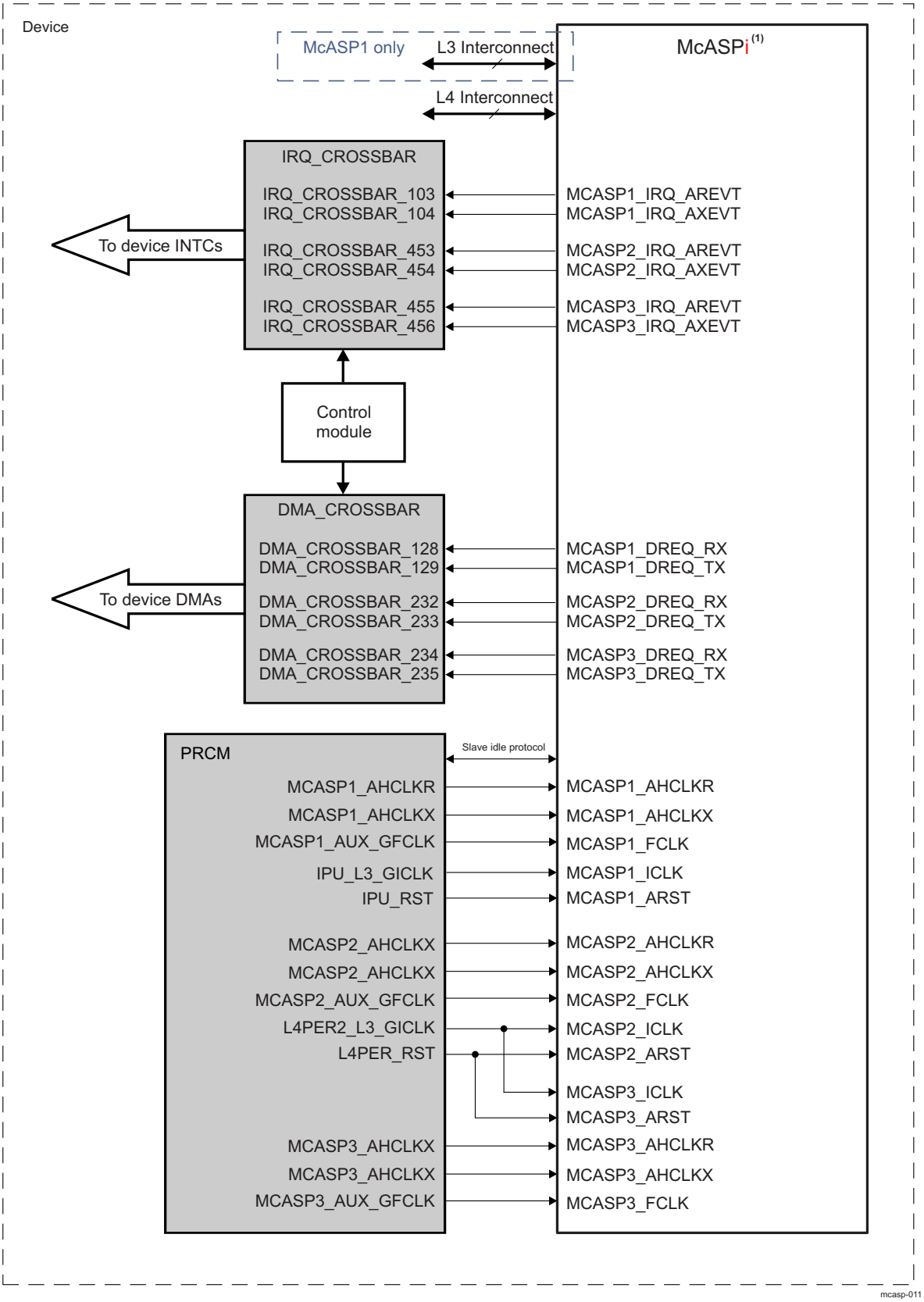
This section describes module integration in the device, including information about clocks, resets, and hardware requests.

McASP module includes the following features:

- Slave idle protocol
- One DMA request for transmit event
- One DMA request for receive event
- One interrupt request (IRQ) for transmit
- One interrupt request (IRQ) for receive

Figure 18-88 shows McASP integration.

Figure 18-88. McASP Integration



(1) i = 1 to 3

NOTE: For more information about the slave idle protocol, see Section 3.1.1.1.2, *Module Level Clock Management* of Chapter 3, *Power, Reset, and Clock Management*.

Table 18-325 through Table 18-327 summarize McASP integration in the device.

Table 18-325. McASP Integration Attributes

Module Instance	Attributes		
	Power Domain	Wake-Up Capability	Interconnect
McASP1	PD_COREAON	No	L3_MAIN
			L4_PER2
McASP2	PD_COREAON	No	L4_PER2
McASP3	PD_COREAON	No	L4_PER2

Table 18-326. McASP Clocks and Resets

Clocks				
Module Instance	Destination Signal Name	Source Signal Name	Source	Description
McASP1	MCASP1_AHCLKR	MCASP1_AHCLKR	PRCM	McASP1 AHCLKR receive high-frequency master clock
	MCASP1_AHCLKX	MCASP1_AHCLKX		McASP1 AHCLKX transmit high-frequency master clock
	MCASP1_FCLK	MCASP1_AUX_GFCLK		McASP1 functional clock
	MCASP1_ICLK	IPU_L3_GICLK		McASP1 interface clock
McASP2	MCASP2_AHCLKR	MCASP2_AHCLKX	PRCM	McASP2 AHCLKR receive high-frequency master clock
	MCASP2_AHCLKX	MCASP2_AHCLKX		McASP2 AHCLKX transmit high-frequency master clock
	MCASP2_FCLK	MCASP2_AUX_GFCLK		McASP2 functional clock
	MCASP2_ICLK	L4PER2_L3_GICLK		McASP2 interface clock
MCASP3	MCASP3_AHCLKR	MCASP3_AHCLKX	PRCM	McASP3 AHCLKR receive high-frequency master clock
	MCASP3_AHCLKX	MCASP3_AHCLKX		McASP3 AHCLKX transmit high-frequency master clock
	MCASP3_FCLK	MCASP3_AUX_GFCLK		McASP3 functional clock
	MCASP3_ICLK	L4PER2_L3_GICLK		McASP3 interface clock
Resets				
McASP1	MCASP1_ARST	IPU_RST	PRCM	McASP1 reset
McASP2	MCASP2_ARST	L4PER_RST	PRCM	McASP2 reset
McASP3	MCASP3_ARST	L4PER_RST	PRCM	McASP3 reset

Table 18-327. McASP Hardware Requests

Interrupt Requests				
Module Instance	Source Signal Name	Destination IRQ_CROSSBAR Input	Default Mapping	Description
McASP1	MCASP1_IRQ_AREVT	IRQ_CROSSBAR_103	-	McASP1 module receive interrupt request
	MCASP1_IRQ_AXEVT	IRQ_CROSSBAR_104	-	McASP1 module transmit interrupt request

Table 18-327. McASP Hardware Requests (continued)

McASP2	MCASP2_IRQ_AREVT	IRQ_CROSSBAR_453	-	McASP2 module receive interrupt request
	MCASP2_IRQ_AXEVT	IRQ_CROSSBAR_454	-	McASP2 module transmit interrupt request
McASP3	MCASP3_IRQ_AREVT	IRQ_CROSSBAR_455	-	McASP3 module receive interrupt request
	MCASP3_IRQ_AXEVT	IRQ_CROSSBAR_456	-	McASP3 module transmit interrupt request
DMA Requests				
Module Instance	**Source Signal Name**	**Destination DMA_CROSSBAR Input**	**Default Mapping**	**Description**
McASP1	MCASP_DREQ_RX	DMA_CROSSBAR_128	DMA_DSP1_DREQ_0 DMA_DSP2_DREQ_0	McASP module receive event request
	MCASP_DREQ_TX	DMA_CROSSBAR_129	DMA_DSP1_DREQ_1 DMA_DSP2_DREQ_1	McASP module transmit event request
McASP2	MCASP2_DREQ_RX	DMA_CROSSBAR_232	-	McASP2 module receive event request
	MCASP2_DREQ_TX	DMA_CROSSBAR_233	-	McASP2 module transmit event request
McASP3	MCASP3_DREQ_RX	DMA_CROSSBAR_234	-	McASP3 module receive event request
	MCASP3_DREQ_TX	DMA_CROSSBAR_235	-	McASP3 module transmit event request

NOTE: The **Default Mapping** column in Table 18-327 *McASP Hardware Requests* shows the default mapping of module IRQ and DREQ source signals. These module IRQ and DREQ source signals can also be mapped to other lines of each device Interrupt or DMA controller through the IRQ_CROSSBAR and DMA_CROSSBAR modules, respectively.
For more information about the IRQ_CROSSBAR module, see Section 13.4.6.4, *IRQ_CROSSBAR Module Functional Description*, in Chapter 13, *Control Module*.
For more information about the DMA_CROSSBAR module, see Section 13.4.6.5, *DMA_CROSSBAR Module Functional Description*, in Chapter 13, *Control Module*.
For more information about the device interrupt controllers, see Chapter 12, *Interrupt Controllers*.
For more information about the device DSP1_EDMA and DSP2_EDMA modules, see Section 4.3.7, *DSP Integrated EDMA Subsystem*.

NOTE:

- For the description of the interrupt source, see Section 18.5.4.12, *McASP Events and Interrupt Requests*.
- For the description of the DMA source, see Section 18.5.4.13, *DMA Requests*.

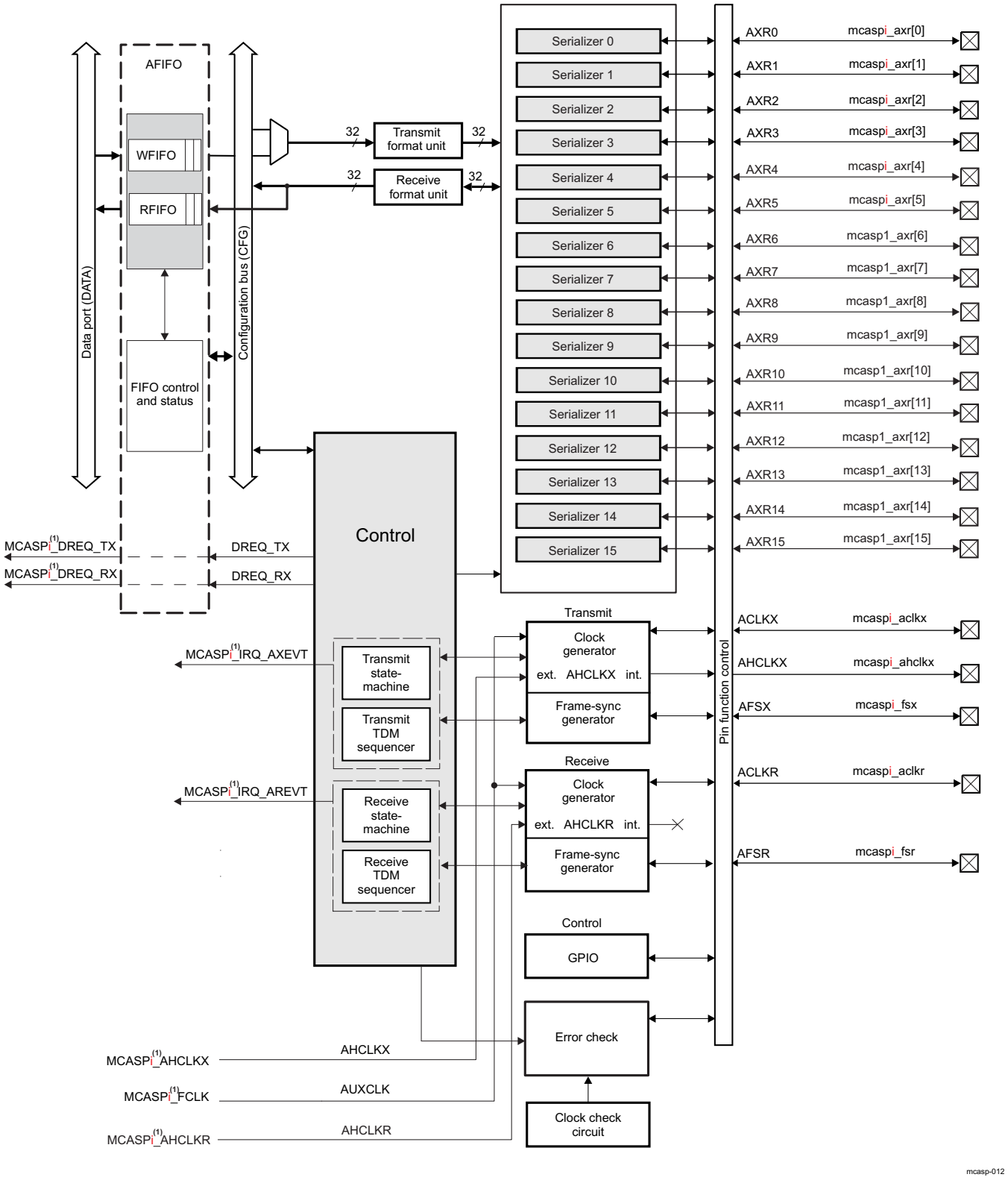
18.5.4 McASP Functional Description

In the text throughout this section a single instance of McASP is described assuming that all modules are functionally identical. For module availability and integration differences, see Section 18.5.2, *McASP Environment*, and Section 18.5.3, *McASP Integration*

18.5.4.1 McASP Block Diagram

Figure 18-89 shows the major blocks of the McASP module. McASP1 has 16 serializers , and McASP2 and McASP3 have 6 serializers each. The serializers share a clock and frame-sync generator, format unit, and error-checking logic independently for the receive and transmit part.

Figure 18-89. McASP Module Block Diagram



(1) i = 1 to 3

NOTE: The internal and external clocks mentioned in this section are with respect to clock and frame-sync generator modules.

18.5.4.2 McASP Clock and Frame-Sync Configurations

There are three scenarios to provide clock source signals for the Tx part and four scenarios for the Rx part of the McASP serializers. The first three scenarios are identical between the Tx and Rx part of the McASP. They feature an asynchronous operation between receiver and transmitter channels using independent Tx/Rx bit rate clock sources (either internal or external).

In the first scenario, the transmit - XCLK and receive - RCLK serial clocks (clock at the bit rate) are generated internally by passing through a couple of clock dividers off the internal functional clock source (AUXCLK). In this case, the bit rate clock is generated internally and is driven out on the pin ACLKX for the Tx part and pin ACLKR for the Rx part, respectively. An internally generated high-frequency clock can be optionally driven out onto the AHCLKX pin for the Tx part to serve as a reference clock for other components in the system.

In the second scenario, an external for the device clock, is passed on the ACLKX (for the TX part) and ACLKR (for the RX part) pins which are configured as inputs. In this case the Rx- /Tx- high-speed clock logic is bypassed for the XCLK/RCLK generation.

In the third (mixed) scenario, an externally driven (master) high-frequency clock is applied on the AHCLKX (for the TX part) pin, which is configured as input. In this case the AHCLKX clock frequency can be divided down via programming the ACLKX associated divider to produce the necessary bit rate clock. The high-speed clock divider can NOT be used.

In the fourth clock generation scenario the bit rate clock for McASP receivers - RCLK is derived from the bit rate clock of the McASP transmitters - XCLK for a synchronous operation between transmitters and receivers. Hence, the whole receiver clock generator logic is bypassed.

A typical role of the McASP frame sync signal is to carry the left/right clock (LRCLK) signal when transmitting and receiving stereo data.

For an asynchronous operation, the AFSX (Tx part) and AFSR (Rx part) frame synchronization signals can be sourced internally or delivered externally independently for the Tx and Rx channels. During synchronous operation the receive frame sync - AFSR signal is derived from the transmit frame sync - AFSX signal. A synchronous and asynchronous mode applies to bit rate clock and frame sync signals at the same time.

18.5.4.2.1 McASP Transmit Clock

The transmit high-speed and transmit clock configuration is controlled by the following registers:

- MCASP_ACLKXCTL
- MCASP_AHCLKXCTL

In case, the transmit bit clock, ACLKX, is generated internally, the MCASP_ACLKXCTL[5] CLKXM bit must be set to 1. Thus, the clock is divided down by a programmable bit clock divider (the MCASP_ACLKXCTL[4:0] CLKXDIV bit field) from the source signal.

If the transmit high-frequency master clock, AHCLKX, is also sourced internally (that is first scenario described in Section 18.5.4.2, the MCASP_AHCLKXCTL[15] HCLKXM bit must be set to 1. Thus, the clock is divided down by a programmable high-clock divider (the MCASP_AHCLKXCTL[11:0] HCLKXDIV bit field) from the McASP internal clock source AUXCLK.

Internally, the McASP always shifts transmit data at the rising edge of the internal transmit clock - XCLK, (see Figure 18-90). The CLKXP mux determines if ACLKX needs to be inverted to become XCLK. If MCASP_ACLKXCTL[7] CLKXP = 0, the CLKXP mux directly passes ACLKX signal to XCLK. As a result, the McASP shifts transmit data at the rising edge of ACLKX. If MCASP_ACLKXCTL[7] CLKXP = 1, the CLKX mux passes the inverted version of ACLKX to XCLK. As a result, the McASP shifts transmit data at the falling edge of ACLKX.

It can be seen in Figure 18-90 that XCLK is propagated to the Rx clock logic, to allow an internally synchronous operation between McASP transmitters and receivers. This is used for example in the McASP loopback mode.

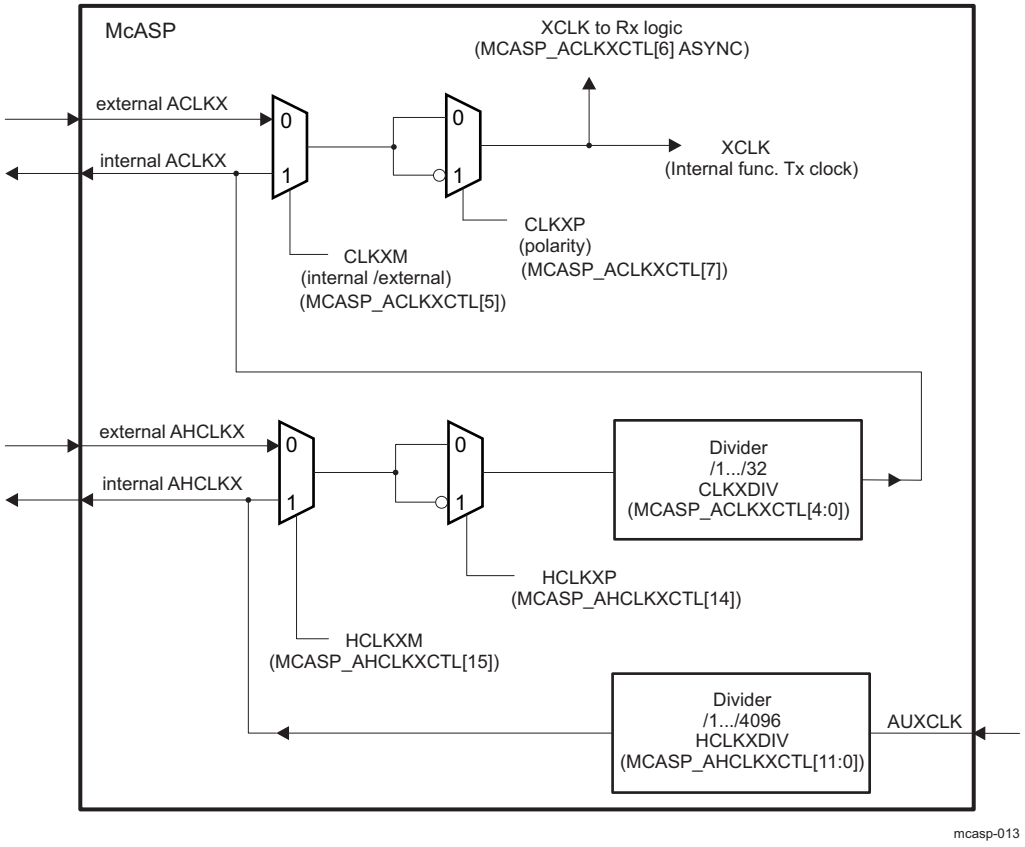
NOTE: The polarity of ACLKX can be controlled in MCASP_ACLKXCTL[7] CLKXP, regardless of ACLKX signal being internally or externally sourced.

In addition, there is an option to invert polarity of the AHCLKX master high speed clock via writing the MCASP_AHCLKXCTL[14] HCLKXP bit.

NOTE: In a similar way, the polarity of AHCLKX clock can be controlled in MCASP_AHCLKXCTL[14] HCLKXP, regardless of the AHCLKX signal being internally or externally sourced.

Figure 18-90 is the block diagram of the transmit clock generator.

Figure 18-90. Transmit Clock Generator Block Diagram



NOTE: In this device:

- ACLKX is mapped on the device ball mcaspi_aclkx, where i = 1 to 3
- internal AHCLKX is mapped on the device ball mcaspi_ahclkx
- external AHCLKX is mapped on MCASPi_AHCLKX clock from the PRCM

For more on McASP integration, see Section 18.5.2, *McASP Environment*, and Section 18.5.3, *McASP Integration*.

18.5.4.2.2 McASP Receive Clock

The McASP receive clock generator is built on a very similar to the transmit clock generator (but independent) circuit.

The receive clock configuration is controlled by the following registers:

- MCASP_ACLKRCTL
- MCASP_AHCLKRCTL

In case, the receive bit clock, ACLKR, is generated internally (but asynchronously to XCLK), the MCASP_ACLKRCTL[5] CLKRM bit must be set to 1. Thus, the clock is divided down by a programmable bit clock divider (the MCASP_ACLKRCTL[4:0] CLKRDIV bit field) from the source signal.

If the receive high-frequency master clock, AHCLKR, is also sourced internally (that is, first scenario described in Section 18.5.4.2) and the MCASP_AHCLKRCTL[15] HCLKRM bit must be set to 1. Thus, the clock is divided down by a programmable high-clock divider (the MCASP_AHCLKRCTL[11:0] HCLKRDIV bit field) from the McASP internal clock source AUXCLK.

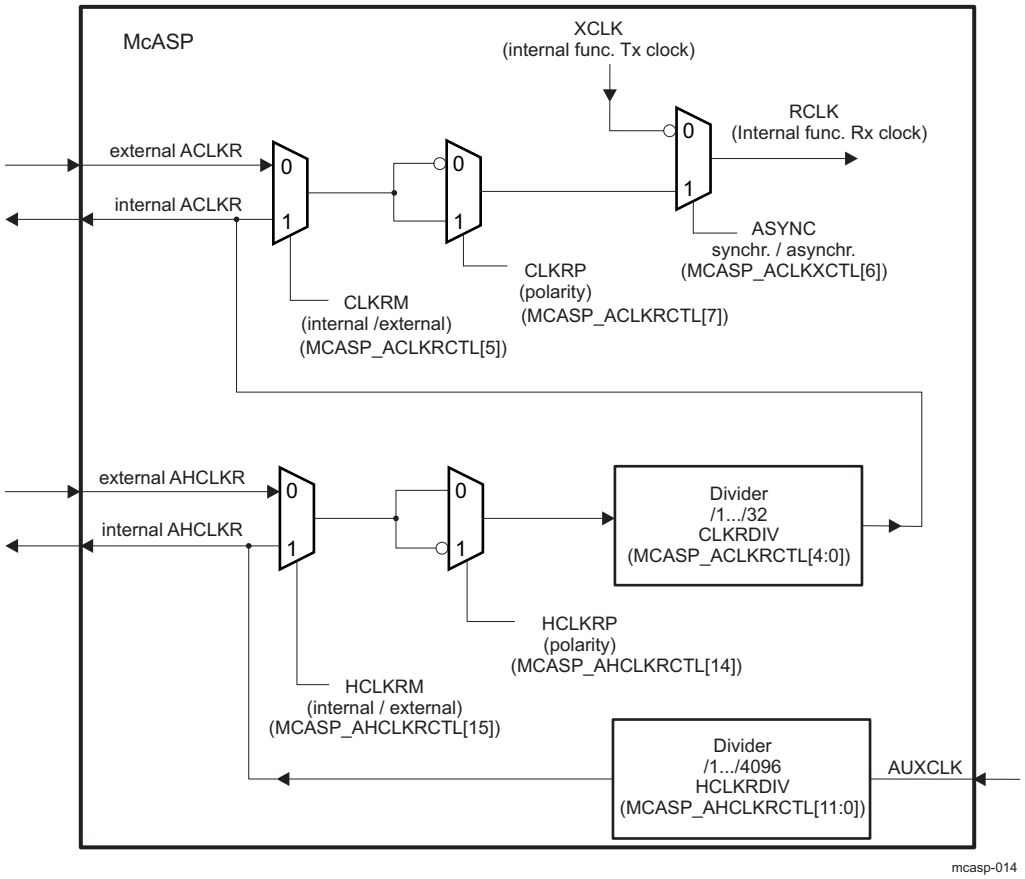
NOTE: The polarity of ACLKR can be controlled in MCASP_ACLKRCTL[7] CLKRP, regardless of ACLKR signal being internally or externally sourced.

In a similar way, the polarity of AHCLKR clock can be controlled in MCASP_AHCLKRCTL[14] HCLKRP, regardless of the AHCLKR signal being internally or externally sourced.

There is an option for the McASP receiver to be configured to operate synchronously to the ACLKX and AFSX signals. The XCLK output of the Tx Clock generator (see Figure 18-90 and Figure 18-91) becomes source of the receive clock (RCLK output), when the MCASP_ACLKXCTL[6] ASYNC bit in the transmit clock control register is set to '0b0'. For more information, refer to Section 18.5.4.2.4.

Figure 18-91 is the block diagram of the receive clock generator.

Figure 18-91. Receive Clock Generator Block Diagram



NOTE: In this device:

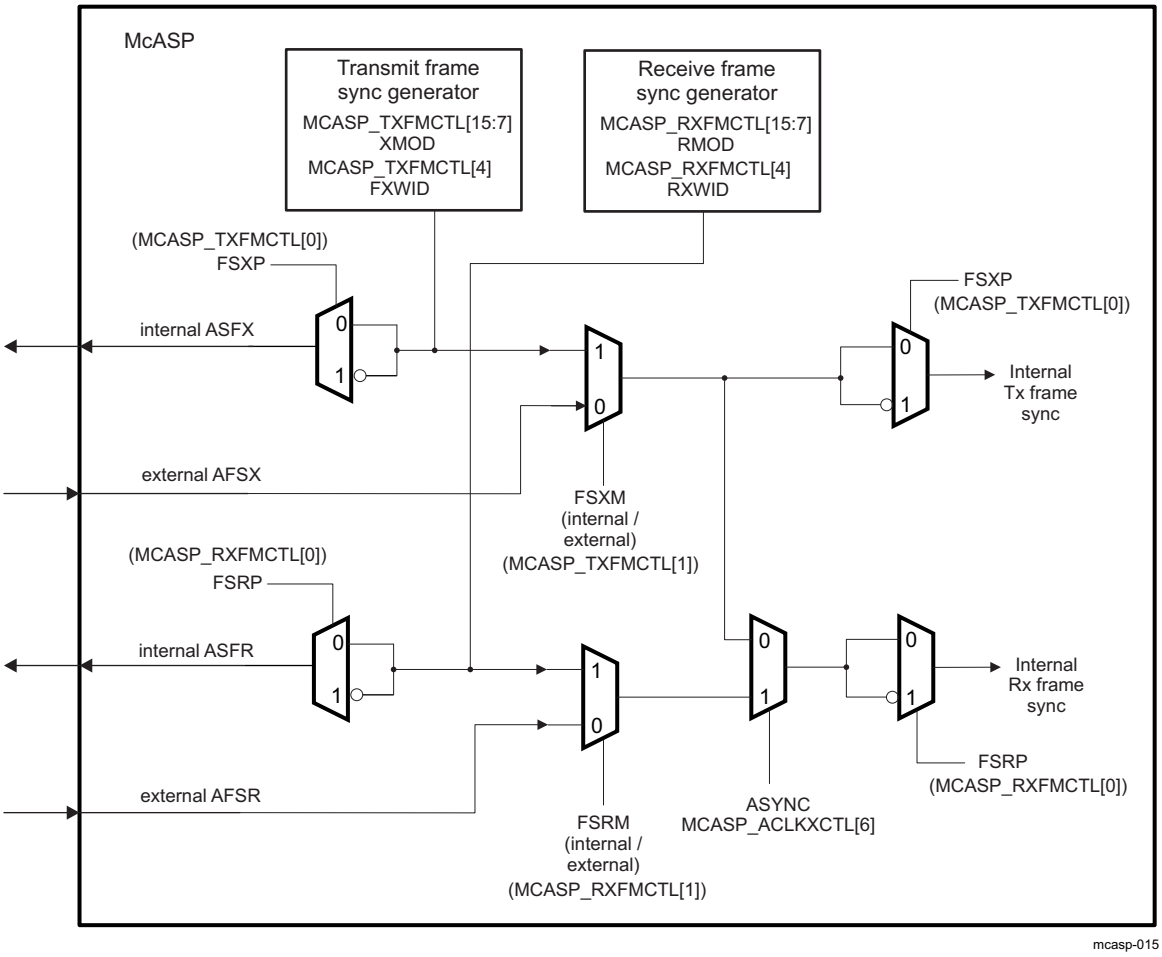
- ACLKR is mapped on the device ball mcaspi_aclkr, where i = 1 to 3
- internal AHCLKR is tied-off
- external AHCLKR is mapped on PRCM MCASPi_AHCLKR from PRCM

For more on McASP integration, see Section 18.5.2, *McASP Environment*, and Section 18.5.3, *McASP Integration*.

18.5.4.2.3 Frame-Sync Generator

There are two different modes for frame sync: burst and TDM. The McASP frame sync generator logic is illustrated in Figure 18-92. I/O buffers are not part of the McASP module, and are not shown in the figure.

Figure 18-92. Frame Sync Generator Block Diagram



NOTE: For more on McASP integration, see Section 18.5.2, *McASP Environment*, and Section 18.5.3, *McASP Integration*.

For the transmit logic, following frame-sync generator configurations can be selected:

- Internally/externally generated frame-sync via configuring bit MCASP_TXFMCTL[1] FSXM
- Frame-sync polarity: Rising edge or falling edge via configuring bit MCASP_TXFMCTL[0] FSXP
- Frame-sync width: "single bit" or "single word" via configuring bit MCASP_TXFMCTL[4] FXWID
- Frame sync mode - the appropriate frame sync generation pattern for the selected transfer mode is defined in the bitfield MCASP_TXFMCTL[15:7] XMOD, as follows:
  - For DIT mode (384 slots) - MCASP_TXFMCTL[15:7] XMOD = 0x180
  - For I2S mode (2 TDM slots) - MCASP_TXFMCTL[15:7] XMOD = 0x2
  - For TDM mode (from 3 to 32 TDM slots) - MCASP_TXFMCTL[15:7] XMOD set in range 0x3 - 0x20
- Bit delay: 0, 1, or 2 cycles before the first data bit. This delay is defined in MCASP_TXFMT[17:16] XDATDLY

For the receive logic, following frame-sync generator configurations can be selected:

- Internally/externally generated frame-sync via configuring bit MCASP_RXFMCTL[1] FSRM
- Frame-sync polarity: Rising edge or falling edge via configuring bit MCASP_RXFMCTL[0] FSRP
- Frame-sync width: "single bit" or "single word" via configuring bit MCASP_RXFMCTL[4] FRWID
- Frame sync mode - the appropriate frame sync generation pattern for the selected transfer mode is

defined in the bitfield MCASP_RXFMCTL[15:7] RMOD, as follows:

- For I2S mode (2 TDM slots) - MCASP_RXFMCTL[15:7] RMOD = 0x2
- For TDM mode (from 3 to 32 TDM slots) - MCASP_RXFMCTL[15:7] RMOD set in range 0x3 - 0x20
- For the special 384-slot TDM mode - MCASP_RXFMCTL[15:7] RMOD=0x180

- Bit delay: 0, 1, or 2 cycles before the first data bit. This delay is defined in MCASP_RXFMT[17:16] RDATDLY
- Selecting the source (AFSX or AFSR) of receiver internal frame synchronization. This is done in the same bit - MCASP_ACLKXCTL[6] ASYNC, used to define the receiver internal clock source. For more details, refer to Section 18.5.4.2.4.

Regardless of the AFSX/AFSR being internally generated or externally sourced, the polarity of AFSX/AFSR is determined by FSXP/FSRP, respectively, to be either rising or falling edge. If FSXP/FSRP = 0, the frame sync polarity is rising edge. If FSXP/FSRP = 1, the frame sync polarity is falling edge.

NOTE: Certain restrictions apply to the receive and transmit logic settings, when MCASP_ACLKXCTL[6] ASYNC is set to 0b0. They are described in Section 18.5.4.2.4.

18.5.4.2.4 Synchronous and Asynchronous Transmit and Receive Operations

Synchronous Transmit and Receive Operations -

When MCASP_ACLKXCTL[6] ASYNC is written to 0b0, the transmit and receive sections operate synchronously to the transmit section clock and transmit frame sync signals.

Though Rx section may have a different data format, it has to be configured to have the same slot size than the transmit section one. As shown on the Figure 18-91, with the ASYNC bit set to 0b0, the RCLK becomes an inverted version of the transmit clock generator XCLK output.

When MCASP_ACLKXCTL[6] ASYNC = 0b0, both Rx and Tx sections use the same clock and frame sync signals. For this reason, they must be aligned on the following settings:

- MCASP_TXDITCTL[0] DITEN = 0 (that is, transmission in TDM mode is enabled)
- The total number of bits per frame must be the same (that is, RSSZ * RMOD product value must equal XSSZ * XMOD product value)
- The settings in MCASP_ACLKRCTL are NOT considered
- FSXM must match FSRM
- FXWID must match FRWID

For all other settings, the transmit and receive sections may be programmed independently.

Asynchronous Transmit and Receive Operations -

When MCASP_ACLKXCTL[6] ASYNC = 0b1, Tx and Rx operate independently from each other with separate clock and frame sync signals.

NOTE: Synchronous transmit and receive operations are allowed only in the McASP TDM (I2S) mode (i.e. when MCASP_TXDITCTL[0] DITEN=0b0).

18.5.4.3 Serializers

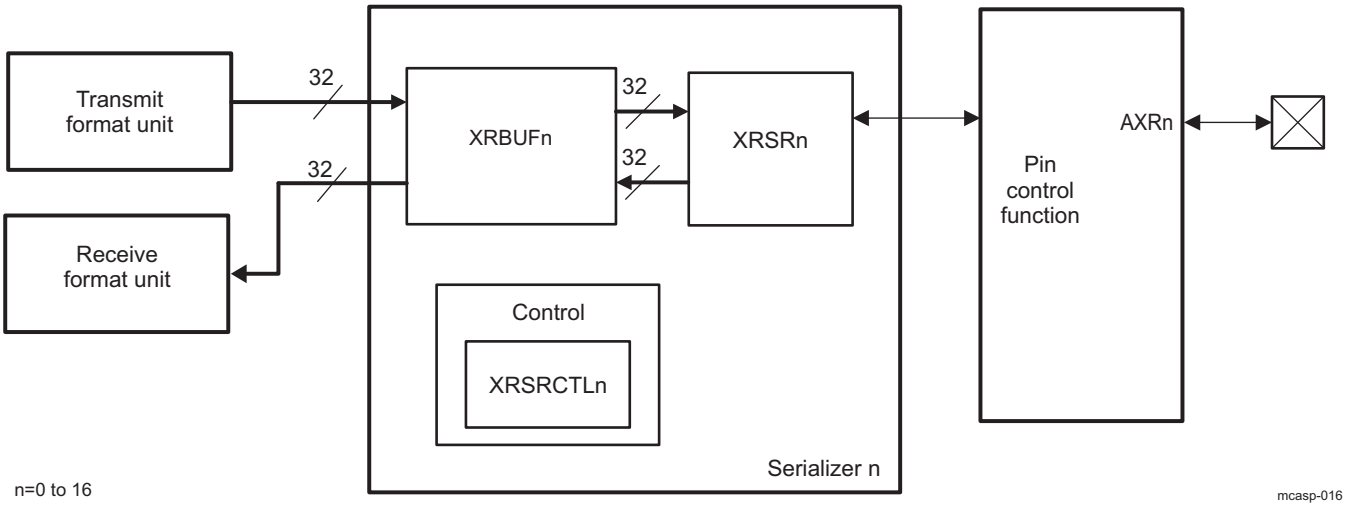
The McASP serializers shift serial data in (Rx) and out (Tx) of the McASP. A given serializer n consists of a shift register (XRSRn) with a single-entry data buffer XRBUFn used either for transmitting (write accessible in register MCASP_TXBUFn) or for receiving (read accessible in register MCASP_RXBUFn) data. In addition, each serializer has a dedicated control register (MCASP_XRSRCTLn) and a serial bidirectional data pin - AXRn. The register MCASP_XRSRCTLn allows n-th serializer to be configured as a transmitter, receiver, or as inactive. There are transmit and receive data formatting units to support data alignment options of the McASP which are shared between all Tx and Rx serializers, respectively.

A given serializer XRSRn shifter configured as a receiver in MCASP_XRSRCTLn, shifts in data through McASP corresponding device level bidirectional data pad AXRn. A given serializer XRSRn shifter configured as a transmitter in MCASP_XRSRCTLn, shifts out data on McASP corresponding device level bidirectional data pad AXRn.

The serializer is clocked from the transmit section clock (ACLKX signal) if configured to transmit or clocked from the receive section clock (ACLKR signal) if configured to receive. A serializer configured to transmit and receive operates in lockstep, which means that for McASP there are at most a couple of zones, one for transmit and one for receive.

Figure 18-93 is the serializer block diagram.

Figure 18-93. Individual Serializer and Connections Within McASP



Transmission on the n-th serializer is performed as follows:

The McASP is serviced by writing data into the register MCASP_TXBUFn, which is an alias of the serializer data buffer - XRBUFn for transmit function. The data automatically passes through the transmit format unit before reaching the XRBUFn register in the serializer. The data is then copied from the XRBUFn to XRSRn and shifted out from AXRn synchronously to the serial clock.

Reception from the n-th serializer is performed as follows:

The data is shifted into the McASP XRSRn serializer register through the AXRn pin, bit by bit. Once the entire slot becomes available within the XRSRn shift register, the data is copied into the serializer data buffer - XRBUFn,and can be accessed in the MCASP_RXBUFn register, which is an alias of the serializer data buffer - XRBUFn for receive function. When software reads the data from this register, the McASP passes the data through the receive format unit, hence it returns the formatted data.

Serializer controls:

A serializer n is configured as inactive via setting bitfield MCASP_XRSRCTLn[1:0] SRMOD to 0x0.

For a transmitting serializer, the MCASP_XRSRCTLn[3:2] DISMOD bitfield, defines the AXRn pin output state, during inactive slots (HIGH, LOW or Hi-Z).

Transmit function for the n-th serializer is selected via setting bitfield MCASP_XRSRCTLn[1:0] SRMOD to 0x1.

Receive function for the n-th serializer is selected via setting bitfield MCASP_XRSRCTLn[1:0] SRMOD to 0x2.

In the DIT-transmission mode (that is S/PDIF format data transmission): in addition to the data, the serializer shifts out other DIT-specific information accordingly (preamble, user data, etc.). For more information, see Section 18.5.2.2.5

18.5.4.4 Format Units

The McASP has one transmit data formatting unit and one receive data formatting unit, shared between the device McASP serializers. These units automatically remap the data bits within the transmitted or received words between a natural format for the device processors (for example, a Q31 representation) and the required format for the external serial device (for example I2S format). During the remapping process, the format unit can also mask off certain bits.

Since all transmitters share the same data formatting unit, the McASP only supports one transmit format at a time. For example, the McASP does NOT transmit in "I2S format" on serializer 0, while transmitting "Left Justified" on serializer 1. Likewise, the receiver section of the McASP only supports one data format at a time, and this format applies to all receiving serializers.

NOTE: The McASP can transmit in one format while receiving in a completely different format.

The bit mask and pad stage of each of Tx and Rx format units includes a full 32-bit mask register, allowing selected individual bits to either pass through the stage unchanged, or be masked off. The bit mask and pad then pad the value of the masked off bits by inserting either a 0, a 1, or one of the original 32 bits as the pad value. The last option allows for sign-extension when the sign bit is selected to pad the remaining bits. The rotate right stage performs bitwise rotation by a multiple of 4 bits (between 0 and 28 bits), programmable by the MCASP_RXFMT/MCASP_TXFMT register. Note that this is a rotation process, not a shifting process, so bit 0 gets shifted back into bit 31 during the rotation. The bit order - reversal stage either passes all 32 bits directly through, or swaps them. This allows for either MSB or LSB first data formats. If bit order reversal is not enabled, then the McASP will naturally transmit and receive in an LSB first order. Finally, note that the RDATDLY/XDATDLY bits in the MCASP_RXFMT/MCASP_TXFMT also determine the data format. For example, the difference between I2S format and left-justified is determined by the delay between the frame sync edge and the first data bit of a given time slot. For I2S format, RDATDLY/XDATDLY should be set to a 1-bit delay, whereas for left-justified format, it should be set to a 0-bit delay. The combination of all the options in MCASP_RXFMT/MCASP_TXFMT means that the McASP supports a wide variety of data formats, both on the serial data lines, and in the device CPU data representation.

18.5.4.4.1 Transmit Format Unit

The McASP transmit formatting unit consists of three stages :

- Bit mask (masks off bits)
- Rotate right (aligns data within word)
- Bit reversal (selects between MSB-first or LSB-first)

Figure 18-94 shows the transmit formatting unit.

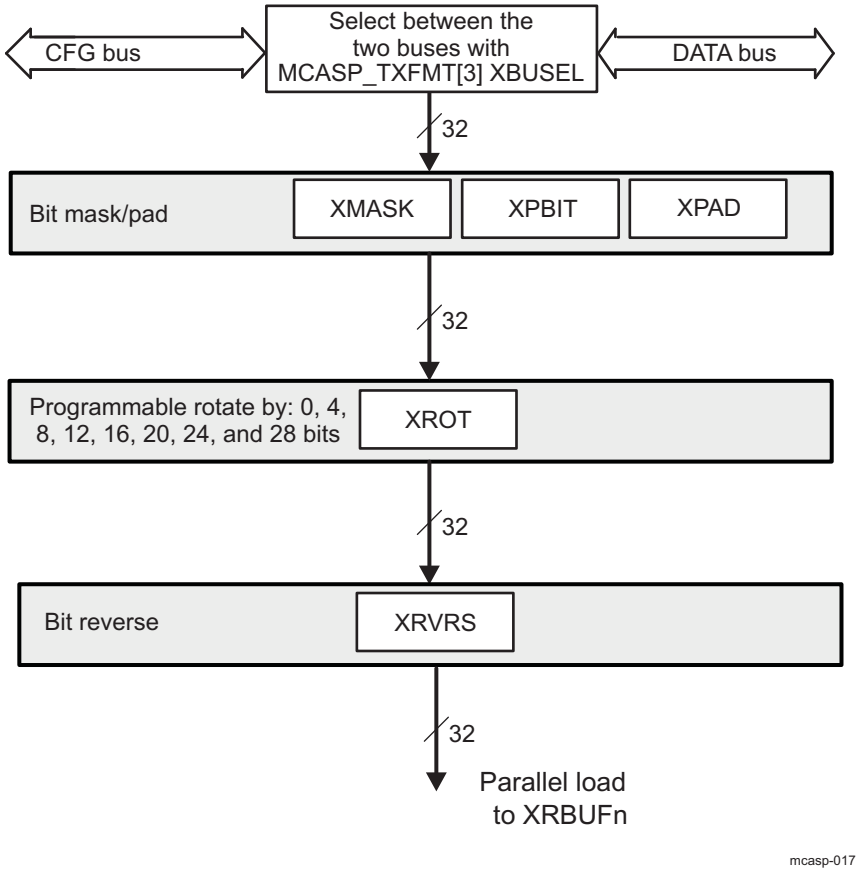
The McASP transmitter supports serial formats of:

- Slot (or Time slot) size = 8, 12, 16, 20, 24, 28, 32 bits
- Word size ≤ Slot size
- Alignment: when more bits/slot than bits/words, then:
  - Left aligned = word shifted first, remaining bits are pad
  - Right aligned = pad bits are shifted first, word occupies the last bits in slot
- Order of bits shifted out:
  - MSB: most-significant bit of word is shifted out first, last bit is LSB
  - LSB: least-significant bit of word is shifted out last, last bit is MSB

Hardware support for these serial formats comes from the programmable options in the bitstream format register - MCASP_TXFMT:

- XRVRS: bit reverse (1) or no bit reverse (0)
- XROT: rotate right by 0, 4, 8, 12, 16, 20, 24, or 28 bits
- XSSZ: transmit slot size of 8, 12, 16, 20, 24, 28, or 32 bits

Figure 18-94. Transmit Format Unit



As shown in Figure 18-94, the data to the transmit format unit can come from the configuration port (CFG) or the data port (DATA). The selection is made through the MCASP_TXFMT[3] XBUSEL bit. According to port type selected, data transfer has different behaviour. For more details, refer to the Section 18.5.4.10.1.3, *Transfers Through the DATA Port*, and Section 18.5.4.10.1.4, *Transfers Through the Configuration (CFG) Bus*.

In the transmit format unit (TFU), the input data bits are first masked-off with the MCASP_TXMASK[31:0] XMASK contents. The masked data is then right-rotated to MCASP_TXFMT[2:0] XROT positions, to produce the output word for a TDM- or DIT- transmission.

The bit mask stage includes a full 32-bit mask register, allowing selected individual bits to pass through the stage unchanged or be masked off.

18.5.4.4.1.1 TDM Mode Transmission Data Alignment Settings

The TDM-mode transmission settings are relevant for I2S-protocol and protocols using more than 2 TDM-slots.

XSSZ should always be programmed to match the slot size of the serial stream.

NOTE: Note that, TDM word size is not directly programmed into the McASP, but rather is used to determine the rotation needed in the XROT field.

The Table 18-328 show the XRVRS and XROT fields for each serial format and for both integer and Q31 fractional internal representations.

The Table 18-328 assumes that all slot size (SLOT in Table 18-328) and word size (WORD in Table 18-328) options are multiples of 4, since the transmit rotate right unit only supports rotation by multiples of 4. However, the bit mask/pad unit does allow for any number of significant digits. For example, a Q31 number may have 19 significant digits (word) and be transmitted in a 24-bit slot; this would be formatted as a word size of 20 bits and a slot size of 24 bits. However, it is possible to set the bit mask unit to only pass the 19 most-significant digits (program the mask value to FFFF E000h). The digits that are not significant can be set to a selected pad value, which can be any one of the significant digits, a fixed value of 0, or a fixed value of 1.

The transmit bit mask/pad unit operates on data as an initial step of the transmit format unit, and the data is aligned in the same representation as it is written to the transmitter (typically Q31 or integer).

Table 18-328. McASP TFU TDM Mode Settings

Bit Stream Order	Bit Stream Alignment	Internal Numeric Representation	MCASP_TXFMT bits	
			XROT[1]	XRVRS
MSB first[2]	Left aligned	Q31 fraction	0	1
MSB first	Right aligned	Q31 fraction	SLOT - WORD	1
LSB first	Left aligned	Q31 fraction	32 - WORD	0
LSB first	Right aligned	Q31 fraction	32 - SLOT	0
MSB first[2]	Left aligned	Integer	WORD	1
MSB first	Right aligned	Integer	SLOT	1
LSB first	Left aligned	Integer	0	0
LSB first	Right aligned	Integer	(32 - (SLOT - WORD)) % 32	0

(1) WORD = Word size rounded up to the nearest multiple of 4; SLOT = slot size; % = modulo operator
(2) To transmit in I2S format, select MSB first, left aligned, and also select XDATDLY = 01 (1 bit delay)

18.5.4.4.1.2 DIT Mode Transmission Data Alignment Settings

In case of a DIT-mode (S/PDIF protocol) transmission, while left-aligned Q31 data should be right-rotated to a multiple by 4 positions, no right-rotation is required for a right-aligned Q31 data. Because this is a rotation process, not a shifting process, bit 0 gets shifted back into bit 31 during the process.

The MCASP_TXFMT[17:16] XDATDLY bit field must be set to a 0-bit delay (0x0 value).

For left-aligned Q31 data, the following transmit format unit settings process the data into right-aligned data, ready for transmission:

- MCASP_TXFMT[2:0] XROT =
  - 0x2 (rotate right by 8 bits) - for a 24-bit output audio data
  - 0x3 (rotate right by 12 bits) - for a 20-bit output audio data
  - 0x4 (rotate right by 16 bits) - for a 16-bit output audio data
- MCASP_TXFMT[15] XRVRS = 0x0 – Bit reversal is not enabled; the McASP naturally transmits and receives in a LSB-first order.
- MCASP_TXMASK[32] XMASK = 0xFFFFFF00 – 0xFFFF0000
- MCASP_TXFMT[14:13] XPAD = 0x0 (Pad extra bits with 0s.)

For right-aligned data, the following transmit format unit settings process the data into right-aligned audio data ready for transmission:

- MCASP_TXFMT[2:0] XROT = 0x0 (rotate right by 0 bits regardless of the audio word length)
- MCASP_TXFMT[15] XRVRS = 0x0 – Bit reversal is not enabled; the McASP naturally transmits and receives in a LSB-first order.
- MCASP_TXMASK[32] XMASK = 0x00FFFFFF – 0x0000FFFF
- MCASP_TXFMT[14:13] XPAD = 0x0 (Pad extra bits with 0s.)

The example settings provided in Table 18-329 should be applied to McASP in cases of DIT-transmitting a Q31 data as a 24-bit, 20-bit and 16-bit left- or right- aligned audio word, respectively. Note that the listed settings let the McASP TFU preserve the most significant bits and cut only the LSBs of the original Q31 CPU data:

Table 18-329. McASP TFU DIT-Mode Example Settings

Output Audio Word Alignment	Audio Word Length	Right-rotation (multiple of 4-bit positions)	XMASK	XROT
LEFT	16	16	0xFFFF0000	0x4
LEFT	20	12	0xFFFFF000	0x3
LEFT	24	8	0xFFFFFF00	0x2
RIGHT	16	0	0x0000FFFF	0x0
RIGHT	20	0	0x000FFFFF	0x0
RIGHT	24	0	0x00FFFFFF	0x0

Assuming that a Q31 data word 0xFA5AFxxx (where x-marked nibbles of the data are applied as padding bits of the word) is generated on the McASP CFG (peripheral) port. To transmit a left-aligned 20-bit version of same word, preserving the MSBs, according to the Table 18-329, the user must set XMASK=0xFFFFF000, and to select a right-rotation to 12 positions (XROT=0x3).

- After applying 0-s (XPAD=0) as masking-off bits at the first TFU stage, word is transformed to the word 0xFA5AF000.
- After a rotation by 12 positions to the right is performed in TFU, the 20-bit output word obtained is: 0x000FA5AF.Thus the word gets ready for transmission being mapped with its LS-bit as bit 8 and its MS-bit as bit 27 within a S/PDIF bitstream. This word is shifted in a LSB-to-MSB order (XRVRS = 0x0) out of the XRSR register during a DIT-transmission.

Assuming that a right-aligned Q31 data word - 0x yyyyE4B4 is generated by software on the McASP CFG (peripheral) port (with the presumption that y-marked nibbles of the input data are applied as padding bits). To transmit a right-aligned 16-bit version of same word, preserving the MSBs, according to the table McASP TFU Example Settings, user is supposed to set XMASK=0x0000FFFF, and to select right-rotation to 0 positions (XROT=0x0).

- After masking-off with 0s at first TFU stage, word is transformed to 0x0000E4B4.
- Since no rotation is applied, the 16-bit output word obtained is actually the one obtained in the masking stage – 0x0000E4B4.

The above examples use internal representation in integer and Q31 notation, but other fractional notations are also possible.

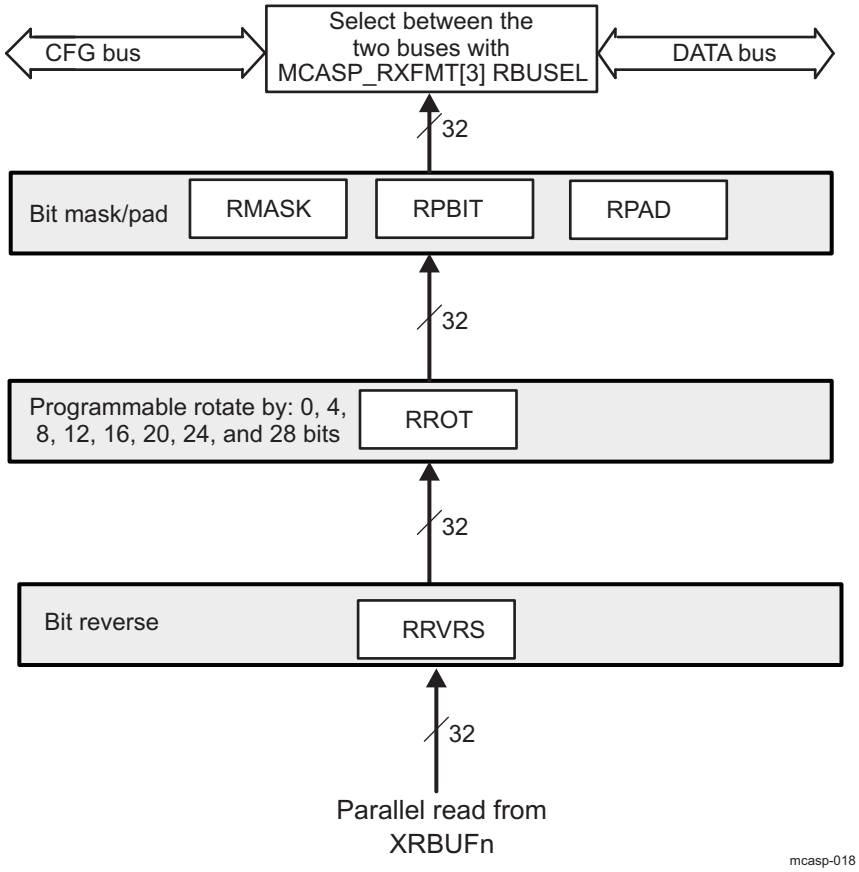
18.5.4.4.2 Receive Format Unit

The McASP receive formatting unit consists of three stages:

- Bit mask (masks off bits)
- Rotate right (aligns data within word)
- Bit reversal (selects between MSB first or LSB first)

Figure 18-95 shows the receive format unit (RFU).

Figure 18-95. Receive Format Unit



The McASP receiver supports serial formats of:

- Slot or time slot size = 8, 12, 16, 20, 24, 28, 32 bits
- Word size ≤ Slot size
- Alignment when more bits are available per slot than bits per word within the slot, then:
  - Left aligned = word shifted first, remaining bits are pad
  - Right aligned = pad bits are shifted first, word occupies the last bits in slot
- Order of bits shifted out:
  - MSB: most-significant bit of word is shifted out first, last bit is LSB
  - LSB: least-significant bit of word is shifted out last, last bit is MSB

Hardware support for these serial formats comes from the programmable options in the receive bitstream format register - MCASP_RXFMT:

- RRVRS: bit reverse (1) or no bit reverse (0)
- RROT: rotate right by 0, 4, 8, 12, 16, 20, 24, or 28 bits
- RSSZ: receive slot size of 8, 12, 16, 20, 24, 28, or 32 bits

As shown on Figure 18-95, the data processed in the RFU can be output to host CPU through the configuration port (CFG) or the data port (DATA). The selection is made through the MCASP_RXFMT[3] RBUSEL bit. According to port type selected, data transfer has different behaviour. For more details, refer to the Section 18.5.4.10.1.3, *Transfers Through the DATA Port*, and Section 18.5.4.10.1.4, *Transfers Through the Configuration (CFG) Bus*.

18.5.4.4.2.1 TDM Mode Reception Data Alignment Settings

RSSZ should always be programmed to match the slot size of the serial stream.

NOTE: Note that the word size is not directly programmed into the McASP, but rather is used to determine the rotation needed in the RROT field.

Table 18-330 shows the RRVRS and RROT fields for each serial format and for both integer and Q31 fractional internal representations.

Table 18-330. McASP RFU Settings

Bit Stream Order	Bit Stream Alignment	Internal Numeric Representation	MCASP_RXFMT bits	
			RROT[1]	RRVRS
MSB first[2]	Left aligned	Q31 fraction	SLOT	1
MSB first	Right aligned	Q31 fraction	WORD	1
LSB first	Left aligned	Q31 fraction	(32 - (SLOT - WORD)) % 32	0
LSB first	Right aligned	Q31 fraction	0	0
MSB first[2]	Left aligned	Integer	SLOT - WORD	1
MSB first	Right aligned	Integer	0	1
LSB first	Left aligned	Integer	32 - SLOT	0
LSB first	Right aligned	Integer	32 - WORD	0

(1) WORD = Word size rounded up to the nearest multiple of 4; SLOT = slot size; % = modulo operator
(2) To receive in I2S format, select MSB first, left aligned, and also select RDATDLY = 01 (1 bit delay)

The Table 18-330 assumes that all slot size and word size options are multiples of 4; since the receive rotate right unit only supports rotation by multiples of 4. However, the bit mask/pad unit does allow for any number of significant digits. For example, a Q31 number may have 19 significant digits (word) and be received in a 24-bit slot; this would be formatted as a word size of 20 bits and a slot size of 24 bits. However, it is possible to set the bit mask unit to only pass the 19 most-significant digits (program the mask value to FFFF E000h). The digits that are not significant can be set to a selected pad value, which can be any one of the significant digits, a fixed value of 0, or a fixed value of 1. The receive bit mask/pad unit operates on data as the final step of the receive format unit (see Figure 18-95), and the data is aligned in the same representation as it is read from the receiver (typically Q31 or integer, for example).

18.5.4.5 State-Machines

The receive and transmit sections have independent state machines.

Each state-machine controls the interactions between the various units in the McASP Rx and Tx sections, respectively. In addition, each state-machine keeps track of error conditions and serial port status. No serial transfers can occur until the RX/TX state-machine is released from reset.

The transmit state-machine is controlled by the transmit bitstream format register (MCASP_TXFMT) and it reports the McASP status and error conditions in the transmitter status register (MCASP_TXSTAT).

Similarly, the receive state-machine is controlled by the receive bitstream format register (MCASP_RXFMT) and it reports the McASP status and error conditions in the receiver status register (MCASP_RXSTAT).

18.5.4.6 TDM Sequencers

There are separate TDM sequencers for the transmit section and the receive section. Each TDM sequencer keeps track of the slot count. In addition, the TDM sequencer checks the bits of MCASP_RXTDM/MCASP_TXTDM and determines if the McASP should receive/transmit in that time slot.

There are two possibilities for a slot: The McASP either performs Rx/Tx operations during the time slot (transmit/receive bit is active), or the McASP skips Rx/Tx operations during the time slot (transmit/receive bit is inactive). In the latter case, no transfers between the XRBUF and XRSR registers in the serializer would occur during that time slot.

In addition, during time of inactive slots, the serializers programmed as transmitters place their data output pins - AXRn in a predetermined state - logic low, high, or high impedance (tri-stated) as programmed in each serializer control register MCASP_XRSRCTLn[3:2] DISMOD. Refer also to Section 18.5.4.9.2.1, *TDM Time Slots Generation and Processing*, for details on how DMA event or interrupt generations are handled during inactive time slots in TDM mode.

In case of a DIT-transmission (S/PDIF transfers): the time division multiplexing (TDM) sequencer is used to count the 384 subframes (slots) in the DIT block. If currently transmitting slot 1, slot 2 (next value of the TDM slot counter) should be used during the encode phase to select the appropriate C, V, and U bit, because the data encoded and written to a MCASP_TXBUFn register during the current time slot (slot 1) is actually shifted out on the next time slot.

The transmit TDM sequencer is controlled by the MCASP_TXTDM register and reports the current transmit slot to the MCASP_TXTDMSLOT[9:0] XSLOTCNT bit field.

18.5.4.7 McASP Software Reset

The McASP can be put into reset through the global transmit and receive control register (MCASP_GBLCTL). A valid serial clock must be supplied to the McASP to assert the software reset bits in the MCASP_GBLCTL register.

18.5.4.8 McASP Power Management

Table 18-331 describes power-management features available to the McASP.

Table 18-331. Local Power-Management Features

Feature	Registers	Description
Slave idle modes	PWRIDLESYSCONFIG[1:0] IDLE_MODE	Force-idle, no-idle, and smart-idle modes are available.

CAUTION

No wakeup schema is supported for the McASP. To ensure a correct behavior after enabling McASP at device PRCM level, the user software is strongly recommended to choose *No Idle* mode, setting PWRIDLESYSCONFIG[1:0] IDLE_MODE to 0x1. Before disabling McASP at device PRCM level, user software is strongly recommended to choose a *Smart-Idle* mode, setting PWRIDLESYSCONFIG[1:0] IDLE_MODE to 0x2.

18.5.4.9 Transfer Modes

18.5.4.9.1 Burst Transfer Mode

The McASP supports a burst transfer mode, which is useful for nonaudio data such as passing control information between two processors. Burst transfer mode uses a synchronous serial format similar to the TDM mode. The frame sync generation is not periodic or time-driven as in TDM mode, but data driven, and the frame sync is generated for each data word transferred.

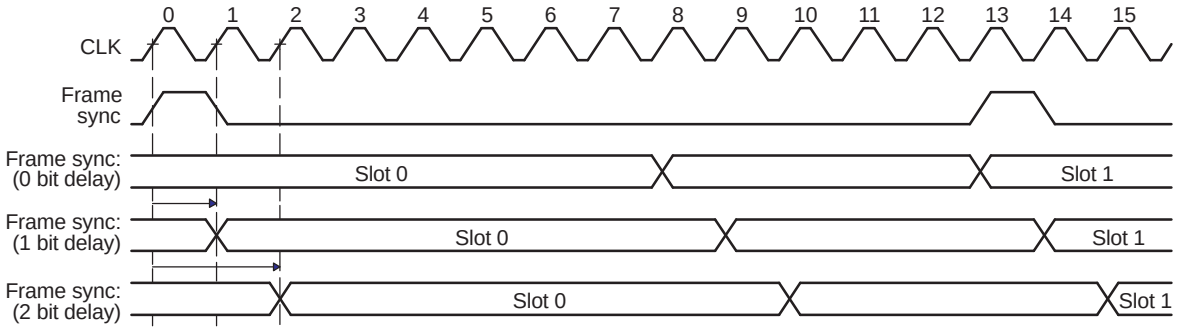
When operating in burst frame sync mode (see Figure 18-96), as specified for transmit (MCASP_TXFMCTL[15:7] = 0) and receive (MCASP_RXFMCTL[15:7] RMOD = 0), one slot is shifted for each active edge of the frame sync signal that is recognized. Additional clocks after the slot and before the next frame sync edge are ignored.

In burst frame sync mode, the frame sync delay may be specified as 0, 1, or 2 serial clock cycles. This is the delay between the frame sync active edge and the start of the slot. The frame sync signal lasts for a single bit clock duration (MCASP_RXFMCTL[4] FRWID = 0, MCASP_TXFMCTL[4] FXWID = 0).

For transmit, when generating the transmit frame sync internally, the frame sync begins when the previous transmission has completed and when all the XBUFn (for every serializer set to operate as a transmitter) has been updated with new data.

For receive, when generating the receive frame sync internally, frame sync begins when the previous transmission has completed and when all the RBUFn (for every serializer set to operate as a receiver) has been read.

Figure 18-96. Burst Frame Sync Mode



The control registers must be configured as follows for the burst transfer mode. The burst mode specific bit fields are in bold face:

- MCASP_PFUNC: The clock, frame, data pins must be configured for McASP function.
- MCASP_PDIR: The clock, frame, data pins must be configured to the direction desired.
- MCASP_PDOUT, MCASP_PDIN, MCASP_PDSET, MCASP_PDCLR: Not applicable. Leave at default.
- MCASP_GBLCTL: Follow the initialization sequence in Section 18.5.5.1.2, *McASP Global Initialization*, to configure this register.
- MCASP_AMUTE: Not applicable. Leave at default.
- MCASP_LBCTL: If loopback mode is desired, configure this register according to Section 24.6.4.14 Loopback Modes, otherwise leave this register at default.
- MCASP_TXDITCTL: DITEN must be left at default 0 to select non-DIT mode. Leave the register at default.
- MCASP_RXMASK/MCASP_TXMASK: Mask desired bits according to Section 18.5.4.4, *Format Units*.
- MCASP_RXFMT/MCASP_TXFMT: Program all fields according to data format desired. See Section 18.5.4.4, *Format Units*.
- MCASP_RXFMT/MCASP_TXFMT: Clear RMOD/XMOD bits to 0 to indicate burst mode. Clear FRWID/FXWID bits to 0 for single bit frame sync duration. Configure other fields as desired.
- MCASP_ACLKRCTL/MCASP_ACLKXCTL: Program all fields according to bit clock desired. See Section 18.5.4.2, *McASP Clock and Frame-Sync Configurations*.
- MCASP_AHCLKRCTL/MCASP_AHCLKXCTL: Program all fields according to high-frequency clock desired. See Section 18.5.4.2, *McASP Clock and Frame-Sync Configurations*.
- MCASP_RXTDM/MCASP_TXTDM: Program RTDMS0/XTDMS0 to 1 to indicate one active slot only. Leave other fields at default.
- MCASP_EVTCTLR/MCASP_EVTCTLX: Program all fields according to interrupts desired.
- MCASP_RXCLKCHK/MCASP_TXCLKCHK: Not applicable. Leave at default.
- MCASP_XRSRCTLn: Program SRMOD to inactive/transmitter/receiver as desired. DISMOD is not applicable and should be left at default.
- MCASP_DITCSRAi, MCASP_DITCSRBi, MCASP_DITUDRAi, MCASP_DITUDRBi: Not applicable. Leave at default.

18.5.4.9.2 Time-Division Multiplexed (TDM) Transfer Mode

The McASP time-division multiplexed (TDM) transfer mode supports the TDM format discussed in Section 18.5.2.2.3.

Transmitting data in the TDM transfer mode requires a minimum set of pins:

- ACLKX - transmit bit clock
- AFSX - transmit frame sync (or commonly called left/right clock)
- One or more serial data pins, AXRn, whose serializers are configured to transmit

For more details on McASP transmitting serializers clock and frame sync options, refer to the Section 18.5.4.2.1, *Transmit Clock*, and Section 18.5.4.2.3, *Frame-Sync Generator* .

Similarly, to receive data in the TDM transfer mode requires a minimum set of pins:

- ACLKR - receive bit clock
- AFSR - receive frame sync (or commonly called left/right clock)
- One or more serial data pins, AXRn, whose serializers are configured to receive

For more details on McASP receiving serializers clock and frame sync options, refer to Section 18.5.4.2.2, *Receive Clock*, and Section 18.5.4.2.3, *Frame-Sync Generator* .

The control registers must be configured as follows for the TDM mode. The TDM mode specific bit fields are highlighted in bold:

- MCASP_PFUNC: The clock, frame, data pins must be configured for McASP function.
- MCASP_PDIR: The clock, frame, data pins must be configured to the direction desired.
- MCASP_PDOUT, MCASP_PDIN, MCASP_PDSET, MCASP_PDCLR: Not applicable. Leave at default.
- MCASP_GBLCTL: Follow the initialization sequence is described in the Section 18.5.5.2, *Operational Modes Configuration*.
- MCASP_AMUTE: Leave this register at default state.
- MCASP_LBCTL: If loopback mode is desired, configure this register according to Section 18.5.4.14, otherwise leave this register at default.
- MCASP_TXDITCTL: DITEN must be left at default 0 to select TDM mode (transmitters only).
- MCASP_RXMASK/MCASP_TXMASK: Mask desired bits according to Section 18.5.4.4, *Format Units*.
- MCASP_RXFMT/MCASP_TXFMT: Program all fields according to data format desired. See the Section 18.5.4.4, *Format Units*.
- MCASP_RXFMCTL/MCASP_TXFMCTL: Set RMOD/XMOD bits to (0x2 - 0x20) for Rx/Tx (2- 32 slots) TDM mode. In addition, set RMOD to 0x180 if 384-slot TDM stream has to be received by McASP. Configure other fields as desired.
- MCASP_ACLKRCTL/MCASP_ACLKXCTL: Program all fields according to bit clock desired. For more inforamtion, refer to Section 18.5.4.2.
- MCASP_AHCLKRCTL/MCASP_AHCLKXCTL: Program all fields according to high-frequency clock desired. For more details, refer to Section 18.5.4.2.
- MCASP_RXTDM/MCASP_TXTDM: Program all fields according to the time slot characteristics desired.
- MCASP_EVTCTLX: Program all fields according to transmit interrupts desired.
- MCASP_RXCLKCHK/MCASP_TXCLKCHK: Program all fields according to clock checking desired.
- MCASP_XRSRCTLn: Program all fields according to serializer operation desired.

NOTE: The MCASP_DITCSRAi, MCASP_DITCSRBi, MCASP_DITUDRAi, MCASP_DITUDRBi (i=0 to 5) settings are NOT applicable in TDM transfer modes. They have to be kept at their default values.

18.5.4.9.2.1 TDM Time Slots Generation and Processing

TDM mode on the McASP can extend to support multiprocessor applications, with up to 32 time slots per frame. For each of the time slots, the McASP may be configured to participate or to be inactive by configuring MCASP_TXTDM and/or MCASP_RXTDM registers.

The TDM sequencer (separate ones for transmit and receive) functions in this mode. The TDM sequencer counts the slots beginning with the frame sync. For each slot, the TDM sequencer checks the respective bit in either MCASP_TXTDM or MCASP_RXTDM to determine if the McASP transmits/receives in that time slot.

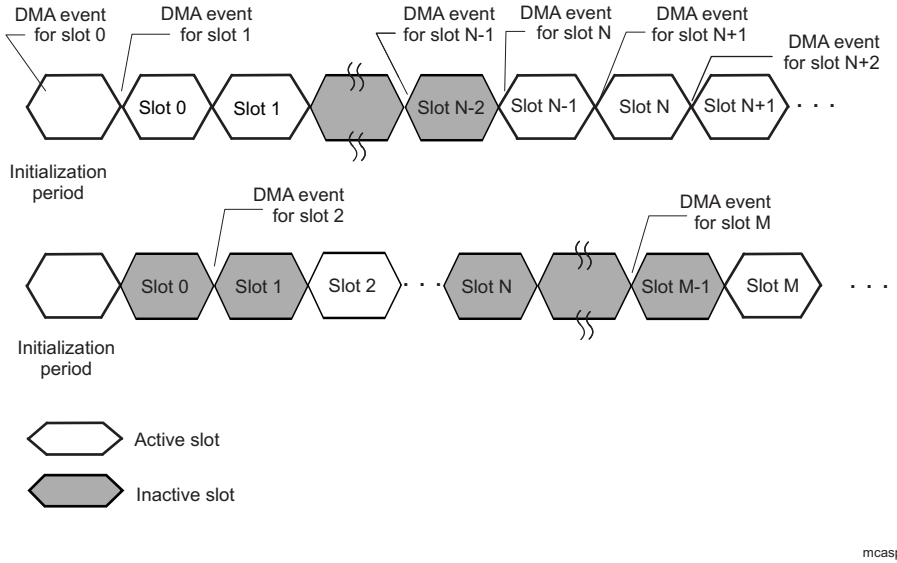
NOTE: If a MCASP_TXTDM/MCASP_RXTDM bit defines an active slot (number of slot matches the bit position), the McASP functions normally during that time slot; otherwise, the McASP is inactive during that time slot; no update to the buffer occurs, and no event is generated. McASP (transmit only) data pins are automatically set to a high-impedance state, 0, or 1 during that slot, as determined by bitfield MCASP_XRSRCTLn[3:2] DISMOD.

Figure 18-97 shows when the transmit DMA event - AXEVT is generated. See Section 18.5.4.10.1, *Data Ready Status and Event/Interrupt Generation* for details on data ready and the initialization period indication. The transmit DMA event for an active time slot (slot N) is generated during the previous time slot (slot N - 1), regardless of the previous time slot (slot N - 1) being active or inactive.

During an active transmit time slot (slot N), if the next time slot (slot N + 1) is configured to be active, the copy from XRBUFn to XRSRn generates the DMA event for time slot N + 1. If the next time slot (slot N + 1) is configured to be inactive, then the DMA event will be delayed to time slot M - 1. In this case, slot M is the next active time slot. The DMA event for time slot M is generated during the first bit time of slot M - 1.

The receive DMA event is generated after data is received in the buffer (looks back in time). If a time slot is disabled, then no data is copied to the buffer for that time slot and no DMA event is generated.

Figure 18-97. Transmit DMA Event (AXEVT) Generation in TDM Time Slots



18.5.4.9.2.2 Special 384-Slot TDM Mode for Connection to External DIR

The McASP receiver also supports a 384 time slot TDM mode (DIR mode), to support S/PDIF receiver ICs whose natural block (block corresponds to McASP frame) size is 384 samples. The receive TDM time slot register (MCASP_RXTDM) should be programmed to all 1s during reception of a DIR block. Other TDM functionalities (for example, inactive slots) are not supported (only the slot counter counts the 384 subframes in a block). To receive data in DIR mode, the following pins are typically needed:

- ACLKR - receive bit clock
- AFSR - receive frame sync (or commonly called left/right clock)
- In this mode, AFSR should be connected to a DIR which outputs a start of block signal, instead of LRCLK
- One or more serial data pins, AXRn, whose serializers have been configured to receive
- For this special DIR mode, the control registers can be configured just as for TDM mode, except set

RMOD in MCASP_RXFMCTL to 384 (0x180) to receive 384 time slots

18.5.4.9.3 DIT Transfer Mode

The DIT transfer mode of the McASP also supports transmission of audio data in S/PDIF, AES-3, and IEC-60958 formats. These formats are designed to carry audio data between different systems through an optical or coaxial cable. The DIT mode applies only to a serializer configured as transmitter, not as receiver. For a description of the S/PDIF format, see Section 18.5.2.2.5, *S/PDIF Coding Format*.

18.5.4.9.3.1 Transmit DIT Encoding

When the McASP operates in DIT mode, the data transmitted is output as a biphase-mark encoded bitstream, with preamble, channel status, user data, validity, and parity automatically stuffed into the bitstream by the McASP. The McASP includes separate validity bits for even/odd subframes and two 384-bit RAM modules to hold channel status and user data bits.

NOTE: The transmit TDM time slot register (MCASP_TXTDM) should be programmed to all 1s during DIT mode. TDM functionality is not supported in DIT mode, except that the TDM slot counter counts the DIT subframes.

To transmit data in DIT mode, the following pins are typically required:

- AHCLKX – transmit high-frequency master clock (The internal clock source can be used instead.)
- One serial data pin (AXRn) of a serializer n configured to transmit.

For DIT Mode Transmission Data Alignment Settings see Section 18.5.4.4.1.2.

If the McASP is configured to transmit in the DIT mode on more than one serial data pin, the bit streams on all pins will be synchronized. In addition, although they will carry unique audio data, they will carry the same channel status, user data, and validity information.

The actual 24-bit audio data must always be in bit positions 23–0 after passing through the first three stages of the transmit format unit.

18.5.4.9.3.2 Transmit DIT Clock and Frame-Sync Generation

The DIT transmitter works only in the following configuration:

- In the transmit frame control register (MCASP_TXFMCTL):
  - Internally generated transmit frame sync, FSXM = 1
  - Rising-edge frame sync, FSXP = 0
  - Bit-width frame sync, FXWID = 0
  - 384-slot TDM, XMOD = 1 1000 0000b
- In the transmit clock control register (MCASP_ACLKXCTL), ASYNC = 1
- In the transmit bitstream format register (MCASP_TXFMT), XSSZ = 1111 (32-bit slot size)

All combinations of AHCLKX and ACLKX are supported.

The following summarizes the register configurations required for DIT mode. DIT mode-specific bit fields are in bold face:

- MCASP_PFUNC: The data pin - AXRn must be configured for McASP function. If AHCLKX is used, it must also be configured for McASP function. Other pins can be configured to function as GPIOs, if desired.
- MCASP_PDIR: The data pin must be configured as output. If internal clock source AUXCLK is used as the reference clock, it may be output as the AHCLKX device level signal by configuring AHCLKX pin as an output.
- MCASP_GBLCTL: Global initialization
- MCASP_AMUTE: Leave this register at default state.
- MCASP_TXDITCTL: The DITEN bit must be set to 0b1 to enable DIT mode. Configure other bits as

desired.

- MCASP_TXMASK: Mask the desired bits, depending upon left-aligned or right-aligned internal data.
- MCASP_TXFMT: XDATDLY = 0. XRVRS = 0. XPAD = 0. XSSZ = Fh (32-bit slot). XBUSEL = configured as desired. The XROT bit is configured, as described in the Section 18.5.4.4.1.2 .
- MCASP_TXFMCTL: Configure the bits according to former discussions.
- MCASP_ACLKXCTL: ASYNC = 1. Program the CLKXDIV bits to obtain the bit clock rate desired. CLKXM = 1.
- MCASP_AHCLKXCTL: Program the HCLKXDIV bits to obtain the high-frequency bit clock rate desired.
- MCASP_TXTDM: Set to FFFF FFFFh for all active slots for DIT transfers.
- MCASP_EVTCTLX: Program all fields according to the interrupts desired.
- MCASP_TXCLKCHK: Program all fields according to the clock checking desired.
- MCASP_XRSRCTLn: Set SRMOD = 1 (transmitter) for the DIT pins.
- MCASP_DITCSRAi and MCASP_DITCSRBi: Program the channel status bits as desired.
- MCASP_DITUDRAi and MCASP_DITUDRBi: Program the user data bits as desired.

NOTE: In DIT mode, the transmitter can support a 192 kHz frame rate (stereo) on up to 2 serial data pins simultaneously (note that the internal bit clock for DIT runs two times faster than the equivalent bit clock for TDM (I2S) mode, due to the need to generate Biphase Mark Encoded Data - see Section 18.5.2.2.5.1).

18.5.4.9.3.3 DIT Channel Status and User Data Register Files

The channel status registers (MCASP_DITCSRAi and MCASP_DITCSRBi) and user data registers (MCASP_DITUDRAi and MCASP_DITUDRBi) are not double-buffered. Typically, programmers use one of the synchronizing interrupts, such as the last slot, to create an event at a safe time so the register may be updated. In addition, the software reads the transmit TDM slot counter to determine which word of the register is being used.

It is a software requirement to avoid writing to the word of user data and channel status that are being used to encode the current time slot; otherwise, it is undetermined whether old or new data is used to encode the bitstream.

The DIT subframe format is defined in Section 18.5.2.2.5.2, *S/PDIF Subframe Format*. The channel status information (C) and user data (U) are defined in the following DIT control registers:

- MCASP_DITCSRA0 to MCASP_DITCSRA5: The 192 bits in these six registers contain the channel status information for the left channel within each frame.
- MCASP_DITCSRB0 to MCASP_DITCSRB5: The 192 bits in these six registers contain the channel status information for the right channel within each frame.
- MCASP_DITUDRA0 to MCASP_DITUDRA5: The 192 bits in these six registers contain the user data information for the left channel within each frame.
- MCASP_DITUDRB0 to MCASP_DITUDRB5: The 192 bits in these six registers contain the user data information for the right channel within each frame.
- The S/PDIF block format is shown in Figure 18-87. There are 192 frames within a block (frame 0 to frame 191). There are two subframes within each frame (subframes 1 and 2 for the left and right channels, respectively).

The channel status and user data information sent on each subframe is summarized in Table 18-332.

Table 18-332. Channel Status and User Data for Each DIT Block

Frame	Subframe	Preamble	Channel Status Defined in:	User Data Defined in:
Defined by DITCSRA0, DITCSRB0, DITUDRA0, DITUDRB0				
0	1 (L)	B	DITCSRA0[0]	DITUDRA0[0]
0	2 (R)	W	DITCSRB0[0]	DITUDRB0[0]

Table 18-332. Channel Status and User Data for Each DIT Block (continued)

Frame	Subframe	Preamble	Channel Status Defined in:	User Data Defined in:
1	1 (L)	M	DITCSRA0[1]	DITUDRA0[1]
1	2 (R)	W	DITCSRB0[1]	DITUDRB0[1]
2	1 (L)	M	DITCSRA0[2]	DITUDRA0[2]
2	2 (R)	W	DITCSRB0[2]	DITUDRB0[2]
…	…	…	…	…
31	1 (L)	M	DITCSRA0[31]	DITUDRA0[31]
31	2 (R)	W	DITCSRB0[31]	DITUDRB0[31]
Defined by DITCSRA1, DITCSRB1, DITUDRA1, DITUDRB1				
32	1 (L)	M	DITCSRA1[0]	DITUDRA1[0]
32	2 (R)	W	DITCSRB1[0]	DITUDRB1[0]
…	…	…	…	…
63	1 (L)	M	DITCSRA1[31]	DITUDRA1[31]
63	2 (R)	W	DITCSRB1[31]	DITUDRB1[31]
Defined by DITCSRA2, DITCSRB2, DITUDRA2, DITUDRB2				
64	1 (L)	M	DITCSRA2[0]	DITUDRA2[0]
64	2 (R)	W	DITCSRB2[0]	DITUDRB2[0]
…	…	…	…	…
95	1 (L)	M	DITCSRA2[31]	DITUDRA2[31]
95	2 (R)	W	DITCSRB2[31]	DITUDRB2[31]
Defined by DITCSRA3, DITCSRB3, DITUDRA3, DITUDRB3				
96	1 (L)	M	DITCSRA3[0]	DITUDRA3[0]
96	2 (R)	W	DITCSRB3[0]	DITUDRB3[0]
…	…	…	…	…
127	1 (L)	M	DITCSRA3[31]	DITUDRA3[31]
127	2 (R)	W	DITCSRB3[31]	DITUDRB3[31]
Defined by DITCSRA4, DITCSRB4, DITUDRA4, DITUDRB4				
128	1 (L)	M	DITCSRA4[0]	DITUDRA4[0]
128	2 (R)	W	DITCSRB4[0]	DITUDRB4[0]
…	…	…	…	…
159	1 (L)	M	DITCSRA4[31]	DITUDRA4[31]
159	2 (R)	W	DITCSRB4[31]	DITUDRB4[31]
Defined by DITCSRA5, DITCSRB5, DITUDRA5, DITUDRB5				
160	1 (L)	M	DITCSRA5[0]	DITUDRA5[0]
160	2 (R)	W	DITCSRB5[0]	DITUDRB5[0]
…	…	…	…	…
191	1 (L)	M	DITCSRA5[31]	DITUDRA5[31]
191	2 (R)	W	DITCSRB5[31]	DITUDRB5[31]

18.5.4.10 Data Transmission and Reception

The McASP is serviced by writing data to the MCASP_TXBUFn registers for transmit operations, and by reading data from the MCASP_RXBUFn registers for receive operations. The McASP sets status flags and notifies the software whenever data is ready to be serviced. The Section 18.5.4.10.1, *Data Ready Status and Event/Interrupt Generation*, discusses data-ready status in details.

The McASP transmit/receive XRBUFn buffer can be accessed through one of the two peripheral ports of the device:

- DATA port: This port is dedicated to DMA initiated data transfers on the device for McASP transmit (Tx) purposes.

- Configuration bus (CFG): The configuration bus- CFG port is used for peripheral configuration control and receive/transmit data transfers initiated by the host CPU in the device.

Section 18.5.4.10.1.3, *Transfers Through the Data Port (DATA)*, and Section 18.5.4.10.1.4, *Transfers Through the Configuration Bus (CFG),* discuss how to perform transfers through the data port (DATA) and the configuration port (CFG), respectively.

A device CPU and DMA usages are discussed in Section 18.5.4.10.1.5, *Using the device CPUs for McASP Servicing*, and Section 18.5.4.10.1.6, *Using the DMA for McASP Servicing*, respectively.

McASP DATA port allows DMAs to access the McASP transmit buffer more efficiently on the L3_MAIN-interconnect or L4_PER2 interconect, using burst transfers. The physical addresses to access these registers are listed in Section 18.5.6.2.5.

18.5.4.10.1 Data Ready Status and Event/Interrupt Generation

18.5.4.10.1.1 Transmit Data Ready

The transmit data ready flag - XDATA in the MCASP_TXSTAT register reflects the data ready status of XRBUFn buffers for all of the active slot transmitting serializers. The XDATA flag is set whenever data is transferred from a transmitting serializer buffer - XRBUFn to its corresponding XRSRn shift register. Thus, the XDATA bit indicates the global event that some of the serializers data buffer - XRBUFn is emptied and ready to accept new data from the host (CPU or DMA). The transmit data ready event is individually indicated per serializer in its corresponding control register MCASP_XRSRCTLn[4] XRDY status bit. When this bit is set to 0b1, it notifies to host that this serializer Tx buffer must be serviced (written). When MCASP_TXBUFn is written to by the host, the MCASP_XRSRCTLn[4] XRDY is deasserted to 0b0. As XDATA global flag is an OR-event of all active serializers XRDY flags, it indicates to software the moment, when write service operation has to be initiated by the McASP host (XDATA=0b1). The XRDY flags have to be sequentially scanned by user software to determine which serializer MCASP_TXBUFn register has to be currently written. Once all requested MCASP_TXBUFn are written, the serializers control XRDY flags are cleared to 0b0. As a consequence, XDATA flag is deasserted to 0b0, to indicate to SW that write operation is completed for all serializers.

The global XDATA flag can be cleared when the MCASP_TXSTAT[5] XDATA bit is written to 0b1, or once MCASP_TXBUFn registers of all the serializers, that have previously raised their XRDY flags, are written with corresponding active slot data by the host.

Whenever XDATA is set, the AXEVT event is automatically generated on MCASPi_DREQ_TX line (if enabled in the MCASP_XEVTCTL register) to notify the DMA of the MCASP_TXBUFn empty status. An interrupt - MCASPi_IRQ_AXEVT can be also generated if the XDATA interrupt is enabled in the MCASP_EVTCTLX register (for details, see Section 18.5.4.12.1, *Transmit Data Ready Interrupt*).

For DMA requests, the McASP does not require that MCASP_TXSTAT be read between DMA events. This means that, even if MCASP_TXSTAT already has the XDATA flag set to 1 from a previous request, the next transfer triggers another DMA request.

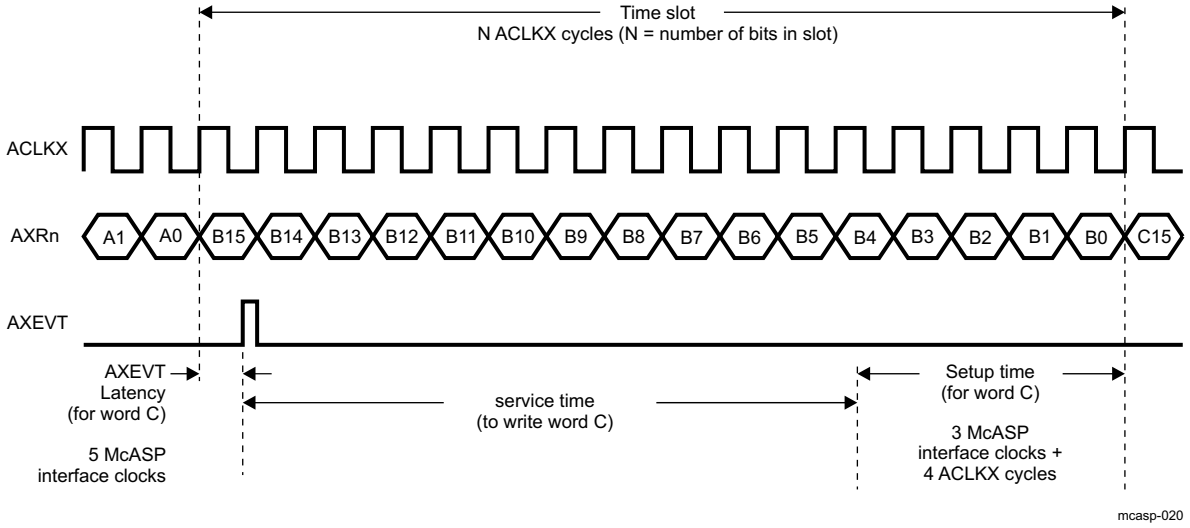
Because the serializer acts in lockstep, only one DMA event is generated to indicate that the transmit serializer is ready to be written to with new data.

Figure 18-98 shows the timing details of when AXEVT is generated at the McASP boundary. In this example, as soon as the last bit (A0) of word A is transmitted, the McASP sets the XDATA flag and generates an AXEVT event. However, it takes up to five McASP interface clocks (AXEVT latency) before AXEVT is active at the McASP boundary. Upon AXEVT, the CPU can begin servicing the McASP by writing word C into the MCASP_TXBUFn (service time). The CPU must write word C into the MCASP_TXBUFn within the setup time required by the McASP (setup time).

The maximum service time (see Figure 18-98) can be calculated as:

Service Time = Time Slot – AXEVT Latency –Setup Time

Figure 18-98. Service Time Upon Transmit DMA Event (AXEVT)



18.5.4.10.1.2 Receive Data Ready

Similarly, the receive data ready flag - RDATA in the MCASP_RXSTAT register reflects the data ready status of XRBUFn buffers for all of the active slot receiving serializers. The RDATA flag is set whenever data is transferred from a receiving serializer shift register XRSRn to its corresponding XRBUFn data buffer. Thus, the RDATA bit indicates the global event that some of the receivers data buffer - RXBUFn already contains received data (i.e. a buffer is full) and is ready to transfer it to the host. The receive data ready event is individually indicated per serializer in its corresponding control register MCASP_XRSRCTLn [5] RRDY status bit. When this bit is set to 0b1, it notifies to host that this serializer Rx buffer must be serviced (read). When MCASP_RXBUFn is read from the host, the MCASP_XRSRCTLn [5] RRDY is deasserted to 0b0. As RDATA global flag is an OR-event of all active serializers RRDY flags, it indicates to software the moment, when read service operation has to be initiated by the McASP host (RDATA=0b1). The RRDY flags have to be sequentially scanned by user software to determine which serializer MCASP_RXBUFn register has to be currently read. Once all requested MCASP_RXBUFn are read, the serializers control RRDY flags are cleared to 0b0. As a consequence, RDATA flag is deasserted to 0b0, to indicate to SW that read operation is completed for all serializers.

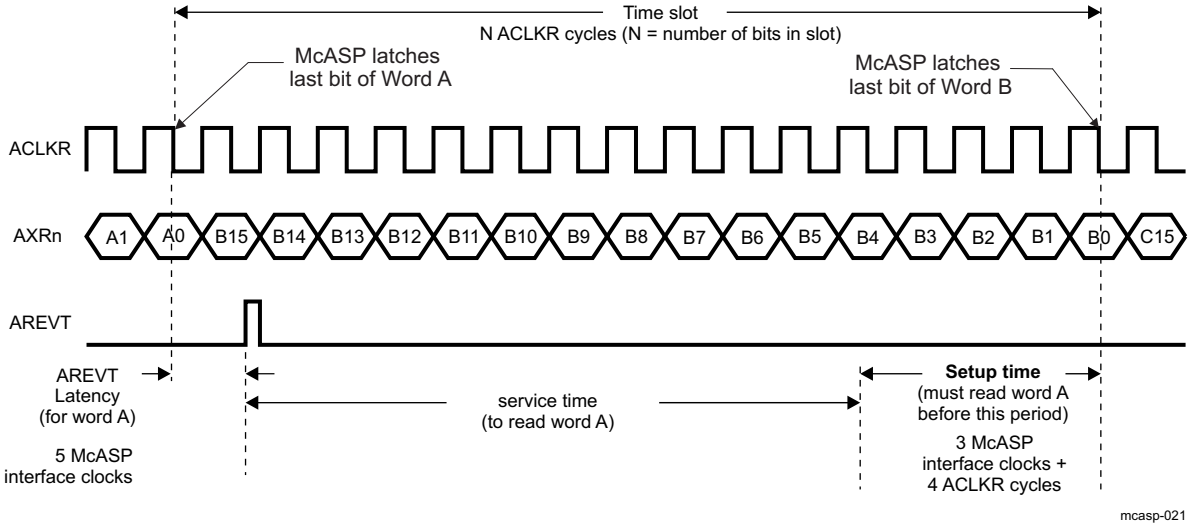
The global RDATA flag can be cleared when the MCASP_RXSTAT[5] RDATA bit is written to 0b1, or once MCASP_RXBUFn registers of all the serializers, that have previously raised their RRDY flags, are read by the host.

Whenever RDATA is set, the AREVT event is automatically generated on MCASPi_DREQ_RX line (if enabled in the MCASP_REVTCTL register) to notify the DMA of the MCASP_RXBUFn full status. An interrupt - MCASPi_IRQ_AREVT can be also generated if the RDATA interrupt is enabled in the MCASP_EVTCTLR register (for details, see Section 18.5.4.12.1, *Receive Data Ready Interrupt*).

Figure 18-99 shows the timing details of when AREVT event is generated at the McASP boundary. In this example, as soon as the last bit (bit A0) of Word A is received, the McASP sets the RDATA flag and generates an AREVT event. However, it takes up to five McASP interface clocks (AREVT Latency) before AREVT is active at the McASP boundary. Upon AREVT, the CPU can begin servicing the McASP by reading Word A from the MCASP_RXBUFn (service time). The CPU must read Word A from the MCASP_RXBUFn register no later than the setup time required by the McASP (Setup Time).

The maximum service time (see Figure 18-99) can be calculated as:
Service Time = Time Slot - AREVT Latency - Setup Time

Figure 18-99. CPU Service Time Upon Receive Event (AREVT)



18.5.4.10.1.3 Transfers Through the Data Port (DATA)

CAUTION

To perform internal transfers through the DATA port, clear the XBUSEL/RBUSEL bit to 0b0 in the MCASP_TXFMT/MCASP_RXFMT register, respectively. Failure to do so may result in software malfunction.

NOTE: McASP1, whose data portis accessible directly via L3_MAIN, does not support FIFO/constant addressing modes. Incrementing transfers must be used instead.

In a typical McASP transfer scenario, the DMA Controller write accesses the XRBUFn transmit buffer through the McASP data port (DATA) on L3_MAIN Interconect for McASP1 and on L4_PER2 Interconect for McASP2/3 . CPU hosts can access both XRBUFn transmit and receive data buffers on their corresponding DATA port address via DATA port corresponding address. To perform transfers through the DATA port, simply have the DMA Controller write the McASP Tx buffer through Interconect DATA port location. Refer to Section 18.5.6.2.5. Although the transfer is passed through an integrated AFIFO transmit/receive buffer, the host (DMA or CPU) must follow the described below procedure to access the data buffers of each serializer, regardless the AFIFO is enabled or disabled. The AFIFO operation is described in Section 18.5.4.11.

For accesses through the DATA port, the DMA/CPU services all the serializers through accessing only a single address. In addition, as can be seen in Section 18.5.6.2.5, the same physical DATA port address is used regardless of a read or write access is performed. The McASP automatically cycles through the active slot transmitting/receiving serializers, internally generating the appropriate offsets.

NOTE: DATA port allows the DMA/CPU to automatically access only the data buffers. There is no way for DMA/CPU to access the McASP configuration registers addressing their corresponding McASP DATA port.

For transmit operations through the DATA port, the host must always write to the same transmit buffer DATA port address (which is same than the receive buffer DATA port adress) to service all of the active slot transmitting serializers. Regardless of McASP serializer 0 being configured inactive or active, the user software must always configure the destination address to match the DATA port location of TXBUF buffer (See Section 18.5.6.2.5).

In addition, the DMA/CPU must write the buffers of all transmitting serializers in incremental (although not necessarily consecutive) order. For example, if only serializers 1 and 3 are set up as active transmitters, to the same transmit buffer DATA port address twice - first data for serializer 1 and second data for serializer 3 upon each transmit data ready event. This exact servicing order must be followed so that data appears in the appropriate serializers.

NOTE: For write transfers through McASP DATA port it is preferable to use DMA on corresponding Interconect. This is because DMAs initiated traffic gets better advantage of the burst transfers supported by DATA port.

For receive operations through the DATA port, the DMA/CPU must always read from the same receive buffer DATA port address (which is same than the transmit buffer DATA port adress) to service all of the active slot receiving serializers. Regardless of McASP serializer 0 being configured inactive or active, the user software must always configure the DMA/CPU source address to match the DATA port location of RXBUF buffer (See Section 18.5.6.2.5).

In addition, reads from the receive buffer for all active slot receiving serializers through the Rx DATA port return data in incremental (although not necessarily consecutive) order. For example, if serializers 0, 1 and 3 are set up as active receivers, the MPU should read from the same receive buffer DATA port address three times to obtain data for serializers 0,1 and 3 in this exact order, upon each receive data ready event.

NOTE: To service a serializer for a transmit or receive operation through the McASP DATA port, the initiator always writes (preferably DMA) and reads from the same address (refer to Section 18.5.6.2.5), respectively.

See Section 18.5.6.2.5, *MCASP_DATA Register Summary*, for more details about XRBUFn buffer physical address corresponding to the McASP DATA port on:

- Main Interconect (L3_MAIN or L4_PER2)

NOTE: When transmitting through the DATA port, the DMA/CPU must write data (at the same address) to each serializer configured as *active* (active slot selected in MCASP_TXTDM) and *transmit* (Tx enabled in MCASP_XRSRCTLn) within each time slot. Failure to do so results in a buffer underrun condition (see Section 18.5.4.15.1, *Buffer Underrun Error - Transmitter*). Similarly, when DMA/CPU receives, data must be read from each serializer configured as *active* (active slot selected in MCASP_RXTDM) and *receive* (Rx enabled in MCASP_XRSRCTLn) within each time slot. Failure to do results in a buffer overrun condition (see Section 18.5.4.15.2, *Buffer Overrun Error - Receiver*).

18.5.4.10.1.4 Transfers Through the Configuration Bus (CFG)

CAUTION

To perform internal transfers through the configuration bus, set the XBUSEL/RBUSEL bit to 1 in the MCASP_TXFMT/MCASP_RXFMT registers, respectively. Failure to do so may result in software malfunction.

NOTE: McASP1, whose data port is accessible directly via L3_MAIN does not support FIFO/constant addressing modes. Incrementing transfers must be used instead.

In this method, the DMA/CPU accesses the XRBUFn transmit or receive buffer through corresponding configuration bus (CFG) address.

The exact XRBUFn transmit/receive buffer physical address for any particular serializer is determined by adding the transmit/receive buffer alias register offset for that particular serializer to the base address of McASP CFG port actual for L4_PER2 accesses. The XRBUFn buffer of the n-th serializer configured as a transmitter is aliased - MCASP_TXBUFn in the CFG port address space. For example, the XRBUF2 transmit buffer is mapped as the MCASP_TXBUF2 register. Similarly, the XRBUFn buffer of the n-th serializer configured as a receiver is aliased - MCASP_RXBUFn in the CFG port address space. For example, the XRBUF3 receive buffer is mapped as the MCASP_RXBUF3 register.

Accessing the XRBUF through the DATA port (see Section 18.5.4.10.1.3) is different than CFG port accesses because the DATA port access demands the same physical address, regardless of transfer direction or current channel index , while accessing through the peripheral configuration port - CFG, the DMA/CPU must provide the exact MCASP_TXBUFn or MCASP_RXBUFn address upon accessing n-th serializer TX or RX buffer, respectively. For more details about MCASP_TXBUFn and MCASP_RXBUFn addresses corresponding to McASP CFG port, see Section 18.5.6.2.1, *MCASP_CFG Register Summary*.

18.5.4.10.1.5 Using a Device CPU for McASP Servicing

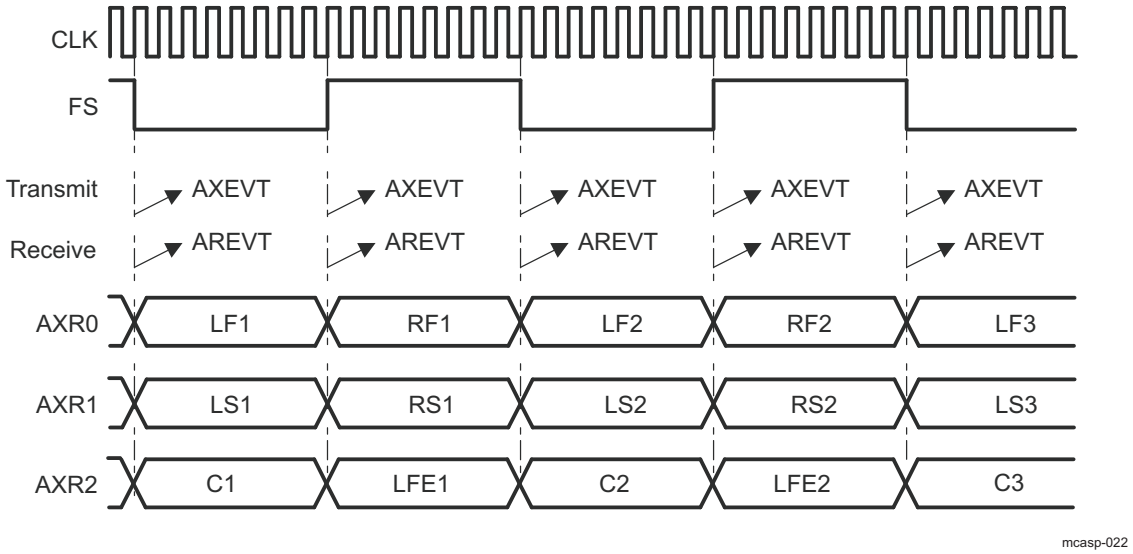
The device CPUs can be used to service the McASP transmit channels through interrupts (upon MCASPi_IRQ_AXEVT and MCASPi_IRQ_AREVT interrupts). Because these interrupt events are connected to device IRQ_CROSSBAR module, they could be software mapped to input IRQ lines of any device CPU. Another way to service the transmit and receive channels, a polling of the XDATA bit in the MCASP_TXSTAT register and RDATA bit in the MCASP_RXSTAT register can be performed by device CPUs, respectively. As discussed in Section 18.5.4.10.1.3, *Transfers Through the Data Port (DATA)*, and Section 18.5.4.10.1.4, *Transfers Through the Configuration Bus (CFG)* , the device CPUs can access McASP XRBUF serializer buffer through their corresponding DATA and CFG port locations.

To use the device CPUs to service the McASP through interrupts, the XDATA/RDATA bit must be enabled in the respective MCASP_EVTCTLX/MCASP_EVTCTLR registers, to generate interrupts MCASPi_IRQ_AXEVT/MCASPi_IRQ_AREVT to the device CPUs upon data ready

18.5.4.10.1.6 Using the DMA for McASP Servicing

The typical scenario is to use the DMA to service the McASP transmit and receive logic through the DATA port, although the DMA can also service the McASP through the configuration bus (CFG). The transfer passes through integrated AFIFO transmit/receive buffer. If AFIFO is enabled, DMA requests are collected and fed to a device DMA controller (see Figure 18-89). The data transfer is managed by the AFIFO according to generated transmit and receive events in the McASP and data is fed to transmit buffers and fetched from receive buffers as described in Section 18.5.4.11. The generation of transmit and receive request is described below. After generation of transmit/receive DMA events from McASP module, these events are collected in AFIFO and on specific AFIFO conditions described in Section 18.5.4.11 the requests (transmit or receive) are forwarded to a DMA controller via MCASPi_DREQ_TX and MCASPi_DREQ_RX outputs. If the AFIFO is disabled (default state) it is transperrant for the McASP module and all request are directly sent to the DMA controller.

Figure 18-100. DMA Transmit and Receive Event in an Audio Example – One Event



In transmit mode, the DMA event - AXEVT (MCASPi_DREQ_TX output), which is triggered upon each XDATA transition from 0 to 1, is used to service the McASP TXBUFn transmit buffers. In receive mode, the DMA event AREVT (MCASPi_DREQ_RX output) which is triggered upon each RDATA transition from 0 to 1, is used to service the McASP RXBUFn receive buffers.

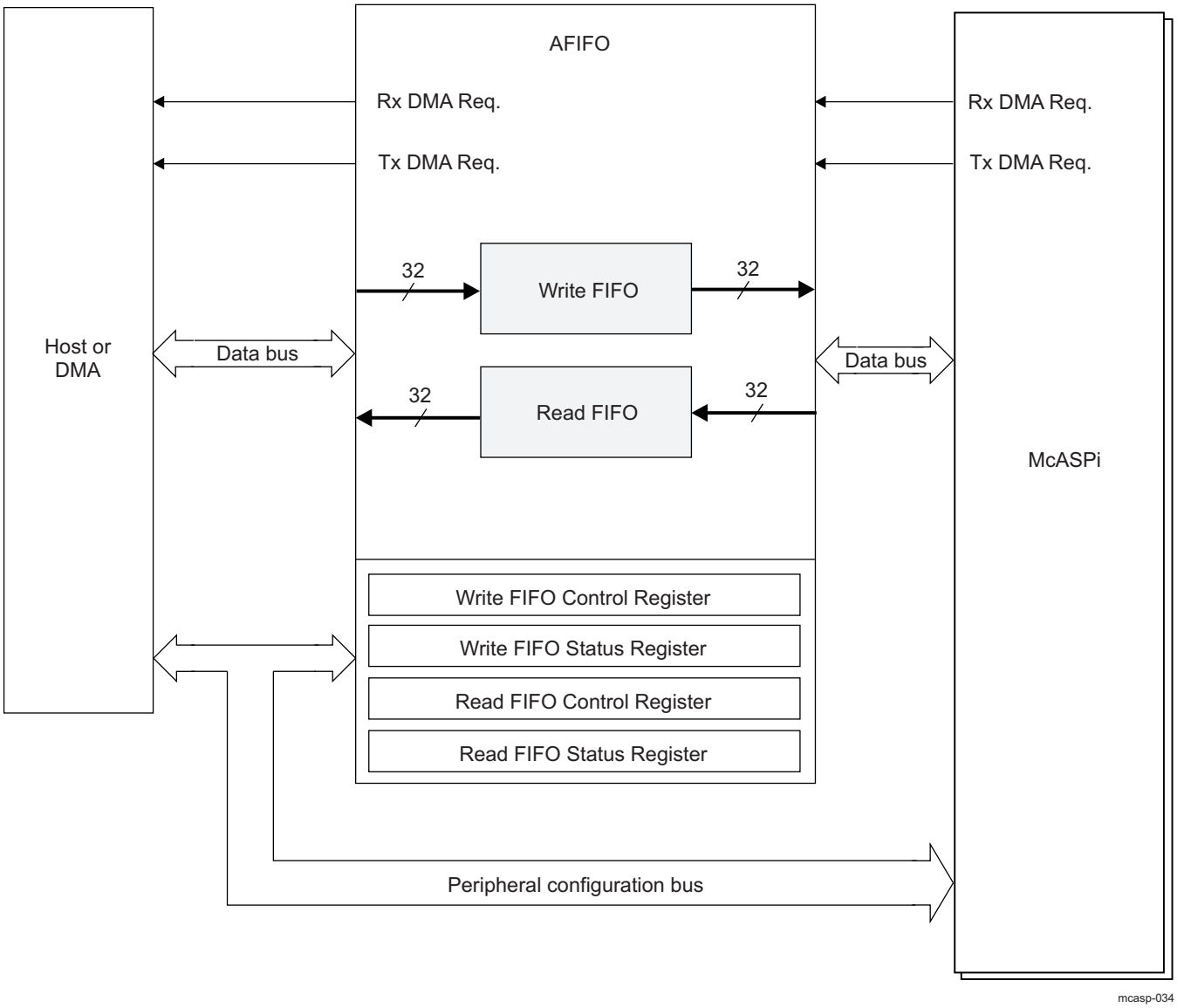
Figure 18-100 is an example of an audio system with six audio channels (LF, RF, LS, RS, C and LFE) transmitted or received through the McASP signals - AXR0, AXR1 and AXR2. It shows the points at which events AXEVT/AREVT are triggered.

In Figure 18-100, a Tx DMA event AXEVT is triggered on each time slot. In the example, AXEVT is triggered for each of the transmit audio channel time slot (time slot for channels LF, LS, and C; and time slot for channels RF, RS, LFE). Transmit DMA events are generated automatically upon transmit data ready, provided that DMA TX requests generation is enabled in the MCASP_XEVTCTL register. Similarly, Rx DMA event AREVT is triggered for each of the receive audio channel time slot. Receive DMA events are generated automatically upon receive data ready, provided that DMA RX requests generation is enabled in the MCASP_REVTCTL register.

18.5.4.11 McASP Audio FIFO (AFIFO)

The AFIFO contains two FIFOs: one Read FIFO (RFIFO), and one Write FIFO (WFIFO). The RFIFO and the WFIFO are the same size: 64 32-bit Words. To ensure backward compatibility with existing software, both the Read and Write FIFOs are disabled by default. See Figure 18-101 for a high-level block diagram of the AFIFO. The AFIFO may be enabled/disabled and configured via the WFIFOCTL and RFIFOCTL registers. Note that if the Read or Write FIFO is to be enabled, it must be enabled prior to initializing the receive/transmit section of the McASP.

Figure 18-101. McASP Audio FIFO (AFIFO) Block Diagram



18.5.4.11.1 AFIFO Data Transmission

When the Write FIFO is disabled, transmit DMA requests pass through directly from the McASP to the host/DMA controller. Whether the WFIFO is enabled or disabled, the McASP generates transmit DMA requests as needed; the AFIFO is “invisible” to the McASP. When the Write FIFO is enabled, transmit DMA requests from the McASP are sent to the AFIFO, which in turn generates transmit DMA requests to the host/DMA controller. If the Write FIFO is enabled, upon a transmit DMA request from the McASP, the WFIFO writes WNUMDMA 32-bit words to the McASP if and when there are at least WNUMDMA words in the Write FIFO. If there are not, the WFIFO waits until this condition has been satisfied. At that point, it writes WNUMDMA words to the McASP. (See description for WFIFOCTL[7:0] WNUMDMA.) If the host CPU writes to the Write FIFO, independent of a transmit DMA request, the WFIFO will accept host writes until full. After this point, excess data will be discarded. Note that when the WFIFO is first enabled, it will immediately issue a transmit DMA request to the host. This is because it begins in an empty state, and is therefore ready to accept data.

18.5.4.11.1.1 Transmit DMA Event Pacer

The AFIFO may be configured to delay making a transmit DMA request to the host until the Write FIFO has enough space for a specified number of words. In this situation, the number of transmit DMA requests to the host or DMA controller is reduced. If the Write FIFO has space to accept WNUMEVT 32-bit words, it generates a transmit DMA request to the host and then waits for a response. Once WNUMEVT words have been written to the FIFO, it checks again to see if there is space for WNUMEVT 32-bit words. If there is space, it generates another transmit DMA request to the host, and so on. In this fashion, the Write FIFO will attempt to stay filled. Note that if transmit DMA event pacing is desired, WFIFOCTL[15:8] WNUMEVT should be set to a non-zero integer multiple of the value in WFIFOCTL[7:0] WNUMDMA. If transmit DMA event pacing is not desired, then the value in WFIFOCTL[15:8] WNUMEVT should be set equal to the value in WFIFOCTL[7:0] WNUMDMA.

18.5.4.11.2 AFIFO Data Reception

When the Read FIFO is disabled, receive DMA requests pass through directly from McASP to the host/DMA controller. Whether the RFIFO is enabled or disabled, the McASP generates receive DMA requests as needed; the AFIFO is “invisible” to the McASP. When the Read FIFO is enabled, receive DMA requests from the McASP are sent to the AFIFO, which in turn generates receive DMA requests to the host/DMA controller. If the Read FIFO is enabled and the McASP makes a receive DMA request, the RFIFO reads RNUMDMA 32-bit words from the McASP, if and when the RFIFO has space for RNUMDMA words. If it does not, the RFIFO waits until this condition has been satisfied; at that point, it reads RNUMDMA words from the McASP. (See description for RFIFOCTL[7:0] RNUMDMA.) If the host CPU reads the Read FIFO, independent of a receive DMA request, and the RFIFO at that time contains less than RNUMEVT words, those words will be read correctly, emptying the FIFO.

18.5.4.11.2.1 Receive DMA Event Pacer

The AFIFO may be configured to delay making a receive DMA request to the host until the Read FIFO contains a specified number of words. In this situation, the number of receive DMA requests to the host or DMA controller is reduced. If the Read FIFO contains at least RNUMEVT 32-bit words, it generates a receive DMA request to the host and then waits for a response. Once RNUMEVT 32-bit words have been read from the RFIFO, the RFIFO checks again to see if it contains at least another RNUMEVT words. If it does, it generates another receive DMA request to the host, and so on. In this fashion, the Read FIFO will attempt to stay empty. Note that if receive DMA event pacing is desired, RFIFOCTL[15:8] RNUMEVT should be set to a non-zero integer multiple of the value in RFIFOCTL[7:0] RNUMDMA. If receive DMA event pacing is not desired, then the value in RFIFOCTL[15:8] RNUMEVT should be set equal to the value in RFIFOCTL[7:0] RNUMDMA.

18.5.4.11.3 Arbitration Between Transmit and Receive DMA Requests

If both the WFIFO and the RFIFO are enabled and a transmit DMA request and receive DMA request occur simultaneously, priority is given to the transmit DMA request. Once a transfer is in progress, it is allowed to complete. If only the WFIFO is enabled and a transmit DMA request and receive DMA request occur simultaneously, priority is given to the transmit DMA request. Once a transfer is in progress, it is allowed to complete. If only the RFIFO is enabled and a transmit DMA request and receive DMA request occur simultaneously, priority is given to the receive DMA request. Once a transfer is in progress, it is allowed to complete.

18.5.4.12 McASP Events and Interrupt Requests

Table 18-333 lists all the transmit event flags. Table 18-334 lists all the Receive event flags. Source of each of these TX/RX events can be a TX/RX channel from any McASPi serializer configured as transmitter or receiver respectivly.

Table 18-333. TX Events[1]

Event Mask	Event Flag	Map to [2]	Description
MCASP_EVTCTLX[0] XUNDRN	MCASP_TXSTAT[0] XUNDRN	MCASPi_IRQ_AXEVT	Transmit buffer underrun

(1) Global events for all transmitting serializers in a single McASPi module.
(2) Every McASPi module generates separate IRQ event.

Table 18-333. TX Events[1] (continued)

Event Mask	Event Flag	Map to [2]	Description
MCASP_EVTCTLX[1] XSYNCERR	MCASP_TXSTAT[1] XSYNCERR	MCASPi_IRQ_AXEVT	Unexpected transmit frame sync
MCASP_EVTCTLX[2] XCKFAIL	MCASP_TXSTAT[2] XCKFAIL	MCASPi_IRQ_AXEVT	Transmit clock failure
MCASP_EVTCTLX[3] XDMAERR	MCASP_TXSTAT[7] XDMAERR	MCASPi_IRQ_AXEVT	DATA port transmit error
MCASP_EVTCTLX[4] XLAST	MCASP_TXSTAT[4] XLAST	MCASPi_IRQ_AXEVT	Transmit last slot interrupt
MCASP_EVTCTLX[5] XDATA	MCASP_TXSTAT[5] XDATA	MCASPi_IRQ_AXEVT	Transmit data-ready interrupt
MCASP_EVTCTLX[7] XSTAFRM	MCASP_TXSTAT[6] XSTAFRM	MCASPi_IRQ_AXEVT	Transmit start of frame interrupt
n.a.	MCASP_TXSTAT[8] XERR	n.a.	OR-event of all Tx-error events: (XDMAERR | XCKFAIL | XUNDRN | XSYNCERR). It is cleared ONLY when all error flags are cleared
n.a.	MCASP_TXSTAT[3] XTDMSLOT	n.a.	Qualifies the current TDM slot as an odd or an even slot.

Table 18-334. RX Events[1]

Event Mask	Event Flag	Map to[2]	Description
MCASP_EVTCTLR[0] ROVRN	MCASP_RXSTAT[0] ROVRN	MCASPi_IRQ_AREVT	Receive buffer overrun
MCASP_EVTCTLR[1] RSYNCERR	MCASP_RXSTAT[1] RSYNCERR	MCASPi_IRQ_AREVT	Unexpected receive frame sync
MCASP_EVTCTLR[2] RCKFAIL	MCASP_RXSTAT[2] RCKFAIL	MCASPi_IRQ_AREVT	Receive clock failure
MCASP_EVTCTLR[3] RDMAERR	MCASP_RXSTAT[7] RDMAERR	MCASPi_IRQ_AREVT	DATA port receive error
MCASP_EVTCTLR[4] RLAST	MCASP_RXSTAT[4] RLAST	MCASPi_IRQ_AREVT	Receive last slot
MCASP_EVTCTLR[5] RDATA	MCASP_RXSTAT[5] RDATA	MCASPi_IRQ_AREVT	Receive data-ready
MCASP_EVTCTLR[7] RSTAFRM	MCASP_RXSTAT[6] RSTAFRM	MCASPi_IRQ_AREVT	Receive start of frame
n.a.	MCASP_RXSTAT[8] RERR	n.a.	OR-event of all Rx-error events: (RDMAERR | RCKFAIL | ROVRN | RSYNCERR). RERR event is cleared once all error flags are cleared.
n.a.	MCASP_RXSTAT[3] RTDMSLOT	n.a.	Qualifies the current TDM slot as an odd or an even slot.

(1) Global events for all receiving serializers in a single McASPi module. These events and masks are available in same format for every McASPi module

(2) Every McASP module generates separate IRQ event.

Software has to read the MCASP_TXSTAT/MCASP_RXSTAT register to determine which event occurs at a global level for McASP Tx/Rx logic. In addition user software has to scan the XRDY/RRDY read-only flags in the MCASP_XRSRCTLn registers to determine which active serializer is the actual source of the event.

A Tx interrupt line (MCASPi_IRQ_AXEVT) is asserted (active high) when one of the MCASP_TXSTAT notified events occurs, provided that it is enabled in its corresponding MCASP_EVTCTLX bit. Similarly, a Rx interrupt line (MCASPi_IRQ_AREVT) is asserted (active high) when one of MCASP_RXSTAT notified events occurs, provided that it is enabled in its corresponding MCASP_EVTCTLR bit. See also Section 18.5.4.12.4, *Multiple Interrupts* and the Section 18.5.4.10.1, *Data Ready Status and Event/Interrupt Generation*.

18.5.4.12.1 Transmit Data Ready Event and Interrupt

The transmit data-ready interrupt (XDATA) is generated if XDATA is 1 in the MCASP_TXSTAT register and XDATA is enabled in MCASP_EVTCTLX. The Section 18.5.4.10.1, *Data Ready Status and Event/Interrupt Generation*, provides details on when XDATA is set in the MCASP_TXSTAT register.

A transmit-start-of-frame interrupt (XSTAFRM) is triggered by the recognition of a transmit frame sync.

A transmit-last-slot interrupt (XLAST) is a qualified version of the data-ready interrupt (XDATA). It has the same behavior than the data-ready interrupt, but is further qualified by having the data requested belonging to the last slot (the slot that just ended is the next-to-last TDM slot, the current slot is the last slot).

18.5.4.12.2 Receive Data Ready Event and Interrupt

The receive data-ready interrupt (RDATA) is generated if RDATA is 1 in the MCASP_RXSTAT register and RDATA is enabled in MCASP_EVTCTLR. The Section 18.5.4.10.1, *Data Ready Status and Event/Interrupt Generation*, provides details on when RDATA flag is set in the MCASP_RXSTAT register.

A receiver start of frame (RSTAFRM) interrupt is triggered by the recognition of a receiver frame sync.

A receiver last slot (RLAST) interrupt is a qualified version of the data ready interrupt (RDATA). It has the same behavior as the data ready interrupt, but is further qualified by having the data in the buffer come from the last TDM time slot (the slot that just ended was last TDM slot).

18.5.4.12.3 Error Interrupt

Upon detection, the following error conditions generate interrupt flags:

In the transmit status register (MCASP_TXSTAT):

- Transmit underrun (XUNDRN)
- Unexpected transmit frame sync (XSYNCERR)
- Transmit clock failure (XCKFAIL)
- Transmit DATA port error (XDMAERR)

Each interrupt source also has a corresponding enable bit in the transmit interrupt control register (MCASP_EVTCTLX). If the enable bit is set, an interrupt is requested when the interrupt flag is set in MCASP_TXSTAT. If the enable bit is not set, no interrupt request is generated. However, the interrupt flag may be polled.

In the receive status register (MCASP_RXSTAT) :

- Receiver overrun (ROVRN)
- Unexpected receive frame sync (RSYNCERR)
- Receive clock failure (RCKFAIL)
- Receive DATA port error (RDMAERR)

Each interrupt source also has a corresponding enable bit in the receive interrupt control register (MCASP_EVTCTLR). If the enable bit is set, an interrupt is requested when the interrupt flag is set in MCASP_RXSTAT. If the enable bit is not set, no interrupt request is generated. However, the interrupt flag may be polled.

18.5.4.12.4 Multiple Interrupts

This only applies to interrupts and not to DMA requests. The following terms are defined:

- **Active Interrupt Request:** a flag in MCASP_TXSTAT is set and the interrupt is enabled in MCASP_EVTCTLX.
- **Outstanding Interrupt Request:** An interrupt request has been issued on one of the McASP transmit interrupt port, but that request has not yet been serviced.
- **Serviced:** The CPUs write to MCASP_TXSTAT to clear one or more of the active interrupt request flags.

The first interrupt request to become active for the serializer with the interrupt flag set in MCASP_TXSTAT/MCASP_RXSTAT and the interrupt enabled in MCASP_EVTCTLX/MCASP_EVTCTLR generates a request on the McASP transmit or receive interrupt port.

If more than one interrupt request becomes active in the same cycle, a single interrupt request is generated on the McASP transmit or receive interrupt port. Subsequent interrupt requests that become active while the first interrupt request is outstanding do not immediately generate a new request pulse on the McASP transmit or receive interrupt port.

The interrupt is serviced with the CPU writing to MCASP_TXSTAT/MCASP_RXSTAT. If any interrupt requests are active after the write, a new request is generated on the McASP transmit or receive interrupt port.

One outstanding interrupt request is allowed on each port, so a transmit and a receive interrupt request may both be outstanding at the same time.

18.5.4.13 DMA Requests

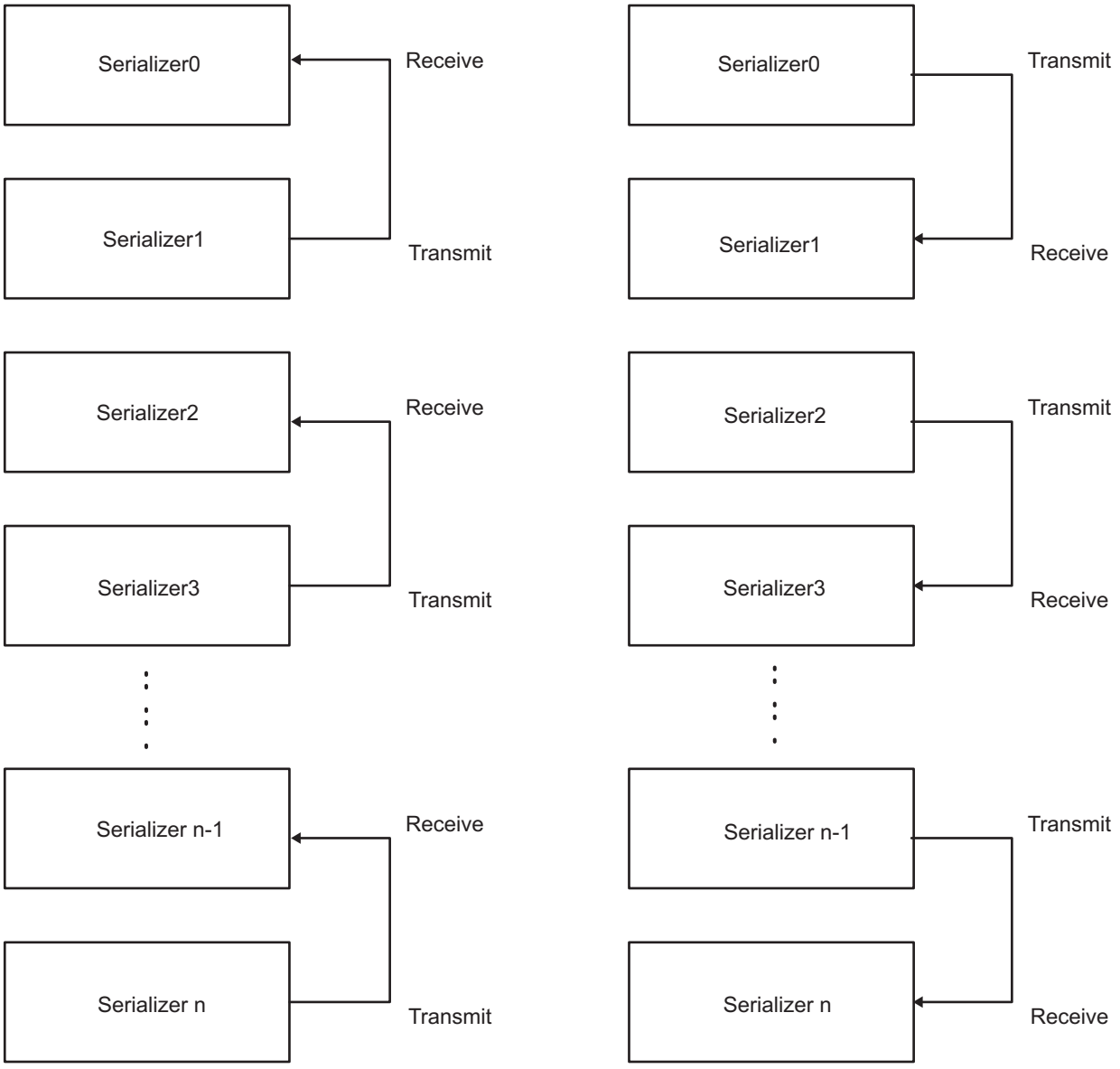
The McASP can generate one DMA request to the DMA_CROSSBAR to transmit (MCASPi_DREQ_TX) or receive (MCASPi_DREQ_RX) data. A DMA request to transmit data is generated if the XDATDMA bit in the MCASP_XEVTCTL register is cleared. A DMA request to receive data is generated if the RDATDMA bit in the MCASP_REVTCTL register is cleared.

18.5.4.14 Loopback Modes

The McASP features a digital loopback mode (DLB) that allows loopback test transfers in TDM mode between McASP transmitters and receivers within the same device. In loopback mode, the output of a transmit serializer is connected internally to the input of a receive serializer. Therefore, a receiver data can be checked against a transmitter data to ensure that the McASP settings are correct. Digital loopback mode applies to TDM mode only (2 to 32 slots in a frame). It does not apply to DIT mode (XMOD = 0x180) or burst mode (XMOD = 0).

Figure 18-102shows the basic logical connection of the serializers in loopback mode.

Figure 18-102. McASP Serializers Operation in Loopback Mode



Two types of loopback connections are possible, selected by the ORD bit in the digital loopback control register - MCASP_LBCTL as follows:

- ORD = 0: Outputs of odd serializers are connected to inputs of even serializers. If this mode is selected, the odd serializers must be configured as transmitters and even serializers as receivers.
- ORD = 1: Outputs of even serializers are connected to inputs of odd serializers. If this mode is selected, the even serializers must be configured as transmitters and odd serializers as receivers.

User can choose in software (bit IOLBEN of the MCASP_LBCTL) between a McASP module internal loopback and a device I/O level loopback.

When a **McASP internal loopback** is selected (MCASP_LBCTL[4] IOLBEN=0b0), it is NOT necessary to configure MCASP_PFUNC and MCASP_PDIR registers for McASP pin settings. Nevertheless, data can be optionally made externally visible at the I/O pin of the transmit serializer, if the pin is configured as a McASP output pin by setting the corresponding MCASP_PFUNC bit to 0 (i.e. to function as McASP, not GPIO) and MCASP_PDIR bit to 1 (output).

When a **device I/O level loopback** is selected (MCASP_LBCTL[4] IOLBEN=0b1), the MCASP_PFUNC and MCASP_PDIR registers must be configured with the appropriate settings for all AXRn pins, according to ORD bit configuration.

In case of device I/O loopback, the connectivity is externally applied between device pads (i.e. reaching device I/O buffers).

Hence, the corresponding padconfiguration registers must be appropriately configured in the device Control Module - CTRL_MODULE_CORE_PAD. For more details, see Section 13.4.6.1, *Pad Configuration Rregisters*, in *Chapter 13, Control Module*.

When In loopback mode, the transmit clock and frame sync are used by both the transmit and receive sections of the McASP. The transmit and receive sections operate synchronously. This is achieved by setting the MODE bitfield of the MCASP_LBCTL register to 0x1 and the ASYNC bit of the MCASP_ACLKXCTL register to 0b0.

18.5.4.14.1 Loopback Mode Configurations

This is a summary of the settings required for digital loopback mode for TDM format :

- The MCASP_LBCTL[0] DLBEN bit must be set to 0b1 to enable a loopback mode. It must be kept at 0b0 during normal McASP operation.
- The MCASP_LBCTL[4] IOLBEN bit must be set to select between internal (McASP local) loopback mode or device I/O level loopback mode.
- The MCASP_LBCTL[3:2] MODE bitfield must be set to 0x1 for both the transmit and receive sections to use the transmit clock and frame sync generator.
- The MCASP_LBCTL[1] ORD must be programmed appropriately to select odd or even serializers to be transmitters or receivers.
- The corresponding serializers must be configured accordingly.
- The bit - MCASP_ACLKXCTL[6] ASYNC must be cleared to 0b0 to ensure synchronous transmit and receive operations.
- The bitfields - MCASP_RXFMCTL[15:7] RMOD and MCASP_TXFMCTL[15:7] XMOD must be set within range (0x2- 0x20) to indicate TDM mode.

NOTE: Loopback mode does not apply to DIT or burst mode, because McASP receivers do NOT natively support DIR - reception.

18.5.4.15 Error Reporting

The McASP includes error-checking capability for the serial protocol and data underrun. In addition, the McASP includes a timer that continually measures the high-frequency master clock every 32 AHCLKX clock cycles. The value of the timer can be read to get a measurement of the clock frequency and has a minimum and maximum range setting that can set an error flag if the master clock goes out of a specified range.

When one or more errors (software selectable) are detected, an interrupt can be generated if desired, based on one or more error sources.

18.5.4.15.1 Buffer Underrun Error -Transmitter

A buffer underrun occurs when a serializer is instructed by the transmit state-machine to transfer data from XRBUFn buffer to XRSRn shift register, but the corresponding (MCASP_TXBUFn) register has not yet been written with new data since the last time the transfer occurred. When this occurs, the transmit state-machine sets the XUNDRN flag.

An underrun is checked only once per time slot. The MCASP_TXSTAT[0] XUNDRN flag is set when an underrun condition occurs. Once set, the XUNDRN flag remains set until the host explicitly writes 1 to the XUNDRN bit to clear it.

In DIT mode, a pair of BMC zeros is shifted out when an underrun occurs (four bit times at 128 bfs). By shifting out a pair of zeros, a clock can be recovered on the receiver. To recover, reset the McASP and restart with the proper initialization.

In TDM mode, during an underrun case, a long stream of zeros are shifted out causing the DACs to mute. To recover, reset the McASP and start again with the proper initialization.

18.5.4.15.2 Buffer Overrun Error-Receiver

A buffer overrun occurs when a serializer is instructed to transfer data from XRSRn shift register to XRBUFn receiver buffer, but the corresponding MCASP_RXBUFn register has not yet been read since the last time the transfer occured. When this occurs, the receiver state machine sets the overrun flag - ROVRN. However, the individual serializer writes over the data in the XRBUFn buffer register (destroying the previous sample) and continues shifting.

An overrun is checked only once per time slot. The MCASP_RXSTAT[0] ROVRN flag is set when an overrun condition occurs. It is possible that an overrun occurs on one time slot but then the host catches up and does not cause an overrun on the following time slots. However, once the ROVRN flag is set, it remains set until the host explicitly writes a 1 to the ROVRN bit to clear the ROVRN bit.

18.5.4.15.3 DATA Port Error - Transmitter

A transmit DATA port error, as indicated by the XDMAERR flag in the MCASP_TXSTAT register, occurs when the DMA or device CPU writes more words to the DATA port of the McASP than it should.

The MCASP_TXSTAT[7] XDMAERR=0b1 indicates that the DMA or device CPU wrote too many words to the McASP DATA port for a given transmit DMA event. Writing too few words results in a transmit underrun error setting XUNDRN in MCASP_TXSTAT.

While XDMAERR occurs infrequently, an occurrence indicates a serious loss of synchronization between the McASP and the DMA or device CPU. The McASP transmitter and the DMA must be reinitialized to resynchronize them.

18.5.4.15.4 DATA Port Error - Receiver

A receive DATA port error, as indicated by the RDMAERR flag in the MCASP_RXSTAT register, occurs when the DMA or device CPU reads more words from the DATA port of the McASP than it should.

The MCASP_RXSTAT[7] RDMAERR indicates that the DMA or device CPU read too many words from the McASP DATA port for a given receive AREVT event. Reading too few words results in a receiver overrun error setting ROVRN in MCASP_RXSTAT.

While RDMAERR occurs infrequently, an occurrence indicates a serious loss of synchronization between the McASP and the DMA or device CPU. The McASP receiver and the DMA must be reinitialized to resynchronize them.

18.5.4.15.5 Unexpected Frame Sync Error

An unexpected frame sync occurs in when:

- in burst mode and TDM mode, the next active edge of the frame sync occurs early such that the current slot will not be completed by the time the next slot is scheduled to begin.
- in TDM mode, an unexpected frame sync occurs also if the frame sync does NOT occur exactly during the correct bit clock (not a cycle earlier or later) and before slot 0.

When an unexpected frame sync occurs, there are two possible actions depending upon when the unexpected frame sync occurs:

1. **Early:** An early unexpected frame sync occurs when the McASP is in the process of completing the current frame and a new frame sync is detected (not including overlap that occurs due to a 1 or 2 bit frame sync delay). When an early unexpected frame sync occurs:

- Error event flag is set (XSYNCERR, if an unexpected transmit frame sync occurs; RSYNCERR, if an unexpected receive frame sync occurs).
- Current frame is not resynchronized. The number of bits in the current frame is completed. The next frame sync, which occurs after the current frame is completed, will be resynchronized.

2. **Late:** A late unexpected frame sync occurs when there is a gap or delay between the last bit of the previous frame and the first bit of the next frame. When a late unexpected frame sync occurs (as soon as the gap is detected):
   - Error event flag is set (XSYNCERR, if an unexpected transmit frame sync occurs; RSYNCERR, if an unexpected receive frame sync occurs).
   - Resynchronization occurs upon the arrival of the next frame sync.

Late frame sync is detected the same way in burst mode and TDM mode. However, in burst mode, late frame sync is not meaningful and its interrupt enable should not be set.

18.5.4.15.6 Clock Failure Detection

18.5.4.15.6.1 Clock Failure Check Startup

It is initially expected of the clock failure circuits to generate an error until at least one measurement is taken. Therefore, the clock failure interrupts, clock switch, and mute functions should not be enabled immediately, but only after a specific startup procedure.

To start the transmit clock failure check procedure:

1. Configure the transmit clock failure detect logic (XMIN, XMAX, XPS) in the transmit clock check control register (MCASP_TXCLKCHK).
2. Clear the transmit clock failure flag (XCKFAIL) in the transmit status register (MCASP_TXSTAT).
3. Wait until the first measurement is taken (> 32 AHCLKX clock periods).
4. Verify that no clock failure is detected.
5. Repeat Step 2 through Step 4 until the clock is running and is no longer issuing clock failure errors.
6. After the transmit clock is measured and falls within the acceptable range, the following can be enabled:
   1. The transmit clock failure interrupt enable bit (XCKFAIL) in the transmitter interrupt control register (MCASP_EVTCTLX)

To start the receive clock failure check procedure:

1. Configure receive clock failure detect logic (RMIN, RMAX, RPS) in the receive clock check control register (MCASP_RXCLKCHK).
2. Clear receive clock failure flag (RCKFAIL) in the receive status register (MCASP_RXSTAT).
3. Wait until first measurement is taken (> 32 AHCLKR clock periods).
4. Verify no clock failure is detected.
5. Repeat steps 2–4 until clock is running and is no longer issuing clock failure errors.
6. After the receive clock is measured and falls within the acceptable range, the following may be enabled:
   1. the receive clock failure (RCKFAIL) interrupt enable bit in the receive interrupt control register (MCASP_EVTCTLR)

18.5.4.15.6.2 Transmit Clock Failure Check and Recovery

The transmit clock failure check circuit (see Figure 18-103) works off the internal McASP interface clock and the external high-frequency serial clock (AHCLKX). It continually counts the number of interface clocks for every 32 high-rate serial clock (AHCLKX) periods, and stores the count in XCNT of the transmit clock check control register (MCASP_TXCLKCHK) every 32 high-rate serial clock cycles.

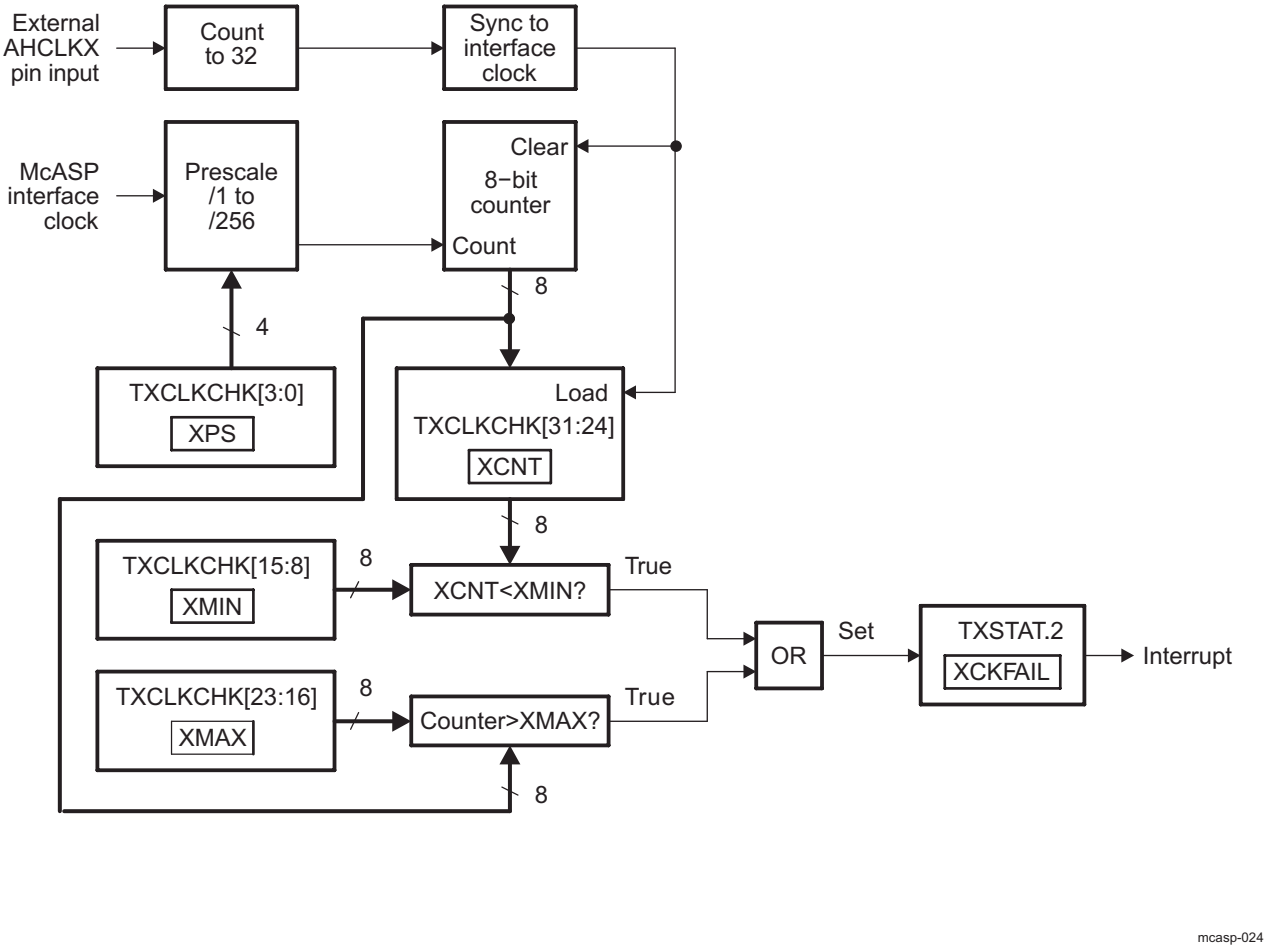
The logic compares the count against a user-defined minimum allowable boundary (XMIN), and automatically flags an interrupt (XCKFAIL in MCASP_TXSTAT) when an out-of-range condition occurs. An out-of-range minimum condition occurs when the count is less than XMIN. The logic continually compares the current count (from the running interface clock counter) to the maximum allowable boundary (XMAX). This is so that if the external clock completely stops, the counter value is not copied to XCNT. An out-of-range maximum condition occurs when the count is greater than XMAX. The XMIN and XMAX fields are 8-bit unsigned values, and the comparison is performed using unsigned arithmetic.

An out-of-range count may indicate that an unstable clock was detected or that the audio source has changed and a new sample rate is being used.

For the transmit clock failure check circuit to operate correctly, the high-frequency serial clock divider must be taken out of reset.

If a clock failure is detected, the transmit clock failure flag (XCKFAIL) in MCASP_TXSTAT is set. This causes an interrupt if the transmit clock failure interrupt enable bit (XCKFAIL) in MCASP_EVTCTLX is set.

Figure 18-103. Transmit Clock Failure Detection Circuit Block Diagram



18.5.4.15.6.3 Receive Clock Failure Check and Recovery

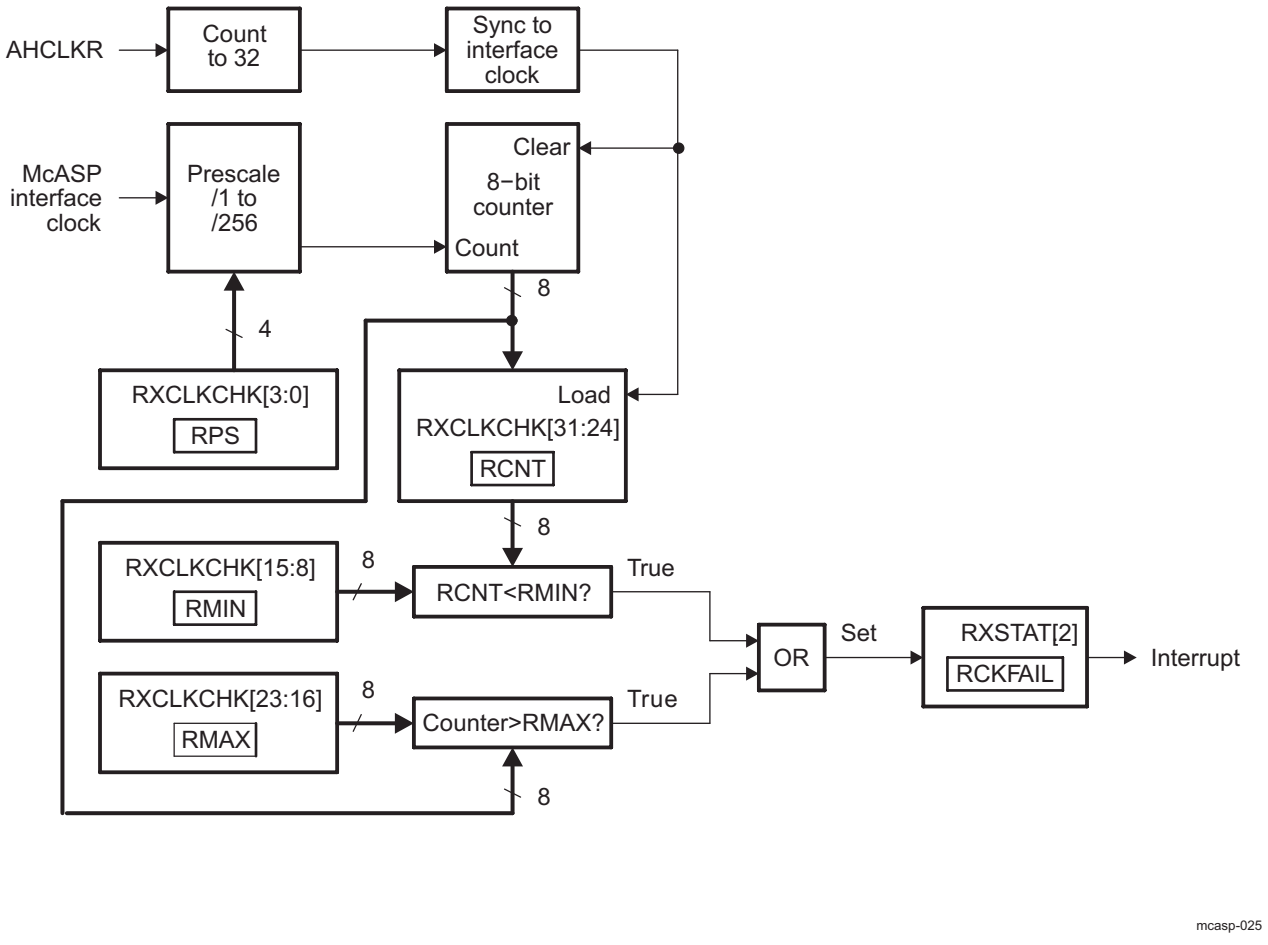
The receive clock failure check circuit (see Figure 18-104) works off both the internal McASP interface clock and the high-frequency serial clock (AHCLKR) coming from the device clock generator. It continually counts the number of interface clocks for every 32 high rate serial clock (AHCLKR) periods, and stores the count in RCNT of the receive clock check control register (MCASP_RXCLKCHK) every 32 high rate serial clock cycles.

The logic compares the count against a user-defined minimum allowable boundary (RMIN) and automatically flags an event (RCKFAIL in MCASP_RXSTAT) when an out-of-range condition occurs. An out-of-range minimum condition occurs when the count is smaller than RMIN. The logic continually compares the current count (from the running interface clock counter) against the maximum allowable boundary (RMAX). This is in case the external clock completely stops, so that the counter value is not copied to RCNT. An out-of-range maximum condition occurs when the count is greater than RMAX. Note that the RMIN and RMAX fields are 8-bit unsigned values, and the comparison is performed using unsigned arithmetic.

An out-of-range count may indicate either that an unstable clock was detected or that the audio source has changed and a new sample rate is being used.

In order for the receive clock failure check circuit to operate correctly, the high-frequency serial clock divider must be taken out of reset.

Figure 18-104. Receive Clock Failure Detection Circuit Block Diagram



18.5.5 McASP Low-Level Programming Model

This section describes the low-level hardware programming sequences for the configuration and use of the McASP module.

18.5.5.1 Global Initialization

18.5.5.1.1 Surrounding Modules Global Initialization

This section identifies the requirements for initializing the surrounding modules when the McASP module is used for the first time after a device reset. This initialization of surrounding modules is based on the integration and environment of the McASP (for more information, see Section 18.5.3, *McASP Integration*, and Section 18.5.2, *McASP Environment*).

Table 18-335, describes the global initialization of surrounding modules.

Table 18-335. Global Initialization of Surrounding Modules

Surrounding Modules	Comments
PRCM	Module functional and interface clocks must be enabled. (See Section 3.6, *Clock Management Functional Description*, in Chapter 3, *Power, Reset, and Clock Management*.)
Control module	Module-specific pad muxing and other pad configurations must be set in the control module. (See Section 13.4.6.1, *Pad Configuration Registers*, in Chapter 13, *Control Module*).
(Optional) IRQ_CROSSBAR	Interrupt crossbar configuration must be done to enable the interrupts from the McASP. For more details on IRQ_CROSSBAR module, see Section 13.4.6.4, *IRQ_CROSSBAR Module Functional Description*, in Chapter 13, *Control Module*.
(Optional) DMA_CROSSBAR	DMA configuration must be done to enable the McASP DMA data channel requests. For more information on DMA_CROSSBAR module configuration, see Section 13.4.6.5, *DMA_CROSSBAR Module Functional Description*, in Chapter 13, *Control Module*.
(Optional) L4_PER2 and L3_MAIN Interconnects	For more information about the interconnect configuration, see Section 9.2.1, *L3_MAIN Interconnect Overview* in Section 9.2, *L3 Interconnect*.

NOTE: The IRQ_CROSSBAR and the DMA_CROSSBAR configurations are required when the interrupt and DMA-based communication modes are used. Further initialization of the selected IRQ and DMA controllers of the host CPU must be done for full functionality of the McASP DMA and IRQ lines.

18.5.5.1.2 McASP Global Initialization

18.5.5.1.2.1 Main Sequence – McASP Global Initialization for DIT-Transmission

The procedure in Table 18-336 initializes the McASP serializers transmitters to operate in DIT-mode (S/PDIF-transmission protocol) after a power-on reset (POR).

CAUTION

Before performing McASP global initialization, If external clock ACLKR is used, it must be running already for proper synchronization of the MCASP_GBLCTL register.

Table 18-336. McASP Transmitters Global Initialization for DIT-Mode Operation

Step	Register/Bit Field/Programming Model	Value
1. Apply software reset to different McASP components.	MCASP_GBLCTL[12:8]	0x00
2. Poll the bits to ensure the active reset value (0x00) is successfully latched into the register.	MCASP_GBLCTL[12:8]	=0x00
3. Configure the local power management.	PWRIDLESYSCONFIG[1:0] IDLE_MODE	0x1
4. Configure the transmit format unit.	See Section 18.5.5.1.2.1.1.	
5. Configure the transmit frame sync generator.	See Section 18.5.5.1.2.1.2.	
6. Configure the transmit clock generator.	See Section 18.5.5.1.2.1.3.	
7. Configure the TDM sequencer—set all slots active.	MCASP_TXTDM[31:0] XTDMS	0xFFFF FFFF
8. Configure the desired n-th serializer (n=0 to 3) for transmit mode operation. (1)	MCASP_XRSRCTLn [1:0] SRMOD	0x1
9. Configure the McASP pins functionality.	See Section 18.5.5.1.2.1.4.	
10. Enable the McASP DIT - transmission mode.	MCASP_TXDITCTL[0] DITEN	0x1 (2)
11. Configure DIT-specific subframe fields.	See Table 18-341.	
12. Release from reset state the divider that outputs the AHCLKX clock. (3)	MCASP_GBLCTL[9] XHCLKRST	0x1
13. Poll the bit to ensure that it is successfully latched in the register.	MCASP_GBLCTL[9] XHCLKRST	=0x1
14. Release from reset state the divider that outputs the ACLKX clock. (3)	MCASP_GBLCTL[8] XCLKRST	0x1
15. Poll the bit to ensure that it is successfully latched in the register.	MCASP_GBLCTL[8] XCLKRST	=0x1

(1) For an unused serializer n, write MCASP_XRSRCTLn [1:0] SRMOD=0x0 to disable it.
(2) This globally configures all active transmitters to operate in DIT-mode.
(3) During reset state the local McASP internal clock dividers maintain a 1:1 ratio at their outputs. The values stored in the MCASP_AHCLKXCTL and MCASP_ACLKXCT registers are ignored; hence, the transmission clock does not stop during the reset state of the dividers.

18.5.5.1.2.1.1 Subsequence – Transmit Format Unit Configuration for DIT-Transmission

The procedure in Table 18-337 configures the transmit frame format unit of the McASP module for a DIT-transmission.

NOTE:

- The first transmit data bit always has a 0-bit delay.
- The bitstream is always transmitted in least-significant-bit (LSB)-first order.
- Pad value for extra bits in a certain slot is always 0.

Table 18-337. Transmit Format Unit Configuration for DIT-Transmission

Step	Register/Bit Field/Programming Model	Value
Configure the slot size to 32 bits.	MCASP_TXFMT[7:4] XSSZ	0xF
IF: the data to transmit is left- aligned	Software test condition	
Set data mask in the range 0xFFFF FF00 – 0xFFFF 0000.	MCASP_TXMASK[31:0] XMASK	0x- (1)
Rotate data right by a multiple-of-4- bit positions.	MCASP_TXFMT[2:0] XROT	0x- (1)
ELSE		
Set data mask in the range 0x00FF FFFF– 0x0000 FFFF.	MCASP_TXMASK[31:0] XMASK	0x- (1)
Rotate data right by 0-bit positions.	MCASP_TXFMT[2:0] XROT	0x0
ENDIF		

(1) Refer to Section 18.5.4.4.1, *Transmit Fromat Unit* and Section 18.5.4.4.1.2, *DIT-Mode Transmission Data Alignment Settings.*

Table 18-337. Transmit Format Unit Configuration for DIT-Transmission (continued)

Step	Register/Bit Field/Programming Model	Value
Select to write data to active serializers transmit buffers using peripheral (CFG) or DATA port	MCASP_TXFMT[3] XBUSEL	0x-

18.5.5.1.2.1.2 Subsequence – Transmit Frame Synchronization Generator Configuration for DIT-Transmission

The procedure in Table 18-338 configures the transmit frame synchronization generator of the McASP module.

NOTE: The frame synchronization signal is always rising-edge active and always has a single-bit width.

Table 18-338. Transmit Frame-Synchronization Generator Configuration for DIT-Transmission

Step	Register/Bit Field/Programming Model	Value
Select 384-slot size block.	MCASP_TXFMCTL[15:7] XMOD	0x180
Select internally-generated transmit frame sync.	MCASP_TXFMCTL[1] FSXM	0x1

18.5.5.1.2.1.3 Subsequence – Transmit Clock Generator Configuration for DIT-Transmission

NOTE: By default, the ACLKX and AHCLKX clocks are generated only from the McASP internal clock source.

The procedure in Table 18-339 configures the transmit clock generator of the McASP module.

Table 18-339. Transmit Clock Generator Configuration in DIT-Mode

Step	Register/Bit Field/Programming Model	Value
Set the divisor for the internally generated high frequency clock– AHCLKX.	MCASP_AHCLKXCTL[11:0] HCLKXDIV	0x-
Set he divisor for the internally generated transmission clock– ACLKX.	MCASP_ACLKXCTL[4:0] CLKXDIV	0x-
Configure the transmit clock failure detect logic.	See Section 18.5.4.15.6.1, *Clock Failure Check Startup*.	

18.5.5.1.2.1.4 Subsequence - McASP Pins Functional Configuration

The procedure in Table 18-340 configures the McASP pins for McASP functionality.

Table 18-340. McASP Pins Functional Configuration

Step	Register/Bit Field/Programming Model	Value
Configure module different pins to have McASP functionality.	MCASP_PFUNC[31:0]	0x0
Configure the McASP pins as outputs: AFSX AHCLKX ACLKX Desired i-th McASP data pin AXRi is configured as an output for DIT-transmission.	MCASP_PDIR[28] AFSX; MCASP_PDIR[27] AHCLKX; MCASP_PDIR[26] ACLKX; MCASP_PDIR [i] AXRi	0x1 0x1 0x1 0x1

18.5.5.1.2.1.5 Subsequence – DIT-specific Subframe Fields Configuration

The procedure in Table 18-341 configures the DIT-specific subframe fields as part of the S/PDIF format data.

Table 18-341. DIT-Specific Subframe Fields Configuration

Step	Register/Bit Field/Programming Model	Value
Configure the valid bit value for odd time slots.	MCASP_TXDITCTL[3] VB	0x-
Configure the valid bit value for even time slots.	MCASP_TXDITCTL[2] VA	0x-
Configure the user data bit for each subframe A and B in a 384-slot S/PDIF block.	MCASP_DITUDRAi[31:0] DITUDRAi, where i = 0 to 5 MCASP_DITUDRBi[31:0] DITUDRBi, where i = 0 to 5	0x- 0x-
Configure the channel status bit for each subframe A and B in a 384-slot S/PDIF block.	MCASP_DITCSRAi[31:0], where i = 0 to 5 MCASP_DITCSRBi[31:0], where i = 0 to 5	0x- 0x-

18.5.5.1.2.2 Main Sequence – McASP Global Initialization for TDM-Reception

The procedure in Table 18-342 initializes a McASP serializer n receiver(s) to operate in TDM-mode (the only mode supported by McASP receivers) after a power-on reset (POR). This is used for I2S (2-slot TDM) and other TDM-based audio protocols reception.

CAUTION

Before performing McASP global initialization, If external clock ACLKR is used, it must be running already for proper synchronization of the MCASP_GBLCTL register.

NOTE: The McASP receivers support only TDM-frames (including 384-TDM frames) reception. DIT-frames reception (i.e. S/PDIF stream) can be implemented indirectly via an external DIR-chip converter with DIT-input and TDM (I2S)-compatible output connected to device McASP receiver input (TDM-only compatible).

Table 18-342. McASP Receivers Global Initialization for TDM-Mode Operation

Step	Register/Bit Field/Programming Model	Value
1. Apply software reset to different McASP receive components.	MCASP_GBLCTL[4:0]	0x00
2. Poll the bits to ensure the active reset value (0x00) is successfully latched into the register.	MCASP_GBLCTL[4:0]	=0x00
3. Configure the local power management.	PWRIDLESYSCONFIG[1:0] IDLE_MODE	0x1
4. Configure the receive format unit.	See Section 18.5.5.1.2.2.1.	
5. Configure the receive frame sync generator.	See Section 18.5.5.1.2.2.2.	
6. Configure the receive clock generator.	See Section 18.5.5.1.2.2.3.	
7. Program all bits -RTDMSk (where k=0 to 31) according to the time slot characteristics desired (positions of active versus inactive slots within a frame).	MCASP_RXTDM [k] RTDMSk , where k=0 to 31	0x-
8. Configure the desired n-th serializer for receive mode operation. (1)	MCASP_XRSRCTLn [1:0] SRMOD	0x2
9. Configure the McASP pins functionality.	See Section 18.5.5.1.2.2.4.	

(1) For an unused serializer n, write MCASP_XRSRCTLn [1:0] SRMOD=0x0 to disable it.

Table 18-342. McASP Receivers Global Initialization for TDM-Mode Operation (continued)

Step	Register/Bit Field/Programming Model	Value
10. Optional: Configure a McASP Rx channel for a loopback operation (TDM mode only) in MCASP_LBCTL [31:0].	See Section 18.5.4.14.1, *Loopback Mode Configurations*.	0x-(2)
11. Release from reset state the divider that outputs the AHCLKR clock.(3) See also (4).	MCASP_GBLCTL[1] RHCLKRST	0x1
12. Poll the bit to ensure that it is successfully latched in the register. See also (4).	MCASP_GBLCTL[1] RHCLKRST	=0x1
13. Release from reset state the divider that outputs the ACLKR clock. (3) See also (5).	MCASP_GBLCTL[0] RCLKRST	0x1
14. Poll the bit to ensure that it is successfully latched in the register. See also (5).	MCASP_GBLCTL[0] RCLKRST	=0x1

(2) In this case the receiver clock and frame sync are derived from the McASP transmitter logic, so MCASP_ACLKXCTL[6] ASYNC must be set to 0b0. Neither McASP internal receiver clock and frame sync generators, nor external clock and frame sync source are used.

(3) During reset state the local McASP internal clock dividers maintain a 1:1 ratio at their outputs. The values stored in the MCASP_AHCLKRCTL and MCASP_ACLKRCTL registers are ignored; hence, the reception clock does not stop during the reset state of the dividers.

(4) This step is necessary even if external high-frequency serial clocks are used.

(5) This step can be skipped if external serial clocks are used and they are running.

18.5.5.1.2.2.1 Subsequence – Receive Format Unit Configuration in TDM Mode

The procedure in Table 18-343 configures the receive frame format unit of the McASP module for TDM slots reception.

Table 18-343. Receive Format Unit Configuration for TDM-Reception

Step	Register/Bit Field/Programming Model	Value
Configure the desired TDM-slot size	MCASP_RXFMT[7:4] RSSZ	0x-(1)
Set data mask out value (0x0000 0000 - 0xFFFF FFFF).	MCASP_RXMASK[31:0] RMASK	0x-(2)
Select a padding value for masked-out bits.	MCASP_RXFMT[14:13] RPAD	0x-
Specify position (0x0-0x1F) of the bit in corresponding register MCASP_RXBUFn which value to be used as a pad value in case MCASP_RXFMT[14:13] RPAD=0x2.	MCASP_RXFMT[12:8] RPBIT	0x-
Rotate data right by a multiple of 4- bit positions.	MCASP_RXFMT[2:0] RROT	0x- (3)
Received stream bit order (LSB- or MSB-first). Must be set to 0x1 for an I2S stream reception (MSB-first).	MCASP_RXFMT[15] RRVRS	0x- (3)
Specify a delay between frame sync and first bit of data in number of bits. Must be set to 0x1 for an I2S stream reception.	MCASP_RXFMT[17:16] RDATDLY	0x-
Select to read data from active serializers receive buffers using peripheral (CFG) or DATA port	MCASP_RXFMT[3] RBUSEL	0x-

(1) Refer to Section 18.5.4.4.2, *Receive Format Unit*, regarding options for received TDM-slot sizes.

(2) For more details on Rx masking value, refer to Section 18.5.4.4.2.1, *TDM - Mode Reception Data Alignment Settings*

(3) For more details on rotation and received TDM stream bit order, refer to Section 18.5.4.4.2.1, *TDM - Mode Reception Data Alignment Settings* and Table 18-330, *McASP RFU Settings*.

18.5.5.1.2.2.2 Subsequence – Receive Frame Synchronization Generator Configuration in TDM Mode

The procedure in Table 18-344 configures the transmit frame synchronization generator of the McASP module.

NOTE: The same bit - MCASP_ACLKXCTL[6] ASYNC which is used to determine if McASP receivers and transmitters work synchronously on the same clock, is also used to define if receiver frame sync is derived from the transmit frame sync generator, or generated independently in the receiver (either inernally or externally sourced). Hence, the settings in below table Table 18-344 have no effect, if MCASP_ACLKXCTL[6] ASYNC = 0.

Table 18-344. Receive Frame-Synchronization Generator Configuration for TDM-Reception

Step	Register/Bit Field/Programming Model	Value
Select number of TDM slots per frame. Must be set to 0x2, in case of an I2S-reception. For more details on frame-sync generator, refer to Section 18.5.4.2.3.	MCASP_RXFMCTL[15:7] RMOD	0x-(1)
Choose the receive frame sync width -single bit/single word. For more details on frame-sync generator, refer to Section 18.5.4.2.3.	MCASP_RXFMCTL[4] FRWID	0x-
Select start of received frame sync polarity - rising /falling edge. For more details on frame-sync generator, refer to Section 18.5.4.2.3.	MCASP_RXFMCTL[0] FSRP	0x-
IFreceive frame sync - FS is internally generated	Software test condition	
Select internally- generated receive frame sync. For more details on frame-sync generator, refer to Section 18.5.4.2.3.	MCASP_RXFMCTL[1] FSRM	0b1
If McASP receiver is required to output internally generated frame, AFSR pin must be set as an output in step 9 of the sequence documented in the Table 18-342. This must not be done in current step because the frame control register - MCASP_TXFMCTL must be appropriately configured prior to AFSR pin outputting a frame to an external device.	MCASP_PDIR[31] AFSR	0b1
ELSE		
Select externally- generated receive frame sync. For more details on frame-sync generator, refer to Section 18.5.4.2.3.	MCASP_RXFMCTL[1] FSRM	0b0
Setup the AFSR pin as input (device level: mcaspi_fsr)	MCASP_PDIR[31] AFSR	0b0
ENDIF		
To generate McASP receive frame sync in receiver logic, select an asynchronous frame sync.	MCASP_ACLKXCTL[6] ASYNC	0b1

(1) Must be set to 0x180 in case of 384-TDM slot frame reception from a DIR component I2S-output. For more details on TDM-frame settings, refer to Section 18.5.4.9.2 .

18.5.5.1.2.2.3 Subsequence – Receive Clock Generator Configuration

The procedure in Table 18-345 configures the receive clock generator of the McASP module.

NOTE: The settings in below table Table 18-345 have no effect, if MCASP_ACLKXCTL[6] ASYNC = 0 (i.e. receive clock is sourced from the inverted version of the transmit clock). For example, such is the case when McASP loopback mode is used.

Table 18-345. Receive Clock Generator Configuration

Step	Register/Bit Field/Programming Model	Value
To use the McASP receive clock generator, select an asynchronous receiver clock schema (ASYNC=1). Otherwise an inverted version of transmit clock XCLK is used (receiver synchronized with transmitter).	MCASP_ACLKXCTL[6] ASYNC	0b1
IF receive clock - RCLK is internally generated	Software test condition	
The high-speed receive clock - AHCLKR is internally generated based on AUXCLK		
Select an internally-generated high-frequency clock.	MCASP_AHCLKRCTL[15] HCLKRM	0b1
Select the internal high-speed clock source polarity: non-inverted or inverted.	MCASP_AHCLKRCTL[14] HCLKRP	0x-
Set the divisor for the internally generated high-frequency clock – AHCLKR in range (1 - 4096).	MCASP_AHCLKRCTL[11:0] HCLKRDIV	0x-
Select an internally-generated receive clock.	MCASP_ACLKRCTL[5] CLKRM	0b1
Receiver samples on rising/falling edge. Select Rx sampling on the rising edge if transmitter shifts out on falling edge, and vice versa.	MCASP_ACLKRCTL[7] CLKRP	0x-
Set the divisor for the internally generated receive clock– ACLKR in range (1 - 32).	MCASP_ACLKRCTL[4:0] CLKRDIV	0x-
Optional: If McASP receiver is required to output internally generated clock, ACLKR pin must be set as an output in step 9 of the sequence documented in the Table 18-342. This must not be done in current step because the clock control register - MCASP_ACLKRCTL must be appropriately configured prior to ACLKR pin outputting a receive clock to an external device.	MCASP_PDIR[29] ACLKR	0b1
ELSE		
Select an externally-generated receive clock. Note that in this case the AHCLKR signal path and the CLKRDIV divider are NOT used.	MCASP_ACLKRCTL[5] CLKRM	0b0
Receiver samples on rising/falling edge. Select Rx sampling on the rising edge if transmitter shifts out on falling edge, and vice versa.	MCASP_ACLKRCTL[7] CLKRP	0x-
Setup an input directon for the ACLKR pin	MCASP_PDIR[29] ACLKR	0b0
ENDIF		
Configure the transmit clock failure detect logic.	See Section 18.5.4.15.6.1, *Clock Failure Check Startup*.	

18.5.5.1.2.2.4 Subsequence—McASP Receiver Pins Functional Configuration

The procedure in Table 18-346 configures the McASP pins for McASP functionality.

Table 18-346. McASP Receiver Pins Functional Configuration

Step	Register/Bit Field/Programming Model	Value
Configure module different pins to have McASP functionality.	MCASP_PFUNC[31:0]	0x0
Configure the McASP pins direction: AFSR ACLKR Desired n-th McASP data pin AXRn is configured as an input for receiving.	MCASP_PDIR[31] AFSR; MCASP_PDIR[29] ACLKR; MCASP_PDIR[n] AXRn;	0x-(1) 0x- (2) 0x0

(1) See Table 18-344.

(2) For more details on McASP clock configurations, refer to Table 18-345.

18.5.5.1.2.3 Main Sequence – McASP Global Initialization for TDM -Transmission

The procedure in Table 18-347 initializes a McASP serializer n transmitter(s) to operate in TDM-mode after a power-on reset (POR). This is used for I2S (2-slot TDM) and other TDM-based audio protocols transmission.

CAUTION

Before performing McASP global initialization, If external clock ACLKR is used, it must be running already for proper synchronization of the MCASP_GBLCTL register.

Table 18-347. McASP Transmitters Global Initialization for TDM-Mode Operation

Step	Register/Bit Field/Programming Model	Value
1. Apply software reset to different McASP transmit components.	MCASP_GBLCTL[12:8]	0x00
2. Poll the bits to ensure the active reset value (0x00) is successfully latched into the register.	MCASP_GBLCTL[12:8]	=0x00
3. Configure the local power management.	PWRIDLESYSCONFIG[1:0] IDLE_MODE	0x1
4. Configure the transmit format unit.	See Section 18.5.5.1.2.3.1.	
5. Configure the transmit frame sync generator.	See Section 18.5.5.1.2.3.2.	
6. Configure the transmit clock generator.	See Section 18.5.5.1.2.3.3.	
7. Program all bits - XTDMSk, where k=0 to 31, according to the time slot characteristics desired (positions of active versus inactive slots within a frame).	MCASP_TXTDM [k] XTDMSk, where k=0 to 31 (1)	0x-
8. Configure the desired n-th serializer for transmit mode operation. (2)	MCASP_XRSRCTLn[1:0] SRMOD;	0x1
9. Setup all active transmitters to operate in TDM mode.	MCASP_TXDITCTL[0] DITEN	0x0 (3)
10. Configure the McASP pins functionality.	See Section 18.5.5.1.2.3.4.	
11. Optional: Configure a McASP Tx channel for loopback operation (TDM mode only) in MCASP_LBCTL [31:0].	See Section 18.5.4.14.1, *Loopback Mode Configurations*.	0x-
12. Release from reset state the divider that outputs the AHCLKR clock. See (4)	MCASP_GBLCTL[9] XHCLKRST	0x1
13. Poll the bit to ensure that it is successfully latched in the register.	MCASP_GBLCTL[9] XHCLKRST	=0x1
14. Release from reset state the divider that outputs the ACLKR clock. See (4)	MCASP_GBLCTL[8] XCLKRST	0x1
15. Poll the bit to ensure that it is successfully latched in the register.	MCASP_GBLCTL[8] XCLKRST	=0x1

(1) Appropriately program in bitfield MCASP_XRSRCTLn [3:2] DISMOD, the desired level (high-impedance state, 0, or 1) at AXRn output, during time of inactive slots. Note, that this setting does NOT apply when all slots are programmed to be active within a frame (in particular DIT-mode).

(2) For an unused serializer n, write MCASP_XRSRCTLn [1:0] SRMOD=0x0 to disable it.

(3) All active transmit channels operate either in TDM mode or in DIT mode depending on DITEN value. There is no option to choose Tx Mode between DIT and TDM separately per serializer transmitter.

(4) During reset state the local McASP internal clock dividers maintain a 1:1 ratio at their outputs. The values stored in the MCASP_AHCLKX and MCASP_ACLKX registers are ignored; hence, the transmission clock does not stop during the reset state of the dividers.

18.5.5.1.2.3.1 Subsequence – Transmit Format Unit Configuration in TDM Mode

The procedure in Table 18-348 configures the transmit frame format unit of the McASP module for TDM slots transmission.

Table 18-348. Transmit Format Unit Configuration for TDM-Transmission

Step	Register/Bit Field/Programming Model	Value
Configure the desired TDM-slot size	MCASP_TXFMT[7:4] XSSZ	0x-(1)
Set data mask out value (0x0000 0000 - 0xFFFF FFFF).	MCASP_TXMASK[31:0] XMASK	0x-(2)
Select a padding value for masked-out bits.	MCASP_TXFMT[14:13] XPAD	0x-
Specify position (0x0-0x1F) of the bit in corresponding register MCASP_TXBUFn which value to be used as a pad value in case MCASP_TXFMT[14:13] XPAD=0x2.	MCASP_TXFMT[12:8] XPBIT	0x-
Rotate data right by a multiple of 4- bit positions.	MCASP_TXFMT[2:0] XROT	0x- (3)
transmitted stream bit order (LSB- or MSB-first). Must be set to 0x1 for an I2S stream transmission (MSB-first).	MCASP_TXFMT[15] XRVRS	0x- (3)
Specify a delay between frame sync and first bit of data in number of bits. Must be set to 0x1 for an I2S stream transmission.	MCASP_TXFMT[17:16] XDATDLY	0x-
Select to write data to active serializers transmit buffers using peripheral (CFG) or DATA port	MCASP_TXFMT[3] XBUSEL	0x-

(1) Refer to Section 18.5.4.4.1, *Transmit Format Unit* , regarding options for transmitted TDM-slot sizes.
(2) For more details on Tx masking value, refer to Section 18.5.4.4.1.1, *TDM - Mode Transmission Data Alignment Settings*
(3) For more details on rotation and transmitd TDM stream bit order, refer to Section 18.5.4.4.1.1, *TDM - Mode Transmission Data Alignment Settings* and Table 18-328, *McASP TFU TDM Mode Settings*.

18.5.5.1.2.3.2 Subsequence – Transmit Frame Synchronization Generator Configuration in TDM Mode

The procedure in Table 18-349 configures the transmit frame synchronization generator of the McASP module.

Table 18-349. Transmit Frame-Synchronization Generator Configuration for TDM-Transmission

Step	Register/Bit Field/Programming Model	Value
Select number of TDM slots per frame (2 - 32). Must be set to 0x2, in case of an I2S-transmission. For more details on frame-sync generator, refer to Section 18.5.4.2.3.	MCASP_TXFMCTL[15:7] XMOD	0x-
Choose the transmit frame sync width -single bit/single word. For more details on frame-sync generator, refer to Section 18.5.4.2.3.	MCASP_TXFMCTL[4] FXWID	0x-
Select start of transmit frame sync polarity - rising /falling edge. For more details on frame-sync generator, refer to Section 18.5.4.2.3.	MCASP_TXFMCTL[0] FSXP	0x-
IF transmit frame sync - FS is internally generated	Software test condition	
Select internally- generated transmit frame sync. For more details on frame-sync generator, refer to Section 18.5.4.2.3.	MCASP_TXFMCTL[1] FSXM	0b1
If McASP transmitter is required to output internally generated frame, AFSX pin must be set as an output in step 10 of the sequence documented in the Table 18-347. This must NOT be done in current step because the frame control register - MCASP_TXFMCTL must be appropriately configured prior to AFSX pin outputting a frame sync to an external device.	MCASP_PDIR[28] AFSX	0b1
ELSE		
Select externally- generated transmit frame sync. For more details on frame-sync generator, refer to Section 18.5.4.2.3.	MCASP_TXFMCTL[1] FSXM	0b0
Setup the AFSX pin as input	MCASP_PDIR[28] AFSX	0b0

18.5.5.1.2.3.3 Subsequence – Transmit Clock Generator Configuration for TDM Cases

The procedure in Table 18-350 configures the transmit clock generator of the McASP module.

Table 18-350. Transmit Clock Generator Configuration for TDM Cases

Step	Register/Bit Field/Programming Model	Value
IF transmit clock - XCLK is internally generated	Software test condition	
IF high-speed transmit clock - AHCLKX is internally generated based on AUXCLK	Software test condition	
Select an internally-generated high-frequency clock.	MCASP_AHCLKXCTL[15] HCLKXM	0b1
Select the high-frequency clock source polarity: non-inverted or inverted.	MCASP_AHCLKXCTL[14] HCLKXP	0x-
Set the divisor for the internally generated high-frequency clock – AHCLKX in range (1 - 4096).	MCASP_AHCLKXCTL[11:0] HCLKXDIV	0x-
Optional: If McASP transmitter is required to output internally generated high-frequency clock, AHCLKX pin must be set as an output in step 10 of the sequence documented in the Table 18-347. This must NOT be done in current step because the clock control register - MCASP_AHCLKXCTL must be appropriately configured prior to AHCLKX pin outputting a high-speed clock to an external device.	MCASP_PDIR[27] AHCLKX	0b1
ELSE		
Select an externally-generated high frequency clock (HCLKXDIV divider can not be used).	MCASP_AHCLKXCTL[15] HCLKXM	0b0
Select the high-speed transmit clock source polarity: non-inverted or inverted.	MCASP_AHCLKXCTL[14] HCLKXP	0x-
Setup an input directon for the AHCLKX pin	MCASP_PDIR[27] AHCLKX	0b0
ENDIF		
Select an internally-generated transmit clock.	MCASP_ACLKXCTL[5] CLKXM	0b1
Transmitter samples on rising/falling edge. Select Tx shifting out data on the rising edge if receiver samples on falling edge, and vice versa.	MCASP_ACLKXCTL[7] CLKXP	0x-
Set the divisor for the internally generated transmit clock– ACLKX in range (1 - 32).	MCASP_ACLKXCTL[4:0] CLKXDIV	0x-
Optional: If McASP transmitter is required to output internally generated clock, ACLKX pin) must be set as an output in step 10 of the sequence documented in the Table 18-347. This must NOT be done in current step because the clock control register - MCASP_ACLKXCTL must be appropriately configured prior to ACLKX pin outputting a transmit clock to an external device.	MCASP_PDIR[26] ACLKX	0b1
ELSE		
Select an externally-generated transmit clock. Note that in this case the AHCLKX signal path and the CLKXDIV divider are NOT used.	MCASP_ACLKXCTL[5] CLKXM	0b0
Transmitter samples on rising/falling edge. Select Tx shifting out data on the rising edge if receiver samples on falling edge, and vice versa.	MCASP_ACLKXCTL[7] CLKXP	0x-
Setup an input directon for the ACLKX pin	MCASP_PDIR[26] ACLKX	0b0
ENDIF		
Configure the transmit clock failure detect logic.	See Section 18.5.4.15.6.1, *Clock Failure Check Startup*.	

18.5.5.1.2.3.4 Subsequence—McASP Transmit Pins Functional Configuration

The procedure in Table 18-351 configures the McASP pins for McASP functionality.

Table 18-351. McASP Transmit Pins Functional Configuration

Step	Register/Bit Field/Programming Model	Value
Configure module different pins to have McASP functionality.	MCASP_PFUNC[31:0]	0x0
Configure the McASP pins direction: AFSX AHCLKX ACLKX Desired n-th McASP data pin AXRn is configured as an ouput for transmission.	MCASP_PDIR[28] AFSR; MCASP_PDIR[27] AHCLKR; MCASP_PDIR[26] ACLKR; MCASP_PDIR[n] AXRn	0x-(1) 0x-(2) 0x- (2) 0x1

(1) See Table 18-349.

(2) For more details on McASP clock configurations, refer to Table 18-345.

18.5.5.2 Operational Modes Configuration

18.5.5.2.1 McASP Transmission Modes

18.5.5.2.1.1 Main Sequence – McASP DIT- /TDM- Polling Transmission Method

Figure 18-105 shows the McASP DIT-/TDM- polling method.

Figure 18-105. McASP DIT- /TDM- Transmission Polling Method

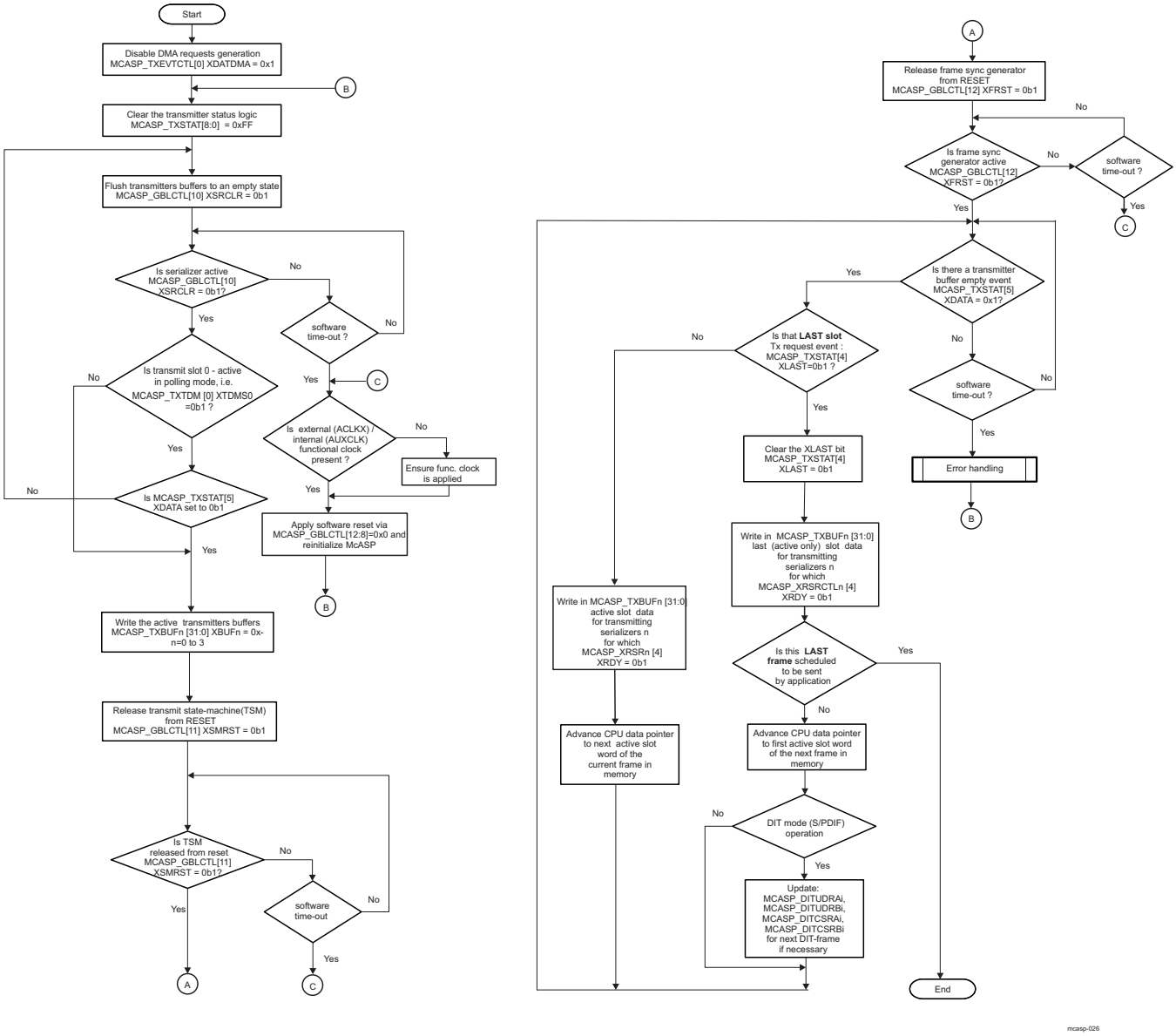


Table 18-352 summarizes the register call for the transmission DIT-/TDM- polling mode.

Table 18-352. Register Call Summary for Main Sequence – McASP DIT-/TDM- Transmission Polling Method

Register Name
MCASP_XEVTCTL
MCASP_TXSTAT
MCASP_GBLCTL
MCASP_TXTDM
MCASP_TXBUFn
MCASP_XRSRCTLn
MCASP_DITUDRAi (i=0 to 5)
MCASP_DITUDRBi (i=0 to 5)
MCASP_DITCSRAi (i=0 to 5)

Table 18-352. Register Call Summary for Main Sequence – McASP DIT-/TDM- Transmission Polling Method (continued)

Register Name
MCASP_DITCSRBi (i=0 to 5)

Table 18-353 summarizes the subprocess call for the DIT-/TDM- transmission polling mode.

Table 18-353. Subprocess Call Summary for Main Sequence – McASP DIT-/TDM- Transmission Polling Method

Subprocess Name	Cross-Reference
Error handling	Figure 18-111

18.5.5.2.1.2 Main Sequence – McASP DIT- /TDM - Interrupt Transmission Method

Figure 18-106 shows the initial setup for interrupt-based transmission.

Figure 18-106. Subsequence – DIT-/TDM- Transmission Startup Procedure

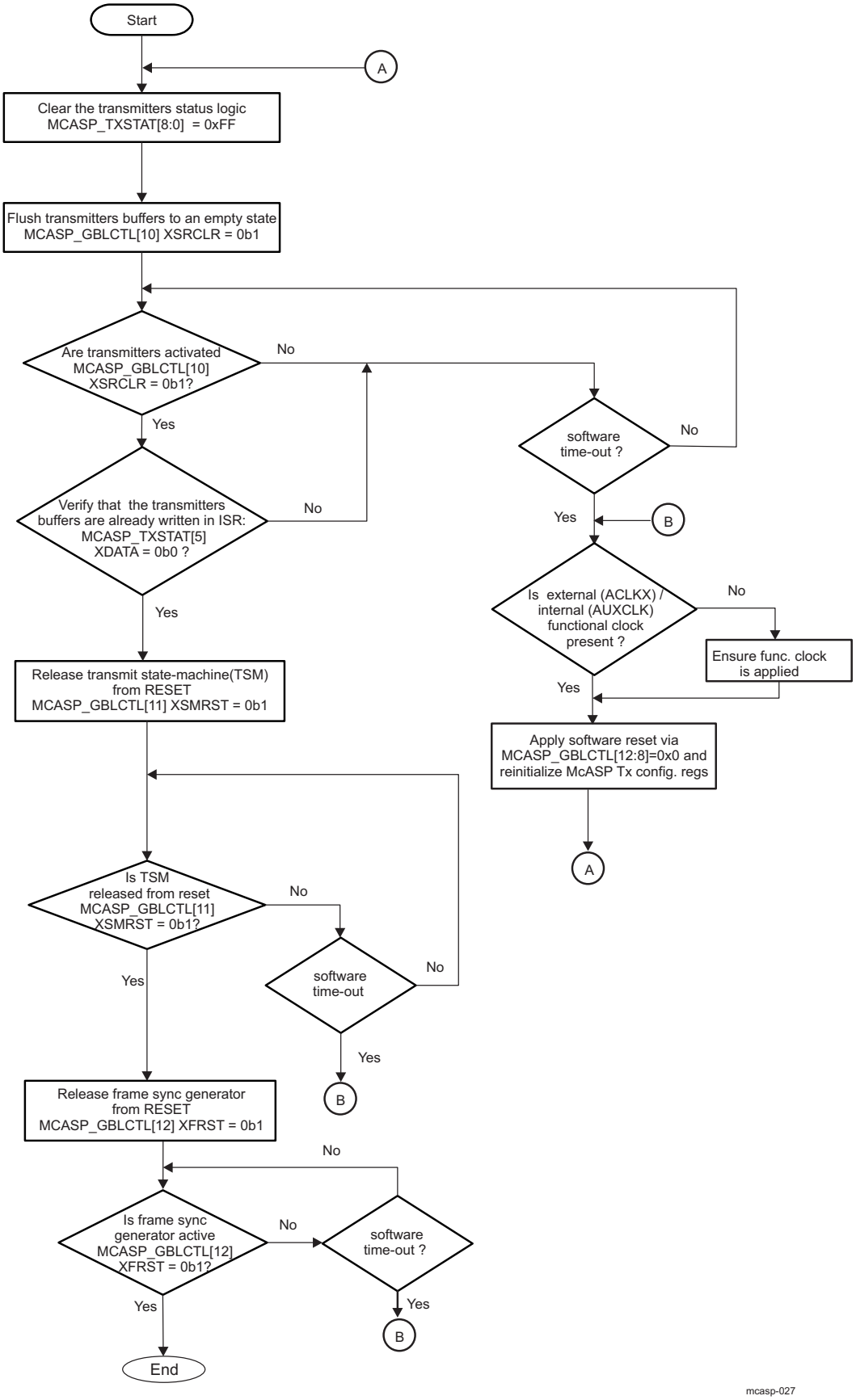


Table 18-354 shows the configuration of the McASP using an interrupt method for DIT-/TDM- transmission.

Table 18-354. McASP DIT-/TDM- Interrupt Transmission Model

Step	Register/Bit Field/Programming Model	Value
Disable Tx DMA requests generation.	MCASP_XEVTCTL[0] XDATDMA	0x1
Enable the data ready event transmit interrupt.	MCASP_EVTCTLX[5] XDATA	0x1
Optional: Enable the transmit error event interrupts.	MCASP_EVTCTLX[2] XCKFAIL MCASP_EVTCTLX[1] XSYNCERR MCASP_EVTCTLX[0] XUNDRN	0x1 0x1 0x1
Optional: Enable the start of frame interrupt. Optional: Enable the last slot data interrupt (useful for DIT user data/ channel status next S/PDIF frame info update.)	MCASP_EVTCTLX [7] XSTAFRM MCASP_EVTCTLX[4] XLAST	0x1 0x1
IFwrite transfer is through the McASP DATA port (MCASP_TXFMT[3] XBUSEL is set to 0b0).	Software test condition (setting is done in step4 of the *McASP Transmitters Global Initialization* - see Table 18-336)	
Enable the DATA port error based interrupt.	MCASP_EVTCTLX[3] XDMAERR	0x1
ELSE		
Disable the DATA port error based interrupt.	MCASP_EVTCTLX[3] XDMAERR	0x0
ENDIF		
DIT/TDM - Transmission Startup Procedure	See Figure 18-106.	

Table 18-355 summarizes the register call to initialize the McASP to transmit using interrupt events.

Table 18-355. Register Call Summary for Subsequence – McASP DIT-/TDM- Transmission Startup Procedure

Register Name	Register Name
MCASP_GBLCTL	MCASP_TXSTAT

18.5.5.2.1.3 Main Sequence –McASP DIT- /TDM - Mode DMA Transmission Method

Table 18-356 shows the configuration of the McASP using the DMA method for transmission. Possible interrupt error event servicing is also considered. Table 18-355 shows the initial setup for DMA - based transmission.

NOTE: Because of the DATA port burst access capability with the DMA method, it is strongly recommended that DMA transfers are initiated through the McASP DATA port.

Table 18-356. McASP DMA Transmission Model with Interrupt Events Servicing

Step	Register/Bit Field/Programming Model	Value
Recommended: Select DATA port to access the transmit buffers.	MCASP_TXFMT[3] XBUSEL	0x0
Enable the Tx DMA requests generation.	MCASP_XEVTCTL[0] XDATDMA	0x0
Enable the Tx DMA error event, because of McASP DATA port usage.	MCASP_EVTCTLX[3] XDMAERR	0x1
Optional: Enable the transmit error event interrupts.	MCASP_EVTCTLX[2] XCKFAIL MCASP_EVTCTLX[1] XSYNCERR MCASP_EVTCTLX[0] XUNDRN	0x1 0x1 0x1
Optional: Enable the start of frame interrupt. Optional: Enable the last slot data interrupt.	MCASP_EVTCTLX [7] XSTAFRM MCASP_EVTCTLX[4] XLAST	0x1 0x1
Disable the data ready event transmit interrupt, as DMA is used to service this request.	MCASP_EVTCTLX[5] XDATA	0x0

Table 18-356. McASP DMA Transmission Model with Interrupt Events Servicing (continued)

Step	Register/Bit Field/Programming Model	Value
DMA startup transmission procedure. This procedure is identical than the one shown in Figure 18-106. The only difference is that DMA automatically services all the AXEVT events raised by the McASP, and no CPU data processing intervention is required. The CPU is involved only in error handling shown in Figure 18-109.	See Figure 18-106.	

18.5.5.2.2 McASP Reception Modes

18.5.5.2.2.1 Main Sequence – McASP Polling Reception Method

Figure 18-107 shows the McASP polling reception method.

NOTE: The McASP polling reception model considers thedevice CPUs as the accessor of audio data from the McASP receive buffers.

Figure 18-107. McASP Polling Reception Method

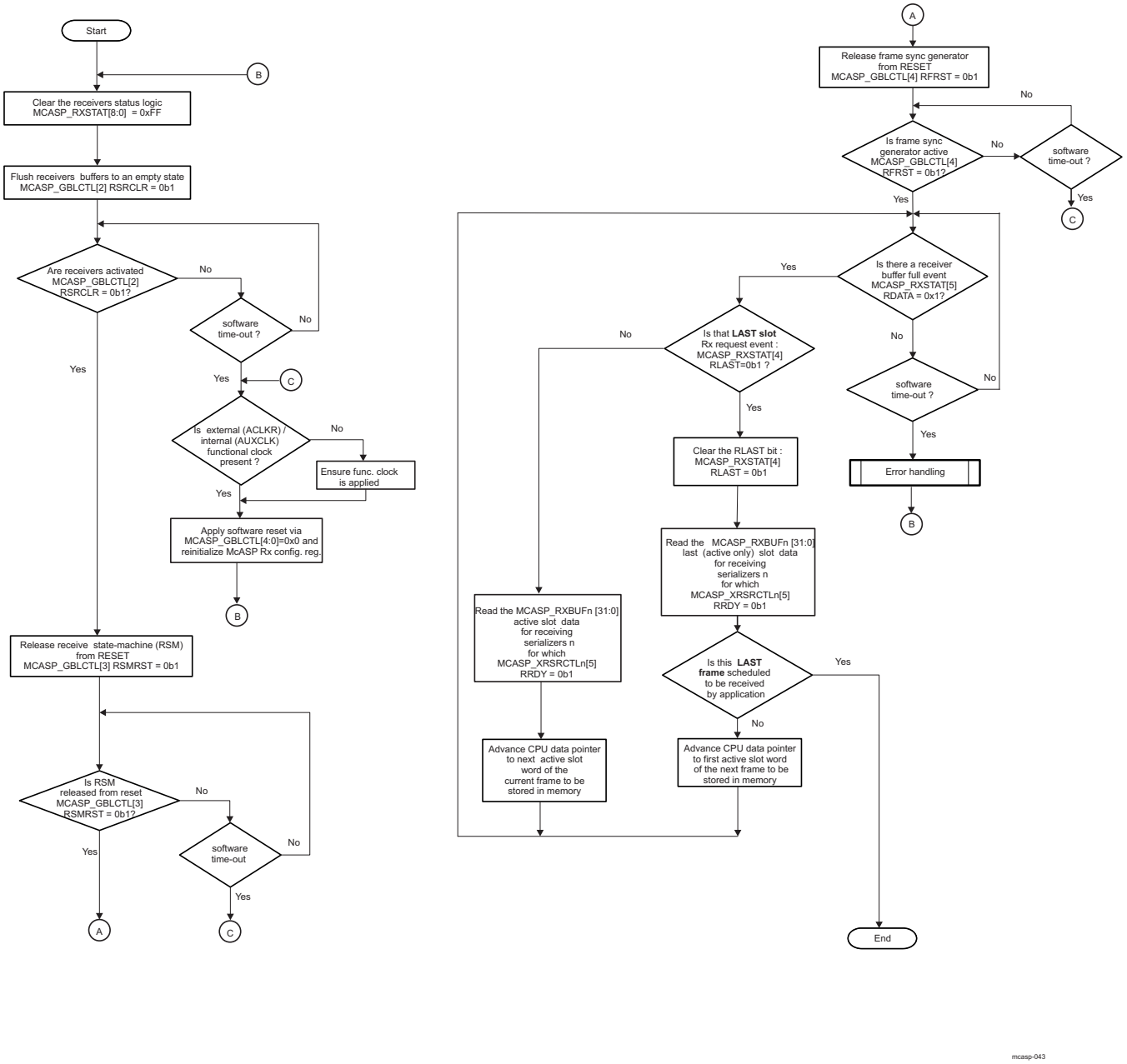


Table 18-357 summarizes the register call for the reception polling mode.

Table 18-357. Register Call Summary for Main Sequence – McASP Reception Polling Method

Register Name
MCASP_RXSTAT
MCASP_GBLCTL
MCASP_RXBUFn
MCASP_XRSRCTLn

Table 18-358 summarizes the subprocess call for the polling mode.

Table 18-358. Subprocess Call Summary for Main Sequence – McASP Reception Polling Method

Subprocess Name	Cross-Reference
Error handling	Figure 18-112

18.5.5.2.2.2 Main Sequence – McASP TDM - Interrupt Reception Method

Figure 18-108 shows the initial setup for interrupt-based reception.

Figure 18-108. Subsequence – TDM - Reception Startup Procedure

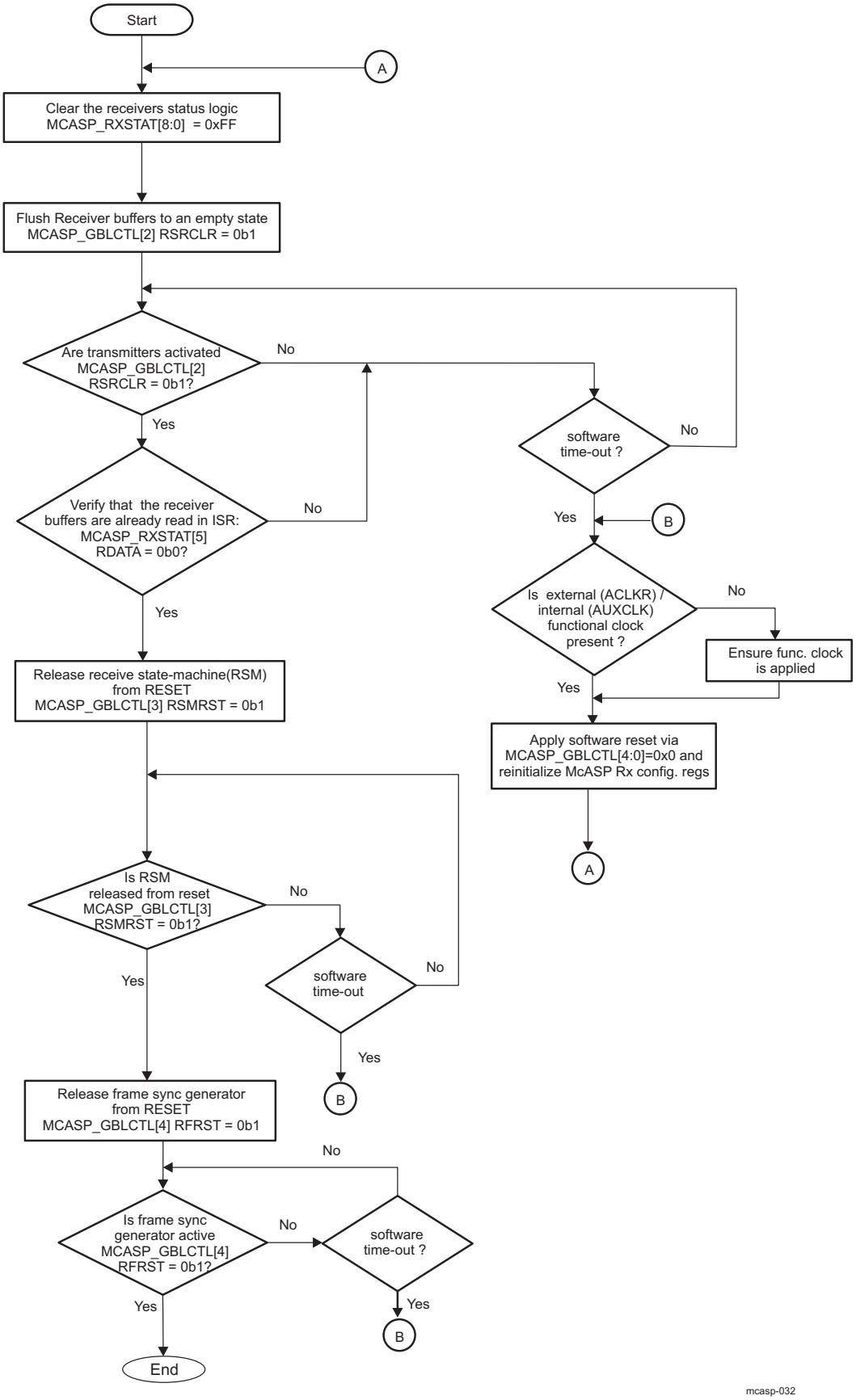


Table 18-359 shows the configuration of the McASP using an interrupt method for TDM- reception.

Table 18-359. McASP TDM- Interrupt Reception Model

Step	Register/Bit Field/Programming Model	Value
Disable Rx DMA requests generation.	MCASP_REVTCTL[0] RDATDMA	0x1
Enable the data ready event receive interrupt.	MCASP_EVTCTLR[5] RDATA	0x1
Optional: Enable the receive error event interrupts.	MCASP_EVTCTLR[2] RCKFAIL MCASP_EVTCTLR[1] RSYNCERR MCASP_EVTCTLR[0] ROVRN	0x1 0x1 0x1
Optional: Enable the start of frame interrupt. Optional: Enable the last slot data interrupt	MCASP_EVTCTLR [7] RSTAFRM MCASP_EVTCTLR[4] RLAST	0x1 0x1
IF read transfer is through the McASP DATA port (MCASP_RXFMT[3] RBUSEL is set to 0b0).	Software test condition (setting is done in step4 of the *McASP Receivers Global Initialization for TDM-Mode Operation* - see Table 18-342)	
Enable the DATA port error based interrupt.	MCASP_EVTCTLR[3] RDMAERR	0x1
ELSE		
Disable the DATA port error based interrupt.	MCASP_EVTCTLR[3] RDMAERR	0x0
ENDIF		
TDM - Transmission Startup Procedure	See Figure 18-108.	

Table 18-360 summarizes the register call to initialize the McASP to transmit using interrupt events.

Table 18-360. Register Call Summary for Subsequence – McASP TDM- Reception Startup Procedure

Register Name	Register Name
MCASP_GBLCTL	MCASP_RXSTAT

18.5.5.2.2.3 Main Sequence – McASP TDM - Mode DMA Reception Method

Table 18-361 shows the configuration of the McASP using the DMA method for reception. Possible interrupt error event servicing is also considered. Table 18-355 shows the initial setup for DMA - based transmission.

NOTE: Because of the DATA port burst access capability with the DMA method, it is strongly recommended that DMA transfers are initiated through the McASP DATA port.

Table 18-361. McASP DMA Reception Model with Interrupt Events Servicing

Step	Register/Bit Field/Programming Model	Value
Recommended: Select DATA port to access the transmit buffers.	MCASP_RXFMT[3] RBUSEL	0x0
Enable the Rx DMA requests generation.	MCASP_REVTCTL[0] RDATDMA	0x0
Enable the Rx DMA error event, because of McASP DATA port usage.	MCASP_EVTCTLR[3] RDMAERR	0x1
Optional: Enable the receive error event interrupts.	MCASP_EVTCTLR2] RCKFAIL MCASP_EVTCTLR[1] RSYNCERR MCASP_EVTCTLR[0] ROVRN	0x1 0x1 0x1
Optional: Enable the start of frame interrupt. Optional: Enable the last slot data interrupt.	MCASP_EVTCTLR [7] RSTAFRM MCASP_EVTCTLR[4] RLAST	0x1 0x1
Disable the data ready event receive interrupt, as DMA is used to service this request.	MCASP_EVTCTLR[5] RDATA	0x0

Table 18-361. McASP DMA Reception Model with Interrupt Events Servicing (continued)

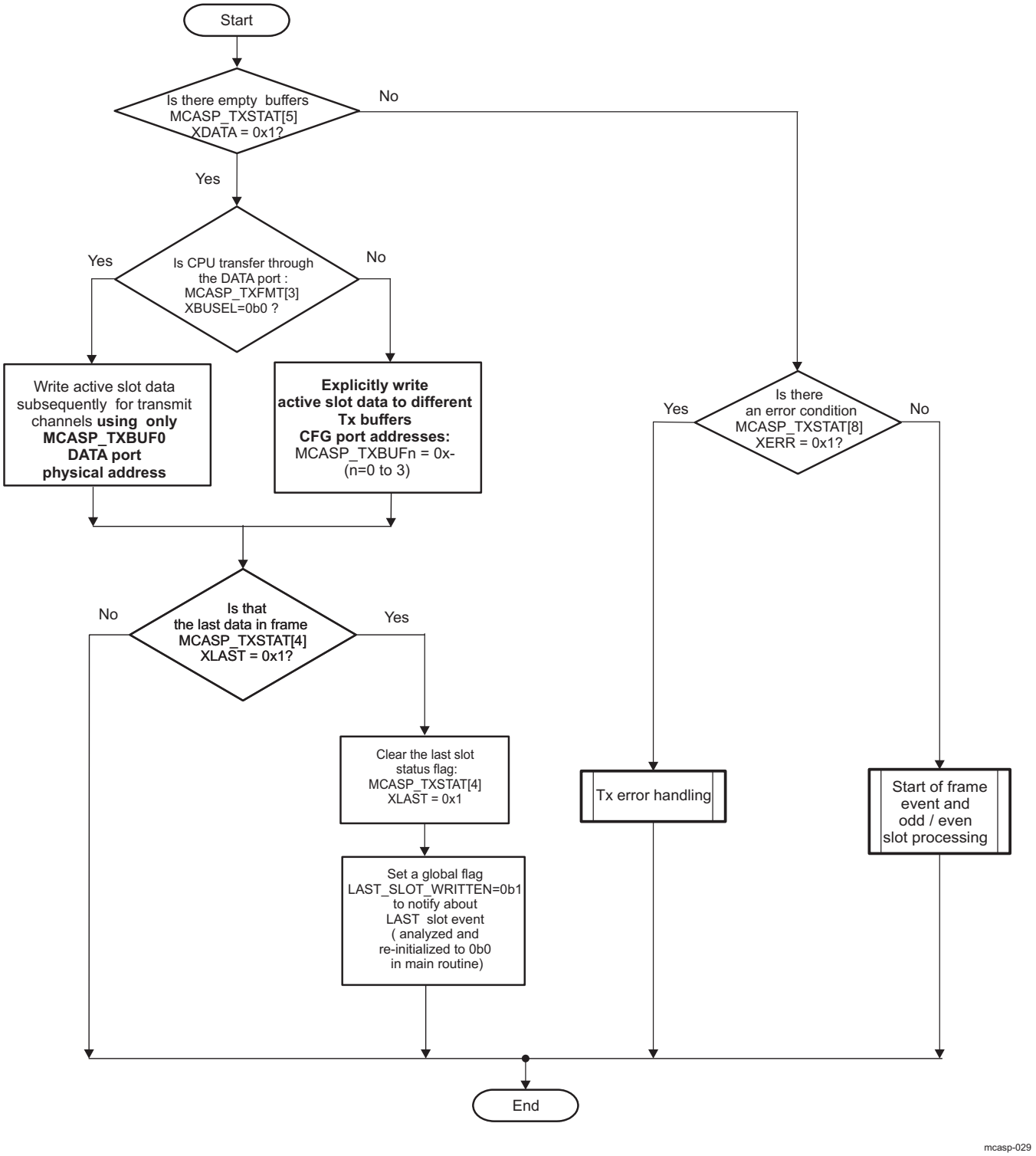
Step	Register/Bit Field/Programming Model	Value
DMA startup reception procedure. This procedure is identical than the one shown in Figure 18-108. The only difference is that DMA automatically services all the AREVT events raised by the McASP, and no CPU data processing intervention is required. The CPU is involved only in error handling shown in Figure 18-110.	See Figure 18-108.	

18.5.5.2.3 McASP Event Servicing

18.5.5.2.3.1 McASP DIT-/TDM- Transmit Interrupt Events Servicing

Figure 18-109 shows the flow of DIT-/TDM- mode transmit interrupt events servicing for the McASP module.

Figure 18-109. McASP Transmit Interrupt Events Servicing



18.5.5.2.3.2 McASP TDM- Receive Interrupt Events Servicing

Figure 18-110 shows the flow of DIT-/TDM- mode transmit interrupt events servicing for the McASP module.

Figure 18-110. McASP Receive Interrupt Events Servicing

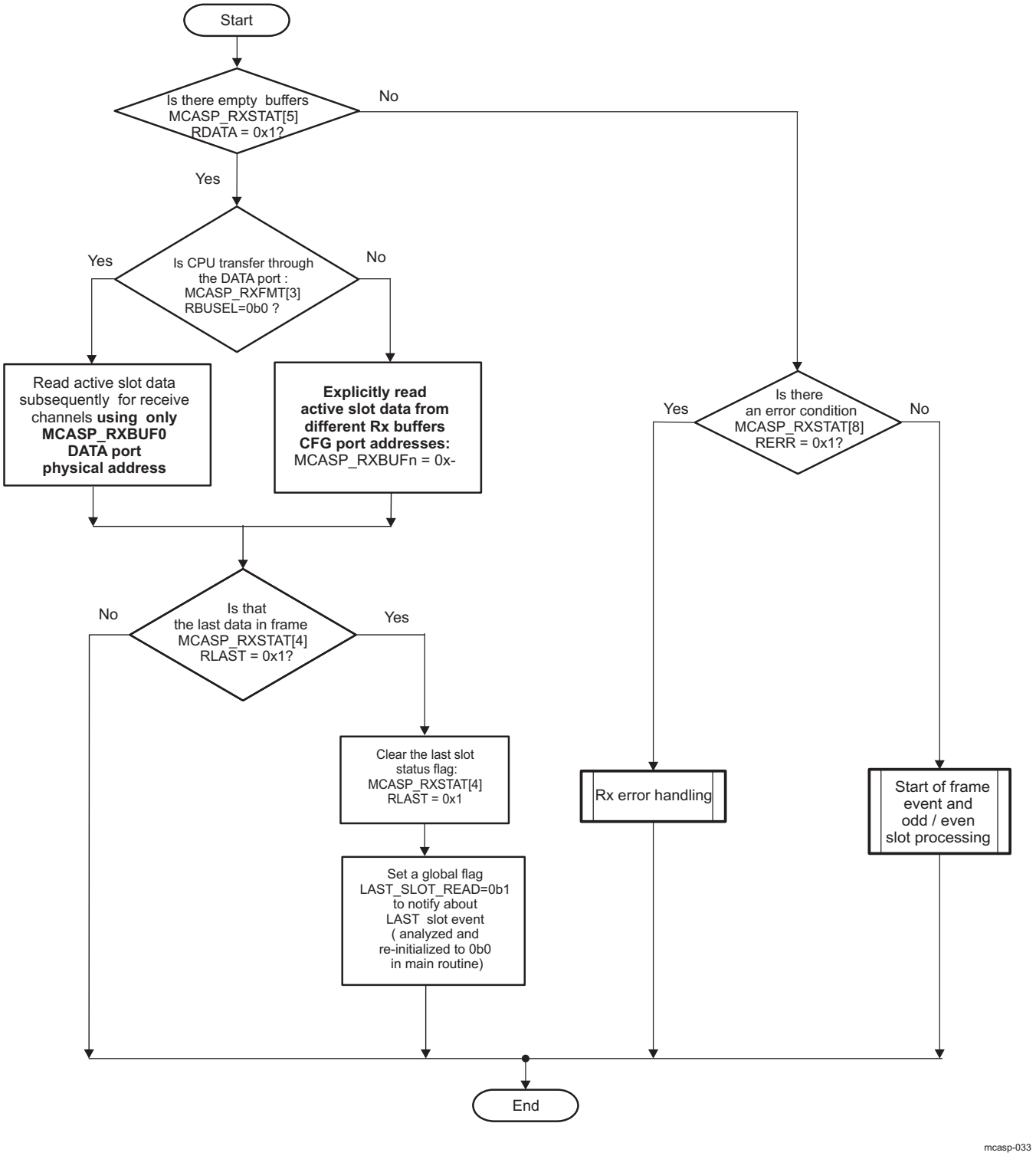


Table 18-362 lists the register call summary for the receive interrupt events servicing.

Table 18-362. Register Call Summary for McASP Receive Interrupt Events Servicing

Register Name	Register Name	Register Name
MCASP_RXSTAT	MCASP_RXBUFn	MCASP_RXFMT

Table 18-363 lists the subprocess call summary for receive interrupt events servicing.

Table 18-363. Subprocess Call Summary for Receive Interrupt Events Servicing

Subprocess Name	Cross-Reference
McASP receive error handling	Figure 18-112
Start of frame handling	Section 18.5.4.12.2

18.5.5.2.3.3 Subsequence – McASP DIT-/TDM -Modes Transmit Error Handling

Figure 18-111 shows the transmit error handling schema for the McASP, which can be implemented as part of the Tx interrupt service routine or as part of the Tx polling sequence.

Figure 18-111. McASP Transmit Error Handling

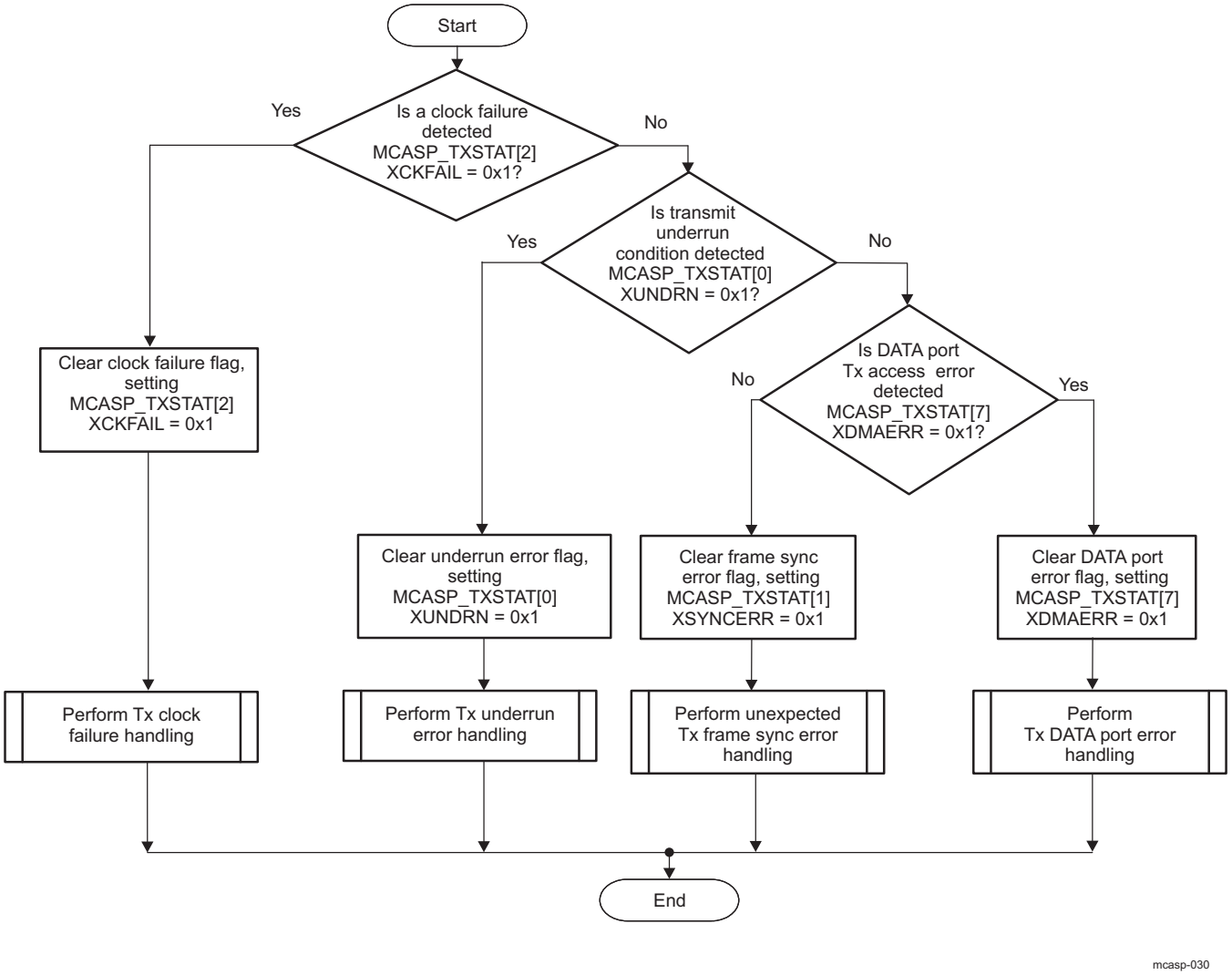


Table 18-364 lists the register call summary for the McASP transmit error handling.

Table 18-364. Register Call Summary for McASP Transmit Error Handling

Register Name
MCASP_TXSTAT

NOTE:

- For more information about transmit clock failure handling, see Section 18.5.4.15.6.2, *Transmit Clock Failure Check and Recovery*.
- For more information about transmit buffer underrun handling, see Section 18.5.4.15.1, *Buffer Underrun Error -Transmitter*.
- For more information about DATA port Tx error handling, see Section 18.5.4.15.3, *DATA Port Error - Transmitter*.
- For more information about unexpected Tx frame sync error handling, see Section 18.5.4.15.5, *Unexpected Frame Sync Error*.

18.5.5.2.3.4 Subsequence – McASP Receive Error Handling

Figure 18-112 shows the receive error handling schema for the McASP, which can ONLY be implemented as part of the Rx polling sequence.

Figure 18-112. McASP Receive Error Handling

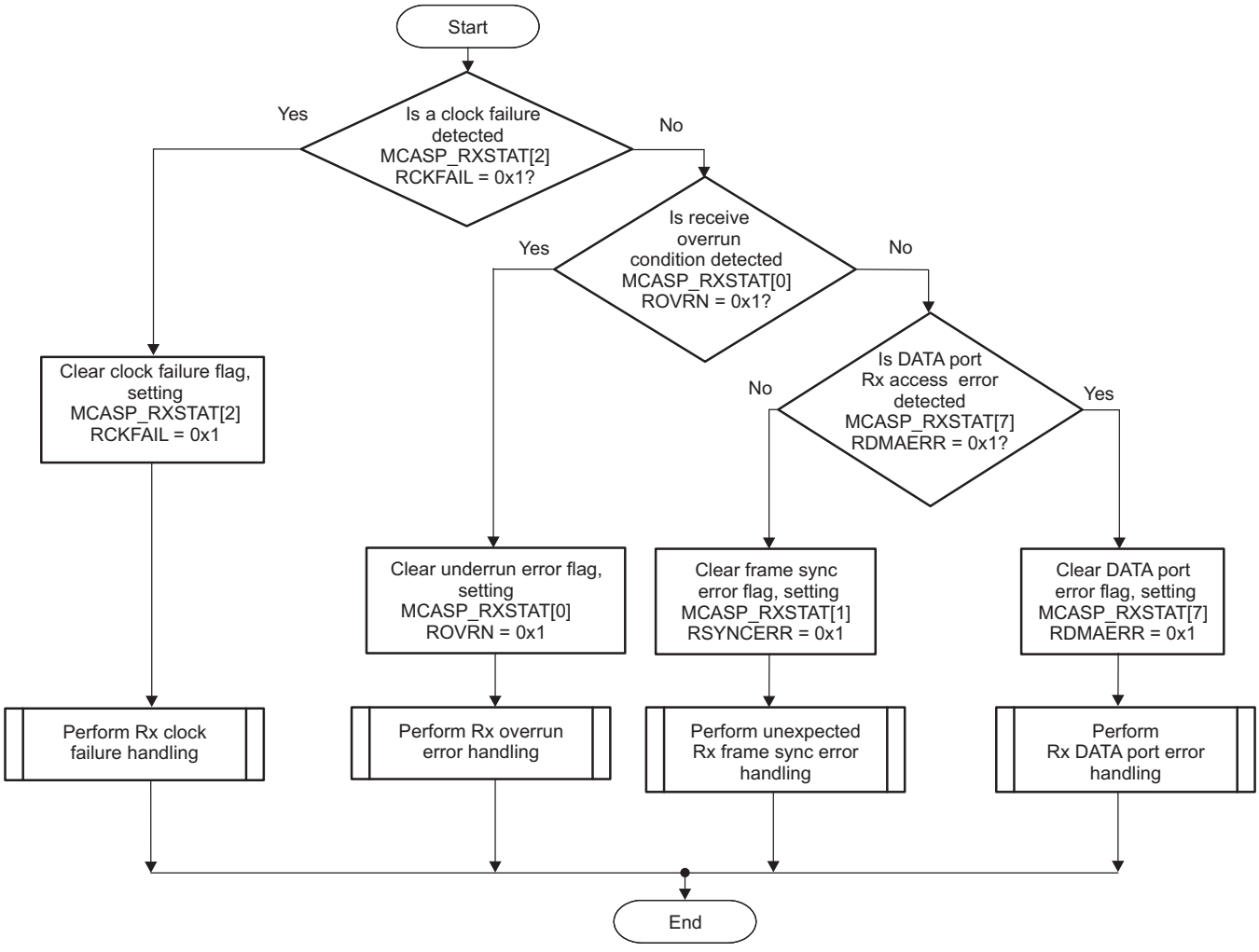


Table 18-365 lists the register call summary for the McASP receive error handling.

Table 18-365. Register Call Summary for McASP Receive Error Handling

Register Name
MCASP_RXSTAT

NOTE:

- For more information about receive clock failure handling, see Section 18.5.4.15.6.3, *Receive Clock Failure Check and Recovery*.
- For more information about receive buffer overrun handling, see Section 18.5.4.15.2, *Buffer Overrun Error - Receiver*.
- For more information about DATA port Rx error handling, see Section 18.5.4.15.4, *DATA Port Error - Receiver*.
- For more information about unexpected Rx frame sync error handling, see Section 18.5.4.15.5, *Unexpected Frame Sync Error*.

18.5.6 McASP Register Manual

18.5.6.1 McASP Instance Summary

Table 18-366 summarizes the McASP instances.

Table 18-366. McASP Instance Summary

Module Name	Base Address L3_MAIN Interconnect	Base Address L4_PER2 Interconnect	Size
MCASP1_CFG	-	0x4846 0000	4 KiB
MCASP2_CFG	-	0x4846 C000	4 KiB
MCASP3_CFG	-	0x4847 0000	4 KiB
MCASP1_AFIFO	-	0x4846 1000	4 KiB
MCASP2_AFIFO	-	0x4846 D000	4 KiB
MCASP3_AFIFO	-	0x4847 1000	4 KiB
MCASP1_DAT	0x4580 0000	-	4MB
MCASP2_DAT	-	0x4843 6000	4 KiB
MCASP3_DAT	-	0x4843 A000	4 KiB

18.5.6.2 MCASP Registers

18.5.6.2.1 MCASP_CFG Register Summary

Table 18-367 and Table 18-368 summarize the MCASP_CFG register mapping.

Table 18-367. MCASP_CFG Register Summary 1

Register Name	Type	Register Width (Bits)	Address Offset	MCASP1_CFG L4_PER2 Physical Address
MCASP_PID	R	32	0x0000 0000	0x4846 0000
PWRIDLESYSCONFIG	RW	32	0x0000 0004	0x4846 0004
MCASP_PFUNC	RW	32	0x0000 0010	0x4846 0010
MCASP_PDIR	RW	32	0x0000 0014	0x4846 0014
MCASP_PDOUT	RW	32	0x0000 0018	0x4846 0018
MCASP_PDIN	R	32	0x0000 001C	0x4846 001C
MCASP_PDSET	W	32	0x0000 001C	0x4846 001C
MCASP_PDCLR	RW	32	0x0000 0020	0x4846 0020
RESERVED	RW	32	0x0000 0030	0x4846 0030
RESERVED	RW	32	0x0000 0034	0x4846 0034
RESERVED	RW	32	0x0000 0038	0x4846 0038
MCASP_GBLCTL	RW	32	0x0000 0044	0x4846 0044
MCASP_AMUTE	RW	32	0x0000 0048	0x4846 0048
MCASP_LBCTL	RW	32	0x0000 004C	0x4846 004C
MCASP_TXDITCTL	RW	32	0x0000 0050	0x4846 0050
MCASP_GBLCTLR	RW	32	0x0000 0060	0x4846 0060
MCASP_RXMASK	RW	32	0x0000 0064	0x4846 0064
MCASP_RXFMT	RW	32	0x0000 0068	0x4846 0068
MCASP_RXFMCTL	RW	32	0x0000 006C	0x4846 006C
MCASP_ACLKRCTL	RW	32	0x0000 0070	0x4846 0070
MCASP_AHCLKRCTL	RW	32	0x0000 0074	0x4846 0074
MCASP_RXTDM	RW	32	0x0000 0078	0x4846 0078
MCASP_EVTCTLR	RW	32	0x0000 007C	0x4846 007C

Table 18-367. MCASP_CFG Register Summary 1 (continued)

Register Name	Type	Register Width (Bits)	Address Offset	MCASP1_CFG L4_PER2 Physical Address
MCASP_RXSTAT	RW	32	0x0000 0080	0x4846 0080
MCASP_RXTDMSLOT	R	32	0x0000 0084	0x4846 0084
MCASP_RXCLKCHK	RW	32	0x0000 0088	0x4846 0088
MCASP_REVTCTL	RW	32	0x0000 008C	0x4846 008C
MCASP_GBLCTLX	RW	32	0x0000 00A0	0x4846 00A0
MCASP_TXMASK	RW	32	0x0000 00A4	0x4846 00A4
MCASP_TXFMT	RW	32	0x0000 00A8	0x4846 00A8
MCASP_TXFMCTL	RW	32	0x0000 00AC	0x4846 00AC
MCASP_ACLKXCTL	RW	32	0x0000 00B0	0x4846 00B0
MCASP_AHCLKXCTL	RW	32	0x0000 00B4	0x4846 00B4
MCASP_TXTDM	RW	32	0x0000 00B8	0x4846 00B8
MCASP_EVTCTLX	RW	32	0x0000 00BC	0x4846 00BC
MCASP_TXSTAT	RW	32	0x0000 00C0	0x4846 00C0
MCASP_TXTDMSLOT	R	32	0x0000 00C4	0x4846 00C4
MCASP_TXCLKCHK	RW	32	0x0000 00C8	0x4846 00C8
MCASP_XEVTCTL	RW	32	0x0000 00CC	0x4846 00CC
MCASP_CLKADJEN	RW	32	0x0000 00D0	0x4846 00D0
MCASP_DITCSRAi (1)	RW	32	0x0000 0100 + (0x4*i)	0x4846 0100 + (0x4*i)
MCASP_DITCSRBi (1)	RW	32	0x0000 0118 + (0x4*i)	0x4846 0118 + (0x4*i)
MCASP_DITUDRAi (1)	RW	32	0x0000 0130 + (0x4*i)	0x4846 0130 + (0x4*i)
MCASP_DITUDRBi (1)	RW	32	0x0000 0148 + (0x4*i)	0x4846 0148 + (0x4*i)
MCASP_XRSRCTLn(2)	RW	32	0x0000 0180 + (0x4*n)	0x4846 0180 + (0x4*n)
MCASP_TXBUFn (2)	RW	32	0x0000 0200 + (0x4*n)	0x4846 0200 + (0x4*n)
MCASP_RXBUFn (2)	RW	32	0x0000 0280 + (0x4*n)	0x4846 0280 + (0x4*n)

(1) i = 0 to 5
(2) n = 0 to 15

Table 18-368. MCASP_CFG Register Summary 2

Register Name	Type	Register Width (Bits)	Address Offset	MCASP2_CFG L4_PER2 Physical Address	MCASP3_CFG L4_PER2 Physical Address
MCASP_PID	R	32	0x0000 0000	0x4846 C000	0x4847 0000
PWRIDLESYSCONFIG	RW	32	0x0000 0004	0x4846 C004	0x4847 0004
MCASP_PFUNC	RW	32	0x0000 0010	0x4846 C010	0x4847 0010
MCASP_PDIR	RW	32	0x0000 0014	0x4846 C014	0x4847 0014
MCASP_PDOUT	RW	32	0x0000 0018	0x4846 C018	0x4847 0018
MCASP_PDIN	R	32	0x0000 001C	0x4846 C01C	0x4847 001C
MCASP_PDSET	W	32	0x0000 001C	0x4846 C01C	0x4847 001C
MCASP_PDCLR	RW	32	0x0000 0020	0x4846 C020	0x4847 0020
RESERVED	RW	32	0x0000 0030	0x4846 C030	0x4847 0030
RESERVED	RW	32	0x0000 0034	0x4846 C034	0x4847 0034
RESERVED	RW	32	0x0000 0038	0x4846 C038	0x4847 0038
MCASP_GBLCTL	RW	32	0x0000 0044	0x4846 C044	0x4847 0044
MCASP_AMUTE	RW	32	0x0000 0048	0x4846 C048	0x4847 0048
MCASP_LBCTL	RW	32	0x0000 004C	0x4846 C04C	0x4847 004C
MCASP_TXDITCTL	RW	32	0x0000 0050	0x4846 C050	0x4847 0050
MCASP_GBLCTLR	RW	32	0x0000 0060	0x4846 C060	0x4847 0060

Table 18-368. MCASP_CFG Register Summary 2 (continued)

Register Name	Type	Register Width (Bits)	Address Offset	MCASP2_CFG L4_PER2 Physical Address	MCASP3_CFG L4_PER2 Physical Address
MCASP_RXMASK	RW	32	0x0000 0064	0x4846 C064	0x4847 0064
MCASP_RXFMT	RW	32	0x0000 0068	0x4846 C068	0x4847 0068
MCASP_RXFMCTL	RW	32	0x0000 006C	0x4846 C06C	0x4847 006C
MCASP_ACLKRCTL	RW	32	0x0000 0070	0x4846 C070	0x4847 0070
MCASP_AHCLKRCTL	RW	32	0x0000 0074	0x4846 C074	0x4847 0074
MCASP_RXTDM	RW	32	0x0000 0078	0x4846 C078	0x4847 0078
MCASP_EVTCTLR	RW	32	0x0000 007C	0x4846 C07C	0x4847 007C
MCASP_RXSTAT	RW	32	0x0000 0080	0x4846 C080	0x4847 0080
MCASP_RXTDMSLOT	R	32	0x0000 0084	0x4846 C084	0x4847 0084
MCASP_RXCLKCHK	RW	32	0x0000 0088	0x4846 C088	0x4847 0088
MCASP_REVTCTL	RW	32	0x0000 008C	0x4846 C08C	0x4847 008C
MCASP_GBLCTLX	RW	32	0x0000 00A0	0x4846 C0A0	0x4847 00A0
MCASP_TXMASK	RW	32	0x0000 00A4	0x4846 C0A4	0x4847 00A4
MCASP_TXFMT	RW	32	0x0000 00A8	0x4846 C0A8	0x4847 00A8
MCASP_TXFMCTL	RW	32	0x0000 00AC	0x4846 C0AC	0x4847 00AC
MCASP_ACLKXCTL	RW	32	0x0000 00B0	0x4846 C0B0	0x4847 00B0
MCASP_AHCLKXCTL	RW	32	0x0000 00B4	0x4846 C0B4	0x4847 00B4
MCASP_TXTDM	RW	32	0x0000 00B8	0x4846 C0B8	0x4847 00B8
MCASP_EVTCTLX	RW	32	0x0000 00BC	0x4846 C0BC	0x4847 00BC
MCASP_TXSTAT	RW	32	0x0000 00C0	0x4846 C0C0	0x4847 00C0
MCASP_TXTDMSLOT	R	32	0x0000 00C4	0x4846 C0C4	0x4847 00C4
MCASP_TXCLKCHK	RW	32	0x0000 00C8	0x4846 C0C8	0x4847 00C8
MCASP_XEVTCTL	RW	32	0x0000 00CC	0x4846 C0CC	0x4847 00CC
MCASP_CLKADJEN	RW	32	0x0000 00D0	0x4846 C0D0	0x4847 00D0
MCASP_DITCSRAi (1)	RW	32	0x0000 0100 + (0x4*i)	0x4846 C100 + (0x4*i)	0x4847 0100 + (0x4*i)
MCASP_DITCSRBi (1)	RW	32	0x0000 0118 + (0x4*i)	0x4846 C118 + (0x4*i)	0x4847 0118 + (0x4*i)
MCASP_DITUDRAi (1)	RW	32	0x0000 0130 + (0x4*i)	0x4846 C130 + (0x4*i)	0x4847 0130 + (0x4*i)
MCASP_DITUDRBi (1)	RW	32	0x0000 0148 + (0x4*i)	0x4846 C148 + (0x4*i)	0x4847 0148 + (0x4*i)
MCASP_XRSRCTLn (2)	RW	32	0x0000 0180 + (0x4*n)	0x4846 C180 + (0x4*n)	0x4847 0180 + (0x4*n)
MCASP_TXBUFn (2)	RW	32	0x0000 0200 + (0x4*n)	0x4846 C200 + (0x4*n)	0x4847 0200 + (0x4*n)
MCASP_RXBUFn (2)	RW	32	0x0000 0280 + (0x4*n)	0x4846 C280 + (0x4*n)	0x4847 0280 + (0x4*n)

(1) i = 0 to 5
(2) n = 0 to 5

NOTE: The address locations listed in Table 18-367 and Table 18-368, *MCASP_CFG Register Mapping Summary*, are relevant for accessing:

- All McASP configuration registers
- MCASP_TXBUFn registers
- MCASP_RXBUFn registers

through the McASP peripheral configuration (CFG) port.

The MCASP_TXFMT[3] XBUSEL bit must be set to 0b1, to allow CFG port write accesses to the McASP XRBUFn buffer. The MCASP_RXFMT[3] RBUSEL bit must be set to 0b1, to allow CFG port read accesses to the McASP XRBUFn buffer.

18.5.6.2.2 MCASP_CFG Register Description

The tables below describe the individual MCASP_CFG register bits.

NOTE: For all of the below described registers the indexes *n* and *N* apply to serializers (not slots).

Register descriptions cover the superset McASP (16 serializers and all signals pinned out). For particular McASP Instance, refer to Section 18.5.2, *McASP Environment*, and Section 18.5.3, *McASP Integration*.

Table 18-369. MCASP_PID

Address Offset	0x0000 0000		
Physical Address	0x4846 0000 0x4846 C000 0x4847 0000	**Instance**	MCASP1_CFG_PER2_L4 MCASP2_CFG_PER2_L4 MCASP3_CFG_PER2_L4
Description	Peripheral identification register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCHEME				RESV				FUNCTION								RTL					REVMAJOR			CUSTOM		REVMINOR						

Bits	Field Name	Description	Type	Reset
31:30	SCHEME	Scheme. Distinguishes between old scheme and current.	R	0x1
29:28	RESV	Reserved.	R	0x0
27:16	FUNCTION	McASP. Indicates a software-compatible module family.	R	0x430
15:11	RTL	RTL version.	R	0x1
10:8	REVMAJOR	Major revision number.	R	0x0
7:6	CUSTOM	Non-custom. Indicates a special version for a given device.	R	0x0
5:0	REVMINOR	Minor revision number.	R	0x0

Table 18-370. Register Call Summary for Register MCASP_PID

Multichannel Audio Serial Port
- MCASP_CFG Register Summary: [0]

Table 18-371. PWRIDLESYSCONFIG

Address Offset	0x0000 0004		
Physical Address	0x4846 0004 0x4846 C004 0x4847 0004	**Instance**	MCASP1_CFG_PER2_L4 MCASP2_CFG_PER2_L4 MCASP3_CFG_PER2_L4
Description	Power idle module configuration register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								OTHER						IDLE_MODE	

Bits	Field Name	Description	Type	Reset
31:6	RESERVED	Reserved	R	0x0000000
5:2	OTHER	Reserved for future expansion	RW	0x0
1:0	IDLE_MODE	0x0: Force-idle mode 0x1: No-idle mode 0x2: Smart-idle mode - default state 0x3: Reserved	RW	0x2

Table 18-372. Register Call Summary for Register PWRIDLESYSCONFIG

Multichannel Audio Serial Port

- MCASP Power Management: [0][1][2]
- MCASP Global Initialization: [3][4][5]
- MCASP_CFG Register Summary: [6]

Table 18-373. MCASP_PFUNC

Address Offset	0x0000 0010		
Physical Address	0x4846 0010 0x4846 C010 0x4847 0010	**Instance**	MCASP1_CFG_PER2_L4 MCASP2_CFG_PER2_L4 MCASP3_CFG_PER2_L4
Description	Specifies the function of the pins as either a McASP pin or a GPIO pin. Some register bits might not be functional for all McASP instances, due to corresponding McASP pin not being pinned out on a particular device part number. Refer to device-specific DM for more information on the supported McASP features.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AFSR	RESERVED	ACLKR	AFSX	AHCLKX	ACLKX	RESERVED										AXR15	AXR14	AXR13	AXR12	AXR11	AXR10	AXR9	AXR8	AXR7	AXR6	AXR5	AXR4	AXR3	AXR2	AXR1	AXR0

Bits	Field Name	Description	Type	Reset
31	AFSR	Determines if AFSR pin functions as McASP or GPIO. 0x0: Pin functions as McASP pin 0x1: Pin functions as GIO pin	RW	0
30	RESERVED	Reserved	RW	0
29	ACLKR	Determines if ACLKR pin functions as McASP or GPIO. 0x0: Pin functions as McASP pin 0x1: Pin functions as GIO pin	RW	0
28	AFSX	Determines if AFSX pin functions as McASP or GPIO. 0x0: Pin functions as McASP pin 0x1: Pin functions as GIO pin	RW	0
27	AHCLKX	Determines if AHCLKX pin functions as McASP or GPIO. 0x0: Pin functions as McASP pin 0x1: Pin functions as GIO pin	RW	0
26	ACLKX	Determines if ACLKX pin functions as McASP or GPIO. 0x0: Pin functions as McASP pin 0x1: Pin functions as GIO pin	RW	0
25:16	RESERVED	Reserved	RW	0x000
15	AXR15	Determines if AXR15 pin functions as McASP or GPIO. 0x0: Pin functions as McASP pin 0x1: Pin functions as GIO pin	RW	0

Bits	Field Name	Description	Type	Reset
14	AXR14	Determines if AXR14 pin functions as McASP or GPIO. 0x0: Pin functions as McASP pin 0x1: Pin functions as GIO pin	RW	0
13	AXR13	Determines if AXR13 pin functions as McASP or GPIO. 0x0: Pin functions as McASP pin 0x1: Pin functions as GIO pin	RW	0
12	AXR12	Determines if AXR12 pin functions as McASP or GPIO. 0x0: Pin functions as McASP pin 0x1: Pin functions as GIO pin	RW	0
11	AXR11	Determines if AXR11 pin functions as McASP or GPIO. 0x0: Pin functions as McASP pin 0x1: Pin functions as GIO pin	RW	0
10	AXR10	Determines if AXR10 pin functions as McASP or GPIO. 0x0: Pin functions as McASP pin 0x1: Pin functions as GIO pin	RW	0
9	AXR9	Determines if AXR9 pin functions as McASP or GPIO. 0x0: Pin functions as McASP pin 0x1: Pin functions as GIO pin	RW	0
8	AXR8	Determines if AXR8 pin functions as McASP or GPIO. 0x0: Pin functions as McASP pin 0x1: Pin functions as GIO pin	RW	0
7	AXR7	Determines if AXR7 pin functions as McASP or GPIO. 0x0: Pin functions as McASP pin 0x1: Pin functions as GIO pin	RW	0
6	AXR6	Determines if AXR6 pin functions as McASP or GPIO. 0x0: Pin functions as McASP pin 0x1: Pin functions as GIO pin	RW	0
5	AXR5	Determines if AXR5 pin functions as McASP or GPIO. 0x0: Pin functions as McASP pin 0x1: Pin functions as GIO pin	RW	0
4	AXR4	Determines if AXR4 pin functions as McASP or GPIO. 0x0: Pin functions as McASP pin 0x1: Pin functions as GIO pin	RW	0
3	AXR3	Determines if AXR3 pin functions as McASP or GPIO. 0x0: Pin functions as McASP pin 0x1: Pin functions as GIO pin	RW	0
2	AXR2	Determines if AXR2 pin functions as McASP or GPIO. 0x0: Pin functions as McASP pin 0x1: Pin functions as GIO pin	RW	0
1	AXR1	Determines if AXR1 pin functions as McASP or GPIO. 0x0: Pin functions as McASP pin 0x1: Pin functions as GIO pin	RW	0
0	AXR0	Determines if AXR0 pin functions as McASP or GPIO. 0x0: Pin functions as McASP pin 0x1: Pin functions as GIO pin	RW	0

Table 18-374. Register Call Summary for Register MCASP_PFUNC

Multichannel Audio Serial Port

- Burst Transfer Mode: [0]
- Time-Division Multiplexed (TDM) Transfer Mode: [1]
- DIT Transfer Mode: [2]
- Loopback Modes: [3][4][5]
- MCASP Global Initialization: [6][7][8]
- MCASP_CFG Register Summary: [9]
- MCASP_CFG Register Description: [13][14][15][16][17][18][19][20][21][22][23][24][25][26][27][28][29][30][31][32][33][34][35][36][37][38][39][40]

Table 18-375. MCASP_PDIR

Address Offset	0x0000 0014		
Physical Address	0x4846 0014 0x4846 C014 0x4847 0014	**Instance**	MCASP1_CFG_PER2_L4 MCASP2_CFG_PER2_L4 MCASP3_CFG_PER2_L4
Description	Pin direction register - specifies the direction of the McASP pins as either an input or an output pin. Some register bits might not be functional for all McASP instances, due to corresponding McASP pin not being pinned out on a particular device part number. Refer to device-specific DM for more information on the supported McASP features.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AFSR	RESERVED	ACLKR	AFSX	AHCLKX	ACLKX	RESERVED										AXR15	AXR14	AXR13	AXR12	AXR11	AXR10	AXR9	AXR8	AXR7	AXR6	AXR5	AXR4	AXR3	AXR2	AXR1	AXR0

Bits	Field Name	Description	Type	Reset
31	AFSR	Determines if AFSR pin functions as an input or output. 0x0: Input 0x1: Output	RW	0
30	RESERVED	Reserved	RW	0
29	ACLKR	Determines if ACLKR pin functions as an input or output. 0x0: Input 0x1: Output	RW	0
28	AFSX	Determines if AFSX pin functions as an input or output. 0x0: Input 0x1: Output	RW	0
27	AHCLKX	Determines if AHCLKX pin functions as an input or output. 0x0: Input 0x1: Output	RW	0
26	ACLKX	Determines if ACLKX pin functions as an input or output. 0x0: Input 0x1: Output	RW	0
25:16	RESERVED	Reserved	RW	0x000
15	AXR15	Determines if AXR15 pin functions as an input or output. 0x0: Input 0x1: Output	RW	0

Bits	Field Name	Description	Type	Reset
14	AXR14	Determines if AXR14 pin functions as an input or output. 0x0: Input 0x1: Output	RW	0
13	AXR13	Determines if AXR13 pin functions as an input or output. 0x0: Input 0x1: Output	RW	0
12	AXR12	Determines if AXR12 pin functions as an input or output. 0x0: Input 0x1: Output	RW	0
11	AXR11	Determines if AXR11 pin functions as an input or output. 0x0: Input 0x1: Output	RW	0
10	AXR10	Determines if AXR10 pin functions as an input or output. 0x0: Input 0x1: Output	RW	0
9	AXR9	Determines if AXR9 pin functions as an input or output. 0x0: Input 0x1: Output	RW	0
8	AXR8	Determines if AXR8 pin functions as an input or output. 0x0: Input 0x1: Output	RW	0
7	AXR7	Determines if AXR7 pin functions as an input or output. 0x0: Input 0x1: Output	RW	0
6	AXR6	Determines if AXR6 pin functions as an input or output. 0x0: Input 0x1: Output	RW	0
5	AXR5	Determines if AXR5 pin functions as an input or output. 0x0: Input 0x1: Output	RW	0
4	AXR4	Determines if AXR4 pin functions as an input or output. 0x0: Input 0x1: Output	RW	0
3	AXR3	Determines if AXR3 pin functions as an input or output. 0x0: Input 0x1: Output	RW	0
2	AXR2	Determines if AXR2 pin functions as an input or output. 0x0: Input 0x1: Output	RW	0
1	AXR1	Determines if AXR1 pin functions as an input or output. 0x0: Input 0x1: Output	RW	0
0	AXR0	Determines if AXR0 pin functions as an input or output. 0x0: Input 0x1: Output	RW	0

Table 18-376. Register Call Summary for Register MCASP_PDIR

Multichannel Audio Serial Port

- MCASP Signals: [0]
- Burst Transfer Mode: [1]
- Time-Division Multiplexed (TDM) Transfer Mode: [2]
- DIT Transfer Mode: [3]
- Loopback Modes: [4][5][6]
- MCASP Global Initialization: [7][8][9][10][11][12][13][14][15][16][17][18][19][20][21][22][23][24][25][26][27]
- MCASP_CFG Register Summary: [28]
- MCASP_CFG Register Description: [32][33][34][35][36][37][38][39][40][41][42][43][44][45][46][47][48][49][50][51][52][53][54][55][56][57][58][59]

Table 18-377. MCASP_PDOUT

Address Offset	0x0000 0018		
Physical Address	0x4846 0018 0x4846 C018 0x4847 0018	**Instance**	MCASP1_CFG_PER2_L4 MCASP2_CFG_PER2_L4 MCASP3_CFG_PER2_L4

Description

Pin data output register - holds a value for data out at all times, and may be read back at all times. The value held by MCASP_PDOUT is not affected by writing to MCASP_PDIR and MCASP_PFUNC. However, the data value in MCASP_PDOUT is driven out onto the McASP pin only if the corresponding bit in MCASP_PFUNC is set to 1 (GPIO function) and the corresponding bit in MCASP_PDIR is set to 1 (output).

When reading data, it returns the corresponding bit value in MCASP_PDOUT[n]; it does not return the input from the I/O pin.

When writing data, writes to the corresponding MCASP_PDOUT[n] bit.

PDOUT has these aliases or alternate addresses:

- MCASP_PDSET - when written to at this address, writing a 1 to a bit in MCASP_PDSET sets the corresponding bit in MCASP_PDOUT to 1; writing a 0 has no effect and keeps the bits in MCASP_PDOUT unchanged.
- MCASP_PDCLR - when written to at this address, writing a 1 to a bit in MCASP_PDCLR clears the corresponding bit in MCASP_PDOUT to 0; writing a 0 has no effect and keeps the bits in MCASP_PDOUT unchanged

There is only one set of data-out bits, MCASP_PDOUT[31:0]. The other registers, MCASP_PDSET and MCASP_PDCLR, are just different addresses for the same control bits, with different behaviors during writes.

Some register bits might not be functional for all McASP instances, due to corresponding McASP pin not being pinned out on a particular device part number. Refer to device-specific DM for more information on the supported McASP features.

Type RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AFSR	AHCLKR	ACLKR	AFSX	AHCLKX	ACLKX	RESERVED										AXR15	AXR14	AXR13	AXR12	AXR11	AXR10	AXR9	AXR8	AXR7	AXR6	AXR5	AXR4	AXR3	AXR2	AXR1	AXR0

Bits	Field Name	Description	Type	Reset
31	AFSR	Determines drive on AFSR output pin when the corresponding MCASP_PFUNC[31] and MCASP_PDIR[31] bits are set to 1. 0x0: The pin drives low. 0x1: The pin drives high.	RW	0
30	AHCLKR	Determines drive on AHCLKR output pin when the corresponding MCASP_PFUNC[30] and MCASP_PDIR[30] bits are set to 1. 0x0: The pin drives low. 0x1: The pin drives high.	RW	0
29	ACLKR	Determines drive on ACLKR output pin when the corresponding MCASP_PFUNC[29] and MCASP_PDIR[29] bits are set to 1 0x0: The pin drives low. 0x1: The pin drives high.	RW	0

Bits	Field Name	Description	Type	Reset
28	AFSX	Determines drive on AFSX output pin when the corresponding MCASP_PFUNC[28] and MCASP_PDIR[28] bits are set to 1. 0x0: The pin drives low. 0x1: The pin drives high.	RW	0
27	AHCLKX	Determines drive on AHCLKX output pin when the corresponding MCASP_PFUNC[27] and MCASP_PDIR[27] bits are set to 1. 0x0: The pin drives low. 0x1: The pin drives high.	RW	0
26	ACLKX	Determines drive on ACLKX output pin when the corresponding MCASP_PFUNC[26] and MCASP_PDIR[26] bits are set to 1 0x0: The pin drives low. 0x1: The pin drives high.	RW	0
25:16	RESERVED	Reserved	RW	0x000
15	AXR15	Determines drive on AXR15 output pin when the corresponding MCASP_PFUNC[15] and MCASP_PDIR[15] bits are set to 1. 0x0: The pin drives low. 0x1: The pin drives high.	RW	0
14	AXR14	Determines drive on AXR14 output pin when the corresponding MCASP_PFUNC[14] and MCASP_PDIR[14] bits are set to 1. 0x0: The pin drives low. 0x1: The pin drives high.	RW	0
13	AXR13	Determines drive on AXR13 output pin when the corresponding MCASP_PFUNC[13] and MCASP_PDIR[13] bits are set to 1. 0x0: The pin drives low. 0x1: The pin drives high.	RW	0
12	AXR12	Determines drive on AXR12 output pin when the corresponding MCASP_PFUNC[12] and MCASP_PDIR[12] bits are set to 1. 0x0: The pin drives low. 0x1: The pin drives high.	RW	0
11	AXR11	Determines drive on AXR11 output pin when the corresponding MCASP_PFUNC[11] and MCASP_PDIR[11] bits are set to 1. 0x0: The pin drives low. 0x1: The pin drives high.	RW	0
10	AXR10	Determines drive on AXR10 output pin when the corresponding MCASP_PFUNC[10] and MCASP_PDIR[10] bits are set to 1. 0x0: The pin drives low. 0x1: The pin drives high.	RW	0
9	AXR9	Determines drive on AXR9 output pin when the corresponding MCASP_PFUNC[9] and MCASP_PDIR[9] bits are set to 1. 0x0: The pin drives low. 0x1: The pin drives high.	RW	0
8	AXR8	Determines drive on AXR8 output pin when the corresponding MCASP_PFUNC[8] and MCASP_PDIR[8] bits are set to 1. 0x0: The pin drives low. 0x1: The pin drives high.	RW	0
7	AXR	Determines drive on AXR7 output pin when the corresponding MCASP_PFUNC[7] and MCASP_PDIR[7] bits are set to 1. 0x0: The pin drives low. 0x1: The pin drives high.	RW	0
6	AXR6	Determines drive on AXR6 output pin when the corresponding MCASP_PFUNC[6] and MCASP_PDIR[6] bits are set to 1. 0x0: The pin drives low. 0x1: The pin drives high.	RW	0

Bits	Field Name	Description	Type	Reset
5	AXR5	Determines drive on AXR5 output pin when the corresponding MCASP_PFUNC[5] and MCASP_PDIR[5] bits are set to 1. 0x0: The pin drives low. 0x1: The pin drives high.	RW	0
4	AXR4	Determines drive on AXR4 output pin when the corresponding MCASP_PFUNC[4] and MCASP_PDIR[4] bits are set to 1. 0x0: The pin drives low. 0x1: The pin drives high.	RW	0
3	AXR3	Determines drive on AXR3 output pin when the corresponding MCASP_PFUNC[3] and MCASP_PDIR[3] bits are set to 1. 0x0: The pin drives low. 0x1: The pin drives high.	RW	0
2	AXR2	Determines drive on AXR2 output pin when the corresponding MCASP_PFUNC[2] and MCASP_PDIR[2] bits are set to 1. 0x0: The pin drives low. 0x1: The pin drives high.	RW	0
1	AXR1	Determines drive on AXR1 output pin when the corresponding MCASP_PFUNC[1] and MCASP_PDIR[1] bits are set to 1. 0x0: The pin drives low. 0x1: The pin drives high.	RW	0
0	AXR0	Determines drive on AXR0 output pin when the corresponding MCASP_PFUNC[0] and MCASP_PDIR[0] bits are set to 1. 0x0: The pin drives low. 0x1: The pin drives high.	RW	0

Table 18-378. Register Call Summary for Register MCASP_PDOUT

Multichannel Audio Serial Port

- Burst Transfer Mode: [0]
- Time-Division Multiplexed (TDM) Transfer Mode: [1]
- MCASP_CFG Register Summary: [2]
- MCASP_CFG Register Description: [6][7][8][9][10][11][12][13][14][15][16][17][18][19][20][21][22][23][24][25][26][27][28][29][30][31][32][33][34][35][36][37][38][39][40][41][42][43][44][45][46][47][48][49][50][51][52][53][54][55][56][57][58][59][60][61][62][63][64][65][66][67][68][69][70][71][72][73][74][75][76][77][78][79][80][81][82][83][84][85][86][87][88][89][90][91][92][93][94][95][96][97][98][99][100][101][102]

Table 18-379. MCASP_PDIN

Address Offset	0x0000 001C		
Physical Address	0x4846 001C 0x4846 C01C 0x4847 001C	**Instance**	MCASP1_CFG_PER2_L4 MCASP2_CFG_PER2_L4 MCASP3_CFG_PER2_L4
Description	Pin data input register - holds the state of all the McASP pins. MCASP_PDIN allows reading the actual value of the pin, regardless of the state of MCASP_PFUNC and MCASP_PDIR. Some register bits might not be functional for all McASP instances, due to corresponding McASP pin not being pinned out on a particular device part number. Refer to device-specific DM for more information on the supported McASP features.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AFSR	RESERVED	ACLKR	AFSX	AHCLKX	ACLKX	RESERVED										AXR15	AXR14	AXR13	AXR12	AXR11	AXR10	AXR9	AXR8	AXR7	AXR6	AXR5	AXR4	AXR3	AXR2	AXR1	AXR0

Bits	Field Name	Description	Type	Reset
31	AFSR	Logic level on AFSR pin 0x0: Pin is logic low. 0x1: Pin is logic high.	R	0
30	RESERVED	Reserved	R	0
29	ACLKR	Logic level on ACLKR pin 0x0: Pin is logic low. 0x1: Pin is logic high.	R	0
28	AFSX	Logic level on AFSX pin 0x0: Pin is logic low. 0x1: Pin is logic high.	R	0
27	AHCLKX	Logic level on AHCLKX pin 0x0: Pin is logic low. 0x1: Pin is logic high.	R	0
26	ACLKX	Logic level on ACLKX pin 0x0: Pin is logic low. 0x1: Pin is logic high.	R	0
25:16	RESERVED	Reserved	R	0x000
15	AXR15	Logic level on AXR15 pin 0x0: Pin is logic low. 0x1: Pin is logic high.	R	0
14	AXR14	Logic level on AXR14 pin 0x0: Pin is logic low. 0x1: Pin is logic high.	R	0
13	AXR13	Logic level on AXR13 pin 0x0: Pin is logic low. 0x1: Pin is logic high.	R	0
12	AXR12	Logic level on AXR12 pin 0x0: Pin is logic low. 0x1: Pin is logic high.	R	0
11	AXR11	Logic level on AXR11 pin 0x0: Pin is logic low. 0x1: Pin is logic high.	R	0
10	AXR10	Logic level on AXR10 pin 0x0: Pin is logic low. 0x1: Pin is logic high.	R	0
9	AXR9	Logic level on AXR9 pin 0x0: Pin is logic low. 0x1: Pin is logic high.	R	0
8	AXR8	Logic level on AXR8 pin 0x0: Pin is logic low. 0x1: Pin is logic high.	R	0
7	AXR7	Logic level on AXR7 pin 0x0: Pin is logic low. 0x1: Pin is logic high.	R	0
6	AXR6	Logic level on AXR6 pin 0x0: Pin is logic low. 0x1: Pin is logic high.	R	0

Bits	Field Name	Description	Type	Reset
5	AXR5	Logic level on AXR5 pin 0x0: Pin is logic low. 0x1: Pin is logic high.	R	0
4	AXR4	Logic level on AXR4 pin 0x0: Pin is logic low. 0x1: Pin is logic high.	R	0
3	AXR3	Logic level on AXR3 pin 0x0: Pin is logic low. 0x1: Pin is logic high.	R	0
2	AXR2	Logic level on AXR2 pin 0x0: Pin is logic low. 0x1: Pin is logic high.	R	0
1	AXR1	Logic level on AXR1 pin 0x0: Pin is logic low. 0x1: Pin is logic high.	R	0
0	AXR0	Logic level on AXR0 pin 0x0: Pin is logic low. 0x1: Pin is logic high.	R	0

Table 18-380. Register Call Summary for Register MCASP_PDIN

Multichannel Audio Serial Port
- Burst Transfer Mode: [0]
- Time-Division Multiplexed (TDM) Transfer Mode: [1]
- MCASP_CFG Register Summary: [2]
- MCASP_CFG Register Description: [6]

Table 18-381. MCASP_PDSET

Address Offset	0x0000 001C		
Physical Address	0x4846 001C 0x4846 C01C 0x4847 001C	**Instance**	MCASP1_CFG_PER2_L4 MCASP2_CFG_PER2_L4 MCASP3_CFG_PER2_L4
Description	The pin data set register is an alias of the pin data output register (MCASP_PDOUT) for writes only. Writing a 1 to the MCASP_PDSET bit sets the corresponding bit in MCASP_PDOUT and, if MCASP_PFUNC = 1 (GPIO function) and MCASP_PDIR = 1 (output), drives a logic high on the pin. Some register bits might not be functional for all McASP instances, due to corresponding McASP pin not being pinned out on a particular device part number. Refer to device-specific DM for more information on the supported McASP features.		
Type	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AFSR	RESERVED	ACLKR	AFSX	AHCLKX	ACLKX	RESERVED										AXR15	AXR14	AXR13	AXR12	AXR11	AXR10	AXR9	AXR8	AXR7	AXR6	AXR5	AXR4	AXR3	AXR2	AXR1	AXR0

Bits	Field Name	Description	Type	Reset
31	AFSR	Allows the corresponding AFSR bit in MCASP_PDOUT to be set to a logic high without affecting other I/O pins controlled by the same port. 0x0: No effect 0x1: MCASP_PDOUT[31] bit is set to 1.	W	0
30	RESERVED	Reserved	W	0

Bits	Field Name	Description	Type	Reset
29	ACLKR	Allows the corresponding ACLKR bit in MCASP_PDOUT to be set to a logic high without affecting other I/O pins controlled by the same port. 0x0: No effect 0x1: MCASP_PDOUT[29] bit is set to 1.	W	0
28	AFSX	Allows the corresponding AFSX bit in MCASP_PDOUT to be set to a logic high without affecting other I/O pins controlled by the same port. 0x0: No effect 0x1: MCASP_PDOUT[28] bit is set to 1.	W	0
27	AHCLKX	Allows the corresponding AHCLKX bit in MCASP_PDOUT to be set to a logic high without affecting other I/O pins controlled by the same port. 0x0: No effect 0x1: MCASP_PDOUT[27] bit is set to 1.	W	0
26	ACLKX	Allows the corresponding ACLKX bit in MCASP_PDOUT to be set to a logic high without affecting other I/O pins controlled by the same port. 0x0: No effect 0x1: MCASP_PDOUT[26] bit is set to 1.	W	0
25:16	RESERVED	Reserved	W	0x000
15	AXR15	Allows the AXR15 bit in MCASP_PDOUT to be set to a logic high without affecting other I/O pins controlled by the same port. 0x0: No effect 0x1: MCASP_PDOUT[15] bit is set to 1.	W	0
14	AXR14	Allows the AXR14 bit in MCASP_PDOUT to be set to a logic high without affecting other I/O pins controlled by the same port. 0x0: No effect 0x1: MCASP_PDOUT[14] bit is set to 1.	W	0
13	AXR13	Allows the AXR13 bit in MCASP_PDOUT to be set to a logic high without affecting other I/O pins controlled by the same port. 0x0: No effect 0x1: MCASP_PDOUT[13] bit is set to 1.	W	0
12	AXR12	Allows the AXR12 bit in MCASP_PDOUT to be set to a logic high without affecting other I/O pins controlled by the same port. 0x0: No effect 0x1: MCASP_PDOUT[12] bit is set to 1.	W	0
11	AXR11	Allows the AXR11 bit in MCASP_PDOUT to be set to a logic high without affecting other I/O pins controlled by the same port. 0x0: No effect 0x1: MCASP_PDOUT[11] bit is set to 1.	W	0
10	AXR10	Allows the AXR10 bit in MCASP_PDOUT to be set to a logic high without affecting other I/O pins controlled by the same port. 0x0: No effect 0x1: MCASP_PDOUT[10] bit is set to 1.	W	0
9	AXR9	Allows the AXR9 bit in MCASP_PDOUT to be set to a logic high without affecting other I/O pins controlled by the same port. 0x0: No effect 0x1: MCASP_PDOUT[9] bit is set to 1.	W	0
8	AXR8	Allows the AXR8 bit in MCASP_PDOUT to be set to a logic high without affecting other I/O pins controlled by the same port. 0x0: No effect 0x1: MCASP_PDOUT[8] bit is set to 1.	W	0
7	AXR7	Allows the AXR7 bit in MCASP_PDOUT to be set to a logic high without affecting other I/O pins controlled by the same port. 0x0: No effect 0x1: MCASP_PDOUT[7] bit is set to 1.	W	0

Bits	Field Name	Description	Type	Reset
6	AXR6	Allows the AXR6 bit in MCASP_PDOUT to be set to a logic high without affecting other I/O pins controlled by the same port. 0x0: No effect 0x1: MCASP_PDOUT[6] bit is set to 1.	W	0
5	AXR5	Allows the AXR5 bit in MCASP_PDOUT to be set to a logic high without affecting other I/O pins controlled by the same port. 0x0: No effect 0x1: MCASP_PDOUT[5] bit is set to 1.	W	0
4	AXR4	Allows the AXR4 bit in MCASP_PDOUT to be set to a logic high without affecting other I/O pins controlled by the same port. 0x0: No effect 0x1: MCASP_PDOUT[4] bit is set to 1.	W	0
3	AXR3	Allows the AXR3 bit in MCASP_PDOUT to be set to a logic high without affecting other I/O pins controlled by the same port. 0x0: No effect 0x1: MCASP_PDOUT[3] bit is set to 1.	W	0
2	AXR2	Allows the AXR2 bit in MCASP_PDOUT to be set to a logic high without affecting other I/O pins controlled by the same port. 0x0: No effect 0x1: MCASP_PDOUT[2] bit is set to 1.	W	0
1	AXR1	Allows the AXR1 bit in MCASP_PDOUT to be set to a logic high without affecting other I/O pins controlled by the same port. 0x0: No effect 0x1: MCASP_PDOUT[1] bit is set to 1.	W	0
0	AXR0	Allows the AXR0 bit in MCASP_PDOUT to be set to a logic high without affecting other I/O pins controlled by the same port. 0x0: No effect 0x1: MCASP_PDOUT[0] bit is set to 1.	W	0

Table 18-382. Register Call Summary for Register MCASP_PDSET

Multichannel Audio Serial Port

- Burst Transfer Mode: [0]
- Time-Division Multiplexed (TDM) Transfer Mode: [1]
- MCASP_CFG Register Summary: [2]
- MCASP_CFG Register Description: [6][7][8][9]

Table 18-383. MCASP_PDCLR

Address Offset	0x0000 0020		
Physical Address	0x4846 0020 0x4846 C020 0x4847 0020	**Instance**	MCASP1_CFG_PER2_L4 MCASP2_CFG_PER2_L4 MCASP3_CFG_PER2_L4
Description	The pin data clear register is an alias of the pin data output register (MCASP_PDOUT) for writes only. Writing a 1 to the MCASP_PDCLR bit clears the corresponding bit in MCASP_PDOUT and, if MCASP_PFUNC = 1 (GPIO function) and MCASP_PDIR = 1 (output), drives a logic low on the pin. Some register bits might not be functional for all McASP instances, due to corresponding McASP pin not being pinned out on a particular device part number. Refer to device-specific DM for more information on the supported McASP features.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AFSR	RESERVED	ACLKR	AFSX	AHCLKX	ACLKX	RESERVED										AXR15	AXR14	AXR13	AXR12	AXR11	AXR10	AXR9	AXR8	AXR7	AXR6	AXR5	AXR4	AXR3	AXR2	AXR1	AXR0

Bits	Field Name	Description	Type	Reset
31	AFSR	Allows the corresponding AFSR bit in MCASP_PDOUT to be cleared to a logic low without affecting other I/O pins controlled by the same port. 0x0: No effect 0x1: MCASP_PDOUT[31] bit is cleared to 0.	RW	0
30	RESERVED	Reserved	RW	0
29	ACLKR	Allows the corresponding ACLKR bit in MCASP_PDOUT to be cleared to a logic low without affecting other I/O pins controlled by the same port. 0x0: No effect 0x1: MCASP_PDOUT[29] bit is cleared to 0.	RW	0
28	AFSX	Allows the corresponding AFSX bit in MCASP_PDOUT to be cleared to a logic low without affecting other I/O pins controlled by the same port. 0x0: No effect 0x1: MCASP_PDOUT[28] bit is cleared to 0.	RW	0
27	AHCLKX	Allows the corresponding AHCLKX bit in MCASP_PDOUT to be cleared to a logic low without affecting other I/O pins controlled by the same port. 0x0: No effect 0x1: MCASP_PDOUT[27] bit is cleared to 0.	RW	0
26	ACLKX	Allows the corresponding ACLKX bit in MCASP_PDOUT to be cleared to a logic low without affecting other I/O pins controlled by the same port. 0x0: No effect 0x1: MCASP_PDOUT[26] bit is cleared to 0.	RW	0
25:16	RESERVED	Reserved	RW	0x000
15	AXR15	Allows the AXR15 bit in MCASP_PDOUT to be cleared to a logic low without affecting other I/O pins controlled by the same port. 0x0: No effect 0x1: MCASP_PDOUT[15] bit is cleared to 0.	RW	0
14	AXR14	Allows the AXR14 bit in MCASP_PDOUT to be cleared to a logic low without affecting other I/O pins controlled by the same port. 0x0: No effect 0x1: MCASP_PDOUT[14] bit is cleared to 0.	RW	0

Bits	Field Name	Description	Type	Reset
13	AXR13	Allows the AXR13 bit in MCASP_PDOUT to be cleared to a logic low without affecting other I/O pins controlled by the same port. 0x0: No effect 0x1: MCASP_PDOUT[13] bit is cleared to 0.	RW	0
12	AXR12	Allows the AXR12 bit in MCASP_PDOUT to be cleared to a logic low without affecting other I/O pins controlled by the same port. 0x0: No effect 0x1: MCASP_PDOUT[12] bit is cleared to 0.	RW	0
11	AXR11	Allows the AXR11 bit in MCASP_PDOUT to be cleared to a logic low without affecting other I/O pins controlled by the same port. 0x0: No effect 0x1: MCASP_PDOUT[11] bit is cleared to 0.	RW	0
10	AXR10	Allows the AXR10 bit in MCASP_PDOUT to be cleared to a logic low without affecting other I/O pins controlled by the same port. 0x0: No effect 0x1: MCASP_PDOUT[10] bit is cleared to 0.	RW	0
9	AXR9	Allows the AXR9 bit in MCASP_PDOUT to be cleared to a logic low without affecting other I/O pins controlled by the same port. 0x0: No effect 0x1: MCASP_PDOUT[9] bit is cleared to 0.	RW	0
8	AXR8	Allows the AXR8 bit in MCASP_PDOUT to be cleared to a logic low without affecting other I/O pins controlled by the same port. 0x0: No effect 0x1: MCASP_PDOUT[8] bit is cleared to 0.	RW	0
7	AXR7	Allows the AXR7 bit in MCASP_PDOUT to be cleared to a logic low without affecting other I/O pins controlled by the same port. 0x0: No effect 0x1: MCASP_PDOUT[7] bit is cleared to 0.	RW	0
6	AXR6	Allows the AXR6 bit in MCASP_PDOUT to be cleared to a logic low without affecting other I/O pins controlled by the same port. 0x0: No effect 0x1: MCASP_PDOUT[6] bit is cleared to 0.	RW	0
5	AXR5	Allows the AXR5 bit in MCASP_PDOUT to be cleared to a logic low without affecting other I/O pins controlled by the same port. 0x0: No effect 0x1: MCASP_PDOUT[5] bit is cleared to 0.	RW	0
4	AXR4	Allows the AXR4 bit in MCASP_PDOUT to be cleared to a logic low without affecting other I/O pins controlled by the same port. 0x0: No effect 0x1: MCASP_PDOUT[4] bit is cleared to 0.	RW	0
3	AXR3	Allows the AXR3 bit in MCASP_PDOUT to be cleared to a logic low without affecting other I/O pins controlled by the same port. 0x0: No effect 0x1: MCASP_PDOUT[3] bit is cleared to 0.	RW	0
2	AXR2	Allows the AXR2 bit in MCASP_PDOUT to be cleared to a logic low without affecting other I/O pins controlled by the same port. 0x0: No effect 0x1: MCASP_PDOUT[2] bit is cleared to 0.	RW	0
1	AXR1	Allows the AXR1 bit in MCASP_PDOUT to be cleared to a logic low without affecting other I/O pins controlled by the same port. 0x0: No effect 0x1: MCASP_PDOUT[1] bit is cleared to 0.	RW	0

Bits	Field Name	Description	Type	Reset
0	AXR0	Allows the AXR0 bit in MCASP_PDOUT to be cleared to a logic low without affecting other I/O pins controlled by the same port. 0x0: No effect 0x1: MCASP_PDOUT[0] bit is cleared to 0.	RW	0

Table 18-384. Register Call Summary for Register MCASP_PDCLR

Multichannel Audio Serial Port

- Burst Transfer Mode: [0]
- Time-Division Multiplexed (TDM) Transfer Mode: [1]
- MCASP_CFG Register Summary: [2]
- MCASP_CFG Register Description: [6][7][8][9]

Table 18-385. MCASP_GBLCTL

Address Offset	0x0000 0044		
Physical Address	0x4846 0044 0x4846 C044 0x4847 0044	**Instance**	MCASP1_CFG_PER2_L4 MCASP2_CFG_PER2_L4 MCASP3_CFG_PER2_L4
Description	Global transmit control register - provides initialization of the transmit and receive sections. The bit fields in MCASP_GBLCTL are synchronized and latched by the transmitter and receiver corresponding clocks - ACLKX (bits [12:8]) and ACLKR (bits [4:0]), respectively. Before programming MCASP_GBLCTL, ensure that the serial clocks are running. If the corresponding external serial clocks - ACLKX and ACLKR, are not yet running, select the internal serial clock source in AHCLKXCTL, AHCLKRCTL, ACLKXCTL and ACLKRCTL before programming the MCASP_GBLCTL. Also, after programming any bits in MCASP_GBLCTL, do not proceed until reading back from MCASP_GBLCTL and verifying that the bits in MCASP_GBLCTL are latched.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																			XFRST	XSMRST	XSRCLR	XHCLKRST	XCLKRST	RESERVED				RFRST	RSMRST	RSRCLR	RHCLKRST	RCLKRST

Bits	Field Name	Description	Type	Reset
31:13	RESERVED	Reserved	RW	0x00000
12	XFRST	Transmit frame-sync generator reset enable bit 0x0: The transmit frame-sync generator is reset. 0x1: The transmit frame-sync generator is active. When released from reset, the transmit frame-sync generator begins counting serial clocks and generating frame sync as programmed.	RW	0
11	XSMRST	Transmit state-machine reset enable bit 0x0: The transmit state-machine is held in reset. AXR[n] pin state: If MCASP_PFUNC[n] = 0 and MCASP_PDIR[n] = 1, the corresponding serializer [n] drives the AXR[n] pin to the state specified for inactive time slot. 0x1: The transmit state-machine is released from reset. When released from reset, the transmit state-machine immediately transfers data from XBUF[n] to XRSR[n]. The transmit state-machine sets the underrun flag (XUNDRN) in MCASP_XSTAT, if XBUF[n] have not been preloaded with data before reset is released. The transmit state-machine also immediately begins detecting frame sync and is ready to transmit. Transmission of TDM time slot begins at slot 0 after reset is released.	RW	0

Bits	Field Name	Description	Type	Reset
10	XSRCLR	Transmit serializer clear enable bit. By clearing and then setting this bit, the transmit buffer is flushed to an empty state (XDATA = 1). If XSMRST = 1, XSRCLR = 1, XDATA = 1, and XBUF is not loaded with new data before the start of the next active time slot, an underrun occurs. 0x0: The transmit serializer is cleared. 0x1: The transmit serializer is active. When the transmit serializer is first taken out of reset (XSRCLR changes from 0 to 1), the transmit data ready bit (XDATA) in MCASP_XSTAT is set to indicate XBUF is ready to be written.	RW	0
9	XHCLKRST	Transmit high-frequency clock divider reset enable bit 0x0: The transmitter high-frequency clock divider is held in reset and passes through its input as divide-by-1. 0x1: The transmitter high-frequency clock divider is running.	RW	0
8	XCLKRST	Transmit clock divider reset enable bit 0x0: The transmit clock divider is held in reset. When the clock divider is in reset, it passes through a divide-by-1 of its input. 0x1: The transmit clock divider is running.	RW	0
7:5	RESERVED	Reserved	RW	0x0
4	RFRST	Receive frame sync generator reset enable bit. 0x0: Receive frame sync generator is reset. 0x1: Receive frame sync generator is active. When released from reset, the receive frame sync generator begins counting serial clocks and generating frame sync as programmed.	RW	0
3	RSMRST	Receive state machine reset enable bit. 0x0: Receive state machine is held in reset. 0x1: Receive state machine is released from reset. When released from reset, the receive state machine immediately begins detecting frame sync and is ready to receive. Receive TDM time slot begins at slot 0 after reset is released.	RW	0
2	RSRCLR	Receive serializer clear enable bit. By clearing then setting this bit, the receive buffer is flushed. 0x0: Receive serializers are cleared. 0x1: Receive serializers are active.	RW	0
1	RHCLKRST	Receive high-frequency clock divider reset enable bit. 0x0: Receive high-frequency clock divider is held in reset and passes through its input as divide-by-1. 0x1: Receive high-frequency clock divider is running.	RW	0
0	RCLKRST	Receive clock divider reset enable bit. 0x0: Receive clock divider is held in reset. When the clock divider is in reset, it passes through a divide-by-1 of its input. 0x1: Receive clock divider is running.	RW	0

Table 18-386. Register Call Summary for Register MCASP_GBLCTL

Multichannel Audio Serial Port

- MCASP Software Reset: [0][1]
- Burst Transfer Mode: [2]
- Time-Division Multiplexed (TDM) Transfer Mode: [3]
- DIT Transfer Mode: [4]
- MCASP Global Initialization: [5][6][7][8][9][10][11][12][13][14][15][16][17][18][19][20][21][22][23][24][25]
- MCASP Transmission Modes: [26][27]
- MCASP Reception Modes: [28][29]
- MCASP_CFG Register Summary: [30]
- MCASP_CFG Register Description: [34][35][36][37][38][39]

Table 18-387. MCASP_AMUTE

Address Offset	0x0000 0048		
Physical Address	0x4846 0048 0x4846 C048 0x4847 0048	**Instance**	MCASP1_CFG_PER2_L4 MCASP2_CFG_PER2_L4 MCASP3_CFG_PER2_L4
Description	Mute control register - Controls the McASP mute output pin - AMUTE (Not implemented at device level)		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															

Bits	Field Name	Description	Type	Reset
31:0	RESERVED	Reserved	RW	0x0000 0000

Table 18-388. Register Call Summary for Register MCASP_AMUTE

Multichannel Audio Serial Port

- Burst Transfer Mode: [0]
- Time-Division Multiplexed (TDM) Transfer Mode: [1]
- DIT Transfer Mode: [2]
- MCASP_CFG Register Summary: [3]

Table 18-389. MCASP_LBCTL

Address Offset	0x0000 004C		
Physical Address	0x4846 004C 0x4846 C04C 0x4847 004C	**Instance**	MCASP1_CFG_PER2_L4 MCASP2_CFG_PER2_L4 MCASP3_CFG_PER2_L4
Description	The digital loopback control register (MCASP_LBCTL) controls the internal (McASP module)-level and chip-level loopback settings of the McASP in TDM mode. Note that loopback is NOT supported if McASP is configured in DIT mode.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																											IOLBEN	MODE			ORD	DLBEN

Bits	Field Name	Description	Type	Reset
31:5	RESERVED	Reserved	RW	0x000 0000
4	IOLBEN	If DLBEN=0b1, the IOLBEN bit selects between **internal-level (McASP module-level)** and **chip I/O-level** loopback modes. IOLBEN bit value is irrelevant, If DLBEN=0b0. 0x0: McASP internal loopback mode enabled. This selects a direct loopback between corresponding McASP AXRn and AXRn+1 pins, bypassing device pad I/O buffers. 0x1: Chip I/O-level loopback mode enabled. The McASP data is looped back through the device pad I/O buffers.	RW	0
3:2	MODE	Loopback generator mode bits. 0x0: RESERVED 0x1: MODE must be set to 0x1 when McASP operates in loopback mode (DLBEN =0b1). This is necessary to allow transmit clock and frame sync generators to be used by both transmit and receive sections. 0x2, 0x3: Reserved	RW	0x0

Bits	Field Name	Description	Type	Reset
1	ORD	Loopback order bit when loopback mode is enabled (DLBEN = 1). 0x0: Odd serializers N + 1 transmit to even serializers N that receive. The corresponding serializers must be programmed properly. 0x1: Even serializers N transmit to odd serializers N+1 that receive. The corresponding serializers must be programmed properly.	RW	0
0	DLBEN	Loop back mode enable bit. 0x0: Loop back mode is disabled (normal McASP operation). 0x1: Loop back is enabled (TDM mode only). Loopback type is selected in IOLBEN bit.	RW	0

Table 18-390. Register Call Summary for Register MCASP_LBCTL

Multichannel Audio Serial Port
- Burst Transfer Mode: [0]
- Time-Division Multiplexed (TDM) Transfer Mode: [1]
- Loopback Modes: [2][3][4][5][6]
- Loopback Mode Configurations: [7][8][9][10]
- MCASP Global Initialization: [11][12]
- MCASP_CFG Register Summary: [13]
- MCASP_CFG Register Description: [17]

Table 18-391. MCASP_TXDITCTL

Address Offset	0x0000 0050		
Physical Address	0x4846 0050 0x4846 C050 0x4847 0050	**Instance**	MCASP1_CFG_PER2_L4 MCASP2_CFG_PER2_L4 MCASP3_CFG_PER2_L4
Description	Transmit DIT mode control register, controls DIT operations of the McASP		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												VB	VA	RESERVED	DITEN

Bits	Field Name	Description	Type	Reset
31:4	RESERVED	Reserved	RW	0x0000000
3	VB	Valid bit for odd time slots (DIT right subframe). 0x0: V bit is 0 during odd DIT subframes. 0x1: V bit is 1 during odd DIT subframes.	RW	0
2	VA	Valid bit for even time slots (DIT left subframe). 0x0: V bit is 0 during even DIT subframes. 0x1: V bit is 1 during even DIT subframes.	RW	0
1	RESERVED	Reserved	RW	0
0	DITEN	DIT mode enable bit 0x0: DIT mode is disabled. 0x1: DIT mode is enabled. Transmitter operates in DIT encoded mode.	RW	0

Table 18-392. Register Call Summary for Register MCASP_TXDITCTL

Multichannel Audio Serial Port

- Synchronous and Asynchronous Transmit and Receive Operations: [0][1]
- Burst Transfer Mode: [2]
- Time-Division Multiplexed (TDM) Transfer Mode: [3]
- DIT Transfer Mode: [4]
- MCASP Global Initialization: [5][6][7][8]
- MCASP_CFG Register Summary: [9]

Table 18-393. MCASP_GBLCTLR

Address Offset	0x0000 0060		
Physical Address	0x4846 0060 0x4846 C060 0x4847 0060	**Instance**	MCASP1_CFG_PER2_L4 MCASP2_CFG_PER2_L4 MCASP3_CFG_PER2_L4
Description	Alias of GBLCTL. When writing to this register, only the TRANSMIT bits of GBLCTL are affected (This means GBLCTL bits 8,9,10,11,12). Reads return GBLCTL		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																			XFRST	XSMRST	XSRCLR	XHCLKRST	XCLKRST	RESERVED				RFRST	RSMRST	RSRCLR	RHCLKRST	RCLKRST

Bits	Field Name	Description	Type	Reset
31:13	RESERVED		RW	0x0
12	XFRST	Frame sync generator reset 0x0: RESET 0x1: ACTIVE	R	0x0
11	XSMRST	XMT state machine reset 0x0: RESET 0x1: ACTIVE	R	0x0
10	XSRCLR	XMT serializer clear 0x0: CLEAR 0x1: ACTIVE	R	0x0
9	XHCLKRST	XMT High Freq. clk Divider 0x0: RESET 0x1: ACTIVE	R	0x0
8	XCLKRST	XMT clock divder reset 0x0: RESET 0x1: ACTIVE	R	0x0
7:5	RESERVED		RW	0x0
4	RFRST	Frame sync generator reset 0x0: RESET 0x1: ACTIVE	RW	0x0
3	RSMRST	RCV state machine reset 0x0: RESET 0x1: ACTIVE	RW	0x0

Bits	Field Name	Description	Type	Reset
2	RSRCLR	RCV serializer clear 0x0: CLEAR 0x1: ACTIVE	RW	0x0
1	RHCLKRST	RCV High Freq. clk Divider 0x0: RESET 0x1: ACTIVE	RW	0x0
0	RCLKRST	RCV clock divder reset 0x0: RESET 0x1: ACTIVE	RW	0x0

Table 18-394. Register Call Summary for Register MCASP_GBLCTLR

Multichannel Audio Serial Port
- MCASP_CFG Register Summary: [0]

Table 18-395. MCASP_RXMASK

Address Offset	0x0000 0064		
Physical Address	0x4846 0064 0x4846 C064 0x4847 0064	**Instance**	MCASP1_CFG_PER2_L4 MCASP2_CFG_PER2_L4 MCASP3_CFG_PER2_L4
Description	The receive format unit bit mask register (MCASP_RXMASK) determines which bits of the received data are masked off and padded with a known value before being read by the CPU.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RMASK[31:0]																															

Bits	Field Name	Description	Type	Reset
31: 0	RMASK[31:0]	Receive data mask enable bit. 0x0: Corresponding bit of receive data (after passing through reverse and rotate units) is masked out and then padded with the selected bit pad value (RPAD and RPBIT bits in RFMT). 0x1: Corresponding bit of receive data (after passing through reverse and rotate units) is returned to CPU or DMA.	RW	0

Table 18-396. Register Call Summary for Register MCASP_RXMASK

Multichannel Audio Serial Port
- Burst Transfer Mode: [0]
- Time-Division Multiplexed (TDM) Transfer Mode: [1]
- MCASP Global Initialization: [2]
- MCASP_CFG Register Summary: [3]
- MCASP_CFG Register Description: [7]

Table 18-397. MCASP_RXFMT

Address Offset	0x0000 0068		
Physical Address	0x4846 0068 0x4846 C068 0x4847 0068	**Instance**	MCASP1_CFG_PER2_L4 MCASP2_CFG_PER2_L4 MCASP3_CFG_PER2_L4
Description	The receive bit stream format register (MCASP_RXFMT) configures the receive data format.		

Table 18-397. MCASP_RXFMT (continued)

Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														RDATDLY		RRVRS	RPAD			RPBIT				RSSZ				RBUSEL	RROT		

Bits	Field Name	Description	Type	Reset
31:18	RESERVED		RW	0x0000
17:16	RDATDLY	Receive Frame sync delay of AXR[n] 0x0: 0-bit delay. The first receive data bit, AXR[n], occurs in same ACLKR cycle as the receive frame sync (AFSR). 0x1: 1-bit delay. The first receive data bit, AXR[n], occurs one ACLKR cycle after the receive frame sync (AFSR). 0x2: 2-bit delay. The first receive data bit, AXR[n], occurs two ACLKR cycles after the receive frame sync (AFSR). 0x3: Reserved	RW	0x0
15	RRVRS	Receive serial bitstream order 0x0: Bitstream is LSB first. No bit reversal is performed in receive format bit reverse unit. 0x1: Bitstream is MSB first. Bit reversal is performed in receive format bit reverse unit.	RW	0
14:13	RPAD	Pad value for extra bits in slot not belonging to the word. This field only applies to bits when RMASK[n] = 0. 0x0: Pad extra bits with 0. 0x1: Pad extra bits with 1. 0x2: Pad extra bits with one of the bits from the word as specified by RPBIT bits. 0x3: Reserved	RW	0x0
12:8	RPBIT	RPBIT value determines which bit (as read by the CPU from RBUF[n]) is used to pad the extra bits. This field only applies when RPAD = 2h. 0x0: Pad with value of bit RBUFn[0]. 0x01 - 0x1F: Pad with value of the bit positioned within the range RBUFn[31:1].	RW	0x00
7:4	RSSZ	Receive slot size. 0x0 - 0x2: Reserved 0x3: Slot size is 8 bits 0x4: Reserved 0x5: Slot size is 12 bits 0x6: Reserved 0x7: Slot size is 16 bits 0x8: Reserved 0x9: Slot size is 20 bits 0xA: Reserved 0xB: Slot size is 24 bits 0xC: Reserved 0xD: Slot size is 28 bits 0xE: Reserved 0xF: Slot size is 32 bits	RW	0x0

Bits	Field Name	Description	Type	Reset
3	RBUSEL	Selects whether reads from serializer buffer RBUF[n] originate from the peripheral configuration CFG port or the DATA port. 0x0: Reads from XRBUF[n] originate on DATA port. Reads from XRBUF[n] on the peripheral configuration port are ignored. 0x1: Reads from XRBUF[n] originate on peripheral configuration port. Reads from XRBUF[n] on the DATA port are ignored.	RW	0
2:0	RROT	Right-rotation value for receive rotate right format unit. 0x0: Rotate right by 0 (no rotation). 0x1: Rotate right by 4 bit positions. 0x2: Rotate right by 8 bit positions. 0x3: Rotate right by 12 bit positions. 0x4: Rotate right by 16 bit positions. 0x5: Rotate right by 20 bit positions. 0x6: Rotate right by 24 bit positions. 0x7: Rotate right by 28 bit positions.	RW	0x0

Table 18-398. Register Call Summary for Register MCASP_RXFMT

Multichannel Audio Serial Port

- Frame-Sync Generator: [0]
- Format Units: [1][2][3]
- Receive Format Unit: [4][5][6]
- State-Machines: [7]
- Burst Transfer Mode: [8][9]
- Time-Division Multiplexed (TDM) Transfer Mode: [10]
- Data Ready Status and Event/Interrupt Generation: [11][12]
- MCASP Global Initialization: [13][14][15][16][17][18][19][20]
- MCASP Reception Modes: [21][22]
- MCASP Event Servicing: [23]
- MCASP_CFG Register Summary: [24][28]
- MCASP_CFG Register Description: [29]
- MCASP_DAT Register Summary: [30]
- MCASP_DAT Register Description: [31]

Table 18-399. MCASP_RXFMCTL

Address Offset	0x0000 006C		
Physical Address	0x4846 006C 0x4846 C06C 0x4847 006C	**Instance**	MCASP1_CFG_PER2_L4 MCASP2_CFG_PER2_L4 MCASP3_CFG_PER2_L4
Description	The receive frame sync control register (MCASP_RXFMCTL) configures the receive frame sync (AFSR).		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RMOD								RESERVED			FRWID	RESERVED		FSRM	FSRP

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		RW	0x0000
15:7	RMOD	Receive frame sync mode select bits. 0x0: Burst mode 0x1: Reserved 0x2: 2-slot TDM mode (I2S receive mode) 0x3 - 0x20: 3-slot TDM to 32-slot TDM mode 0x21 - 0x17F: Reserved 0x180: 384-slot TDM (external DIR IC inputting 384-slot DIR frames to McASP) 0x181 - 0x1FF: Reserved	RW	0x000
6:5	RESERVED		RW	0x0
4	FRWID	Receive frame sync width select bit indicates the width of the receive frame sync (AFSR) during its active period. 0x0: Single bit 0x1: Single word. Single word is not supported if RMOD is set to burst mode.	RW	0
3:2	RESERVED		RW	0x0
1	FSRM	Receive frame sync generation select bit. 0x0: Externally-generated receive frame sync 0x1: Internally-generated receive frame sync	RW	0
0	FSRP	Receive frame sync polarity select bit. 0x0: A rising edge on receive frame sync (AFSR) indicates the beginning of a frame. 0x1: A falling edge on receive frame sync (AFSR) indicates the beginning of a frame.	RW	0

Table 18-400. Register Call Summary for Register MCASP_RXFMCTL

Multichannel Audio Serial Port

- Frame-Sync Generator: [0][1][2][3][4][5][6]
- Burst Transfer Mode: [7][8]
- Time-Division Multiplexed (TDM) Transfer Mode: [9][10]
- Loopback Mode Configurations: [11]
- MCASP Global Initialization: [12][13][14][15][16]
- MCASP_CFG Register Summary: [17]
- MCASP_CFG Register Description: [21]

Table 18-401. MCASP_ACLKRCTL

Address Offset	0x0000 0070		
Physical Address	0x4846 0070 0x4846 C070 0x4847 0070	**Instance**	MCASP1_CFG_PER2_L4 MCASP2_CFG_PER2_L4 MCASP3_CFG_PER2_L4
Description	The receive clock control register (MCASP_ACLKRCTL) configures the receive bit clock (ACLKR) and the receive clock generator.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED											BUSY	DIVBUSY	ADJBUSY	CLKRADJ		RESERVED								CLKRP	RESERVED	CLKRM	CLKRDIV				

Bits	Field Name	Description	Type	Reset
31:21	RESERVED		RW	0x000000
20	BUSY	Status: logical OR of DIVBUSY, ADJBUSY. Not supported. 0x0: NOTBUSY 0x1: BUSY	RW	0
19	DIVBUSY	Status: divide ratio change in progress? Not supported. 0x0: NOTBUSY 0x1: BUSY	RW	0x0
18	ADJBUSY	Status: one-shot adjustment in progress? Not supported. 0x0: NOTBUSY 0x1: BUSY	RW	0x0
17:16	CLKRADJ	CLKRDIV one-shot adjustment. Not supported. Bit field must always be written as 0x0.If CLKRDIV is set such that there are “m” input clocks per one output clock, then for one output cycle: 00 = (m+0) input clocks per output clock, i.e. no adjustment 01 = (m-1) input clocks per output clock 10 = (m+1) input clocks per output clock 11 = (m+0) input clocks per output clock, i.e. no adjustment NOTE: writes to these bits are ineffective if CLKADJEN:ENABLE bit is set to 0b. These bits are ALWAYS read back as zero.	W	0x0
15:8	RESERVED		RW	0x0
7	CLKRP	Receive bitstream clock polarity select bit. Note that this bitfield does not have any effect, if MCASP_ACLKXCTL[6] ASYNC = 0 0x0: Falling edge. Receiver samples data on the falling edge of the serial clock, so the external transmitter driving this receiver must shift data out on the rising edge of the serial clock. 0x1: Rising edge. Receiver samples data on the rising edge of the serial clock, so the external transmitter driving this receiver must shift data out on the falling edge of the serial clock.	RW	0
6	RESERVED		RW	0
5	CLKRM	Receive bit clock source bit. Note that this bitfield does not have any effect, if MCASP_ACLKXCTL[6] ASYNC = 0 0x0: External receive clock source from ACLKR pin. 0x1: Internal receive clock source from output of programmable bit clock divider.	RW	1
4:0	CLKRDIV	Receive bit clock divide ratio bits determine the divide-down ratio from AHCLKR to ACLKR. Note that this bitfield does not have any effect, if MCASP_ACLKXCTL[6] ASYNC = 0. 0x0: Divide-by-1 0x1: Divide-by-2 0x2 - 0x1F: Divide-by-3 to divide-by-32	RW	0x00

Table 18-402. Register Call Summary for Register MCASP_ACLKRCTL

Multichannel Audio Serial Port

- MCASP Receive Clock: [0][1][2][3]
- Synchronous and Asynchronous Transmit and Receive Operations: [4]
- Burst Transfer Mode: [5]
- Time-Division Multiplexed (TDM) Transfer Mode: [6]
- MCASP Global Initialization: [7][8][9][10][11][12][13]
- MCASP_CFG Register Summary: [14]
- MCASP_CFG Register Description: [18]

Table 18-403. MCASP_AHCLKRCTL

Address Offset	0x0000 0074		
Physical Address	0x4846 0074 0x4846 C074 0x4847 0074	**Instance**	MCASP1_CFG_PER2_L4 MCASP2_CFG_PER2_L4 MCASP3_CFG_PER2_L4
Description	The receive high-frequency clock control register (MCASP_AHCLKRCTL) configures the receive high-frequency master clock (AHCLKR) and the receive clock generator.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED											BUSY	DIVBUSY	ADJBUSY	HCLKRADJ		HCLKRM	HCLKRP	RESERVED		HCLKRDIV											

Bits	Field Name	Description	Type	Reset
31:21	RESERVED		RW	0x0000
20	BUSY	Status: logical OR of DIVBUSY, ADJBUSY. Not supported. 0x0: NOTBUSY 0x1: BUSY	RW	0x0
19	DIVBUSY	Status: divide ratio change in progress? Not supported. 0x0: NOTBUSY 0x1: BUSY	RW	0x0
18	ADJBUSY	Status: one-shot adjustment in progress? Not supported. 0x0: NOTBUSY 0x1: BUSY	RW	0x0
17:16	HCLKRADJ	HCLKRDIV one-shot adjustment. Not supported. Bit field must always be written as 0x0. If HCLKRDIV is set such that there are “m” input clocks per one output clock, then for one output cycle: 00 = (m+0) input clocks per output clock, i.e. no adjustment 01 = (m-1) input clocks per output clock 10 = (m+1) input clocks per output clock 11 = (m+0) input clocks per output clock, i.e. no adjustment NOTE: writes to these bits are ineffective if CLKADJEN:ENABLE bit is set to 0b. These bits are ALWAYS read back as zero.	W	0x0
15	HCLKRM	High Freq. RCV clock Source 0x0: EXTERNAL 0x1: INTERNAL	RW	0x1

Bits	Field Name	Description	Type	Reset
14	HCLKRP	Receive bitstream high-frequency clock polarity select bit. 0x0: Not inverted. AHCLKR is not inverted before programmable bit clock divider. 0x1: Inverted. AHCLKR is inverted before programmable bit clock divider.	RW	0
13:12	RESERVED		RW	0x0
11:0	HCLKRDIV	Receive high-frequency clock divide ratio bits determine the divide-down ratio from AUXCLK to AHCLKR. 0x0: Divide-by-1 0x1: Divide-by-2 0x2 - 0xFFF: Divide-by-3 to divide-by-4096	RW	0x000

Table 18-404. Register Call Summary for Register MCASP_AHCLKRCTL

Multichannel Audio Serial Port
- MCASP Receive Clock: [0][1][2][3]
- Burst Transfer Mode: [4]
- Time-Division Multiplexed (TDM) Transfer Mode: [5]
- MCASP Global Initialization: [6][7][8][9]
- MCASP_CFG Register Summary: [10]
- MCASP_CFG Register Description: [14]

Table 18-405. MCASP_RXTDM

Address Offset	0x0000 0078		
Physical Address	0x4846 0078 0x4846 C078 0x4847 0078	**Instance**	MCASP1_CFG_PER2_L4 MCASP2_CFG_PER2_L4 MCASP3_CFG_PER2_L4
Description	The receive TDM time slot register (MCASP_RXTDM) specifies which TDM time slot the receiver is active.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RTDMS[31:0]																															

Bits	Field Name	Description	Type	Reset
31:0	RTDMS[31:0]	Receiver mode during TDM time slot n. 0x0: Receive TDM time slot n is inactive. The receive serializer does not shift in data during this slot. 0x1: Receive TDM time slot n is active. The receive serializer shifts in data during this slot.	RW	0

Table 18-406. Register Call Summary for Register MCASP_RXTDM

Multichannel Audio Serial Port
- TDM Sequencers: [0]
- Burst Transfer Mode: [1]
- Time-Division Multiplexed (TDM) Transfer Mode: [2][3][4][5][6]
- Data Ready Status and Event/Interrupt Generation: [7]
- MCASP Global Initialization: [8]
- MCASP_CFG Register Summary: [9]
- MCASP_CFG Register Description: [13]

Table 18-407. MCASP_EVTCTLR

Address Offset	0x0000 007C		
Physical Address	0x4846 007C 0x4846 C07C 0x4847 007C	**Instance**	MCASP1_CFG_PER2_L4 MCASP2_CFG_PER2_L4 MCASP3_CFG_PER2_L4
Description	Receiver Interrupt control register - controls generation of the McASP receive interrupt (RINT). When the register bit(s) is set to 1, the occurrence of the enabled McASP condition(s) generates RINT.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								RSTAFRM	RESERVED	RDATA	RLAST	RDMAERR	RCKFAIL	RSYNCERR	ROVRN

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Reserved	RW	0x000000
7	RSTAFRM	Receive start of frame interrupt enable bit 0x0: Interrupt is disabled. A receive-start-of-frame interrupt does not generate a McASP receive interrupt (RINT). 0x1: Interrupt is enabled. A receive-start-of-frame interrupt generates a McASP receive interrupt (RINT).	RW	0
6	RESERVED	Reserved	RW	0
5	RDATA	Receive data-ready interrupt enable bit 0x0: Interrupt is disabled. A receive data-ready interrupt does not generate a McASP receive interrupt (RINT). 0x1: Interrupt is enabled. A receive data-ready interrupt generates a McASP receive interrupt (RINT).	RW	0
4	RLAST	Receive last slot interrupt enable bit 0x0: Interrupt is disabled. A receive-last-slot interrupt does not generate a McASP receive interrupt (RINT). 0x1: Interrupt is enabled. A receive-last-slot interrupt generates a McASP receive interrupt (RINT).	RW	0
3	RDMAERR	Receive DMA error interrupt enable bit 0x0: Interrupt is disabled. A receive DMA error interrupt does not generate a McASP receive interrupt (RINT). 0x1: Interrupt is enabled. A receive DMA error interrupt generates a McASP receive interrupt (RINT).	RW	0
2	RCKFAIL	Receive clock failure interrupt enable bit 0x0: Interrupt is disabled. A receive clock failure interrupt does not generate a McASP receive interrupt (RINT). 0x1: Interrupt is enabled. A receive clock failure interrupt generates a McASP receive interrupt (RINT).	RW	0
1	RSYNCERR	Unexpected receive frame-sync interrupt enable bit 0x0: Interrupt is disabled. An unexpected receive frame-sync interrupt does not generate a McASP receive interrupt (RINT). 0x1: Interrupt is enabled. An unexpected receive frame-sync interrupt generates a McASP receive interrupt (RINT).	RW	0
0	ROVRN	Receiver overrun interrupt enable bit 0x0: Interrupt is disabled. A receiver overrun interrupt does not generate a McASP receive interrupt (RINT). 0x1: Interrupt is enabled. A receiver overrun interrupt generates a McASP receive interrupt (RINT).	RW	0

Table 18-408. Register Call Summary for Register MCASP_EVTCTLR

Multichannel Audio Serial Port

- Burst Transfer Mode: [0]
- Data Ready Status and Event/Interrupt Generation: [1][2]
- MCASP Events and Interrupt Requests: [3][4][5][6][7][8][9][10]
- Receive Data Ready Event and Interrupt: [11]
- Error Interrupt: [12]
- Multiple Interrupts: [13]
- Clock Failure Detection: [14]
- MCASP Reception Modes: [15][16][17][18][19][20][21][22][23][24][25][26][27][28][29]
- MCASP_CFG Register Summary: [30]

Table 18-409. MCASP_RXSTAT

Address Offset	0x0000 0080		
Physical Address	0x4846 0080 0x4846 C080 0x4847 0080	**Instance**	MCASP1_CFG_PER2_L4 MCASP2_CFG_PER2_L4 MCASP3_CFG_PER2_L4
Description	The receiver status register (MCASP_RXSTAT) provides the receiver status and receive TDM time slot number.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								RERR	RDMAERR	RSTAFRM	RDATA	RLAST	RTDMSLOT	RCKFAIL	RSYNCERR	ROVRN

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		RW	0x00 0000
8	RERR	RERR bit always returns a logic-OR of: ROVRN | RSYNCERR | RCKFAIL | RDMAERR Allows a single bit to be checked to determine if a receiver error has occurred. 0x0: No errors have occurred. 0x1: An error has occurred.	RW	0
7	RDMAERR	Receive DMA error flag. RDMAERR is set when the CPU or DMA reads more serializers through the DMA port in a given time slot than were programmed as receivers. This bit is cleared by writing a 1 to this bit. Writing a 0 to this bit has no effect. 0x0: Receive DMA error did not occur. 0x1: Receive DMA error did occur.	RW	0
6	RSTAFRM	Receive start of frame flag. This bit is cleared by writing a 1 to this bit. Writing a 0 to this bit has no effect. 0x0: No new receive frame sync (AFSR) is detected. 0x1: A new receive frame sync (AFSR) is detected.	RW	0
5	RDATA	Receive data ready flag. This bit is cleared by writing a 1 to this bit. Writing a 0 to this bit has no effect. 0x0: No new data in RBUF. 0x1: Data is transferred from XRSR to RBUF and ready to be serviced by the CPUs or DMA. When RDATA is set, it always causes a DMA event (AREVT).	RW	0

Bits	Field Name	Description	Type	Reset
4	RLAST	Receive last slot flag. RLAST is set along with RDATA, if the current slot is the last slot in a frame. This bit is cleared by writing a 1 to this bit. Writing a 0 to this bit has no effect. 0x0: Current slot is not the last slot in a frame. 0x1: Current slot is the last slot in a frame. RDATA is also set.	RW	0
3	RTDMSLOT	Returns the LSB of RSLOT. Allows a single read of MCASP_RXSTAT to determine whether the current TDM time slot is even or odd. 0x0: Current TDM time slot is odd. 0x1: Current TDM time slot is even.	RW	0
2	RCKFAIL	Receive clock failure flag. RCKFAIL is set when the receive clock failure detection circuit reports an error. This bit is cleared by writing a 1 to this bit. Writing a 0 to this bit has no effect. 0x0: Receive clock failure did not occur. 0x1: Receive clock failure did occur.	RW	0
1	RSYNCERR	Unexpected receive frame sync flag. RSYNCERR is set when a new receive frame sync (AFSR) occurs before it is expected. This bit is cleared by writing a 1 to this bit. Writing a 0 to this bit has no effect. 0x0: Unexpected receive frame sync did not occur. 0x1: Unexpected receive frame sync did occur.	RW	0
0	ROVRN	Receiver overrun flag. ROVRN is set when the receive serializer is instructed to transfer data from XRSR to RBUF, but the former data in RBUF has not yet been read by the CPU or DMA. This bit is cleared by writing a 1 to this bit. Writing a 0 to this bit has no effect. 0x0: Receiver overrun did not occur. 0x1: Receiver overrun did occur.	RW	0

Table 18-410. Register Call Summary for Register MCASP_RXSTAT

Multichannel Audio Serial Port

- State-Machines: [0]
- Data Ready Status and Event/Interrupt Generation: [1][2][3]
- MCASP Events and Interrupt Requests: [4][5][6][7][8][9][10][11][12][13][14]
- Receive Data Ready Event and Interrupt: [15][16]
- Error Interrupt: [17][18]
- Multiple Interrupts: [19][20]
- Buffer Overrun Error-Receiver: [21]
- DATA Port Error - Receiver: [22][23][24]
- Clock Failure Detection: [25][26]
- MCASP Reception Modes: [27][28]
- MCASP Event Servicing: [29][30]
- MCASP_CFG Register Summary: [31]
- MCASP_CFG Register Description: [35][36][37]

Table 18-411. MCASP_RXTDMSLOT

Address Offset	0x0000 0084		
Physical Address	0x4846 0084 0x4846 C084 0x4847 0084	**Instance**	MCASP1_CFG_PER2_L4 MCASP2_CFG_PER2_L4 MCASP3_CFG_PER2_L4
Description	The current receive TDM time slot register (MCASP_RXTDMSLOT) indicates the current time slot for the receive data frame.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								RSLOTCNT							

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x00 0000
8:0	RSLOTCNT	0x0 - 0x17F: Current receive time slot count. Legal values: 0 to 383 (17Fh). TDM function is not supported for > 32 time slots. However, TDM time slot counter may count to 383 when used to receive a DIR block (transferred over TDM format).	R	0x000

Table 18-412. Register Call Summary for Register MCASP_RXTDMSLOT

Multichannel Audio Serial Port

- MCASP_CFG Register Summary: [0]
- MCASP_CFG Register Description: [4]

Table 18-413. MCASP_RXCLKCHK

Address Offset	0x0000 0088		
Physical Address	0x4846 0088 0x4846 C088 0x4847 0088	**Instance**	MCASP1_CFG_PER2_L4 MCASP2_CFG_PER2_L4 MCASP3_CFG_PER2_L4
Description	The receive clock check control register (RCLKCHK) configures the receive clock failure detection circuit.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RCNT								RMAX								RMIN								RESERVED				RPS			

Bits	Field Name	Description	Type	Reset
31:24	RCNT	0x0 - 0xFF: Receive clock count value (from previous measurement). The clock circuit continually counts the number of interface clocks for every 32 receive high-frequency master clock (AHCLKR) signals, and stores the count in RCNT until the next measurement is taken.	R	0x00
23:16	RMAX	0x00-0xFF: Receive clock maximum boundary. This 8-bit unsigned value sets the maximum allowed boundary for the clock check counter after 32 receive high-frequency master clock (AHCLKR) signals have been received. If the current counter value is greater than RMAX after counting 32 AHCLKR signals, RCKFAIL in MCASP_RXSTAT is set. The comparison is performed using unsigned arithmetic.	RW	0x00
15:8	RMIN	0x00 - 0xFF: Receive clock minimum boundary. This 8-bit unsigned value sets the minimum allowed boundary for the clock check counter after 32 receive high-frequency master clock (AHCLKR) signals have been received. If RCNT is less than RMIN after counting 32 AHCLKR signals, RCKFAIL in RSTAT is set. The comparison is performed using unsigned arithmetic.	RW	0x00
7:4	RESERVED		RW	0x0

Bits	Field Name	Description	Type	Reset
3:0	RPS	Receive clock check prescaler value. 0x0: McASP interface clock divided by 1 0x1: McASP interface clock divided by 2 0x2: McASP interface clock divided by 4 0x3: McASP interface clock divided by 8 0x4: McASP interface clock divided by 16 0x5: McASP interface clock divided by 32 0x6: McASP interface clock divided by 64 0x7: McASP interface clock divided by 128 0x8: McASP interface clock divided by 256 0x9 - 0xF: Reserved	RW	0x0

Table 18-414. Register Call Summary for Register MCASP_RXCLKCHK

Multichannel Audio Serial Port
- Burst Transfer Mode: [0]
- Time-Division Multiplexed (TDM) Transfer Mode: [1]
- Clock Failure Detection: [2][3]
- MCASP_CFG Register Summary: [4]

Table 18-415. MCASP_REVTCTL

Address Offset	0x0000 008C		
Physical Address	0x4846 008C 0x4846 C08C 0x4847 008C	**Instance**	MCASP1_CFG_PER2_L4 MCASP2_CFG_PER2_L4 MCASP3_CFG_PER2_L4
Description	Receiver DMA event control register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															RDATDMA

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved	RW	0x0000 0000
0	RDATDMA	Receive data DMA request enable bit. 0x0: The receive data DMA request is enabled. 0x1: The receive data DMA request is disabled.	RW	0

Table 18-416. Register Call Summary for Register MCASP_REVTCTL

Multichannel Audio Serial Port
- Data Ready Status and Event/Interrupt Generation: [0][1]
- DMA Requests: [2]
- MCASP Reception Modes: [3][4]
- MCASP_CFG Register Summary: [5]

Table 18-417. MCASP_GBLCTLX

Address Offset	0x0000 00A0		
Physical Address	0x4846 00A0 0x4846 C0A0 0x4847 00A0	**Instance**	MCASP1_CFG_PER2_L4 MCASP2_CFG_PER2_L4 MCASP3_CFG_PER2_L4
Description	Alias of GBLCTL. When writing to this register, only the TRANSMIT bits of GBLCTL are affected (This means GBLCTL bits 8,9,10,11,12.). Reads return GBLCTL.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																			XFRST	XSMRST	XSRCLR	XHCLKRST	XCLKRST	RESERVED			RFRST	RSMRST	RSRCLKR	RHCLKRST	RCLKRST

Bits	Field Name	Description	Type	Reset
31:13	RESERVED		RW	0x0
12	XFRST	Frame sync generator reset 0x0: RESET 0x1: ACTIVE	RW	0x0
11	XSMRST	XMT state machine reset 0x0: RESET 0x1: ACTIVE	RW	0x0
10	XSRCLR	XMT serializer clear 0x0: CLEAR 0x1: ACTIVE	RW	0x0
9	XHCLKRST	XMT High Freq. clk Divider 0x0: RESET 0x1: ACTIVE	RW	0x0
8	XCLKRST	XMT clock divder reset 0x0: RESET 0x1: ACTIVE	RW	0x0
7:5	RESERVED		RW	0x0
4	RFRST	Frame sync generator reset 0x0: RESET 0x1: ACTIVE	R	0x0
3	RSMRST	RCV state machine reset 0x0: RESET 0x1: ACTIVE	R	0x0
2	RSRCLKR	RCV serializer clear 0x0: CLEAR 0x1: ACTIVE	R	0x0
1	RHCLKRST	RCV High Freq. clk Divider 0x0: RESET 0x1: ACTIVE	R	0x0
0	RCLKRST	RCV clock divder reset 0x0: RESET 0x1: ACTIVE	R	0x0

Table 18-418. Register Call Summary for Register MCASP_GBLCTLX

Multichannel Audio Serial Port

- MCASP_CFG Register Summary: [0]

Table 18-419. MCASP_TXMASK

Address Offset	0x0000 00A4		
Physical Address	0x4846 00A4 0x4846 C0A4 0x4847 00A4	**Instance**	MCASP1_CFG_PER2_L4 MCASP2_CFG_PER2_L4 MCASP3_CFG_PER2_L4
Description	Transmit format unit bit mask register - Determines which bits of the transmitted data are masked off before being shifted out the McASP		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
XMASK[31:0]																															

Bits	Field Name	Description	Type	Reset
31:0	XMASK[31:0]	Transmit data mask enable bit 0x0: The corresponding bit of transmit data is masked out and then transmitted out the McASP in place of the original bit. 0x1: The corresponding bit of transmit data is transmitted out the McASP.	RW	0

Table 18-420. Register Call Summary for Register MCASP_TXMASK

Multichannel Audio Serial Port

- Transmit Format Unit: [0][1][2]
- Burst Transfer Mode: [3]
- Time-Division Multiplexed (TDM) Transfer Mode: [4]
- DIT Transfer Mode: [5]
- MCASP Global Initialization: [6][7][8]
- MCASP_CFG Register Summary: [9]

Table 18-421. MCASP_TXFMT

Address Offset	0x0000 00A8		
Physical Address	0x4846 00A8 0x4846 C0A8 0x4847 00A8	**Instance**	MCASP1_CFG_PER2_L4 MCASP2_CFG_PER2_L4 MCASP3_CFG_PER2_L4
Description	Transmit bitstream format register - configures the transmit data format		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED														XDATDLY		XRVRS	XPAD			XPBIT				XSSZ				XBUSEL	XROT		

Bits	Field Name	Description	Type	Reset
31:18	RESERVED	Reserved	RW	0x0000
17:16	XDATDLY	Transmit sync bit delay 0x0: 0 bit delay - The first transmit data bit, on the AXR[n], occurs in the same ACLKX cycle as the transmit frame sync (AFSX). 0x1: 1-bit delay. The first transmit data bit, AXR[n], occurs one ACLKX cycle after the transmit frame sync (AFSX). 0x2: 2-bit delay. The first transmit data bit, AXR[n], occurs two ACLKX cycles after the transmit frame sync (AFSX). 0x3: Reserved	RW	0x0
15	XRVRS	Transmit serial bitstream order 0x0: Bitstream is LSB first. No bit reversal is performed in transmit format unit. 0x1: Bitstream is MSB first. Bit reversal is performed in transmit format bit reverse unit.	RW	0x0
14:13	XPAD	Pad value for extra bits in slot not belonging to word defined by XMASK. This field only applies to bits when XMASK[n] = 0. 0x0: Pad extra bits with 0. 0x1: Pad extra bits with 1. 0x2: Pad extra bits with one of the bits from the word as specified by XPBIT bits. 0x3: Reserved	RW	0x00
12:8	XPBIT	XPBIT value determines which bit (as written by the CPU or DMA to XBUF[n]) is used to pad the extra bits before shifting. This field only applies when XPAD = 0x2. 0x0: Pad with bit 0 value. 0x1 - 0x1F: Pad with bit 1 to bit 31 value.	RW	0x0
7:4	XSSZ	Transmit slot size 0x0 - 0x2: Reserved 0x3: Slot size is 8 bits 0x4: Reserved 0x5: Slot size is 12 bits 0x6: Reserved 0x7: Slot size is 16 bits 0x8: Reserved 0x9: Slot size is 20 bits 0xA: Reserved 0xB: Slot size is 24 bits 0xC: Reserved 0xD: Slot size is 28 bits 0xE: Reserved 0xF: Slot size is 32 bits.	RW	0x0
3	XBUSEL	Selects whether writes to the serializer buffer XBUF[n] originate from the peripheral configuration CFG port or the DATA port. 0x0: Writes to XBUF[n] originate from the DATA port. Writes to XBUF[n] from the peripheral configuration port are ignored with no effect on the McASP. 0x1: Writes to XBUF[n] originate from the peripheral configuration port - CFG port. Writes to XBUF[n] from the DATA port are ignored with no effect on the McASP.	RW	0

Bits	Field Name	Description	Type	Reset
2:0	XROT	Right-rotation value for transmit rotate right format unit 0x0: Rotate right by 0 (no rotation). 0x1: Rotate right by 4 bit positions. 0x2: Rotate right by 8 bit positions. 0x3: Rotate right by 12 bit positions. 0x4: Rotate right by 16 bit positions. 0x5: Rotate right by 20 bit positions. 0x6: Rotate right by 24 bit positions. 0x7: Rotate right by 28 bit positions.	RW	0x0

Table 18-422. Register Call Summary for Register MCASP_TXFMT

Multichannel Audio Serial Port

- Frame-Sync Generator: [0]
- Format Units: [1][2][3]
- Transmit Format Unit: [4][5][6][7][8][9][10][11][12][13][14]
- State-Machines: [15]
- Burst Transfer Mode: [16][17]
- Time-Division Multiplexed (TDM) Transfer Mode: [18]
- DIT Transfer Mode: [19][20]
- Data Ready Status and Event/Interrupt Generation: [21][22]
- MCASP Global Initialization: [23][24][25][26][27][28][29][30][31][32][33][34]
- MCASP Transmission Modes: [35][36]
- MCASP_CFG Register Summary: [37][41]
- MCASP_DAT Register Summary: [42]
- MCASP_DAT Register Description: [43]

Table 18-423. MCASP_TXFMCTL

Address Offset	0x0000 00AC		
Physical Address	0x4846 00AC 0x4846 C0AC 0x4847 00AC	**Instance**	MCASP1_CFG_PER2_L4 MCASP2_CFG_PER2_L4 MCASP3_CFG_PER2_L4
Description	Transmit frame-sync control register - configures the transmit frame sync (AFSX).		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																XMOD								RESERVED			FXWID	RESERVED		FSXM	FSXP

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Reserved	RW	0x0000
15:7	XMOD	Transmit frame-sync mode select bits 0x0: Burst mode 0x1: Reserved 0x2: 2-slot TDM mode (I2S transmit mode) 0x3 - 0x20: 3-slot TDM to 32-slot TDM mode 0x21 - 0x17F: Reserved 0x180: 384-slot DIT mode All other: Reserved	RW	0x000
6:5	RESERVED	Reserved	RW	0x0

Bits	Field Name	Description	Type	Reset
4	FXWID	The transmit frame-sync width select bit indicates the width of the transmit frame sync (AFSX) during its active period. 0x0: Single bit 0x1: Single word. Single word is not supported if XMOD is set to burst mode.	RW	0
3:2	RESERVED	Reserved	RW	0x0
1	FSXM	Transmit frame-sync generation select bit 0x0: Externally-generated transmit frame 0x1: Internally-generated transmit frame sync	RW	0
0	FSXP	Transmit frame-sync polarity select bit 0x0: Rising Edge - A rising edge on transmit frame sync (AFSX) indicates the beginning of a frame. 0x1: Falling Edge - A falling edge on transmit frame sync (AFSX) indicates the beginning of a frame.	RW	0

Table 18-424. Register Call Summary for Register MCASP_TXFMCTL

Multichannel Audio Serial Port

- Frame-Sync Generator: [0][1][2][3][4][5][6]
- Burst Transfer Mode: [7][8]
- Time-Division Multiplexed (TDM) Transfer Mode: [9]
- DIT Transfer Mode: [10][11]
- Loopback Mode Configurations: [12]
- MCASP Global Initialization: [13][14][15][16][17][18][19][20][21]
- MCASP_CFG Register Summary: [22]

Table 18-425. MCASP_ACLKXCTL

Address Offset	0x0000 00B0		
Physical Address	0x4846 00B0 0x4846 C0B0 0x4847 00B0	**Instance**	MCASP1_CFG_PER2_L4 MCASP2_CFG_PER2_L4 MCASP3_CFG_PER2_L4
Description	Transmit clock control register - Configures the transmit bit clock (ACLKX) and the transmit clock generator.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED											BUSY	DIVBUSY	ADJBUSY	CLKXADJ		RESERVED								CLKXP	ASYNC	CLKXM	CLKXDIV				

Bits	Field Name	Description	Type	Reset
31:21	RESERVED	Reserved	RW	0x00
20	BUSY	Status: logical OR of DIVBUSY, ADJBUSY. Not supported. 0x0: NOTBUSY 0x1: BUSY	RW	0x0
19	DIVBUSY	Status: divide ratio change in progress? Not supported. 0x0: NOTBUSY 0x1: BUSY	RW	0x0
18	ADJBUSY	Status: one-shot adjustment in progress? Not supported. 0x0: NOTBUSY 0x1: BUSY	RW	0x0

Bits	Field Name	Description	Type	Reset
17:16	CLKXADJ	CLKXDIV one-shot adjustment. Not supported. Bit field must always be written as 0x0. If CLKXDIV is set such that there are “m” input clocks per one output clock, then for one output cycle: 00 = (m+0) input clocks per output clock, i.e. no adjustment 01 = (m-1) input clocks per output clock 10 = (m+1) input clocks per output clock 11 = (m+0) input clocks per output clock, i.e. no adjustment NOTE: writes to these bits are ineffective if CLKADJEN:ENABLE bit is set to 0b. These bits are ALWAYS read back as zero.	W	0x0
15:8	RESERVED		RW	0x0
7	CLKXP	Transmit bitstream clock polarity select bit. 0x0: Rising edge. External receiver samples data on the falling edge of the serial clock, so the transmitter must shift data out on the rising edge of the serial clock. 0x1: Falling edge. External receiver samples data on the rising edge of the serial clock, so the transmitter must shift data out on the falling edge of the serial clock.	RW	0
6	ASYNC	Transmit operation asynchronous enable bit 0x0: Synchronous. Transmit clock and frame sync provides the source for both the transmit and receive sections. Note that in this mode, the receive bit clock is an inverted version of the transmit bit clock. 0x1: Asynchronous. Separate clock and frame sync used by transmit and receive sections.	RW	1
5	CLKXM	Transmit bit clock source bit 0x0: External transmit clock source from ACLKX pin. 0x1: Internal (output of divider)	RW	1
4:0	CLKXDIV	Transmit bit clock divide ratio bits, determine the divide-down ratio from AHCLKX to ACLKX. 0x0: Divide-by-1 0x1: Divide-by-2 0x2 to 0x1F: Divide-by-3 to divide-by-32	RW	0x00

Table 18-426. Register Call Summary for Register MCASP_ACLKXCTL

Multichannel Audio Serial Port

- MCASP Transmit Clock: [0][1][2][3][4][5]
- MCASP Receive Clock: [6]
- Frame-Sync Generator: [7][8]
- Synchronous and Asynchronous Transmit and Receive Operations: [9][10][11]
- Burst Transfer Mode: [12]
- Time-Division Multiplexed (TDM) Transfer Mode: [13]
- DIT Transfer Mode: [14][15]
- Loopback Modes: [16]
- Loopback Mode Configurations: [17]
- MCASP Global Initialization: [18][19][20][21][22][23][24][25][26][27][28][29][30]
- MCASP_CFG Register Summary: [31]
- MCASP_CFG Register Description: [35][36][37]

Table 18-427. MCASP_AHCLKXCTL

Address Offset	0x0000 00B4		
Physical Address	0x4846 00B4 0x4846 C0B4 0x4847 00B4	**Instance**	MCASP1_CFG_PER2_L4 MCASP2_CFG_PER2_L4 MCASP3_CFG_PER2_L4
Description	High-frequency transmit clock control register - Configures the transmit high-frequency master clock (AHCLKX) and the transmit clock generator.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED											BUSY	DIVBUSY	ADJBUSY	HCLKXADJ		HCLKXM	HCLKXP	RESERVED		HCLKXDIV											

Bits	Field Name	Description	Type	Reset
31:21	RESERVED	Reserved	RW	0x000
20	BUSY	Status: logical OR of DIVBUSY, ADJBUSY. Not supported. 0x0: NOTBUSY 0x1: BUSY	RW	0x0
19	DIVBUSY	Status: divide ratio change in progress? Not supported. 0x0: NOTBUSY 0x1: BUSY	RW	0x0
18	ADJBUSY	Status: one-shot adjustment in progress? Not supported. 0x0: NOTBUSY 0x1: BUSY	RW	0x0
17:16	HCLKXADJ	HCLKXDIV one-shot adjustment. Not supported. Bit field must always be written as 0x0. If HCLKXDIV is set such that there are “m” input clocks per one output clock, then for one output cycle: 00 = (m+0) input clocks per output clock, i.e. no adjustment 01 = (m-1) input clocks per output clock 10 = (m+1) input clocks per output clock 11 = (m+0) input clocks per output clock, i.e. no adjustment NOTE: writes to these bits are ineffective if CLKADJEN:ENABLE bit is set to 0b. These bits are ALWAYS read back as zero.	W	0x0
15	HCLKXM	Transmit high-frequency clock source bit 0x0: External transmit high-frequency clock source from AHCLKX pin. 0x1: Internal transmit high-frequency clock source from output of programmable high clock divider	RW	1
14	HCLKXP	Transmit bitstream high-frequency clock polarity select bit. 0x0: Not inverted. AHCLKX is not inverted before programmable bit clock divider. 0x1: Inverted. AHCLKX is inverted before programmable bit clock divider.	RW	0
13:12	RESERVED	Reserved	RW	0x0
11:0	HCLKXDIV	Transmit high-frequency clock divide ratio bits determine the divide-down ratio from AUXCLK to AHCLKX. 0x0: Divide-by-1 0x1: Divide-by-2 0x2 to 0xFFF: Divide-by-3 to divide-by-4096	RW	0x000

Table 18-428. Register Call Summary for Register MCASP_AHCLKXCTL

Multichannel Audio Serial Port

- MCASP Transmit Clock: [0][1][2][3][4]
- Burst Transfer Mode: [5]
- Time-Division Multiplexed (TDM) Transfer Mode: [6]
- DIT Transfer Mode: [7]
- MCASP Global Initialization: [8][9][10][11][12][13][14][15]
- MCASP_CFG Register Summary: [16]

Table 18-429. MCASP_TXTDM

Address Offset	0x0000 00B8		
Physical Address	0x4846 00B8 0x4846 C0B8 0x4847 00B8	**Instance**	MCASP1_CFG_PER2_L4 MCASP2_CFG_PER2_L4 MCASP3_CFG_PER2_L4
Description	Transmit TDM slot 0-31 register - TDM time slot counter range is to 384 slots (to support SPDIF blocks of 384 subframes).		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
XTDMS[31:0]																															

Bits	Field Name	Description	Type	Reset
31:0	XTDMS[31:0]	Transmitter mode during TDM time slot n (n = 0..31) 0x0: Transmit TDM time slot n is inactive. The transmit serializer does not shift out data during this slot. 0x1: The transmit TDM time slot n is active. The transmit serializer shifts out data during this slot according to the serializer control registers - MCASP_XRSRCTLn.	RW	0

Table 18-430. Register Call Summary for Register MCASP_TXTDM

Multichannel Audio Serial Port

- TDM Sequencers: [0][1]
- Burst Transfer Mode: [2]
- Time-Division Multiplexed (TDM) Transfer Mode: [3][4][5][6]
- DIT Transfer Mode: [7][8]
- Data Ready Status and Event/Interrupt Generation: [9]
- MCASP Global Initialization: [10][11]
- MCASP Transmission Modes: [12]
- MCASP_CFG Register Summary: [13]

Table 18-431. MCASP_EVTCTLX

Address Offset	0x0000 00BC		
Physical Address	0x4846 00BC 0x4846 C0BC 0x4847 00BC	**Instance**	MCASP1_CFG_PER2_L4 MCASP2_CFG_PER2_L4 MCASP3_CFG_PER2_L4
Description	Transmitter Interrupt control register - controls generation of the McASP transmit interrupt (XINT). When the register bit(s) is set to 1, the occurrence of the enabled McASP condition(s) generates XINT.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								XSTAFRM	RESERVED	XDATA	XLAST	XDMAERR	XCKFAIL	XSYNCERR	XUNDRN

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Reserved	RW	0x000000
7	XSTAFRM	Transmit start of frame interrupt enable bit 0x0: Interrupt is disabled. A transmit-start-of-frame interrupt does not generate a McASP transmit interrupt (XINT). 0x1: Interrupt is enabled. A transmit-start-of-frame interrupt generates a McASP transmit interrupt (XINT).	RW	0
6	RESERVED	Reserved	RW	0
5	XDATA	Transmit data-ready interrupt enable bit 0x0: Interrupt is disabled. A transmit data-ready interrupt does not generate a McASP transmit interrupt (XINT). 0x1: Interrupt is enabled. A transmit data-ready interrupt generates a McASP transmit interrupt (XINT).	RW	0
4	XLAST	Transmit last slot interrupt enable bit 0x0: Interrupt is disabled. A transmit-last-slot interrupt does not generate a McASP transmit interrupt (XINT). 0x1: Interrupt is enabled. A transmit-last-slot interrupt generates a McASP transmit interrupt (XINT).	RW	0
3	XDMAERR	Transmit DMA error interrupt enable bit 0x0: Interrupt is disabled. A transmit DMA error interrupt does not generate a McASP transmit interrupt (XINT). 0x1: Interrupt is enabled. A transmit DMA error interrupt generates a McASP transmit interrupt (XINT).	RW	0
2	XCKFAIL	Transmit clock failure interrupt enable bit 0x0: Interrupt is disabled. A transmit clock failure interrupt does not generate a McASP transmit interrupt (XINT). 0x1: Interrupt is enabled. A transmit clock failure interrupt generates a McASP transmit interrupt (XINT).	RW	0
1	XSYNCERR	Unexpected transmit frame-sync interrupt enable bit 0x0: Interrupt is disabled. An unexpected transmit frame-sync interrupt does not generate a McASP transmit interrupt (XINT). 0x1: Interrupt is enabled. An unexpected transmit frame-sync interrupt generates a McASP transmit interrupt (XINT).	RW	0
0	XUNDRN	Transmitter underrun interrupt enable bit 0x0: Interrupt is disabled. A transmitter underrun interrupt does not generate a McASP transmit interrupt (XINT). 0x1: Interrupt is enabled. A transmitter underrun interrupt generates a McASP transmit interrupt (XINT).	RW	0

Table 18-432. Register Call Summary for Register MCASP_EVTCTLX

Multichannel Audio Serial Port

- Burst Transfer Mode: [0]
- Time-Division Multiplexed (TDM) Transfer Mode: [1]
- DIT Transfer Mode: [2]
- Data Ready Status and Event/Interrupt Generation: [3][4]
- MCASP Events and Interrupt Requests: [5][6][7][8][9][10][11][12]
- Transmit Data Ready Event and Interrupt: [13]
- Error Interrupt: [14]
- Multiple Interrupts: [15][16]
- Clock Failure Detection: [17][18]
- MCASP Transmission Modes: [19][20][21][22][23][24][25][26][27][28][29][30][31][32][33]
- MCASP_CFG Register Summary: [34]
- MCASP_CFG Register Description: [38][39][40][41][42][43][44]

Table 18-433. MCASP_TXSTAT

Address Offset	0x0000 00C0		
Physical Address	0x4846 00C0 0x4846 C0C0 0x4847 00C0	**Instance**	MCASP1_CFG_PER2_L4 MCASP2_CFG_PER2_L4 MCASP3_CFG_PER2_L4
Description	Transmitter status register - If the McASP logic attempts to set an interrupt flag in the same cycle that the CPU writes to the flag to clear it, the McASP logic has priority and the flag remains set. This also causes the generation of a new interrupt request.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							XERR	XDMAERR	XSTAFRM	XDATA	XLAST	XTDMSLOT	XCKFAIL	XSYNCERR	XUNDRN

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reserved	RW	0x000000
8	XERR	XERR bit always returns a logic-OR of: XUNDRN | XSYNCERR | XCKFAIL | XDMAERR. Allows a single bit to be checked to determine if a transmitter error interrupt has occurred. 0x0: No errors have occurred. 0x1: An error has occurred.	RW	0
7	XDMAERR	Transmit DMA error flag. XDMAERR is set when the CPU or DMA writes more words to the DATA port of the McASP in a given time slot than it should. Causes a transmit interrupt (XINT) if this bit and XDMAERR in MCASP_EVTCTLX are set. This bit is cleared by writing a 1 to it. Writing a 0 has no effect. 0x0: Transmit DMA error did not occur. 0x1: Transmit DMA error occurred.	RW	0
6	XSTAFRM	Transmit start of frame flag. Causes a transmit interrupt (XINT) if this bit and XSTAFRM in MCASP_EVTCTLX are set. This bit is cleared by writing a 1 to it. Writing a 0 has no effect. 0x0: No new transmit frame sync (AFSX) is detected. 0x1: A new transmit frame sync (AFSX) is detected.	RW	0

Bits	Field Name	Description	Type	Reset
5	XDATA	Transmit data ready flag. Causes a transmit interrupt (XINT) if this bit and XDATA in MCASP_EVTCTLX are set. This bit is cleared by writing a 1 to iit. Writing a 0 has no effect 0x0: XBUF[n] is written and is full 0x1: Data is copied from XBUF[n] to XRSR[n]. XBUF[n] is empty and ready to be written. XDATA is also set when the transmit serializers are taken out of reset. When XDATA is set, it always causes a DMA event (AXEVT).	RW	0
4	XLAST	Transmit last slot flag. XLAST, along with XDATA, are set if the current slot is the last slot in a frame. Causes a transmit interrupt (XINT) if this bit and XLAST in MCASP_EVTCTLX are set. This bit is cleared by writing a 1 to iit. Writing a 0 has no effect. 0x0: Current slot is not the last slot in a frame. 0x1: Current slot is the last slot in a frame. XDATA is also set.	RW	0
3	XTDMSLOT	Returns the LSB of XSLOT. Allows a single read of XSTAT to determine whether the current TDM time slot is even or odd. read 0x0: Current TDM time slot is odd. read 0x1: Current TDM time slot is even.	R	0
2	XCKFAIL	Transmit clock failure flag. XCKFAIL is set when the transmit clock failure detection circuit reports an error. Causes a transmit interrupt (XINT) if this bit and XCKFAIL in MCASP_EVTCTLX are set. This bit is cleared by writing a 1 to iit. Writing a 0 has no effect. 0x0: Transmit clock failure did not occur. 0x1: Transmit clock failure occurred	RW	0
1	XSYNCERR	Unexpected transmit frame-sync flag. XSYNCERR is set when a new transmit frame sync (AFSX) occurs before it is expected. Causes a transmit interrupt (XINT) if this bit and XSYNCERR in MCASP_EVTCTLX are set. This bit is cleared by writing a 1 to iit. Writing a 0 has no effect. 0x0: Unexpected transmit frame sync did not occur 0x1: Unexpected transmit frame sync occurred.	RW	0
0	XUNDRN	Transmitter underrun flag. XUNDRN is set when the transmit serializer is instructed to transfer data from XBUF[n] to XRSR[n], but XBUF[n] has not yet been serviced with new data since the last transfer. Causes a transmit interrupt (XINT) if this bit and XUNDRN in MCASP_EVTCTLX are set. This bit is cleared by writing a 1 to iit. Writing a 0 has no effect. 0x0: Transmitter underrun did not occur 0x1: Transmitter underrun occurred.	RW	0

Table 18-434. Register Call Summary for Register MCASP_TXSTAT

Multichannel Audio Serial Port

- State-Machines: [0]
- Data Ready Status and Event/Interrupt Generation: [1][2][3][4][5]
- MCASP Events and Interrupt Requests: [6][7][8][9][10][11][12][13][14][15][16]
- Transmit Data Ready Event and Interrupt: [17][18]
- Error Interrupt: [19][20]
- Multiple Interrupts: [21][22][23][24]
- Buffer Underrun Error -Transmitter: [25]
- DATA Port Error - Transmitter: [26][27][28]
- Clock Failure Detection: [29][30][31]
- MCASP Transmission Modes: [32][33]
- MCASP Event Servicing: [34]
- MCASP_CFG Register Summary: [35]

Table 18-435. MCASP_TXTDMSLOT

Address Offset	0x0000 00C4		
Physical Address	0x4846 00C4 0x4846 C0C4 0x4847 00C4	**Instance**	MCASP1_CFG_PER2_L4 MCASP2_CFG_PER2_L4 MCASP3_CFG_PER2_L4
Description	Current transmit TDM time slot register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								XSLOTCNT							

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reserved	R	0x000000
8:0	XSLOTCNT	Current transmit time slot count. the value of this register is 0b0101111111 (0x17f) during reset and 0 after reset.	R	0x000

Table 18-436. Register Call Summary for Register MCASP_TXTDMSLOT

Multichannel Audio Serial Port

- TDM Sequencers: [0]
- MCASP_CFG Register Summary: [1]

Table 18-437. MCASP_TXCLKCHK

Address Offset	0x0000 00C8		
Physical Address	0x4846 00C8 0x4846 C0C8 0x4847 00C8	**Instance**	MCASP1_CFG_PER2_L4 MCASP2_CFG_PER2_L4 MCASP3_CFG_PER2_L4
Description	Transmit clock check control register - configures the transmit clock failure detection circuit.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
XCNT								XMAX								XMIN								RESERVED				XPS			

Bits	Field Name	Description	Type	Reset
31:24	XCNT	Transmit clock count value (from previous measurement). The clock circuit continually counts the number of interface clocks for every 32 transmit high-frequency master clock (AHCLKX) signals, and stores the count in XCNT until the next measurement is taken	R	0x00
23:16	XMAX	0x0 to 0xFF: Transmit clock maximum boundary. This 8-bit unsigned value sets the maximum allowed boundary for the clock check counter after 32 transmit high-frequency master clock (AHCLKX) signals have been received. If the current counter value is greater than XMAX after counting 32 AHCLKX signals, XCKFAIL in XSTAT is set. The comparison is performed using unsigned arithmetic.	RW	0x00
15:8	XMIN	0x0 to 0xFF: Transmit clock minimum boundary. This 8-bit unsigned value sets the minimum allowed boundary for the clock check counter after 32 transmit high-frequency master clock (AHCLKX) signals have been received. If XCNT is less than XMIN after counting 32 AHCLKX signals, XCKFAIL in XSTAT is set. The comparison is performed using unsigned arithmetic.	RW	0x00
7:4	RESERVED	Reserved	RW	0x0

Bits	Field Name	Description	Type	Reset
3:0	XPS	Transmit clock check prescaler value 0x0: McASP interface clock divided by 1 0x1: McASP interface clock divided by 2 0x2: McASP interface clock divided by 4 0x3: McASP interface clock divided by 8 0x4: McASP interface clock divided by 16 0x5: McASP interface clock divided by 32 0x6: McASP interface clock divided by 64 0x7: McASP interface clock divided by 128 0x8: McASP interface clock divided by 256 0x9 to 0xF: Reserved	RW	0x0

Table 18-438. Register Call Summary for Register MCASP_TXCLKCHK

Multichannel Audio Serial Port
- Burst Transfer Mode: [0]
- Time-Division Multiplexed (TDM) Transfer Mode: [1]
- DIT Transfer Mode: [2]
- Clock Failure Detection: [3][4]
- MCASP_CFG Register Summary: [5]

Table 18-439. MCASP_XEVTCTL

Address Offset	0x0000 00CC		
Physical Address	0x4846 00CC 0x4846 C0CC 0x4847 00CC	**Instance**	MCASP1_CFG_PER2_L4 MCASP2_CFG_PER2_L4 MCASP3_CFG_PER2_L4
Description	Transmitter DMA event control register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															XDATDMA

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved	RW	0x0000 0000
0	XDATDMA	Transmit data DMA request enable bit. 0x0: The transmit data DMA request is enabled. 0x1: The transmit data DMA request is disabled.	RW	0

Table 18-440. Register Call Summary for Register MCASP_XEVTCTL

Multichannel Audio Serial Port
- Data Ready Status and Event/Interrupt Generation: [0][1]
- DMA Requests: [2]
- MCASP Transmission Modes: [3][4][5]
- MCASP_CFG Register Summary: [6]

Table 18-441. MCASP_CLKADJEN

Address Offset	0x0000 00D0		
Physical Address	0x4846 00D0 0x4846 C0D0 0x4847 00D0	**Instance**	MCASP1_CFG_PER2_L4 MCASP2_CFG_PER2_L4 MCASP3_CFG_PER2_L4
Description	One-Shot Clock Adjustment Enable		

Table 18-441. MCASP_CLKADJEN (continued)

Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															ENABLE

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		RW	0x0
0	ENABLE	One-shot clock adjust enable. Not supported. Bit field must always be written as 0x0. 0x0: DISABLE 0x1: ENABLE	RW	0x0

Table 18-442. Register Call Summary for Register MCASP_CLKADJEN

Multichannel Audio Serial Port

- MCASP_CFG Register Summary: [0]

Table 18-443. MCASP_DITCSRAi

Address Offset	0x0000 0100 + (0x4*i)	**Index**	i = 0 to 5
Physical Address	0x4846 0100 + (0x04*i) 0x4846 C100 + (0x04*i) 0x4847 0100 + (0x04*i)	**Instance**	MCASP1_CFG_PER2_L4 MCASP2_CFG_PER2_L4 MCASP3_CFG_PER2_L4
Description	DIT left channel status register - All six 32-bit registers (i = 0 to 5) can store 192 bits of channel status data for a complete block of transmission. The DIT reuses the same data for the next block. Make sure to update the register file before a different set of data needs to be sent.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DITCSRAi																															

Bits	Field Name	Description	Type	Reset
31:0	DITCSRAi	Left (even TDM slot) channel status	RW	0x0000 0000

Table 18-444. Register Call Summary for Register MCASP_DITCSRAi

Multichannel Audio Serial Port

- Burst Transfer Mode: [0]
- Time-Division Multiplexed (TDM) Transfer Mode: [1]
- DIT Transfer Mode: [2][3]
- MCASP Global Initialization: [4]
- MCASP Transmission Modes: [5]
- MCASP_CFG Register Summary: [6]

Table 18-445. MCASP_DITCSRBi

Address Offset	0x0000 0118+ (0x4*i)	**Index**	i = 0 to 5
Physical Address	0x4846 0118 + (0x04*i) 0x4846 C118 + (0x04*i) 0x4847 0118 + (0x04*i)	**Instance**	MCASP1_CFG_PER2_L4 MCASP2_CFG_PER2_L4 MCASP3_CFG_PER2_L4
Description	DIT right channel status register - All six 32-bit registers (i = 0 to 5) can store 192 bits of channel status data for a complete block of transmission. The DIT reuses the same data for the next block. Make sure to update the register file before a different set of data needs to be sent.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DITCSRBi																															

Bits	Field Name	Description	Type	Reset
31:0	DITCSRBi	Right (odd TDM slot) channel status	RW	0x0000 0000

Table 18-446. Register Call Summary for Register MCASP_DITCSRBi

Multichannel Audio Serial Port

- Burst Transfer Mode: [0]
- Time-Division Multiplexed (TDM) Transfer Mode: [1]
- DIT Transfer Mode: [2][3]
- MCASP Global Initialization: [4]
- MCASP Transmission Modes: [5]
- MCASP_CFG Register Summary: [6]

Table 18-447. MCASP_DITUDRAi

Address Offset	0x0000 0130 + (0x4*i)	**Index**	i = 0 to 5
Physical Address	0x4846 0130 + (0x04*i) 0x4846 C130 + (0x04*i) 0x4847 0130 + (0x04*i)	**Instance**	MCASP1_CFG_PER2_L4 MCASP2_CFG_PER2_L4 MCASP3_CFG_PER2_L4
Description	DIT left channel user data register - provides the user data of each left channel (even TDM time slot). All six 32-bit registers (i = 0 to 5) can store 192 bits of user data for a complete block of transmission. The DIT reuses the same data for the next block. Make sure to update the register before a different set of data needs to be sent.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DITUDRAi																															

Bits	Field Name	Description	Type	Reset
31:0	DITUDRAi	Left (even TDM slot) user data	RW	0x0000 0000

Table 18-448. Register Call Summary for Register MCASP_DITUDRAi

Multichannel Audio Serial Port

- Burst Transfer Mode: [0]
- Time-Division Multiplexed (TDM) Transfer Mode: [1]
- DIT Transfer Mode: [2][3]
- MCASP Global Initialization: [4]
- MCASP Transmission Modes: [5]
- MCASP_CFG Register Summary: [6]

Table 18-449. MCASP_DITUDRBi

Address Offset	0x0000 0148+ (0x4*i)	**Index**	i = 0 to 5
Physical Address	0x4846 0148 + (0x04*i) 0x4846 C148 + (0x04*i) 0x4847 0148 + (0x04*i)	**Instance**	MCASP1_CFG_PER2_L4 MCASP2_CFG_PER2_L4 MCASP3_CFG_PER2_L4
Description	DIT right user data register - provides the user data of each right channel (odd TDM time slot). All six 32-bit registers (i = 0 to 5) can store 192 bits of user data for a complete block of transmission. The DIT reuses the same data for the next block. Make sure to update the register before a different set of data needs to be sent.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DITUDRBi																															

Bits	Field Name	Description	Type	Reset
31:0	DITUDRBi	Right (odd TDM slot) user data	RW	0x0000 0000

Table 18-450. Register Call Summary for Register MCASP_DITUDRBi

Multichannel Audio Serial Port

- Burst Transfer Mode: [0]
- Time-Division Multiplexed (TDM) Transfer Mode: [1]
- DIT Transfer Mode: [2][3]
- MCASP Global Initialization: [4]
- MCASP Transmission Modes: [5]
- MCASP_CFG Register Summary: [6]

Table 18-451. MCASP_XRSRCTLn

Address Offset	0x0000 0180 + (0x4*n)	**Index**	n = 0 to 15
Physical Address	0x4846 0180 + (0x04*n) 0x4846 C180 + (0x04*n) 0x4847 0180 + (0x04*n)	**Instance**	MCASP1_CFG_PER2_L4 MCASP2_CFG_PER2_L4 MCASP3_CFG_PER2_L4
Description	Serializer n control register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																										RRDY	XRDY	DISMOD		SRMOD	

Bits	Field Name	Description	Type	Reset
31:6	RESERVED	Reserved	RW	0x0000000
5	RRDY	Receive buffer ready bit. RRDY indicates the current receive buffer state. Always reads 0 when programmed as a transmitter or as inactive. If SRMOD bit is set to receive (2h), RRDY switches from 0 to 1 whenever data is transferred from XRSRn to RBUFn. Read 0x0: Receive buffer (MCASP_RXBUFn) is empty. Read 0x1: Receive buffer (MCASP_RXBUFn) contains data and needs to be read before the start of the next time slot or a receiver overrun occurs.	R	0

Bits	Field Name	Description	Type	Reset
4	XRDY	Transmit buffer ready bit. XRDY indicates the current transmit buffer state. Always reads 0 when programmed as a receiver or as inactive. If SRMOD bit is set to transmit (1h), XRDY switches from 0 to 1 when XSRCLR in GBLCTL is switched from 0 to 1 to indicate an empty transmitter. XRDY remains set until XSRCLR is forced to 0, data is written to the corresponding transmit buffer, or SRMOD bit is changed to receive (2h) or inactive (0). Read 0x0: The transmit buffer (MCASP_TXBUFn) contains data. Read 0x1: The transmit buffer (MCASP_TXBUFn) is empty and needs to be written before the start of the next time slot or a transmit underrun occurs.	R	0
3:2	DISMOD	Serializer pin drive mode bit. Drive on pin when in inactive TDM slot of transmit mode or when serializer is inactive. This field only applies if the pin is configured as a McASP pin (PFUNC = 0). 0x0: Drive on pin is 3-state. 0x1: Reserved 0x2: Drive on pin is logic low. 0x3: Drive on pin is logic high.	RW	0x0
1:0	SRMOD	Serializer mode bit 0x0:The serializer is inactive 0x1:The serializer is operating in transmit mode. 0x2: The serializer is operating in receive mode. 0x3: Reserved	RW	0x0

Table 18-452. Register Call Summary for Register MCASP_XRSRCTLn

Multichannel Audio Serial Port

- MCASP Signals: [0]
- Serializers: [1][2][3][4][5][6][7][8]
- TDM Sequencers: [9]
- Burst Transfer Mode: [10]
- Time-Division Multiplexed (TDM) Transfer Mode: [11][12]
- DIT Transfer Mode: [13]
- Data Ready Status and Event/Interrupt Generation: [14][15][16][17][18][19]
- MCASP Events and Interrupt Requests: [20]
- MCASP Global Initialization: [21][22][23][24][25][26][27]
- MCASP Transmission Modes: [28]
- MCASP Reception Modes: [29]
- MCASP_CFG Register Summary: [30]
- MCASP_CFG Register Description: [34]

Table 18-453. MCASP_TXBUFn

Address Offset	0x0000 0200 + (0x4*n)	**Index**	n = 0 to 15
Physical Address	0x4846 0200 + (0x04*n) 0x4846 C200 + (0x04*n) 0x4847 0200 + (0x04*n)	**Instance**	MCASP1_CFG_PER2_L4 MCASP2_CFG_PER2_L4 MCASP3_CFG_PER2_L4
Description	Transmit buffer n - The transmit buffer for the serializer n holds data from the transmit format unit.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
XBUFn																															

Bits	Field Name	Description	Type	Reset
31:0	XBUFn	Transmit buffer n	RW	0x0000 0000

Table 18-454. Register Call Summary for Register MCASP_TXBUFn

Multichannel Audio Serial Port

- Serializers: [0][1]
- TDM Sequencers: [2]
- Data Transmission and Reception: [3]
- Data Ready Status and Event/Interrupt Generation: [4][5][6][7][8][9][10][11][12][13]
- Buffer Underrun Error -Transmitter: [14]
- MCASP Global Initialization: [15]
- MCASP Transmission Modes: [16]
- MCASP_CFG Register Summary: [17][21]
- MCASP_CFG Register Description: [22][23]

Table 18-455. MCASP_RXBUFn

Address Offset	0x0000 0280 + (0x4*n)	**Index**	n = 0 to 15
Physical Address	0x4846 0280 + (0x04*n) 0x4846 C280 + (0x04*n) 0x4847 0280 + (0x04*n)	**Instance**	MCASP1_CFG_PER2_L4 MCASP2_CFG_PER2_L4 MCASP3_CFG_PER2_L4
Description	Receive buffer n - The receive buffer for the serializer n holds data before the data goes to the receive format unit.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RBUFn																															

Bits	Field Name	Description	Type	Reset
31:0	RBUFn	Receive Buffer n	RW	0x0000 0000

Table 18-456. Register Call Summary for Register MCASP_RXBUFn

Multichannel Audio Serial Port

- Serializers: [0][1]
- Data Transmission and Reception: [2]
- Data Ready Status and Event/Interrupt Generation: [3][4][5][6][7][8][9][10][11][12]
- Buffer Overrun Error-Receiver: [13]
- MCASP Global Initialization: [14]
- MCASP Reception Modes: [15]
- MCASP Event Servicing: [16]
- MCASP_CFG Register Summary: [17][21]
- MCASP_CFG Register Description: [22][23]

18.5.6.2.3 MCASP_AFIFO Register Summary

Table 18-457. MCASP_AFIFO Register Summary 1

Register Name	Type	Register Width (Bits)	Address Offset	MCASP1_AFIFO L4_PER2 Physical Address
WFIFOCTL	RW	32	0x0000 0000	0x4846 1000
WFIFOSTS	R	32	0x0000 0004	0x4846 1004
RFIFOCTL	RW	32	0x0000 0008	0x4846 1008
RFIFOSTS	R	32	0x0000 000C	0x4846 100C

Table 18-458. MCASP_AFIFO Register Summary 2

Register Name	Type	Register Width (Bits)	Address Offset	MCASP2_AFIFO L4_PER2 Physical Address	MCASP3_AFIFO L4_PER2 Physical Address
WFIFOCTL	RW	32	0x0000 0000	0x4846 D000	0x4847 1000
WFIFOSTS	R	32	0x0000 0004	0x4846 D004	0x4847 1004
RFIFOCTL	RW	32	0x0000 0008	0x4846 D008	0x4847 1008
RFIFOSTS	R	32	0x0000 000C	0x4846 D00C	0x4847 100C

18.5.6.2.4 MCASP_AFIFO Register Description

Table 18-459. WFIFOCTL

Address Offset	0x0000 0000		
Physical Address	0x4846 1000 0x4846 D000 0x4847 1000	**Instance**	MCASP1_AFIFO_PER2_L4 MCASP2_AFIFO_PER2_L4 MCASP3_AFIFO_PER2_L4
Description	The Write FIFO control register. The WNUMEVT and WNUMDMA values must be set prior to enabling the Write FIFO. If the Write FIFO is to be enabled, it must be enabled prior to taking the McASP out of reset.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED															WENA	WNUMEVT								WNUMDMA							

Bits	Field Name	Description	Type	Reset
31:17	RESERVED	Reserved	R	0x0000 0000
16	WENA	Write FIFO enable bit. 0x0: Write FIFO is disabled (default). Data access by the host must pass through the FIFO block to the McASP transparently. DMA requests must also pass through the FIFO block transparently. WLVL is reset to 0 and pointers are initialized, i.e., the write FIFO is “flushed.” 0x1: Write FIFO is enabled. If write FIFO is to be enabled, it must be enabled prior to enabling McASP.	RW	0
15:8	WNUMEVT	Write word count (32-bit) to generate TX event to host. When Write FIFO has word space for more or equal to this value then transmit event will be generated to host/DMA. This value must be set prior to enabling the write FIFO. 0x0: 0 words. 0x1: 1 word. 0x2: 2 words. 0x3 - 0x40: 3 to 64 words currently in write FIFO. 0x41 - 0xFF: Reserved.	RW	0x10

Bits	Field Name	Description	Type	Reset
7:0	WNUMDMA	Write word count (32-bit words). On the transmit DMA event from McASP the WNUMDMA word will be transferred from DMA engine to McASP. This value must equal the number of McASP serializers used as transmitters. This value must be set prior to enabling the write FIFO. 0x0: 0 words. 0x1: 1 word. 0x2: 2 words. 0x3 - 0x10: 3 to 16 words. 0x11 - 0xFF: Reserved.	RW	0x04

Table 18-460. Register Call Summary for Register WFIFOCTL

Multichannel Audio Serial Port
- MCASP Audio FIFO (AFIFO): [0]
- AFIFO Data Transmission: [1][2][3][4][5]
- MCASP_AFIFO Register Summary: [6]

Table 18-461. WFIFOSTS

Address Offset	0x0000 0004		
Physical Address	0x4846 1004 0x4846 D004 0x4847 1004	**Instance**	MCASP1_AFIFO_PER2_L4 MCASP2_AFIFO_PER2_L4 MCASP3_AFIFO_PER2_L4
Description	The Write FIFO status register.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								WLVL							

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Reserved	R	0x0000 0000
7:0	WLVL	Write level (read-only). Number of 32-bit words currently in write FIFO. 0x0: 0 words currently in write FIFO. 0x1: 1 word currently in write FIFO. 0x2: 2 words currently in write FIFO. 0x3 - 0x40: 3 to 64 words currently in write FIFO. 0x41 - 0xFF: Reserved.	R	0

Table 18-462. Register Call Summary for Register WFIFOSTS

Multichannel Audio Serial Port
- MCASP_AFIFO Register Summary: [0]

Table 18-463. RFIFOCTL

Address Offset	0x0000 0008		
Physical Address	0x4846 1008 0x4846 D008 0x4847 1008	**Instance**	MCASP1_AFIFO_PER2_L4 MCASP2_AFIFO_PER2_L4 MCASP3_AFIFO_PER2_L4
Description	The Read FIFO control register. The RNUMEVT and RNUMDMA values must be set prior to enabling the Read FIFO. If the Read FIFO is to be enabled, it must be enabled prior to taking the McASP out of reset.		

Table 18-463. RFIFOCTL (continued)

Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED															RENA	RNUMEVT								RNUMDMA							

Bits	Field Name	Description	Type	Reset
31:17	RESERVED	Reserved	R	0x0000 0000
16	RENA	Read FIFO enable bit. 0x0: Read FIFO is disabled (default). Data access by the host must pass through the FIFO block to the McASP transparently. DMA requests must also pass through the FIFO block transparently. RLVL is reset to 0 and pointers are initialized, i.e., the read FIFO is "flushed." 0x1: Read FIFO is enabled. If read FIFO is to be enabled, it must be enabled prior to enabling McASP.	RW	0
15:8	RNUMEVT	Read word count (32-bit) to generate RX event to host. When Read FIFO has number of word available which is more or equal to this value then receive event will be generated to host/DMA. This value must be set prior to enabling the write FIFO. 0x0: 0 words currently in read FIFO. 0x1: 1 word currently in read FIFO. 0x2: 2 words currently in read FIFO. 0x3 - 0x40: 3 to 64 words currently in read FIFO. 0x41 - 0xFF: Reserved	RW	0x10
7:0	RNUMDMA	Read word count (32-bit words). On receive DMA event from McASP, the DMA engine will read specified number of words from McASP. This value must equal the number of McASP serializers used as transmitters. This value must be set prior to enabling the read FIFO. 0x0: 0 words 0x1: 1 word 0x2: 2 words 0x3 - 0x10: 3-16 words 0x11 - 0xFF: Reserved		

Table 18-464. Register Call Summary for Register RFIFOCTL

Multichannel Audio Serial Port

- MCASP Audio FIFO (AFIFO): [0]
- AFIFO Data Reception: [1][2][3][4][5]
- MCASP_AFIFO Register Summary: [6]

Table 18-465. RFIFOSTS

Address Offset	0x0000 000C		
Physical Address	0x4846 100C 0x4846 D00C 0x4847 100C	**Instance**	MCASP1_AFIFO_PER2_L4 MCASP2_AFIFO_PER2_L4 MCASP3_AFIFO_PER2_L4
Description	The Read FIFO status register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								RLVL							

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Reserved	R	0x0000 0000
7:0	RLVL	Read level (read-only). Number of 32-bit words currently in read FIFO. 0x0: 0 words currently in read FIFO. 0x1: 1 word currently in read FIFO. 0x2: 2 words currently in read FIFO. 0x3 - 0x40: 3 to 64 words currently in read FIFO. 0x41 - 0xFF: Reserved.	R	0

Table 18-466. Register Call Summary for Register RFIFOSTS

Multichannel Audio Serial Port
- MCASP_AFIFO Register Summary: [0]

18.5.6.2.5 MCASP_DAT Register Summary

Table 18-467 and Table 18-468 summarize the MCASP_DAT register mapping.

Table 18-467. MCASP_DAT Register Summary 1

Register Name	Type	Register Width (Bits)	Address Offset	MCASP1_DAT L3_MAIN Physical Address
MCASP_RXBUF	R	32	0x0000 0000[1]	0x4580 0000
MCASP_TXBUF	W	32	0x0000 0000[1]	0x4580 0000

[1] 0x000 is just an example DATA port offset value. Actually, whatever the offset value is added to base address, it is ignored (don't care) when an access is performed to XRBUFn RX/TX buffers through the McASP DATA port.

Table 18-468. MCASP_DAT Register Summary 2

Register Name	Type	Register Width (Bits)	Address Offset	MCASP2_DAT L4_PER2 Physical Address	MCASP3_DAT L4_PER2 Physical Address
MCASP_RXBUF	R	32	0x0000 0000[1]	0x4843 6000	0x4843 A000
MCASP_TXBUF	W	32	0x0000 0000[1]	0x4843 6000	0x4843 A000

[1] 0x000 is just an example DATA port offset value. Actually, whatever the offset value is added to base address, it is ignored (don't care) when an access is performed to XRBUFn RX/TX buffers through the McASP DATA port.

NOTE: For MCASP_RXBUF and MCASP_TXBUF buffer accesses through the McASP DATA port, the destination physical address is always the same regardless of current channel index or transfer direction. The MCASP_TXFMT[3] XBUSEL bit must be set to 0b0, to allow write transfers through the DATA port. The MCASP_RXFMT[3] RBUSEL bit must be set to 0b0, to allow read transfers through the DATA port.

NOTE: The McASP DATA port is exclusively assigned for DMAs/device CPUs accesses to the McASP channels transmit and receive buffer registers. All other McASP module registers must be accessed through the McASP CFG (peripheral) port.

NOTE: The McASP1, whose data port is accessible directly via L3_MAIN, do not support FIFO/constant addressing modes. Incrementing transfers must be used instead.

18.5.6.2.6 MCASP_DAT Register Description

Table 18-469. MCASP_RXBUF

Address Offset	0x0000 0000		
Physical Address	0x4580 0000 0x4843 6000 0x4843 A000	**Instance**	MCASP1_DAT_MAIN_L3 MCASP2_DAT_PER2_L4 MCASP3_DAT_PER2_L4
Description	Through the DATA port, the Host can service all serializers through a single address and the McASP automatically cycles through the appropriate serializers. For receive operations through the DATA port, the Host should read from the same RBUF DATA port address to service all of the active receive serializers upon each receive data ready event. To enable accesses from the Host to the McASP XRBUF registers through the DATA port, one must clear the RBUSEL bits to 0 in the respective MCASP_RXFMT registers in the MCASP_CFG Memory Map.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RXBUF																															

Bits	Field Name	Description	Type	Reset
31:0	RXBUF	Rx buffer data.	R	0x0000 0000

Table 18-470. Register Call Summary for Register MCASP_RXBUF

Multichannel Audio Serial Port
- Data Ready Status and Event/Interrupt Generation: [0]
- MCASP_DAT Register Summary: [1][5]

Table 18-471. MCASP_TXBUF

Address Offset	0x0000 0000		
Physical Address	0x4580 0000 0x4843 6000 0x4843 A000	**Instance**	MCASP1_DAT_MAIN_L3 MCASP2_DAT_PER2_L4 MCASP3_DAT_PER2_L4
Description	Through the DATA port, the Host can service all serializers through a single address and the McASP automatically cycles through the appropriate serializers. For transmit operations through the DATA port, the Host should write to the same DATA port address to service all of the active transmit serializers upon each transmit data ready event. To enable accesses from the Host to the McASP XRBUF registers through the DATA port, one must clear the XBUSEL bits to 0 in the respective MCASP_TXFMT registers in the MCASP_CFG Memory Map.		
Type	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TXBUF																															

Bits	Field Name	Description	Type	Reset
31:0	TXBUF	Tx buffer data.	W	0x0000 0000

Table 18-472. Register Call Summary for Register MCASP_TXBUF

Multichannel Audio Serial Port • Data Ready Status and Event/Interrupt Generation: [0] • MCASP_DAT Register Summary: [1][5]

18.6 DCAN

This chapter describes the controller area network module (DCAN) in the device.

18.6.1 DCAN Overview

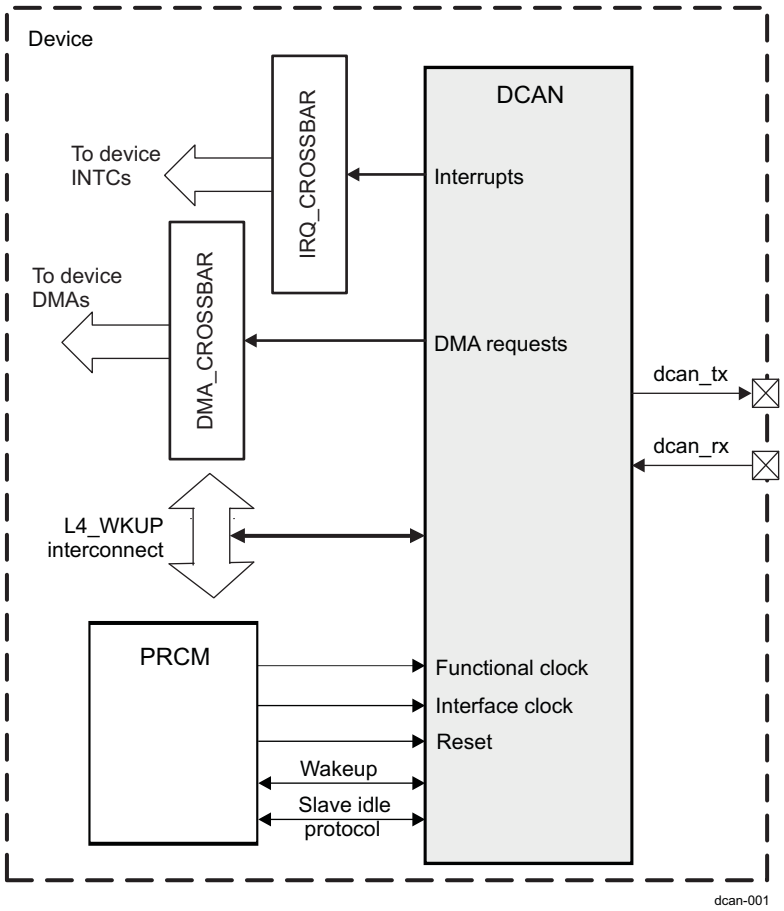
The Controller Area Network (CAN) is a serial communications protocol which efficiently supports distributed real-time applications. CAN has high immunity to electrical interference and the ability to self-diagnose and repair data errors. In a CAN network, many short messages are broadcast to the entire network, which provides for data consistency in every node of the system.

The device supports a single DCAN module.

Connection to the CAN network is performed through external (for the device) transceivers. The DCAN module supports bit rates up to 1 Mbit/s and is compliant to the CAN 2.0B protocol specification.

Figure 18-113 shows the DCAN module highlights.

Figure 18-113. DCAN Overview



18.6.1.1 Features

The DCAN module implements the following features:

- Support for CAN protocol version 2.0 part A, B
- Bit rates up to 1 Mbit/s
- 64 message objects in a dedicated message RAM
- Individual identifier mask for each message object
- Programmable FIFO mode for message objects
- Programmable loop-back modes for self-test operation

- Suspend mode for debug support
- Automatic bus on after Bus-Off state by a programmable 32-bit timer
- Message RAM single error correction and double error detection (SECDED) mechanism
- Direct access to message RAM during test mode
- Support for two interrupt lines: Level 0 and Level 1, plus separate ECC interrupt line
- Local power down and wakeup support
- Automatic message RAM initialization
- Support for DMA access

18.6.2 DCAN Environment

CAN network physical layer consists of two-wire differential bus, usually twisted pair, and provides high level of interference immunity. External CAN transceiver IC is needed to access a CAN bus by the DCAN.

Figure 18-114 shows an overview of a typical DCAN application.

Figure 18-114. DCAN Typical Application

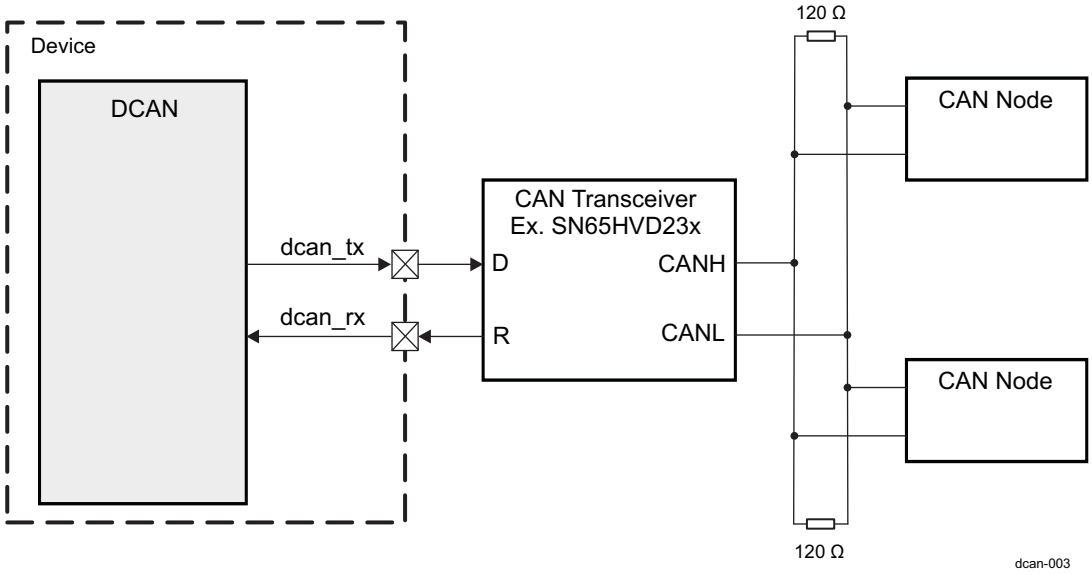


Table 18-473 describes the external signals of the DCAN module.

Table 18-473. DCAN I/O Description

Module Signal	Device Signal	I/O	Description	Value at Reset
CAN_RX	dcan_rx	I	Serial data input from external CAN tranceiver	HiZ
CAN_TX	dcan_tx	O	Serial data output to external CAN tranceiver	1

NOTE: The path from a module pin to device pad(s) is defined at the device I/O logic level. The control module registers assign the specific function to the device pads. For more information on control module settings, see Section 13.4.6.1, *Pad Configuration Registers* of *Chapter 13, Control Module*.

18.6.2.1 CAN Network Basics

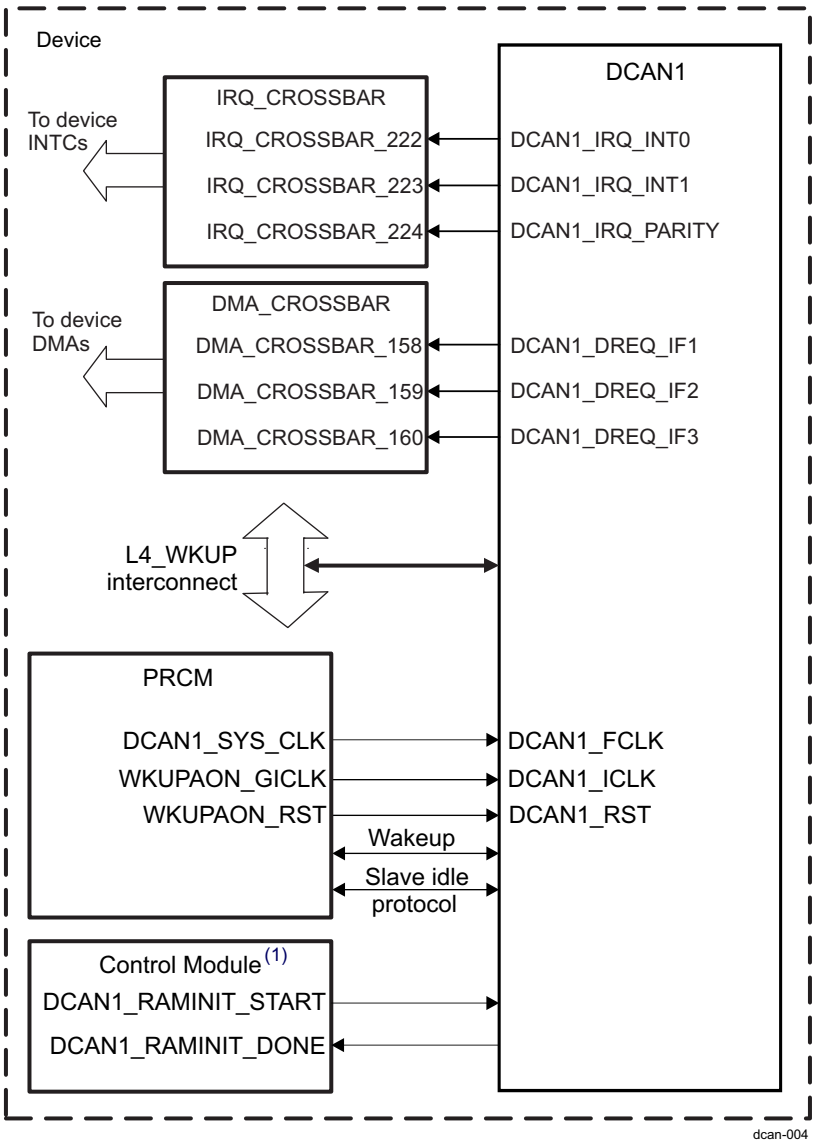
- CAN bus is a 2-wire differential bus using NRZ encoding and has two states:
  - Recessive state (logic 1)
  - Dominant state (logic 0)
- The network is multimaster. When two or more nodes attempt to transmit at the same time, a non-destructive arbitration technique guarantees messages are sent in order of priority and no messages are lost
- The message transmission is multicast. Data messages transmitted are identifier based, not address based
- Content of message is labeled by the identifier that is unique throughout the network (for example, rpm, temperature, position, pressure, etc.)
- All nodes on network receive the message and each performs an acceptance test on the identifier. If message is relevant, it is processed, otherwise it is ignored

- The unique identifier also determines the priority of the message (the lower the numerical value of the identifier, the higher the priority is)
- Data is transmitted and received using message frames, consisting of:
  - Arbitration field
  - Control field
  - Data field (0 ÷ 8 bytes)
  - CRC field
  - ACK field

18.6.3 DCAN Integration

Figure 18-115 shows the integration of the DCAN module in the device. For detailed information on the integration of DCAN see also Table 18-474 through Table 18-476.

Figure 18-115. DCAN Integration



(1) For more information on DCAN RAM Initialization see Section 18.6.4.12.1.3, *DCAN RAM Hardware Initialization*

Table 18-474 through Table 18-476 summarize the integration of the DCAN modules in the device.

Table 18-474. DCAN Integration Attributes

Module Instance	Attributes		
	Power Domain	Wake-Up Capability	Interconnect
DCAN	PD_WKUPAON	Yes	L4_WKUP

Table 18-475. DCAN Clocks and Resets

Clocks				
Module Instance	**Destination Signal Name**	**Source Signal Name**	**Source**	**Description**
DCAN	DCAN_ICLK	WKUPAON_GICLK	PRCM	Interface clock for the DCAN module
	DCAN_FCLK	DCAN1_SYS_CLK	PRCM	Functional clock for the DCAN core (CAN_CLK). Gated SYS_CLK1 or SYS_CLK2 version.
Resets				
Module Instance	**Destination Signal Name**	**Source Signal Name**	**Source**	**Description**
DCAN	DCAN_RST	WKUPAON_RST	PRCM	Asynchronous reset signal to the DCAN module

Table 18-476. DCAN Hardware Requests

Interrupt Requests				
Module Instance	**Source Signal Name**	**IRQ_CROSSBAR Input**	**Default Mapping**	**Description**
DCAN	DCAN_IRQ_INT0	IRQ_CROSSBAR_222	-	Error, Status, and Message Objects interrupt
	DCAN_IRQ_INT1	IRQ_CROSSBAR_223	-	Message Objects interrupt
	DCAN_IRQ_PARITY	IRQ_CROSSBAR_224	-	ECC double bit or single bit error (when correction is disabled) error interrupt
DMA Requests				
Module Instance	**Source Signal Name**	**DMA_CROSSBAR Input**	**Default Mapping**	**Description**
DCAN	DCAN_DREQ_IF1	DMA_CROSSBAR_158	-	DMA request for IF1 register set
	DCAN_DREQ_IF2	DMA_CROSSBAR_159	-	DMA request for IF2 register set
	DCAN_DREQ_IF3	DMA_CROSSBAR_160	-	DMA request for IF3 register set

NOTE: DCAN has no default IRQ mappings through the IRQ_CROSSBAR. For DCAN, the IRQ_CROSSBAR module must be configured prior to unmask interrupts in the interrupt controller(s).
For more information about the IRQ_CROSSBAR module, see Section 13.4.6.4, *IRQ_CROSSBAR Module Functional Description*, in Chapter 13, *Control Module*.
For more information about the device interrupt controllers, see Chapter 12, *Interrupt Controllers*.

NOTE: For more information about the DMA_CROSSBAR module, see Section 13.4.6.5, *DMA_CROSSBAR Module Functional Description*, in Chapter 13, *Control Module*.

NOTE: For the description of the interrupt source, see Section 18.6.4.2, *Interrupt Functionality*.

18.6.4 DCAN Functional Description

The DCAN module performs CAN protocol communication according to ISO 11898-1. The bit rate can be programmed to values up to 1 Mbit/s. Additional transceiver hardware is required for the connection to the physical layer (CAN bus).

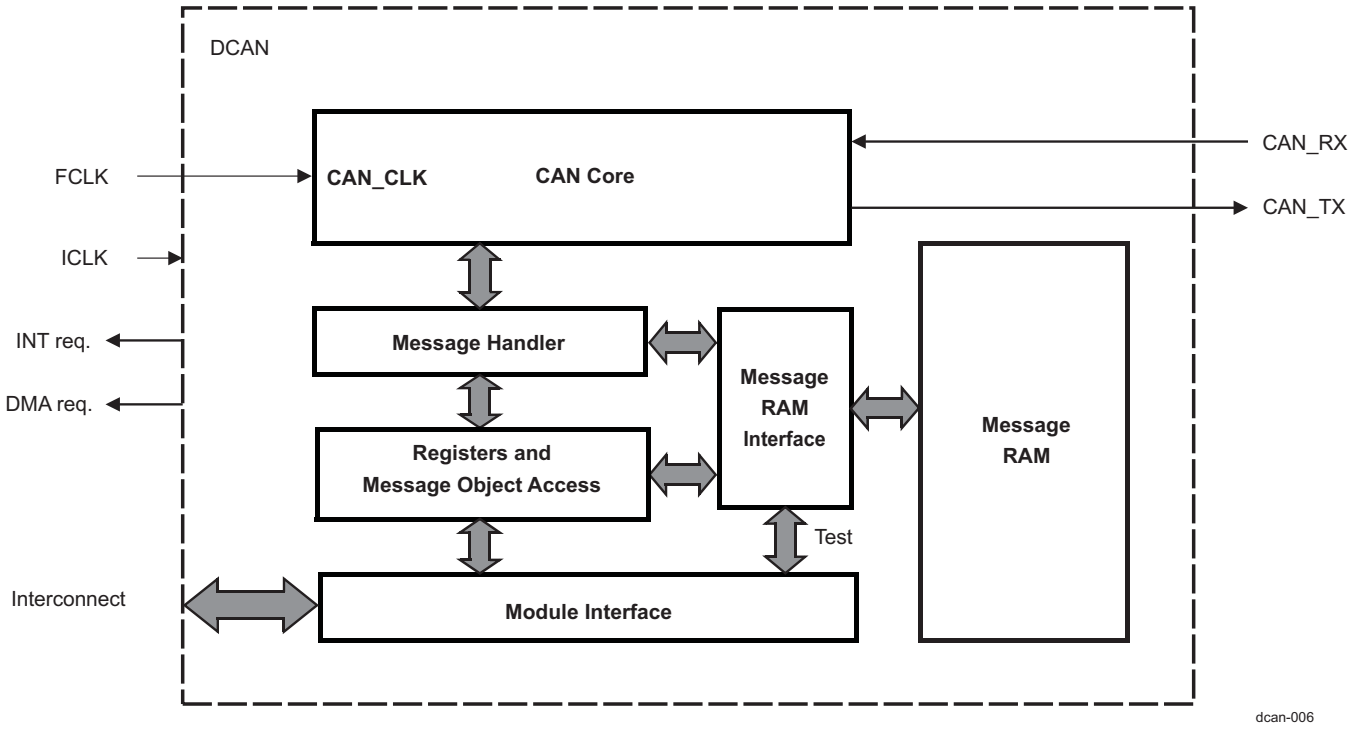
For communication on a CAN network, individual message objects can be configured. The message objects and identifier masks are stored in the message RAM.

All functions concerning the handling of messages are implemented in the message handler. Those functions are acceptance filtering, the transfer of messages between the CAN core and the message RAM, and the handling of transmission requests, as well as the generation of interrupts or DMA requests.

The register set of the DCAN module can be accessed directly via the module interface. These registers are used to control and configure the CAN core and the message handler, and to access the message RAM.

Figure 18-116 shows the DCAN block diagram and its features are described below.

Figure 18-116. DCAN Block Diagram



CAN Core: The CAN core consists of the CAN protocol controller and the Rx/Tx shift register. It handles all ISO 11898-1 protocol functions.

Message Handler: The message handler is a state machine that controls the data transfer between the single-ported message RAM and the CAN core's Rx/Tx shift register. It also handles acceptance filtering and the interrupt/DMA request generation as programmed in the control registers.

Message RAM: The DCAN enables a storage of 64 CAN messages.

Message RAM Interface: Three interface register sets control the MPU read and write accesses to the message RAM. There are two interface registers sets for read and write access, IF1 and IF2, and one interface register set for read access only, IF3. Additional information can be found in Section 18.6.4.8.12, *Reading From a FIFO Buffer*.

The interface registers have the same word-length as the message RAM.

Registers and Message Object Access: Data consistency is ensured by indirect accesses to the message objects. During normal operation, all software and DMA accesses to the message RAM are done through interface registers. In a dedicated test mode, the message RAM is memory mapped and can be directly accessed by either MPU or DMA.

Module Interface: The DCAN module registers are accessed by the user software through a 32-bit peripheral bus interface.

Clocking: Two clocks are provided to the DCAN module: the peripheral synchronous clock (interface clock [ICLK]) and the peripheral asynchronous clock (functional clock [FCLK]) .

18.6.4.1 Module Clocking Requirements

Two clocks are provided to the DCAN module:

- the peripheral synchronous clock (ICLK) as the general module clock source
- and the peripheral asynchronous clock (FCLK) provided to the CAN core for generating the CAN bit timing.

NOTE: ICLK must always be higher or equal to FCLK, in order to achieve a stable functionality of the DCAN. Here, also the frequency shift of the modulated ICLK has to be considered:

$f_{0,\ ICLK}(OCP) \pm \Delta f_{FM,\ ICLK}(OCP) \geq f_{FCLK}$

For more information on how to configure the relevant clock source registers, see Chapter 3, *PRCM* and the device data manual.

18.6.4.2 Interrupt Functionality

Interrupts can be generated on two interrupt lines: INT0 and INT1. These lines can be enabled by setting the DCAN_CTL[1] IE0 and [17] IE1 bits, respectively. The interrupts are level triggered at the chip level.

The DCAN provides three groups of interrupt sources: message object interrupts, status change interrupts, and error interrupts (see Figure 18-117, *Error and Status Change Interrupts* and Figure 18-118, *Message Objects Interrupts*).

The source of an interrupt can be determined by the interrupt identifiers DCAN_INT[15:0] INT0ID/[23:16] INT1ID. When no interrupt is pending, the register will hold the value zero.

Each interrupt line remains active until the dedicated field in the interrupt register DCAN_INT[15:0] INT0ID/[23:16] INT1ID again reach zero (this means the cause of the interrupt is reset), or until IE0/IE1 are reset.

The value 0x8000 in the INT0ID field indicates that an interrupt is pending because the CAN core has updated (not necessarily changed) the Error and Status register (DCAN_ES). This interrupt has the highest priority. The software can update (reset) the status bits [9] WAKEUPPND, [4] RXOK, [3] TXOK and [2:0] LEC by reading DCAN_ES, but a write access of the software will never generate or reset an interrupt.

Values between 1 and the number of the last message object indicates that the source of the interrupt is one of the message objects, INT0ID resp. INT1ID will point to the pending message interrupt with the highest priority. The Message Object 1 has the highest priority; the last message object has the lowest priority.

An interrupt service routine that reads the message that is the source of the interrupt may read the message and reset the message object's IntPnd at the same time (DCAN_IF1CMD/DCAN_IF2CMD[19] CLRINTPND bit). When IntPnd is cleared, DCAN_INT will point to the next message object with a pending interrupt.

18.6.4.2.1 Message Object Interrupts

Message object interrupts are generated by events from the message objects. They are controlled by the flags IntPnd, TxIE and RxIE that are described in Section 18.6.4.11.1, *Structure of Message Objects*.

Message object interrupts can be routed to either INT0 or INT1 line, controlled by the interrupt multiplexer registers (DCAN_INTMUX12 to DCAN_INTMUX78).

18.6.4.2.2 Status Change Interrupts

The events WAKEUPPND, RXOK, TXOK and LEC in the error and status register (DCAN_ES) belong to the status change interrupts. The status change interrupt group can be enabled by DCAN_CTL[2] SIE bit.

If SIE is set, a status change interrupt will be generated at each CAN frame, independent of bus errors or valid CAN communication, and also independent of the message RAM configuration.

Status change interrupts can only be routed to interrupt line INT0, which has to be enabled by setting DCAN_CTL[1] IE0 = 1.

NOTE: Reading DCAN_ES will clear the WAKEUPPND flag. If in global power-down mode, the WAKEUPPND flag is cleared by such a read access before the DCAN module has been waken up by the system, the DCAN may re-assert the WAKEUPPND flag, and a second interrupt may occur.

18.6.4.2.3 Error Interrupts

The events PER, BOFF and EWARN, monitored in the DCAN_ES register belong to the error interrupts. The error interrupt group can be enabled by setting bit DCAN_CTL[3] EIE = 1.

Error interrupts can only be routed to interrupt line INT0, which has to be enabled by setting DCAN_CTL[1] IE0 = 1.

Figure 18-117. Error and Status Change Interrupts

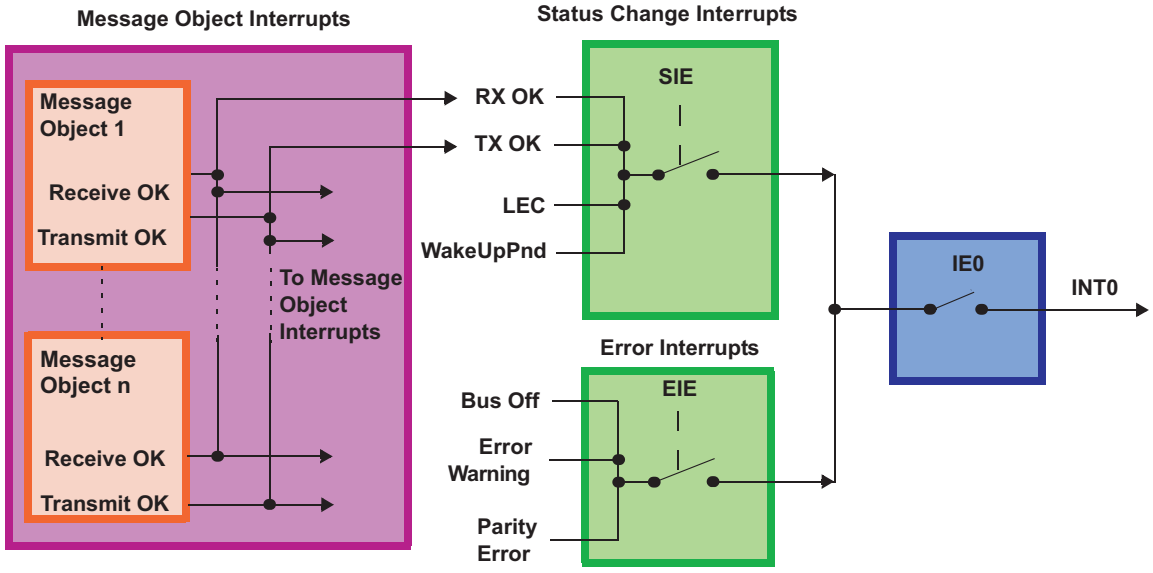
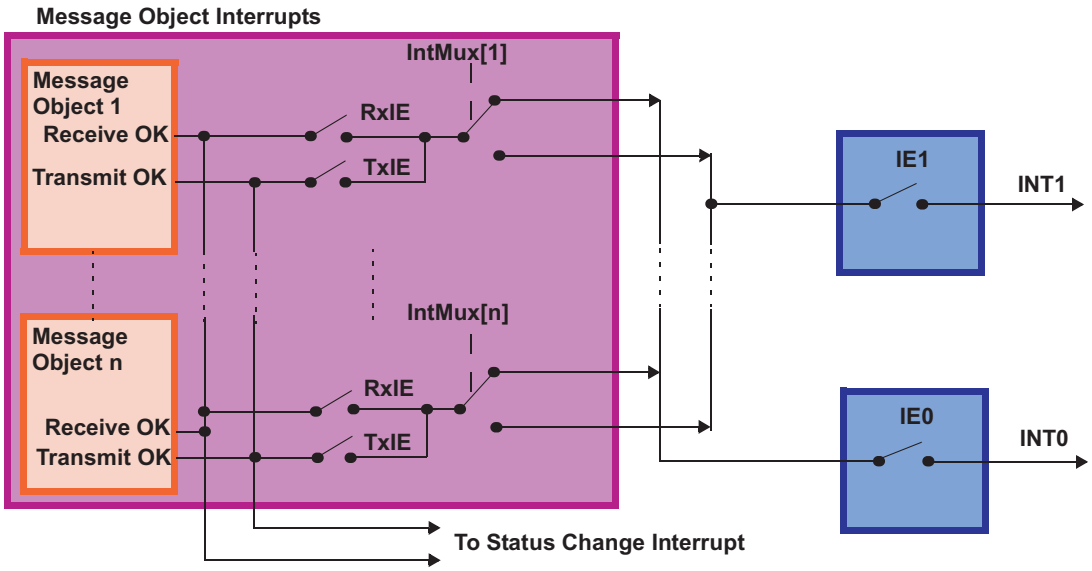


Figure 18-118. Message Objects Interrupts



18.6.4.3 DMA Functionality

The DCAN provides three DMA request lines, each indicating new data in one of the three interface register sets IF1, IF2 and IF3.

The update of IF1 and IF2 registers will be initiated by a write access to the IF1 respective IF2 Command Registers (DCAN_IF1CMD, DCAN_IF2CMD).

The IF3 registers content can be automatically updated on reception of CAN messages in message objects which are programmed for automatic IF3 update, see Section 18.6.4.10.2, *IF3 Register Set*.

When a DCAN internal IFx (x = 1 to 3) update is complete, a DMA request will be activated and stays active until the first access to one of the relevant IFx registers. The DMA functionality has to be enabled by setting bit [18] DE0/[19] DE1/[20] DE3 in DCAN_CTL.

18.6.4.4 Local Power-Down Mode

The DCAN supports a local power-down mode, which can be controlled within the DCAN registers.

18.6.4.4.1 Entering Local Power-Down Mode

The local power-down mode is requested by setting the DCAN_CTL[24] PDR bit (=1).

The DCAN then finishes all transmit requests of the message objects. When all requests are done, DCAN waits until a bus idle state is recognized. Then it will automatically set the DCAN_CTL[0] INIT bit to prevent any further CAN transfers, and it will also set the DCAN_ES[10] PDA bit. With setting the PDA bit, the DCAN module indicates that the local power-down mode has been entered.

During local power-down mode, the internal clocks of the DCAN module are turned off, but there is a wakeup logic (see Section 18.6.4.4.2, *Wakeup From Local Power Down*) that can be active, if enabled. Also, the actual contents of the control registers can be read back.

NOTE: In local low-power mode, the software must not clear the INIT bit while PDR is set. If there are any messages in the message RAM which are configured as transmit messages and the application resets the INIT bit, these messages may get sent.

18.6.4.4.2 Wakeup From Local Power Down

There are two ways to wake up the DCAN from local power-down mode:

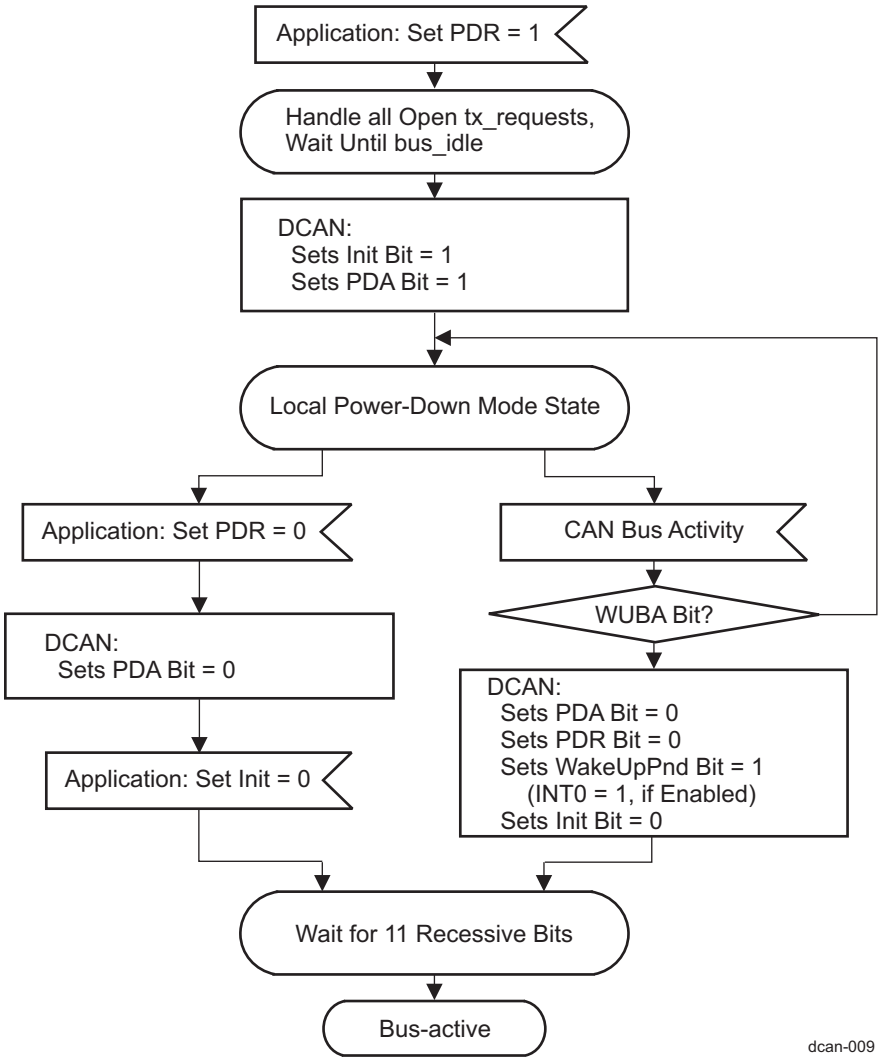
- The application could wake up the DCAN module manually by clearing the DCAN_CTL[24] PDR bit and then clearing the DCAN_CTL[0] INIT.
- Alternatively, a CAN bus activity detection circuit can be activated by setting the wakeup on bus activity bit (DCAN_CTL[25] WUBA). If this circuit is active, on occurrence of a dominant CAN bus level, the DCAN will automatically start the wakeup sequence. It will clear the DCAN_CTL[24] PDR bit and also clear the DCAN_ES[10] PDA bit. The DCAN_ES[10] WAKEUPPND bit will be set. If status interrupts are enabled, also an interrupt will be generated. Finally the DCAN_CTL[0] INIT bit will be cleared.

After the INIT bit has been cleared, the module waits until it detects 11 consecutive recessive bits on the CAN_RX pin and then goes bus-active again.

NOTE: The CAN transceiver circuit has to stay active in order to detect any CAN bus activity while the DCAN is in local power down mode. The first CAN message, which initiates the bus activity, cannot be received. This means that the first message received in power-down and automatic wake-up mode, is lost.

Figure 18-119 shows a flow diagram about entering and leaving local power-down mode.

Figure 18-119. Local Power-Down Mode Flow Diagram



18.6.4.5 SECDED Mechanism

The DCAN module provides a single error correction and double error detection (SECDED) mechanism to ensure data integrity of Message RAM data. For each message object (136 bits) in the Message RAM, 9 ECC bits will be calculated.

The ECC bits are stored in a dedicated RAM. They will be generated on write accesses and will be checked on read accesses.

The SECDED functionality can be enabled or disabled by DCAN_CTL[13:10] PMD bit field.

In case of disabled SECDED, the ECC bits will be left unchanged on write access to data area and no correction or check will be done on read access.

If SECDED is enabled, ECC bits will be automatically generated and checked. The ECC bits are memory mapped as described in Section 18.6.4.11.3, *ECC RAM*.

With the ECCMODE field in the DCAN_ECC_CS register the single bit error correction can be enabled or disabled (default: enabled).

NOTE: During RAM initialization, no SECDED check will be done, but if the PMD bit is set, the ECC bits will be generated.

18.6.4.5.1 Behavior on Single Bit Error

If a single bit error is detected with single bit error correction enabled, the correction will be done and the SEFLG in the DCAN_ECC_CS register will be set.

If single bit error correction is disabled and a single bit error is detected then the SEFLG in the DCAN_ECC_CS register and the the PER bit in the Error and Status register will be set. If error interrupts are enabled, also an interrupt would be generated. In order to avoid the transmission of invalid data over the CAN bus, the MsgVal bit of the message object will be reset.

If single bit error correction is disabled and a single bit error is detected then the IRQ_PARITY signal is generating a high pulse for one interface clock period.

The message object number where the single bit error has occurred will be indicated in the DCAN_ECC_SERR Register.

When single bit error correction is disabled the message object data can be read by the host CPU, independently of single bit errors. Thus, the application has to ensure that the read data is valid, for example, by immediately checking the DCAN_ECC_SERR on single bit error interrupt.

18.6.4.5.2 Behavior on Double Bit Error

If a double bit error is detected, then the DEFLG in the DCAN_ECC_CS register and the PER bit in DCAN_ES will be set. If error interrupts are enabled, also an interrupt would be generated. In order to avoid the transmission of invalid data over the CAN bus, the MsgVal bit of the message object will be reset. The message object number will be indicated in the DCAN_PERR register.

Additionally the signal IRQ_PARITY signalizes - by generating a high pulse for one interface clock period - the double bit error occurrence for the host CPU. The message object data can be read by the host CPU, independently of double bit errors. Thus, the application has to ensure that the read data is valid, for example, by immediately checking the DCAN_PERR register on double bit error interrupt.

18.6.4.5.3 SECDED Testing

Testing of the SECDED mechanism can be implemented by using the diagnostic mode, which is enabled with the DCAN_ECCDIAG register. The following procedure can be used:

1. Disable SECDED using DCAN_CTL register. Enable diagnostic mode using the DCAN_ECCDIAG register
2. Write to corrupt the data (in RDA mode) or ECC bits.
3. Enable SECDED and read data for which ECC is corrupted (either in RDA mode or via IFx registers).

4. Single bit error or double bit error flag will be set in the diagnostic status register (DCAN_ECCDIAG_STAT) and in the DCAN_ECC_CS register, accordingly. A double bit error or a single bit error with single bit error correction disabled also triggers the PER flag.
5. Disable diagnostic mode.

18.6.4.6 Debug/Suspend Mode

The module supports the usage of an external debug unit by providing functions like pausing DCAN activities and making message RAM content accessible via interconnect interface.

Before entering debug/suspend mode, the DCAN will either wait until a started transmission or reception will be finished and bus idle state is recognized, or immediately interrupt a current transmission or reception. This is depending on bit DCAN_CTL[8] IDS in the CAN control register.

Afterwards, the DCAN enters debug/suspend mode, indicated by DCAN_CTL [16] INITDBG flag.

During debug/suspend mode, all DCAN registers can be accessed. Reading reserved bits will return '0'. Writing to reserved bits will have no effect.

Also, the message RAM will be memory mapped. This allows the external debug unit to read the message RAM. For the memory organization, see Section 18.6.4.11.4, *Message RAM Representation in Debug/Suspend Mode*.

NOTE: During debug/suspend mode, the message RAM cannot be accessed via the IFx register sets.

NOTE: Writing to control registers in debug/suspend mode may influence the CAN state machine and further message handling.

For debug support, the auto clear functionality of the following DCAN registers is disabled:

- DCAN_ES register (clear of status flags by read)
- DCAN_IF1CMD/DCAN_IF2CMD command registers (clear of [14] DMAACTIVE flag by read/write)

18.6.4.7 Configuration of Message Objects Description

The whole message RAM should be configured before the end of the initialization, however it is also possible to change the configuration of message objects during CAN communication.

The CAN software driver must offer subroutines that:

- Transfer a complete message structure into a message object. (Configuration)
- Transfer the data bytes of a message into a message object and set TxRqst and NewDat. (Start a new transmission)
- Get the data bytes of a message from a message object and clear NewDat (and IntPnd). (Read received data)
- Get the complete message from a message object and clear NewDat (and IntPnd). (Read a received message, including identifier, from a message object with UMask = '1')

Parameters of the subroutines are the Message Number and a pointer to a complete message structure or to the data bytes of a message structure.

Two examples of assigning the IFx interface register sets to these subroutines are shown here:

In the first method, the tasks of the application program that may access the module are divided into two groups. Each group is restricted to the use of one of the interface register sets. The tasks of one group may interrupt tasks of the other group, but not of the same group.

In the second method, which may be a special case of the first method, there are only two tasks in the application program that access the module. A Read_Message task that uses IF2 or IF3 to get received messages from the message RAM and a Write_Message task that uses IF1 to write messages to be transmitted (or to be configured) into the message RAM. Both tasks may interrupt each other.

18.6.4.7.1 Configuration of a Transmit Object for Data Frames

Table 18-477 shows how a transmit object can be initialized.

Table 18-477. Initialization of a Transmit Object

MsgVal	Arb	Data	Mask	EoB	Dir	NewDat	MsgLst	RxIE	TxIE	IntPnd	RmtEn	TxRqst
1	appl.	appl.	appl.	1	1	0	0	0	appl.	0	appl.	0

The arbitration bits (ID[28:0] and Xtd bit) are given by the application. They define the identifier and type of the outgoing message. If an 11-bit identifier (standard frame) is used (Xtd = '0'), it is programmed to ID[28:18]. In this case, ID[17:0] can be ignored.

The data length and data itself (DLC[3:0] and Data0-7) are given by the application. TxRqst and RmtEn should not be set before the data is valid.

If the TXIE bit is set, the IntPnd bit will be set after a successful transmission of the message object.

If the RmtEn bit is set, a matching received remote frame will cause the TxRqst bit to be set; the remote frame will autonomously be answered by a data frame.

The mask bits (Msk[28:0], UMask, MXtd, and MDir bits) may be used (UMask='1') to allow groups of remote frames with similar identifiers to set the TxRqst bit. The Dir bit should not be masked. For details, see Section 18.6.4.8.8, *Reception of Remote Frames*.

Identifier masking must be disabled (UMask = '0') if no remote frames are allowed to set the TxRqst bit (RmtEn = '0').

18.6.4.7.2 Configuration of a Transmit Object for Remote Frames

It is not necessary to configure transmit objects for the transmission of remote frames. Setting TxRqst for a receive object causes the transmission of a remote frame with the same identifier as the data rrame for which this receive object is configured.

18.6.4.7.3 Configuration of a Single Receive Object for Data Frames

Table 18-478 shows how a receive object for data frames can be initialized.

Table 18-478. Initialization of a single Receive Object for Data Frames

MsgVal	Arb	Data	Mask	EoB	Dir	NewDat	MsgLst	RxIE	TxIE	IntPnd	RmtEn	TxRqst
1	appl.	appl.	appl.	1	0	0	0	appl.	0	0	0	0

The arbitration bits (ID[28:0] and Xtd bit) are given by the application. They define the identifier and type of accepted received messages. If an 11-bit Identifier (standard frame) is used (Xtd = '0'), it is programmed to ID[28:18]. In this case, ID[17:0] can be ignored. When a data frame with an 11-bit Identifier is received, ID[17:0] is set to '0'.

The data length code (DLC[3:0]) is given by the application. When the message handler stores a data frame in the message object, it will store the received data length code and eight data bytes. If the data length code is less than 8, the remaining bytes of the message object may be overwritten by non specified values.

The mask bits (Msk[28:0], UMask, MXtd, and MDir bits) may be used (UMask = '1') to allow groups of data frames with similar identifiers to be accepted. The Dir bit should not be masked in typical applications. If some bits of the mask bits are set to "don't care," the corresponding bits of the arbitration register (DCAN_IF1ARB, DCAN_IF2ARB, DCAN_IF3ARB) will be overwritten by the bits of the stored data frame.

If the RxIE bit is set, the IntPnd bit will be set when a received data frame is accepted and stored in the message object.

If the TxRqst bit is set, the transmission of a remote frame with the same identifier as actually stored in the arbitration bits will be triggered. The content of the arbitration bits may change if the mask bits are used (UMask = '1') for acceptance filtering.

18.6.4.7.4 Configuration of a Single Receive Object for Remote Frames

Table 18-479 shows how a receive object for remote frames can be initialized.

Table 18-479. Initialization of a Single Receive Object for Remote Frames

MsgVal	Arb	Data	Mask	EoB	Dir	NewDat	MsgLst	RxIE	TxIE	IntPnd	RmtEn	TxRqst
1	appl.	appl.	appl.	1	1	0	0	appl.	0	0	0	0

A receive object for remote frames may be used to monitor remote frames on the CAN bus. The remote frame stored in the receive object will not trigger the transmission of a data frame. Receive objects for remote frames may be expanded to a FIFO buffer (see Section 18.6.4.7.5, *Configuration of a FIFO Buffer*).

UMask must be set to '1.' The mask bits (Msk[28:0], UMask, MXtd, and MDir bits) may be set to "must-match" or to "don't care," to allow groups of remote frames with similar identifiers to be accepted. The Dir bit should not be masked in typical applications. For details, see Section 18.6.4.8.8, *Reception of Remote Frames*.

The arbitration bits (ID[28:0] and Xtd bit) may be given by the application. They define the identifier and type of accepted received remote frames. If some bits of the mask bits are set to "don't care," the corresponding bits of the arbitration bits will be overwritten by the bits of the stored remote frame. If an 11-bit Identifier (standard frame) is used (Xtd = '0'), it is programmed to ID[28:18]. In this case, ID[17:0] can be ignored. When a remote frame with an 11-bit Identifier is received, ID[17:0] will be set to '0.'

The data length code (DLC[3:0]) may be given by the application. When the message handler stores a remote frame in the message object, it will store the received data length code. The data bytes of the message object will remain unchanged.

If the RxIE bit is set, the IntPnd bit will be set when a received remote frame is accepted and stored in the message object.

18.6.4.7.5 Configuration of a FIFO Buffer

With the exception of the EoB bit, the configuration of receive objects belonging to a FIFO buffer is the same as the configuration of a single receive object.

To concatenate multiple message objects to a FIFO buffer, the identifiers and masks (if used) of these message objects have to be programmed to matching values. Due to the implicit priority of the message objects, the message object with the lowest number will be the first message object of the FIFO buffer. The EoB bit of all message objects of a FIFO buffer except the last one have to be programmed to zero. The EoB bits of the last message object of a FIFO Buffer is set to one, configuring it as the end of the block.

18.6.4.8 Message Handling

When initialization is finished, the DCAN module synchronizes itself to the traffic on the CAN bus. It does acceptance filtering on received messages and stores those frames that are accepted into the designated message objects. The application has to update the data of the messages to be transmitted and to enable and request their transmission. The transmission is requested automatically when a matching remote frame is received.

The application may read messages which are received and accepted. Messages that are not read before the next messages is accepted for the same message object will be overwritten.

Messages may be read interrupt-driven or after polling of NewDat.

18.6.4.8.1 Message Handler Overview

The message handler state machine controls the data transfer between the Rx/Tx shift register of the CAN core and the message RAM. It performs the following tasks:

- Data transfer from message RAM to CAN core (messages to be transmitted)
- Data transfer from CAN core to the message RAM (received messages)
- Data transfer from CAN core to the acceptance filtering unit
- Scanning of message RAM for a matching message object (acceptance filtering)
- Scanning the same message object after being changed by IF1/IF2 registers when priority is the same or higher as the message the object found by last scanning
- Handling of TxRqst flags
- Handling of interrupt flags

The message handler registers contains status flags of all message objects grouped into the following topics:

- Transmission Request Flags
- New Data Flags
- Interrupt Pending Flags
- Message Valid Registers

Instead of collecting above listed status information of each message object via IFx registers separately, these message handler registers provides a fast and easy way to get an overview, for example, about all pending transmission requests.

All message handler registers are read-only.

18.6.4.8.2 Receive/Transmit Priority

The receive/transmit priority for the message objects is attached to the message number, not to the CAN identifier. Message object 1 has the highest priority, while the last implemented message object has the lowest priority. If more than one transmission request is pending, they are serviced due to the priority of the corresponding message object so messages with the highest priority, for example, can be placed in the message objects with the lowest numbers.

The acceptance filtering for received data frames or remote frames is also done in ascending order of message objects, so a frame that has been accepted by a message object cannot be accepted by another message object with a higher message number. The last message object may be configured to accept any data frame or remote frame that was not accepted by any other message object, for nodes that need to log the complete message traffic on the CAN bus.

18.6.4.8.3 Transmission of Messages in Event Driven CAN Communication

If the shift register of the CAN core is ready for loading and if there is no data transfer between the IFx registers and message RAM, the MSGVAL bits in the Message Valid register (DCAN_MSGVAL12 to DCAN_MSGVAL78) and the TXRQST bits in the transmission request register (DCAN_TXRQ12 to DCAN_TXRQ78) are evaluated. The valid message object with the highest priority pending transmission request is loaded into the shift register by the message handler and the transmission is started. The message object's NewDat bit is reset.

After a successful transmission and if no new data was written to the message object (NewDat = '0') since the start of the transmission, the TxRqst bit will be reset. If TxIE is set, IntPnd will be set after a successful transmission. If the DCAN has lost the arbitration or if an error occurred during the transmission, the message will be retransmitted as soon as the CAN bus is free again. If meanwhile the transmission of a message with higher priority has been requested, the messages will be transmitted in the order of their priority.

If automatic retransmission mode is disabled by setting the DCAN_CTL[5] DAR bit, the behavior of bits TXRQST and NEWDAT in the DCAN_IF1CMD/DCAN_IF2CMD register of the interface register set is as follows:

- When a transmission starts, the TxRqst bit of the respective interface register set is reset, while bit

NewDat remains set.

- When the transmission has been successfully completed, the NewDat bit is reset.

When a transmission failed (lost arbitration or error), bit NewDat remains set. To restart the transmission, the application has to set TxRqst again.

Received remote frames do not require a receive object. They will automatically trigger the transmission of a data frame, if in the matching transmit object the RmtEn bit is set.

18.6.4.8.4 Updating a Transmit Object

The software may update the data bytes of a transmit object any time via the IF1/IF2 interface registers, neither MSGVAL, nor TXRQST have to be reset before the update.

Even if only a part of the data bytes are to be updated, all four bytes in the corresponding IF1/IF2 Data A register (DCAN_IF1DATA/DCAN_IF2DATA) or IF1/IF2 Data B register (DCAN_IF1DATB/DCAN_IF2DATB) have to be valid before the content of that register is transferred to the message object. Either the software has to write all four bytes into the IF1/IF2 data register or the message object is transferred to the IF1/IF2 data register before the software writes the new data bytes.

When only the data bytes are updated, first 0x87 can be written to bits [23:16] of the IF1/IF2 Command register (DCAN_IF1CMD/DCAN_IF2CMD) and then the number of the message object is written to bits [7:0] MESSAGE_NUMBER of the command register, concurrently updating the data bytes and setting TXRQST with NEWDAT.

To prevent the reset of TxRqst at the end of a transmission that may already be in progress while the data is updated, NewDat has to be set together with TxRqst in event driven CAN communication. For details, see Section 18.6.4.8.3, *Transmission of Messages in Event Driven CAN Communication*.

When NewDat is set together with TxRqst, NewDat will be reset as soon as the new transmission has started.

18.6.4.8.5 Changing a Transmit Object

If the number of implemented message objects is not sufficient to be used as permanent message objects only, the transmit objects may be managed dynamically. The software can write the whole message (arbitration, control, and data) into the interface register. The bits [23:16] of the command register (DCAN_IF1CMD/DCAN_IF2CMD) can be set to 0xB7 for the transfer of the whole message object content into the message object. Neither Dir, nor TxRqst have to be reset before this operation.

If a previously requested transmission of this message object is not completed but already in progress, it will be continued; however, it will not be repeated if it is disturbed.

To only update the data bytes of a message to be transmitted, bits [23:16] of the command register (DCAN_IF1CMD/ DCAN_IF2CMD) should be set to 0x87.

NOTE: After the update of the transmit object, the interface register set will contain a copy of the actual contents of the object, including the part that had not been updated.

18.6.4.8.6 Acceptance Filtering of Received Messages

When the arbitration and control bits (Identifier + IDE + RTR + DLC) of an incoming message are completely shifted into the shift register of the CAN core, the message handler starts the scan of the message RAM for a matching valid message object:

- The acceptance filtering unit is loaded with the arbitration bits from the CAN core shift register.
- Then the arbitration and mask bits (including MsgVal, UMask, NewDat, and EoB) of message object 1 are loaded into the acceptance filtering unit and are compared with the arbitration bits from the shift register. This is repeated for all following message objects until a matching message object is found, or until the end of the message RAM is reached.
- If a match occurs, the scanning is stopped and the message handler proceeds depending on the type of the frame (data frame or remote frame) received.

18.6.4.8.7 Reception of Data Frames

The message handler stores the message from the CAN core shift register into the respective message object in the message RAM. Not only the data bytes, but all arbitration bits and the data length code are stored into the corresponding message object. This ensures that the data bytes stay associated to the identifier even if arbitration mask registers are used.

The NewDat bit is set to indicate that new data (not yet seen by the software) has been received. The software should reset the NewDat bit when it reads the message object. If at the time of the reception the NewDat bit was already set, MsgLst is set to indicate that the previous data (supposedly not seen by the software) is lost. If the RxIE bit is set, the IntPnd bit is set, causing the DCAN_INT to point to this message object.

The TxRqst bit of this message object is reset to prevent the transmission of a remote frame, while the requested data frame has just been received.

18.6.4.8.8 Reception of Remote Frames

When a remote frame is received, three different configurations of the matching message object have to be considered:

- Dir = '1' (direction = transmit), RmtEn = '1', UMask = '1' or '0'

  The TxRqst bit of this message object is set at the reception of a matching remote frame. The rest of the message object remains unchanged.
- Dir = '1' (direction = transmit), RmtEn = '0', UMask = '0'

  The remote frame is ignored, this message object remains unchanged.
- Dir = '1' (direction = transmit), RmtEn = '0', UMask = '1'

  The remote frame is treated similar to a received data frame. At the reception of a matching Remote Frame, the TxRqst bit of this message object is reset. The arbitration and control bits (Identifier + IDE + RTR + DLC) from the shift register are stored in the message object in the message RAM and the NewDat bit of this message object is set. The data bytes of the message object remain unchanged.

18.6.4.8.9 Reading Received Messages

The software may read a received message any time via the IFx interface register. The data consistency is guaranteed by the message handler state machine. Typically the software will write first 0x7F to bits [23:16] and then the number of the message object to bits [7:0] MESSAGE_NUMBER of the command register (DCAN_IF1CMD/ DCAN_IF2CMD). That combination will transfer the entire received message from the message RAM into the interface register set. Additionally, the bits NewDat and IntPnd are cleared in the message RAM (not in the interface register set). The values of these bits in the message control register (DCAN_IF1MCTL/DCAN_IF2MCTL/DCAN_IF3MCTL) always reflect the status before resetting the bits. If the message object uses masks for acceptance filtering, the arbitration bits show which of the different matching messages has been received.

The actual value of NewDat shows whether a new message has been received since last time when this message object was read. The actual value of MsgLst shows whether more than one message has been received since the last time when this message object was read. MsgLst will not be automatically reset.

18.6.4.8.10 Requesting New Data for a Receive Object

By means of a remote frame, the software may request another CAN node to provide new data for a receive object. Setting the TxRqst bit of a receive object will cause the transmission of a remote frame with the identifier of the receive object. This remote frame triggers the other CAN node to start the transmission of the matching data frame. If the matching data frame is received before the remote frame could be transmitted, the TxRqst bit is automatically reset. Setting the TxRqst bit without changing the contents of a message object requires the value 0x84 in bits [23:16] of the command register (DCAN_IF1CMD/ DCAN_IF2CMD).

18.6.4.8.11 Storing Received Messages in FIFO Buffers

Several message objects may be grouped to form one or more FIFO buffers. Each FIFO buffer configured to store received messages with a particular (group of) identifier(s). arbitration and mask registers of the FIFO buffer's message objects are identical. The end of buffer (EoB) bits of all but the last of the FIFO buffer's message objects are '0'; in the last one the EoB bit is '1.'

Received messages with identifiers matching to a FIFO buffer are stored into a message object of this FIFO buffer, starting with the message object with the lowest message number. When a message is stored into a message object of a FIFO buffer, the NewDat bit of this message object is set. By setting NewDat while EoB is '0', the message object is locked for further write accesses by the message handler until the software has cleared the NewDat bit.

Messages are stored into a FIFO buffer until the last message object of this FIFO buffer is reached. If none of the preceding message objects is released by writing NewDat to '0,' all further messages for this FIFO buffer will be written into the last message object of the FIFO buffer (EoB = '1') and therefore overwrite previous messages in this message object.

18.6.4.8.12 Reading From a FIFO Buffer

Several messages may be accumulated in a set of message objects which are concatenated to form a FIFO buffer before the application program is required (in order to avoid the loss of data) to empty the buffer.

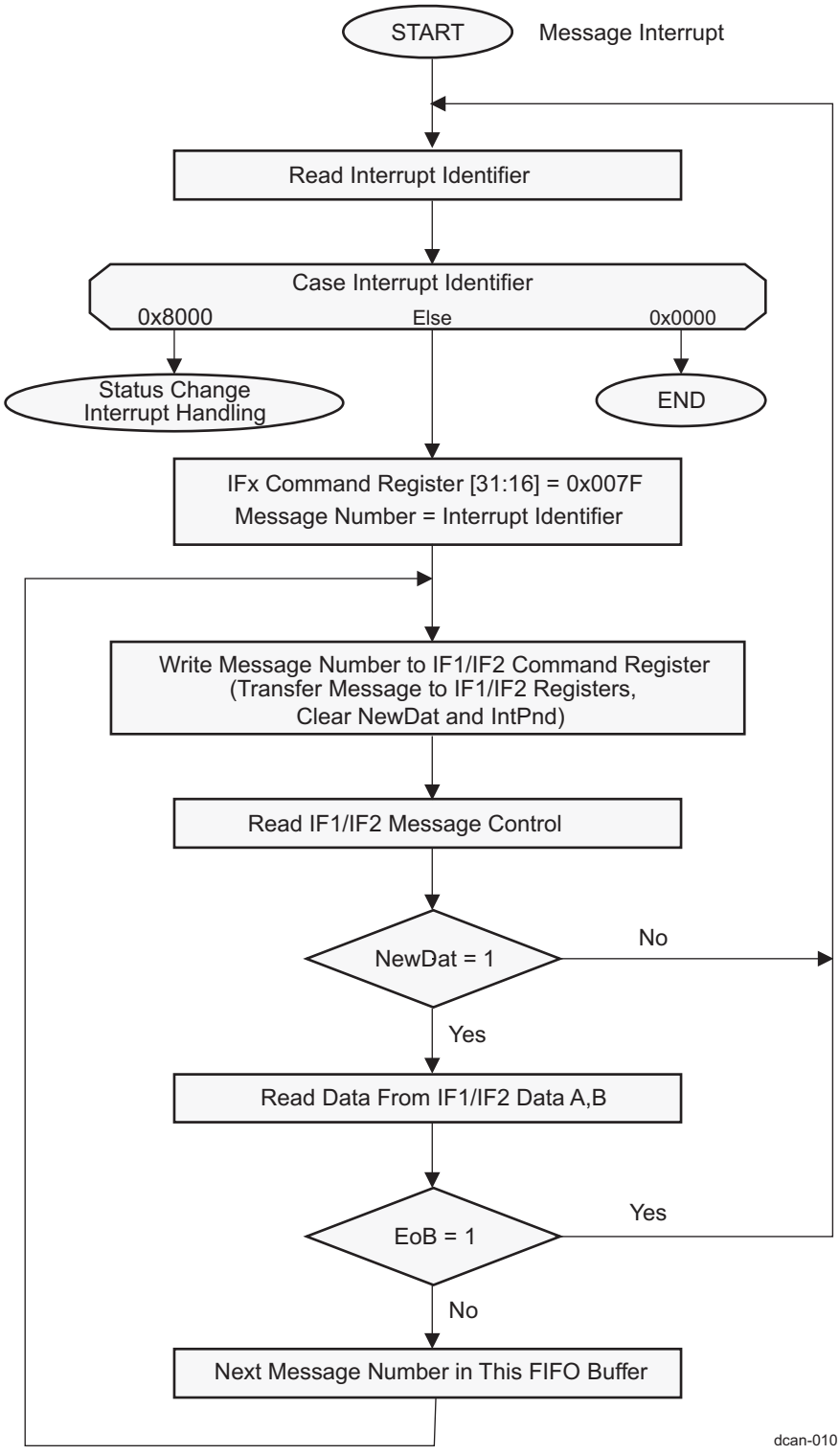
A FIFO buffer of length N will store N–1, plus the last received message since last time it was cleared.

A FIFO buffer is cleared by reading and resetting the NewDat bits of all its message objects, starting at the FIFO Object with the lowest message number. This should be done in a subroutine following the example shown in Figure 18-120.

NOTE: All message objects of a FIFO buffer needs to be read and cleared before the next batch of messages can be stored. Otherwise, true FIFO functionality can not be guaranteed, since the message objects of a partly read buffer will be re-filled according to the normal (descending) priority.

Reading from a FIFO buffer message object and resetting its NewDat bit is handled the same way as reading from a single message object.

Figure 18-120. Software Handling of a FIFO Buffer (Interrupt Driven)



18.6.4.9 CAN Bit Timing

The DCAN supports bit rates between < 1 kBit/s and 1000 kBit/s.

Each member of the CAN network has its own clock generator, typically derived from a crystal oscillator. The bit timing parameters can be configured individually for each CAN node, creating a common bit rate even though the CAN nodes' oscillator periods (f_{osc}) may be different.

The frequencies of these oscillators are not absolutely stable. Small variations are caused by changes in temperature or voltage and by deteriorating components. As long as the variations remain inside a specific oscillator tolerance range (df), the CAN nodes are able to compensate for the different bit rates by resynchronizing to the bit stream.

In many cases, the CAN bit synchronization will amend a faulty configuration of the CAN bit timing to such a degree that only occasionally an error frame is generated. In the case of arbitration however, when two or more CAN nodes simultaneously try to transmit a frame, a misplaced sample point may cause one of the transmitters to become error passive.

The analysis of such sporadic errors requires a detailed knowledge of the CAN bit synchronization inside a CAN node and of the CAN nodes' interaction on the CAN bus.

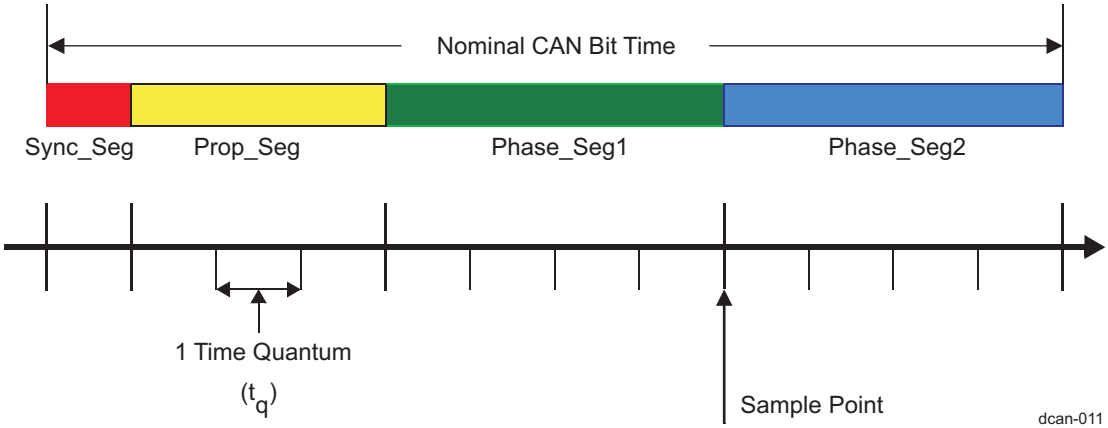
Even if minor errors in the configuration of the CAN bit timing do not result in immediate failure, the performance of a CAN network can be reduced significantly.

18.6.4.9.1 Bit Time and Bit Rate

According to the CAN specification, the bit time is divided into four segments (see Figure 18-121):

- Synchronization segment (Sync_Seg)
- Propagation time segment (Prop_Seg)
- Phase buffer segment 1 (Phase_Seg1)
- Phase buffer segment 2 (Phase_Seg2)

Figure 18-121. Bit Timing



Each segment consists of a specific number of time quanta. The length of one time quantum (t_q), which is the basic time unit of the bit time, is given by the FCLK and the baud rate prescalers (BRPE and BRP). With these two baud rate prescalers combined, divider values from 1 to 1024 can be programmed:

t_q = Baud Rate Prescaler / FCLK

Apart from the fixed length of the synchronization segment, these numbers are programmable. Table 18-480 describes the minimum programmable ranges required by the CAN protocol.

A given bit rate may be met by different bit time configurations.

Table 18-480. Parameters of the CAN Bit Time

Parameter	Range	Remark
Sync_Seg	1 t_q (fixed)	Synchronization of bus input to FCLK
Prop_Seg	[1 … 8] t_q	Compensates for the physical delay times
Phase_Seg1	[1 … 8] t_q	May be lengthened temporarily by synchronization

Table 18-480. Parameters of the CAN Bit Time (continued)

Parameter	Range	Remark
Phase_Seg2	[1 … 8] t_q	May be shortened temporarily by synchronization
Synchronization Jump Width (SJW)	[1 … 4] t_q	May not be longer than either Phase Buffer Segment

NOTE: For proper functionality of the CAN network, the physical delay times and the oscillator's tolerance range have to be considered.

18.6.4.9.1.1 Synchronization Segment

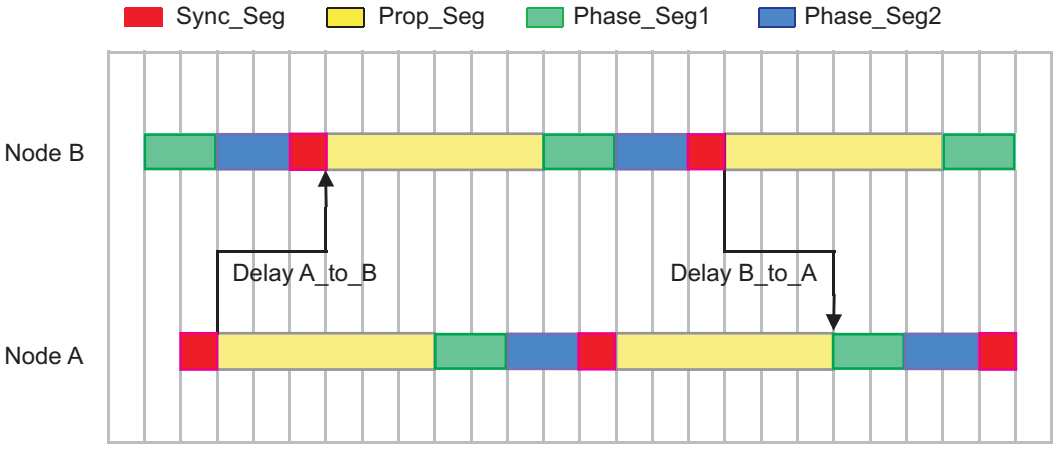
The synchronization segment (Sync_Seg) is the part of the bit time where edges of the CAN bus level are expected to occur. If an edge occurs outside of Sync_Seg, its distance to the Sync_Seg is called the phase error of this edge.

18.6.4.9.1.2 Propagation Time Segment

This part of the bit time is used to compensate physical delay times within the CAN network. These delay times consist of the signal propagation time on the bus and the internal delay time of the CAN nodes.

Any CAN node synchronized to the bit stream on the CAN bus can be out of phase with the transmitter of the bit stream, caused by the signal propagation time between the two nodes. The CAN protocol's nondestructive bitwise arbitration and the dominant acknowledge bit provided by receivers of CAN messages require that a CAN node transmitting a bit stream must also be able to receive dominant bits transmitted by other CAN nodes that are synchronized to that bit stream. The example in Figure 18-122 shows the phase shift and propagation times between two CAN nodes.

Figure 18-122. The Propagation Time Segment



Delay A_to_B ≥ node output delay(A) + bus line delay(AÆB) + node input delay(B)

Prop_Seg ≥ Delay A_to_B + Delay B_to_A

Prop_Seg ≥ 2 [max(node output delay + bus line delay + node input delay)]

dcan-012

In this example, both nodes A and B are transmitters performing an arbitration for the CAN bus. The node A has sent its start of frame bit less than one bit time earlier than node B, therefore node B has synchronized itself to the received edge from recessive to dominant. Since node B has received this edge delay(A_to_B) after it has been transmitted, node B's bit timing segments are shifted with regard to node A. Node B sends an identifier with higher priority and so it will win the arbitration at a specific identifier bit when it transmits a dominant bit while node A transmits a recessive bit. The dominant bit transmitted by node B will arrive at node A after the delay (B_to_A).

Due to oscillator tolerances, the actual position of node A's sample point can be anywhere inside the nominal range of node A's Phase Buffer Segments, so the bit transmitted by node B must arrive at node A before the start of Phase_Seg1. This condition defines the length of Prop_Seg.

If the edge from recessive to dominant transmitted by node B would arrive at node A after the start of Phase_Seg1, it could happen that node A samples a recessive bit instead of a dominant bit, resulting in a bit error and the destruction of the current frame by an error flag.

This error only occurs when two nodes arbitrate for the CAN bus which have oscillators of opposite ends of the tolerance range and are separated by a long bus line; this is an example of a minor error in the bit timing configuration (Prop_Seg too short) that causes sporadic bus errors.

Some CAN implementations provide an optional 3-Sample Mode. The DCAN does not. In this mode, the CAN bus input signal passes a digital low-pass filter, using three samples and a majority logic to determine the valid bit value. This results in an additional input delay of 1 t_q, requiring a longer Prop_Seg.

18.6.4.9.1.3 Phase Buffer Segments and Synchronization

The phase buffer segments (Phase_Seg1 and Phase_Seg2) and the synchronization jump width (SJW) are used to compensate for the oscillator tolerance.

The phase buffer segments surround the sample point and may be lengthened or shortened by synchronization.

The synchronization jump width (SJW) defines how far the resynchronizing mechanism may move the sample point inside the limits defined by the phase buffer segments to compensate for edge phase errors.

Synchronizations occur on edges from recessive to dominant. Their purpose is to control the distance between edges and sample points.

Edges are detected by sampling the actual bus level in each time quantum and comparing it with the bus level at the previous sample point. A synchronization may be done only if a recessive bit was sampled at the previous sample point and if the actual time quantum's bus level is dominant.

An edge is synchronous if it occurs inside of Sync_Seg; otherwise, its distance to the Sync_Seg is the edge phase error, measured in time quanta. If the edge occurs before Sync_Seg, the phase error is negative, else it is positive.

Two types of synchronization exist: hard synchronization and resynchronizing. A hard synchronization is done once at the start of a frame; inside a frame, only resynchronization is possible.

- Hard Synchronization

  After a hard synchronization, the bit time is restarted with the end of Sync_Seg, regardless of the edge phase error. Thus hard synchronization forces the edge which has caused the hard synchronization, to lie within the synchronization segment of the restarted bit time.

- Bit Resynchronizations

  Resynchronization leads to a shortening or lengthening of the Bit time such that the position of the sample point is shifted with regard to the edge.

  When the phase error of the edge which causes resynchronization is positive, Phase_Seg1 is lengthened. If the magnitude of the phase error is less than SJW, Phase_Seg1 is lengthened by the magnitude of the phase error, else it is lengthened by SJW.

  When the phase error of the edge which causes Resynchronization is negative, Phase_Seg2 is shortened. If the magnitude of the phase error is less than SJW, Phase_Seg2 is shortened by the magnitude of the phase error, else it is shortened by SJW.

If the magnitude of the phase error of the edge is less than or equal to the programmed value of SJW, the results of hard synchronization and resynchronization are the same. If the magnitude of the phase error is larger than SJW, the resynchronization cannot compensate the phase error completely, and an error of (phase error - SJW) remains.

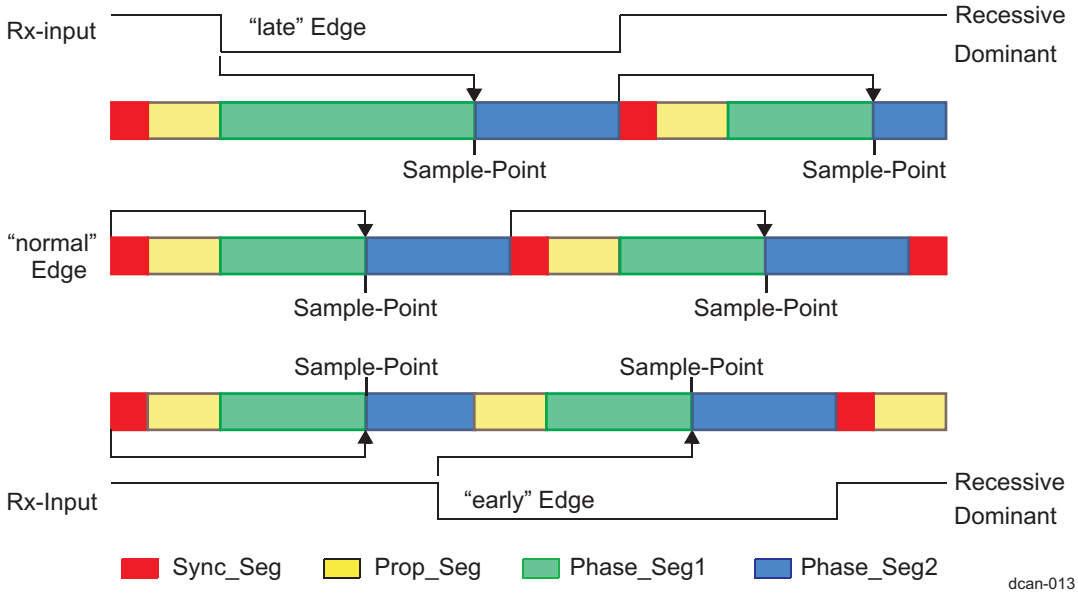
Only one synchronization may be done between two sample points. The synchronizations maintain a minimum distance between edges and sample points, giving the bus level time to stabilize and filtering out spikes that are shorter than (Prop_Seg + Phase_Seg1).

Apart from noise spikes, most synchronizations are caused by arbitration. All nodes synchronize “hard” on the edge transmitted by the “leading” transceiver that started transmitting first, but due to propagation delay times, they cannot become ideally synchronized. The leading transmitter does not necessarily win the arbitration; therefore, the receivers have to synchronize themselves to different transmitters that subsequently take the lead and that are differently synchronized to the previously leading transmitter. The same happens at the acknowledge field, where the transmitter and some of the receivers will have to synchronize to that receiver that takes the lead in the transmission of the dominant acknowledge bit.

Synchronizations after the end of the arbitration will be caused by oscillator tolerance, when the differences in the oscillator’s clock periods of transmitter and receivers sum up during the time between synchronizations (at most ten bits). These summarized differences may not be longer than the SJW, limiting the oscillator’s tolerance range.

Figure 18-123 shows how the phase buffer segments are used to compensate for phase errors. There are three drawings of each two consecutive bit timings. The upper drawing shows the synchronization on a “late” edge, the lower drawing shows the synchronization on an “early” edge, and the middle drawing is the reference without synchronization.

Figure 18-123. Synchronization on Late and Early Edges



In the first example, an edge from recessive to dominant occurs at the end of Prop_Seg. The edge is "late" since it occurs after the Sync_Seg. Reacting to the late edge, Phase_Seg1 is lengthened so that the distance from the edge to the sample point is the same as it would have been from the Sync_Seg to the sample point if no edge had occurred. The phase error of this late edge is less than SJW, so it is fully compensated and the edge from dominant to recessive at the end of the bit, which is one nominal bit time long, occurs in the Sync_Seg.

In the second example, an edge from recessive to dominant occurs during Phase_Seg2. The edge is "early" since it occurs before a Sync_Seg. Reacting to the early edge, Phase_Seg2 is shortened and Sync_Seg is omitted, so that the distance from the edge to the sample point is the same as it would have been from a Sync_Seg to the sample point if no edge had occurred. As in the previous example, the magnitude of this early edge’s phase error is less than SJW, so it is fully compensated.

The phase buffer segments are lengthened or shortened temporarily only; at the next bit time, the segments return to their nominal programmed values.

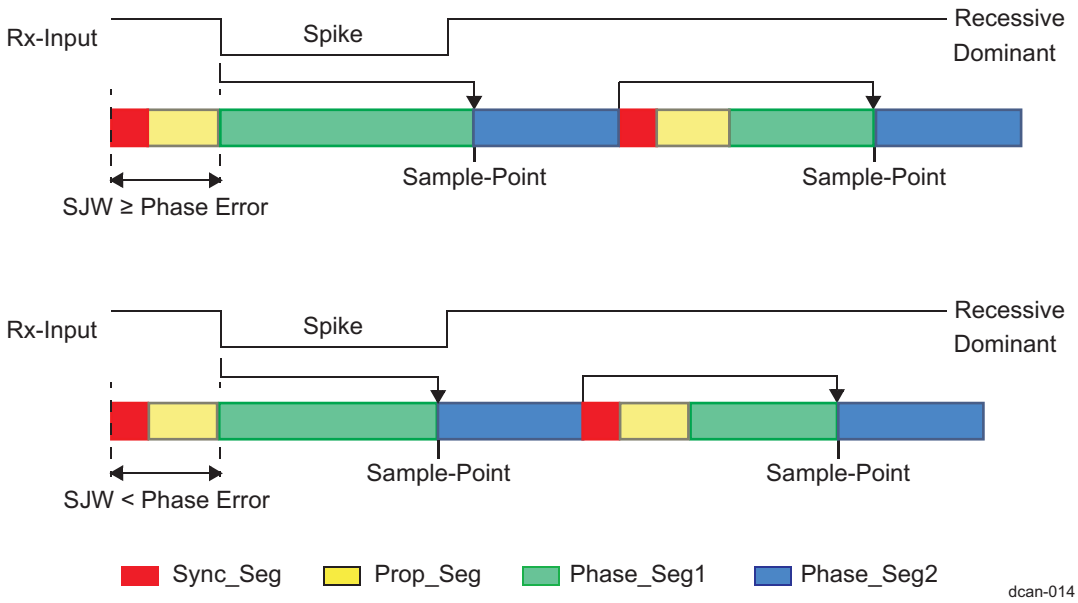
In these examples, the bit timing is seen from the point of view of the CAN implementation’s state machine, where the bit time starts and ends at the sample points. The state machine omits Sync_Seg when synchronizing on an early edge because it cannot subsequently redefine that time quantum of Phase_Seg2 where the edge occurs to be the Sync_Seg.

Figure 18-124 shows how short dominant noise spikes are filtered by synchronizations. In both examples, the spike starts at the end of Prop_Seg and has the length of (Prop_Seg + Phase_Seg1).

In the first example, the synchronization jump width is greater than or equal to the phase error of the spike's edge from recessive to dominant. Therefore the sample point is shifted after the end of the spike; a recessive bus level is sampled.

In the second example, SJW is shorter than the phase error, so the sample point cannot be shifted far enough; the dominant spike is sampled as actual bus level.

Figure 18-124. Filtering of Short Dominant Spikes



18.6.4.9.1.4 Oscillator Tolerance Range

With the introduction of CAN protocol version 1.2, the option to synchronize on edges from dominant to recessive became obsolete. Only edges from recessive to dominant are considered for synchronization. The protocol update to version 2.0 (A and B) had no influence on the oscillator tolerance.

The tolerance range df for an oscillator's frequency f_{osc} around the nominal frequency f_{nom} with:

$$(1 - df) \bullet f_{nom} \le f_{osc} \le (1 + df) \bullet f_{nom}$$

dcan-e015

depends on the proportions of Phase_Seg1, Phase_Seg2, SJW, and the bit time. The maximum tolerance df is the defined by two conditions (both shall be met):

$$\text{I:} \quad df \le \frac{\min(TSeg1, TSeg2)}{2x\left(13x\left(bit\_time - TSeg2\right)\right)}$$

$$\text{II:} \quad df \le \frac{SJW}{20xbit\_time}$$

dcan-e016

It has to be considered that SJW may not be larger than the smaller of the phase buffer segments and that the propagation time segment limits that part of the bit time that may be used for the phase buffer segments.

The combination Prop_Seg = 1 and Phase_Seg1 = Phase_Seg2 = SJW = 4 allows the largest possible oscillator tolerance of 1.58%. This combination with a propagation time segment of only 10% of the bit time is not suitable for short bit times; it can be used for bit rates of up to 125 kBit/s (bit time = 8 µs) with a bus length of 40 m.

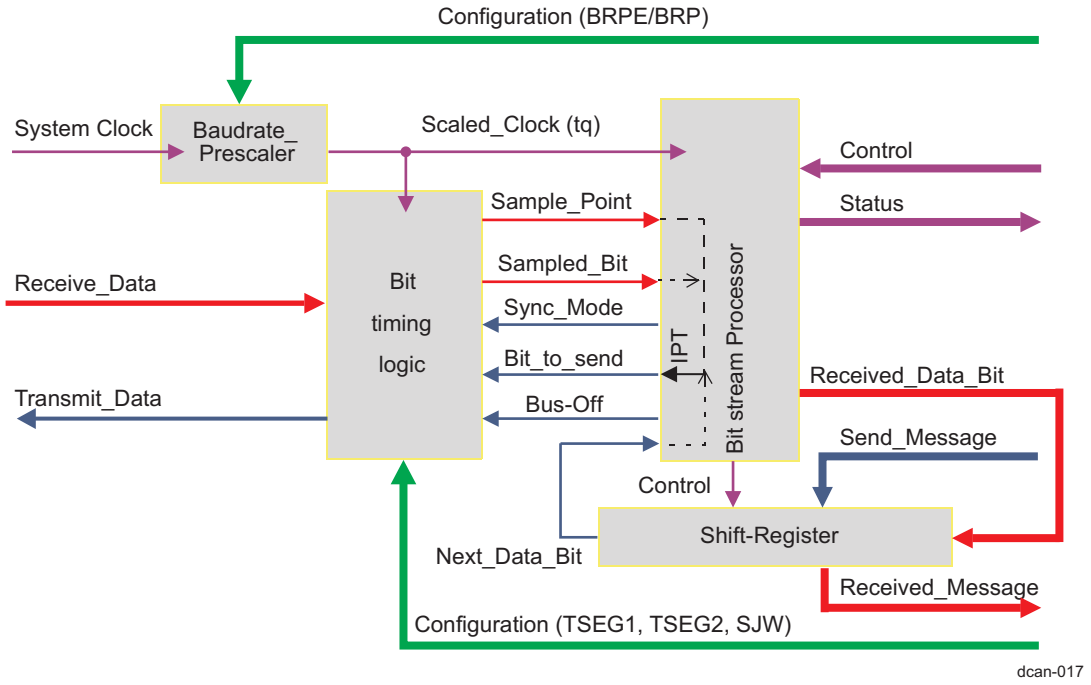
18.6.4.9.2 DCAN Bit Timing Registers

In the DCAN, the bit timing configuration is programmed in DCAN_BTR[14:0], additionally a baud rate prescaler extension of four bits (DCAN_BTR[19:16] BRPE) is provided.

- The sum of Prop_Seg and Phase_Seg1 is set in [11:8] TSEG1
- Phase_Seg2 in [14:12] TSEG2
- SynchronizationJumpWidth in [7:6] SJW
- and baud rate prescaler [5:0] BRP (plus [19:16] BRPE)

Figure 18-125 shows the CAN protocol controller.

Figure 18-125. Structure of the CAN Core's CAN Protocol Controller



In DCAN_BTR register, the components TSEG1, TSEG2, SJW and BRP have to be programmed to a numerical value that is one less than its functional value; so instead of values in the range of [1…n], values in the range of [0…n–1] are programmed. That way, e.g., SJW (functional range of [1…4]) is represented by only two bits.

Therefore, the length of the bit time is (programmed values) [TSEG1 + TSEG2 + 3] t_q or (functional values) [Sync_Seg + Prop_Seg + Phase_Seg1 + Phase_Seg2] t_q.

The data in the bit timing register (DCAN_BTR) is the configuration input of the CAN protocol controller. The baud rate prescaler (configured by BRPE/BRP) defines the length of the time quantum (the basic time unit of the bit time); the bit timing logic (configured by TSEG1, TSEG2, and SJW) defines the number of time quanta in the bit time.

The processing of the bit time, the calculation of the position of the sample point, and occasional synchronizations are controlled by the bit timing state machine, which is evaluated once each time quantum. The rest of the CAN protocol controller, the bit stream processor (BSP) state machine, is evaluated once each bit time, at the sample point.

The shift register serializes the messages to be sent and parallelizes received messages. Its loading and shifting is controlled by the BSP. The BSP translates messages into frames and vice versa. It generates and discards the enclosing fixed format bits, inserts and extracts stuff bits, calculates and checks the CRC code, performs the error management, and decides which type of synchronization is to be used. It is evaluated at the sample point and processes the sampled bus input bit. The time after the sample point that is needed to calculate the next bit to be sent (e.g., data bit, CRC bit, stuff bit, error flag, or idle) is called the information processing time (IPT), which is 0 tq for the DCAN.

Generally, the IPT is CAN controller-specific, but may not be longer than 2 tq. The IPC length is the lower limit of the programmed length of Phase_Seg2. In case of a synchronization, Phase_Seg2 may be shortened to a value less than IPT, which does not affect bus timing.

18.6.4.9.2.1 Calculation of the Bit Timing Parameters

Usually, the calculation of the bit timing configuration starts with a desired bit rate or bit time. The resulting bit time (1 / Bit rate) must be an integer multiple of the CAN clock period.

The bit time may consist of 8 to 25 time quanta. The length of the time quantum t_q is defined by the baud rate prescaler with t_q = (Baud Rate Prescaler) / FCLK. Several combinations may lead to the desired bit time, allowing iterations of the following steps.

First part of the bit time to be defined is the Prop_Seg. Its length depends on the delay times measured in the system. A maximum bus length as well as a maximum node delay has to be defined for expandible CAN bus systems. The resulting time for Prop_Seg is converted into time quanta (rounded up to the nearest integer multiple of t_q).

The Sync_Seg is 1 t_q long (fixed), leaving (bit time – Prop_Seg – 1) t_q for the two Phase Buffer Segments. If the number of remaining t_q is even, the Phase Buffer Segments have the same length, Phase_Seg2 = Phase_Seg1, else Phase_Seg2 = Phase_Seg1 + 1.

The minimum nominal length of Phase_Seg2 has to be regarded as well. Phase_Seg2 may not be shorter than any CAN controller's Information Processing Time in the network, which is device dependent and can be in the range of [0…2] t_q.

The length of the synchronization jump width is set to its maximum value, which is the minimum of four (4) and Phase_Seg1.

The oscillator tolerance range necessary for the resulting configuration is calculated by the formulas given in Table 18-481.

If more than one configurations are possible to reach a certain Bit rate, it is recommended to choose the configuration which allows the highest oscillator tolerance range.

CAN nodes with different clocks require different configurations to come to the same bit rate. The calculation of the propagation time in the CAN network, based on the nodes with the longest delay times, is done once for the whole network.

The CAN system's oscillator tolerance range is limited by the node with the lowest tolerance range.

The calculation may show that bus length or bit rate have to be decreased or that the oscillator frequencies' stability has to be increased in order to find a protocol compliant configuration of the CAN bit timing.

The resulting configuration is written into the bit timing register (DCAN_BTR):

[14:12] TSEG2 = Phase_Seg2 - 1

[11:8] TSEG1 = Phase_Seg1+ Prop_Seg - 1

[7:6] SJW = SynchronizationJumpWidth - 1

[5:0] BRP = Prescaler - 1

18.6.4.9.2.2 Example for Bit Timing Calculation

In this example, the frequency of FCLK is 20 MHz, BRP is 2, the bit rate is 500 KBit/s.

Table 18-481. Example For Bit Timing

Parameter	Formula	Value	t_q
bit time (500 KBit/s)	1/bit rate, consists of $t_{Sync\text{-}Seg} + t_{TSeg1} + t_{TSeg2}$	2000 ns	20
delay of bus driver		280 ns	-
delay of receiver circuit		29 ns	-
delay of bus line (16 m)	16 × 5.5 ns/m	88 ns	-
t_q	BRP/FCLK	100 ns	1
t_{Sync_Seg}	1 × t_q (fixed)	100 ns	1
t_{Prop_Seg}	INT (2×delays + 1) = 8 × t_q	800 ns	8
t_{Seg1}	$t_{Prop\_Seg} + t_{Phase\_Seg1}$	1400 ns	14

Table 18-481. Example For Bit Timing (continued)

Parameter	Formula	Value	t_q
t_{Seg2}	bit time - ($_{Sync\_Seg}$ + t_{Seg1})	500 ns	5
t_{SJWmax}	MIN (4 × t_q, t_{Phase_Seg1})	400 ns	4

In this example, the bit timing register DCAN_BTR is programmed to:

- BRP = 2 - 1 = 1
- BRPE = 0
- TSEG1 = 14 - 1 = 13 (0xC)
- TSEG2 = 5 - 1 = 4
- SJW = 4 - 1 = 3

18.6.4.10 Message Interface Register Sets

The interface register sets control the software read and write accesses to the message RAM. There are two interface registers sets for read/write access, IF1 and IF2 and one interface register set for read access only, IF3.

Due to the structure of the message RAM, it is not possible to change single bits or bytes of a message object. Instead, always a complete message object in the message RAM is accessed. Therefore the data transfer from the IF1/IF2 registers to the message RAM requires the message handler to perform a read-modify-write cycle: First those parts of the message object that are not to be changed are read from the message RAM into the interface register set, and after the update the whole content of the interface register set is written into the message object.

After the partial write of a message object, those parts of the interface register set that are not selected in DCAN_IF1CMD/DCAN_IF2CMD, will be set to the actual contents of the selected message object.

After the partial read of a message object, those parts of the interface register set that are not selected in DCAN_IF1CMD/DCAN_IF2CMD, will be left unchanged.

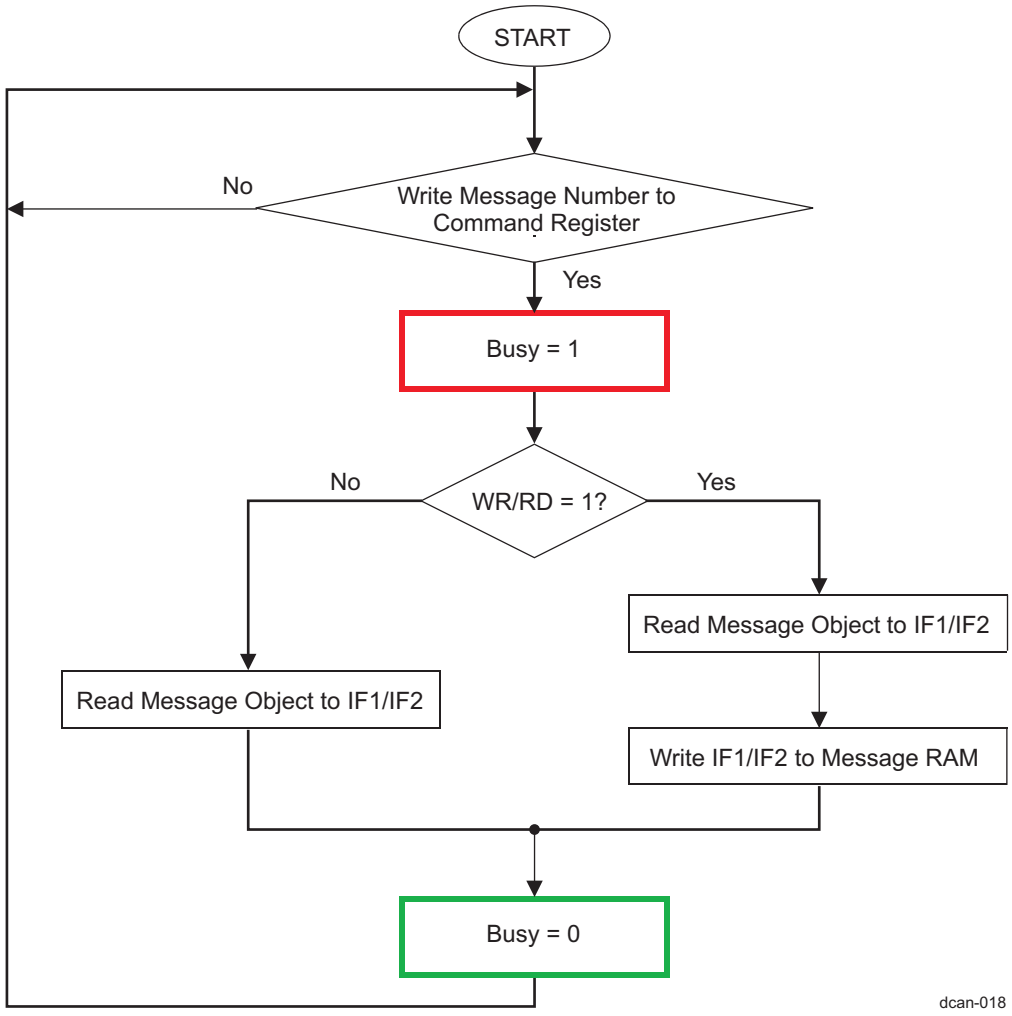
By buffering the data to be transferred, the interface register sets avoid conflicts between concurrent software accesses to the message RAM and CAN message reception and transmission. A complete message object (see Section 18.6.4.11.1, *Structure of Message Objects*) or parts of the message object may be transferred between the message RAM and the IF1/IF2 register set (see Section 18.6.5, *DCAN Register Manual*) in one single transfer. This transfer, performed in parallel on all selected parts of the message object, guarantees the data consistency of the CAN message.

18.6.4.10.1 Message Interface Register Sets 1 and 2

The IF1 and IF2 register sets control the data transfer to and from the message object. DCAN_IF1CMD/DCAN_IF2CMD address the desired message object in the message RAM and specifies whether a complete message object or only parts should be transferred. The data transfer is initiated by writing the message number to the bits [7:0] MESSAGE_NUMBER.

When the software initiates a data transfer between the IF1/IF2 registers and message RAM, the message handler sets the [15] BUSY bit in respective DCAN_IF1CMD/DCAN_IF2CMD to 1. After the transfer has completed, the BUSY bit is set back to 0 (see Figure 18-126).

Figure 18-126. Data Transfer Between IF1/IF2 Registers and Message RAM



18.6.4.10.2 IF3 Register Set

The IF3 register set can automatically be updated with received message objects without the need to initiate the transfer from message RAM by software. The intention of this feature of IF3 is to provide an interface for the DMA to read packets efficiently. The automatic update functionality can be programmed for each message object (DCAN_IF3UPD12 to DCAN_IF3UPD78).

All valid message objects in message RAM which are configured for automatic update, will be checked for active NewDat flags. If such a message object is found, it will be transferred to the IF3 register (if no previous DMA transfers are ongoing), controlled by IF3 Observation register (DCAN_IF3OBS). If more than one NewDat flag is active, the message object with the lowest number has the highest priority for automatic IF3 update.

The NewDat bit in the message object will be reset by a transfer to IF3.

If DCAN internal IF3 update is complete, a DMA request is generated. The DMA request stays active until first read access to one of the IF3 registers. The DMA functionality has to be enabled by setting bit DCAN_CTL[20] DE3 = 1.

NOTE: The IF3 register set cannot be used for transferring data into message objects.

18.6.4.11 Message RAM

The DCAN message RAM contains message objects. There are up to 64 message objects in the message RAM.

During normal operation, accesses to the message RAM are performed via the interface register sets, and the software cannot directly access the message RAM.

The interface register sets IF1 and IF2 provide indirect read/write access from the software to the message RAM. The IF1 and IF2 register sets can buffer control and user data to be transferred to and from the message objects.

The third interface register set IF3 can be configured to automatically receive control and user data from the message RAM when a message object has been updated after reception of a CAN message. The software does not need to initiate the transfer from message RAM to IF3 register set.

The message handler avoids potential conflicts between concurrent accesses to message RAM and CAN frame reception/transmission.

There are two modes where the message RAM can be directly accessed by the software:

- Debug/Suspend mode (see Section 18.6.4.11.4, *Message RAM Representation in Debug/Suspend Mode*)
- RAM Direct Access (RDA) mode (see Section 18.6.4.11.5, *Message RAM Representation in Direct Access Mode*)

CAUTION

Writes to the DCAN RAM must not be done while it is in low-power mode. If such writes occur, the behavior is undefined and RAM content is corrupted. It has to be ensured in software that the low-power mode is removed from the DCAN RAM before writing to it.

18.6.4.11.1 Structure of Message Objects

Table 18-482 shows the structure of a message object.

The highlighted fields are those parts of the message object which are represented in dedicated registers. For example, the transmit request flags of all message objects are represented in centralized transmit request registers.

Table 18-482. Structure of a Message Object

UMask	Msk[28:0]	MXtd	MDir	EoB	unused	NewDat	MsgLst	RxIE	TxIE	IntPnd	RmtEn	TxRqst
MsgVal	ID[28:0]	Xtd	Dir	DLC[3:0]	Data 0	Data 1	Data 2	Data 3	Data 4	Data 5	Data 6	Data 7

Table 18-483. Message Object Field Descriptions

Field Name	Value	Description
MsgVal		Message valid
	0	The message object is ignored by the message handler.
	1	The message object is to be used by the message handler.
		Note: This bit may be kept at level '1' even when the identifier bits ID[28:0], the control bits Xtd, Dir, or the data length code are changed. It should be reset if the Messages Object is no longer required. The software must reset the MsgVal bit of all unused Messages Objects during the initialization before it resets the INIT bit in the DCAN_CTL register.

Table 18-483. Message Object Field Descriptions (continued)

Field Name	Value	Description
UMask		Use acceptance mask
	0	Mask bits (Msk[28:0], MXtd and MDir) are ignored and not used for acceptance filtering.
	1	Mask bits are used for acceptance filtering.
		Note: If the UMask bit is set to one, the message object's mask bits have to be programmed during initialization of the message object before MsgVal is set to one.
ID[28:0]		Message identifier
	ID[28:0]	29-bit ("extended") identifier bits
	ID[28:18]	11-bit ("standard") identifier bits
Msk[28:0]		Identifier mask
	0	The corresponding bit in the message identifier is not used for acceptance filtering (don't care).
	1	The corresponding bit in the message identifier is used for acceptance filtering.
Xtd		Mask extended identifier
	0	The extended identifier bit (IDE) has no effect on the acceptance filtering.
	1	The extended identifier bit (IDE) is used for acceptance filtering.
		Note: When 11-bit ("standard") Identifiers are used for a message object, the identifiers of received data frames are written into bits ID[28:18]. For acceptance filtering, only these bits together with mask bits Msk[28:18] are considered.
Dir		Message direction
	0	Direction = receive: On TxRqst, a remote frame with the identifier of this message object is transmitted. On reception of a data frame with matching identifier, the message is stored in this message object.
	1	Direction = transmit: On TxRqst, a data frame is transmitted. On reception of a remote frame with matching identifier, the TxRqst bit of this message object is set (if RmtEn = one).
MDir		Mask message direction
	0	The message direction bit (Dir) has no effect on the acceptance filtering.
	1	The message direction bit (Dir) is used for acceptance filtering.
EOB		End of block
	0	The message object is part of a FIFO Buffer block and is not the last message object of this FIFO Buffer block.
	1	The message object is a single message object or the last message object in a FIFO Buffer Block.
		Note: This bit is used to concatenate multiple message objects to build a FIFO Buffer. For single message objects (not belonging to a FIFO Buffer), this bit must always be set to one.
NewDat		New data
	0	No new data has been written into the data bytes of this message object by the message handler since the last time when this flag was cleared by the software.
	1	The message handler or the software has written new data into the data bytes of this message object.
MsgLst		Message lost (only valid for message objects with direction = receive)
	0	No message was lost since the last time when this bit was reset by the software.
	1	The message handler stored a new message into this message object when NewDat was still set, so the previous message has been overwritten.
RxIE		Receive interrupt enable
	0	IntPnd will not be triggered after the successful reception of a frame.
	1	IntPnd will be triggered after the successful reception of a frame.
TxIE		Transmit interrupt enable
	0	IntPnd will not be triggered after the successful transmission of a frame.
	1	IntPnd will be triggered after the successful transmission of a frame.
IntPnd		Interrupt pending
	0	This message object is not the source of an interrupt.
	1	This message object is the source of an interrupt. The interrupt Identifier in the interrupt register (DCAN_INT) will point to this message object if there is no other interrupt source with higher priority.

Table 18-483. Message Object Field Descriptions (continued)

Field Name	Value	Description
RmtEn		Remote enable
	0	At the reception of a remote frame, TxRqst is not changed.
	1	At the reception of a remote frame, TxRqst is set.
TxRqst		Transmit request
	0	This message object is not waiting for a transmission.
	1	The transmission of this message object is requested and is not yet done.
DLC[3:0]		Data length code
	0 - 8	Data frame has 0 - 8 data bytes.
	9 - 15	Data frame has 8 data bytes.
		Note: The data length code of a message object must be defined to the same value as in the corresponding objects with the same identifier at other nodes. When the message handler stores a data frame, it will write the DLC to the value given by the received message.
Data 0		1st data byte of a CAN data frame
Data 1		2nd data byte of a CAN data frame
Data 2		3rd data byte of a CAN data frame
Data 3		4th data byte of a CAN data frame
Data 4		5th data byte of a CAN data frame
Data 5		6th data byte of a CAN data frame
Data 6		7th data byte of a CAN data frame
Data 7		8th data byte of a CAN data frame
		Note: Byte Data 0 is the first data byte shifted into the shift register of the CAN core during a reception, byte Data 7 is the last. When the message handler stores a data frame, it will write all the eight data bytes into a message object. If the data length code is less than 8, the remaining bytes of the message object may be overwritten by undefined values.

18.6.4.11.2 Addressing Message Objects in RAM

The starting location of a particular message object in RAM is:
Message RAM base address + (message object number) × 0x20

That is, message object 1 starts at offset 0x0020; message object 2 starts at offset 0x0040, etc.

NOTE: Because 0 is not a valid message object number, at offset 0x0000 is not located message object 0, but the last implemented: 64.

The base address for DCAN RAM is 0x4AE3 D000.

Message object number 1 has the highest priority.

Table 18-484. Message RAM Addressing in Debug/Suspend and RDA Modes

Message Object Number	Offset	Word Number	Debug/Suspend Mode, see Section 18.6.4.11.4	RDA Mode, see Section 18.6.4.11.5
64 (last implemented)	0x0000	1	Reserved	Data Bytes 4-7
	0x0004	2	MXtd,MDir,Mask	Data Bytes 0-3
	0x0008	3	Xtd,Dir,ID	ID[27:0],DLC
	0x000C	4	Ctrl	Mask,Xtd,Dir,ID[28]
	0x0010	5	Data Bytes 3-0	Reserved,Ctrl,MXtd,MDir
	0x0014	6	Data Bytes 7-4	-

Table 18-484. Message RAM Addressing in Debug/Suspend and RDA Modes (continued)

Message Object Number	Offset	Word Number	Debug/Suspend Mode, see Section 18.6.4.11.4	RDA Mode, see Section 18.6.4.11.5
1	0x0020	1	Reserved	Data Bytes 4-7
	0x0024	2	MXtd,MDir,Mask	Data Bytes 0-3
	0x0028	3	Xtd,Dir,ID	ID[27:0],DLC
	0x002C	4	Ctrl	Mask,Xtd,Dir,ID[28]
	0x0030	5	Data Bytes 3-0	Reserved,Ctrl,MXtd,MDir
	0x0034	6	Data Bytes 7-4	-
2	0x0040	1	Reserved	Data Bytes 4-7
	0x0044	2	MXtd,MDir,Mask	Data Bytes 0-3
	0x0048	3	Xtd,Dir,ID	ID[27:0],DLC
	0x004C	4	Ctrl	Mask,Xtd,Dir,ID[28]
	0x0050	5	Data Bytes 3-0	Reserved,Ctrl,MXtd,MDir
	0x0054	6	Data Bytes 7-4	-
…	…	…	…	…
63	0x07E0	1	Reserved	Data Bytes 4-7
	0x07E4	2	MXtd,MDir,Mask	Data Bytes 0-3
	0x07E8	3	Xtd,Dir,ID	ID[27:0],DLC
	0x07EC	4	Ctrl	Mask,Xtd,Dir,ID[28]
	0x07F0	5	Data Bytes 3-0	Reserved,Ctrl,MXtd,MDir
	0x07F4	6	Data Bytes 7-4	-

18.6.4.11.3 ECC RAM

On devices with SECDED implementation for the message RAM, the ECC bits are stored in a dedicated ECC RAM area which is memory mapped as follows:

The location of the ECC bits for a particular message object in RAM is:

Message RAM base address + 0x1000 + (message object number) × 0x20

NOTE: 0 is not a valid message object number. Therefore, at address 0x1000, the ECC bits of the last implemented message object are located, that is, 64-th.

Table 18-485. ECC RAM Representation

Address	Msg RAM base + 0x1000

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								ECC[8:0] last implemented Message Object (here: 64)								

Address	Msg RAM base + 0x1020

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								ECC[8:0] Message Object 1							

Address	Msg RAM base + 0x17E0

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								ECC[8:0] Message Object 63							

As shown in Figure 1-19 the ECC bits for the last implemented Message Object (here: 128) are located at offset 0x1000; the ECC bits for Message Object 1 are located at offset 0x1020, and the ECC bits for Message Object 127 are located at offset 0x1FE0.

The ECC RAM is only memory mapped if SECDED diagnostic mode is enabled.

18.6.4.11.4 Message RAM Representation in Debug/Suspend Mode

In debug/suspend mode, the message RAM will be memory mapped. This allows the external debug unit to access the message RAM.

NOTE: During debug/suspend mode, the message RAM cannot be accessed via the IFx register sets.

Table 18-486. Message RAM Representation in Debug/Suspend Mode

Offset Within MO	0x00

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																											Reserved for parity				

Offset Within MO	0x04

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MXtd	MDir	RESERVED	Msk[28:0]																												

Offset Within MO	0x08

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	Xtd	Dir	ID[28:0]																												

Offset Within MO	0x0C

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																MsgLst	RESERVED	UMask	TxIE	RxTE	RmtEn	RESERVED	EOB	RESERVED				DLC[3:0]			

Offset Within MO	0x10

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA_3								DATA_2								DATA_1								DATA_0							

Offset Within MO	0x14

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA_7								DATA_6								DATA_5								DATA_4							

18.6.4.11.5 Message RAM Representation in Direct Access Mode

When the DCAN_TEST[9] RDA bit is set while the DCAN module is in test mode (DCAN_CTL[7] TEST = 1), the software has direct access to the message RAM. Due to the 32-bit bus structure, the RAM is split into word lines to support this feature. The software has access to one word line at a time only.

In RAM direct access mode, the RAM is represented by a continuous memory space within the address frame of the DCAN module, starting at the message RAM base address.

Note: During direct access mode, the message RAM cannot be accessed via the IFx register sets.

Any read or write to the RAM addresses for RAM Direct Access during normal operation mode (TEST bit or RDA bit not set) will be ignored.

Table 18-487. Message RAM Representation in RAM Direct Access Mode

Offset Within MO	0x00																											
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA_4								DATA_5								DATA_6								DATA_7							

Offset Within MO	0x04																											
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA_0								DATA_1								DATA_2								DATA_3							

Offset Within MO	0x08																											
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ID[27:0]																												DLC[3:0]			

Offset Within MO	0x0C																											
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Msk[28:0]																												Xtd	Dir	ID[28]	

Offset Within MO	0x10																											
31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																Reserved for parity				Unused				MsgLst	UMask	TxIE	RxTE	RmtEn	EOB	MXtd	MDir

NOTE: Writes to unused bits have no effect.

18.6.4.12 CAN Operation

After device reset, the DCAN_CTL[0] INIT is set and all CAN protocol functions are disabled. The CAN module must be initialized before operating it. Figure 18-127 illustrates the basic initialization flow for the CAN module.

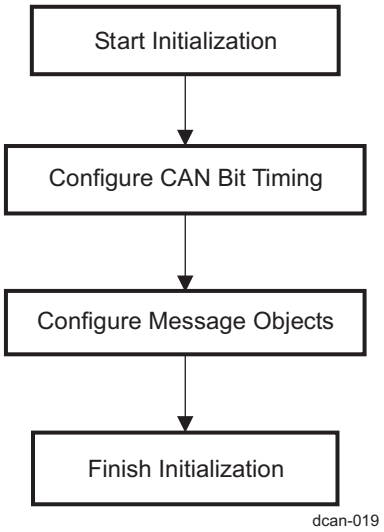
18.6.4.12.1 CAN Module Initialization

A general CAN module initialization would mean the following two critical steps:

- Configuration of the CAN bit timing
- Configuration of message objects

To initialize the CAN controller, the software has to set up the CAN bit timing and those message objects that have to be used for CAN communication. Message objects that are not needed, can be deactivated.

Figure 18-127. CAN Module General Initialization Flow



18.6.4.12.1.1 Configuration of CAN Bit Timing

The CAN module must be in initialization mode to configure the CAN bit timing.

For CAN bit timing software configuration flow, see Figure 18-128, *CAN Bit-Timing Configuration*.

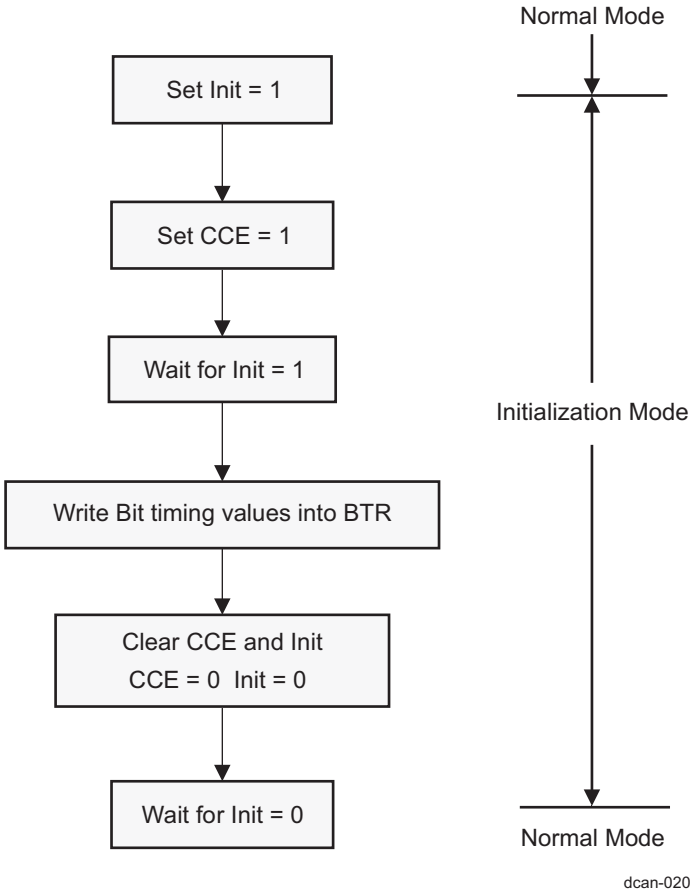
Step 1: Enter *initialization mode* (DCAN_CTL[0] INIT = 1).

While the INIT bit is set, the message transfer from and to the CAN bus is stopped, and the status of the CAN_TX output is recessive (high).

The CAN error counters are not updated. Setting the INIT bit does not change any other configuration register.

Also, note that the CAN module is in initialization mode on device reset and during Bus-Off.

Figure 18-128. CAN Bit-Timing Configuration



Step 2: Set the Configure Change Enable bit (DCAN_CTL[6] CCE = 1).

The access to the Bit Timing register (DCAN_BTR) for the configuration of the bit timing is enabled when both INIT = 1 and CCE = 1.

Step 3: Wait for the INIT bit to get set (=1). This would make sure that the module has entered Initialization mode.

Step 4: Write the bit timing values into DCAN_BTR. See Section 18.6.4.9.2, *DCAN Bit Timing Registers* for the values calculation for a given bit timing.

Step 5: Clear the CCE and INIT bits (=0).

Step 6: Wait for the INIT bit to clear (=0). This would ensure that the module has come out of Initialization mode.

Following these steps, the module comes to operation by synchronizing itself to the CAN bus, provided the DCAN_BTR is configured as per the CAN bus baud rate, although the message objects have to be configured before carrying out any communication.

NOTE: The module will not come out of the Initialization mode if any incorrect DCAN_BTR values are written in step 4.

NOTE: The required message objects should be configured as transmit or receive objects before the start of data transfer as explained in Section 18.6.4.12.1, *CAN Module Initialization*.

18.6.4.12.1.2 Configuration of Message Objects

The message objects can be configured only through the interface registers; the software does not have direct access to the message object (message RAM) . Familiarize yourself with the interface register set (IFx) usage (see Section 18.6.4.10, *Message Interface Register Sets*) and the message object structure (see Section 18.6.4.11, *Message RAM*) before configuring the message objects.

For more information regarding the procedure to configure the message objects, see Section 18.6.4.7, *Configuration of Message Objects Description*. All the message objects should be configured to particular identifiers or set to not valid before the message transfer is started. It is possible to change the configuration of message objects during normal operation (that is between data transfers).

NOTE: The message objects initialization is independent of the bit-timing configuration.

18.6.4.12.1.3 DCAN RAM Hardware Initialization

The memory hardware initialization for the DCAN module is enabled in the device control module. Setting RAMINIT_START to 1, causes RAM initialization with zeros and sets parity bits accordingly. Software must wait for the RAMINIT_DONE bit to be set to ensure successful RAM initialization.

For more details on CTRL_CORE_CONTROL_IO_2 register, see Section 13.5, *Control Module Register Manual*.

18.6.4.12.2 CAN Message Transfer (Normal Operation)

Once the DCAN is initialized and DCAN_CTL[0] INIT bit is reset (=0), the CAN core synchronizes itself to the CAN bus and is ready for message transfer as per the configured message objects.

The software may enable the interrupt lines (DCAN_CTL[1] IE0 and [17] IE1 = 1) at the same time when it clears [0] INIT and [6] CCE = 0. The status interrupts [3] EIE and [2] SIE may be enabled (=1) simultaneously.

The CAN communication can be carried out in any of the following two modes: interrupt and polling.

The DCAN_INT register points to those message objects with IntPnd = 1. It is updated even if the interrupt lines to the host processor are disabled (DCAN_CTL[1] IE0 and [17] IE1 = 0).

The software may poll all MessageObject's NewDat and TxRqst bits in parallel from the DCAN_NWDAT_X and the DCAN_TXRQ_X registers respectively. Polling can be made easier if all transmit objects are grouped at the low numbers and all receive objects are grouped at the high numbers.

Received messages are stored into their appropriate message objects if they pass acceptance filtering.

The whole message (including all arbitration bits, DLC and up to eight data bytes) is stored into the message object. As a consequence (e.g., when the identifier mask is used), the arbitration bits which are masked to "don't care" may change in the message object when a received message is stored.

The software may read or write each message at any time via the interface registers, as the message handler guarantees data consistency in case of concurrent accesses.

If a permanent message object (arbitration and control bits set up during configuration and leaving unchanged for multiple CAN transfers) exists for the message, it is possible to only update the data bytes.

If several transmit messages should be assigned to one message object, the whole message object has to be configured before the transmission of this message is requested.

The transmission of multiple message objects may be requested at the same time. They are subsequently transmitted, according to their internal priority.

Messages may be updated or set to not valid at any time, even if a requested transmission is still pending. However, the data bytes will be discarded if a message is updated before a pending transmission has started.

Depending on the configuration of the message object, a transmission may be automatically requested by the reception of a remote frame with a matching identifier.

18.6.4.12.2.1 Automatic Retransmission

According to the CAN Specification (ISO11898), the DCAN provides a mechanism to automatically retransmit frames which have lost arbitration or have been disturbed by errors during transmission. The frame transmission service will not be confirmed to the user before the transmission is successfully completed.

By default, this automatic retransmission is enabled. It can be disabled by setting bit disable automatic retransmission (DCAN_CTL[5] DAR = 1). Further details to this mode are provided in Section 18.6.4.8.3, *Transmission of Messages in Event Driven CAN Communication*.

18.6.4.12.2.2 Auto-Bus-On

By default, after the DCAN has entered Bus-Off state, the software can start a Bus-Off-Recovery sequence by resetting the DCAN_CTL[0] INIT bit to 0. If this is not done, the module will stay in Bus-Off state.

The DCAN provides an automatic Auto-Bus-On feature which is enabled by bit DCAN_CTL[9] ABO. If set, the DCAN will automatically start the Bus-Off-Recovery sequence. The sequence can be delayed by a user-defined number of interface clock cycles which can be defined in the Auto-Bus-On Time register (DCAN_ABOTR).

NOTE: If the DCAN goes to Bus-Off state due to a massive occurrence of CAN bus errors, it stops all bus activities and automatically sets the INIT bit. Once the INIT bit has been reset by the software or due to the Auto-Bus-On feature, the device will wait for 129 occurrences of bus Idle (equal to 129 × 11 consecutive recessive bits) before resuming normal operation. At the end of the Bus-Off recovery sequence, the error counters will be reset.

18.6.4.12.3 Test Modes

The DCAN module provides several test modes which are mainly intended for production tests or self test.

For all test modes, the DCAN_CTL[7] TEST bit needs to be set to 1. This enables write access to the test register (DCAN_TEST).

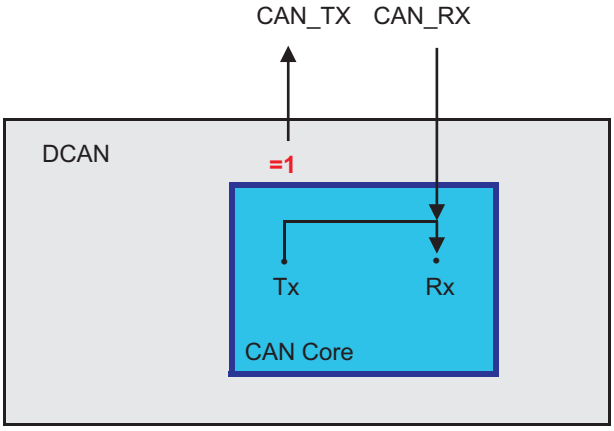
NOTE: It must be ensured by software that all message transfers are completed before entering test mode.

18.6.4.12.3.1 Silent Mode

The silent mode may be used to analyze the traffic on the CAN bus without affecting it by sending dominant bits (e.g., acknowledge bit, overload flag, active error flag). The DCAN is still able to receive valid data frames and valid remote frames, but it will not send any dominant bits. However, these are internally routed to the CAN core.

Figure 18-129 shows the connection of signals CAN_TX and CAN_RX to the CAN core in silent mode. Silent mode can be activated by setting the DCAN_TEST[3] SILENT bit to 1. In ISO 11898-1, the silent mode is called the bus monitoring mode.

Figure 18-129. CAN Core in Silent Mode



18.6.4.12.3.2 Loopback Mode

The loopback mode is mainly intended for hardware self-test functions. In this mode, the CAN core uses internal feedback from Tx output to Rx input. Transmitted messages are treated as received messages, and can be stored into message objects if they pass acceptance filtering. The actual value of the CAN_RX input pin is disregarded by the CAN core. Transmitted messages can still be monitored at the CAN_TX pin.

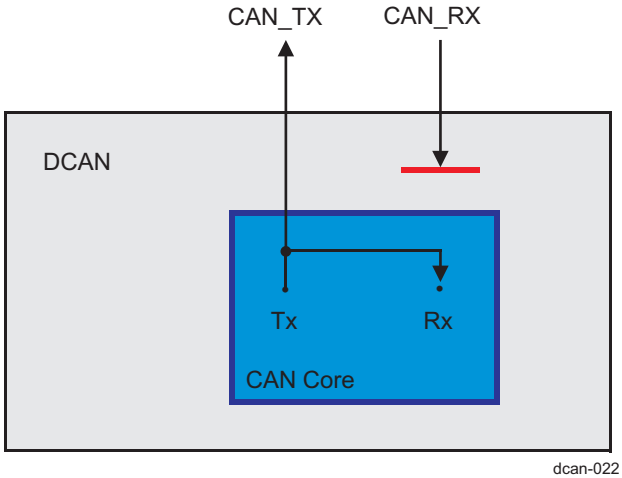
In order to be independent from external stimulation, the CAN core ignores acknowledge sampled in the acknowledge slot of a data/remote frame.

Figure 18-130 shows the connection of signals CAN_TX and CAN_RX to the CAN core in loopback mode.

Loopback mode can be activated by setting bit DCAN_TEST[4] LBACK to 1.

NOTE: In loopback mode, the signal path from CAN core to Tx pin, the Tx pin itself, and the signal path from Tx pin back to CAN core are disregarded. For including these into the testing, see Section 18.6.4.12.3.3, *External Loopback Mode*.

Figure 18-130. CAN Core in Loopback Mode



18.6.4.12.3.3 External Loopback Mode

The external loopback mode is similar to the loopback mode; however, it includes the signal path from CAN core to Tx pin, the Tx pin itself, and the signal path from Tx pin back to CAN core. When external loopback mode is selected, the input of the CAN core is connected to the input buffer of the Tx pin.

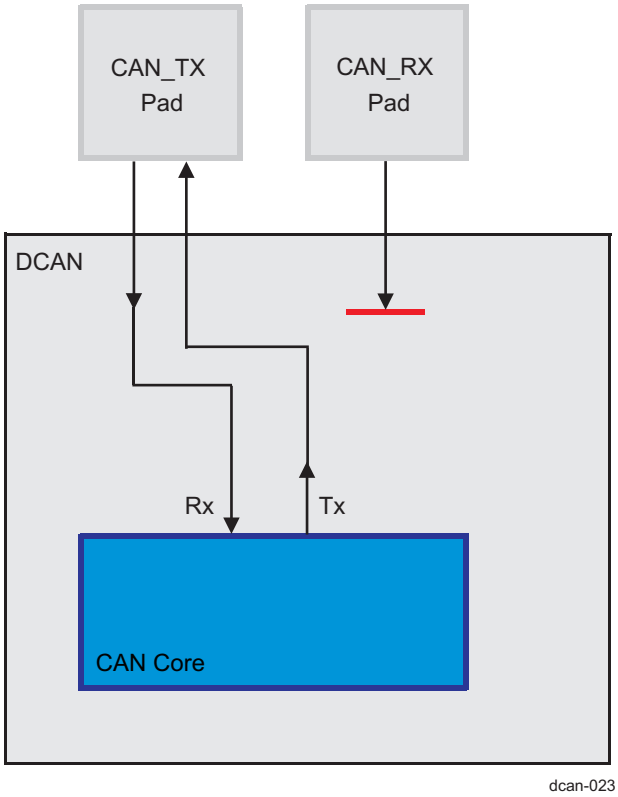
With this configuration, the Tx pin IO circuit can be tested.

External loopback mode can be activated by setting bit DCAN_TEST[8] EXL to 1.

Figure 18-131 shows the connection of signals CAN_TX and CAN_RX to the CAN core in external loopback mode.

NOTE: When loopback mode is active (LBACK bit set), the EXL bit will be ignored.

Figure 18-131. CAN Core in External Loopback Mode

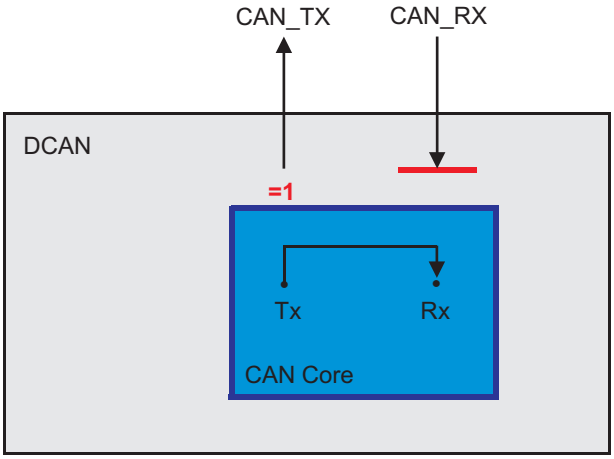


18.6.4.12.3.4 Loopback Mode Combined With Silent Mode

It is also possible to combine loopback mode and silent mode by setting bits DCAN_TEST[4] LBACK and [3] SILENT at the same time. This mode can be used for a "Hot Selftest", that is, the DCAN hardware can be tested without affecting the CAN network. In this mode, the CAN_RX pin is disconnected from the CAN core and no dominant bits will be sent on the CAN_TX pin.

Figure 18-132 shows the connection of the signals CAN_TX and CAN_RX to the CAN core in case of the combination of loopback mode with silent mode.

Figure 18-132. CAN Core in Loop Back Combined With Silent Mode



18.6.4.12.3.5 Software Control of CAN_TX Pin

Four output functions are available for the CAN transmit pin CAN_TX. In addition to its default function (serial data output), the CAN_TX pin can drive constant dominant or recessive values, or it can drive the CAN sample point signal to monitor the CAN core's bit timing.

Combined with the readable value of the CAN_RX pin, this function can be used to check the physical layer of the CAN bus.

The output mode of pin CAN_TX is selected by programming the DCAN_TEST[6:5] TX:

- 0x0: Normal operation, CAN_TX is controlled by the DCAN core
- 0x1: Sample point can be monitored at CAN_TX pin
- 0x2: CAN_TX pin drives a dominant value
- 0x3: CAN_TX pin drives a recessive value.

NOTE: The software control for the CAN_TX pin interferes with CAN protocol functions. For CAN message transfer or any of the test modes (loopback mode, external loopback mode or silent mode), the CAN_TX pin should operate in its default functionality.

18.6.4.13 GPIO Support

The CAN_RX and CAN_TX pins of the DCAN module can be used as general purpose IO pins, if CAN functionality is not needed. This function is controlled by the CAN TX IO control register (DCAN_TIOC) and the CAN RX IO control register (DCAN_RIOC).

18.6.5 DCAN Register Manual

18.6.5.1 DCAN Instance Summary

Table 18-488. DCAN Instance Summary

Module Name	Base Address	Size
DCAN	0x4AE3 C000	8 KiB

18.6.5.2 DCAN Registers

NOTE: After device reset, the registers of the DCAN hold the values shown in the register descriptions.

Additionally, the Bus-Off state is reset and the CAN_TX pin is set to recessive (HIGH). The INIT bit in the DCAN_CTL is set to enable the software initialization. The DCAN will not influence the CAN bus until the software resets INIT to 0.

18.6.5.2.1 DCAN Register Summary

Table 18-489. DCAN Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	DCAN Physical Address
DCAN_CTL	RW	32	0x0000 0000	0x4AE3 C000
DCAN_ES	R	32	0x0000 0004	0x4AE3 C004
DCAN_ERRC	R	32	0x0000 0008	0x4AE3 C008
DCAN_BTR	RW	32	0x0000 000C	0x4AE3 C00C
DCAN_INT	R	32	0x0000 0010	0x4AE3 C010
DCAN_TEST	RW	32	0x0000 0014	0x4AE3 C014
DCAN_PERR	R	32	0x0000 001C	0x4AE3 C01C
DCAN_REL	R	32	0x0000 0020	0x4AE3 C020
DCAN_ECCDIAG	RW	32	0x0000 0024	0x4AE3 C024
DCAN_ECCDIAG_STAT	RW	32	0x0000 0028	0x4AE3 C028
DCAN_ECC_CS	RW	32	0x0000 002C	0x4AE3 C02C
DCAN_ECC_SERR	R	32	0x0000 0030	0x4AE3 C030
DCAN_ABOTR	RW	32	0x0000 0080	0x4AE3 C080
DCAN_TXRQ_X	R	32	0x0000 0084	0x4AE3 C084
DCAN_TXRQ12	R	32	0x0000 0088	0x4AE3 C088
DCAN_TXRQ34	R	32	0x0000 008C	0x4AE3 C08C
DCAN_TXRQ56	R	32	0x0000 0090	0x4AE3 C090
DCAN_TXRQ78	R	32	0x0000 0094	0x4AE3 C094
DCAN_NWDAT_X	R	32	0x0000 0098	0x4AE3 C098
DCAN_NWDAT12	R	32	0x0000 009C	0x4AE3 C09C
DCAN_NWDAT34	R	32	0x0000 00A0	0x4AE3 C0A0
DCAN_NWDAT56	R	32	0x0000 00A4	0x4AE3 C0A4
DCAN_NWDAT78	R	32	0x0000 00A8	0x4AE3 C0A8
DCAN_INTPND_X	R	32	0x0000 00AC	0x4AE3 C0AC
DCAN_INTPND12	R	32	0x0000 00B0	0x4AE3 C0B0
DCAN_INTPND34	R	32	0x0000 00B4	0x4AE3 C0B4
DCAN_INTPND56	R	32	0x0000 00B8	0x4AE3 C0B8

Table 18-489. DCAN Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	DCAN Physical Address
DCAN_INTPND78	R	32	0x0000 00BC	0x4AE3 C0BC
DCAN_MSGVAL_X	R	32	0x0000 00C0	0x4AE3 C0C0
DCAN_MSGVAL12	R	32	0x0000 00C4	0x4AE3 C0C4
DCAN_MSGVAL34	R	32	0x0000 00C8	0x4AE3 C0C8
DCAN_MSGVAL56	R	32	0x0000 00CC	0x4AE3 C0CC
DCAN_MSGVAL78	R	32	0x0000 00D0	0x4AE3 C0D0
DCAN_INTMUX12	RW	32	0x0000 00D8	0x4AE3 C0D8
DCAN_INTMUX34	RW	32	0x0000 00DC	0x4AE3 C0DC
DCAN_INTMUX56	RW	32	0x0000 00E0	0x4AE3 C0E0
DCAN_INTMUX78	RW	32	0x0000 00E4	0x4AE3 C0E4
DCAN_IF1CMD	RW	32	0x0000 0100	0x4AE3 C100
DCAN_IF1MSK	RW	32	0x0000 0104	0x4AE3 C104
DCAN_IF1ARB	RW	32	0x0000 0108	0x4AE3 C108
DCAN_IF1MCTL	RW	32	0x0000 010C	0x4AE3 C10C
DCAN_IF1DATA	RW	32	0x0000 0110	0x4AE3 C110
DCAN_IF1DATB	RW	32	0x0000 0114	0x4AE3 C114
DCAN_IF2CMD	RW	32	0x0000 0120	0x4AE3 C120
DCAN_IF2MSK	RW	32	0x0000 0124	0x4AE3 C124
DCAN_IF2ARB	RW	32	0x0000 0128	0x4AE3 C128
DCAN_IF2MCTL	RW	32	0x0000 012C	0x4AE3 C12C
DCAN_IF2DATA	RW	32	0x0000 0130	0x4AE3 C130
DCAN_IF2DATB	RW	32	0x0000 0134	0x4AE3 C134
DCAN_IF3OBS	RW	32	0x0000 0140	0x4AE3 C140
DCAN_IF3MSK	RW	32	0x0000 0144	0x4AE3 C144
DCAN_IF3ARB	R	32	0x0000 0148	0x4AE3 C148
DCAN_IF3MCTL	R	32	0x0000 014C	0x4AE3 C14C
DCAN_IF3DATA	R	32	0x0000 0150	0x4AE3 C150
DCAN_IF3DATB	R	32	0x0000 0154	0x4AE3 C154
DCAN_IF3UPD12	RW	32	0x0000 0160	0x4AE3 C160
DCAN_IF3UPD34	RW	32	0x0000 0164	0x4AE3 C164
DCAN_IF3UPD56	RW	32	0x0000 0168	0x4AE3 C168
DCAN_IF3UPD78	RW	32	0x0000 016C	0x4AE3 C16C
DCAN_TIOC	RW	32	0x0000 01E0	0x4AE3 C1E0
DCAN_RIOC	RW	32	0x0000 01E4	0x4AE3 C1E4

18.6.5.2.2 DCAN Register Description

Table 18-490. DCAN_CTL

Address Offset	0x0000 0000
Physical Address	0x4AE3 C000 **Instance** DCAN
Description	DCAN control register **NOTE**: The Bus-Off recovery sequence (refer to CAN specification) cannot be shortened by setting or resetting INIT bit. If the module goes Bus-Off, it will automatically set the INIT bit and stop all bus activities. When the INIT bit is cleared by the application again, the module will then wait for 129 occurrences of Bus Idle (129 × 11 consecutive recessive bits) before resuming normal operation. At the end of the bus-off recovery sequence, the error counters will be reset. After the INIT bit is reset, each time when a sequence of 11 recessive bits is monitored, a Bit0 error code is written to DCAN_ES, enabling the software to check whether the CAN bus is stuck at dominant or continuously disturbed, and to monitor the proceeding of the bus-off recovery sequence.
Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED						WUBA	PDR	RESERVED			DE3	DE2	DE1	IE1	INITDBG	SWR	RESERVED	PMD				ABO	IDS	TEST	CCE	DAR	RESERVED	EIE	SIE	IE0	INIT

Bits	Field Name	Description	Type	Reset
31:26	RESERVED	These bits are always read as 0. Writes have no effect.	R	0x00
25	WUBA	Automatic wake up on bus activity when in local power-down mode. Note: The CAN message, which initiates the bus activity, cannot be received. This means that the first message received in power down and automatic wake-up mode, will be lost. 0: No detection of a dominant CAN bus level while in local power-down mode. 1: Detection of a dominant CAN bus level while in local power-down mode is enabled. On occurrence of a dominant CAN bus level, the wake up sequence is started (Additional information can be found in *Local Power-Down Mode*).	RW	0
24	PDR	Request for local low power-down mode 0: No application request for local low power-down mode. If the application has cleared this bit while DCAN in local power-down mode, also the INIT bit has to be cleared. 1: Local power-down mode has been requested by application. The DCAN will acknowledge the local power-down mode by setting bit PDA in the DCAN_ES register. The local clocks will be turned off by DCAN internal logic (Additional information can be found in *Local Power-Down Mode*).	RW	0
23:21	RESERVED	These bits are always read as 0. Writes have no effect.	R	0x0
20	DE3	Enable DMA request line for IF3. Note: A pending DMA request for IF3 remains active until first access to one of the IF3 registers. 0: Disabled 1: Enabled	RW	0
19	DE2	Enable DMA request line for IF2. Note: A pending DMA request for IF2 remains active until first access to one of the IF2 registers. 0: Disabled 1: Enabled	RW	0

Bits	Field Name	Description	Type	Reset
18	DE1	Enable DMA request line for IF1. Note: A pending DMA request for IF1 remains active until first access to one of the IF1 registers. 0: Disabled 1: Enabled	RW	0
17	IE1	Interrupt line 1 enable 0: Disabled - Module interrupt INT1 is always low. 1: Enabled - interrupts will assert line INT1 to one; line remains active until pending interrupts are processed.	RW	0
16	INITDBG	Internal init state while debug access 0: Not in debug mode, or debug mode requested but not entered. 1: Debug mode requested and internally entered; the DCAN is ready for debug accesses.	RW	0
15	SWR	Software reset enable. Note: To execute software reset, the following procedure is necessary: 1. Set INIT bit to shut down CAN communication. 2. Set SWR (this) bit additionally to INIT bit. 0: Normal Operation 1: Module is forced to reset state. This bit will automatically get cleared after execution of software reset after one OCP clock cycle.	RW	0
14	RESERVED	This bit is always read as 0. Writes have no effect.	R	0
13:10	PMD	ECCon/off 0x5: function disabled Others: function enabled	RW	0x5
9	ABO	Auto-Bus-On enable 0: The Auto-Bus-On feature is disabled 1: The Auto-Bus-On feature is enabled	RW	0
8	IDS	Interruption debug support enable 0: When Debug/Suspend mode is requested, DCAN will wait for a started transmission or reception to be completed before entering Debug/Suspend mode 1: When Debug/Suspend mode is requested, DCAN will interrupt any transmission or reception, and enter Debug/Suspend mode immediately.	RW	0
7	TEST	Test mode enable 0: Normal Operation 1: Test Mode	RW	0
6	CCE	Configuration change enable 0: The software has no write access to the configuration registers. 1: The software has write access to the configuration registers (when INIT bit is set).	RW	0
5	DAR	Disable automatic retransmission 0: Automatic retransmission of not successful messages enabled. 1: Automatic retransmission disabled.	RW	0
4	RESERVED	This bit is always read as 0. Writes have no effect.	R	0
3	EIE	Error interrupt enable 0: Disabled - PER, BOFF and EWARN bits can not generate an interrupt. 1: Enabled - PER, BOFF and EWARN bits can generate an interrupt at INT0 line and affect the interrupt register.	RW	0

Bits	Field Name	Description	Type	Reset
2	SIE	Status change interrupt enable 0: Disabled - WAKEUPPND, RXOK, TXOK and LEC bits can not generate an interrupt. 1: Enabled - WAKEUPPND, RXOK, TXOK and LEC can generate an interrupt at INT0 line and affect the interrupt register.	RW	0
1	IE0	Interrupt line 0 enable 0: Disabled - Module interrupt INT0 is always low. 1: Enabled - interrupts will assert line INT0 to one; line remains active until pending interrupts are processed.	RW	0
0	INIT	Initialization 0: Normal operation 1: Initialization mode is entered	RW	1

Table 18-491. Register Call Summary for Register DCAN_CTL

DCAN

- Interrupt Functionality: [0]
- Status Change Interrupts: [1][2]
- Error Interrupts: [3][4]
- DMA Functionality: [5]
- Entering Local Power-Down Mode: [6][7]
- Wakeup From Local Power Down: [8][9][10][11][12]
- SECDED Mechanism: [13]
- SECDED Testing: [14]
- Debug/Suspend Mode: [15][16]
- Transmission of Messages in Event Driven CAN Communication: [17]
- IF3 Register Set: [18]
- Structure of Message Objects: [19]
- Message RAM Representation in Direct Access Mode: [20]
- CAN Operation: [21]
- CAN Module Initialization: [22][23]
- CAN Message Transfer (Normal Operation): [24][25][26][27][28][29]
- Test Modes: [30]
- DCAN Registers: [31]
- DCAN Register Summary: [32]
- DCAN Register Description: [33][34][35][36][37][38][39][40][41][42][43][44][45][46][47][48][49][50][51]

Table 18-492. DCAN_ES

Address Offset	0x0000 0004		
Physical Address	0x4AE3 C004	**Instance**	DCAN
Description	Error and Status Register Interrupts are generated by bits PER, BOFF and EWARN (if EIE bit in DCAN_CTL is 1) and by bits WAKEUPPND, RXOK, TXOK, and LEC (if SIE bit in DCAN_CTL is 1). A change of bit EPASS will not generate an interrupt. Reading the DCAN_ES clears the WAKEUPPND, PER, RXOK and TXOK bits and set the LEC to value '7.' Additionally, the status interrupt value (0x8000) in the DCAN_INT will be replaced by the next lower priority interrupt value. For debug support, the auto clear functionality of DCAN_ES (clear of status flags by read) is disabled when in debug/suspend mode.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																					PDA	WAKEUPPND	PER	BOFF	EWARN	EPASS	RXOK	TXOK	LEC		

Bits	Field Name	Description	Type	Reset
31:11	RESERVED	These bits are always read as 0. Writes have no effect.	R	0x00 0000
10	PDA	Local power-down mode acknowledge 0: DCAN is not in local power-down mode. 1: Application request for setting DCAN to local power-down mode was successful. DCAN is in local power-down mode.	R	0
9	WAKEUPPND	Wake up pending. This bit can be used by the software to identify the DCAN as the source to wake up the system. This bit will be reset if DCAN_ES is read. 0: No Wake Up is requested by DCAN. 1: DCAN has initiated a wake up of the system due to dominant CAN bus while module power down.	R	0
8	PER	Parity error detected. This bit will be reset if DCAN_ES register is read. Read 0: No parity error has been detected since last read access. Read 1: The parity check mechanism has detected a parity error in the Message RAM. Write 0: No effect Write 1: End of interrupt (EOI) for parity error on DCAN_PARITY interrupt line	RW	0
7	BOFF	Bus-Off state 0: The CAN module is not bus-off state. 1: The CAN module is in bus-off state.	R	0
6	EWARN	Warning state 0: Both error counters are below the error warning limit of 96. 1: At least one of the error counters has reached the error warning limit of 96.	R	0
5	EPASS	Error passive state 0: On CAN Bus error, the DCAN could send active error frames. 1: The CAN core is in the error passive state as defined in the CAN Specification.	R	0

Bits	Field Name	Description	Type	Reset
4	RXOK	Received a message successfully. This bit will be reset if DCAN_ES register is read. 0: No message has been successfully received since the last time when this bit was read by the software. This bit is never reset by DCAN internal events. 1: A message has been successfully received since the last time when this bit was reset by a read access of the software (independent of the result of acceptance filtering).	R	0
3	TXOK	Transmitted a message successfully. This bit will be reset if DCAN_ES register is read. 0: No message has been successfully transmitted since the last time when this bit was read by the software. This bit is never reset by DCAN internal events. 1: A message has been successfully transmitted (error free and acknowledged by at least one other node) since the last time when this bit was reset by a read access of the software.	R	0
2:0	LEC	Last error code. The LEC field indicates the type of the last error on the CAN bus. This field will be cleared to '0' when a message has been transferred (reception or transmission) without error. 0x0: No error 0x1: Stuff error: More than five equal bits in a row have been detected in a part of a received message where this is not allowed. 0x2: Form error: A fixed format part of a received frame has the wrong format. 0x3: Ack error: The message this CAN core transmitted was not acknowledged by another node. 0x4: Bit1 error: During the transmission of a message (with the exception of the arbitration field), the device wanted to send a recessive level (bit of logical value '1'), but the monitored bus value was dominant. 0x5: Bit0 error: During the transmission of a message (or acknowledge bit, or active error flag, or overload flag), the device wanted to send a dominant level (logical value '0'), but the monitored bus level was recessive. During Bus-Off recovery, this status is set each time a sequence of 11 recessive bits has been monitored. This enables the software to monitor the proceeding of the Bus-Off recovery sequence (indicating the bus is not stuck at dominant or continuously disturbed). 0x6: CRC error: In a received message, the CRC check sum was incorrect. (CRC received for an incoming message does not match the calculated CRC for the received data). 0x7: No CAN bus event was detected since the last time the software read DCAN_ES. Any read access to DCAN_ES re-initializes the LEC to value '7.'	R	0x7

Table 18-493. Register Call Summary for Register DCAN_ES

DCAN

- Interrupt Functionality: [0][1]
- Status Change Interrupts: [2][3]
- Error Interrupts: [4]
- Entering Local Power-Down Mode: [5]
- Wakeup From Local Power Down: [6][7]
- Behavior on Double Bit Error: [8]
- Debug/Suspend Mode: [9]
- DCAN Register Summary: [10]
- DCAN Register Description: [11][12][13][14][15][16][17][18][19][20][21][22][23]

Table 18-494. DCAN_ERRC

Address Offset	0x0000 0008		
Physical Address	0x4AE3 C008	**Instance**	DCAN
Description	Error Counter Register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RP	REC							TEC							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	These bits are always read as 0. Writes have no effect.	R	0x0000
15	RP	Receive error passive 0: The receive error counter is below the error passive level. 1: The receive error counter has reached the error passive level as defined in the CAN specification.	R	0
14:8	REC	Receive error counter. Actual state of the receive error counter	R	0x00
7:0	TEC	Transmit error counter. Actual state of the transmit error counter	R	0x00

Table 18-495. Register Call Summary for Register DCAN_ERRC

DCAN

- DCAN Register Summary: [0]

Table 18-496. DCAN_BTR

Address Offset	0x0000 000C		
Physical Address	0x4AE3 C00C	**Instance**	DCAN
Description	Bit timing register This register is only writable if CCE and INIT bits in the DCAN_CTL are set. The CAN bit time may be programmed in the range of 8 to 25 time quanta The CAN time quantum may be programmed in the range of 1 to 1024 CAN_CLK periods.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												BRPE				RESERVED	TSEG2			TSEG1				SJW			BRP				

Bits	Field Name	Description	Type	Reset
31:20	RESERVED	These bits are always read as 0. Writes have no effect.	R	0x000
19:16	BRPE	Baud rate prescaler extension. Valid programmed values are 0 to 15. By programming BRPE the baud rate prescaler can be extended to values up to 1024.	RW	0x0
15	RESERVED	These bits are always read as 0. Writes have no effect.	R	0
14:12	TSEG2	Time segment after the sample point Valid programmed values are 0 to 7. The actual TSeg2 value which is interpreted for the bit timing will be the programmed TSeg2 value + 1.	RW	0x2
11:8	TSEG1	Time segment before the sample point Valid programmed values are 1 to15. The actual TSeg1 value interpreted for the bit timing will be the programmed TSeg1 value + 1.	RW	0x3
7:6	SJW	Synchronization Jump Width Valid programmed values are 0 to 3. The actual SJW value interpreted for the synchronization will be the programmed SJW value + 1.	RW	0x0
5:0	BRP	Baud rate prescaler Value by which the CAN_CLK frequency is divided for generating the bit time quanta. The bit time is built up from a multiple of this quanta. Valid programmed values are 0 to 63. The actual BRP value interpreted for the bit timing will be the programmed BRP value + 1.	RW	0x1

Table 18-497. Register Call Summary for Register DCAN_BTR

DCAN
- DCAN Bit Timing Registers: [0][1][2][3][4][5]
- CAN Module Initialization: [6][7][8][9]
- DCAN Register Summary: [10]

Table 18-498. DCAN_INT

Address Offset	0x0000 0010		
Physical Address	0x4AE3 C010	**Instance**	DCAN
Description	Interrupt register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								INT1ID								INT0ID															

Bits	Field Name	Description	Type	Reset
31:24	RESERVED	These bits are always read as 0. Writes have no effect.	R	0x00
23:16	INT1ID	Interrupt 1 Identifier (indicates the message object with the highest pending interrupt) 0x00: No interrupt is pending 0x01-0x80: Number of message object which caused the interrupt. 0x81-0xFF: Unused If several interrupts are pending, DCAN_INT will point to the pending interrupt with the highest priority. The INT1 interrupt line remains active until INT1ID reaches value 0 (the cause of the interrupt is reset) or until IE1 is cleared. A message interrupt is cleared by clearing the message object's IntPnd bit. Among the message interrupts, the message object's interrupt priority decreases with increasing message number.	R	0x00
15:0	INT0ID	Interrupt Identifier (the number here indicates the source of the interrupt) 0x0000: No interrupt is pending 0x0001-0x0080: Number of message object which caused the interrupt. 0x0081-0x7FFF: Unused 0x8000: DCAN_ES value is not 0x07. 0x8001-0xFFFF: Unused If several interrupts are pending,DCAN_INTwill point to the pending interrupt with the highest priority. The INT0 interrupt line remains active until INT0ID reaches value 0 (the cause of the interrupt is reset) or until IE0 is cleared. The Status interrupt has the highest priority. Among the message interrupts, the message object's interrupt priority decreases with increasing message number.	R	0x0000

Table 18-499. Register Call Summary for Register DCAN_INT

DCAN

- Interrupt Functionality: [0][1][2]
- Reception of Data Frames: [3]
- Structure of Message Objects: [4]
- CAN Message Transfer (Normal Operation): [5]
- DCAN Register Summary: [6]
- DCAN Register Description: [7][8][9][10][11][12][13][14][15]

Table 18-500. DCAN_TEST

Address Offset	0x0000 0014		
Physical Address	0x4AE3 C014	**Instance**	DCAN
Description	Test Register For all test modes, the TEST bit in DCAN_CTL control register needs to be set to 1. If TEST bit is set, the RDA, EXL, TX1, TX0, LBACK and SILENT bits are writable. Bit RX monitors the state of pin CAN_RX and therefore is only readable. All test register functions are disabled when TEST bit is cleared.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						RDA	EXL	RX	TX		LBACK	SILENT	RESERVED		

Bits	Field Name	Description	Type	Reset
31:10	RESERVED	These bits are always read as 0. Writes have no effect.	R	0x00 0000
9	RDA	RAM direct access enable 0: Normal operation 1: Direct access to the RAM is enabled while in test mode	RW	0
8	EXL	External loopback mode. When the internal loop-back mode is active (bit LBACK is set), bit EXL will be ignored. 0: Disabled 1: Enabled	RW	0
7	RX	Receive pin. Monitors the actual value of the CAN_RX pin 0: The CAN bus is dominant 1: The CAN bus is recessive	R	-
6:5	TX	Control of CAN_TX pin. Setting Tx[1:0] other than '00' will disturb message transfer. 0x0: Normal operation, CAN_TX is controlled by the CAN core. 0x1: Sample point can be monitored at CAN_TX pin. 0x2: CAN_TX pin drives a dominant value. 0x3: CAN_TX pin drives a recessive value.	RW	0x0
4	LBACK	Loopback mode. When the internal loop-back mode is active (bit LBACK is set), bit EXL will be ignored. 0: Disabled 1: Enabled	RW	0
3	SILENT	Silent mode 0: Disabled 1: Enabled	RW	0
2:0	RESERVED	These bits are always read as 0. Writes have no effect.	R	0x0

Table 18-501. Register Call Summary for Register DCAN_TEST

DCAN

- Message RAM Representation in Direct Access Mode: [0]
- Test Modes: [1][2][3][4][5][6]
- DCAN Register Summary: [7]

Table 18-502. DCAN_PERR

Address Offset	0x0000 001C		
Physical Address	0x4AE3 C01C	**Instance**	DCAN
Description	Parity Error Code Register If a parity error is detected, the PER flag will be set in the DCAN_ES. This bit is not reset by the parity check mechanism; it must be reset by reading DCAN_ES. In addition to the PER flag, the parity error code register will indicate the memory area where the parity error has been detected (message number and word number). If more than one word with a parity error was detected, the highest word number with a parity error will be displayed. After a parity error has been detected, the register will hold the last error code until power is removed.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																					WORD_NUMBER			MESSAGE_NUMBER							

Bits	Field Name	Description	Type	Reset
31:11	RESERVED	These bits are always read as 0. Writes have no effect.	R	0x00 0000
10:8	WORD_NUMBER	Word number where parity error has been detected RDA word number (1 to 5) of the message object (according to the message RAM representation in RDA mode).	R	0x-
7:0	MESSAGE_NUMBER	Message object number where parity error has been detected (0x01-0x80)	R	0x-

Table 18-503. Register Call Summary for Register DCAN_PERR

DCAN

- Behavior on Double Bit Error: [0][1]
- DCAN Register Summary: [2]

Table 18-504. DCAN_REL

Address Offset	0x0000 0020		
Physical Address	0x4AE3 C020	**Instance**	DCAN
Description	Core revision register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REVISION																															

Bits	Field Name	Description	Type	Reset
31:0	REVISION	DCAN core revision number	R	0x-

Table 18-505. Register Call Summary for Register DCAN_REL

DCAN

- DCAN Register Summary: [0]

Table 18-506. DCAN_ECCDIAG

Address Offset	0x0000 0024		
Physical Address	0x4AE3 C024	**Instance**	DCAN
Description	ECC Diagnostic Register. This register is writable only in privileged mode.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												ECCDIAG			

Bits	Field Name	Description	Type	Reset
31:4	RESERVED	Reserved	R	0x000 0000
3:0	ECCDIAG	SECDED diagnostic mode enable/disable 0x5: Diagnostic mode is enabled. Single and double bit errors are shown in the DCAN_ECCDIAG_STAT and the DCAN_ECC_CS. A double bit error (or single bit error with single bit error correction disabled) also triggers the parity interrupt flag (PER). Memory mapping of ECC RAM is enabled 0xA: Diagnostic mode is disabled, single and double bit errors are shown only in the DCAN_ECC_CS.	RW	0xA

Table 18-507. Register Call Summary for Register DCAN_ECCDIAG

DCAN

- SECDED Testing: [0][1]
- DCAN Register Summary: [2]

Table 18-508. DCAN_ECCDIAG_STAT

Address Offset	0x0000 0028		
Physical Address	0x4AE3 C028	**Instance**	DCAN
Description	ECC Diagnostic Status Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							DEFLG_DIAG	RESERVED							SEFLG_DIAG

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reserved	R	0x00 0000
8	DEFLG_DIAG	Double bit error flag diagnostic Read 0: No double bit error detected. Write 0: The bit is unchanged. Read 1: Double bit error detected in diagnostic mode. Write 1: The bit is cleared to 0.	RW	0
7:1	RESERVED	Reserved	R	0x00
0	SEFLG_DIAG	Single bit error flag diagnostic Read 0: No single bit error detected. Write 0: The bit is unchanged. Read 1: Single bit error detected in diagnostic mode. Write 1: The bit is cleared to 0.	RW	0

Table 18-509. Register Call Summary for Register DCAN_ECCDIAG_STAT

DCAN

- SECDED Testing: [0]
- DCAN Register Summary: [1]
- DCAN Register Description: [2]

Table 18-510. DCAN_ECC_CS

Address Offset	0x0000 002C		
Physical Address	0x4AE3 C02C	**Instance**	DCAN
Description	ECC Control and Status Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				SBE_EVT_EN				RESERVED				ECCMODE				RESERVED							DEFLG	RESERVED							SEFLG

Bits	Field Name	Description	Type	Reset
31:28	RESERVED	Reserved	R	0x0
27:24	SBE_EVT_EN	Enable/disable SECDED single bit error event (CAN_SERR signal). Write in privileged mode only. 0x5: SECDED single bit error event is disabled, single bit errors are not signaled with a high pulse on CAN_SERR signal. Others: SECDED single bit error event is enabled, single bit errors are signaled with a high pulse on CAN_SERR signal.	RW	0x5
23:20	RESERVED	Reserved	R	0x0
19:16	ECCMODE	Enable/disable SECDED single bit error correction. Write in privileged mode only. 0x5: SECDED single bit error correction disabled Others: SECDED single bit error correction enabled	RW	0xA
15:9	RESERVED	Reserved	R	0x00
8	DEFLG	Double bit error flag Read 0: No double bit error detected. Write 0: The bit is unchanged. Read 1: Double bit error detected. Write 1: The bit is cleared to 0.	RW	0
7:1	RESERVED	Reserved	R	0x00
0	SEFLG	Single bit error flag Read 0: No single bit error detected. Write 0: The bit is unchanged. Read 1: Single bit error detected. Write 1: The bit is cleared to 0.	RW	0

Table 18-511. Register Call Summary for Register DCAN_ECC_CS

DCAN

- SECDED Mechanism: [0]
- Behavior on Single Bit Error: [1][2]
- Behavior on Double Bit Error: [3]
- SECDED Testing: [4]
- DCAN Register Summary: [5]
- DCAN Register Description: [6][7][8]

Table 18-512. DCAN_ECC_SERR

Address Offset	0x0000 0030		
Physical Address	0x4AE3 C030	**Instance**	DCAN
Description	ECC Single Bit Error Code Register. If an ECC single bit error is detected, the SEFLG flag will be set in the DCAN_ECC_CS. In addition, MESSAGE_NUMBER will indicate the memory area where the single bit error has been detected (message object number only). If more than one word with an ECC single bit error was detected, the highest word number with an ECC single bit error error will be displayed. After an ECC single bit error has been detected, the register will hold the last error code until power is removed.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								MESSAGE_NUMBER							

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Reserved	R	0x00 0000
7:0	MESSAGE_NUMBER	Message object number where ECC single bit error has been detected. 0x0: Reserved	RW	0x-

Table 18-513. Register Call Summary for Register DCAN_ECC_SERR

DCAN

- Behavior on Single Bit Error: [0][1]
- DCAN Register Summary: [2]

Table 18-514. DCAN_ABOTR

Address Offset	0x0000 0080		
Physical Address	0x4AE3 C080	**Instance**	DCAN
Description	Auto-Bus-On Time Register On write access to the DCAN_CTL while Auto-Bus-On timer is running, the Auto-Bus-On procedure will be aborted. During Debug/Suspend mode, running Auto-Bus-On timer will be paused.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ABO_TIME																															

Bits	Field Name	Description	Type	Reset
31:0	ABO_TIME	Number of OCP clock cycles before a Bus-Off recovery sequence is started by clearing the INIT bit. This function has to be enabled by setting bit ABO in DCAN_CTL. The Auto-Bus-On timer is realized by a 32-bit counter which starts to count down to zero when the module goes Bus-Off. The counter will be reloaded with the preload value of the DCAN_ABOTR after this phase.	RW	0x0000 0000

Table 18-515. Register Call Summary for Register DCAN_ABOTR

DCAN

- CAN Message Transfer (Normal Operation): [0]
- DCAN Register Summary: [1]
- DCAN Register Description: [2]

Table 18-516. DCAN_TXRQ_X

Address Offset	0x0000 0084		
Physical Address	0x4AE3 C084	**Instance**	DCAN
Description	Transmission Request X Register The software can detect if one or more bits in the different transmission request registers are set. Each register bit represents a group of eight message objects. If at least one of the TxRqst bits of these message objects are set, the corresponding bit in the transmission request X register will be set.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																TXRQSTREG8		TXRQSTREG7		TXRQSTREG6		TXRQSTREG5		TXRQSTREG4		TXRQSTREG3		TXRQSTREG2		TXRQSTREG1	

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	RESERVED	R	0x0000
15:14	TXRQSTREG8	Transmission request bits (aggregate for 113-128 message objects). Lower bit represents first 8 message objects. Higher bit represents second 8 message objects.	R	0x0
13:12	TXRQSTREG7	Transmission request bits (aggregate for 97-112 message objects). Lower bit represents first 8 message objects. Higher bit represents second 8 message objects.	R	0x0
11:10	TXRQSTREG6	Transmission request bits (aggregate for 81-96 message objects). Lower bit represents first 8 message objects. Higher bit represents second 8 message objects.	R	0x0
9:8	TXRQSTREG5	Transmission request bits (aggregate for 65-80 message objects). Lower bit represents first 8 message objects. Higher bit represents second 8 message objects.	R	0x0
7:6	TXRQSTREG4	Transmission request bits (aggregate for 49-64 message objects). Lower bit represents first 8 message objects. Higher bit represents second 8 message objects.	R	0x0
5:4	TXRQSTREG3	Transmission request bits (aggregate for 33-48 message objects). Lower bit represents first 8 message objects. Higher bit represents second 8 message objects.	R	0x0
3:2	TXRQSTREG2	Transmission request bits (aggregate for 17-32 message objects). Lower bit represents first 8 message objects. Higher bit represents second 8 message objects.	R	0x0
1:0	TXRQSTREG1	Transmission request bits (aggregate for 1-16 message objects). Lower bit represents first 8 message objects. Higher bit represents second 8 message objects.	R	0x0

Table 18-517. Register Call Summary for Register DCAN_TXRQ_X

DCAN

- CAN Message Transfer (Normal Operation): [0]
- DCAN Register Summary: [1]

Table 18-518. DCAN_TXRQ12

Address Offset	0x0000 0088		
Physical Address	0x4AE3 C088	**Instance**	DCAN
Description	Transmission Request Register This register holds the TxRqst bits of the implemented message objects. By reading out these bits, the software can check for pending transmission requests. The TxRqst bit in a specific message object can be set/reset by the software via the IF1/IF2 message interface registers, or by the message handler after reception of a remote frame or after a successful transmission.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TXRQS																															

Bits	Field Name	Description	Type	Reset
31:0	TXRQS	Transmission request bits (for 1-32 message objects) 0: No transmission has been requested for this message object. 1: The transmission of this message object is requested and is not yet done.	R	0x0000 0000

Table 18-519. Register Call Summary for Register DCAN_TXRQ12

DCAN

- Transmission of Messages in Event Driven CAN Communication: [0]
- DCAN Register Summary: [1]

Table 18-520. DCAN_TXRQ34

Address Offset	0x0000 008C		
Physical Address	0x4AE3 C08C	**Instance**	DCAN
Description	Transmission Request Register This register holds the TxRqst bits of the implemented message objects. By reading out these bits, the software can check for pending transmission requests. The TxRqst bit in a specific message object can be set/reset by the software via the IF1/IF2 message interface registers, or by the message handler after reception of a remote frame or after a successful transmission.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TXRQS																															

Bits	Field Name	Description	Type	Reset
31:0	TXRQS	Transmission request bits (for 33-64 message objects) 0: No transmission has been requested for this message object. 1: The transmission of this message object is requested and is not yet done.	R	0x0000 0000

Table 18-521. Register Call Summary for Register DCAN_TXRQ34

DCAN

- DCAN Register Summary: [0]

Table 18-522. DCAN_TXRQ56

Address Offset	0x0000 0090		
Physical Address	0x4AE3 C090	**Instance**	DCAN
Description	Transmission Request Register This register holds the TxRqst bits of the implemented message objects. By reading out these bits, the software can check for pending transmission requests. The TxRqst bit in a specific message object can be set/reset by the software via the IF1/IF2 message interface registers, or by the message handler after reception of a remote frame or after a successful transmission.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TXRQS																															

Bits	Field Name	Description	Type	Reset
31:0	TXRQS	Transmission request bits (for 65-96 message objects) 0: No transmission has been requested for this message object. 1: The transmission of this message object is requested and is not yet done.	R	0x0000 0000

Table 18-523. Register Call Summary for Register DCAN_TXRQ56

DCAN

- DCAN Register Summary: [0]

Table 18-524. DCAN_TXRQ78

Address Offset	0x0000 0094		
Physical Address	0x4AE3 C094	**Instance**	DCAN
Description	Transmission Request Register This register holds the TxRqst bits of the implemented message objects. By reading out these bits, the software can check for pending transmission requests. The TxRqst bit in a specific message object can be set/reset by the software via the IF1/IF2 message interface registers, or by the message handler after reception of a remote frame or after a successful transmission.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TXRQS																															

Bits	Field Name	Description	Type	Reset
31:0	TXRQS	Transmission request bits (for 97-128 message objects) 0: No transmission has been requested for this message object. 1: The transmission of this message object is requested and is not yet done.	R	0x0000 0000

Table 18-525. Register Call Summary for Register DCAN_TXRQ78

DCAN

- Transmission of Messages in Event Driven CAN Communication: [0]
- DCAN Register Summary: [1]

Table 18-526. DCAN_NWDAT_X

Address Offset	0x0000 0098		
Physical Address	0x4AE3 C098	**Instance**	DCAN
Description	New Data X Register With the new data X register, the software can detect if one or more bits in the different new data registers are set. Each register bit represents a group of eight message objects. If at least on of the NewDat bits of these message objects are set, the corresponding bit in the new data X register will be set		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																NEWDATREG8		NEWDATREG7		NEWDATREG6		NEWDATREG5		NEWDATREG4		NEWDATREG3		NEWDATREG2		NEWDATREG1	

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Reserved	R	0x0000
15:14	NEWDATREG8	New data bits (aggregate for 113-128 message objects). Lower bit represents first 8 message objects. Higher bit represents second 8 message objects.	R	0x0
13:12	NEWDATREG7	New data bits (aggregate for 97-112 message objects). Lower bit represents first 8 message objects. Higher bit represents second 8 message objects.	R	0x0
11:10	NEWDATREG6	New data bits (aggregate for 81-96 message objects). Lower bit represents first 8 message objects. Higher bit represents second 8 message objects.	R	0x0
9:8	NEWDATREG5	New data bits (aggregate for 65-80 message objects). Lower bit represents first 8 message objects. Higher bit represents second 8 message objects.	R	0x0
7:6	NEWDATREG4	New data bits (aggregate for 49-64 message objects). Lower bit represents first 8 message objects. Higher bit represents second 8 message objects.	R	0x0
5:4	NEWDATREG3	New data bits (aggregate for 33-48 message objects). Lower bit represents first 8 message objects. Higher bit represents second 8 message objects.	R	0x0
3:2	NEWDATREG2	New data bits (aggregate for 17-32 message objects). Lower bit represents first 8 message objects. Higher bit represents second 8 message objects.	R	0x0
1:0	NEWDATREG1	New data bits (aggregate for 1-16 message objects). Lower bit represents first 8 message objects. Higher bit represents second 8 message objects.	R	0x0

Table 18-527. Register Call Summary for Register DCAN_NWDAT_X

DCAN

- CAN Message Transfer (Normal Operation): [0]
- DCAN Register Summary: [1]

Table 18-528. DCAN_NWDAT12

Address Offset	0x0000 009C		
Physical Address	0x4AE3 C09C	**Instance**	DCAN
Description	New Data Register This register hold the NewDat bits of the implemented message objects. By reading out these bits, the software can check for new data in the message objects. The NewDat bit of a specific message object can be set/reset by the software via the IF1/IF2 interface register sets, or by the message handler after reception of a data frame or after a successful transmission.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NEWDAT																															

Bits	Field Name	Description	Type	Reset
31:0	NEWDAT	New Data Bits (for 1-32 message objects) 0: No new data has been written into the data portion of this message object by the Message Handler since the last time when this flag was cleared by the software. 1: The Message Handler or the software has written new data into the data portion of this message object.	R	0x0000 0000

Table 18-529. Register Call Summary for Register DCAN_NWDAT12

DCAN

- DCAN Register Summary: [0]

Table 18-530. DCAN_NWDAT34

Address Offset	0x0000 00A0		
Physical Address	0x4AE3 C0A0	**Instance**	DCAN
Description	New Data Register This register hold the NewDat bits of the implemented message objects. By reading out these bits, the software can check for new data in the message objects. The NewDat bit of a specific message object can be set/reset by the software via the IF1/IF2 interface register sets, or by the message handler after reception of a data frame or after a successful transmission.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NEWDAT																															

Bits	Field Name	Description	Type	Reset
31:0	NEWDAT	New Data Bits (for 33-64 message objects) 0: No new data has been written into the data portion of this message object by the Message Handler since the last time when this flag was cleared by the software. 1: The Message Handler or the software has written new data into the data portion of this message object.	R	0x0000 0000

Table 18-531. Register Call Summary for Register DCAN_NWDAT34

DCAN

- DCAN Register Summary: [0]

Table 18-532. DCAN_NWDAT56

Address Offset	0x0000 00A4		
Physical Address	0x4AE3 C0A4	**Instance**	DCAN
Description	New Data Register This register hold the NewDat bits of the implemented message objects. By reading out these bits, the software can check for new data in the message objects. The NewDat bit of a specific message object can be set/reset by the software via the IF1/IF2 interface register sets, or by the message handler after reception of a data frame or after a successful transmission.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NEWDAT																															

Bits	Field Name	Description	Type	Reset
31:0	NEWDAT	New Data Bits (for 65-96 message objects) 0: No new data has been written into the data portion of this message object by the Message Handler since the last time when this flag was cleared by the software. 1: The Message Handler or the software has written new data into the data portion of this message object.	R	0x0000 0000

Table 18-533. Register Call Summary for Register DCAN_NWDAT56

DCAN
- DCAN Register Summary: [0]

Table 18-534. DCAN_NWDAT78

Address Offset	0x0000 00A8		
Physical Address	0x4AE3 C0A8	**Instance**	DCAN
Description	New Data Register This register hold the NewDat bits of the implemented message objects. By reading out these bits, the software can check for new data in the message objects. The NewDat bit of a specific message object can be set/reset by the software via the IF1/IF2 interface register sets, or by the message handler after reception of a data frame or after a successful transmission.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NEWDAT																															

Bits	Field Name	Description	Type	Reset
31:0	NEWDAT	New Data Bits (for 97-128 message objects) 0: No new data has been written into the data portion of this message object by the Message Handler since the last time when this flag was cleared by the software. 1: The Message Handler or the software has written new data into the data portion of this message object.	R	0x0000 0000

Table 18-535. Register Call Summary for Register DCAN_NWDAT78

DCAN
- DCAN Register Summary: [0]

Table 18-536. DCAN_INTPND_X

Address Offset	0x0000 00AC		
Physical Address	0x4AE3 C0AC	**Instance**	DCAN
Description	Interrupt Pending X Register With the interrupt pending X register, the software can detect if one or more bits in the different interrupt pending registers are set. Each bit of this register represents a group of eight message objects. If at least one of the IntPnd bits of these message objects are set, the corresponding bit in the interrupt pending X register will be set.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																INTPNDREG8		INTPNDREG7		INTPNDREG6		INTPNDREG5		INTPNDREG4		INTPNDREG3		INTPNDREG2		INTPNDREG1	

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Reserved	R	0x0000
15:14	INTPNDREG8	Interrupt Pending bits (aggregate for 113-128 message objects). Lower bit represents first 8 message objects. Higher bit represents second 8 message objects.	R	0x0
13:12	INTPNDREG7	Interrupt Pending bits (aggregate for 97-112 message objects). Lower bit represents first 8 message objects. Higher bit represents second 8 message objects.	R	0x0
11:10	INTPNDREG6	Interrupt Pendingbits (aggregate for 81-96 message objects). Lower bit represents first 8 message objects. Higher bit represents second 8 message objects.	R	0x0
9:8	INTPNDREG5	Interrupt Pending bits (aggregate for 65-80 message objects). Lower bit represents first 8 message objects. Higher bit represents second 8 message objects.	R	0x0
7:6	INTPNDREG4	Interrupt Pending bits (aggregate for 49-64 message objects). Lower bit represents first 8 message objects. Higher bit represents second 8 message objects.	R	0x0
5:4	INTPNDREG3	Interrupt Pending bits (aggregate for 33-48 message objects). Lower bit represents first 8 message objects. Higher bit represents second 8 message objects.	R	0x0
3:2	INTPNDREG2	Interrupt Pending bits (aggregate for 17-32 message objects). Lower bit represents first 8 message objects. Higher bit represents second 8 message objects.	R	0x0
1:0	INTPNDREG1	Interrupt Pending bits (aggregate for 1-16 message objects). Lower bit represents first 8 message objects. Higher bit represents second 8 message objects.	R	0x0

Table 18-537. Register Call Summary for Register DCAN_INTPND_X

DCAN

- DCAN Register Summary: [0]

Table 18-538. DCAN_INTPND12

Address Offset	0x0000 00B0		
Physical Address	0x4AE3 C0B0	**Instance**	DCAN
Description	Interrupt Pending Register This register holds the IntPnd bits of the implemented message objects. By reading out these bits, the software can check for pending interrupts in the message objects. The IntPnd bit of a specific message object can be set/reset by the software via the IF1/IF2 interface register sets, or by the message handler after a reception or a successful transmission.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTPND																															

Bits	Field Name	Description	Type	Reset
31:0	INTPND	Interrupt Pending Bits (for 1-32 message objects) 0: This message object is not the source of an interrupt. 1: This message object is the source of an interrupt.	R	0x0000 0000

Table 18-539. Register Call Summary for Register DCAN_INTPND12

DCAN
- DCAN Register Summary: [0]

Table 18-540. DCAN_INTPND34

Address Offset	0x0000 00B4		
Physical Address	0x4AE3 C0B4	**Instance**	DCAN
Description	Interrupt Pending Register This register holds the IntPnd bits of the implemented message objects. By reading out these bits, the software can check for pending interrupts in the message objects. The IntPnd bit of a specific message object can be set/reset by the software via the IF1/IF2 interface register sets, or by the message handler after a reception or a successful transmission.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTPND																															

Bits	Field Name	Description	Type	Reset
31:0	INTPND	Interrupt Pending Bits (for 33-64 message objects) 0: This message object is not the source of an interrupt. 1: This message object is the source of an interrupt.	R	0x0000 0000

Table 18-541. Register Call Summary for Register DCAN_INTPND34

DCAN
- DCAN Register Summary: [0]

Table 18-542. DCAN_INTPND56

Address Offset	0x0000 00B8		
Physical Address	0x4AE3 C0B8	**Instance**	DCAN
Description	Interrupt Pending Register This register holds the IntPnd bits of the implemented message objects. By reading out these bits, the software can check for pending interrupts in the message objects. The IntPnd bit of a specific message object can be set/reset by the software via the IF1/IF2 interface register sets, or by the message handler after a reception or a successful transmission.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTPND																															

Bits	Field Name	Description	Type	Reset
31:0	INTPND	Interrupt Pending Bits (for 65-96 message objects) 0: This message object is not the source of an interrupt. 1: This message object is the source of an interrupt.	R	0x0000 0000

Table 18-543. Register Call Summary for Register DCAN_INTPND56

DCAN

- DCAN Register Summary: [0]

Table 18-544. DCAN_INTPND78

Address Offset	0x0000 00BC		
Physical Address	0x4AE3 C0BC	**Instance**	DCAN
Description	Interrupt Pending Register This register holds the IntPnd bits of the implemented message objects. By reading out these bits, the software can check for pending interrupts in the message objects. The IntPnd bit of a specific message object can be set/reset by the software via the IF1/IF2 interface register sets, or by the message handler after a reception or a successful transmission.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTPND																															

Bits	Field Name	Description	Type	Reset
31:0	INTPND	Interrupt Pending Bits (for 97-128 message objects) 0: This message object is not the source of an interrupt. 1: This message object is the source of an interrupt.	R	0x0000 0000

Table 18-545. Register Call Summary for Register DCAN_INTPND78

DCAN

- DCAN Register Summary: [0]

Table 18-546. DCAN_MSGVAL_X

Address Offset	0x0000 00C0		
Physical Address	0x4AE3 C0C0	**Instance**	DCAN
Description	Message Valid X Register With the message valid X register, the software can detect if one or more bits in the different message valid registers are set. Each bit of this register represents a group of eight message objects. If at least one of the MsgVal bits of these message objects are set, the corresponding bit in the message valid X register will be set.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																MSGVALREG8		MSGVALREG7		MSGVALREG6		MSGVALREG5		MSGVALREG4		MSGVALREG3		MSGVALREG2		MSGVALREG1	

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Reserved	R	0x0000
15:14	MSGVALREG8	Message valid bits (aggregate for 113-128 message objects). Lower bit represents first 8 message objects. Higher bit represents second 8 message objects.	R	0x0
13:12	MSGVALREG7	Message valid bits (aggregate for 97-112 message objects). Lower bit represents first 8 message objects. Higher bit represents second 8 message objects.	R	0x0
11:10	MSGVALREG6	Message valid bits (aggregate for 81-96 message objects). Lower bit represents first 8 message objects. Higher bit represents second 8 message objects.	R	0x0
9:8	MSGVALREG5	Message valid bits (aggregate for 65-80 message objects). Lower bit represents first 8 message objects. Higher bit represents second 8 message objects.	R	0x0
7:6	MSGVALREG4	Message valid bits (aggregate for 49-64 message objects). Lower bit represents first 8 message objects. Higher bit represents second 8 message objects.	R	0x0
5:4	MSGVALREG3	Message valid bits (aggregate for 33-48 message objects). Lower bit represents first 8 message objects. Higher bit represents second 8 message objects.	R	0x0
3:2	MSGVALREG2	Message valid bits (aggregate for 17-32 message objects). Lower bit represents first 8 message objects. Higher bit represents second 8 message objects.	R	0x0
1:0	MSGVALREG1	Message valid bits (aggregate for 1-16 message objects). Lower bit represents first 8 message objects. Higher bit represents second 8 message objects.	R	0x0

Table 18-547. Register Call Summary for Register DCAN_MSGVAL_X

DCAN

- DCAN Register Summary: [0]

Table 18-548. DCAN_MSGVAL12

Address Offset	0x0000 00C4		
Physical Address	0x4AE3 C0C4	**Instance**	DCAN
Description	Message Valid Register These registers hold the MsgVal bits of the implemented message objects. By reading out these bits, the software can check which message objects are valid. The MsgVal bit of a specific message object can be set/reset by the software via the IF1/IF2 interface register sets, or by the message handler after a reception or a successful transmission		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MSGVAL																															

Bits	Field Name	Description	Type	Reset
31:0	MSGVAL	Message valid Bits (for 1-32 message objects) 0: This message object is ignored by the message handler. 1: This message object is configured and will be considered by the message handler.	R	0x0000 0000

Table 18-549. Register Call Summary for Register DCAN_MSGVAL12

DCAN

- Transmission of Messages in Event Driven CAN Communication: [0]
- DCAN Register Summary: [1]

Table 18-550. DCAN_MSGVAL34

Address Offset	0x0000 00C8		
Physical Address	0x4AE3 C0C8	**Instance**	DCAN
Description	Message Valid Register These registers hold the MsgVal bits of the implemented message objects. By reading out these bits, the software can check which message objects are valid. The MsgVal bit of a specific message object can be set/reset by the software via the IF1/IF2 interface register sets, or by the message handler after a reception or a successful transmission		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MSGVAL																															

Bits	Field Name	Description	Type	Reset
31:0	MSGVAL	Message valid Bits (for 33-64 message objects) 0: This message object is ignored by the message handler. 1: This message object is configured and will be considered by the message handler.	R	0x0000 0000

Table 18-551. Register Call Summary for Register DCAN_MSGVAL34

DCAN

- DCAN Register Summary: [0]

Table 18-552. DCAN_MSGVAL56

Address Offset	0x0000 00CC		
Physical Address	0x4AE3 C0CC	**Instance**	DCAN
Description	Message Valid Register These registers hold the MsgVal bits of the implemented message objects. By reading out these bits, the software can check which message objects are valid. The MsgVal bit of a specific message object can be set/reset by the software via the IF1/IF2 interface register sets, or by the message handler after a reception or a successful transmission		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MSGVAL																															

Bits	Field Name	Description	Type	Reset
31:0	MSGVAL	Message valid Bits (for 65-96 message objects) 0: This message object is ignored by the message handler. 1: This message object is configured and will be considered by the message handler.	R	0x0000 0000

Table 18-553. Register Call Summary for Register DCAN_MSGVAL56

DCAN

- DCAN Register Summary: [0]

Table 18-554. DCAN_MSGVAL78

Address Offset	0x0000 00D0		
Physical Address	0x4AE3 C0D0	**Instance**	DCAN
Description	Message Valid Register These registers hold the MsgVal bits of the implemented message objects. By reading out these bits, the software can check which message objects are valid. The MsgVal bit of a specific message object can be set/reset by the software via the IF1/IF2 interface register sets, or by the message handler after a reception or a successful transmission		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MSGVAL																															

Bits	Field Name	Description	Type	Reset
31:0	MSGVAL	Message valid Bits (for 97-128 message objects) 0: This message object is ignored by the message handler. 1: This message object is configured and will be considered by the message handler.	R	0x0000 0000

Table 18-555. Register Call Summary for Register DCAN_MSGVAL78

DCAN

- Transmission of Messages in Event Driven CAN Communication: [0]
- DCAN Register Summary: [1]

Table 18-556. DCAN_INTMUX12

Address Offset	0x0000 00D8		
Physical Address	0x4AE3 C0D8	**Instance**	DCAN
Description	Interrupt Multiplexer Register The IntMux flag determine for each message object, which of the two interrupt lines (INT0 or INT1) will be asserted when the IntPnd of this message object is set. Both interrupt lines can be globally enabled or disabled by setting or clearing IE0 and IE1 bits in DCAN_CTL. The IntPnd bit of a specific message object can be set or reset by the software via the IF1/IF2 interface register sets, or by message handler after reception or successful transmission of a frame. This will also affect the INT0ID resp. INT1ID flags in the DCAN_INT register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTMUX																															

Bits	Field Name	Description	Type	Reset
31:0	INTMUX	Multiplexes IntPnd value to either INT0 or INT1 interrupt lines (bit 0 -> last implemented message object) (bits 1:31 -> 1-31 message objects) 0: INT0 line is active if corresponding IntPnd flag is one. 1: INT1 line is active if corresponding IntPnd flag is one.	RW	0x0000 0000

Table 18-557. Register Call Summary for Register DCAN_INTMUX12

DCAN

- Message Object Interrupts: [0]
- DCAN Register Summary: [1]

Table 18-558. DCAN_INTMUX34

Address Offset	0x0000 00DC		
Physical Address	0x4AE3 C0DC	**Instance**	DCAN
Description	Interrupt Multiplexer Register The IntMux flag determine for each message object, which of the two interrupt lines (INT0 or INT1) will be asserted when the IntPnd of this message object is set. Both interrupt lines can be globally enabled or disabled by setting or clearing IE0 and IE1 bits in CAN control register. The IntPnd bit of a specific message object can be set or reset by the software via the IF1/IF2 interface register sets, or by message handler after reception or successful transmission of a frame. This will also affect the INT0ID resp INT1ID flags in the DCAN_INT register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTMUX																															

Bits	Field Name	Description	Type	Reset
31:0	INTMUX	Multiplexes IntPnd value to either INT0 or INT1 interrupt lines (bits 0:31 -> 32-63 message objects) 0: INT0 line is active if corresponding IntPnd flag is one. 1: INT1 line is active if corresponding IntPnd flag is one.	RW	0x0000 0000

Table 18-559. Register Call Summary for Register DCAN_INTMUX34

DCAN

- DCAN Register Summary: [0]

Table 18-560. DCAN_INTMUX56

Address Offset	0x0000 00E0		
Physical Address	0x4AE3 C0E0	**Instance**	DCAN
Description	Interrupt Multiplexer Register The IntMux flag determine for each message object, which of the two interrupt lines (INT0 or INT1) will be asserted when the IntPnd of this message object is set. Both interrupt lines can be globally enabled or disabled by setting or clearing IE0 and IE1 bits in CAN control register. The IntPnd bit of a specific message object can be set or reset by the software via the IF1/IF2 interface register sets, or by message handler after reception or successful transmission of a frame. This will also affect the INT0ID resp INT1ID flags in the DCAN_INT register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTMUX																															

Bits	Field Name	Description	Type	Reset
31:0	INTMUX	Multiplexes IntPnd value to either INT0 or INT1 interrupt lines (bits 0:31 -> 64-95 message objects) 0: INT0 line is active if corresponding IntPnd flag is one. 1: INT1 line is active if corresponding IntPnd flag is one.	RW	0x0000 0000

Table 18-561. Register Call Summary for Register DCAN_INTMUX56

DCAN
- DCAN Register Summary: [0]

Table 18-562. DCAN_INTMUX78

Address Offset	0x0000 00E4		
Physical Address	0x4AE3 C0E4	**Instance**	DCAN
Description	Interrupt Multiplexer Register The IntMux flag determine for each message object, which of the two interrupt lines (INT0 or INT1) will be asserted when the IntPnd of this message object is set. Both interrupt lines can be globally enabled or disabled by setting or clearing IE0 and IE1 bits in CAN control register. The IntPnd bit of a specific message object can be set or reset by the software via the IF1/IF2 interface register sets, or by message handler after reception or successful transmission of a frame. This will also affect the INT0ID resp INT1ID flags in the DCAN_INT register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTMUX																															

Bits	Field Name	Description	Type	Reset
31:0	INTMUX	Multiplexes IntPnd value to either INT0 or INT1 interrupt lines (bits 0:31 -> 96-127 message objects) 0: INT0 line is active if corresponding IntPnd flag is one. 1: INT1 line is active if corresponding IntPnd flag is one.	RW	0x0000 0000

Table 18-563. Register Call Summary for Register DCAN_INTMUX78

DCAN
- Message Object Interrupts: [0]
- DCAN Register Summary: [1]

Table 18-564. DCAN_IF1CMD

Address Offset	0x0000 0100
Physical Address	0x4AE3 C100 **Instance** DCAN
Description	IF1 Command Register The IF1 Command Register (DCAN_IF1CMD) configure and initiate the transfer between the IF1 register set and the message RAM. It is configurable which portions of the message object should be transferred. A transfer is started when the software writes the message number to bits [7:0] MESSAGE_NUMBER. With this write operation, the BUSY bit is automatically set to 1 to indicate that a transfer is in progress. After 4 to 14 OCP clock cycles, the transfer between the interface register and the message RAM will be completed and the BUSY bit is cleared. The maximum number of cycles is needed when the message transfer concurs with a CAN message transmission, acceptance filtering, or message storage. If the software writes to both DCAN_IF1CMD/DCAN_IF2CMD consecutively (request of a second transfer while first transfer is still in progress), the second transfer will start after the first one has been completed. While BUSY bit is one, IF1/IF2 register sets are write protected. For debug support, the auto clear functionality of the IF1/IF2 command registers (clear of DMAACTIVE flag by r/w) is disabled during Debug/Suspend mode. If an invalid Message Number is written to bits [7:0] MESSAGE_NUMBER, the message handler may access an implemented (valid) message object instead.
Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								WR_RD	MASK	ARB	CONTROL	CLRINTPND	TXRQST_NEWDAT	DATA_A	DATA_B	BUSY	DMAACTIVE	RESERVED						MESSAGE_NUMBER							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED	These bits are always read as 0. Writes have no effect.	R	0x00
23	WR_RD	Write/Read 0: Direction = Read: Transfer direction is from the message object addressed by MESSAGE_NUMBER to the IF1 register set. 1: Direction = Write: Transfer direction is from the IF1 register set to the message object addressed by MESSAGE_NUMBER.	RW	0
22	MASK	Access mask bits 0: Mask bits will not be changed 1: Direction = Read: The mask bits (identifier mask + MDir + MXtd) will be transferred from the message object addressed by MESSAGE_NUMBER to the IF1 register set. 1: Direction = Write: The mask bits (identifier mask + MDir + MXtd) will be transferred from the IF1 register set to the message object addressed by MESSAGE_NUMBER.	RW	0
21	ARB	Access arbitration bits 0: Arbitration bits will not be changed 1: Direction = Read: The Arbitration bits (Identifier + Dir + Xtd + MsgVal) will be transferred from the message object addressed by MESSAGE_NUMBER to the corresponding IF1 register set. 1: Direction = Write: The Arbitration bits (Identifier + Dir + Xtd + MsgVal) will be transferred from the IF1 register set to the message object addressed by MESSAGE_NUMBER.	RW	0

Bits	Field Name	Description	Type	Reset
20	CONTROL	Access control bits 0: Control bits will not be changed 1: Direction = Read: The message control bits will be transferred from the message object addressed by MESSAGE_NUMBER to the IF1 register set. 1: Direction = Write: The message control bits will be transferred from the IF1 registerset to the message object addressed by MESSAGE_NUMBER. If the TXRQST_NEWDAT bit in this register(Bit [18]) is set, the TXRQST/ NEWDAT bits in the DCAN_IF1MCTL will be ignored.	RW	0
19	CLRINTPND	Clear interrupt pending bit 0: IntPnd bit will not be changed 1: Direction = Read: Clears IntPnd bit in the message object. 1: Direction = Write: This bit is ignored. Copying of IntPnd flag from IF1 Registers to message RAM can only be controlled by the CONTROL flag (Bit [20]).	RW	0
18	TXRQST_NEWDAT	Access transmission request bit 0: Direction = Read: NewDat bit will not be changed. Direction = Write: TxRqst/NewDat bit will be handled according to the CONTROL bit. 1: Direction = Read: Clears NewDat bit in the message object. 1: Direction = Write: Sets TxRqst/NewDat in message object. Note: If a CAN transmission is requested by setting TXRQST_NEWDAT in this register, the TxRqst/NewDat bits in the message object will be set to one independent of the values in DCAN_IF1MCTL. Note: A read access to a message object can be combined with the reset of the control bits IntPnd and NewDat. The values of these bits transferred to the DCAN_IF1MCTL always reflect the status before resetting them.	RW	0
17	DATA_A	Access Data Bytes 0-3 0: Data Bytes 0-3 will not be changed. 1: Direction = Read: The data bytes 0-3 will be transferred from the message object addressed by the MESSAGE_NUMBER to the corresponding IF1 registerset. 1: Direction = Write: The data bytes 0-3 will be transferred from the IF1 registerset to the message object addressed by the MESSAGE_NUMBER. Note: The duration of the message transfer is independent of the number of bytes to be transferred.	RW	0
16	DATA_B	Access Data Bytes 4-7 0: Data Bytes 4-7 will not be changed. 1: Direction = Read: The data bytes 4-7 will be transferred from the message object addressed by MESSAGE_NUMBER to the corresponding IF1 registerset. 1: Direction = Write: The data bytes 4-7 will be transferred from the IF1 registerset to the message object addressed by MESSAGE_NUMBER. Note: The duration of the message transfer is independent of the number of bytes to be transferred.	RW	0

Bits	Field Name	Description	Type	Reset
15	BUSY	Busy flag 0: No transfer between IF1 register set and message RAM is in progress. 1: Transfer between IF1 register set and message RAM is in progress. This bit is set to one after the message number has been written to bits [7:0] MESSAGE_NUMBER. IF1 register set will be write protected. The bit is cleared after read/write action has been finished.	RW	0
14	DMAACTIVE	Activation of DMA feature for subsequent internal IF1 update 0: DMA request line is independent of IF1 activities. 1: DMA is requested after completed transfer between IF1 register set and message RAM. The DMA request remains active until the first read or write to one of the IF1 registers; an exception is a write to MESSAGE_NUMBER when DMAACTIVE is one. Note: Due to the auto reset feature of the DMAACTIVE bit, this bit has to be set for each subsequent DMA cycle separately.	RW	0
13:8	RESERVED	These bits are always read as 0. Writes have no effect.	R	0x00
7:0	MESSAGE_NUMBER	Number of message object in message RAM which is used for data transfer 0x00: Invalid message number 0x01-0x80: Valid message numbers 0x81-0xFF: Invalid message numbers	RW	0x1

Table 18-565. Register Call Summary for Register DCAN_IF1CMD

DCAN

- Interrupt Functionality: [0]
- DMA Functionality: [1]
- Debug/Suspend Mode: [2]
- Transmission of Messages in Event Driven CAN Communication: [3]
- Updating a Transmit Object: [4]
- Changing a Transmit Object: [5][6]
- Reading Received Messages: [7]
- Requesting New Data for a Receive Object: [8]
- Message Interface Register Sets: [9][10]
- Message Interface Register Sets 1 and 2: [11][12]
- DCAN Register Summary: [13]
- DCAN Register Description: [14][15][16][17][18][19][20][21][22][23][24][25][26]

Table 18-566. DCAN_IF1MSK

Address Offset	0x0000 0104		
Physical Address	0x4AE3 C104	**Instance**	DCAN
Description	IF1 Mask Register The bits of the IF1/IF2 mask registers mirror the mask bits of a message object. The function of the relevant message objects bits is described in Section 18.6.4.11.1 *Structure of Message Objects*. While BUSY bit of DCAN_IF1CMD/DCAN_IF2CMD register is one, IF1/IF2 register set is write protected.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MXTD	MDIR	RESERVED	MSK																												

Bits	Field Name	Description	Type	Reset
31	MXTD	Mask Extended Identifier 0: The extended identifier bit (IDE) has no effect on the acceptance filtering. 1: The extended identifier bit (IDE) is used for acceptance filtering. When 11-bit (“standard”) identifiers are used for a message object, the identifiers of received data frames are written into bits ID[28:18]. For acceptance filtering, only these bits together with mask bits Msk[28:18] are considered.	RW	1
30	MDIR	Mask Message Direction 0: The message direction bit (Dir) has no effect on the acceptance filtering. 1: The message direction bit (Dir) is used for acceptance filtering.	RW	1
29	RESERVED	This bit is always read as 1. Writes have no effect.	R	1
28:0	MSK	Identifier Mask 0: The corresponding bit in the identifier of the message object is not used for acceptance filtering (don't care). 1: The corresponding bit in the identifier of the message object is used for acceptance filtering.	RW	0x1FFF FFFF

Table 18-567. Register Call Summary for Register DCAN_IF1MSK

DCAN
- DCAN Register Summary: [0]

Table 18-568. DCAN_IF1ARB

Address Offset	0x0000 0108		
Physical Address	0x4AE3 C108	**Instance**	DCAN
Description	IF1 arbitration register The Arbitration bits ID[28:0], XTD, and DIR are used to define the identifier and type of outgoing messages and (together with the mask bits MSK[28:0], MXTD, and MDIR) for acceptance filtering of incoming messages. A received message is stored into the valid message object with matching identifier and Direction = receive (data frame) or Direction = transmit (remote frame). Extended frames can be stored only in message objects with XTD = 1, standard frames in message objects with XTD = 0. If a received message (data frame or remote frame) matches more than one valid message objects, it is stored into the one with the lowest message number. The bits of the IF1/IF2 arbitration registers mirror the arbitration bits of a message object. The function of the relevant message objects bits is described in Section 18.6.4.11.1 *Structure of Message Objects* While BUSY bit of DCAN_IF1CMD/DCAN_IF2CMD register is one, IF1/IF2 register set is write protected.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MSGVAL	XTD	DIR	ID																												

Bits	Field Name	Description	Type	Reset
31	MSGVAL	Message valid 0: The message object is ignored by the message handler. 1: The message object is to be used by the message handler. The software should reset the MsgVal bit of all unused Messages Objects during the initialization before it resets bit INIT in the DCAN_CTL. This bit must also be reset if the messages object is no longer required.	RW	0
30	XTD	Extended identifier 0: The 11-bit (“standard”) Identifier is used for this message object. 1: The 29-bit (“extended”) Identifier is used for this message object.	RW	0
29	DIR	Message direction 0: Direction = receive: On TxRqst, a remote frame with the identifier of this message object is transmitted. On reception of a data frame with matching identifier, this message is stored in this message object. 1: Direction = transmit: On TxRqst, the respective message object is transmitted as a data frame. On reception of a remote frame with matching identifier, the TxRqst bit of this message object is set (if RmtEn = 1).	RW	0
28:0	ID	Message identifier ID[28:0]: 29-bit identifier (extended frame) ID[28:18]: 11-bit identifier (standard frame)	RW	0x0000 0000

Table 18-569. Register Call Summary for Register DCAN_IF1ARB

DCAN

- Configuration of a Single Receive Object for Data Frames: [0]
- DCAN Register Summary: [1]

Table 18-570. DCAN_IF1MCTL

Address Offset	0x0000 010C
Physical Address	0x4AE3 C10C **Instance** DCAN
Description	IF1 Message Control Register The bits of the IF1/IF2 message control registers mirror the message control bits of a message object. The function of the relevant message objects bits is described in Section 18.6.4.11.1 *Structure of Message Objects* While BUSY bit of DCAN_IF1CMD/DCAN_IF2CMD register is one, IF1/IF2 register set is write protected.
Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																NEWDAT	MSGLST	INTPND	UMASK	TXIE	RXIE	RMTEN	TXRQST	EOB	RESERVED			DLC			

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	These bits are always read as 0. Writes have no effect.	R	0x0000
15	NEWDAT	New data 0: No new data has been written into the data portion of this message object by the message handler since the last time when this flag was cleared by the software. 1: The message handler or the software has written new data into the data portion of this message object.	RW	0
14	MSGLST	Message lost (only valid for message objects with direction = receive) 0: No message lost since the last time when this bit was reset by the software. 1: The message handler stored a new message into this object when NewDat was still set, so the previous message has been overwritten.	RW	0
13	INTPND	Interrupt pending 0: This message object is not the source of an interrupt. 1: This message object is the source of an interrupt. The Interrupt Identifier in DCAN_INT will point to this message object if there is no other interrupt source with higher priority.	RW	0
12	UMASK	Use acceptance mask 0: Mask ignored 1: Use mask (Msk[28:0], MXtd, and MDir) for acceptance filtering If the UMASK bit is set to one, the message object's mask bits have to be programmed during initialization of the message object before MsgVal is set to one.	RW	0
11	TXIE	Transmit interrupt enable 0: IntPnd will not be triggered after the successful transmission of a frame. 1: IntPnd will be triggered after the successful transmission of a frame.	RW	0
10	RXIE	Receive interrupt enable 0: IntPnd will not be triggered after the successful reception of a frame. 1: IntPnd will be triggered after the successful reception of a frame.	RW	0

Bits	Field Name	Description	Type	Reset
9	RMTEN	Remote enable 0: At the reception of a remote frame, TxRqst is not changed. 1: At the reception of a remote frame, TxRqst is set.	RW	0
8	TXRQST	Transmit request 0: This message object is not waiting for a transmission. 1: The transmission of this message object is requested and is not yet done.	RW	0
7	EOB	End of Block 0: The message object is part of a FIFO Buffer block and is not the last message object of the FIFO Buffer block. 1: The message object is a single message object or the last message object in a FIFO Buffer Block. Note: This bit is used to concatenate multiple message objects to build a FIFO Buffer. For single message objects (not belonging to a FIFO Buffer), this bit must always be set to 1.	RW	0
6:4	RESERVED	These bits are always read as 0. Writes have no effect.	R	0x0
3:0	DLC	Data length code 0-8: Data frame has 0-8 data bytes. 9-15 Data frame has 8 data bytes. Note: The data length code of a message object must be defined the same as in all the corresponding objects with the same identifier at other nodes. When the message handler stores a data frame, it will write the DLC to the value given by the received message.	RW	0x0

Table 18-571. Register Call Summary for Register DCAN_IF1MCTL

DCAN

- Reading Received Messages: [0]
- DCAN Register Summary: [1]
- DCAN Register Description: [2][3][4][5][6][7]

Table 18-572. DCAN_IF1DATA

Address Offset	0x0000 0110		
Physical Address	0x4AE3 C110	**Instance**	DCAN
Description	IF1 Data A Register The data bytes of CAN messages are stored in the IF1/IF2 registers in the following order: In a CAN data frame, Data 0 is the first, and Data 7 is the last byte to be transmitted or received. In CAN's serial bit stream, the MSB of each byte will be transmitted first. While BUSY bit of DCAN_IF1CMD/DCAN_IF2CMD register is one, IF1/IF2 register set is write protected.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA_3								DATA_2								DATA_1								DATA_0							

Bits	Field Name	Description	Type	Reset
31:24	DATA_3	Data byte 3	RW	0x0
23:16	DATA_2	Data byte 2	RW	0x0
15:8	DATA_1	Data byte 1	RW	0x0
7:0	DATA_0	Data byte 0	RW	0x0

Table 18-573. Register Call Summary for Register DCAN_IF1DATA

DCAN

- Updating a Transmit Object: [0]
- DCAN Register Summary: [1]

Table 18-574. DCAN_IF1DATB

Address Offset	0x0000 0114		
Physical Address	0x4AE3 C114	**Instance**	DCAN
Description	IF1 Data B Register The data bytes of CAN messages are stored in the IF1/IF2 registers in the following order: In a CAN data frame, Data 0 is the first, and Data 7 is the last byte to be transmitted or received. In CAN's serial bit stream, the MSB of each byte will be transmitted first. While BUSY bit of DCAN_IF1CMD/DCAN_IF2CMD register is one, IF1/IF2 register set is write protected.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA_7								DATA_6								DATA_5								DATA_4							

Bits	Field Name	Description	Type	Reset
31:24	DATA_7	Data byte 7	RW	0x0
23:16	DATA_6	Data byte 6	RW	0x0
15:8	DATA_5	Data byte 5	RW	0x0
7:0	DATA_4	Data byte 4	RW	0x0

Table 18-575. Register Call Summary for Register DCAN_IF1DATB

DCAN

- Updating a Transmit Object: [0]
- DCAN Register Summary: [1]

Table 18-576. DCAN_IF2CMD

Address Offset	0x0000 0120
Physical Address	0x4AE3 C120 **Instance** DCAN
Description	IF2 Command Register The IF2 Command Register (DCAN_IF2CMD) configure and initiate the transfer between the IF2 register set and the message RAM. It is configurable which portions of the message object should be transferred. A transfer is started when the software writes the message number to bits [7:0] MESSAGE_NUMBER. With this write operation, the BUSY bit is automatically set to 1 to indicate that a transfer is in progress. After 4 to 14 OCP clock cycles, the transfer between the interface register and the message RAM will be completed and the BUSY bit is cleared. The maximum number of cycles is needed when the message transfer concurs with a CAN message transmission, acceptance filtering, or message storage. If the software writes to both DCAN_IF1CMD/DCAN_IF2CMD consecutively (request of a second transfer while first transfer is still in progress), the second transfer will start after the first one has been completed. While BUSY bit is one, IF1/IF2 register sets are write protected. For debug support, the auto clear functionality of the IF1/IF2 command registers (clear of DMAACTIVE flag by r/w) is disabled during Debug/Suspend mode. If an invalid Message Number is written to bits [7:0] MESSAGE_NUMBER, the message handler may access an implemented (valid) message object instead.
Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								WR_RD	MASK	ARB	CONTROL	CLRINTPND	TXRQST_NEWDAT	DATA_A	DATA_B	BUSY	DMAACTIVE	RESERVED						MESSAGE_NUMBER							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED	These bits are always read as 0. Writes have no effect.	R	0x00
23	WR_RD	Write/Read 0: Direction = Read: Transfer direction is from the message object addressed by MESSAGE_NUMBER to the IF2 register set. 1: Direction = Write: Transfer direction is from the IF2 register set to the message object addressed by MESSAGE_NUMBER.	RW	0
22	MASK	Access mask bits 0: Mask bits will not be changed 1: Direction = Read: The mask bits (identifier mask + MDir + MXtd) will be transferred from the message object addressed by MESSAGE_NUMBER to the IF2 register set. 1: Direction = Write: The mask bits (identifier mask + MDir + MXtd) will be transferred from the IF2 register set to the message object addressed by MESSAGE_NUMBER.	RW	0
21	ARB	Access arbitration bits 0: Arbitration bits will not be changed 1: Direction = Read: The Arbitration bits (Identifier + Dir + Xtd + MsgVal) will be transferred from the message object addressed by MESSAGE_NUMBER to the corresponding IF2 register set. 1: Direction = Write: The Arbitration bits (Identifier + Dir + Xtd + MsgVal) will be transferred from the IF2 register set to the message object addressed by MESSAGE_NUMBER.	RW	0

Bits	Field Name	Description	Type	Reset
20	CONTROL	Access control bits 0: Control bits will not be changed 1: Direction = Read: The message control bits will be transferred from the message object addressed by MESSAGE_NUMBER to the IF2 register set. 1: Direction = Write: The message control bits will be transferred from the IF2 registerset to the message object addressed by MESSAGE_NUMBER. If the TXRQST_NEWDAT bit in this register(Bit [18]) is set, the TXRQST/ NEWDAT bits in the DCAN_IF1MCTL/DCAN_IF2MCTL will be ignored.	RW	0
19	CLRINTPND	Clear interrupt pending bit 0: IntPnd bit will not be changed 1: Direction = Read: Clears IntPnd bit in the message object. 1: Direction = Write: This bit is ignored. Copying of IntPnd flag from IF2 Registers to message RAM can only be controlled by the CONTROL flag (Bit [20]).	RW	0
18	TXRQST_NEWDAT	Access transmission request bit 0: Direction = Read: NewDat bit will not be changed. Direction = Write: TxRqst/NewDat bit will be handled according to the CONTROL bit. 1: Direction = Read: Clears NewDat bit in the message object. 1: Direction = Write: Sets TxRqst/NewDat in message object. Note: If a CAN transmission is requested by setting TXRQST_NEWDAT in this register, the TxRqst/NewDat bits in the message object will be set to one independent of the values in DCAN_IF1MCTL/DCAN_IF2MCTL. Note: A read access to a message object can be combined with the reset of the control bits IntPnd and NewDat. The values of these bits transferred to the DCAN_IF1MCTL/DCAN_IF2MCTL always reflect the status before resetting them.	RW	0
17	DATA_A	Access Data Bytes 0-3 0: Data Bytes 0-3 will not be changed. 1: Direction = Read: The data bytes 0-3 will be transferred from the message object addressed by the MESSAGE_NUMBER to the corresponding IF2 registerset. 1: Direction = Write: The data bytes 0-3 will be transferred from the IF2 registerset to the message object addressed by the MESSAGE_NUMBER. Note: The duration of the message transfer is independent of the number of bytes to be transferred.	RW	0
16	DATA_B	Access Data Bytes 4-7 0: Data Bytes 4-7 will not be changed. 1: Direction = Read: The data bytes 4-7 will be transferred from the message object addressed by MESSAGE_NUMBER to the corresponding IF2 registerset. 1: Direction = Write: The data bytes 4-7 will be transferred from the IF2 registerset to the message object addressed by MESSAGE_NUMBER. Note: The duration of the message transfer is independent of the number of bytes to be transferred.	RW	0

Bits	Field Name	Description	Type	Reset
15	BUSY	Busy flag 0: No transfer between IF2 register set and message RAM is in progress. 1: Transfer between IF2 register set and message RAM is in progress. This bit is set to one after the message number has been written to bits [7:0] MESSAGE_NUMBER. IF2 register set will be write protected. The bit is cleared after read/write action has been finished.	RW	0
14	DMAACTIVE	Activation of DMA feature for subsequent internal IF2 update 0: DMA request line is independent of IF2 activities. 1: DMA is requested after completed transfer between IF2 register set and message RAM. The DMA request remains active until the first read or write to one of the IF2 registers; an exception is a write to MESSAGE_NUMBER when DMAACTIVE is one. Note: Due to the auto reset feature of the DMAACTIVE bit, this bit has to be set for each subsequent DMA cycle separately.	RW	0
13:8	RESERVED	These bits are always read as 0. Writes have no effect.	R	0x00
7:0	MESSAGE_NUMBER	Number of message object in message RAM which is used for data transfer 0x00: Invalid message number 0x01-0x80: Valid message numbers 0x81-0xFF: Invalid message numbers	RW	0x1

Table 18-577. Register Call Summary for Register DCAN_IF2CMD

DCAN

- Interrupt Functionality: [0]
- DMA Functionality: [1]
- Debug/Suspend Mode: [2]
- Transmission of Messages in Event Driven CAN Communication: [3]
- Updating a Transmit Object: [4]
- Changing a Transmit Object: [5][6]
- Reading Received Messages: [7]
- Requesting New Data for a Receive Object: [8]
- Message Interface Register Sets: [9][10]
- Message Interface Register Sets 1 and 2: [11][12]
- DCAN Register Summary: [13]
- DCAN Register Description: [14][15][16][17][18][19][20][21][22][23][24][25][26]

Table 18-578. DCAN_IF2MSK

Address Offset	0x0000 0124		
Physical Address	0x4AE3 C124	**Instance**	DCAN
Description	IF2 Mask Register The bits of the IF1/IF2 mask registers mirror the mask bits of a message object. The function of the relevant message objects bits is described in Section 18.6.4.11.1 *Structure of Message Objects*. While BUSY bit of DCAN_IF1CMD/DCAN_IF2CMD register is one, IF1/IF2 register set is write protected.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MXTD	MDIR	RESERVED	MSK																												

Bits	Field Name	Description	Type	Reset
31	MXTD	Mask Extended Identifier 0: The extended identifier bit (IDE) has no effect on the acceptance filtering. 1: The extended identifier bit (IDE) is used for acceptance filtering. When 11-bit (“standard”) identifiers are used for a message object, the identifiers of received data frames are written into bits ID[28:18]. For acceptance filtering, only these bits together with mask bits Msk[28:18] are considered.	RW	1
30	MDIR	Mask Message Direction 0: The message direction bit (Dir) has no effect on the acceptance filtering. 1: The message direction bit (Dir) is used for acceptance filtering.	RW	1
29	RESERVED	This bit is always read as 1. Writes have no effect.	R	1
28:0	MSK	Identifier Mask 0: The corresponding bit in the identifier of the message object is not used for acceptance filtering (don't care). 1: The corresponding bit in the identifier of the message object is used for acceptance filtering.	RW	0x1FFF FFFF

Table 18-579. Register Call Summary for Register DCAN_IF2MSK

DCAN

- DCAN Register Summary: [0]

Table 18-580. DCAN_IF2ARB

Address Offset	0x0000 0128		
Physical Address	0x4AE3 C128	**Instance**	DCAN
Description	IF2 arbitration register The Arbitration bits ID[28:0], XTD, and DIR are used to define the identifier and type of outgoing messages and (together with the mask bits MSK[28:0], MXTD, and MDIR) for acceptance filtering of incoming messages. A received message is stored into the valid message object with matching identifier and Direction = receive (data frame) or Direction = transmit (remote frame). Extended frames can be stored only in message objects with Xtd = 1, standard frames in message objects with Xtd = 0. If a received message (data frame or remote frame) matches more than one valid message objects, it is stored into the one with the lowest message number. The bits of the IF1/IF2 arbitration registers mirror the arbitration bits of a message object. The function of the relevant message objects bits is described in Section 18.6.4.11.1 *Structure of Message Objects* While BUSY bit of DCAN_IF1CMD/DCAN_IF2CMD register is one, IF1/IF2 register set is write protected.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MSGVAL	XTD	DIR	ID																												

Bits	Field Name	Description	Type	Reset
31	MSGVAL	Message valid 0: The message object is ignored by the message handler. 1: The message object is to be used by the message handler. The software should reset the MsgVal bit of all unused Messages Objects during the initialization before it resets bit INIT in the DCAN_CTL. This bit must also be reset if the messages object is no longer required.	RW	0
30	XTD	Extended identifier 0: The 11-bit (“standard”) Identifier is used for this message object. 1: The 29-bit (“extended”) Identifier is used for this message object.	RW	0
29	DIR	Message direction 0: Direction = receive: On TxRqst, a remote frame with the identifier of this message object is transmitted. On reception of a data frame with matching identifier, this message is stored in this message object. 1: Direction = transmit: On TxRqst, the respective message object is transmitted as a data frame. On reception of a remote frame with matching identifier, the TxRqst bit of this message object is set (if RmtEn = 1).	RW	0
28:0	ID	Message identifier ID[28:0]: 29-bit identifier (extended frame) ID[28:18]: 11-bit identifier (standard frame)	RW	0x000 0000

Table 18-581. Register Call Summary for Register DCAN_IF2ARB

DCAN

- Configuration of a Single Receive Object for Data Frames: [0]
- DCAN Register Summary: [1]

Table 18-582. DCAN_IF2MCTL

Address Offset	0x0000 012C
Physical Address	0x4AE3 C12C **Instance** DCAN
Description	IF2 Message Control Register The bits of the IF1/IF2 message control registers mirror the message control bits of a message object. The function of the relevant message objects bits is described in Section 18.6.4.11.1 *Structure of Message Objects* While BUSY bit of DCAN_IF1CMD/DCAN_IF2CMD register is one, IF1/IF2 register set is write protected.
Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																NEWDAT	MSGLST	INTPND	UMASK	TXIE	RXIE	RMTEN	TXRQST	EOB	RESERVED			DLC			

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	These bits are always read as 0. Writes have no effect.	R	0x0000
15	NEWDAT	New data 0: No new data has been written into the data portion of this message object by the message handler since the last time when this flag was cleared by the software. 1: The message handler or the software has written new data into the data portion of this message object.	RW	0
14	MSGLST	Message lost (only valid for message objects with direction = receive) 0: No message lost since the last time when this bit was reset by the software. 1: The message handler stored a new message into this object when NewDat was still set, so the previous message has been overwritten.	RW	0
13	INTPND	Interrupt pending 0: This message object is not the source of an interrupt. 1: This message object is the source of an interrupt. The Interrupt Identifier in DCAN_INT will point to this message object if there is no other interrupt source with higher priority.	RW	0
12	UMASK	Use acceptance mask 0: Mask ignored 1: Use mask (Msk[28:0], MXtd, and MDir) for acceptance filtering If the UMask bit is set to one, the message object's mask bits have to be programmed during initialization of the message object before MsgVal is set to one.	RW	0
11	TXIE	Transmit interrupt enable 0: IntPnd will not be triggered after the successful transmission of a frame. 1: IntPnd will be triggered after the successful transmission of a frame.	RW	0
10	RXIE	Receive interrupt enable 0: IntPnd will not be triggered after the successful reception of a frame. 1: IntPnd will be triggered after the successful reception of a frame.	RW	0

Bits	Field Name	Description	Type	Reset
9	RMTEN	Remote enable 0: At the reception of a remote frame, TxRqst is not changed. 1: At the reception of a remote frame, TxRqst is set.	RW	0
8	TXRQST	Transmit request 0: This message object is not waiting for a transmission. 1: The transmission of this message object is requested and is not yet done.	RW	0
7	EOB	End of Block 0: The message object is part of a FIFO Buffer block and is not the last message object of the FIFO Buffer block. 1: The message object is a single message object or the last message object in a FIFO Buffer Block. Note: This bit is used to concatenate multiple message objects to build a FIFO Buffer. For single message objects (not belonging to a FIFO Buffer), this bit must always be set to one.	RW	0
6:4	RESERVED	These bits are always read as 0. Writes have no effect.	R	0x0
3:0	DLC	Data length code 0-8: Data frame has 0-8 data bytes. 9-15 Data frame has 8 data bytes. Note: The data length code of a message object must be defined the same as in all the corresponding objects with the same identifier at other nodes. When the message handler stores a data frame, it will write the DLC to the value given by the received message.	RW	0x0

Table 18-583. Register Call Summary for Register DCAN_IF2MCTL

DCAN

- Reading Received Messages: [0]
- DCAN Register Summary: [1]
- DCAN Register Description: [2][3][4]

Table 18-584. DCAN_IF2DATA

Address Offset	0x0000 0130		
Physical Address	0x4AE3 C130	**Instance**	DCAN
Description	IF2 Data A Register The data bytes of CAN messages are stored in the IF1/IF2 registers in the following order: In a CAN data frame, Data 0 is the first, and Data 7 is the last byte to be transmitted or received. In CAN's serial bit stream, the MSB of each byte will be transmitted first. While BUSY bit of DCAN_IF1CMD/DCAN_IF2CMD register is one, IF1/IF2 register set is write protected.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA_3								DATA_2								DATA_1								DATA_0							

Bits	Field Name	Description	Type	Reset
31:24	DATA_3	Data byte 3	RW	0x0
23:16	DATA_2	Data byte 2	RW	0x0
15:8	DATA_1	Data byte 1	RW	0x0
7:0	DATA_0	Data byte 0	RW	0x0

Table 18-585. Register Call Summary for Register DCAN_IF2DATA

DCAN

- Updating a Transmit Object: [0]
- DCAN Register Summary: [1]

Table 18-586. DCAN_IF2DATB

Address Offset	0x0000 0134		
Physical Address	0x4AE3 C134	**Instance**	DCAN
Description	IF2 Data B Register The data bytes of CAN messages are stored in the IF1/IF2 registers in the following order: In a CAN data frame, Data 0 is the first, and Data 7 is the last byte to be transmitted or received. In CAN's serial bit stream, the MSB of each byte will be transmitted first. While BUSY bit of DCAN_IF1CMD/DCAN_IF2CMD register is one, IF1/IF2 register set is write protected.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA_7								DATA_6								DATA_5								DATA_4							

Bits	Field Name	Description	Type	Reset
31:24	DATA_7	Data byte 7	RW	0x0
23:16	DATA_6	Data byte 6	RW	0x0
15:8	DATA_5	Data byte 5	RW	0x0
7:0	DATA_4	Data byte 4	RW	0x0

Table 18-587. Register Call Summary for Register DCAN_IF2DATB

DCAN

- Updating a Transmit Object: [0]
- DCAN Register Summary: [1]

Table 18-588. DCAN_IF3OBS

Address Offset	0x0000 0140		
Physical Address	0x4AE3 C140	**Instance**	DCAN
Description	IF3 Observation Register The IF3 register set can automatically be updated with received message objects without the need to initiate the transfer from message RAM by software (Additional information can be found in Section 18.6.4.11.1 *Structure of Message Objects*). The observation flags (Bits [4:0]) are used to determine, which data sections of the IF3 interface register set have to be read in order to complete a DMA read cycle. After all marked data sections are read, the DCAN is enabled to update the IF3 interface register set with new data. Any access order of single bytes or half-words is supported. When using byte or half-word accesses, a data section is marked as completed, if all bytes are read. NOTE: If IF3 Update Enable is used and no Observation flag is set, the corresponding message objects will be copied to IF3 without activating the DMA request line and without waiting for DMA read accesses. A write access to this register aborts a pending DMA cycle by resetting the DMA line and enables updating of IF3 interface register set with new data. To avoid data inconsistency, the DMA controller should be disabled before reconfiguring IF3 observation register. The status of the current read-cycle can be observed via status flags (Bits [12:8]).		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																IF3_UPD	RESERVED		IF3_SDB	IF3_SDA	IF3_SC	IF3_SA	IF3_SM	RESERVED			DATAB	DATAA	CTRL	ARB	MASK

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	These bits are always read as 0. Writes have no effect.	R	0x0000
15	IF3_UPD	IF3 Update Data 0: No new data has been loaded since last IF3 read. 1: New data has been loaded since last IF3 read.	R	0
14:13	RESERVED	These bits are always read as 0. Writes have no effect.	R	0x0
12	IF3_SDB	IF3 Status of Data B read access 0: All Data B bytes are already read out, or are not marked to be read. 1: Data B section has still data to be read out.	R	0
11	IF3_SDA	IF3 Status of Data A read access 0: All Data A bytes are already read out, or are not marked to be read. 1: Data A section has still data to be read out.	R	0
10	IF3_SC	IF3 Status of control bits read access 0: All control section bytes are already read out, or are not marked to be read. 1: Control section has still data to be read out.	R	0
9	IF3_SA	IF3 Status of Arbitration data read access 0: All Arbitration data bytes are already read out, or are not marked to be read. 1: Arbitration section has still data to be read out.	R	0
8	IF3_SM	IF3 Status of Mask data read access 0: All mask data bytes are already read out, or are not marked to be read. 1: Mask section has still data to be read out.	R	0
7:5	RESERVED	These bits are always read as 0. Writes have no effect.	R	0x0

Bits	Field Name	Description	Type	Reset
4	DATAB	Data B read observation 0: Data B section has not to be read. 1: Data B section has to be read to enable next IF3 update.	RW	0
3	DATAA	Data A read observation 0: Data A section has not to be read. 1: Data A section has to be read to enable next IF3 update.	RW	0
2	CTRL	Ctrl read observation 0: Ctrl section has not to be read. 1: Ctrl section has to be read to enable next IF3 update.	RW	0
1	ARB	Arbitration data read observation 0: Arbitration data has not to be read. 1: Arbitration data has to be read to enable next IF3 update.	RW	0
0	MASK	Mask data read observation 0: Mask data has not to be read. 1: Mask data has to be read to enable next IF3 update.	RW	0

Table 18-589. Register Call Summary for Register DCAN_IF3OBS

DCAN
- IF3 Register Set: [0]
- DCAN Register Summary: [1]

Table 18-590. DCAN_IF3MSK

Address Offset	0x0000 0144		
Physical Address	0x4AE3 C144	**Instance**	DCAN
Description	IF3 Mask Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MXTD	MDIR	RESERVED						MSK																							

Bits	Field Name	Description	Type	Reset
31	MXTD	Mask Extended Identifier 0: The extended identifier bit (IDE) has no effect on the acceptance filtering. 1: The extended identifier bit (IDE) is used for acceptance filtering. When 11-bit (“standard”) identifiers are used for a message object, the identifiers of received data frames are written into bits ID[28:18]. For acceptance filtering, only these bits together with mask bits Msk[28:18] are considered.	R	1
30	MDIR	Mask Message Direction 0: The message direction bit (Dir) has no effect on the acceptance filtering. 1: The message direction bit (Dir) is used for acceptance filtering.	R	1

Bits	Field Name	Description	Type	Reset
29	RESERVED	These bits are always read as 1. Writes have no effect.	R	1
28:0	MSK	Identifier Mask 0: The corresponding bit in the identifier of the message object is not used for acceptance filtering (don't care). 1: The corresponding bit in the identifier of the message object is used for acceptance filtering.	RW	0x1FFF FFFF

Table 18-591. Register Call Summary for Register DCAN_IF3MSK

DCAN
- DCAN Register Summary: [0]

Table 18-592. DCAN_IF3ARB

Address Offset	0x0000 0148		
Physical Address	0x4AE3 C148	**Instance**	DCAN
Description	IF3 Arbitration Register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MSGVAL	XTD	DIR	ID																												

Bits	Field Name	Description	Type	Reset
31	MSGVAL	Message Valid 0: The message object is ignored by the message handler. 1: The message object is to be used by the message handler. The software should reset the MsgVal bit of all unused Messages Objects during the initialization before it resets bit INIT in the DCAN_CTL. This bit must also be reset before the identifier ID[28:0], the control bits Xtd, Dir or DLC[3:0] are modified, or if the messages object is no longer required.	R	0
30	XTD	Extended Identifier 0: The 11-bit (“standard”) Identifier is used for this message object. 1: The 29-bit (“extended”) Identifier is used for this message object.	R	0
29	DIR	Message Direction 0: Direction = receive: On TxRqst, a remote frame with the identifier of this message object is transmitted. On reception of a data frame with matching identifier, this message is stored in this message object. 1: Direction = transmit: On TxRqst, the respective message object is transmitted as a data frame. On reception of a remote frame with matching identifier, the TxRqst bit of this message object is set (if RmtEn = 1).	R	0
28:0	ID	Message Identifier ID[28:0]: 29-bit Identifier (“extended frame”) ID[28:18]: 11-bit Identifier (“standard frame”)	R	0x0000 0000

Table 18-593. Register Call Summary for Register DCAN_IF3ARB

DCAN

- Configuration of a Single Receive Object for Data Frames: [0]
- DCAN Register Summary: [1]

Table 18-594. DCAN_IF3MCTL

Address Offset	0x0000 014C		
Physical Address	0x4AE3 C14C	**Instance**	DCAN
Description	IF3 Message Control Register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																NEWDAT	MSGLST	INTPND	UMASK	TXIE	RXIE	RMTEN	TXRQST	EOB	RESERVED			DLC			

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	These bits are always read as 0. Writes have no effect.	R	0x0000
15	NEWDAT	New Data 0: No new data has been written into the data portion of this message object by the message handler since the last time when this flag was cleared by the software. 1: The message handler or the software has written new data into the data portion of this message object.	R	0
14	MSGLST	Message Lost (only valid for message objects with direction = receive) 0: No message lost since the last time when this bit was reset by the software. 1: The message handler stored a new message into this object when NewDat was still set, so the previous message has been overwritten.	R	0
13	INTPND	Interrupt Pending 0: This message object is not the source of an interrupt. 1: This message object is the source of an interrupt. The Interrupt Identifier in DCAN_INT will point to this message object if there is no other interrupt source with higher priority.	R	0
12	UMASK	Use Acceptance Mask 0: Mask ignored 1: Use mask (Msk[28:0], MXtd, and MDir) for acceptance filtering If the UMASK bit is set to one, the message object's mask bits have to be programmed during initialization of the message object before MsgVal is set to one.	R	0
11	TXIE	Transmit Interrupt enable 0: IntPnd will not be triggered after the successful transmission of a frame. 1: IntPnd will be triggered after the successful transmission of a frame.	R	0
10	RXIE	Receive Interrupt enable 0: IntPnd will not be triggered after the successful reception of a frame. 1: IntPnd will be triggered after the successful reception of a frame.	R	0

Bits	Field Name	Description	Type	Reset
9	RMTEN	Remote enable 0: At the reception of a remote frame, TxRqst is not changed. 1: At the reception of a remote frame, TxRqst is set.	R	0
8	TXRQST	Transmit Request 0: This message object is not waiting for a transmission. 1: The transmission of this message object is requested and is not yet done.	R	0
7	EOB	End of Block 0: The message object is part of a FIFO Buffer block and is not the last message object of the FIFO Buffer block. 1: The message object is a single message object or the last message object in a FIFO Buffer Block. Note: This bit is used to concatenate multiple message objects to build a FIFO Buffer. For single message objects (not belonging to a FIFO Buffer), this bit must always be set to one.	R	0
6:4	RESERVED	These bits are always read as 0. Writes have no effect.	R	0x0
3:0	DLC	Data Length Code 0-8: Data frame has 0-8 data bits. 9-15: Data frame has 8 data bytes. Note: The data length code of a message object must be defined the same as in all the corresponding objects with the same identifier at other nodes. When the message handler stores a data frame, it will write the DLC to the value given by the received message.	R	0x0

Table 18-595. Register Call Summary for Register DCAN_IF3MCTL

DCAN
- Reading Received Messages: [0]
- DCAN Register Summary: [1]

Table 18-596. DCAN_IF3DATA

Address Offset	0x0000 0150		
Physical Address	0x4AE3 C150	**Instance**	DCAN
Description	IF3 Data A The data bytes of CAN messages are stored in the IF3 registers in the following order: In a CAN data frame, Data 0 is the first, and Data 7 is the last byte to be transmitted or received. In CAN's serial bit stream, the MSB of each byte will be transmitted first.		
Type	R		

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
DATA_3	DATA_2	DATA_1	DATA_0

Bits	Field Name	Description	Type	Reset
31: 24	DATA_3	Data byte 3	R	0x0
23:16	DATA_2	Data byte 2	R	0x0
15:8	DATA_1	Data byte 1	R	0x0
7:0	DATA_0	Data byte 0	R	0x0

Table 18-597. Register Call Summary for Register DCAN_IF3DATA

DCAN

- DCAN Register Summary: [0]

Table 18-598. DCAN_IF3DATB

Address Offset	0x0000 0154		
Physical Address	0x4AE3 C154	**Instance**	DCAN
Description	IF3 Data B The data bytes of CAN messages are stored in the IF3 registers in the following order: In a CAN data frame, Data 0 is the first, and Data 7 is the last byte to be transmitted or received. In CAN's serial bit stream, the MSB of each byte will be transmitted first.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA_7								DATA_6								DATA_5								DATA_4							

Bits	Field Name	Description	Type	Reset
31:24	DATA_7	Data byte 7	R	0x0
23:16	DATA_6	Data byte 6	R	0x0
15:8	DATA_5	Data byte 5	R	0x0
7:0	DATA_4	Data byte 4	R	0x0

Table 18-599. Register Call Summary for Register DCAN_IF3DATB

DCAN

- DCAN Register Summary: [0]

Table 18-600. DCAN_IF3UPD12

Address Offset	0x0000 0160		
Physical Address	0x4AE3 C160	**Instance**	DCAN
Description	Update Enable Register The automatic update functionality of the IF3 register set can be configured for each message object. A message object is enabled for automatic IF3 update, if the dedicated IF3UPDEN flag is set. This means that an active NewDat flag of this message object (e.g due to reception of a CAN frame) will trigger an automatic copy of the whole message object to IF3 register set NOTE: IF3 Update enable should not be set for transmit objects.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IF3UPDEN																															

Bits	Field Name	Description	Type	Reset
31:0	IF3UPDEN	IF3 Update Enabled (for 1-32 message objects) 0: Automatic IF3 update is disabled for this message object. 1: Automatic IF3 update is enabled for this message object. A message object is scheduled to be copied to IF3 register set, if NewDat flag of the message object is active.	RW	0x0000 0000

Table 18-601. Register Call Summary for Register DCAN_IF3UPD12

DCAN

- IF3 Register Set: [0]
- DCAN Register Summary: [1]

Table 18-602. DCAN_IF3UPD34

Address Offset	0x0000 0164		
Physical Address	0x4AE3 C164	**Instance**	DCAN
Description	Update Enable Register The automatic update functionality of the IF3 register set can be configured for each message object. A message object is enabled for automatic IF3 update, if the dedicated IF3UPDEN flag is set. This means that an active NewDat flag of this message object (e.g due to reception of a CAN frame) will trigger an automatic copy of the whole message object to IF3 register set NOTE: IF3 Update enable should not be set for transmit objects.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IF3UPDEN																															

Bits	Field Name	Description	Type	Reset
31:0	IF3UPDEN	IF3 Update Enabled (for 33-64 message objects) 0: Automatic IF3 update is disabled for this message object. 1: Automatic IF3 update is enabled for this message object. A message object is scheduled to be copied to IF3 register set, if NewDat flag of the message object is active.	RW	0x0000 0000

Table 18-603. Register Call Summary for Register DCAN_IF3UPD34

DCAN

- DCAN Register Summary: [0]

Table 18-604. DCAN_IF3UPD56

Address Offset	0x0000 0168		
Physical Address	0x4AE3 C168	**Instance**	DCAN
Description	Update Enable Register The automatic update functionality of the IF3 register set can be configured for each message object. A message object is enabled for automatic IF3 update, if the dedicated IF3UPDEN flag is set. This means that an active NewDat flag of this message object (e.g due to reception of a CAN frame) will trigger an automatic copy of the whole message object to IF3 register set NOTE: IF3 Update enable should not be set for transmit objects.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IF3UPDEN																															

Bits	Field Name	Description	Type	Reset
31:0	IF3UPDEN	IF3 Update Enabled (for 65-96 message objects) 0: Automatic IF3 update is disabled for this message object. 1: Automatic IF3 update is enabled for this message object. A message object is scheduled to be copied to IF3 register set, if NewDat flag of the message object is active.	RW	0x0000 0000

Table 18-605. Register Call Summary for Register DCAN_IF3UPD56

DCAN
- DCAN Register Summary: [0]

Table 18-606. DCAN_IF3UPD78

Address Offset	0x0000 016C		
Physical Address	0x4AE3 C16C	**Instance**	DCAN
Description	Update Enable Register The automatic update functionality of the IF3 register set can be configured for each message object. A message object is enabled for automatic IF3 update, if the dedicated IF3UPDEN flag is set. This means that an active NewDat flag of this message object (e.g due to reception of a CAN frame) will trigger an automatic copy of the whole message object to IF3 register set NOTE: IF3 Update enable should not be set for transmit objects.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IF3UPDEN																															

Bits	Field Name	Description	Type	Reset
31:0	IF3UPDEN	IF3 Update Enabled (for 97-128 message objects) 0: Automatic IF3 update is disabled for this message object. 1: Automatic IF3 update is enabled for this message object. A message object is scheduled to be copied to IF3 register set, if NewDat flag of the message object is active.	RW	0x0000 0000

Table 18-607. Register Call Summary for Register DCAN_IF3UPD78

DCAN
- IF3 Register Set: [0]
- DCAN Register Summary: [1]

Table 18-608. DCAN_TIOC

Address Offset	0x0000 01E0		
Physical Address	0x4AE3 C1E0	**Instance**	DCAN
Description	TX I/O Control Register The CAN_TX pin of the DCAN module can be used as general purpose IO pin if CAN function is not needed. The values of the IO control registers are only writable if INIT bit of the DCAN_CTL is set to 1.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED													PU	PD	OD	RESERVED												FUNC	DIR	OUT	IN

Bits	Field Name	Description	Type	Reset
31:19	RESERVED	These bits are always read as 0. Writes have no effect.	R	0x0000
18	PU	CAN_TX pull up/pull down select. This bit is only active when CAN_TX is configured to be an input. 0: CAN_TX pull down is selected, when pull logic is active ($\overline{PD}$ = 0). 1: CAN_TX pull up is selected, when pull logic is active($\overline{PD}$ = 0).	RW	0

Bits	Field Name	Description	Type	Reset
17	PD	CAN_TX pull disable. This bit is only active when CAN_TX is configured to be an input. 0: CAN_TX pull is active 1: CAN_TX pull is disabled	RW	0
16	OD	CAN_TX open drain enable. This bit is only active when CAN_TX is configured to be in GIO mode (FUNC=0). 0: The CAN_TX pin is configured in push/pull mode. 1: The CAN_TX pin is configured in open drain mode. Forced to '0' if INIT bit of DCAN_CTL is reset.	RW	0
15:4	RESERVED	These bits are always read as 0. Writes have no effect.	R	0x000
3	FUNC	CAN_TX function. This bit changes the function of the CAN_TX pin 0: CAN_TX pin is in GIO mode. 1: CAN_TX pin is in functional mode (as an output to transmit CAN data). Forced to Tx output of the CAN core, if INIT bit of DCAN_CTL is reset.	RW	0
2	DIR	CAN_TX data direction. This bit controls the direction of the CAN_TX pin when it is configured to be in GIO mode only (FUNC=0) 0: The CAN_TX pin is an input. 1: The CAN_TX pin is an output Forced to '1' if INIT bit of DCAN_CTL is reset.	RW	0
1	OUT	CAN_TX data out write. This bit is only active when CAN_TX pin is configured to be in GIO mode (FUNC = 0) and configured to be an output pin (DIR = 1). The value of this bit indicates the value to be output to the CAN_TX pin. 0: The CAN_TX pin is driven to logic low 1: The CAN_TX pin is driven to logic high Forced to 1 if INIT bit of DCAN_CTL is reset.	RW	0
0	IN	CAN_TX data in 0: The CAN_TX pin is at logic low 1: The CAN_TX pin is at logic high Note: When CAN_TX pin is connected to a CAN transceiver, an external pullup resistor has to be used to ensure that the CAN bus will not be disturbed (e.g. while reset of the DCAN module).	RW	-

Table 18-609. Register Call Summary for Register DCAN_TIOC

DCAN

- GPIO Support: [0]
- DCAN Register Summary: [1]

Table 18-610. DCAN_RIOC

Address Offset	0x0000 01E4
Physical Address	0x4AE3 C1E4 **Instance** DCAN
Description	RX I/O Control Register The CAN_RX pin of the DCAN_module can be used as general purpose IO pin if CAN function is not needed. The values of the IO control registers are only writable if INIT bit of the DCAN_CTL is set to 1.
Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED													PU	PD	OD	RESERVED												FUNC	DIR	OUT	IN

Bits	Field Name	Description	Type	Reset
31:19	RESERVED	These bits are always read as 0. Writes have no effect.	R	0x0000
18	PU	CAN_RX pull up/pull down select. This bit is only active when CAN_RX is configured to be an input. 0: CAN_RX pull down is selected, when pull logic is active (PD = 0). 1: CAN_RX pull up is selected, when pull logic is active(PD = 0).	RW	0
17	PD	CAN_RX pull disable. This bit is only active when CAN_TX is configured to be an input. 0: CAN_RX pull is active 1: CAN_RX pull is disabled	RW	0
16	OD	CAN_RX open drain enable. This bit is only active when CAN_RX is configured to be in GIO mode (FUNC=0). 0: The CAN_RX pin is configured in push/pull mode. 1: The CAN_RX pin is configured in open drain mode. Forced to '0' if INIT bit of DCAN_CTL is reset.	RW	0
15:4	RESERVED	These bits are always read as 0. Writes have no effect.	R	0x000
3	FUNC	CAN_RX function. This bit changes the function of the CAN_RX pin 0: CAN_RX pin is in GIO mode. 1: CAN_RX pin is in functional mode (as an input to receive CAN data). Forced to '1' if INIT bit of DCAN_CTL is reset.	RW	0
2	DIR	CAN_RX data direction. This bit controls the direction of the CAN_RX pin when it is configured to be in GIO mode only (FUNC=0) 0: The CAN_RX pin is an input. 1: The CAN_RX pin is an output Forced to '0' if INIT bit DCAN_CTL is reset.	RW	0
1	OUT	CAN_RX data out write. This bit is only active when CAN_RX pin is configured to be in GIO mode (FUNC = 0) and configured to be an output pin (DIR = 1). The value of this bit indicates the value to be output to the CAN_RX pin. 0: The CAN_RX pin is driven to logic low 1: The CAN_RX pin is driven to logic high	RW	0
0	IN	CAN_RX data in 0: The CAN_RX pin is at logic low 1: The CAN_RX pin is at logic high	RW	-

Table 18-611. Register Call Summary for Register DCAN_RIOC

DCAN

- GPIO Support: [0]
- DCAN Register Summary: [1]

18.7 MCAN

This chapter describes the Modular Controller Area Network (MCAN) module.

18.7.1 MCAN Overview

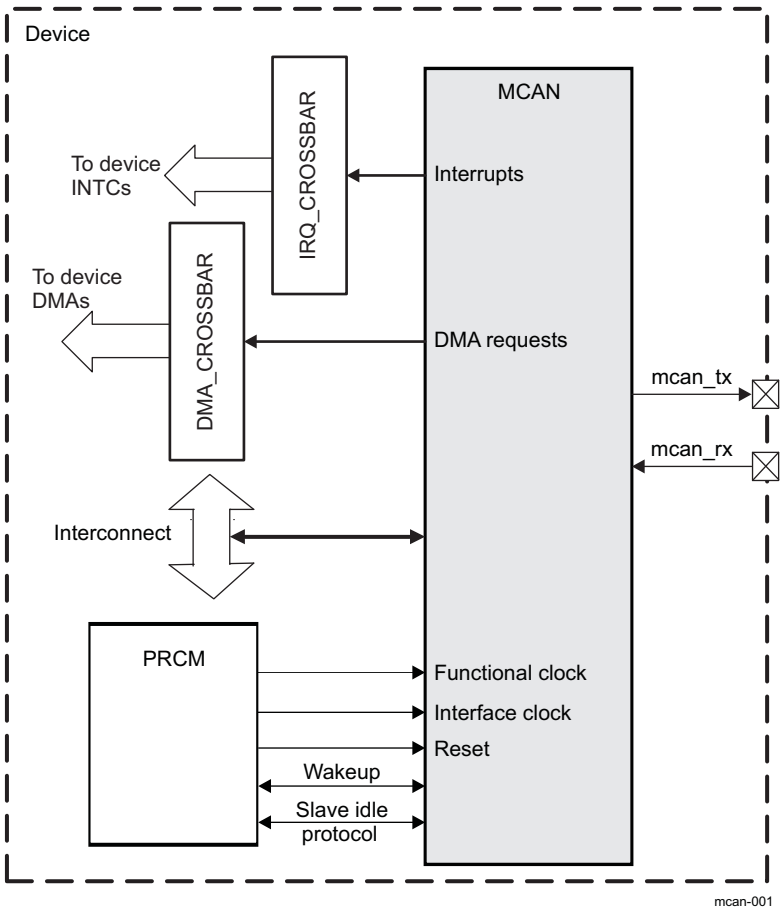
The Controller Area Network (CAN) is a serial communications protocol which efficiently supports distributed real-time control with a high level of security. CAN has high immunity to electrical interference and the ability to self-diagnose and repair data errors. In a CAN network, many short messages are broadcast to the entire network, which provides for data consistency in every node of the system.

The MCAN module supports both classic CAN and CAN FD (CAN with Flexible Data-Rate) specifications. CAN FD feature allows high throughput and increased payload per data frame. The classic CAN and CAN FD devices can coexist on the same network without any conflict.

The device supports one MCAN module connecting to the CAN network through external (for the device) transceiver for connection to the physical layer. The MCAN module supports up to 5 Mbit/s data rate and is compliant to ISO 11898-1:2015.

Figure 18-133 shows the MCAN module overview.

Figure 18-133. MCAN Module Overview



18.7.1.1 Features

The MCAN module implements the following features:

- Conforms with ISO 11898-1:2015
- Full CAN FD support (up to 64 data bytes)
- AUTOSAR and SAE J1939 support

- Up to 32 dedicated Transmit Buffers
- Configurable Transmit FIFO, up to 32 elements
- Configurable Transmit Queue, up to 32 elements
- Configurable Transmit Event FIFO, up to 32 elements
- Up to 64 dedicated Receive Buffers
- Two configurable Receive FIFOs, up to 64 elements each
- Up to 128 filter elements
- Internal Loopback mode for self-test
- Maskable interrupts, two interrupt lines
- Two clock domains (CAN clock/Host clock)
- Parity/ECC support - Message RAM single error correction and double error detection (SECDED) mechanism
- Local power-down and wakeup support
- Timestamp Counter

Not supported features:

- Full Message Memory capacity (4352 words). Only 1600 words implemented.
- Debug on CAN (Debug DMA)
- Host bus read and write bursts
- Host bus firewall
- GPIO mode
- External (IO) Loopback mode
- Device clock domains monitoring (using DCC module)

18.7.2 MCAN Environment

CAN network physical layer consists of two-wire differential bus, usually twisted pair, and provides high level of interference immunity. External CAN transceiver IC is needed to access a CAN bus by the MCAN.

Figure 18-134 shows an overview of a typical MCAN application.

Figure 18-134. MCAN Typical Application

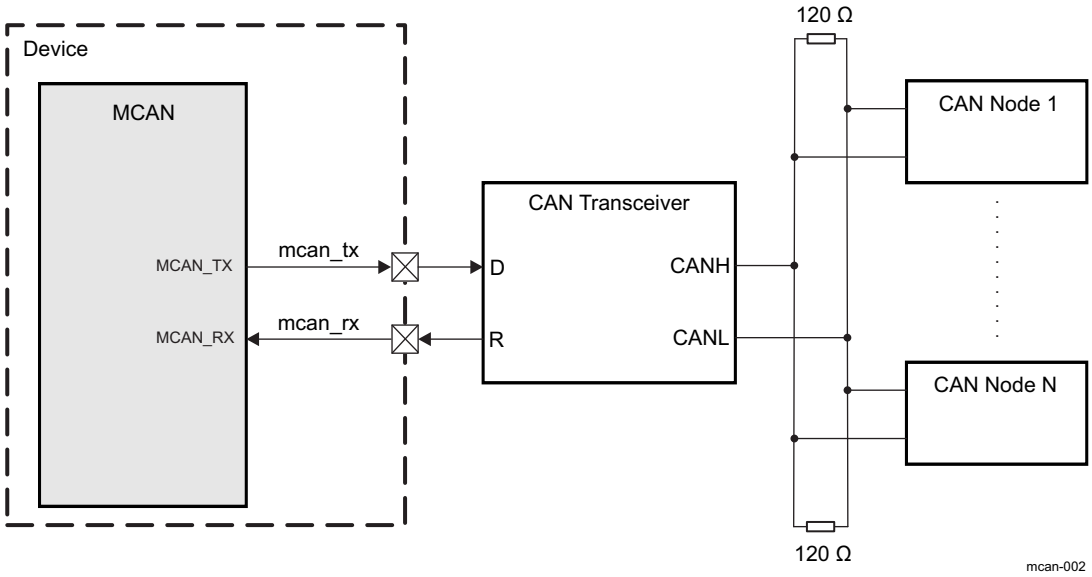


Table 18-612 describes the external signals of the MCAN module.

Table 18-612. MCAN I/O Description

Module Signal	Device Signal	I/O[1]	Description	Value at Reset
MCAN_RX	mcan_rx	I	Serial data input from external CAN tranceiver	HiZ
MCAN_TX	mcan_tx	O	Serial data output to external CAN tranceiver	1

(1) I = Input; O = Output

NOTE: The path from a module pin to device pad(s) is defined at the device I/O logic level. The control module registers assign the specific function to the device pads. For more information on control module settings, see Section 13.4.6.1, *Pad Configuration Registers* of Chapter 13, *Control Module*.

18.7.2.1 CAN Network Basics

- CAN bus is a 2-wire differential bus using Non-Return-to-Zero (NRZ) encoding and has two states:
  - Recessive state (logical 1)
  - Dominant state (logical 0)
- The network is multimaster. When two or more nodes (ECUs) attempt to transmit at the same time, a non-destructive arbitration technique guarantees messages are sent in order of priority and no messages are lost.
- The message transmission is multicast. Data messages transmitted are identifier based, not address based.
- Content of message is labeled by the identifier that is unique throughout the network (for example: rpm, temperature, position, pressure, and so forth).
- All nodes on network receive the message and each performs an acceptance test on the identifier. If

message is relevant, it is processed, otherwise it is ignored.

- The unique identifier also determines the priority of the message (the lower the numerical value of the identifier, the higher the priority is).
- Data is transmitted and received using message frames, consisting of the following basic fields:
  - Arbitration field
  - Control field
  - Data field (up to 8 bytes for Classical CAN and up to 64 bytes for CAN FD)
  - CRC field
  - ACK field

For more information, see *ISO 11898-1:2015: CAN data link layer and physical signalling*.

18.7.3 MCAN Integration

Figure 18-135 shows the integration of the MCAN module in the device.

Figure 18-135. MCAN Integration

Table 18-613 through Table 18-615 summarize the integration of the MCAN module in the device.

Table 18-613. MCAN Integration Attributes

Module Instance	Attributes		
	Power Domain	Wake-Up Capability	Interconnect
MCAN	Yes		

Table 18-614. MCAN Clocks and Resets

Clocks				
Module Instance	**Destination Signal Name**	**Source Signal Name**	**Source**	**Description**
MCAN	MCAN_ICLK		PRCM	Interface clock for the MCAN module
	MCAN_FCLK	MCAN_CLK	PRCM	Functional clock for the MCAN core
Resets				
Module Instance	**Destination Signal Name**	**Source Signal Name**	**Source**	**Description**
MCAN	MCAN_RST		PRCM	Asynchronous reset signal to the MCAN module

Table 18-615. MCAN Hardware Requests

Interrupt Requests				
Module Instance	**Source Signal Name**	**IRQ_CROSSBAR Input**	**Default Mapping**	**Description**
MCAN	MCAN_IRQ_INT0	IRQ_CROSSBAR_		MCAN interrupt 0
	MCAN_IRQ_INT1	IRQ_CROSSBAR_		MCAN interrupt 1
	MCAN_IRQ_ECC	IRQ_CROSSBAR_		MCAN ECC interrupt
	MCAN_IRQ_TS	IRQ_CROSSBAR_		MCAN timestamp interrupt
DMA Requests				
Module Instance	**Source Signal Name**	**DMA_CROSSBAR Input**	**Default Mapping**	**Description**
MCAN	MCAN_DREQ_TX	DMA_CROSSBAR_		MCAN TX DMA Event
	MCAN_DREQ_RX_FE1	DMA_CROSSBAR_		MCAN RX Filter Event 1
	MCAN_DREQ_RX_FE2	DMA_CROSSBAR_		MCAN RX Filter Event 2

NOTE: For more information about the IRQ_CROSSBAR module, see Section 13.4.6.4, *IRQ_CROSSBAR Module Functional Description*, in Chapter 13, *Control Module*.

For more information about the device interrupt controllers, see Chapter 12, *Interrupt Controllers*.

NOTE: For more information about the DMA_CROSSBAR module, see Section 13.4.6.5, *DMA_CROSSBAR Module Functional Description*, in Chapter 13, *Control Module*.

NOTE: For the description of the interrupt source, see Section 18.7.4.2, *Interrupt and DMA Requests*.

18.7.4 MCAN Functional Description

The MCAN module performs CAN protocol communication according to ISO 11898-1:2015. The bit rate can be programmed to values up to 5 Mbit/s. Additional transceiver hardware is required for the connection to the physical layer (CAN bus).

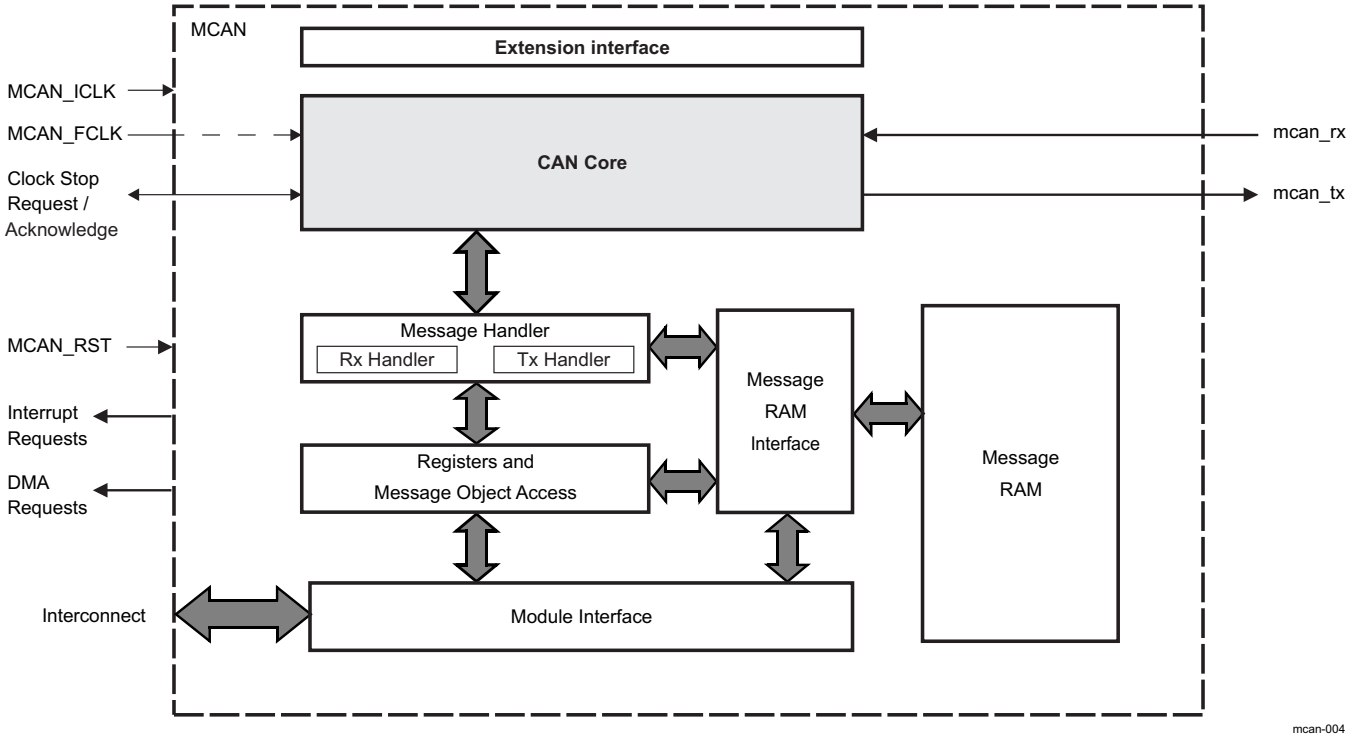
For communication on a CAN network, individual message frames can be configured. The message frames and identifier masks are stored in the Message RAM.

All functions concerning the handling of messages are implemented in the Message Handler.

The register set of the MCAN module can be accessed directly via the module interface. These registers are used to control and configure the CAN core and the Message Handler, and to access the Message RAM.

Figure 18-136 shows the MCAN module block diagram.

Figure 18-136. MCAN Block Diagram



The MCAN module blocks description:

- **CAN Core**: The CAN core consists of the CAN protocol controller and the Rx/Tx shift register. It handles all ISO 11898-1:2015 protocol functions and supports 11-bit and 29-bit identifiers.
- **Message Handler**: the Message Handler (Rx Handler and Tx Handler) is a state machine that controls the data transfer between the single-ported Message RAM and the CAN core's Rx/Tx shift register. It also handles the acceptance filtering and the Interrupt/DMA request generation as programmed in the control registers.
- **Message RAM**: the main purpose of the Message RAM is to store Rx/Tx messages, Tx Event elements, and Message ID Filter elements (for more information, see Section 18.7.4.11, *Message RAM*).
- **Message RAM Interface**: enables connection between the Message RAM and the other blocks in the MCAN module.
- **Registers and Message Object Access**: Data consistency is ensured by indirect accesses to the message objects. During normal operation, all software and DMA accesses to the Message RAM are done through interface registers. The interface registers have the same word-length as the Message RAM.

- **Module Interface**: The MCAN module registers are accessed by the user software through a 32-bit peripheral bus interface.
- **Clocking**: Two clocks are provided to the MCAN module: the peripheral synchronous clock (interface clock - MCAN_ICLK) and the peripheral asynchronous clock (functional clock - MCAN_FCLK).
- **Extension Interface**: All flags from the Interrupt Register (MCAN_IR) as well as selected internal status and control signals are routed to this interface.

18.7.4.1 Module Clocking Requirements

Two clocks are provided to the MCAN module:

- the peripheral synchronous clock (MCAN_ICLK) as the general module clock source
- and the peripheral asynchronous clock (MCAN_FCLK) provided to the CAN core for generating the CAN bit timing.

Within the MCAN module there is a synchronization mechanism implemented to ensure safe data transfer between the two clock domains. There are synchronization between the signals from the Host clock domain to the CAN clock domain and vice versa and between the reset signal (MCAN_RST) to the Host clock domain and to the CAN clock domain.

NOTE: MCAN_ICLK must always be higher or equal to MCAN_FCLK, in order to achieve a stable functionality of the MCAN module. Here, also the frequency shift of the modulated MCAN_ICLK has to be considered:

$f_{0,\ ICLK}(OCP) \pm \Delta f_{FM,\ ICLK}(OCP) \geq f_{FCLK}$

CAN-FD supports higher speeds of operation and as such has more stringent timing requirements than Classic CAN. For optimal performance, TI recommends using the lowest N-divider value that maintains a working PLL REF_CLK (GMAC_DSP_DPLL_CLK) for the system. Lower N-divider values increase the loop bandwidth of the PLL which in turn improves timing margins for CAN-FD.

For CAN-FD operations > 2 Mbps:

- For 20 MHz input clocks, N = 0 is the preferred configuration.

For CAN-FD operations < 2 Mbps:

- For 20 MHz input clocks, N = 0 is the preferred configuration.
- For 19.2 MHz input clocks, N = 11 is the preferred configuration.

For more information on how to configure the relevant clock source registers, see Chapter 3, *PRCM* and the device data manual.

18.7.4.2 Interrupt and DMA Requests

The MCAN module provides interrupt and DMA requests. They are configured via the Host CPU. The Suspend Mode prevents the interrupt and DMA requests from propagating to the Host CPU (for more information, see Section 18.7.4.4.8.2, *Suspend Mode*).

18.7.4.2.1 Interrupt Requests

The MCAN module has two interrupt lines. The first interrupt line (INT0) is associated with the MCAN core. There are 30 internal interrupt sources. The interrupts are 'level high' interrupts.

For more information, see the following registers:

- Interrupt Register (MCAN_IR)
- Interrupt Enable (MCAN_IE)
- Interrupt Line Select (MCAN_ILS)
- Interrupt Line Enable (MCAN_ILE)

The MCAN module is capable of issuing an ECC interrupt. After clearing the ECC interrupt source, the application software must also write 1 to MCANSS_ECC_EOI[8] ECC_EOI bit (for more information, see Section 18.7.4.7.2, *ECC Aggregator*).

The second interrupt line (INT1) is associated with the External Timestamp Counter. When the External Timestamp Counter rolls over it produces an interrupt (see Section 18.7.4.5.1, *External Timestamp Counter*).

For more information, see the following registers:

- Interrupt Clear Shadow Register (MCANSS_ICS)
- Interrupt Raw Status Register (MCANSS_IRS)
- Interrupt Enable Clear Shadow Register (MCANSS_IECS)
- Interrupt Enable Register (MCANSS_IE)
- Interrupt Enable Status (MCANSS_IES)
- End Of Interrupt (MCANSS_EOI)
- External Timestamp Prescaler (MCANSS_EXT_TS_PRESCALER)
- External Timestamp Unserviced Interrupts Counter (MCANSS_EXT_TS_UNSERVICED_INTR_CNTR)

18.7.4.2.2 DMA Requests

Functional transmit and Filter DMA requests are generated by the MCAN module based on the signaling in the Extention Interface. The DMA signaling uses a simple DMA request active high pulse. Only one Tx DMA event is provided by the MCAN module.

Standard and Extended message filters can be set to issue a pulse when a filter match occurs. These 'Filter Events' can be used to DMA messages from the Rx FIFO. The events are high level single clock cycle (MCAN_ICLK) pulses. Only two Filter DMA events are provided by the MCAN module.

For more information about available Interrupt and DMA Requests, see Section 18.7.3, *MCAN Integration*.

18.7.4.3 Fuseable CAN FD Operation Enable

The Flexible Datarate feature of the MCAN module can be enabled by writing 1 to MCAN_CCCR[8] FDOE bit. A value of 0 on the primary configuration port (mcanss_enable_fdoe) will force the MCAN_CCCR[8] FDOE bit during write to the MCAN_CCCR register which will prevent the device from enabling and using the CAN FD mode.

18.7.4.4 Operating Modes

18.7.4.4.1 Software Initialization

Setting the MCAN_CCCR[0] INIT bit to 1 starts a software initialization. This is done either by software or by a hardware reset, when an uncorrected bit error was detected in the Message RAM, or by going Bus_Off state. While the MCAN_CCCR[0] INIT bit is set, the message transfer is stopped and the status of the output MCAN_TX pin is recessive (high). The counters of the Error Management Logic (EML) are unchanged. Setting the MCAN_CCCR[0] INIT bit does not change any configuration register. Resetting the MCAN_CCCR[0] INIT bit finishes the software initialization. After waiting for the occurrence of a sequence of 11 consecutive recessive bits (indication for Bus_Idle state) the message transfer starts.

Access to the MCAN configuration registers is only enabled when both MCAN_CCCR[0] INIT and MCAN_CCCR[1] CCE bits are set (write protection).

The MCAN_CCCR[1] CCE bit can only be set/reset while the MCAN_CCCR[0] INIT = 1. The MCAN_CCCR[1] CCE bit is automatically reset when the MCAN_CCCR[0] INIT bit is reset.

The following registers are reset when the MCAN_CCCR[1] CCE bit is set:

- MCAN_HPMS - High Priority Message Status
- MCAN_RXF0S - Rx FIFO 0 Status
- MCAN_RXF1S - Rx FIFO 1 Status

- MCAN_TXFQS - Tx FIFO/Queue Status
- MCAN_TXBRP - Tx Buffer Request Pending
- MCAN_TXBTO - Tx Buffer Transmission Occurred
- MCAN_TXBCF - Tx Buffer Cancellation Finished
- MCAN_TXEFS - Tx Event FIFO Status

The Timeout Counter value MCAN_TOCV[15:0] TOC field is preset to the value configured by the MCAN_TOCC[31:16] TOP field when the MCAN_CCCR[1] CCE bit is set.

In addition the Tx Handler and Rx Handler are held in idle state while MCAN_CCCR[1] CCE = 1.

The following registers are only writeable while MCAN_CCCR[1] CCE = 0

- MCAN_TXBAR - Tx Buffer Add Request
- MCAN_TXBCR - Tx Buffer Cancellation Request

MCAN_CCCR[7] TEST and MCAN_CCCR[5] MON bits can only be set by the Host CPU while MCAN_CCCR[0] INIT = 1 and MCAN_CCCR[1] CCE = 1. Both bits may be reset at any time. The MCAN_CCCR[6] DAR bit can only be set/reset while MCAN_CCCR[0] INIT = 1 and MCAN_CCCR[1] CCE = 1.

18.7.4.4.2 Normal Operation

Once the MCAN module is initialized and the MCAN_CCCR[0] INIT bit is reset to zero, the MCAN module synchronizes itself to the CAN bus and is ready for communication. After passing the acceptance filtering, received messages including Message Identifier (ID) and Data Length Code (DLC) are stored into a dedicated Rx Buffer or into Rx FIFO 0/Rx FIFO 1.

For messages to be transmitted dedicated Tx Buffers and/or a Tx FIFO or a Tx Queue can be initialized or updated.

NOTE: Automated transmission on reception of remote frames is not supported.

18.7.4.4.3 CAN FD Operation

There are two variants of CAN FD frame transmission:

- CAN FD frame transmission without bit rate switching
- CAN FD frame transmission where control field, data field, and CRC field are transmitted with a higher bit rate than the beginning and the end of the frame

In the CAN frames FDF = recessive (logical 1) signifies a CAN FD frame, FDF = dominant (logical 0) signifies a Classic CAN frame. In a CAN FD frame, the two bits following FDF - res and BRS, decide whether the bit rate inside of this CAN FD frame is switched. A CAN FD bit rate switch is signified by res = dominant and BRS = recessive. Note that the coding of res = recessive is reserved for future expansion of the protocol. In case the MCAN module receives a frame with FDF = recessive and res = recessive, it will signal a Protocol Exception Event by setting the MCAN_PSR[14] EXE bit. When Protocol Exception Handling is enabled (MCAN_CCCR[12] PXHD = 0), this causes the operation state to change from Receiver (MCAN_PSR[4:3] ACT = 10) to Integrating (MCAN_PSR[4:3] ACT = 00) at the next sample point. In case Protocol Exception Handling is disabled (MCAN_CCCR[12] PXHD = 1), the MCAN will treat a recessive res bit as an form error and will respond with an error frame.

CAN FD operation is enabled by programming the MCAN_CCCR[8] FDOE bit. In case MCAN_CCCR[8] FDOE = 1, transmission and reception of CAN FD frames is enabled. Transmission and reception of Classic CAN frames is always possible. Whether a CAN FD frame or a Classic CAN frame is transmitted can be configured via the FDF bit in the respective Tx Buffer element.

With MCAN_CCCR[8] FDOE = 0, received frames are interpreted as Classic CAN frames, which leads to the transmission of an error frame when receiving a CAN FD frame. When CAN FD operation is disabled, no CAN FD frames are transmitted even if the FDF bit of a Tx Buffer element is set. The MCAN_CCCR[8] FDOE and MCAN_CCCR[9] BRSE bits can only be changed while the MCAN_CCCR[0] INIT and MCAN_CCCR[1] CCE bits are both set. With MCAN_CCCR[8] FDOE = 0, the setting of bits FDF and BRS is ignored and frames are transmitted in Classic CAN format.

With MCAN_CCCR[8] FDOE = 1 and MCAN_CCCR[9] BRSE = 0, only FDF bit of a Tx Buffer element is evaluated. With MCAN_CCCR[8] FDOE = 1 and MCAN_CCCR[9] BRSE = 1, transmission of CAN FD frames with bit rate switching is enabled. All Tx Buffer elements with bits FDF and BRS set are transmitted in CAN FD format with bit rate switching.

A mode change during CAN operation is only recommended under the following conditions:

- The failure rate in the CAN FD data phase is significant higher than in the CAN FD arbitration phase. In this case disable the CAN FD bit rate switching option for transmissions.
- During system startup all nodes are transmitting Classic CAN messages until it is verified that they are able to communicate in CAN FD format. If this is true, all nodes switch to CAN FD operation.
- Wake-up messages in CAN Partial Networking have to be transmitted in Classic CAN format.
- End-of-line programming in case not all nodes are CAN FD capable. Non CAN FD nodes are held in Silent mode until programming has completed. Then all nodes switch back to Classic CAN communication.

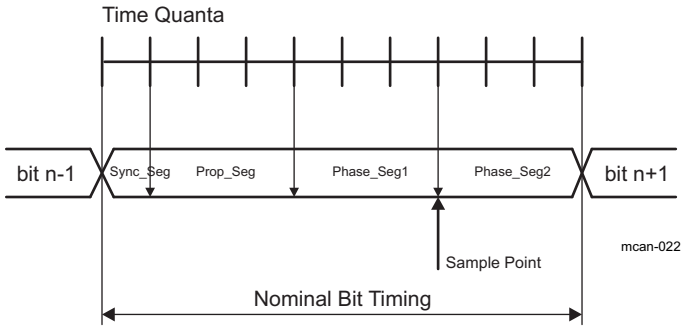
In the CAN FD format, the DLC coding differs from the standard CAN format (see Table 18-616).

Table 18-616. DLC Coding

DLC	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
Number of Data Bytes in Standard CAN	0	1	2	3	4	5	6	7	8	8	8	8	8	8	8	8
Number of Data Bytes in CAN FD	0	1	2	3	4	5	6	7	8	12	16	20	24	32	48	64

For CAN FD frames with bit rate switchingIn, the bit timing will be switched inside the frame after the BRS (Bit Rate Switch) bit in case this bit is recessive. In the CAN FD arbitration phase, before the BRS bit, the nominal CAN bit timing (see Figure 18-137) is used as configured by the Nominal Bit Timing and Prescaler Register MCAN_NBTP. In the following CAN FD data phase, the data phase bit timing is used as configured by the Data Bit Timing and Prescaler Register MCAN_DBTP. The bit timing is switched back from the data phase timing at the CRC delimiter or when an error is detected, whichever occurs first.

Figure 18-137. CAN Bit Timing



The maximum configurable data phase bit timing depends on the CAN clock frequency (MCAN_FCLK). Example: with MCAN_FCLK = 20 MHz and the shortest configurable bit time of 4 t_q (time quanta), the bit rate in the data phase is 5 Mbit/s.

For both CAN FD without and CAN FD with bit rate switching the value of the ESI (Error Status Indicator) bit depends on transmitter's error state (see MCAN_PSR[11] RESI bit) monitored at the start of the transmission. If the transmitter has error passive flag the ESI bit is transmitted recessive, else it is transmitted dominant.

18.7.4.4.4 Transmitter Delay Compensation

18.7.4.4.4.1 Description

When only one CAN FD node is transmitting and all others are receivers the length of the bus line has no impact. When transmitting via the MCAN_TX pin the MCAN module receives the transmitted data from its local CAN transceiver via the MCAN_RX pin. The received data is delayed. If the transmitter delay is greater than TSEG1 (time segment before sample point), a bit error is detected.

The MCAN module provides a delay compensation mechanism to compensate the transmitter delay. The compensation mechanism enables transmission with higher bit rates during the CAN FD data phase independent of the delay of a specific CAN transceiver. Without transmitter delay compensation the bit rate in the data phase is limited by the transmitter delay.

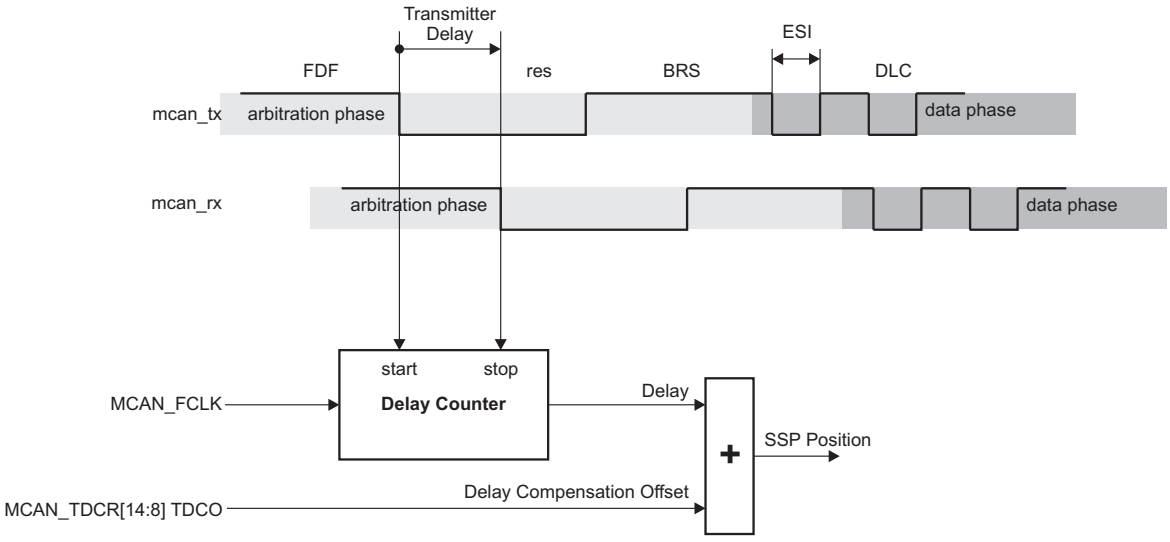
The mechanism enables configurations where the data bit time is shorter than the transmitter delay (it is described in detail in ISO 11898-1:2015). The transmitter delay compensation is enabled by setting the MCAN_DBTP[23] TDC bit to 1.

The delayed transmit data is compared against the received data at the Secondary Sample Point (SSP) in order to check for bit errors during the data phase of transmitting nodes. If a bit error is detected, the transmitter will react on this bit error at the next following regular sample point. During arbitration phase the delay compensation is always disabled.

The received bit is compared against the transmitted bit at the SSP. The SSP position is defined as the sum of the measured delay from the MCAN's transmit output MCAN_TX pin through the transceiver to the receive input MCAN_RX pin plus the transmitter delay compensation offset configured by the MCAN_TDCR[14:8] TDCO field (see Figure 18-138). The transmitter delay compensation offset is used to adjust the position of the SSP inside the received bit (example: half of the bit time in the data phase). The position of the SSP is rounded down to the next integer number of mtq.

The actual transmitter delay compensation value can be checked by reading the MCAN_PSR[22:16] TDCV field. This filed is cleared when the MCAN_CCCR[0] INIT bit is set and is updated at each transmission of CAN FD frame while the MCAN_DBTP[23] TDC bit is set.

Figure 18-138. Transmitter Delay Measurement



18.7.4.4.4.2 Transmitter Delay Compensation Measurement

When transmitter delay compensation is enabled (by programming MCAN_DBTP[23] TDC = 1), the measurement is started within each transmitted CAN FD frame at the falling edge of FDF bit to bit res. The measurement is stopped when this edge is seen at the receive input MCAN_RX pin of the transmitter. The resolution of this measurement is one mtq (see Figure 18-138). The mtq (minimum time quantum) dimension is equal to the CAN clock period (MCAN_FCLK).

The use of a transmitter delay compensation filter window can be enabled by programming MCAN_TDCR[6:0] TDCF field. This filter feature defines a minimum value for the SSP position to avoid the case in which a dominant glitch inside the received FDF bit ends the delay compensation measurement before the falling edge of the received res bit, resulting in an early taken SSP position. Dominant edges on the MCAN_RX pin, that would result in an earlier SSP position are ignored for transmitter delay measurement. The measurement is stopped when the SSP position is at least MCAN_TDCR[6:0] TDCF field and the MCAN_RX pin is low.

The following boundary conditions have to be considered:

- The sum of the measured delay from the MCAN_TX pin to the MCAN_RX pin and the configured transmitter delay compensation offset (MCAN_TDCR[14:8] TDCO field) has to be less than 6 bit times in the data phase.
- The sum of the measured delay from the MCAN_TX pin to the MCAN_RX pin and the configured transmitter delay compensation offset (MCAN_TDCR[14:8] TDCO) field has to be less or equal 127 mtq. In case this sum exceeds 127 mtq, the maximum value of 127 mtq is used for transmitter delay compensation.
- The data phase ends at the sample point of the CRC delimiter, that stops checking of receive bits at the SSPs.

18.7.4.4.5 Restricted Operation Mode

In Restricted Operation Mode the CAN node is able to receive data and remote frames and to give acknowledge to valid frames, but it does not send data frames, remote frames, active error frames, or overload frames. In case of an error condition or overload condition, it does not send dominant bits, instead it waits for the occurrence of bus idle condition to resynchronize itself to the CAN communication. The recieve and transmit error counters (MCAN_ECR[14:8] REC and MCAN_ECR[7:0] TEC) are frozen while CAN error logging (MCAN_ECR[23:16] CEL) is active. The Host CPU can set the MCAN module into Restricted Operation Mode by setting MCAN_CCCR[2] ASM bit. The bit can only be set by the Host CPU at any time when both MCAN_CCCR[2] CCE and MCAN_CCCR[1] INIT bits are set to 1.

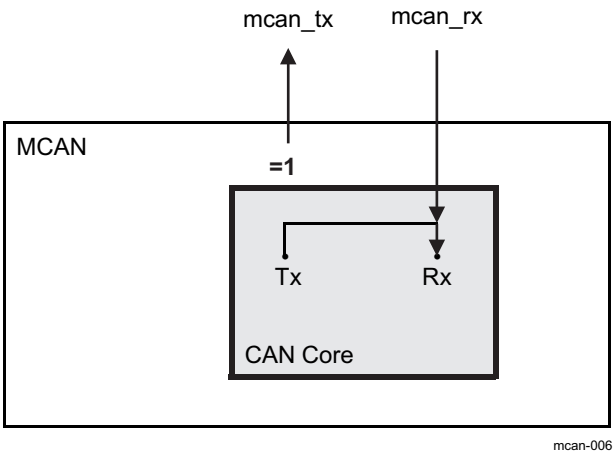
The Restricted Operation Mode is automatically entered when the Tx Handler was not able to read data from the Message RAM in time. To leave Restricted Operation Mode, the Host CPU has to reset MCAN_CCCR[2] ASM bit. This mode can be used in applications that adapt themselves to different CAN bit rates. In this case the application tests different bit rates and leaves the Restricted Operation Mode after it has received a valid frame.

NOTE: The Restricted Operation Mode must not be combined with the Loop Back Mode.

18.7.4.4.6 Bus Monitoring Mode

Entering Bus Monitoring Mode is done by setting the MCAN_CCCR[5] MON bit to 1. In this mode (see ISO 11898-1:2015, *Bus Monitoring* section), the MCAN module is able to receive valid data and remote frames, but cannot start a transmission. The MCAN module sends only recessive bits on the CAN bus. If the MCAN module is required to send a dominant bit (ACK bit, overload flag, active error flag), the bit is rerouted internally so that the MCAN module monitors this dominant bit, although the CAN bus may remain in recessive state. In Bus Monitoring Mode the MCAN_TXBRP register is held in reset state. The Bus Monitoring Mode can be used to analyze the traffic on a CAN bus without affecting it by the transmission of dominant bits. Figure 18-139 shows the connection of the MCAN_TX and MCAN_RX signals to the MCAN module in Bus Monitoring Mode.

Figure 18-139. Connection of Signals in Bus Monitoring Mode



18.7.4.4.7 Disabled Automatic Retransmission (DAR) Mode

According to the CAN Specification (see ISO11898-1:2015, *Recovery Management* section), the MCAN module provides means for automatic retransmission of frames that have lost arbitration or that have been disturbed by errors during transmission. By default automatic retransmission is enabled (see the MCAN_CCCR[6] DAR bit).

18.7.4.4.7.1 Frame Transmission in DAR Mode

In DAR mode all transmissions are automatically cancelled after they started on the CAN bus. A Tx Buffer's Tx Request Pending MCAN_TXBRP[xx] TRPx bit is reset after successful transmission, when a transmission has not yet been started at the point of cancellation, has been aborted due to lost arbitration, or when an error occurred during frame transmission.

Successful transmission:

- Corresponding Tx Buffer Transmission Occurred MCAN_TXBTO[xx] TOx bit is set
- Corresponding Tx Buffer Cancellation Finished MCAN_TXBCF[xx] CFx bit is not set

Successful transmission in spite of cancellation:

- Corresponding Tx Buffer Transmission Occurred MCAN_TXBTO[xx] TOx bit is set
- Corresponding Tx Buffer Cancellation Finished MCAN_TXBCF[xx] CFx bit is set

Arbitration lost or frame transmission disturbed:

- Corresponding Tx Buffer Transmission Occurred MCAN_TXBTO[xx] TOx bit is not set
- Corresponding Tx Buffer Cancellation Finished MCAN_TXBCF[xx] CFx bit is set

In case of a successful frame transmission, and if storage of Tx events is enabled, a Tx Event FIFO element is written with Event Type ET = 10 (transmission in spite of cancellation).

18.7.4.4.8 Power Down (Sleep Mode)

Entering Power Down mode is controlled via the input clock stop request signal (mcanss_clkstp_clkstop_req) or MCAN_CCCR[4] CSR bit. As long as the clock stop request signal is active, the MCAN_CCCR[4] CSR bit is read as 1. When all pending transmission requests have completed, the MCAN module waits until bus idle state is detected. Then the MCAN module sets the MCAN_CCCR[1] INIT to 1 to prevent any further CAN transfers. The MCAN module acknowledges that it is ready for power down by setting the output clock stop acknowledge signal (mcanss_clkstop_clkstop_ack) to 1 and the MCAN_CCCR[3] CSA bit to 1. In this state, before the clocks are switched off, further register accesses can be made. A write access to the MCAN_CCCR[1] INIT bit will have no effect. Now the module clock inputs MCAN_ICLK and MCAN_FCLK may be switched off.

To leave power down mode, the application has to turn on the module clocks before resetting the input clock stop request signal respectively the MCAN_CCCR[4] CSR flag bit. The MCAN will acknowledge this by resetting the output clock stop acknowledge signal respectively the MCAN_CCCR[3] CSA flag bit. Afterwards, the application can restart CAN communication by resetting MCAN_CCCR[1] INIT bit.

18.7.4.4.8.1 External Clock Stop Mode

The MCAN module supports two external clock stop modes:

- Immediate
- Graceful

In a graceful clock stop mode, when the clock stop request is asserted, the MCAN core will respond with clock stop acknowledge when all pending Tx messages have been processed and an Idle line had been detected. The MCAN_CCCR[0] INIT bit will be set, the MCAN core will go and stay Idle.

The automatic wakeup feature is enabled by setting the MCANSS_CTRL[5] AUTOWAKEUP and MCANSS_CTRL[4] WAKEUPREQEN bits to 1 (for more information, see Section 18.7.4.4.8.3, *Wakeup request*). When external clock stop request is removed and no suspend request is active, a read-modify-write to the MCAN_CCCR[0] INIT bit is performed to clear it.

18.7.4.4.8.2 Suspend Mode

The MCAN module supports two suspend modes:

- Immediate
- Graceful

In a graceful suspend mode (see the MCANSS_CTRL[3] FREE and MCANSS_CTRL[2] SOFT bits), when the suspend request is asserted, a clock stop request to the MCAN core is performed. The MCAN core will respond with clock stop acknowledge when all pending Tx messages have been processed and an Idle line had been detected. At that point the MCAN_CCCR[0] INIT bit will be set, the MCAN core will go and stay Idle. The suspend state can be verified by reading MCAN_CCCR[0] INIT bit.

The automatic wakeup feature is enabled by setting the MCANSS_CTRL[5] AUTOWAKEUP and MCANSS_CTRL[4] WAKEUPREQEN bits to 1 (for more information, see Section 18.7.4.4.8.3, *Wakeup request*). When suspend request is removed, if no external clock stop request is active, a read-modify-write to the MCAN_CCCR[0] INIT bit is performed to clear it.

During suspend mode the auto-clear feature is disabled. The following register fields have an auto-clear feature:

- MCAN_ECR[23:16] CEL
- MCAN_PSR[2:0] LEC
- MCAN_PSR[10:8] DLEC
- MCAN_PSR[11] RESI
- MCAN_PSR[12] RBRS
- MCAN_PSR[13] RFDF
- MCAN_PSR[14] PXE

18.7.4.4.8.3 Wakeup request

Issuing a clock stop request puts the MCAN module into Power Down mode (Sleep Mode). During transition from IDLE to ACTIVE, if the MCANSS_CTRL[5] AUTOWAKEUP and MCANSS_CTRL[4] WAKEUPREQEN bits are enabled, after the MCAN Core respond to the removal of the clock stop request with removing the clock stop acknowledge, a read-modify-write will be issued to clear the MCAN_CCCR[0] INIT bit and the MCAN core will resume operation.

If the MCANSS_CTRL[4] WAKEUPREQEN bit is set, the MCAN module provides a wakeup request (SWakeup) on any of the following wakeup events:

- The receive MCAN_RX pin is dominant (logical 0)
- OCP access is performed

To clear the SWakeup in case any of these events is active, the MCANSS_CTRL[4] WAKEUPREQEN bit should be cleared. The MCAN module adds a third wakeup event source - interrupt line 0 (INT0). In this case the SWakeup is cleared by clearing the interrupt source.

18.7.4.4.9 Test Modes

The MCAN_TEST register write access is enabled by setting the test mode enable MCAN_CCCR[7] TEST bit to 1. The MCAN_TEST register allows the configuration of the test modes and test functions.

The CAN transmit MCAN_TX pin has four output functions. One of those functions can be selected by programming the MCAN_TEST[6:5] TX filed. Additionally to its default function (the serial data output) it can drive the CAN Sample Point signal to monitor the MCAN's bit timing and it can drive constant dominant or recessive values.

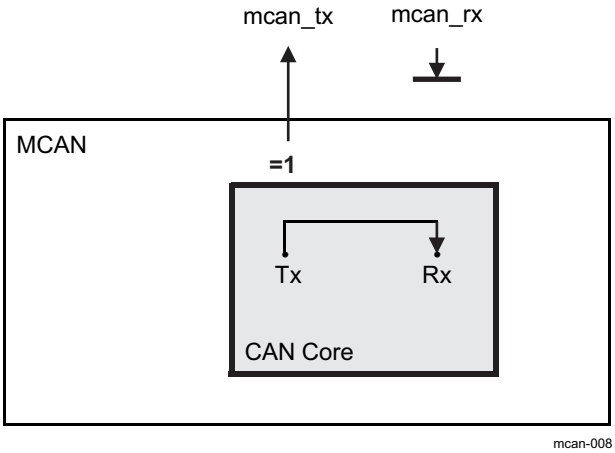
The actual value of the CAN receive MCAN_RX pin can be monitored from MCAN_TEST[7] RX bit. Both functions can be used to check the CAN bus physical layer. Due to the synchronization mechanism between CAN clock (MCAN_FCLK) and Host clock (MCAN_ICLK) domain, there may be a delay of several Host clock periods between writing to the MCAN_TEST[6:5] TX filed until the new configuration is visible at the output MCAN_TX pin. This applies also when reading input MCAN_RX pin via the MCAN_TEST[7] RX bit.

NOTE: Test modes should be used for self test only. The software control for MCAN_TX pin interferes with all CAN protocol functions. It is not recommended to use test modes for application.

18.7.4.4.9.1 Internal Loop Back Mode

The MCAN module can be set into Internal Loop Back Mode by programming MCAN_TEST[4] LBCK and MCAN_CCCR[5] MON bits to 1. The Internal Loop Back Mode is used for a 'Hot Selftest'. The 'Hot Selftest' allows the MCAN module to be tested without affecting a running CAN system connected to the MCAN_TX and MCAN_RX pins. In this mode MCAN_RX pin is disconnected from the MCAN module and MCAN_TX pin is held recessive. Figure 18-140 shows the connection of the MCAN_TX and MCAN_RX pins to the MCAN module in case of Internal Loop Back Mode.

Figure 18-140. Internal Loop Back Mode



18.7.4.5 Timestamp Generation

The MCAN module has integrated a 16-bit wrap-around counter for timestamp generation. The timestamp counter prescaler MCAN_TSCC[19:16] TCP field can be configured to clock the counter in multiples of CAN bit times (1-16). The counter is readable via the MCAN_TSCV[15:0] TSC field. A write access to the MCAN_TSCV register resets the counter to zero. When the timestamp counter wraps around the interrupt MCAN_IR[16] TSW flag is set. On start of a frame reception/transmission the counter value is captured and stored into the timestamp section of an Rx Buffer/Rx FIFO (RXTS[15:0]) or Tx Event FIFO (TXTS[15:0]) element. For more information, see Section 18.7.4.11, *Message RAM*.

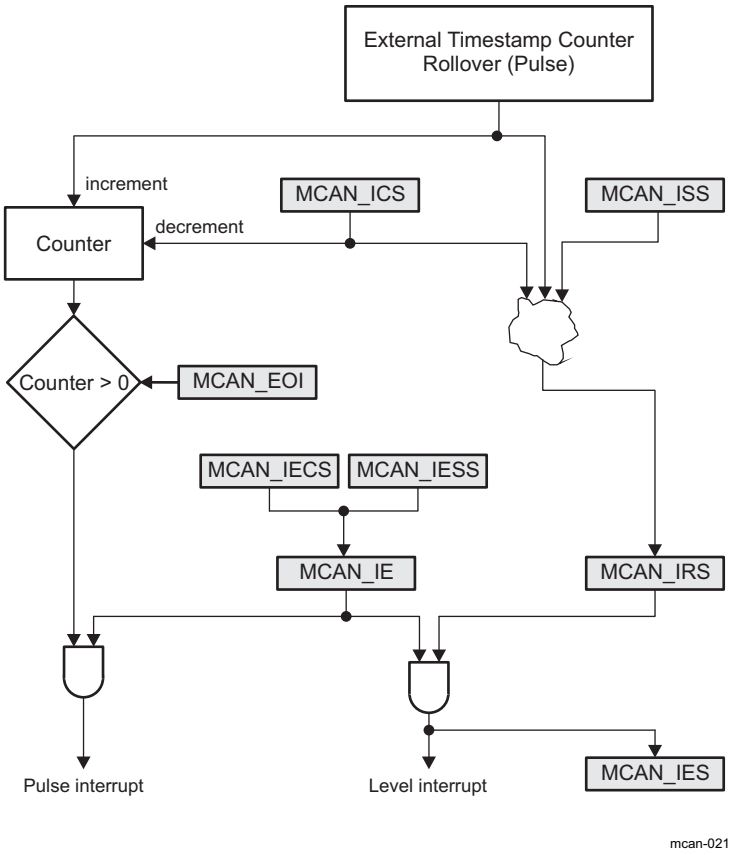
18.7.4.5.1 External Timestamp Counter

For CAN FD operation mode the MCAN core requires an External Timestamp Counter. An externally generated 16-bit vector may substitute the integrated 16-bit CAN bit time counter (internal timestamp counter) for receive and transmit timestamp generation. An external 16-bit timestamp counter can be used by programming the MCAN_TSCC[1:0] TSS field.

The External Timestamp Counter uses the interface clock (MCAN_ICLK) as a reference clock. The MCAN Core accepts a 16-bit timestamp. A 24-bit prescaler provides a programmable resolution for the timestamp (see MCANSS_EXT_TS_PRESCALER[23:0] PRESCALER field). When disabled the counter is reset back to zero. While enabled the counter keeps incrementing. When the timestamp rolls over the MCAN_IRQ_TS interrupt is generated. The MCAN module provides both pulse and level interrupt type for this interrupt.

When the timestamp rolls over the MCANSS_IRS register is set (see Figure 18-141). The MCANSS_IE register can be affected by writing to the MCAN_IESS register to set or to the MCANSS_IECS register to clear. The level interrupt is a reflection of both MCANSS_IRS and MCANSS_IE being set. The MCANSS_IES register reflects the level interrupt. When an rollover event occurs the interrupt counter is incremented. Writing to the MCANSS_ICS register to clear the MCANSS_IRS register will also decrement the interrupt counter. Writing to the MCANSS_EOI register will issue another pulse if the interrupt counter is not zero.

Figure 18-141. External Timestamp Counter Interrupt



18.7.4.6 Timeout Counter

The MCAN module has integrated a 16-bit Timeout Counter. It is used to signal timeout conditions for the Rx FIFO 0, Rx FIFO 1, and Tx Event FIFO Message RAM elements. The Timeout Counter is configured via the MCAN_TOCC register. It is enabled via the MCAN_TOCC[0] ETOC bit. The Timeout Counter operates as down-counter and uses the same prescaler programmed by the MCAN_TSCC[19:16] TCP field as the Timestamp Counter. The actual counter value can be monitored from the MCAN_TOCV[15:0] TOC field. The Timeout Counter can be started only when MCAN_CCCR[1] INIT = 0 and stopped when MCAN_CCCR[1] INIT = 1 (example: when the MCAN enters Bus_Off state). The operation mode is selected by the MCAN_TOCC[2:1] TOS field. When Continuous Mode is selected, the counter starts when MCAN_CCCR[1] INIT = 0, a write to the MCAN_TOCV register presets the counter to the value configured by the MCAN_TOCC[31:16] TOP field and continues down-counting.

In case the Timeout Counter is controlled by one of the FIFOs, an empty FIFO presets the counter to the value configured by the MCAN_TOCC[31:16] TOP field. Down-counting is started when the first FIFO element is stored. Writing to the MCAN_TOCV register has no effect. When the counter reaches zero, the interrupt MCAN_IR[18] TOO flag is set.

In Continuous Mode, the counter is immediately restarted at the value configured by the MCAN_TOCC[31:16] TOP field.

18.7.4.7 Safety

The Message Memory is wrapped in an ECC wrapper provididing SECDED parity functionalilty. The ECC wrapper is controlled by an ECC Aggregator.

18.7.4.7.1 ECC Wrapper

The ECC wrapper provides Single Error Correction (SEC) and Double Error Detection (DED) parity to the Message Memory content. It has side band signals for error notification. The ECC Wrapper implements an error injection test mode.

The error correction is done using a lazy write back. When an error is detected, it is noted in a FIFO Queue which waits for an access gap to write the data back and refresh the memory. If a transaction writes new data to the compromised entry before the lazy write back completes, the write back is discarded.

18.7.4.7.2 ECC Aggregator

This section describes the functional details of the ECC Aggregator module.

18.7.4.7.2.1 ECC Aggregator Overview

The ECC Aggregator module supports the following general features:

- Provides a mechanism to control and monitor the ECC RAM in the MCAN module.
- Provides software access to all the ECC related registers.
- Supports software readable status of ECC single/double-bit errors and associated info such as RAM address and data bit(s) that are in error.
- Aggregates level pending status from the ECC RAM into a single interrupt to the Host CPU.

The following feature is not supported:

- Statistics such as tracking the number of single and double-bit errors. If needed, these operations can be handled by software.

18.7.4.7.2.2 ECC Aggregator Registers

There are 3 groups of registers in the ECC aggregator module:

- Global registers - Aggregator Revision Register (MCANSS_ECC_AGGR_REVISION), ECC Vector Register (MCANSS_ECC_VECTOR), Misc Status Register (MCANSS_ECC_MISC_STATUS), ECC Control Register (MCANSS_ECC_CONTROL), and ECC Wrapper Revision Register (MCANSS_ECC_WRAP_REVISION).
- Control and status registers - ECC Error Control Registers (MCANSS_ECC_ERR_CTRL1 and MCANSS_ECC_ERR_CTRL2) and ECC Error Status Registers (MCANSS_ECC_ERR_STAT1 and MCANSS_ECC_ERR_STAT2).
- Interrupt registers - interrupt status, interrupt enable set, interrupt enable clear and EOI (End Of Interrupt) registers that are part of a standard interrupt module. For more information, see the following registers:
  - MCANSS_ECC_SEC_EOI_REG
  - MCANSS_ECC_SEC_STATUS_REG0
  - MCANSS_ECC_SEC_ENABLE_SET_REG0
  - MCANSS_ECC_SEC_ENABLE_CLR_REG0
  - MCANSS_ECC_DED_EOI_REG
  - MCANSS_ECC_DED_STATUS_REG0
  - MCANSS_ECC_DED_ENABLE_SET_REG0
  - MCANSS_ECC_DED_ENABLE_CLR_REG0

18.7.4.7.2.3 Reads to ECC Control and Status Registers

The reads to the ECC control and status registers are triggered by writing a 'read message' to the ECC Vector Register as described below:

- Software writes value (the ECC RAM ID) to the MCANSS_ECC_VECTOR[10-0] ECC_VECTOR field to select the ECC RAM for control or status.

- Software writes 1 to the MCANSS_ECC_VECTOR[15] RD_SVBUS bit to trigger a read.
- Software writes read address to the MCANSS_ECC_VECTOR[23-16] RD_SVBUS_ADDRESS field.
- Software then polls the MCANSS_ECC_VECTOR[24] RD_SVBUS_DONE bit to check if it is 1. This bit indicates that the read operation has completed.
- Software reads the data from the ECC control or status register. The following clock cycle (MCAN_ICLK) returns the read data.

18.7.4.7.2.4 ECC Interrupts

The ECC aggregator module aggregates the level pending status from the ECC RAM into a single EOI-handshake based interrupt to the Host CPU. Software is expected to follow the sequence described below:

- Software enables the interrupts for the ECC RAM by writing to the MCANSS_ECC_SEC_ENABLE_SET_REG0/MCANSS_ECC_DED_ENABLE_SET_REG0 register.
- Software writes the ECC RAM ID in the MCANSS_ECC_VECTOR[10-0] ECC_VECTOR.
- Software writes the MCANSS_ECC_VECTOR[15] RD_SVBUS bit to trigger the read.
- Software writes the MCANSS_ECC_ERR_STAT1 register address to the MCANSS_ECC_VECTOR[23-16] RD_SVBUS_ADDRESS field. Software will need to load the 'read message' in the MCANSS_ECC_VECTOR register again if it needs to read the MCANSS_ECC_ERR_STAT2 register.
- Software polls the MCANSS_ECC_VECTOR[24] RD_SVBUS_DONE bit. When this bit is set, a read of the MCANSS_ECC_ERR_STAT1/MCANSS_ECC_ERR_STAT2 register is performed.
- After the interrupt has been serviced, software will clear the interrupt status by writing to the MCANSS_ECC_ERR_STAT1[8] CLR_ECC_SEC or MCANSS_ECC_ERR_STAT1[9] CLR_ECC_DED bit depending on the type of the ECC error.
- Software has to poll the MCANSS_ECC_ERR_STAT1 register to guarantee that the status bit has been cleared.
- Software will write to the MCANSS_ECC_SEC_EOI_REG/MCANSS_ECC_DED_EOI_REG register to clear the interrupt.
- After clearing the ECC interrupt source, the application software must also write 1 to the MCANSS_ECC_EOI[8] ECC_EOI bit.

18.7.4.8 Rx Handling

The Rx Handler controls the following operations:

- Acceptance filtering
- The transfer of received messages to the Rx Buffers or to one of the two Rx FIFOs (Rx FIFO 0 or Rx FIFO 1)
- Rx FIFO Put and Get Index operations

18.7.4.8.1 Acceptance Filtering

The MCAN module is capable to configure two sets of acceptance filters - one set for standard and one set for extended identifiers. These filters can be assigned to an Rx Buffer or to one of the two Rx FIFOs.

The main features of the filter elements are:

- Each filter element can be configured as:
  - Range Filter (from - to)
  - Filter for specific IDs (for one or two dedicated IDs)
  - Classic Bit Mask Filter
- Each filter element can be enabled/disabled individually
- Each filter element can be configured for acceptance or rejection filtering
- Filters are checked sequentially and execution (acceptance filtering procedure) stops at the first matching filter element or when the end of the filter list is reached

Related configuration registers are:

- Global Filter Configuration (MCAN_GFC) register
- Standard ID Filter Configuration (MCAN_SIDFC) register
- Extended ID Filter Configuration (MCAN_XIDFC) register
- Extended ID AND Mask (MCAN_XIDAM) register

Depending on the configuration of the filter element (see SFEC/EFEC in Section 18.7.4.11, *Message RAM*) if filter matches, one of the following actions is performed:

- Received frame is stored in FIFO 0 or FIFO 1
- Received frame is stored in Rx Buffer
- Received frame is stored in Rx Buffer and generation of pulse at filter event pin is performed. This is high level single MCAN_ICLK pulse. For more information, see Section 18.7.4.2.1, *DMA Requests*.
- Received frame is rejected
- Set High Priority Message interrupt flag MCAN_IR[8] HPM
- Set High Priority Message interrupt flag MCAN_IR[8] HPM and store received frame in FIFO 0 or FIFO 1

Acceptance filtering starts when complete Message ID is received. Acceptance filtering stops at the first matching enabled filter element or when the end of the filter list is reached. If a filter element matches - the Rx Handler starts writing the received message data in portions of 32 bit to the matching Rx Buffer or Rx FIFO. If an error condition occurs (for example: CRC error), this message is rejected with the following impact on the affected Rx Buffer or Rx FIFO:

- Rx Buffer:
  New Data flag (MCAN_NDAT1/MCAN_NDAT2) of matching Rx Buffer is not set, but Rx Buffer (partly) overwritten with received data (for error type see MCAN_PSR[2:0] LEC respectively MCAN_PSR[10:8] DLEC fields).
- Rx FIFO:
  Put index of matching Rx FIFO is not updated, but related Rx FIFO element (partly) overwritten with received data (for error type see MCAN_PSR[2:0] LEC respectively MCAN_PSR[10:8] DLEC fields). If matching Rx FIFO is configured to operate in overwrite mode, the boundary conditions described in Section 18.7.4.8.2.2 have to be considered.

18.7.4.8.1.1 Range Filter

Each filter element can be configured to operate as Range Filter (Standard Filter Type SFT = 00/Extended Filter Type EFT = 00). The filter matches for all received message frames with IDs in the range from SFID1 to SFID2 (SFID2 ≥ SFID1) respectively in the range from EFID1 to EFID2 (EFID2 ≥ EFID1). For more information see Section 18.7.4.11.5, *Standard Message ID Filter Element* and Section 18.7.4.11.6, *Extended Message ID Filter Element*.

There are two options for range filtering of extended frames:

- Extended Filter Type EFT = 00: The Extended ID AND Mask (MCAN_XIDAM) is used for Range Filtering. The Message ID of received frames is ANDed with the Extended ID AND Mask (MCAN_XIDAM) before the range filter is applied.
- Extended Filter Type EFT = 11: The Extended ID AND Mask (MCAN_XIDAM) is not used for Range Filtering.

18.7.4.8.1.2 Filter for specific IDs

Each filter element can be configured to filter one or two dedicated Message IDs (Standard Filter Type SFT =01/Extended Filter Type EFT =01). To filter only one specific Message ID, the filter element has to be configured with SFID1 = SFID2 respectively EFID1 = EFID2. For more information see Section 18.7.4.11.5, *Standard Message ID Filter Element* and Section 18.7.4.11.6, *Extended Message ID Filter Element*.

18.7.4.8.1.3 Classic Bit Mask Filter

Classic bit mask filtering can filter groups of Message IDs (Standard Filter Type SFT =10/Extended Filter Type EFT =10). This is done by masking single bits of a received Message ID. In this case SFID1/EFID1 element is used as Message ID filter, while SFID2/EFID2 element is used as filter mask.

A 0 bit at the filter mask (SFID2/EFID2) will mask out the corresponding bit position of the configured Message ID filter (SFID1/EFID1) and the value of the received Message ID at that bit position is not relevant for acceptance filtering. Only those bits of the received Message ID where the corresponding mask bits are 1 are relevant for acceptance filtering.

There are two interesting cases:

- All mask bits are 1: a match occurs only when the received Message ID and the configured Message ID filter are identical.
- All mask bits are 0: all Message IDs match.

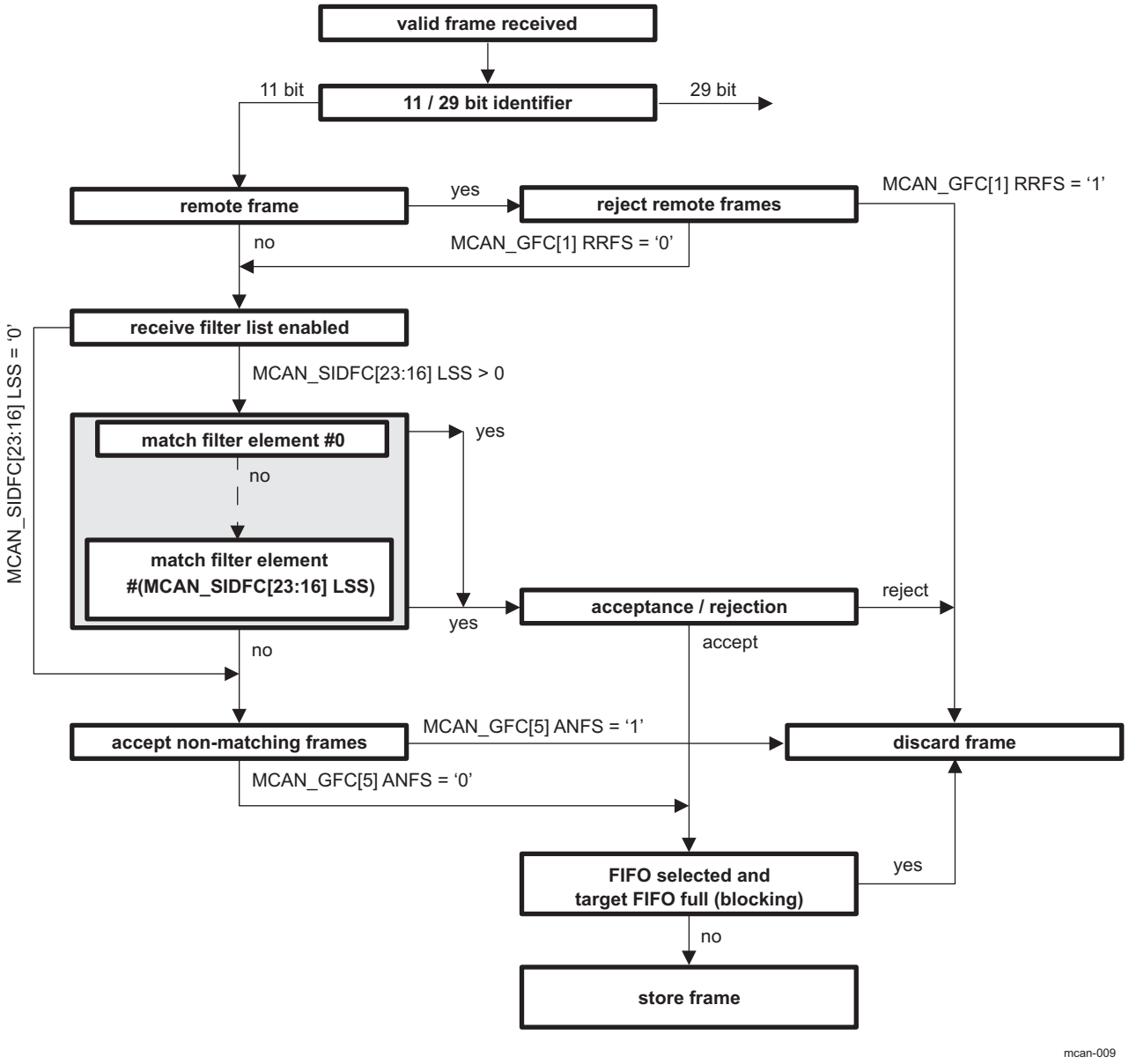
18.7.4.8.1.4 Standard Message ID Filtering

The standard Message ID (11-bit ID) filtering flow is shown in Figure 18-142. Section 18.7.4.11.5, *Standard Message ID Filter Element* describes the standard Message ID filter element.

The Remote Transmission Request (RTR) and Extended Identifier (XTD) bits of the received frames are compared against the list of configured filter elements. This is controlled by the following registers:

- Global Filter Configuration (MCAN_GFC) register
- Standard ID Filter Configuration (MCAN_SIDFC) register

Figure 18-142. Standard Message ID Filter Path



18.7.4.8.1.5 Extended Message ID Filtering

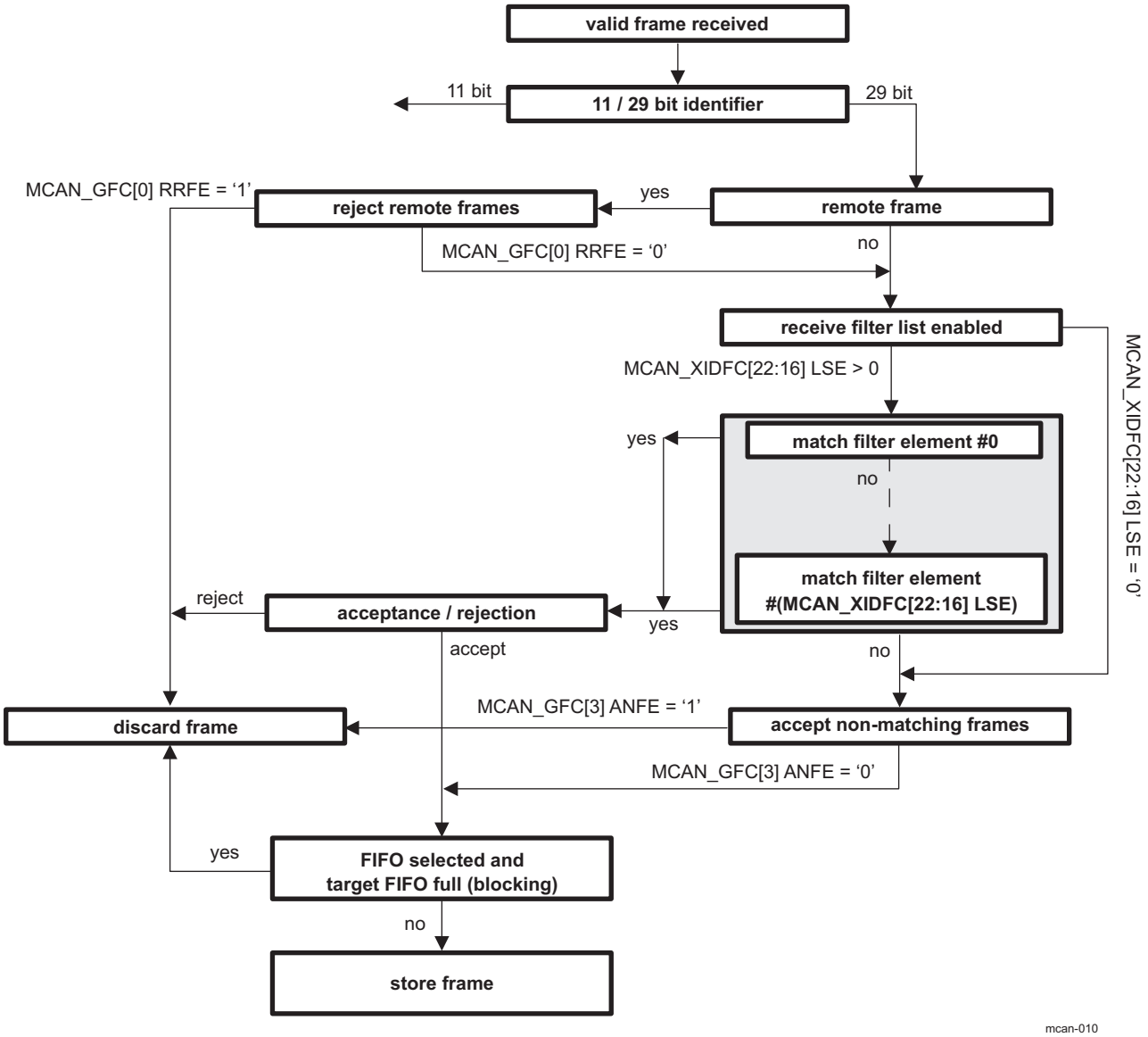
The extended Message ID (29-bit ID) filtering flow is shown in Figure 18-143. Section 18.7.4.11.6, *Extended Message ID Filter Element* describes the extended Message ID filter element.

The Remote Transmission Request (RTR) and Extended Identifier (XTD) bits of the received frames are compared against the list of configured filter elements. This is controlled by the following registers:

- Global Filter Configuration (MCAN_GFC) register
- Extended ID Filter Configuration (MCAN_XIDFC) register

Note that before the filter list is executed the received identifier is ANDed with the Extended ID AND Mask (MCAN_XIDAM).

Figure 18-143. Extended Message ID Filter Path



18.7.4.8.2 Rx FIFOs

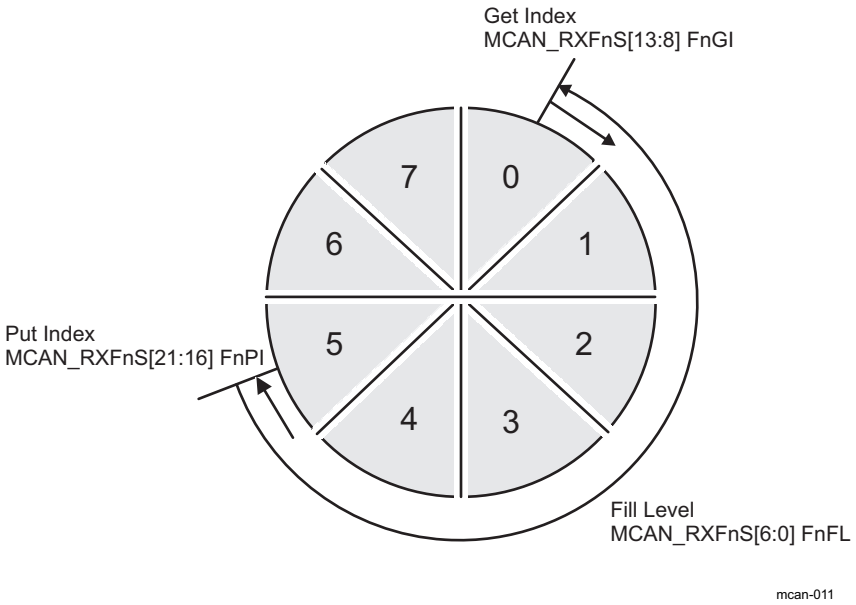
The configuration of the Rx FIFOs (Rx FIFO 0 and Rx FIFO 1) can be done via the MCAN_RXF0C and MCAN_RXF1C registers. Each Rx FIFO can be configured to store up to 64 received messages.

After acceptance filtering the received messages that passed are transferred to the Rx FIFO. The filter mechanisms available for the Rx FIFO 0 and Rx FIFO 1 is described in Section 18.7.4.8.1, *Acceptance Filtering*. Section 18.7.4.11.2, *Rx Buffer and FIFO Element* describes the Rx FIFO element.

The Rx FIFO watermark can be used to prevent an Rx FIFO overflow. If the Rx FIFO fill level reaches the Rx FIFO watermark configured by the MCAN_RXFnC[30:24] FnWM filed (where: n = 0 or 1) an interrupt flag MCAN_IR[1] RF0W/MCAN_IR[5] RF1W is set.

When the Rx FIFO Put Index reaches the Rx FIFO Get Index (MCAN_RXFnS[21:16] FnPI = MCAN_RXFnS[13:8] FnGI) an Rx FIFO Full condition is signalled by the MCAN_RXFnS[24] FnF status bit and interrupt flag MCAN_IR[2] RF0F/MCAN_IR[6] RF1F is set. Figure 18-144 shows Rx FIFO Status. The FIFOs fill level is presented in the MCAN_RXFnS[6:0] FnFL field (the number of elements stored in Rx FIFO).

Figure 18-144. Rx FIFO Status



Rx FIFOs start address in the Message RAM (MCAN_RXFnC[15:2]FnSA field) have to be configured when reading from an Rx FIFO (Rx FIFO Get Index - MCAN_RXFnS[13:8] FnGI). Table 18-617 presents Rx Buffer/Rx FIFO Element Size for different Rx Buffer / Rx FIFO Data Field Size which is configured via the MCAN_RXESC register.

Table 18-617. Rx Buffer/Rx FIFO Element Size

MCAN_RXESC[10:8] RBDS MCAN_RXESC[2:0] F0DS/MCAN_RXESC[6:4] F1DS	Data Field [bytes]	FIFO Element Size [RAM words]
000	8	4
001	12	5
010	16	6
011	20	7
100	24	8
101	32	10
110	48	14
111	64	18

18.7.4.8.2.1 Rx FIFO Blocking Mode

The Rx FIFO blocking mode is the default operation mode for the Rx FIFOs. It is configured by the MCAN_RXFnC[31] FnOM = 0.

If an Rx FIFO full condition is reached (MCAN_RXFnS[21:16] FnPI = MCAN_RXFnS[13:8] FnGI), no further messages are written to the corresponding Rx FIFO until at least one message has been read out and the Rx FIFO Get Index has been incremented. An Rx FIFO full condition is signalled by the MCAN_RXFnS[24] FnF = 1 and interrupt flag MCAN_IR[2] RF0F/MCAN_IR[6] RF1F is set.

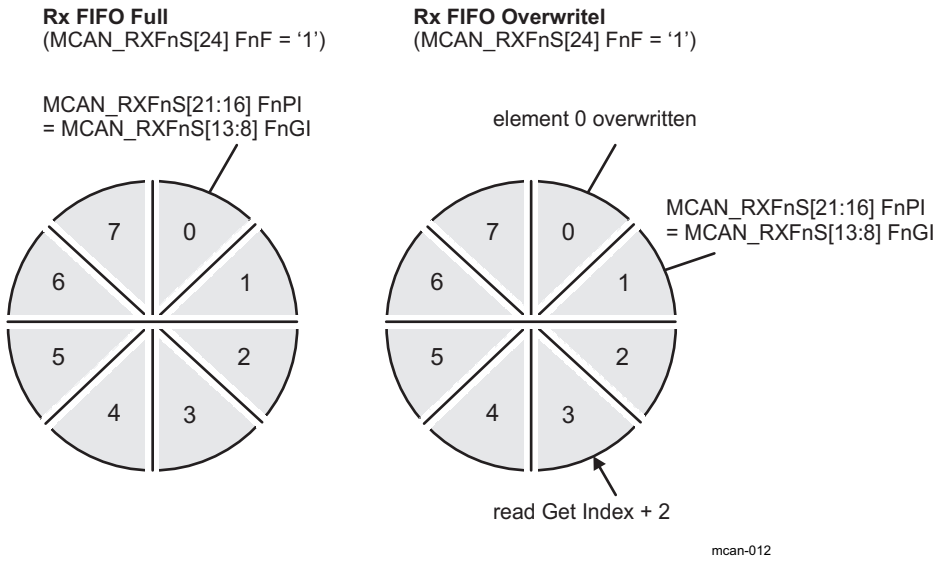
In case a message is received while the corresponding Rx FIFO is full, this message is rejected and the message lost condition is signalled by MCAN_RXFnS[25] RFnL = 1 and interrupt flag MCAN_IR[3] RFnL/MCAN_IR[25] RFnL is set.

18.7.4.8.2.2 Rx FIFO Overwrite Mode

The Rx FIFO overwrite mode is configured by the MCAN_RXFnC[31] FnOM = 1. When an Rx FIFO full condition is reached (MCAN_RXFnS[21:16] FnPI = MCAN_RXFnS[13:8] FnGI) signalled by MCAN_RXFnS[24] FnF = 1, the next accepted message for the FIFO will overwrite the oldest FIFO message. Put index/Get index are both incremented by one.

In overwrite mode if an Rx FIFO full condition is signalled, reading of the Rx FIFO elements should start at least at get index + 1. The reason for that is, that it might happen, that a received message is written to the Message RAM (Put index) while the Host CPU is reading from the Message RAM (Get index). In this case inconsistent data may be read from the respective Rx FIFO element. The problem is solved by adding an offset to the Get index when reading from the Rx FIFO. The offset depends on how fast the Host CPU accesses the Rx FIFO. Figure 18-145 shows an offset of two with respect to the Get index when reading the Rx FIFO. In this case the two messages stored in element 1 and 2 are lost.

Figure 18-145. Rx FIFO Overflow Handling



After reading from the Rx FIFO, the number of the last element read has to be written to the Rx FIFO Acknowledge Index MCAN_RXFnA[5:0] FnAI. This increments the get index to that element number. In case the Put index has not been incremented to this Rx FIFO element, the Rx FIFO full condition is reset (MCAN_RXFnS[24] FnF = 0).

18.7.4.8.3 Dedicated Rx Buffers

The MCAN supports up to 64 dedicated Rx Buffers. The start address of the Rx Buffers section in the Message RAM is configured via MCAN_RXBC[15:2] RBSA field. To store in an Rx Buffer a Standard or Extended Message ID Filter Element with SFEC/EFEC = 111 and SFID2/EFID2[10:9] = 00 has to be configured (see Section 18.7.4.11.5, *Standard Message ID Filter Element* and Section 18.7.4.11.6, *Extended Message ID Filter Element*).

After a received message has been accepted by a filter element, the message is stored into the Rx Buffer in the Message RAM referenced by the filter element (the format is the same as for an Rx FIFO element). In addition the flag MCAN_IR[19] DRX (Message stored in Dedicated Rx Buffer) is set.

Table 18-618 shows Example Filter Configuration for Rx Buffers.

Table 18-618. Example Filter Configuration for Rx Buffers

Filter Element	SFID1[10:0] EFID1[28:0]	SFID2[10:9] EFID2[10:9]	SFID2[5:0] EFID2[5:0]
0	ID message 1	00	00 0000
1	ID message 2	00	00 0001
2	ID message 3	00	00 0010

After the last word of a matching received message has been written to the Message RAM, the respective New Data flag in register MCAN_NDAT1/MCAN_NDAT2 is set. As long as the New Data flag is set, the respective Rx Buffer is locked against updates from received matching frames. The New Data flags have to be reset by the Host CPU by writing a 1 to the respective bit position.

While an Rx Buffer's New Data flag is set, a Message ID Filter Element referencing this specific Rx Buffer will not match, causing the acceptance filtering to continue. Following Message ID Filter Elements may cause the received message to be stored into another Rx Buffer, or into an Rx FIFO, or the message may be rejected, depending on filter configuration.

18.7.4.8.3.1 Rx Buffer Handling

Rx Buffer Handling include the following steps:

- Reset interrupt flag MCAN_IR[19] DRX
- Read New Data registers
- Read messages from Message RAM
- Reset New Data flags of processed messages

18.7.4.9 Tx Handling

The Tx Handler is used to handle the Tx requests. It controls the transfer of transmit messages from the dedicated Tx Buffers, the Tx FIFO, and the Tx Queue to the CAN Core, the Tx Event FIFO, and the Put and Get Index operations. The MCAN module supports up to 32 Tx Buffers. These Tx Buffers can be configured as dedicated Tx Buffers, Tx FIFO, or Tx Queue and as combination of dedicated Tx Buffers/Tx FIFO or dedicated Tx Buffers/Tx Queue. For each Tx Buffer element Classical CAN or CAN FD transmission mode can be configured. Section 18.7.4.11.3 describes the Tx Buffer Element. Table 18-619 shows the possible configurations for message transmission.

Table 18-619. Possible Configurations for Message Transmission

MCAN_CCCR		Tx Buffer Element		Frame Transmission
MCAN_CCCR[9] BRSE	MCAN_CCCR[8] FDOE	FDF	BRS	
ignored	0	ignored	ignored	Classic CAN
0	1	0	ignored	Classic CAN
0	1	1	ignored	CAN FD without bit rate switching
1	1	0	ignored	Classic CAN
1	1	1	0	CAN FD without bit rate switching
1	1	1	1	CAN FD with bit rate switching

When the Tx Buffer Request Pending MCAN_TXBRP register is updated, or when a transmission has been started the Tx Handler starts scanning to check for the highest priority pending Tx request. The Tx Buffer with lowest Message ID has highest priority.

NOTE: AUTOSAR requires at least three Tx Queue Buffers and support of transmit cancellation.

18.7.4.9.1 Transmit Pause

The transmit pause feature is intended for use in CAN networks where the CAN Message IDs are specific and cannot easily be changed. These Message IDs may have a higher priority than other defined Message IDs, while in a specific application their relative priority should be inverse. This allows for a case where one ECU sends a burst of CAN messages that cause another ECU's CAN messages to be delayed (paused).

The transmit pause feature is enabled by the MCAN_CCCR[14] TXP bit. By default this bit is disabled (MCAN_CCCR[14] TXP = 0). Each time after successfully transmitted message, a pause for two CAN bit times occurs before the start of the next transmission. This allows the other CAN nodes in the network to transmit messages even if their Message IDs have lower priority.

18.7.4.9.2 Dedicated Tx Buffers

Dedicated Tx Buffers are intended for message transmission under complete control of the Host CPU.

There are two options:

- Each dedicated Tx Buffer is configured with a specific Message ID.
- Two or more dedicated Tx Buffers are configured with the same Message ID. In this case the Tx Buffer with the lowest buffer number is transmitted first.

After the data section has been updated, a transmission is requested by an Add Request. This is done via the MCAN_TXBAR[x]ARn bit (where x = 0 - 31). The requested messages arbitrate internally with messages from an optional Tx FIFO or Tx Queue and externally with messages on the CAN bus, and are sent out according to their Message ID.

Table 18-620 shows Tx Buffer/Tx FIFO/Tx Queue Element Size. A Dedicated Tx Buffer allocates Element Size 32-bit words in the Message RAM. The start address of a dedicated Tx Buffer in the Message RAM is calculated by adding transmit buffer index from 0 to 31 (MCAN_TXFQS[20:16] TFQPI) × Element Size to the Tx Buffer Start Address MCAN_TXBC[15:2] TBSA field.

Table 18-620. Tx Buffer/Tx FIFO/Tx Queue Element Size

MCAN_TXESC[2:0] TBDS	Data Field [bytes]	Element Size [RAM words]
000	8	4
001	12	5
010	16	6
011	20	7
100	24	8
101	32	10
110	48	14
111	64	18

18.7.4.9.3 Tx FIFO

Tx FIFO mode is configured by setting bit MCAN_TXBC[30] TFQM = 0. The stored in the Tx FIFO messages are transmitted starting with the message referenced by the Get Index MCAN_TXFQS[12:8] TFGI field. After each transmission the Get Index is incremented until the Tx FIFO is empty. The Tx FIFO Free Level MCAN_TXFQS[5:0] TFFL field indicates the number of the available free Tx FIFO elements. The Tx FIFO allows transmission of messages with the same Message ID from different Tx Buffers in the order these messages have been written to the Tx FIFO.

New transmit messages have to be written to the Tx FIFO starting with the Tx Buffer referenced by the Put Index MCAN_TXFQS[20:16] TFQPI field. After each Add Request (MCAN_TXBAR[x] ARn = 1) the Put Index is incemented to the next free Tx FIFO element. When the Put Index reaches the Get Index (MCAN_TXFQS[20:16] TFQPI = MCAN_TXFQS[12:8] TFGI), Tx FIFO Full condition is signalled by bit MCAN_TXFQS[21] TFQF = 1. In this case no further messages should be written to the Tx FIFO until the next message has been transmitted and the Get Index has been incremented.

The number of requested Tx buffers should not exceed the number of free Tx Buffers as indicated by the Tx FIFO Free Level MCAN_TXFQS[5:0] TFFL field.

In case a transmission request for the Tx Buffer referenced by the Get Index is cancelled, the Get Index is incremented to the next Tx Buffer with pending transmission request and the Tx FIFO Free Level MCAN_TXFQS[5:0] TFFL field is recalculated. In case transmission cancellation is applied to any other Tx Buffer - the Get Index and the FIFO Free Level remain unchanged.

A Tx FIFO element allocates Element Size 32-bit words in the Message RAM (see Table 18-620). The start address of the next available (free) Tx FIFO Buffer is calculated by adding Tx FIFO/Queue Put Index MCAN_TXFQS[20:16] TFQPI (from 0 to 31) × Element Size to the Tx Buffer Start Address MCAN_TXBC[15:2] TBSA field.

18.7.4.9.4 Tx Queue

Tx Queue mode is configured by setting bit MCAN_TXBC[30] TFQM = 1. The stored in the Tx Queue messages are transmitted starting with the highest priority message (lowest Message ID). In case two or more Queue Buffers are configured with the same Message ID, the Queue Buffer with the lowest buffer number is transmitted first.

New transmit messages have to be written to the Tx FIFO starting with the Tx Buffer referenced by the Put Index MCAN_TXFQS[20:16] TFQPI field. Each Add Request cyclically increments the Put Index to the next free Tx Buffer. In case of Tx Queue Full condition (MCAN_TXFQS[21] TFQF = 1), the Put Index is not valid and no further message should be written to the Tx Queue until at least one of the requested messages has been sent out or a pending transmission request has been cancelled.

The application may use the MCAN_TXBRP register instead of the Put Index and may place messages to any Tx Buffer without pending transmission request.

A Tx Queue Buffer allocates Element Size 32-bit words in the Message RAM (see Table 18-620). The start address of the next available (free) Tx Queue Buffer is calculated by adding Tx FIFO/Queue Put Index MCAN_TXFQS[20:16] TFQPI (from 0 to 31) × Element Size to the Tx Buffer Start Address MCAN_TXBC[15:2] TBSA field.

18.7.4.9.5 Mixed Dedicated Tx Buffers/Tx FIFO

For this combination the Tx Buffers section in the Message RAM is separated in two parts:

- Dedicated Tx Buffers: the number of Dedicated Tx Buffers is configured by the MCAN_TXBC[21:16] NDTB field
- Tx FIFO: the number of Tx Buffers assigned to the Tx FIFO is configured by the MCAN_TXBC[29:24] TFQS field

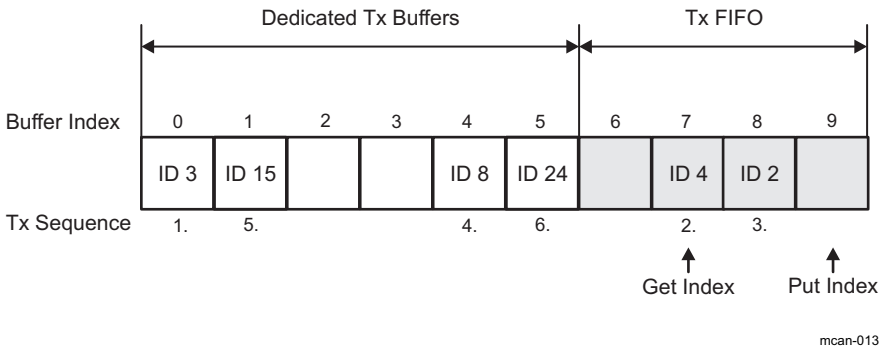
If the MCAN_TXBC[29:24] TFQS field is empty (zero) - only Dedicated Tx Buffers are used.

Tx prioritization:

- Scan Dedicated Tx Buffers and oldest pending Tx FIFO Buffer (referenced by the MCAN_TXFQS[12:8] TFGI field)
- Buffer with lowest Message ID gets highest priority and is transmitted next

Figure 18-146 shows Mixed Dedicated Tx Buffers/Tx FIFO example.

Figure 18-146. Mixed Dedicated Tx Buffers /Tx FIFO (example)



18.7.4.9.6 Mixed Dedicated Tx Buffers/Tx Queue

For this combination the Tx Buffers section in the Message RAM is separated in two parts:

- Dedicated Tx Buffers: the number of Dedicated Tx Buffers is configured by the MCAN_TXBC[21:16] NDTB field
- Tx Queue: the number of Tx Buffers assigned to the Tx Queue is configured by the MCAN_TXBC[29:24] TFQS field

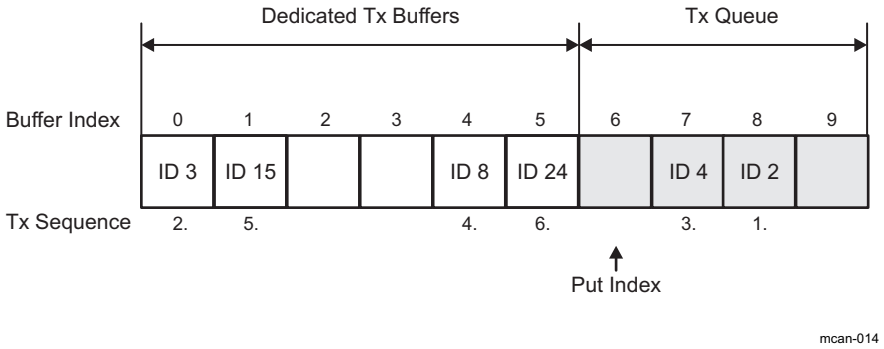
If MCAN_TXBC[29:24] TFQS field is empty (zero) - only Dedicated Tx Buffers are used.

Tx prioritization:

- Scan all Tx Buffers with activated transmission request
- Tx Buffer with lowest Message ID gets highest priority and is transmitted next

Figure 18-147 shows Mixed Dedicated Tx Buffers/Tx Queue example.

Figure 18-147. Mixed Dedicated Tx Buffers /Tx Queue (example)



18.7.4.9.7 Transmit Cancellation

This feature is especially intended for gateway and AUTOSAR based applications. The Host CPU can cancel a requested transmission from a dedicated Tx Buffer or a Tx Queue Buffer by setting bit MCAN_TXBCR[n] CRn = 1 (where n = 0 - 31). The corresponding bit position n is equivalent to the number of the Tx Buffer.

Transmit cancellation is not intended for Tx FIFO operation.

Successful cancellation is signalled by setting the corresponding bit of the MCAN_TXBCF register (MCAN_TXBCF[n] CFn = 1).

If transmission from a Tx Buffer is already ongoing and a transmit cancellation is requested, the corresponding MCAN_TXBRP[n] TRPn bit remains set as long as the transmission is in progress. If the transmission was successful, the corresponding MCAN_TXBTO[n] TOn and MCAN_TXBCF[n] CFn bits are set. If the transmission was not successful, only the corresponding bit MCAN_TXBCF[n] CFn = 1.

NOTE: If pending transmission is cancelled immediately before this transmission could have been started, a short time window occurs where no transmission is started even if another message is also pending in this node. This may enable another node to transmit a message which may have a lower priority than the second message in this node.

18.7.4.9.8 Tx Event Handling

To support Tx Event Handling the Message RAM has implemented a Tx Event FIFO section. Up to 32 Tx Event FIFO elements can be configured. Section 18.7.4.11.4 describes the Tx Event FIFO element. After message transmission on the CAN bus, Message ID and Timestamp are stored in a Tx Event FIFO element. To link a Tx Event to a Tx Event FIFO element, the Message Marker from the transmitted Tx Buffer is copied into the Tx Event FIFO element.

A Tx Event FIFO full condition is signalled by the MCAN_IR[14] TEFF bit. In this case no further elements are written to the Tx Event FIFO until at least one element has been read out and the Tx Event FIFO Get Index has been incremented (MCAN_TXEFS[12:8] EFGI). In case a Tx Event occurs while the Tx Event FIFO is full, this event is rejected and interrupt flag MCAN_IR[15] TEFL bit is set.

The Tx Event FIFO watermark can be configured to avoid a Tx Event FIFO overflow. When the Tx Event FIFO fill level reaches the Tx Event FIFO watermark configured by the MCAN_TXEFC[29:24] EFWM field, interrupt flag MCAN_IR[13] TEFW is set. When reading from the Tx Event FIFO, two times the Tx Event FIFO Get Index MCAN_TXEFS[12:8] EFGI field has to be added to the Tx Event FIFO start address MCAN_TXEFC[15:2] EFSA field.

18.7.4.10 FIFO Acknowledge Handling

The Get Indices of the two Rx FIFOs (Rx FIFO 0 or Rx FIFO 1) and the Tx Event FIFO are controlled by writing to the corresponding FIFO Acknowledge Index (see MCAN_RXF0A, MCAN_RXF1A, and MCAN_TXEFA). Writing to the FIFO Acknowledge Index will set the FIFO Get Index to the FIFO Acknowledge Index plus one and thereby updates the FIFO Fill Level.

There are two use cases:

- A single element has been read from the FIFO: the Get Index value is written to the FIFO Acknowledge Index.
- A sequence of elements has been read from the FIFO: the Get Index value (Index of the last element read) is written to the FIFO Acknowledge Index at the end of that read sequence.

The Host CPU has free access to the Message RAM. The special care has to be taken when reading FIFO elements in an arbitrary order (Get Index not considered). This can be useful when reading a High Priority Message from one of the two Rx FIFOs. In this case the FIFO's Acknowledge Index should not be written because this would set the Get Index to a wrong position and also changes the FIFO's Fill Level. In this case some of the older FIFO elements would be lost.

NOTE: The application has to ensure that a valid value is written to the FIFO Acknowledge Index. The MCAN module does not check for erroneous values.

18.7.4.11 Message RAM

The MCAN module has implemented Message RAM. The main purpose of the Message RAM is to store:

- Receive Messages
- Transmit Messages
- Tx Event Elements
- Message ID Filter Elements

18.7.4.11.1 Message RAM Configuration

The MCAN module is configured to allocate 1600 words in the Message RAM. The Message RAM has a width of 32 bits.

The address range of the Message RAM is from to .

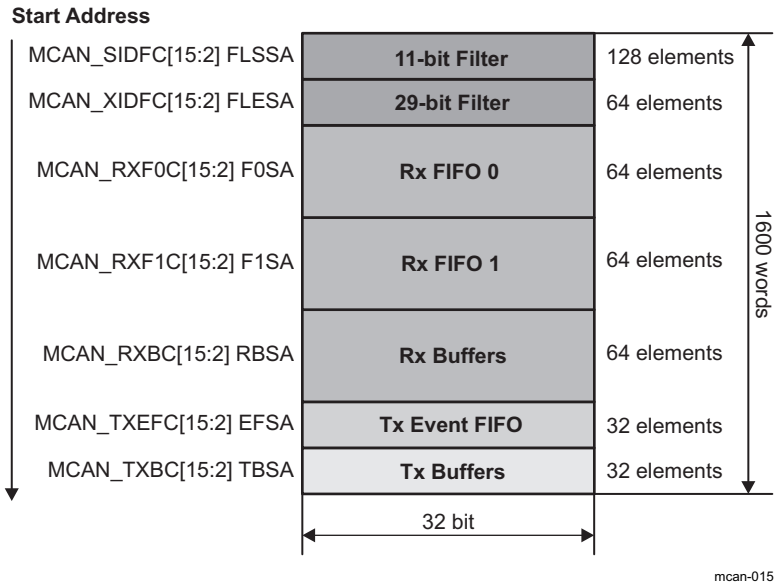
The Message RAM is capable to include each of the sections listed in Figure 18-148. It is not necessary to configure each of the sections (a section in the Message RAM may be 0) and there is not restriction with respect to the sequence of the sections. For parity checking or ECC a respective number of bits has to be added to each word.

When the MCAN module addresses the Message RAM it addresses 32-bit words. The start addresses are configurable and they are 32-bit word addresses.

The element size can be configured for:

- Rx FIFO 0 via the MCAN_RXESC[2:0] F0DS field
- Rx FIFO 1 via the MCAN_RXESC[6:4] F1DS field
- Rx Buffers via the MCAN_RXESC[10:8] RBDS field
- Tx Buffers via the MCAN_TXESC[2:0] TBDS field

Figure 18-148. Message RAM Configuration



The Host CPU configures the following information in the Message RAM:

- Start addresses of the memory sections
- Number of elements in each section
- The size of the elements in some sections

NOTE: The MCAN module does not check for errors in the Message RAM configuration. The configuration of the start addresses of the different sections and the number of elements of each section has to be done carefully. This will prevent falsification or loss of data.

18.7.4.11.2 Rx Buffer and FIFO Element

Up to 64 Rx Buffers and two Rx FIFOs can be configured in the Message RAM. Each Rx FIFO section can be configured to store up to 64 received messages. The element size can be configured for storage of CAN FD messages with up to 64 bytes data field via the MCAN_RXESC register.

Figure 18-149 shows Rx Buffer/Rx FIFO element structure.

Figure 18-149. Rx Buffer/Rx FIFO Element Structure

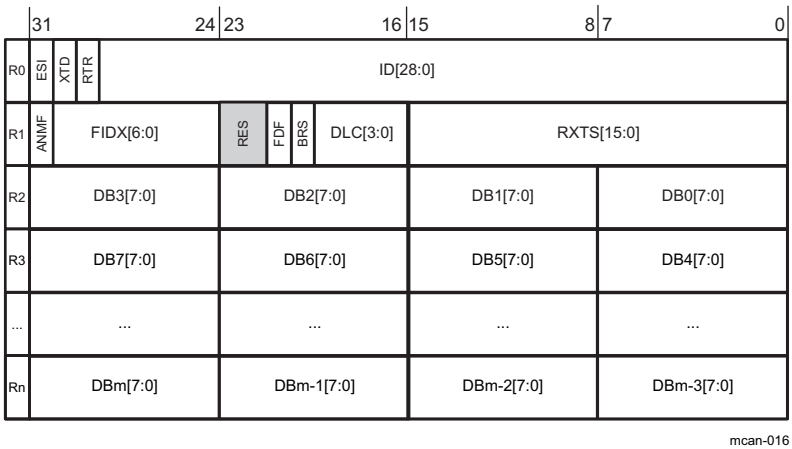


Table 18-621 shows Rx Buffer/Rx FIFO element field descriptions.

Table 18-621. Rx Buffer/Rx FIFO Element Field Descriptions

Word	Bits	Field Name	Description
R0	31	ESI	Error State Indicator • 0x0: Transmitting node is error active • 0x1: Transmitting node is error passive
	30	XTD	Extended Identifier Signals to the Host CPU whether the received frame has a standard or extended identifier. • 0x0: 11-bit standard identifier • 0x1: 29-bit extended identifier
	29	RTR	Remote Transmission Request Signals to the Host CPU whether the received frame is a data frame or a remote frame. • 0x0: Received frame is a data frame • 0x1: Received frame is a remote frame **Note:** There are no remote frames in CAN FD format. In case a CAN FD frame was received (FDF = 1), RTR bit reflects the state of the reserved r1 bit (RES[23]).
	28:0	ID[28:0]	Identifier Standard or extended identifier depending on XTD bit. A standard identifier is stored into ID[28:18].
R1	31	ANMF	Accepted Non-matching Frame Acceptance of non-matching frames may be enabled via the MCAN_GFC[5:4] ANFS and MCAN_GFC[3:2] ANFE fields. • 0x0: Received frame matching filter index FIDX field • 0x1: Received frame did not match any Rx filter element
	30:24	FIDX[6:0]	Filter Index 0x0-0x7F (0-127): Index of matching Rx acceptance filter element (invalid if ANMF = 1). Range is 0 to MCAN_SIDFC[23:16] LSS - 1 respectively MCAN_XIDFC[22:16] LSE - 1.
	23:22	RES	Reserved
	21	FDF	FD Format • 0x0: Standard frame format • 0x1: CAN FD frame format (new DLC-coding and CRC)
	20	BRS	Bit Rate Switch • 0x0: Frame received without bit rate switching • 0x1: Frame received with bit rate switching
	19:16	DLC[3:0]	Data Length Code • 0x0-0x8 (0-8): CAN + CAN FD: received frame has 0-8 data bytes • 0x9-0xF (9-15): CAN: received frame has 8 data bytes • 0x9-0xF (9-15): CAN FD: received frame has 12/16/20/24/32/48/64 data bytes
	15:0	RXTS[15:0]	Rx Timestamp Timestamp Counter value captured on start of frame reception. Resolution depending on configuration of the Timestamp Counter Prescaler MCAN_TSCC[19:16] TCP.
R2	31:24	DB3[7:0]	Data Byte 3
	23:16	DB2[7:0]	Data Byte 2
	15:8	DB1[7:0]	Data Byte 1
	7:0	DB0[7:0]	Data Byte 0

Table 18-621. Rx Buffer/Rx FIFO Element Field Descriptions (continued)

Word	Bits	Field Name	Description
R3	31:24	DB7[7:0]	Data Byte 7
	23:16	DB6[7:0]	Data Byte 6
	15:8	DB5[7:0]	Data Byte 5
	7:0	DB4[7:0]	Data Byte 4
...	...	...	...
Rn	31:24	DBm[7:0]	Data Byte m
	23:16	DBm-1[7:0]	Data Byte m-1
	15:8	DBm-2[7:0]	Data Byte m-2
	7:0	DBm-3[7:0]	Data Byte m-3

Note: Depending on the configuration of the element size (MCAN_RXESC), between two and sixteen 32-bit words (Rn = 3-17) are used for storage of a CAN message's data field.

18.7.4.11.3 Tx Buffer Element

The Tx Buffers section can be configured to hold dedicated Tx Buffers as well as a Tx FIFO/Tx Queue. In case that the Tx Buffers section is shared by dedicated Tx buffers and a Tx FIFO/Tx Queue, the dedicated Tx Buffers start at the beginning of the Tx Buffers section followed by the buffers assigned to the Tx FIFO or Tx Queue. The Tx Handler makes difference between dedicated Tx Buffers and Tx FIFO/Tx Queue via the MCAN_TXBC[29:24] TFQS and MCAN_TXBC[21:16] NDTB fields. The element size can be configured for storage of CAN FD messages with up to 64 bytes data field via the MCAN_TXESC register.

Figure 18-150 shows Tx Buffer element structure.

Figure 18-150. Tx Buffer Element Structure

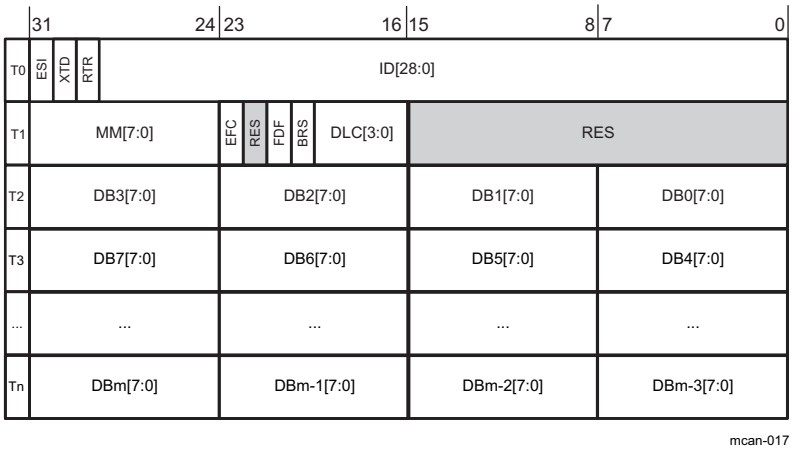


Table 18-622 shows Tx Buffer element field descriptions.

Table 18-622. Tx Buffer Element Field Descriptions

Word	Bits	Field Name	Description
T0	31	ESI	Error State Indicator • 0x0: ESI bit in CAN FD format depends only on error passive flag • 0x1: ESI bit in CAN FD format transmitted recessive **Note:** The ESI bit of the transmit buffer is or'ed with the error passive flag to decide the value of the ESI bit in the transmitted CAN FD frame. As required by the CAN FD protocol specification, an error active node may optionally transmit the ESI bit recessive, but an error passive node will always transmit the ESI bit recessive.
	30	XTD	Extended Identifier • 0x0: 11-bit standard identifier • 0x1: 29-bit extended identifier
	29	RTR	Remote Transmission Request • 0x0: Transmit data frame • 0x1: Transmit remote frame **Note:** When RTR = 1, the MCAN module transmits a remote frame according to ISO11898-1:2015, even if the MCAN_CCCR[8] FDOE bit enables the transmission in CAN FD format.
	28:0	ID[28:0]	Identifier Standard or extended identifier depending on XTD bit. A standard identifier has to be written to ID[28:18].
T1	31:24	MM[7:0]	Message Marker Written by Host CPU during Tx Buffer configuration. Copied into Tx Event FIFO element for identification of Tx message status (see also MM[7:0] field in Table 18-623).
	23	EFC	Event FIFO Control • 0x0: Don't store Tx events • 0x1: Store Tx events
	22	RES	Reserved
	21	FDF	FD Format • 0x0: Frame transmitted in Classic CAN format • 0x1: Frame transmitted in CAN FD format
	20	BRS	Bit Rate Switch • 0x0: CAN FD frames transmitted without bit rate switching • 0x1: CAN FD frames transmitted with bit rate switching **Note:** ESI, FDF, and BRS bits are only evaluated when CAN FD operation is enabled vie the MCAN_CCCR[8] FDOE bit. BRS bit is only evaluated when in addition the MCAN_CCCR[9] BRSE = 1.
	19:16	DLC[3:0]	Data Length Code • 0x0-0x8 (0-8): CAN + CAN FD: transmit frame has 0-8 data bytes • 0x9-0xF (9-15): CAN: transmit frame has 8 data bytes • 0x9-0xF (9-15): CAN FD: transmit frame has 12/16/20/24/32/48/64 data bytes
	15:0	RES	Reserved
T2	31:24	DB3[7:0]	Data Byte 3
	23:16	DB2[7:0]	Data Byte 2
	15:8	DB1[7:0]	Data Byte 1
	7:0	DB0[7:0]	Data Byte 0

Table 18-622. Tx Buffer Element Field Descriptions (continued)

Word	Bits	Field Name	Description
T3	31:24	DB7[7:0]	Data Byte 7
	23:16	DB6[7:0]	Data Byte 6
	15:8	DB5[7:0]	Data Byte 5
	7:0	DB4[7:0]	Data Byte 4
...	...	...	...
Tn	31:24	DBm[7:0]	Data Byte m
	23:16	DBm-1[7:0]	Data Byte m-1
	15:8	DBm-2[7:0]	Data Byte m-2
	7:0	DBm-3[7:0]	Data Byte m-3

Note: Depending on the configuration of the element size (MCAN_TXESC), between two and sixteen 32-bit words (Tn = 3-17) are used for storage of a CAN message's data field.

18.7.4.11.4 Tx Event FIFO Element

Each element stores information about transmitted messages. By reading the Tx Event FIFO the Host CPU gets this information in the order the messages were transmitted. Status information about the Tx Event FIFO can be obtained from the MCAN_TXEFS register.

Figure 18-151 shows Tx Event FIFO element structure.

Figure 18-151. Tx Event FIFO Element Structure

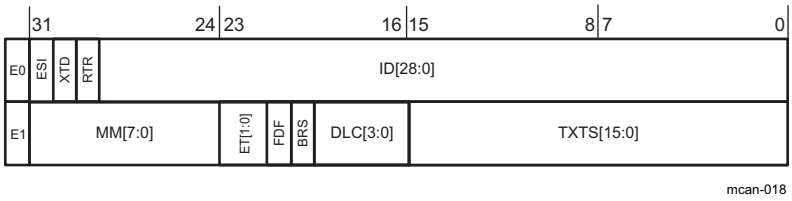


Table 18-623 shows Tx Event FIFO element field descriptions.

Table 18-623. Tx Event FIFO Element Field Descriptions

Word	Bits	Field Name	Description
E0	31	ESI	Error State Indicator • 0x0: Transmitting node is error active • 0x1: Transmitting node is error passive
	30	XTD	Extended Identifier • 0x0: 11-bit standard identifier • 0x1: 29-bit extended identifier
	29	RTR	Remote Transmission Request • 0x0: Data frame transmitted • 0x1: Remote frame transmitted
	28:0	ID[28:0]	Identifier Standard or extended identifier depending on XTD bit. A standard identifier has to be written to ID[28:18].

Table 18-623. Tx Event FIFO Element Field Descriptions (continued)

Word	Bits	Field Name	Description
E1	31:24	MM[7:0]	Message Marker Copied from Tx Buffer into Tx Event FIFO element for identification of Tx message status (see also MM[7:0] field in Table 18-622).
	23:22	ET[1:0]	Event Type • 0x0: Reserved • 0x1: Tx event • 0x2: Transmission in spite of cancellation (always set for transmissions in DAR mode) • 0x3: Reserved
	21	FDF	FD Format • 0x0: Standard frame format • 0x1: CAN FD frame format (new DLC-coding and CRC)
	20	BRS	Bit Rate Switch • 0x0: Frame transmitted without bit rate switching • 0x1: Frame transmitted with bit rate switching
	19:16	DLC[3:0]	Data Length Code • 0x0-0x8 (0-8): CAN + CAN FD: frame with 0-8 data bytes transmitted • 0x9-0xF (9-15): CAN: frame with 8 data bytes transmitted • 0x9-0xF (9-15): CAN FD: frame with 12/16/20/24/32/48/64 data bytes transmitted
	15:0	TXTS[15:0]	Tx Timestamp Timestamp Counter value captured on start of frame transmission. Resolution depending on configuration of the Timestamp Counter Prescaler MCAN_TSCC[19:16] TCP filed.

18.7.4.11.5 Standard Message ID Filter Element

Up to 128 filter elements can be configured for 11-bit standard IDs. When accessing a Standard Message ID Filter element, its address is the Filter List Standard Start Address MCAN_SIDFC[15:2] FLSSA field plus the index of the filter element (0-127).

Figure 18-152 shows Standard Message ID Filter element structure.

Figure 18-152. Standard Message ID Filter Element Structure

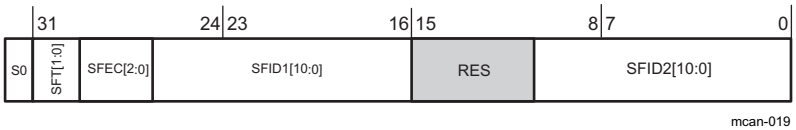


Table 18-624 shows Standard Message ID Filter element field descriptions.

Table 18-624. Standard Message ID Filter Element Field Descriptions

Word	Bits	Field Name	Description
S0	31:30	SFT[1:0]	Standard Filter Type • 0x0: Range filter from SFID1 to SFID2 (SFID2 ≥ SFID1) • 0x1: Dual ID filter for SFID1 or SFID2 • 0x2: Classic filter: SFID1 = filter; SFID2 = mask • 0x3: Filter element disabled **Note:** With SFT = 11 the filter element is disabled and the acceptance filtering continues (same behaviour as with SFEC = 000)
	29:27	SFEC[2:0]	Standard Filter Element Configuration All enabled filter elements are used for acceptance filtering of standard frames. Acceptance filtering stops at the first matching enabled filter element or when the end of the filter list is reached. If SFEC = 100, 101, or 110 a match sets interrupt flag MCAN_IR[8]HPM and, if enabled, an interrupt is generated. In this case the MCAN_HPMS register is updated with the status of the priority match. • 0x0: Disable filter element • 0x1: Store in Rx FIFO 0 if filter matches • 0x2: Store in Rx FIFO 1 if filter matches • 0x3: Reject ID if filter matches • 0x4: Set priority if filter matches • 0x5: Set priority and store in FIFO 0 if filter matches • 0x6: Set priority and store in FIFO 1 if filter matches • 0x7: Store into Rx Buffer, configuration of SFT[1:0] ignored
	26:16	SFID1[10:0]	Standard Filter ID 1 When filtering for Rx Buffers this field defines the ID of a standard message to be stored. The received identifiers must match exactly, no masking mechanism is used.
	15:11	RES	Reserved
	10:0	SFID2[10:0]	Standard Filter ID 2 This bit field has a different meaning depending on the configuration of SFEC: • 1) SFEC = 001 - 110 Second ID of standard ID filter element • 2) SFEC = 111 Filter for Rx Buffers
		SFID2[10:9]	This filed is decides whether the received message is stored into an Rx Buffer or treated as message A, B, or C of the debug message sequence. • 0x0: Store message into an Rx Buffer • 0x1: Debug Message A • 0x2: Debug Message B • 0x3: Debug Message C **Note:** Debug feature is not supported.
		SFID2[8:6]	This field is used to control the filter event pins at the Extension Interface. A one at the respective bit position enables generation of a pulse at the related filter event pin with the duration of one MCAN_ICKL period in case the filter matches. **Note:** Only two filter event pins are supported.
		SFID2[5:0]	This field defines the offset to the Rx Buffer Start Address MCAN_RXBC[15:2] RBSA field for storage of a matching message.

18.7.4.11.6 Extended Message ID Filter Element

Up to 64 filter elements can be configured for 29-bit extended IDs. When accessing an Extended Message ID Filter element, its address is the Filter List Extended Start Address MCAN_XIDFC[15:2] FLESA field plus two times the index of the filter element (0-63).

Figure 18-153 shows Extended Message ID Filter element structure.

Figure 18-153. Extended Message ID Filter Element Structure

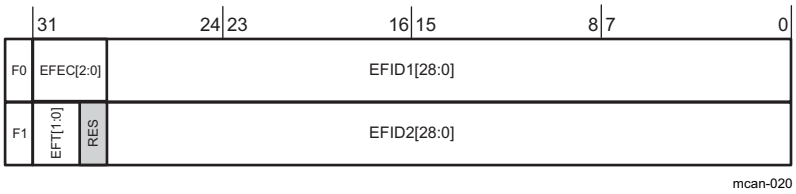


Table 18-625 shows Extended Message ID Filter element field descriptions.

Table 18-625. Extended Message ID Filter Element Field Descriptions

Word	Bits	Field Name	Description
F0	31:29	EFEC[2:0]	Extended Filter Element Configuration All enabled filter elements are used for acceptance filtering of extended frames. Acceptance filtering stops at the first matching enabled filter element or when the end of the filter list is reached. If EFEC = 100, 101, or 110 a match sets interrupt flag MCAN_IR[8]HPM and, if enabled, an interrupt is generated. In this case the MCAN_HPMS register is updated with the status of the priority match. • 0x0: Disable filter element • 0x1: Store in Rx FIFO 0 if filter matches • 0x2: Store in Rx FIFO 1 if filter matches • 0x3: Reject ID if filter matches • 0x4: Set priority if filter matches • 0x5: Set priority and store in FIFO 0 if filter matches • 0x6: Set priority and store in FIFO 1 if filter matches • 0x7: Store into Rx Buffer or as debug message, configuration of EFT[1:0] ignored
	28:0	EFID1[28:0]	Extended Filter ID 1 First ID of extended ID filter element. When filtering for Rx Buffers this field defines the ID of an extended message to be stored. The received identifiers must match exactly, only XIDAM masking mechanism (see Section 18.7.4.8.1.5, *Extended Message ID Filtering*) is used.

Table 18-625. Extended Message ID Filter Element Field Descriptions (continued)

Word	Bits	Field Name	Description
F1	31:30	EFT[1:0]	Extended Filter Type • 0x0: Range filter from EFID1 to EFID2 (EFID2 ≥ EFID1) • 0x1: Dual ID filter for EFID1 or EFID2 • 0x2: Classic filter: EFID1 = filter, EFID2 = mask • 0x3: Range filter from EFID1 to EFID2 (EFID2 ≥ EFID1), XIDAM mask not applied
	29	RES	Reserved
	28:0	EFID2[28:0]	Extended Filter ID 2 This bit field has a different meaning depending on the configuration of EFEC: • 1) EFEC = 001 - 110 Second ID of extended ID filter element • 2) EFEC = 111 Filter for Rx Buffers
		EFID2[10:9]	This field decides whether the received message is stored into an Rx Buffer or treated as message A, B, or C of the debug message sequence. • 0x0: Store message into an Rx Buffer • 0x1: Debug Message A • 0x2: Debug Message B • 0x3: Debug Message C **Note:** Debug feature is not supported.
		EFID2[8:6]	This field is used to control the filter event pins at the Extension Interface. A one at the respective bit position enables generation of a pulse at the related filter event pin with the duration of one MCAN_ICKL period in case the filter matches. **Note:** Only two filter event pins are supported.
		EFID2[5:0]	This field defines the offset to the Rx Buffer Start Address MCAN_RXBC[15:2] RBSA field for storage of a matching message.

18.7.5 MCAN Register Manual

18.7.5.1 MCAN Instance Summary

Table 18-626. MCAN Instance Summary

Module Name	Base Address	Size
MCAN		8 KiB

18.7.5.2 MCAN Registers

NOTE: After hardware reset, the registers of the MCAN module hold the values shown in the register descriptions.

Additionally, the Bus_Off state is reset and the MCAN_TX pin is set to recessive (high). The MCAN_CCCR[0] INIT bit is set to enable the software initialization. The MCAN module will not influence the CAN bus until the software resets the MCAN_CCCR[0] INIT bit.

18.7.5.2.1 MCAN Register Summary

18.7.5.2.2 MCAN Register Description

Table 18-627. MCANSS_PID

Address Offset	0x0000 1900
Physical Address	**Instance**
Description	Revision Register The Revision Register contains the major and minor revisions for the module.
Type	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REVISION																															

Bits	Field Name	Description	Type	Reset
31:0	REVISION	MCAN revision version	R	0x-

Table 18-628. Register Call Summary for Register MCANSS_PID

MCAN

- MCAN Register Summary:

Table 18-629. MCANSS_CTRL

Address Offset	0x0000 1904
Physical Address	**Instance**
Description	Control Register The Control Register contains general control bits for the MCAN module.
Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								EXT_TS_CNTR_EN	AUTOWAKEUP	WAKEUPREQEN	FREE	SOFT	CLKFACK	RESET	

Bits	Field Name	Description	Type	Reset
31:7	RESERVED	Reserved	R	0x0
6	EXT_TS_CNTR_EN	External Timestamp Counter Enable	RW	0x0
5	AUTOWAKEUP	Automatic Wakeup Enable	RW	0x0
4	WAKEUPREQEN	Wakeup Request Enable	RW	0x0
3	FREE	0x0: Disregard debug suspend 0x1: Enable Debug Suspend	RW	0x1
2	SOFT	If FREE = 0x1: 0x0: debug suspend doesn't wait for Idle 0x1: debug suspend waits for Idle	RW	0x0
1	CLKFACK	Clock Fast Ack	RW	0x0
0	RESET	Initiates a Soft Reset **Note**: Software application should complete all pending MCAN services before applying the soft reset. Accesses to MCAN core registers will be stalled until soft reset is completed.	W	0x0

Table 18-630. Register Call Summary for Register MCANSS_CTRL

MCAN

- Power Down (Sleep Mode): [0][1][2][3][4][5][6][7][8][9]
- MCAN Register Summary:

Table 18-631. MCANSS_STAT

Address Offset	0x0000 1908
Physical Address	**Instance**
Description	Status Regsiter The Status register provide general status bits for the MCAN module.
Type	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																										STATE			ENABLE_FDOE	MEM_INIT_DONE	RESET

Bits	Field Name	Description	Type	Reset
31:6	RESERVED	Reserved	R	0x0
5:3	STATE	0x0: Active 0x1: In transition to Idle 0x2: Idle 0x3: In transition to Active	R	0x0
2	ENABLE_FDOE	Enable CAN FD configuration	R	0x-
1	MEM_INIT_DONE	0x0: Memory Initialization is in progress 0x1: Memory Intialization Done	R	0x0
0	RESET	0x0: Not in reset 0x1: Reset is in progress	R	0x0

Table 18-632. Register Call Summary for Register MCANSS_STAT

MCAN
- MCAN Register Summary:

Table 18-633. MCANSS_ICS

Address Offset	0x0000 190C	
Physical Address		**Instance**
Description	Interrupt Clear Shadow Register Write to clear interrupt bits.	
Type	RW	

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															EXT_TS_CNTR_OVFL

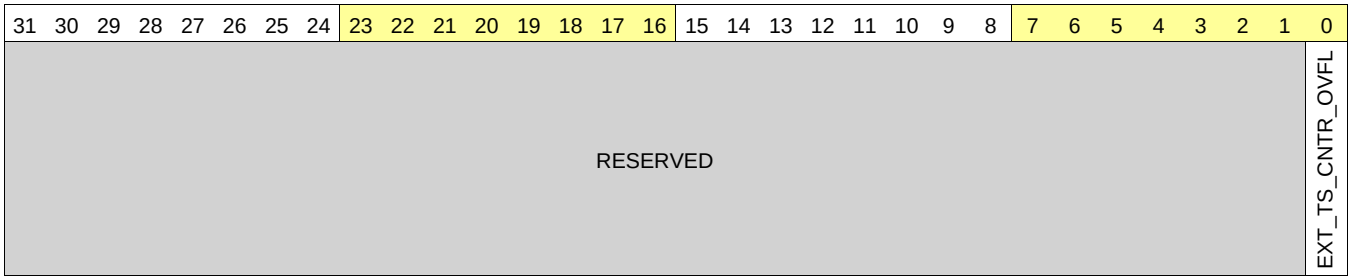
Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved	R	0x0
0	EXT_TS_CNTR_OVFL	External Timestamp Counter Overflow Interrupt status. Write 1 to clear bits.	W	0x0

Table 18-634. Register Call Summary for Register MCANSS_ICS

MCAN
- Interrupt Requests: [0]
- External Timestamp Counter: [1]
- MCAN Register Summary:

Table 18-635. MCANSS_IRS

Address Offset	0x0000 1910
Physical Address	**Instance**
Description	Interrupt Raw Status Register Read raw interrupt status. Write 1 to set interrupt bits.
Type	RW



Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved	R	0x0
0	EXT_TS_CNTR_OVFL	External Timestamp Counter Overflow Interrupt status.	RW1TS	0x0

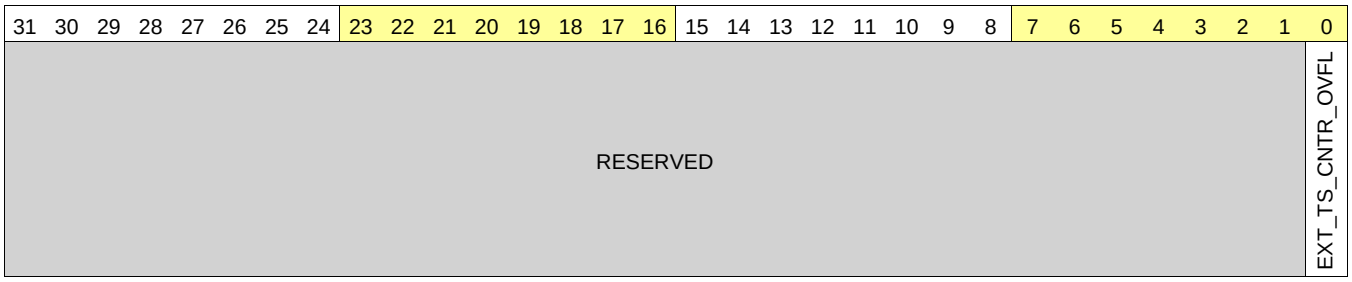
Table 18-636. Register Call Summary for Register MCANSS_IRS

MCAN

- Interrupt Requests: [0]
- External Timestamp Counter: [1][2][3]
- MCAN Register Summary:

Table 18-637. MCANSS_IECS

Address Offset	0x0000 1914
Physical Address	**Instance**
Description	Interrupt Enable Clear Shadow Register Write to clear interrupt enable bits.
Type	RW



Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved	R	0x0
0	EXT_TS_CNTR_OVFL	External Timestamp Counter Overflow Interrupt. Write 1 to clear bits.	W	0x0

Table 18-638. Register Call Summary for Register MCANSS_IECS

MCAN

- Interrupt Requests: [0]
- External Timestamp Counter: [1]
- MCAN Register Summary:

Table 18-639. MCANSS_IE

Address Offset	0x0000 1918
Physical Address	**Instance**
Description	Interrupt Enable Register Read interrupt Enable.
Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															EXT_TS_CNTR_OVFL

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved	R	0x0
0	EXT_TS_CNTR_OVFL	External Timestamp Counter Overflow Interrupt.	RW1TS	0x0

Table 18-640. Register Call Summary for Register MCANSS_IE

MCAN

- Interrupt Requests: [0]
- External Timestamp Counter: [1][2]
- MCAN Register Summary:

Table 18-641. MCANSS_IES

Address Offset	0x0000 191C
Physical Address	**Instance**
Description	Interrupt Enable Status Read Enabled Interrupts.
Type	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															EXT_TS_CNTR_OVFL

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved	R	0x0
0	EXT_TS_CNTR_OVFL	External Timestamp Counter Overflow Interrupt.	R	0x0

Table 18-642. Register Call Summary for Register MCANSS_IES

MCAN
- Interrupt Requests: [0]
- External Timestamp Counter: [1]
- MCAN Register Summary:

Table 18-643. MCANSS_EOI

Address Offset	0x0000 1920
Physical Address	**Instance**
Description	End Of Interrupt End of Interrupt Register.
Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								EOI							

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Reserved	R	0x0
7:0	EOI	Write with bit position of targetted interrupt (example: External TS is bit 0). Upon write, level interrupt will clear and if unserviced interrupt counter > 1 will issue another pulse interrupt.	W	0x0

Table 18-644. Register Call Summary for Register MCANSS_EOI

MCAN
- Interrupt Requests: [0]
- External Timestamp Counter: [1]
- MCAN Register Summary:

Table 18-645. MCANSS_EXT_TS_PRESCALER

Address Offset	0x0000 1924
Physical Address	**Instance**
Description	External Timestamp PreScaler 0 External TImeStamp PreScaler.
Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								PRESCALER																							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED	Reserved	R	0x0
23:0	PRESCALER	External Timestamp Prescaler reload value. External Timestamp count rate is Host clock (MCAN_ICLK) rate divided by this vlaue.	RW	0x0

Table 18-646. Register Call Summary for Register MCANSS_EXT_TS_PRESCALER

MCAN

- Interrupt Requests: [0]
- External Timestamp Counter: [1]
- MCAN Register Summary:

Table 18-647. MCANSS_EXT_TS_UNSERVICED_INTR_CNTR

Address Offset	0x0000 1980
Physical Address	**Instance**
Description	External Timestamp PreScaler 0 Unserviced Interrupts Counter External TImeStamp Unserviced Interrupts Counter.
Type	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								EXT_TS_INTR_CNTR							

Bits	Field Name	Description	Type	Reset
31:5	RESERVED	Reserved	R	0x0
4:0	EXT_TS_INTR_CNTR	Number of unserviced rollover interupts. If > 1 an EOI write will issue another pulse interrupt.	R	0x0

Table 18-648. Register Call Summary for Register MCANSS_EXT_TS_UNSERVICED_INTR_CNTR

MCAN

- Interrupt Requests: [0]
- MCAN Register Summary:

Table 18-649. MCANSS_ECC_EOI

Address Offset	0x0000 1980
Physical Address	**Instance**
Description	ECC EOI End Of Interrupt for ECC interrupt.
Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							ECC_EOI	RESERVED							

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reserved	R	0x0
8	ECC_EOI	ECC EOI	W	0x0
7:0	RESERVED	Reserved	R	0x0

Table 18-650. Register Call Summary for Register MCANSS_ECC_EOI

MCAN

- Interrupt Requests: [0]
- ECC Aggregator: [1]
- MCAN Register Summary:

Table 18-651. MCAN_CREL

Address Offset	0x0000 1A00
Physical Address	**Instance**
Description	Core Release Register Release dependent constant (version + date).
Type	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REL				STEP				SUBSTEP				YEAR				MON								DAY							

Bits	Field Name	Description	Type	Reset
31:28	REL	Core Release One digit, BCD-coded.	R	0x3
27:24	STEP	Step of Core Release One digit, BCD-coded.	R	0x2
23:20	SUBSTEP	Sub-step of Core Release One digit, BCD-coded.	R	0x1
19:16	YEAR	Time Stamp Year One digit, BCD-coded.	R	0x5
15:8	MON	Time Stamp Month Two digits, BCD-coded.	R	0x3
7:0	DAY	Time Stamp Day Two digits, BCD-coded.	R	0x20

Table 18-652. Register Call Summary for Register MCAN_CREL

MCAN

- MCAN Register Summary:

Table 18-653. MCAN_ENDN

Address Offset	0x0000 1A04
Physical Address	**Instance**
Description	Endian Register Constant 0x8765 4321.
Type	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ETV																															

Bits	Field Name	Description	Type	Reset
31:0	ETV	Endianness Test Value The endianness test value is 0x8765 4321.	R	0x8765 4321

Table 18-654. Register Call Summary for Register MCAN_ENDN

MCAN

- MCAN Register Summary:

Table 18-655. MCAN_DBTP

Address Offset	0x0000 1A0C
Physical Address	**Instance**
Description	Data Bit Timing & Prescaler Register Configuration of data phase bit timing, transmitter delay compensation enable. This register is only writable if the MCAN_CCCR[1] CCE and MCAN_CCCR[0] INIT bits are set. The CAN bit time may be programed in the range of 4 to 49 time quanta. The CAN time quantum may be programmed in the range of 1 to 32 MCAN_FCLK periods. t_q = (MCAN_DBTP[20:16] DBRP + 1) mtq (minimum time quantum = CAN clock period (MCAN_FCLK)). The MCAN_DBTP[12:8] DTSEG1 field is the sum of Prop_Seg and Phase_Seg1. The MCAN_DBTP[7:4] DTSEG2 field is Phase_Seg2. Therefore the length of the bit time is (programmed values) [MCAN_DBTP[12:8] DTSEG1 + MCAN_DBTP[7:4] DTSEG2 + 3] t_q or (functional values) [Sync_Seg + Prop_Seg + Phase_Seg1 + Phase_Seg2] t_q. The Information Processing Time (IPT) is zero, meaning the data for the next bit is available at the first clock edge after the sample point. **Note:** With a CAN clock (MCAN_FCLK) of 8 MHz, the reset value of 0x0000 0A33 configures the MCAN module for a data phase bit rate of 500 kBit/s. **Note:** The bit rate configured for the CAN FD data phase via the MCAN_DBTP register must be higher or equal to the bit rate configured for the arbitration phase via the MCAN_NBTP register.
Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								TDC	RESERVED		DBRP					RESERVED			DTSEG1					DTSEG2				DSJW			

Bits	Field Name	Description	Type	Reset
31:24	RESERVED	Reserved	R	0x0
23	TDC	Transmitter Delay Compensation 0x0: Transmitter Delay Compensation disabled 0x1: Transmitter Delay Compensation enabled	RW	0x0
22:21	RESERVED	Reserved	R	0x0
20:16	DBRP	Data Baud Rate Prescaler The value by which the oscillator frequency is divided for generating the bit time quanta. The bit time is built up from a multiple of this quanta. Valid values for the Baud Rate Prescaler are 0 to 31 (0x00-0x1F). The actual interpretation by the hardware of this value is such that one more than the value programmed here is used.	RW	0x0
15:13	RESERVED	Reserved	R	0x0
12:8	DTSEG1	Data time segment before sample point Valid values are 0 to 31 (0x00-0x1F). The actual interpretation by the hardware of this value is such that one more than the programmed value is used.	RW	0xA
7:4	DTSEG2	Data time segment after sample point Valid values are 0 to 15 (0x0-0xF). The actual interpretation by the hardware of this value is such that one more than the programmed value is used.	RW	0x3

Bits	Field Name	Description	Type	Reset
3:0	DSJW	Data (Re)Synchronization Jump Width Valid values are 0 to 15 (0x0-0xF). The actual interpretation by the hardware of this value is such that one more than the value programmed here is used.	RW	0x3

Table 18-656. Register Call Summary for Register MCAN_DBTP

MCAN
- CAN FD Operation: [0]
- Transmitter Delay Compensation: [1][2][3]
- MCAN Register Summary:
- MCAN Register Description: [6][7][8][9][10][11]

Table 18-657. MCAN_TEST

Address Offset	0x0000 1A10
Physical Address	**Instance**
Description	Test Register Test mode selection. Write access to the Test Register has to be enabled by setting the MCAN_CCCR[7] TEST bit. All Test Register functions are set to their reset values when the MCAN_CCCR[7] TEST bit is reset. Loop Back Mode and software control of the MCAN_TX pin are hardware test modes. Programming of the MCAN_TEST[6:5] TX field ≠ 00 may disturb the message transfer on the CAN bus.
Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								RX	TX		LBCK	RESERVED			

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Reserved	R	0x0
7	RX	Receive Pin Monitors the actual value of the MCAN_RX pin 0x0: The CAN bus is dominant (MCAN_RX = 0) 0x1: The CAN bus is recessive (MCAN_RX = 1)	R	0x0
6:5	TX	Control of Transmit Pin 0x0: Reset value, the MCAN_TX pin controlled by the CAN Core, updated at the end of the CAN bit time 0x1: Sample Point can be monitored at the MCAN_TX pin 0x2: Dominant (0) level at the MCAN_TX pin 0x3: Recessive (1) at the MCAN_TX pin	RW	0x0
4	LBCK	Loop Back Mode 0x0: Reset value, Loop Back Mode is disabled 0x1: Loop Back Mode is enabled (see Section 18.7.4.4.9, *Test Modes*)	RW	0x0
3:0	RESERVED	Reserved	R	0x0

Table 18-658. Register Call Summary for Register MCAN_TEST

MCAN
- Test Modes: [0][1][2][3][4][5][7]
- MCAN Register Summary:
- MCAN Register Description: [10][11][12]

Table 18-659. MCAN_RWD

Address Offset	0x0000 1A14
Physical Address	**Instance**
Description	RAM Watchdog Monitors the READY output of the Message RAM. The RAM Watchdog monitors the READY output of the Message RAM. A Message RAM access starts the Message RAM Watchdog Counter with the value configured by the MCAN_RWD[7:0] WDC field. The counter is reloaded with the MCAN_RWD[7:0] WDC field when the Message RAM signals successful completion by activating its READY output. In case there is no response from the Message RAM until the counter has counted down to zero, the counter stops and interrupt flag MCAN_IR[26] WDI is set. The RAM Watchdog Counter is clocked by the Host clock (MCAN_ICLK).
Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																WDV								WDC							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Reserved	R	0x0
15:8	WDV	Watchdog Value Actual Message RAM Watchdog Counter Value.	R	0x0
7:0	WDC	Watchdog Configuration Start value of the Message RAM Watchdog Counter. With the reset value of 00 the counter is disabled.	RW	0x0

Table 18-660. Register Call Summary for Register MCAN_RWD

MCAN

- MCAN Register Summary:
- MCAN Register Description: [2][3]

Table 18-661. MCAN_CCCR

Address Offset	0x0000 1A18
Physical Address	**Instance**
Description	CC Control Register Operation mode configuration. For details about setting and resetting of single bits, see Section 18.7.4.4.1, *Software Initialization*.
Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																TXP	EFBI	PXHD	RESERVED			BRSE	FDOE	TEST	DAR	MON	CSR	CSA	ASM	CCE	INIT

Bits	Field Name	Description	Type	Reset
31:15	RESERVED	Reserved	R	0x0
14	TXP	Transmit Pause If this bit is set, the MCAN module pauses for two CAN bit times before starting the next transmission after itself has successfully transmitted a frame (see Section 18.7.4.9, *Tx Handling*) 0x0: Transmit pause disabled 0x1: Transmit pause enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
13	EFBI	Edge Filtering during Bus Integration 0x0: Edge filtering disabled 0x1: Two consecutive dominant t_q required to detect an edge for hard synchronization	RW	0x0
12	PXHD	Protocol Exception Handling Disable 0x0: Protocol exception handling enabled 0x1: Protocol exception handling disabled **Note:** When protocol exception handling is disabled, the MCAN module will transmit an error frame when it detects a protocol exception condition.	RW	0x0
11:10	RESERVED	Reserved	R	0x0
9	BRSE	Bit Rate Switch Enable 0x0: Bit rate switching for transmissions disabled 0x1: Bit rate switching for transmissions enabled **Note:** When CAN FD operation is disabled the MCAN_CCCR[8] FDOE = 0, the MCAN_CCCR[9] BRSE bit is not evaluated.	RW	0x0
8	FDOE	FD Operation Enable 0x0: FD operation disabled 0x1: FD operation enabled	RW	0x0
7	TEST	Test Mode Enable 0x0: Normal operation. The MCAN_TEST register holds reset values 0x1: Test Mode. Write access to the MCAN_TEST register enabled	RW	0x0
6	DAR	Disable Automatic Retransmission 0x0: Automatic retransmission of messages not transmitted successfully enabled 0x1: Automatic retransmission disabled	RW	0x0
5	MON	Bus Monitoring Mode The MCAN_CCCR[5] MON bit can only be set by the Host CPU when both MCAN_CCCR[1] CCE and MCAN_CCCR[0] INIT bits are set to 1. The bit can be reset by the Host CPU at any time. 0x0: Bus Monitoring Mode is disabled 0x1: Bus Monitoring Mode is enabled	RW	0x0
4	CSR	Clock Stop Request 0x0: No clock stop is requested 0x1: Clock stop requested. When clock stop is requested, first the MCAN_CCCR[0] INIT bit and then the MCAN_CCCR[3] CSA bit will be set after all pending transfer requests have been completed and the CAN bus reached idle.	RW	0x0
3	CSA	Clock Stop Acknowledge 0x0: No clock stop acknowledged 0x1: The MCAN module may be set in power down by stopping MCAN_ICLK and MCAN_FCLK	R	0x0
2	ASM	Restricted Operation Mode The MCAN_CCCR[2] ASM bit can only be set by the Host CPU when both MCAN_CCCR[1] CCE and MCAN_CCCR[0] INIT bits are set to 1. The bit can be reset by the Host CPU at any time. For a description of the Restricted Operation Mode, see Section 18.7.4.4.5. 0x0: Normal CAN operation 0x1: Restricted Operation Mode active	RW	0x0

Bits	Field Name	Description	Type	Reset
1	CCE	Configuration Change Enable 0x0: The Host CPU has no write access to the protected configuration registers 0x1: The Host CPU has write access to the protected configuration registers (while the MCAN_CCCR[0] INIT = 1)	RW	0x0
0	INIT	Initialization 0x0: Normal Operation 0x1: Initialization is started **Note:** Due to the synchronization mechanism between the two clock domains, there may be a delay until the value written to the MCAN_CCCR[0] INIT bit can be read back. Therefore the software has to assure that the previous value written to the MCAN_CCCR[0] INIT bit has been accepted by reading the MCAN_CCCR[0] INIT bit before setting the MCAN_CCCR[0] INIT bit to a new value.	RW	0x1

Table 18-662. Register Call Summary for Register MCAN_CCCR

MCAN

- Fuseable CAN FD Operation Enable: [0][1][2]
- Software Initialization: [3][4][5][6][7][8][9][10][11][12][13][14][15][16][17][18][19][20][21][22][23]
- Normal Operation: [24]
- CAN FD Operation: [25][26][27][28][29][30][31][32][33][34][35][36][37][38]
- Transmitter Delay Compensation: [39]
- Restricted Operation Mode: [40][41][42][43]
- Bus Monitoring Mode: [47]
- Disabled Automatic Retransmission (DAR) Mode: [48]
- Power Down (Sleep Mode): [50][51][52][53][54][55][56][57][58][59][60][61][62][63]
- Test Modes: [66][67]
- Timeout Counter: [68][69][70]
- Tx Handling: [71][72][73]
- Transmit Pause: [74][75]
- Tx Buffer Element: [76][77][78]
- MCAN Registers: [79][80]
- MCAN Register Summary:
- MCAN Register Description: [83][84][85][86][87][88][89][90][91][92][93][94][95][96][97][98][99][100][101][102][103][104][105][106][107][108][109][110]

Table 18-663. MCAN_NBTP

Address Offset	0x0000 1A1C
Physical Address	**Instance**
Description	Nominal Bit Timing & Prescaler Register Configuration of arbitration phase bit timing. This register is only writable if the MCAN_CCCR[1] CCE and MCAN_CCCR[0] INIT bits are set. The CAN bit time may be programed in the range of 4 to 385 time quanta. The CAN time quantum may be programmed in the range of 1 to 512 MCAN_FCLK periods. t_q = (MCAN_NBTP[24:16] NBRP + 1) mtq. The MCAN_NBTP[15:8] NTSEG1 field is the sum of Prop_Seg and Phase_Seg1. The MCAN_NBTP[6:0] NTSEG2 field is Phase_Seg2. Therefore the length of the bit time is (programmed values) [MCAN_NBTP[15:8] NTSEG1 + MCAN_NBTP[6:0] NTSEG2 + 3] t_q or (functional values) [Sync_Seg + Prop_Seg + Phase_Seg1 + Phase_Seg2] t_q. The Information Processing Time (IPT) is zero, meaning the data for the next bit is available at the first clock edge after the sample point.
Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NSJW								NBRP								NTSEG1								RESERVED	NTSEG2						

Bits	Field Name	Description	Type	Reset
31:25	NSJW	Nominal (Re)Synchronization Jump Width Valid values are 0 to 127 (0x00-0x7F). The actual interpretation by the hardware of this value is such that one more than the value programmed here is used.	RW	0x3
24:16	NBRP	Nominal Baud Rate Prescaler The value by which the oscillator frequency is divided for generating the bit time quanta. The bit time is built up from a multiple of this quanta. Valid values for the Baud Rate Prescaler are 0 to 511 (0x000-0x1FF). The actual interpretation by the hardware of this value is such that one more than the value programmed here is used.	RW	0x0
15:8	NTSEG1	Nominal Time segment before sample point Valid values are 1 to 255 (0x01-0xFF). The actual interpretation by the hardware of this value is such that one more than the programmed value is used.	RW	0xA
7	RESERVED	Reserved	R	0x0
6:0	NTSEG2	Nominal Time segment after sample point Valid values are 0 to 127 (0x00-0x7F). The actual interpretation by the hardware of this value is such that one more than the programmed value is used. **Note:** With a CAN clock (MCAN_FCLK) of 8 MHz, the reset value of 0x0600 0A03 configures the MCAN module for a bit rate of 500 kBit/s.	RW	0x3

Table 18-664. Register Call Summary for Register MCAN_NBTP

MCAN

- CAN FD Operation: [0]
- MCAN Register Summary:
- MCAN Register Description: [3][4][5][6][7][8]

Table 18-665. MCAN_TSCC

Address Offset	0x0000 1A20
Physical Address	**Instance**
Description	Timestamp Counter Configuration Timestamp counter prescaler setting, selection of internal/external timestamp vector. For a description of the Timestamp Counter, see Section 18.7.4.5, *Timestamp Generation*.
Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												TCP				RESERVED														TSS	

Bits	Field Name	Description	Type	Reset
31:20	RESERVED	Reserved	R	0x0
19:16	TCP	Timestamp Counter Prescaler Configures the timestamp and timeout counters time unit in multiples of CAN bit times [1-16 (0x0-0xF)]. The actual interpretation by the hardware of this value is such that one more than the value programmed here is used. **Note:** With CAN FD an external counter is required for timestamp generation (MCAN_TSCC[1:0] TSS = 10)	RW	0x0
15:2	RESERVED	Reserved	R	0x0
1:0	TSS	Timestamp Select 0x0: Timestamp counter value always 0x0000 0x1: Timestamp counter value incremented according to the MCAN_TSCC[19:16] TCP field 0x2: External timestamp counter value used 0x3: Same as 00	RW	0x0

Table 18-666. Register Call Summary for Register MCAN_TSCC

MCAN

- Timestamp Generation: [0]
- External Timestamp Counter: [1]
- Timeout Counter: [2]
- Rx Buffer and FIFO Element: [3]
- Tx Event FIFO Element: [4]
- MCAN Register Summary:
- MCAN Register Description: [7][8][9][10][11][12]

Table 18-667. MCAN_TSCV

Address Offset	0x0000 1A24
Physical Address	**Instance**
Description	Timestamp Counter Value Read/reset timestamp counter.
Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																TSC															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Reserved	R	0x0
15:0	TSC	Timestamp Counter The internal/external Timestamp Counter value is captured on start of frame (both Rx and Tx). When the MCAN_TSCC[1:0] TSS = 01, the Timestamp Counter is incremented in multiples of CAN bit times [1-16] depending on the configuration of the MCAN_TSCC[19:16] TCP field. A wrap around sets interrupt flag MCAN_IR[16] TSW. Write access resets the counter to zero. When the MCAN_TSCC[1:0] TSS = 10, the MCAN_TSCV[15:0] TSC field reflects the external Timestamp Counter value. A write access has no impact. **Note:** A 'wrap around' is a change of the Timestamp Counter value from non-zero to zero not caused by write access to the MCAN_TSCV register.	RWTC	0x0

Table 18-668. Register Call Summary for Register MCAN_TSCV

MCAN
- Timestamp Generation: [0][1]
- MCAN Register Summary:
- MCAN Register Description: [4][5]

Table 18-669. MCAN_TOCC

Address Offset	0x0000 1A28
Physical Address	**Instance**
Description	Timeout Counter Configuration Configuration of timeout period, selection of timeout counter operation mode. For a description of the Timeout Counter, see Section 18.7.4.6, *Timeout Counter*.
Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TOP																RESERVED														TOS		ETOC

Bits	Field Name	Description	Type	Reset
31:16	TOP	Timeout Period Start value of the Timeout Counter (down-counter). Configures the Timeout Period.	RW	0xFFFF
15:3	RESERVED	Reserved	R	0x0
2:1	TOS	Timeout Select When operating in Continuous mode, a write to the MCAN_TOCV[15:0] TOC field presets the counter to the value configured by the MCAN_TOCC[31:16] TOP field and continues down-counting. When the Timeout Counter is controlled by one of the FIFOs, an empty FIFO presets the counter to the value configured by the MCAN_TOCC[31:16] TOP field. Down-counting is started when the first FIFO element is stored. 0x0: Continuous operation 0x1: Timeout controlled by Tx Event FIFO 0x2: Timeout controlled by Rx FIFO 0 0x3: Timeout controlled by Rx FIFO 1	RW	0x0

Bits	Field Name	Description	Type	Reset
0	ETOC	Enable Timeout Counter 0x0: Timeout Counter disabled 0x1: Timeout Counter enabled	RW	0x0

Table 18-670. Register Call Summary for Register MCAN_TOCC

MCAN
- Software Initialization: [0]
- Timeout Counter: [1][2][3][4][5][6]
- MCAN Register Summary:
- MCAN Register Description: [9][10][11]

Table 18-671. MCAN_TOCV

Address Offset	0x0000 1A2C
Physical Address	**Instance**
Description	Timeout Counter Value Read/reset timeout counter.
Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																TOC															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Reserved	R	0x0
15:0	TOC	Timeout Counter The Timeout Counter is decremented in multiples of CAN bit times [1-16] depending on the configuration of the MCAN_TSCC[19:16] TCP field. When decremented to zero, interrupt flag MCAN_IR[18] TOO is set and the Timeout Counter is stopped. Start and reset/restart conditions are configured via the MCAN_TOCC[2:1] TOS field.	RWTC	0xFFFF

Table 18-672. Register Call Summary for Register MCAN_TOCV

MCAN
- Software Initialization: [0]
- Timeout Counter: [1][2][3]
- MCAN Register Summary:
- MCAN Register Description: [6]

Table 18-673. MCAN_ECR

Address Offset	0x0000 1A40
Physical Address	**Instance**
Description	Error Counter Register State of Rx/Tx Error Counter, CAN Error Logging.
Type	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								CEL								RP	REC							TEC							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED	Reserved	R	0x0
23:16	CEL	CAN Error Logging The counter is incremented each time when a CAN protocol error causes the Transmit Error Counter or the Receive Error Counter to be incremented. It is reset by read access to the MCAN_ECR[23:16] CEL field. The counter stops at 0xFF; the next increment of the MCAN_ECR[7:0] TEC or MCAN_ECR[14:8] REC fields sets interrupt flag MCAN_IR[22] ELO.	R	0x0
15	RP	Receive Error Passive 0x0: The Receive Error Counter is below the error passive level of 128 0x1: The Receive Error Counter has reached the error passive level of 128	R	0x0
14:8	REC	Receive Error Counter Actual state of the Receive Error Counter, values between 0 and 127.	R	0x0
7:0	TEC	Transmit Error Counter Actual state of the Transmit Error Counter, values between 0 and 255. **Note:** When the MCAN_CCCR[2] ASM bit is set, the CAN protocol controller does not increment the MCAN_ECR[7:0] TEC and MCAN_ECR[14:8] REC fields when a CAN protocol error is detected, but the MCAN_ECR[23:16] CEL field is still incremented.	R	0x0

Table 18-674. Register Call Summary for Register MCAN_ECR

MCAN
- Restricted Operation Mode: [0][1][2]
- Power Down (Sleep Mode): [3]
- MCAN Register Summary:
- MCAN Register Description: [6][7][8][9][10][11][12]

Table 18-675. MCAN_PSR

Address Offset	0x0000 1A44
Physical Address	**Instance**
Description	Protocol Status Register CAN protocol controller status, transmitter delay compensation value.
Type	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								TDCV								RESERVED	PXE	RFDF	RBRS	RESI	DLEC			BO	EW	EP	ACT			LEC	

Bits	Field Name	Description	Type	Reset
31:23	RESERVED	Reserved	R	0x0
22:16	TDCV	Transmitter Delay Compensation Value Position of the secondary sample point, defined by the sum of the measured delay from the MCAN_TX to MCAN_RX pins and the MCAN_TDCR[14:8] TDCO field. The SSP position is, in the data phase, the number of mtq between the start of the transmitted bit and the secondary sample point. Valid values are 0 to 127 mtq (0x00-0x7F).	R	0x0

Bits	Field Name	Description	Type	Reset
15	RESERVED	Reserved	R	0x0
14	PXE	Protocol Exception Event 0x0: No protocol exception event occurred since last read access 0x1: Protocol exception event occurred	R	0x0
13	RFDF	Received a CAN FD Message This bit is set independent of acceptance filtering. 0x0: Since this bit was reset by the Host CPU, no CAN FD message has been received 0x1: Message in CAN FD format with FDF flag set has been received	R	0x0
12	RBRS	BRS flag of last received CAN FD Message This bit is set together with the MCAN_PSR[13] RFDF bit, independent of acceptance filtering. 0x0: Last received CAN FD message did not have its BRS flag set 0x1: Last received CAN FD message had its BRS flag set	R	0x0
11	RESI	ESI flag of last received CAN FD Message This bit is set together with the MCAN_PSR[13] RFDF bit, independent of acceptance filtering. 0x0: Last received CAN FD message did not have its ESI flag set 0x1: Last received CAN FD message had its ESI flag set	R	0x0
10:8	DLEC	Data Phase Last Error Code Type of last error that occurred in the data phase of a CAN FD format frame with its BRS flag set. Coding is the same as for the MCAN_PSR[2:0] LEC field. This field will be cleared to zero when a CAN FD format frame with its BRS flag set has been transferred (reception or transmission) without error.	R	0x7
7	BO	Bus_Off Status 0x0: The MCAN module is not Bus_Off 0x1: The MCAN module is in Bus_Off state	R	0x0
6	EW	Warning Status 0x0: Both error counters are below the Error_Warning limit of 96 0x1: At least one of error counter has reached the Error_Warning limit of 96	R	0x0
5	EP	Error Passive 0x0: The MCAN module is in the Error_Active state. It normally takes part in bus communication and sends an active error flag when an error has been detected 0x1: The MCAN module is in the Error_Passive state	R	0x0
4:3	ACT	Activity Monitors the module's CAN communication state. 0x0: Synchronizing - node is synchronizing on CAN communication 0x1: Idle - node is neither receiver nor transmitter 0x2: Receiver - node is operating as receiver 0x3: Transmitter - node is operating as transmitter **Note:** ACT is set to 00 by a Protocol Exception Event.	R	0x0

Bits	Field Name	Description	Type	Reset
2:0	LEC	Last Error Code The MCAN_PSR[2:0] LEC field indicates the type of the last error to occur on the CAN bus. This field will be cleared to 0 when a message has been transferred (reception or transmission) without error. 0x0: No Error: No error occurred since the MCAN_PSR[2:0] LEC field has been reset by successful reception or transmission. 0x1: Stuff Error: More than 5 equal bits in a sequence have occurred in a part of a received message where this is not allowed. 0x2: Form Error: A fixed format part of a received frame has the wrong format. 0x3: AckError: The message transmitted by the MCAN module was not acknowledged by another node. 0x4: Bit1Error: During the transmission of a message (with the exception of the arbitration field), the device wanted to send a recessive level (bit of logical value 1), but the monitored bus value was dominant. 0x5: Bit0Error: During the transmission of a message (or acknowledge bit, or active error flag, or overload flag), the device wanted to send a dominant level (data or identifier bit logical value 0), but the monitored bus value was recessive. During Bus_Off recovery this status is set each time a sequence of 11 recessive bits has been monitored. This enables the Host CPU to monitor the proceeding of the Bus_Off recovery sequence (indicating the bus is not stuck at dominant or continuously disturbed). 0x6: CRCError: The CRC check sum of a received message was incorrect. The CRC of an incom-ing message does not match with the CRC calculated from the received data. 0x7: NoChange: Any read access to the Protocol Status Register re-initializes the MCAN_PSR[2:0] LEC field to '0x7'. When the MCAN_PSR[2:0] LEC field shows the value '0x7', no CAN bus event was detected since the last Host CPU read access to the Protocol Status Register. **Note:** When a frame in CAN FD format has reached the data phase with BRS flag set, the next CAN event (error or valid frame) will be shown in the MCAN_PSR[10:8] DLEC field instead of the MCAN_PSR[2:0] LEC field. An error in a fixed stuff bit of a CAN FD CRC sequence will be shown as a Form Error, not Stuff Error. **Note:** The Bus_Off recovery sequence (see ISO11898-1:2015) cannot be shortened by setting or resetting the MCAN_CCCR[0] INIT bit. If the device goes Bus_Off, it will set the MCAN_CCCR[0] INIT bit of its own accord, stopping all bus activities. Once the MCAN_CCCR[0] INIT bit has been cleared by the Host CPU, the device will then wait for 129 occurrences of Bus Idle (129 × 11 consecutive recessive bits) before resuming normal operation. At the end of the Bus_Off recovery sequence, the Error Management Counters will be reset. During the waiting time after the resetting of the MCAN_CCCR[0] INIT bit, each time a sequence of 11 recessive bits has been monitored, a Bit0Error code is written to the MCAN_PSR[2:0] LEC field, enabling the Host CPU to readily check up whether the CAN bus is stuck at dominant or continuously disturbed and to monitor the Bus_Off recovery sequence. The MCAN_ECR[14:8] REC field is used to count these sequences.	R	0x7

Table 18-676. Register Call Summary for Register MCAN_PSR

MCAN

- CAN FD Operation: [0][1][2][3]
- Transmitter Delay Compensation: [4]
- Power Down (Sleep Mode): [5][6][7][8][9][10]
- Acceptance Filtering: [11][12][13][14]
- MCAN Register Summary:
- MCAN Register Description: [17][18][19][20][21][22][23][24][25][26][27][28]

Table 18-677. MCAN_TDCR

Address Offset	0x0000 1A48
Physical Address	**Instance**
Description	Transmitter Delay Comensation Register Configuration of transmitter delay compensation offset and filter window length.
Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																TDCO								RESERVED	TDCF						

Bits	Field Name	Description	Type	Reset
31:15	RESERVED	Reserved	R	0x0
14:8	TDCO	Transmitter Delay Compensation Offset Offset value defining the distance between the measured delay from the MCAN_RX and MCAN_TX pins and the secondary sample point. Valid values are 0 to 127 mtq (0x00-0x7F).	RW	0x0
7	RESERVED	Reserved	R	0x0
6:0	TDCF	Transmitter Delay Compensation Filter Window Length Defines the minimum value for the SSP position, dominant edges on the MCAN_RX pin that would result in an earlier SSP position are ignored for transmitter delay measure-ment. The feature is enabled when the MCAN_TDCR[6:0] TDCF field is configured to a value greater than the MCAN_TDCR[14:8] TDCO filed. Valid values are 0 to 127 mtq (0x00-0x7F).	RW	0x0

Table 18-678. Register Call Summary for Register MCAN_TDCR

MCAN

- Transmitter Delay Compensation: [0][1][2][3][4]
- MCAN Register Summary:
- MCAN Register Description: [7][8][9]

Table 18-679. MCAN_IR

Address Offset	0x0000 1A50
Physical Address	**Instance**
Description	Interrupt Register The flags are set when one of the listed conditions is detected (edge-sensitive). The flags remain set until the Host CPU clears them. A flag is cleared by writing a 1 to the corresponding bit position. Writing a 0 has no effect. A hard reset will clear the register. The configuration of the MCAN_IE register controls whether an interrupt is generated. The configuration of the MCAN_ILS register controls on which interrupt line an interrupt is signalled.
Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED		ARA	PED	PEA	WDI	BO	EW	EP	ELO	BEU	BEC	DRX	TOO	MRAF	TSW	TEFL	TEFF	TEFW	TEFN	TFE	TCF	TC	HPM	RF1L	RF1F	RF1W	RF1N	RF0L	RF0F	RF0W	RF0N

Bits	Field Name	Description	Type	Reset
31:30	RESERVED	Reserved	R	0x0
29	ARA	Access to Reserved Address 0x0: No access to reserved address occurred 0x1: Access to reserved address occurred	RW1TC	0x0
28	PED	Protocol Error in Data Phase 0x0: No protocol error in data phase 0x1: Protocol error in data phase detected (MCAN_PSR[10:8] DLEC ≠ 0.7)	RW1TC	0x0
27	PEA	Protocol Error in Arbitration Phase 0x0: No protocol error in arbitration phase 0x1: Protocol error in arbitration phase detected (MCAN_PSR[2:0] LEC ≠ 0.7)	RW1TC	0x0
26	WDI	Watchdog Interrupt 0x0: No Message RAM Watchdog event occurred 0x1: Message RAM Watchdog event due to missing READY	RW1TC	0x0
25	BO	Bus_Off Status 0x0: Bus_Off status unchanged 0x1: Bus_Off status changed	RW1TC	0x0
24	EW	Warning Status 0x0: Error_Warning status unchanged 0x1: Error_Warning status changed	RW1TC	0x0
23	EP	Error Passive 0x0: Error_Passive status unchanged 0x1: Error_Passive status changed	RW1TC	0x0
22	ELO	Error Logging Overflow 0x0: CAN Error Logging Counter did not overflow 0x1: Overflow of CAN Error Logging Counter occurred	RW1TC	0x0

Bits	Field Name	Description	Type	Reset
21	BEU	Bit Error Uncorrected Message RAM bit error detected, uncorrected. Controlled by input signal generated by parity/ECC logic attached to the Message RAM. An uncorrected Message RAM bit error sets the MCAN_CCCR[0] INIT bit to 1. This is done to avoid transmission of corrupted data. 0x0: No bit error detected when reading from Message RAM 0x1: Bit error detected, uncorrected (example: parity logic)	RW1TC	0x0
20	BEC	Bit Error Corrected Message RAM bit error detected and corrected. Controlled by input signal generated by parity/ECC logic attached to the Message RAM. 0x0: No bit error detected when reading from Message RAM 0x1: Bit error detected and corrected (example: ECC)	RW1TC	0x0
19	DRX	Message stored to Dedicated Rx Buffer The flag is set whenever a received message has been stored into a dedicated Rx Buffer. 0x0: No Rx Buffer updated 0x1: At least one received message stored into an Rx Buffer	RW1TC	0x0
18	TOO	Timeout Occurred 0x0: No timeout 0x1: Timeout reached	RW1TC	0x0
17	MRAF	Message RAM Access Failure The flag is set, when the Rx Handler: • has not completed acceptance filtering or storage of an accepted message until the arbitration field of the following message has been received. In this case acceptance filtering or message storage is aborted and the Rx Handler starts processing of the following message. • was not able to write a message to the Message RAM. In this case message storage is aborted. In both cases the FIFO put index is not updated respectively the New Data flag for a dedicated Rx Buffer is not set, a partly stored message is overwritten when the next message is stored to this location. The flag is also set when the Tx Handler was not able to read a message from the Message RAM in time. In this case message transmission is aborted. In case of a Tx Handler access failure the MCAN module is switched into Restricted Operation Mode (see Section 18.7.4.4.5). To leave Restricted Operation Mode, the Host CPU has to reset the MCAN_CCCR[2] ASM bit. 0x0: No Message RAM access failure occurred 0x1: Message RAM access failure occurred	RW1TC	0x0
16	TSW	Timestamp Wraparound 0x0: No timestamp counter wrap-around 0x1: Timestamp counter wrapped around	RW1TC	0x0
15	TEFL	Tx Event FIFO Element Lost 0x0: No Tx Event FIFO element lost 0x1: Tx Event FIFO element lost, also set after write attempt to Tx Event FIFO of size zero	RW1TC	0x0
14	TEFF	Tx Event FIFO Full 0x0: Tx Event FIFO not full 0x1: Tx Event FIFO full	RW1TC	0x0

Bits	Field Name	Description	Type	Reset
13	TEFW	Tx Event FIFO Watermark Reached 0x0: Tx Event FIFO fill level below watermark 0x1: Tx Event FIFO fill level reached watermark	RW1TC	0x0
12	TEFN	Tx Event FIFO New Entry 0x0: Tx Event FIFO unchanged 0x1: Tx Handler wrote Tx Event FIFO element	RW1TC	0x0
11	TFE	Tx FIFO Empty 0x0: Tx FIFO non-empty 0x1: Tx FIFO empty	RW1TC	0x0
10	TCF	Transmission Cancellation Finished 0x0: No transmission cancellation finished 0x1: Transmission cancellation finished	RW1TC	0x0
9	TC	Transmission Completed 0x0: No transmission completed 0x1: Transmission completed	RW1TC	0x0
8	HPM	High Priority Message 0x0: No high priority message received 0x1: High priority message received	RW1TC	0x0
7	RF1L	Rx FIFO 1 Message Lost 0x0: No Rx FIFO 1 message lost 0x1: Rx FIFO 1 message lost, also set after write attempt to Rx FIFO 1 of size zero	RW1TC	0x0
6	RF1F	Rx FIFO 1 Full 0x0: Rx FIFO 1 not full 0x1: Rx FIFO 1 full	RW1TC	0x0
5	RF1W	Rx FIFO 1 Watermark Reached 0x0: Rx FIFO 1 fill level below watermark 0x1: Rx FIFO 1 fill level reached watermark	RW1TC	0x0
4	RF1N	Rx FIFO 1 New Message 0x0: No new message written to Rx FIFO 1 0x1: New message written to Rx FIFO 1	RW1TC	0x0
3	RF0L	Rx FIFO 0 Message Lost 0x0: No Rx FIFO 0 message lost 0x1: Rx FIFO 0 message lost, also set after write attempt to Rx FIFO 0 of size zero	RW1TC	0x0
2	RF0F	Rx FIFO 0 Full 0x0: Rx FIFO 0 not full 0x1: Rx FIFO 0 full	RW1TC	0x0
1	RF0W	Rx FIFO 0 Watermark Reached 0x0: Rx FIFO 0 fill level below watermark 0x1: Rx FIFO 0 fill level reached watermark	RW1TC	0x0
0	RF0N	Rx FIFO 0 New Message 0x0: No new message written to Rx FIFO 0 0x1: New message written to Rx FIFO 0	RW1TC	0x0

Table 18-680. Register Call Summary for Register MCAN_IR

MCAN

- MCAN Functional Description: [0]
- Interrupt Requests: [1]
- Timestamp Generation: [2]
- Timeout Counter: [3]
- Acceptance Filtering: [4][5]
- Rx FIFOs: [6][7][8][9][10][11][12][13]
- Dedicated Rx Buffers: [14][15]
- Tx Event Handling: [16][17][18]
- Standard Message ID Filter Element: [19]
- Extended Message ID Filter Element: [20]
- MCAN Register Summary:
- MCAN Register Description: [23][24][25][26][27][28][29][30][31][32][33][34][35]

Table 18-681. MCAN_IE

Address Offset	0x0000 1A54
Physical Address	**Instance**
Description	Interrupt Enable The settings in the Interrupt Enable register determine which status changes in the Interrupt Register are signalled on an interrupt line.
Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED		ARAE	PEDE	PEAE	WDIE	BOE	EWE	EPE	ELOE	BEUE	BECE	DRX	TOOE	MRAFE	TSWE	TEFLE	TEFFE	TEFWE	TEFNE	TFEE	TCFE	TCE	HPME	RF1LE	RF1FE	RF1WE	RF1NE	RF0LE	RF0FE	RF0WE	RF0NE

Bits	Field Name	Description	Type	Reset
31:30	RESERVED	Reserved	R	0x0
29	ARAE	Access to Reserved Address Enable 0x0: Interrupt disabled 0x1: Interrupt enabled	RW	0x0
28	PEDE	Protocol Error in Data Phase Enable 0x0: Interrupt disabled 0x1: Interrupt enabled	RW	0x0
27	PEAE	Protocol Error in Arbitration Phase Enable 0x0: Interrupt disabled 0x1: Interrupt enabled	RW	0x0
26	WDIE	Watchdog Interrupt Enable 0x0: Interrupt disabled 0x1: Interrupt enabled	RW	0x0
25	BOE	Bus_Off Status Interrupt Enable 0x0: Interrupt disabled 0x1: Interrupt enabled	RW	0x0
24	EWE	Warning Status Interrupt Enable 0x0: Interrupt disabled 0x1: Interrupt enabled	RW	0x0
23	EPE	Error Passive Interrupt Enable 0x0: Interrupt disabled 0x1: Interrupt enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
22	ELOE	Error Logging Overflow Interrupt Enable 0x0: Interrupt disabled 0x1: Interrupt enabled	RW	0x0
21	BEUE	Bit Error Uncorrected Interrupt Enable 0x0: Interrupt disabled 0x1: Interrupt enabled	RW	0x0
20	BECE	Bit Error Corrected Interrupt Enable 0x0: Interrupt disabled 0x1: Interrupt enabled	RW	0x0
19	DRX	Message stored to Dedicated Rx Buffer Interrupt Enable	RW	0x0
18	TOOE	Timeout Occurred Interrupt Enable 0x0: Interrupt disabled 0x1: Interrupt enabled	RW	0x0
17	MRAFE	Message RAM Access Failure Interrupt Enable 0x0: Interrupt disabled 0x1: Interrupt enabled	RW	0x0
16	TSWE	Timestamp Wraparound Interrupt Enable 0x0: Interrupt disabled 0x1: Interrupt enabled	RW	0x0
15	TEFLE	Tx Event FIFO Event Lost Interrupt Enable 0x0: Interrupt disabled 0x1: Interrupt enabled	RW	0x0
14	TEFFE	Tx Event FIFO Full Interrupt Enable 0x0: Interrupt disabled 0x1: Interrupt enabled	RW	0x0
13	TEFWE	Tx Event FIFO Watermark Reached Interrupt Enable 0x0: Interrupt disabled 0x1: Interrupt enabled	RW	0x0
12	TEFNE	Tx Event FIFO New Entry Interrupt Enable 0x0: Interrupt disabled 0x1: Interrupt enabled	RW	0x0
11	TFEE	Tx FIFO Empty Interrupt Enable 0x0: Interrupt disabled 0x1: Interrupt enabled	RW	0x0
10	TCFE	Transmission Cancellation Finished Interrupt Enable 0x0: Interrupt disabled 0x1: Interrupt enabled	RW	0x0
9	TCE	Transmission Completed Interrupt Enable 0x0: Interrupt disabled 0x1: Interrupt enabled	RW	0x0
8	HPME	High Priority Message Interrupt Enable 0x0: Interrupt disabled 0x1: Interrupt enabled	RW	0x0
7	RF1LE	Rx FIFO 1 Message Lost Interrupt Enable 0x0: Interrupt disabled 0x1: Interrupt enabled	RW	0x0
6	RF1FE	Rx FIFO 1 Full Interrupt Enable 0x0: Interrupt disabled 0x1: Interrupt enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
5	RF1WE	Rx FIFO 1 Watermark Reached Interrupt Enable 0x0: Interrupt disabled 0x1: Interrupt enabled	RW	0x0
4	RF1NE	Rx FIFO 1 New Message Interrupt Enable 0x0: Interrupt disabled 0x1: Interrupt enabled	RW	0x0
3	RF0LE	Rx FIFO 0 Message Lost Interrupt Enable 0x0: Interrupt disabled 0x1: Interrupt enabled	RW	0x0
2	RF0FE	Rx FIFO 0 Full Interrupt Enable 0x0: Interrupt disabled 0x1: Interrupt enabled	RW	0x0
1	RF0WE	Rx FIFO 0 Watermark Reached Interrupt Enable 0x0: Interrupt disabled 0x1: Interrupt enabled	RW	0x0
0	RF0NE	Rx FIFO 0 New Message Interrupt Enable 0x0: Interrupt disabled 0x1: Interrupt enabled	RW	0x0

Table 18-682. Register Call Summary for Register MCAN_IE

MCAN
- Interrupt Requests: [0]
- MCAN Register Summary:
- MCAN Register Description: [3]

Table 18-683. MCAN_ILS

Address Offset	0x0000 1A58
Physical Address	**Instance**
Description	Interrupt Line Select The Interrupt Line Select register assigns an interrupt generated by a specific interrupt flag from the Interrupt Register to one of the two module interrupt lines. For interrupt generation the respective interrupt line has to be enabled via the MCAN_ILE[0] EINT0 and MCAN_ILE[1] EINT1 bits.
Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED		ARAL	PEDL	PEAL	WDIL	BOL	EWL	EPL	ELOL	BEUL	BECL	DRXL	TOOL	MRAFL	TSWL	TEFLL	TEFFL	TEFWL	TEFNL	TFEL	TCFL	TCL	HPML	RF1LL	RF1FL	RF1WL	RF1NL	RF0LL	RF0FL	RF0WL	RF0NL

Bits	Field Name	Description	Type	Reset
31:30	RESERVED	Reserved	R	0x0
29	ARAL	Access to Reserved Address Line 0x0: Interrupt assigned to interrupt line INT0 0x1: Interrupt assigned to interrupt line INT1	RW	0x0
28	PEDL	Protocol Error in Data Phase Line 0x0: Interrupt assigned to interrupt line INT0 0x1: Interrupt assigned to interrupt line INT1	RW	0x0
27	PEAL	Protocol Error in Arbitration Phase Line 0x0: Interrupt assigned to interrupt line INT0 0x1: Interrupt assigned to interrupt line INT1	RW	0x0

Bits	Field Name	Description	Type	Reset
26	WDIL	Watchdog Interrupt Line 0x0: Interrupt assigned to interrupt line INT0 0x1: Interrupt assigned to interrupt line INT1	RW	0x0
25	BOL	Bus_Off Status Interrupt Line 0x0: Interrupt assigned to interrupt line INT0 0x1: Interrupt assigned to interrupt line INT1	RW	0x0
24	EWL	Warning Status Interrupt Line 0x0: Interrupt assigned to interrupt line INT0 0x1: Interrupt assigned to interrupt line INT1	RW	0x0
23	EPL	Error Passive Interrupt Line 0x0: Interrupt assigned to interrupt line INT0 0x1: Interrupt assigned to interrupt line INT1	RW	0x0
22	ELOL	Error Logging Overflow Interrupt Line 0x0: Interrupt assigned to interrupt line INT0 0x1: Interrupt assigned to interrupt line INT1	RW	0x0
21	BEUL	Bit Error Uncorrected Interrupt Line 0x0: Interrupt assigned to interrupt line INT0 0x1: Interrupt assigned to interrupt line INT1	RW	0x0
20	BECL	Bit Error Corrected Interrupt Line 0x0: Interrupt assigned to interrupt line INT0 0x1: Interrupt assigned to interrupt line INT1	RW	0x0
19	DRXL	Message stored to Dedicated Rx Buffer Interrupt Line 0x0: Interrupt assigned to interrupt line INT0 0x1: Interrupt assigned to interrupt line INT1	RW	0x0
18	TOOL	Timeout Occurred Interrupt Line 0x0: Interrupt assigned to interrupt line INT0 0x1: Interrupt assigned to interrupt line INT1	RW	0x0
17	MRAFL	Message RAM Access Failure Interrupt Line 0x0: Interrupt assigned to interrupt line INT0 0x1: Interrupt assigned to interrupt line INT1	RW	0x0
16	TSWL	Timestamp Wraparound Interrupt Line 0x0: Interrupt assigned to interrupt line INT0 0x1: Interrupt assigned to interrupt line INT1	RW	0x0
15	TEFLL	Tx Event FIFO Event Lost Interrupt Line 0x0: Interrupt assigned to interrupt line INT0 0x1: Interrupt assigned to interrupt line INT1	RW	0x0
14	TEFFL	Tx Event FIFO Full Interrupt Line 0x0: Interrupt assigned to interrupt line INT0 0x1: Interrupt assigned to interrupt line INT1	RW	0x0
13	TEFWL	Tx Event FIFO Watermark Reached Interrupt Line 0x0: Interrupt assigned to interrupt line INT0 0x1: Interrupt assigned to interrupt line INT1	RW	0x0
12	TEFNL	Tx Event FIFO New Entry Interrupt Line 0x0: Interrupt assigned to interrupt line INT0 0x1: Interrupt assigned to interrupt line INT1	RW	0x0
11	TFEL	Tx FIFO Empty Interrupt Line 0x0: Interrupt assigned to interrupt line INT0 0x1: Interrupt assigned to interrupt line INT1	RW	0x0
10	TCFL	Transmission Cancellation Finished Interrupt Line 0x0: Interrupt assigned to interrupt line INT0 0x1: Interrupt assigned to interrupt line INT1	RW	0x0

Bits	Field Name	Description	Type	Reset
9	TCL	Transmission Completed Interrupt Line 0x0: Interrupt assigned to interrupt line INT0 0x1: Interrupt assigned to interrupt line INT1	RW	0x0
8	HPML	High Priority Message Interrupt Line 0x0: Interrupt assigned to interrupt line INT0 0x1: Interrupt assigned to interrupt line INT1	RW	0x0
7	RF1LL	Rx FIFO 1 Message Lost Interrupt Line 0x0: Interrupt assigned to interrupt line INT0 0x1: Interrupt assigned to interrupt line INT1	RW	0x0
6	RF1FL	Rx FIFO 1 Full Interrupt Line 0x0: Interrupt assigned to interrupt line INT0 0x1: Interrupt assigned to interrupt line INT1	RW	0x0
5	RF1WL	Rx FIFO 1 Watermark Reached Interrupt Line 0x0: Interrupt assigned to interrupt line INT0 0x1: Interrupt assigned to interrupt line INT1	RW	0x0
4	RF1NL	Rx FIFO 1 New Message Interrupt Line 0x0: Interrupt assigned to interrupt line INT0 0x1: Interrupt assigned to interrupt line INT1	RW	0x0
3	RF0LL	Rx FIFO 0 Message Lost Interrupt Line 0x0: Interrupt assigned to interrupt line INT0 0x1: Interrupt assigned to interrupt line INT1	RW	0x0
2	RF0FL	Rx FIFO 0 Full Interrupt Line 0x0: Interrupt assigned to interrupt line INT0 0x1: Interrupt assigned to interrupt line INT1	RW	0x0
1	RF0WL	Rx FIFO 0 Watermark Reached Interrupt Line 0x0: Interrupt assigned to interrupt line INT0 0x1: Interrupt assigned to interrupt line INT1	RW	0x0
0	RF0NL	Rx FIFO 0 New Message Interrupt Line 0x0: Interrupt assigned to interrupt line INT0 0x1: Interrupt assigned to interrupt line INT1	RW	0x0

Table 18-684. Register Call Summary for Register MCAN_ILS

MCAN

- Interrupt Requests: [0]
- MCAN Register Summary:
- MCAN Register Description: [3]

Table 18-685. MCAN_ILE

Address Offset	0x0000 1A5C
Physical Address	**Instance**
Description	Interrupt Line Enable Enable/disable interrupt lines INT0/INT1. Each of the two interrupt lines to the Host CPU can be enabled/disabled separately by programming the MCAN_ILE[0] EINT0 and MCAN_ILE[1] EINT1 bits.
Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														EINT1	EINT0

Bits	Field Name	Description	Type	Reset
31:2	RESERVED	Reserved	R	0x0
1	EINT1	Enable Interrupt Line 1 0x0: Interrupt line INT1 disabled 0x1: Interrupt line INT1 enabled	RW	0x0
0	EINT0	Enable Interrupt Line 0 0x0: Interrupt line INT0 disabled 0x1: Interrupt line INT0 enabled	RW	0x0

Table 18-686. Register Call Summary for Register MCAN_ILE

MCAN
- Interrupt Requests: [0]
- MCAN Register Summary:
- MCAN Register Description: [3][4][5][6]

Table 18-687. MCAN_GFC

Address Offset	0x0000 1A80
Physical Address	**Instance**
Description	Global Filter Configuration Handling of non-matching frames and remote frames. Global settings for Message ID filtering. The Global Filter Configuration controls the filter path for standard and extended messages (see Figure 18-142 and Figure 18-143).
Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								ANFS		ANFE		RRFS	RRFE		

Bits	Field Name	Description	Type	Reset
31:6	RESERVED	Reserved	R	0x0
5:4	ANFS	Accept Non-matching Frames Standard Defines how received messages with 11-bit IDs that do not match any element of the filter list are treated. 0x0: Accept in Rx FIFO 0 0x1: Accept in Rx FIFO 1 0x2: Reject 0x3: Reject	RW	0x0
3:2	ANFE	Accept Non-matching Frames Extended Defines how received messages with 29-bit IDs that do not match any element of the filter list are treated. 0x0: Accept in Rx FIFO 0 0x1: Accept in Rx FIFO 1 0x2: Reject 0x3: Reject	RW	0x0
1	RRFS	Reject Remote Frames Standard 0x0: Filter remote frames with 11-bit standard IDs 0x1: Reject all remote frames with 11-bit standard IDs	RW	0x0
0	RRFE	Reject Remote Frames Extended 0x0: Filter remote frames with 29-bit extended IDs 0x1: Reject all remote frames with 29-bit extended IDs	RW	0x0

Table 18-688. Register Call Summary for Register MCAN_GFC

MCAN

- Acceptance Filtering: [0][1][2]
- Rx Buffer and FIFO Element: [3][4]
- MCAN Register Summary:

Table 18-689. MCAN_SIDFC

Address Offset	0x0000 1A84
Physical Address	**Instance**
Description	Standard ID Filter Configuration Number of filter elements, pointer to start of filter list. Settings for 11-bit standard Message ID filtering. The Standard ID Filter Configuration controls the filter path for standard messages (see Figure 18-142).
Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								LSS								FLSSA														RESERVED	

Bits	Field Name	Description	Type	Reset
31:24	RESERVED	Reserved	R	0x0
23:16	LSS	List Size Standard 0x0: No standard Message ID filter 0x1-0x80 (1-128): Number of standard Message ID filter elements > 0x80 (128): Values greater than 128 are interpreted as 128	RW	0x0
15:2	FLSSA	Filter List Standard Start Address Start address of standard Message ID filter list (32-bit word address, see Figure 18-148).	RW	0x0
1:0	RESERVED	Reserved	R	0x0

Table 18-690. Register Call Summary for Register MCAN_SIDFC

MCAN

- Acceptance Filtering: [0][1]
- Rx Buffer and FIFO Element: [2]
- Standard Message ID Filter Element: [3]
- MCAN Register Summary:
- MCAN Register Description: [6]

Table 18-691. MCAN_XIDFC

Address Offset	0x0000 1A88
Physical Address	**Instance**
Description	Extended ID Filter Configuration Number of filter elements, pointer to start of filter list. Settings for 29-bit extended Message ID filtering. The Extended ID Filter Configuration controls the filter path for standard messages (see Figure 18-143).
Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								LSE								FLESA														RESERVED	

Bits	Field Name	Description	Type	Reset
31:23	RESERVED	Reserved	R	0x0
22:16	LSE	List Size Extended 0x0: No extended Message ID filter 0x1-0x40 (1-64): Number of extended Message ID filter elements > 0x40 (64): Values greater than 64 are interpreted as 64	RW	0x0
15:2	FLESA	Filter List Extended Start Address Start address of extended Message ID filter list (32-bit word address, see Figure 18-148).	RW	0x0
1:0	RESERVED	Reserved	R	0x0

Table 18-692. Register Call Summary for Register MCAN_XIDFC

MCAN

- Acceptance Filtering: [0][1]
- Rx Buffer and FIFO Element: [2]
- Extended Message ID Filter Element: [3]
- MCAN Register Summary:
- MCAN Register Description: [6]

Table 18-693. MCAN_XIDAM

Address Offset	0x0000 1A90
Physical Address	**Instance**
Description	Extended ID AND Mask 29-bit logical AND mask for J1939.
Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								EIDM																							

Bits	Field Name	Description	Type	Reset
31:29	RESERVED	Reserved	R	0x0
28:0	EIDM	Extended ID Mask For acceptance filtering of extended frames the Extended ID AND Mask is ANDed with the Message ID of a received frame. Intended for masking of 29-bit IDs in SAE J1939. With the reset value of all bits set to one the mask is not active.	RW	0x1FFFFFFF

Table 18-694. Register Call Summary for Register MCAN_XIDAM

MCAN
- Acceptance Filtering: [0][1][2][3][4]
- MCAN Register Summary:

Table 18-695. MCAN_HPMS

Address Offset	0x0000 1A94
Physical Address	**Instance**
Description	High Priority Message Status Status monitoring of incoming high priority messages. This register is updated every time a Message ID filter element configured to generate a priority event matches. This can be used to monitor the status of incoming high priority messages and to enable fast access to these messages.
Type	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																FLST	FIDX							MSI		BIDX					

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Reserved	R	0x0
15	FLST	Filter List Indicates the filter list of the matching filter element. 0x0: Standard Filter List 0x1: Extended Filter List	R	0x0
14:8	FIDX	Filter Index Index of matching filter element. Range is 0 to MCAN_SIDFC[23:16] LSS - 1 respectively MCAN_XIDFC[22:16] LSE - 1.	R	0x0
7:6	MSI	Message Storage Indicator 0x0: No FIFO selected 0x1: FIFO message lost 0x2: Message stored in FIFO 0 0x3: Message stored in FIFO 1	R	0x0
5:0	BIDX	Buffer Index Index of Rx FIFO element to which the message was stored. Only valid when the MCAN_HPMS[7:6] MSI = 1.	R	0x0

Table 18-696. Register Call Summary for Register MCAN_HPMS

MCAN
- Software Initialization: [0]
- Standard Message ID Filter Element: [1]
- Extended Message ID Filter Element: [2]
- MCAN Register Summary:
- MCAN Register Description: [5]

Table 18-697. MCAN_NDAT1

Address Offset	0x0000 1A98
Physical Address	**Instance**
Description	New Data 1 NewDat flags of dedicated Rx buffers 0-31. The register holds the New Data flags of Rx Buffers 0 to 31. The flags are set when the respective Rx Buffer has been updated from a received frame. The flags remain set until the Host CPU clears them. Aflag is cleared by writing a '1' to the corresponding bit position. Writing a '0' has no effect. Ahard reset will clear the register.
Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ND31	ND30	ND29	ND28	ND27	ND26	ND25	ND24	ND23	ND22	ND21	ND20	ND19	ND18	ND17	ND16	ND15	ND14	ND13	ND12	ND11	ND10	ND9	ND8	ND7	ND6	ND5	ND4	ND3	ND2	ND1	ND0

Bits	Field Name	Description	Type	Reset
31	ND31	New Data 0x0: Rx Buffer not updated 0x1: Rx Buffer updated from new message	RW1TC	0x0
30	ND30	New Data 0x0: Rx Buffer not updated 0x1: Rx Buffer updated from new message	RW1TC	0x0
29	ND29	New Data 0x0: Rx Buffer not updated 0x1: Rx Buffer updated from new message	RW1TC	0x0
28	ND28	New Data 0x0: Rx Buffer not updated 0x1: Rx Buffer updated from new message	RW1TC	0x0
27	ND27	New Data 0x0: Rx Buffer not updated 0x1: Rx Buffer updated from new message	RW1TC	0x0
26	ND26	New Data 0x0: Rx Buffer not updated 0x1: Rx Buffer updated from new message	RW1TC	0x0
25	ND25	New Data 0x0: Rx Buffer not updated 0x1: Rx Buffer updated from new message	RW1TC	0x0
24	ND24	New Data 0x0: Rx Buffer not updated 0x1: Rx Buffer updated from new message	RW1TC	0x0
23	ND23	New Data 0x0: Rx Buffer not updated 0x1: Rx Buffer updated from new message	RW1TC	0x0
22	ND22	New Data 0x0: Rx Buffer not updated 0x1: Rx Buffer updated from new message	RW1TC	0x0
21	ND21	New Data 0x0: Rx Buffer not updated 0x1: Rx Buffer updated from new message	RW1TC	0x0
20	ND20	New Data 0x0: Rx Buffer not updated 0x1: Rx Buffer updated from new message	RW1TC	0x0

Bits	Field Name	Description	Type	Reset
19	ND19	New Data 0x0: Rx Buffer not updated 0x1: Rx Buffer updated from new message	RW1TC	0x0
18	ND18	New Data 0x0: Rx Buffer not updated 0x1: Rx Buffer updated from new message	RW1TC	0x0
17	ND17	New Data 0x0: Rx Buffer not updated 0x1: Rx Buffer updated from new message	RW1TC	0x0
16	ND16	New Data 0x0: Rx Buffer not updated 0x1: Rx Buffer updated from new message	RW1TC	0x0
15	ND15	New Data 0x0: Rx Buffer not updated 0x1: Rx Buffer updated from new message	RW1TC	0x0
14	ND14	New Data 0x0: Rx Buffer not updated 0x1: Rx Buffer updated from new message	RW1TC	0x0
13	ND13	New Data 0x0: Rx Buffer not updated 0x1: Rx Buffer updated from new message	RW1TC	0x0
12	ND12	New Data 0x0: Rx Buffer not updated 0x1: Rx Buffer updated from new message	RW1TC	0x0
11	ND11	New Data 0x0: Rx Buffer not updated 0x1: Rx Buffer updated from new message	RW1TC	0x0
10	ND10	New Data 0x0: Rx Buffer not updated 0x1: Rx Buffer updated from new message	RW1TC	0x0
9	ND9	New Data 0x0: Rx Buffer not updated 0x1: Rx Buffer updated from new message	RW1TC	0x0
8	ND8	New Data 0x0: Rx Buffer not updated 0x1: Rx Buffer updated from new message	RW1TC	0x0
7	ND7	New Data 0x0: Rx Buffer not updated 0x1: Rx Buffer updated from new message	RW1TC	0x0
6	ND6	New Data 0x0: Rx Buffer not updated 0x1: Rx Buffer updated from new message	RW1TC	0x0
5	ND5	New Data 0x0: Rx Buffer not updated 0x1: Rx Buffer updated from new message	RW1TC	0x0
4	ND4	New Data 0x0: Rx Buffer not updated 0x1: Rx Buffer updated from new message	RW1TC	0x0
3	ND3	New Data 0x0: Rx Buffer not updated 0x1: Rx Buffer updated from new message	RW1TC	0x0

Bits	Field Name	Description	Type	Reset
2	ND2	New Data 0x0: Rx Buffer not updated 0x1: Rx Buffer updated from new message	RW1TC	0x0
1	ND1	New Data 0x0: Rx Buffer not updated 0x1: Rx Buffer updated from new message	RW1TC	0x0
0	ND0	New Data 0x0: Rx Buffer not updated 0x1: Rx Buffer updated from new message	RW1TC	0x0

Table 18-698. Register Call Summary for Register MCAN_NDAT1

MCAN
- Acceptance Filtering: [0]
- Dedicated Rx Buffers: [1]
- MCAN Register Summary:

Table 18-699. MCAN_NDAT2

Address Offset	0x0000 1A9C
Physical Address	**Instance**
Description	New Data 2 NewDat flags of dedicated Rx buffers 32-63. The register holds the New Data flags of Rx Buffers 32 to 63. The flags are set when the respective Rx Buffer has been updated from a received frame. The flags remain set until the Host CPU clears them. Aflag is cleared by writing a '1' to the corresponding bit position. Writing a '0' has no effect. Ahard reset will clear the register.
Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ND63	ND62	ND61	ND60	ND59	ND58	ND57	ND56	ND55	ND54	ND53	ND52	ND51	ND50	ND49	ND48	ND47	ND46	ND45	ND44	ND43	ND42	ND41	ND40	ND39	ND38	ND37	ND36	ND35	ND34	ND33	ND32

Bits	Field Name	Description	Type	Reset
31	ND63	New Data 0x0: Rx Buffer not updated 0x1: Rx Buffer updated from new message	RW1TC	0x0
30	ND62	New Data 0x0: Rx Buffer not updated 0x1: Rx Buffer updated from new message	RW1TC	0x0
29	ND61	New Data 0x0: Rx Buffer not updated 0x1: Rx Buffer updated from new message	RW1TC	0x0
28	ND60	New Data 0x0: Rx Buffer not updated 0x1: Rx Buffer updated from new message	RW1TC	0x0
27	ND59	New Data 0x0: Rx Buffer not updated 0x1: Rx Buffer updated from new message	RW1TC	0x0
26	ND58	New Data 0x0: Rx Buffer not updated 0x1: Rx Buffer updated from new message	RW1TC	0x0

Bits	Field Name	Description	Type	Reset
25	ND57	New Data 0x0: Rx Buffer not updated 0x1: Rx Buffer updated from new message	RW1TC	0x0
24	ND56	New Data 0x0: Rx Buffer not updated 0x1: Rx Buffer updated from new message	RW1TC	0x0
23	ND55	New Data 0x0: Rx Buffer not updated 0x1: Rx Buffer updated from new message	RW1TC	0x0
22	ND54	New Data 0x0: Rx Buffer not updated 0x1: Rx Buffer updated from new message	RW1TC	0x0
21	ND53	New Data 0x0: Rx Buffer not updated 0x1: Rx Buffer updated from new message	RW1TC	0x0
20	ND52	New Data 0x0: Rx Buffer not updated 0x1: Rx Buffer updated from new message	RW1TC	0x0
19	ND51	New Data 0x0: Rx Buffer not updated 0x1: Rx Buffer updated from new message	RW1TC	0x0
18	ND50	New Data 0x0: Rx Buffer not updated 0x1: Rx Buffer updated from new message	RW1TC	0x0
17	ND49	New Data 0x0: Rx Buffer not updated 0x1: Rx Buffer updated from new message	RW1TC	0x0
16	ND48	New Data 0x0: Rx Buffer not updated 0x1: Rx Buffer updated from new message	RW1TC	0x0
15	ND47	New Data 0x0: Rx Buffer not updated 0x1: Rx Buffer updated from new message	RW1TC	0x0
14	ND46	New Data 0x0: Rx Buffer not updated 0x1: Rx Buffer updated from new message	RW1TC	0x0
13	ND45	New Data 0x0: Rx Buffer not updated 0x1: Rx Buffer updated from new message	RW1TC	0x0
12	ND44	New Data 0x0: Rx Buffer not updated 0x1: Rx Buffer updated from new message	RW1TC	0x0
11	ND43	New Data 0x0: Rx Buffer not updated 0x1: Rx Buffer updated from new message	RW1TC	0x0
10	ND42	New Data 0x0: Rx Buffer not updated 0x1: Rx Buffer updated from new message	RW1TC	0x0
9	ND41	New Data 0x0: Rx Buffer not updated 0x1: Rx Buffer updated from new message	RW1TC	0x0

Bits	Field Name	Description	Type	Reset
8	ND40	New Data 0x0: Rx Buffer not updated 0x1: Rx Buffer updated from new message	RW1TC	0x0
7	ND39	New Data 0x0: Rx Buffer not updated 0x1: Rx Buffer updated from new message	RW1TC	0x0
6	ND38	New Data 0x0: Rx Buffer not updated 0x1: Rx Buffer updated from new message	RW1TC	0x0
5	ND37	New Data 0x0: Rx Buffer not updated 0x1: Rx Buffer updated from new message	RW1TC	0x0
4	ND36	New Data 0x0: Rx Buffer not updated 0x1: Rx Buffer updated from new message	RW1TC	0x0
3	ND35	New Data 0x0: Rx Buffer not updated 0x1: Rx Buffer updated from new message	RW1TC	0x0
2	ND34	New Data 0x0: Rx Buffer not updated 0x1: Rx Buffer updated from new message	RW1TC	0x0
1	ND33	New Data 0x0: Rx Buffer not updated 0x1: Rx Buffer updated from new message	RW1TC	0x0
0	ND32	New Data 0x0: Rx Buffer not updated 0x1: Rx Buffer updated from new message	RW1TC	0x0

Table 18-700. Register Call Summary for Register MCAN_NDAT2

MCAN
- Acceptance Filtering: [0]
- Dedicated Rx Buffers: [1]
- MCAN Register Summary:

Table 18-701. MCAN_RXF0C

Address Offset	0x0000 1AA0
Physical Address	**Instance**
Description	Rx FIFO 0 Configuration FIFO 0 operation mode, watermark, size and start address.
Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
F0OM	F0WM							RESERVED	F0S							F0SA														RESERVED	

Bits	Field Name	Description	Type	Reset
31	F0OM	FIFO 0 Operation Mode FIFO 0 can be operated in blocking or in overwrite mode (see Section 18.7.4.8.2). 0x0: FIFO 0 blocking mode 0x1: FIFO 0 overwrite mode	RW	0x0
30:24	F0WM	Rx FIFO 0 Watermark 0x0: Watermark interrupt disabled 0x1-0x40 (1-64): Level for Rx FIFO 0 watermark interrupt (MCAN_IR[1] RF0W) > 0x40 (64): Watermark interrupt disabled	RW	0x0
23	RESERVED	Reserved	R	0x0
22:16	F0S	Rx FIFO 0 Size 0x0: No Rx FIFO 0 0x1-0x40 (1-64): Number of Rx FIFO 0 elements > 0x40 (64): Values greater than 64 are interpreted as 64 The Rx FIFO 0 elements are indexed from 0 to MCAN_RXF0C[22:16] F0S - 1	RW	0x0
15:2	F0SA	Rx FIFO 0 Start Address Start address of Rx FIFO 0 in Message RAM (32-bit word address, see Figure 18-148).	RW	0x0
1:0	RESERVED	Reserved	R	0x0

Table 18-702. Register Call Summary for Register MCAN_RXF0C

MCAN
- Rx FIFOs: [0]
- MCAN Register Summary:
- MCAN Register Description: [3][4]

Table 18-703. MCAN_RXF0S

Address Offset	0x0000 1AA4
Physical Address	**Instance**
Description	Rx FIFO 0 Status FIFO 0 message lost/full indication, put index, get index and fill level.
Type	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED						RF0L	F0F	RESERVED		F0PI						RESERVED		F0GI						RESERVED	F0FL						

Bits	Field Name	Description	Type	Reset
31:26	RESERVED	Reserved	R	0x0
25	RF0L	Rx FIFO 0 Message Lost This bit is a copy of interrupt flag MCAN_IR[3] RF0L. When the MCAN_IR[3] RF0L flag is reset, this bit is also reset. 0x0: No Rx FIFO 0 message lost 0x1: Rx FIFO 0 message lost, also set after write attempt to Rx FIFO 0 of size zero **Note:** Overwriting the oldest message when the MCAN_RXF0C[31] F0OM = 1 will not set this flag.	R	0x0

Bits	Field Name	Description	Type	Reset
24	F0F	Rx FIFO 0 Full 0x0: Rx FIFO 0 not full 0x1: Rx FIFO 0 full	R	0x0
23:22	RESERVED	Reserved	R	0x0
21:16	F0PI	Rx FIFO 0 Put Index Rx FIFO 0 write index pointer, range 0 to 63.	R	0x0
15:14	RESERVED	Reserved	R	0x0
13:8	F0GI	Rx FIFO 0 Get Index Rx FIFO 0 read index pointer, range 0 to 63.	R	0x0
7	RESERVED	Reserved	R	0x0
6:0	F0FL	Rx FIFO 0 Fill Level Number of elements stored in Rx FIFO 0, range 0 to 64.	R	0x0

Table 18-704. Register Call Summary for Register MCAN_RXF0S

MCAN
- Software Initialization: [0]
- MCAN Register Summary:
- MCAN Register Description: [3][4]

Table 18-705. MCAN_RXF0A

Address Offset	0x0000 1AA8
Physical Address	**Instance**
Description	Rx FIFO 0 Acknowledge FIFO 0 acknowledge last index of read buffers, updates get index and fill level.
Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								F0AI							

Bits	Field Name	Description	Type	Reset
31:6	RESERVED	Reserved	R	0x0
5:0	F0AI	Rx FIFO 0 Acknowledge Index After the Host CPU has read a message or a sequence of messages from Rx FIFO 0 it has to write the buffer index of the last element read from Rx FIFO 0 to the MCAN_RXF0A[5:0] F0AI field. This will set the Rx FIFO 0 Get Index MCAN_RXF0S[13:8] F0GI field to the MCAN_RXF0A[5:0] F0AI field + 1 and update the FIFO 0 Fill Level MCAN_RXF0S[6:0] F0FL field.	RW	0x0

Table 18-706. Register Call Summary for Register MCAN_RXF0A

MCAN
- FIFO Acknowledge Handling: [0]
- MCAN Register Summary:
- MCAN Register Description: [3][4]

Table 18-707. MCAN_RXBC

Address Offset	0x0000 1AAC
Physical Address	**Instance**
Description	Rx Buffer Configuration Start address of Rx buffer section.
Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RBSA														RESERVED	

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Reserved	R	0x0
15:2	RBSA	Rx Buffer Start Address Configures the start address of the Rx Buffers section in the Message RAM (32-bit word address, see Figure 18-148). Also used to reference debug messages A,B,C. **Note:** Debug feature is not supported.	RW	0x0
1:0	RESERVED	Reserved	R	0x0

Table 18-708. Register Call Summary for Register MCAN_RXBC

MCAN

- Dedicated Rx Buffers: [0]
- Standard Message ID Filter Element: [1]
- Extended Message ID Filter Element: [2]
- MCAN Register Summary:

Table 18-709. MCAN_RXF1C

Address Offset	0x0000 1AB0
Physical Address	**Instance**
Description	Rx FIFO 1 Configuration FIFO 1 operation mode, watermark, size and start address.
Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
F1OM	F1WM							RESERVED	F1S							F1SA														RESERVED	

Bits	Field Name	Description	Type	Reset
31	F1OM	FIFO 1 Operation Mode FIFO 1 can be operated in blocking or in overwrite mode (see Section 18.7.4.8.2). 0x0: FIFO 1 blocking mode 0x1: FIFO 1 overwrite mode	RW	0x0

Bits	Field Name	Description	Type	Reset
30:24	F1WM	Rx FIFO 1 Watermark 0x0: Watermark interrupt disabled 0x1-0x40 (1-64): Level for Rx FIFO 1 watermark interrupt (MCAN_IR[5] RF1W) > 0x40 (64): Watermark interrupt disabled	RW	0x0
23	RESERVED	Reserved	R	0x0
22:16	F1S	Rx FIFO 1 Size 0x0: No Rx FIFO 1 0x1-0x40 (1-64): Number of Rx FIFO 1 elements > 0x40 (64): Values greater than 64 are interpreted as 64 The Rx FIFO 1 elements are indexed from 0 to MCAN_RXF1C[22:16] F1S - 1	RW	0x0
15:2	F1SA	Rx FIFO 1 Start Address Start address of Rx FIFO 1 in Message RAM (32-bit word address, see Figure 18-148).	RW	0x0
1:0	RESERVED	Reserved	R	0x0

Table 18-710. Register Call Summary for Register MCAN_RXF1C

MCAN
- Rx FIFOs: [0]
- MCAN Register Summary:
- MCAN Register Description: [3][4]

Table 18-711. MCAN_RXF1S

Address Offset	0x0000 1AB4
Physical Address	**Instance**
Description	Rx FIFO 1 Status FIFO 1 message lost/full indication, put index, get index and fill level.
Type	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMS		RESERVED				RF1L	F1F	RESERVED		F1PI						RESERVED		F1GI						RESERVED		F1FL					

Bits	Field Name	Description	Type	Reset
31:30	DMS	Debug Message Status 0x0: Idle state, wait for reception of debug messages, DMA request is cleared 0x1: Debug message A received 0x2: Debug messages A, B received 0x3: Debug messages A, B, C received, DMA request is set **Note:** Debug feature is not supported.	R	0x0
29:26	RESERVED	Reserved	R	0x0

Bits	Field Name	Description	Type	Reset
25	RF1L	Rx FIFO 1 Message Lost This bit is a copy of interrupt flag MCAN_IR[7] RF1L. When the MCAN_IR[7] RF1L flag is reset, this bit is also reset. 0x0: No Rx FIFO 1 message lost 0x1: Rx FIFO 1 message lost, also set after write attempt to Rx FIFO 1 of size zero **Note:** Overwriting the oldest message when the MCAN_RXF1C[31] F1OM = 1 will not set this flag.	R	0x0
24	F1F	Rx FIFO 1 Full 0x0: Rx FIFO 1 not full 0x1: Rx FIFO 1 full	R	0x0
23:22	RESERVED	Reserved	R	0x0
21:16	F1PI	Rx FIFO 1 Put Index Rx FIFO 1 write index pointer, range 0 to 63.	R	0x0
15:14	RESERVED	Reserved	R	0x0
13:8	F1GI	Rx FIFO 1 Get Index Rx FIFO 1 read index pointer, range 0 to 63.	R	0x0
7	RESERVED	Reserved	R	0x0
6:0	F1FL	Rx FIFO 1 Fill Level Number of elements stored in Rx FIFO 1, range 0 to 64.	R	0x0

Table 18-712. Register Call Summary for Register MCAN_RXF1S

MCAN
- Software Initialization: [0]
- MCAN Register Summary:
- MCAN Register Description: [3][4]

Table 18-713. MCAN_RXF1A

Address Offset	0x0000 1AB8
Physical Address	**Instance**
Description	Rx FIFO 1 Acknowledge FIFO 1 acknowledge last index of read buffers, updates get index and fill level.
Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								F1AI							

Bits	Field Name	Description	Type	Reset
31:6	RESERVED	Reserved	R	0x0
5:0	F1AI	Rx FIFO 1 Acknowledge Index After the Host CPU has read a message or a sequence of messages from Rx FIFO 1 it has to write the buffer index of the last element read from Rx FIFO 1 to the MCAN_RXF1A[5:0] F1AI field. This will set the Rx FIFO 1 Get Index MCAN_RXF1S[13:8] F1GI field to the MCAN_RXF1A[5:0] F1AI field + 1 and update the FIFO 1 Fill Level MCAN_RXF1S[6:0] F1FL field.	RW	0x0

Table 18-714. Register Call Summary for Register MCAN_RXF1A

MCAN

- FIFO Acknowledge Handling: [0]
- MCAN Register Summary:
- MCAN Register Description: [3][4]

Table 18-715. MCAN_RXESC

Address Offset	0x0000 1ABC
Physical Address	**Instance**
Description	Rx Buffer/FIFO Element Size Configuration Configure data field size for storage of accepted frames.
Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																					RBDS			RESERVED		F1DS		RESERVED		F0DS	

Bits	Field Name	Description	Type	Reset
31:11	RESERVED	Reserved	R	0x0
10:8	RBDS	Rx Buffer Data Field Size 0x0: 8 byte data field 0x1: 12 byte data field 0x2: 16 byte data field 0x3: 20 byte data field 0x4: 24 byte data field 0x5: 32 byte data field 0x6: 48 byte data field 0x7: 64 byte data field	RW	0x0
7	RESERVED	Reserved	R	0x0
6:4	F1DS	Rx FIFO 1 Data Field Size 0x0: 8 byte data field 0x1: 12 byte data field 0x2: 16 byte data field 0x3: 20 byte data field 0x4: 24 byte data field 0x5: 32 byte data field 0x6: 48 byte data field 0x7: 64 byte data field	RW	0x0
3	RESERVED	Reserved	R	0x0

Bits	Field Name	Description	Type	Reset
2:0	F0DS	Rx FIFO 0 Data Field Size 0x0: 8 byte data field 0x1: 12 byte data field 0x2: 16 byte data field 0x3: 20 byte data field 0x4: 24 byte data field 0x5: 32 byte data field 0x6: 48 byte data field 0x7: 64 byte data field **Note:** In case the data field size of an accepted CAN frame exceeds the data field size configured for the matching Rx Buffer or Rx FIFO, only the number of bytes as configured by the MCAN_RXESC register are stored to the Rx Buffer respectively Rx FIFO element. The rest of the frame's data field is ignored.	RW	0x0

Table 18-716. Register Call Summary for Register MCAN_RXESC

MCAN
- Rx FIFOs: [0][1][2][3]
- Message RAM Configuration: [4][5][6]
- Rx Buffer and FIFO Element: [7][8]
- MCAN Register Summary:
- MCAN Register Description: [11]

Table 18-717. MCAN_TXBC

Address Offset	0x0000 1AC0
Physical Address	**Instance**
Description	Tx Buffer Configuration Configure Tx FIFO/Queue mode, Tx FIFO/Queue size, number of dedicated Tx buffers, Tx buffer start address.
Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	TFQM	TFQS						RESERVED		NDTB						TBSA														RESERVED	

Bits	Field Name	Description	Type	Reset
31	RESERVED	Reserved	R	0x0
30	TFQM	Tx FIFO/Queue Mode 0x0: Tx FIFO operation 0x1: Tx Queue operation	RW	0x0
29:24	TFQS	Transmit FIFO/Queue Size 0x0: No Tx FIFO/Queue 0x1-0x20 (1-32): Number of Tx Buffers used for Tx FIFO/Queue > 0x20 (32): Values greater than 32 are interpreted as 32	RW	0x0
23:22	RESERVED	Reserved	R	0x0
21:16	NDTB	Number of Dedicated Transmit Buffers 0x0: No Dedicated Tx Buffers 0x1-0x20 (1-32): Number of Dedicated Tx Buffers > 0x20 (32): Values greater than 32 are interpreted as 32	RW	0x0

Bits	Field Name	Description	Type	Reset
15:2	TBSA	Tx Buffers Start Address Start address of Tx Buffers section in Message RAM (32-bit word address, see Figure 18-148). **Note:** Be aware that the sum of the MCAN_TXBC[29:24] TFQS and MCAN_TXBC[21:16] NDTB fields may be not greater than 32. There is no check for erroneous configurations. The Tx Buffers section in the Message RAM starts with the dedicated Tx Buffers.	RW	0x0
1:0	RESERVED	Reserved	R	0x0

Table 18-718. Register Call Summary for Register MCAN_TXBC

MCAN

- Dedicated Tx Buffers: [0]
- Tx FIFO: [1][2]
- Tx Queue: [3][4]
- Mixed Dedicated Tx Buffers / Tx FIFO: [5][6][7]
- Mixed Dedicated Tx Buffers / Tx Queue: [8][9][10]
- Tx Buffer Element: [11][12]
- MCAN Register Summary:
- MCAN Register Description: [15][16][17][18][19][20][21]

Table 18-719. MCAN_TXFQS

Address Offset	0x0000 1AC4
Physical Address	**Instance**
Description	Tx FIFO/Queue Status Tx FIFO/Queue full indication and put index, Tx FIFO get index and fill level. The Tx FIFO/Queue status is related to the pending Tx requests listed in the MCAN_TXBRP register. Therefore the effect of Add/Cancellation requests may be delayed due to a running Tx scan (the MCAN_TXBRP register not yet updated).
Type	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED										TFQF	TFQPI					RESERVED			TFGI					RESERVED			TFFL				

Bits	Field Name	Description	Type	Reset
31:22	RESERVED	Reserved	R	0x0
21	TFQF	Tx FIFO/Queue Full 0x0: Tx FIFO/Queue not full 0x1: Tx FIFO/Queue full	R	0x0
20:16	TFQPI	Tx FIFO/Queue Put Index Tx FIFO/Queue write index pointer, range 0 to 31.	R	0x0
15:13	RESERVED	Reserved	R	0x0
12:8	TFGI	Tx FIFO Get Index Tx FIFO read index pointer, range 0 to 31. Read as zero when Tx Queue operation is configured (MCAN_TXBC[30] TFQM = 1).	R	0x0
7:6	RESERVED	Reserved	R	0x0

Bits	Field Name	Description	Type	Reset
5:0	TFFL	Tx FIFO Free Level Number of consecutive free Tx FIFO elements starting from the MCAN_TXFQS[12:8] TFGI field, range 0 to 32. Read as zero when Tx Queue operation is configured (MCAN_TXBC[30] TFQM = 1) **Note:** In case of mixed configurations where dedicated Tx Buffers are combined with a Tx FIFO or a Tx Queue, the Put and Get Indices indicate the number of the Tx Buffer starting with the first dedicated Tx Buffers. **Example:** For a configuration of 12 dedicated Tx Buffers and a Tx FIFO of 20 Buffers a Put Index of 15 points to the fourth buffer of the Tx FIFO.	R	0x0

Table 18-720. Register Call Summary for Register MCAN_TXFQS

MCAN

- Software Initialization: [0]
- Dedicated Tx Buffers: [1]
- Tx FIFO: [2][3][4][5][6][7][8][9][10]
- Tx Queue: [11][12][13]
- Mixed Dedicated Tx Buffers / Tx FIFO: [14]
- MCAN Register Summary:
- MCAN Register Description: [17]

Table 18-721. MCAN_TXESC

Address Offset	0x0000 1AC8
Physical Address	**Instance**
Description	Tx Buffer Element Size Configuration Configure data field size for frame transmission. Configures the number of data bytes belonging to a Tx Buffer element. Data field sizes > 8 bytes are intended for CAN FD operation only.
Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																													TBDS		

Bits	Field Name	Description	Type	Reset
31:3	RESERVED	Reserved	R	0x0
2:0	TBDS	Tx Buffer Data Field Size 0x0: 8 byte data field 0x1: 12 byte data field 0x2: 16 byte data field 0x3: 20 byte data field 0x4: 24 byte data field 0x5: 32 byte data field 0x6: 48 byte data field 0x7: 64 byte data field **Note:** In case the data length code DLC of a Tx Buffer element is configured to a value higher than the Tx Buffer data field size MCAN_TXESC[2:0] TBDS, the bytes not defined by the Tx Buffer are transmitted as '0xCC' (padding bytes).	RW	0x0

Table 18-722. Register Call Summary for Register MCAN_TXESC

MCAN

- Dedicated Tx Buffers: [0]
- Message RAM Configuration: [1]
- Tx Buffer Element: [2][3]
- MCAN Register Summary:
- MCAN Register Description: [6]

Table 18-723. MCAN_TXBRP

Address Offset	0x0000 1ACC
Physical Address	**Instance**
Description	Tx Buffer Request Pending Tx buffers with pending transmission request. Each Tx Buffer has its own Transmission Request Pending bit. The bits are set via the MCAN_TXBAR register. The bits are reset after a requested transmission has completed or has been cancelled via the MCAN_TXBCR register. The MCAN_TXBRP bits are set only for those Tx Buffers configured via the MCAN_TXBC register. After a MCAN_TXBRP bit has been set, a Tx scan (see Section 18.7.4.9, *Tx Handling*) is started to check for the pending Tx request with the highest priority (Tx Buffer with lowest Message ID). A cancellation request resets the corresponding transmission request pending bit of register the MCAN_TXBRP register. In case a transmission has already been started when a cancellation is requested, this is done at the end of the transmission, regardless whether the transmission was successful or not. The cancellation request bits are reset directly after the corresponding MCAN_TXBRP bit has been reset. After a cancellation has been requested, a finished cancellation is signalled via the MCAN_TXBCF • after successful transmission together with the corresponding MCAN_TXBTO bit • when the transmission has not yet been started at the point of cancellation • when the transmission has been aborted due to lost arbitration • when an error occurred during frame transmission In DAR mode all transmissions are automatically cancelled if they are not successful. The corresponding MCAN_TXBCF bit is set for all unsuccessful transmissions. **Note:** The TXBRP bits which are set while a Tx scan is in progress are not considered during this particular Tx scan. In case a cancellation is requested for such a Tx Buffer, this Add Request is cancelled immediately, the corresponding MCAN_TXBRP bit is reset.
Type	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TRP31	TRP30	TRP29	TRP28	TRP27	TRP26	TRP25	TRP24	TRP23	TRP22	TRP21	TRP20	TRP19	TRP18	TRP17	TRP16	TRP15	TRP14	TRP13	TRP12	TRP11	TRP10	TRP9	TRP8	TRP7	TRP6	TRP5	TRP4	TRP3	TRP2	TRP1	TRP0

Bits	Field Name	Description	Type	Reset
31	TRP31	Transmission Request Pending 0x0: No transmission request pending 0x1: Transmission request pending	R	0x0
30	TRP30	Transmission Request Pending 0x0: No transmission request pending 0x1: Transmission request pending	R	0x0
29	TRP29	Transmission Request Pending 0x0: No transmission request pending 0x1: Transmission request pending	R	0x0
28	TRP28	Transmission Request Pending 0x0: No transmission request pending 0x1: Transmission request pending	R	0x0

Bits	Field Name	Description	Type	Reset
27	TRP27	Transmission Request Pending 0x0: No transmission request pending 0x1: Transmission request pending	R	0x0
26	TRP26	Transmission Request Pending 0x0: No transmission request pending 0x1: Transmission request pending	R	0x0
25	TRP25	Transmission Request Pending 0x0: No transmission request pending 0x1: Transmission request pending	R	0x0
24	TRP24	Transmission Request Pending 0x0: No transmission request pending 0x1: Transmission request pending	R	0x0
23	TRP23	Transmission Request Pending 0x0: No transmission request pending 0x1: Transmission request pending	R	0x0
22	TRP22	Transmission Request Pending 0x0: No transmission request pending 0x1: Transmission request pending	R	0x0
21	TRP21	Transmission Request Pending 0x0: No transmission request pending 0x1: Transmission request pending	R	0x0
20	TRP20	Transmission Request Pending 0x0: No transmission request pending 0x1: Transmission request pending	R	0x0
19	TRP19	Transmission Request Pending 0x0: No transmission request pending 0x1: Transmission request pending	R	0x0
18	TRP18	Transmission Request Pending 0x0: No transmission request pending 0x1: Transmission request pending	R	0x0
17	TRP17	Transmission Request Pending 0x0: No transmission request pending 0x1: Transmission request pending	R	0x0
16	TRP16	Transmission Request Pending 0x0: No transmission request pending 0x1: Transmission request pending	R	0x0
15	TRP15	Transmission Request Pending 0x0: No transmission request pending 0x1: Transmission request pending	R	0x0
14	TRP14	Transmission Request Pending 0x0: No transmission request pending 0x1: Transmission request pending	R	0x0
13	TRP13	Transmission Request Pending 0x0: No transmission request pending 0x1: Transmission request pending	R	0x0
12	TRP12	Transmission Request Pending 0x0: No transmission request pending 0x1: Transmission request pending	R	0x0
11	TRP11	Transmission Request Pending 0x0: No transmission request pending 0x1: Transmission request pending	R	0x0

Bits	Field Name	Description	Type	Reset
10	TRP10	Transmission Request Pending 0x0: No transmission request pending 0x1: Transmission request pending	R	0x0
9	TRP9	Transmission Request Pending 0x0: No transmission request pending 0x1: Transmission request pending	R	0x0
8	TRP8	Transmission Request Pending 0x0: No transmission request pending 0x1: Transmission request pending	R	0x0
7	TRP7	Transmission Request Pending 0x0: No transmission request pending 0x1: Transmission request pending	R	0x0
6	TRP6	Transmission Request Pending 0x0: No transmission request pending 0x1: Transmission request pending	R	0x0
5	TRP5	Transmission Request Pending 0x0: No transmission request pending 0x1: Transmission request pending	R	0x0
4	TRP4	Transmission Request Pending 0x0: No transmission request pending 0x1: Transmission request pending	R	0x0
3	TRP3	Transmission Request Pending 0x0: No transmission request pending 0x1: Transmission request pending	R	0x0
2	TRP2	Transmission Request Pending 0x0: No transmission request pending 0x1: Transmission request pending	R	0x0
1	TRP1	Transmission Request Pending 0x0: No transmission request pending 0x1: Transmission request pending	R	0x0
0	TRP0	Transmission Request Pending 0x0: No transmission request pending 0x1: Transmission request pending	R	0x0

Table 18-724. Register Call Summary for Register MCAN_TXBRP

MCAN

- Software Initialization: [0]
- Bus Monitoring Mode: [1]
- Disabled Automatic Retransmission (DAR) Mode: [2]
- Tx Handling: [3]
- Tx Queue: [4]
- Transmit Cancellation: [5]
- MCAN Register Summary:
- MCAN Register Description: [8][9][10][11][12][13][14][15][16][17][18][19][20]

Table 18-725. MCAN_TXBAR

Address Offset	0x0000 1AD0
Physical Address	**Instance**
Description	Tx Buffer Add Request Add transmission requests. Each Tx Buffer has its own Add Request bit. Writing a '1' will set the corresponding Add Request bit; writing a '0' has no impact. This enables the Host CPU to set transmission requests for multiple Tx Buffers with one write to the MCAN_TXBAR register. The MCAN_TXBAR bits are set only for those Tx Buffers configured via the MCAN_TXBC register. When no Tx scan is running, the bits are reset immediately, else the bits remain set until the Tx scan process has completed. **Note:** If an add request is applied for a Tx Buffer with pending transmission request (corresponding MCAN_TXBRP bit already set), this add request is ignored.
Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
AR31	AR30	AR29	AR28	AR27	AR26	AR25	AR24	AR23	AR22	AR21	AR20	AR19	AR18	AR17	AR16	AR15	AR14	AR13	AR12	AR11	AR10	AR9	AR8	AR7	AR6	AR5	AR4	AR3	AR2	AR1	AR0

Bits	Field Name	Description	Type	Reset
31	AR31	Add Request 0x0: No transmission request added 0x1: Transmission requested added	RW1TC	0x0
30	AR30	Add Request 0x0: No transmission request added 0x1: Transmission requested added	RW1TC	0x0
29	AR29	Add Request 0x0: No transmission request added 0x1: Transmission requested added	RW1TC	0x0
28	AR28	Add Request 0x0: No transmission request added 0x1: Transmission requested added	RW1TC	0x0
27	AR27	Add Request 0x0: No transmission request added 0x1: Transmission requested added	RW1TC	0x0
26	AR26	Add Request 0x0: No transmission request added 0x1: Transmission requested added	RW1TC	0x0
25	AR25	Add Request 0x0: No transmission request added 0x1: Transmission requested added	RW1TC	0x0
24	AR24	Add Request 0x0: No transmission request added 0x1: Transmission requested added	RW1TC	0x0
23	AR23	Add Request 0x0: No transmission request added 0x1: Transmission requested added	RW1TC	0x0
22	AR22	Add Request 0x0: No transmission request added 0x1: Transmission requested added	RW1TC	0x0
21	AR21	Add Request 0x0: No transmission request added 0x1: Transmission requested added	RW1TC	0x0

Bits	Field Name	Description	Type	Reset
20	AR20	Add Request 0x0: No transmission request added 0x1: Transmission requested added	RW1TC	0x0
19	AR19	Add Request 0x0: No transmission request added 0x1: Transmission requested added	RW1TC	0x0
18	AR18	Add Request 0x0: No transmission request added 0x1: Transmission requested added	RW1TC	0x0
17	AR17	Add Request 0x0: No transmission request added 0x1: Transmission requested added	RW1TC	0x0
16	AR16	Add Request 0x0: No transmission request added 0x1: Transmission requested added	RW1TC	0x0
15	AR15	Add Request 0x0: No transmission request added 0x1: Transmission requested added	RW1TC	0x0
14	AR14	Add Request 0x0: No transmission request added 0x1: Transmission requested added	RW1TC	0x0
13	AR13	Add Request 0x0: No transmission request added 0x1: Transmission requested added	RW1TC	0x0
12	AR12	Add Request 0x0: No transmission request added 0x1: Transmission requested added	RW1TC	0x0
11	AR11	Add Request 0x0: No transmission request added 0x1: Transmission requested added	RW1TC	0x0
10	AR10	Add Request 0x0: No transmission request added 0x1: Transmission requested added	RW1TC	0x0
9	AR9	Add Request 0x0: No transmission request added 0x1: Transmission requested added	RW1TC	0x0
8	AR8	Add Request 0x0: No transmission request added 0x1: Transmission requested added	RW1TC	0x0
7	AR7	Add Request 0x0: No transmission request added 0x1: Transmission requested added	RW1TC	0x0
6	AR6	Add Request 0x0: No transmission request added 0x1: Transmission requested added	RW1TC	0x0
5	AR5	Add Request 0x0: No transmission request added 0x1: Transmission requested added	RW1TC	0x0
4	AR4	Add Request 0x0: No transmission request added 0x1: Transmission requested added	RW1TC	0x0

Bits	Field Name	Description	Type	Reset
3	AR3	Add Request 0x0: No transmission request added 0x1: Transmission requested added	RW1TC	0x0
2	AR2	Add Request 0x0: No transmission request added 0x1: Transmission requested added	RW1TC	0x0
1	AR1	Add Request 0x0: No transmission request added 0x1: Transmission requested added	RW1TC	0x0
0	AR0	Add Request 0x0: No transmission request added 0x1: Transmission requested added	RW1TC	0x0

Table 18-726. Register Call Summary for Register MCAN_TXBAR

MCAN
- Software Initialization: [0]
- Dedicated Tx Buffers: [1]
- Tx FIFO: [2]
- MCAN Register Summary:
- MCAN Register Description: [5][6][7][8][9]

Table 18-727. MCAN_TXBCR

Address Offset	0x0000 1AD4
Physical Address	**Instance**
Description	Tx Buffer Cancellation Request Request cancellation of pending transmissions. Each Tx Buffer has its own Cancellation Request bit. Writing a '1' will set the corresponding Cancellation Request bit; writing a '0' has no impact. This enables the Host CPU to set cancellation requests for multiple Tx Buffers with one write to the MCAN_TXBCR register. The MCAN_TXBCR bits are set only for those Tx Buffers configured via the MCAN_TXBC register. The bits remain set until the corresponding bit of the MCAN_TXBRP register is reset.
Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CR31	CR30	CR29	CR28	CR27	CR26	CR25	CR24	CR23	CR22	CR21	CR20	CR19	CR18	CR17	CR16	CR15	CR14	CR13	CR12	CR11	CR10	CR9	CR8	CR7	CR6	CR5	CR4	CR3	CR2	CR1	CR0

Bits	Field Name	Description	Type	Reset
31	CR31	Cancellation Request 0x0: No cancellation pending 0x1: Cancellation pending	RW1TC	0x0
30	CR30	Cancellation Request 0x0: No cancellation pending 0x1: Cancellation pending	RW1TC	0x0
29	CR29	Cancellation Request 0x0: No cancellation pending 0x1: Cancellation pending	RW1TC	0x0
28	CR28	Cancellation Request 0x0: No cancellation pending 0x1: Cancellation pending	RW1TC	0x0

Bits	Field Name	Description	Type	Reset
27	CR27	Cancellation Request 0x0: No cancellation pending 0x1: Cancellation pending	RW1TC	0x0
26	CR26	Cancellation Request 0x0: No cancellation pending 0x1: Cancellation pending	RW1TC	0x0
25	CR25	Cancellation Request 0x0: No cancellation pending 0x1: Cancellation pending	RW1TC	0x0
24	CR24	Cancellation Request 0x0: No cancellation pending 0x1: Cancellation pending	RW1TC	0x0
23	CR23	Cancellation Request 0x0: No cancellation pending 0x1: Cancellation pending	RW1TC	0x0
22	CR22	Cancellation Request 0x0: No cancellation pending 0x1: Cancellation pending	RW1TC	0x0
21	CR21	Cancellation Request 0x0: No cancellation pending 0x1: Cancellation pending	RW1TC	0x0
20	CR20	Cancellation Request 0x0: No cancellation pending 0x1: Cancellation pending	RW1TC	0x0
19	CR19	Cancellation Request 0x0: No cancellation pending 0x1: Cancellation pending	RW1TC	0x0
18	CR18	Cancellation Request 0x0: No cancellation pending 0x1: Cancellation pending	RW1TC	0x0
17	CR17	Cancellation Request 0x0: No cancellation pending 0x1: Cancellation pending	RW1TC	0x0
16	CR16	Cancellation Request 0x0: No cancellation pending 0x1: Cancellation pending	RW1TC	0x0
15	CR15	Cancellation Request 0x0: No cancellation pending 0x1: Cancellation pending	RW1TC	0x0
14	CR14	Cancellation Request 0x0: No cancellation pending 0x1: Cancellation pending	RW1TC	0x0
13	CR13	Cancellation Request 0x0: No cancellation pending 0x1: Cancellation pending	RW1TC	0x0
12	CR12	Cancellation Request 0x0: No cancellation pending 0x1: Cancellation pending	RW1TC	0x0
11	CR11	Cancellation Request 0x0: No cancellation pending 0x1: Cancellation pending	RW1TC	0x0

Bits	Field Name	Description	Type	Reset
10	CR10	Cancellation Request 0x0: No cancellation pending 0x1: Cancellation pending	RW1TC	0x0
9	CR9	Cancellation Request 0x0: No cancellation pending 0x1: Cancellation pending	RW1TC	0x0
8	CR8	Cancellation Request 0x0: No cancellation pending 0x1: Cancellation pending	RW1TC	0x0
7	CR7	Cancellation Request 0x0: No cancellation pending 0x1: Cancellation pending	RW1TC	0x0
6	CR6	Cancellation Request 0x0: No cancellation pending 0x1: Cancellation pending	RW1TC	0x0
5	CR5	Cancellation Request 0x0: No cancellation pending 0x1: Cancellation pending	RW1TC	0x0
4	CR4	Cancellation Request 0x0: No cancellation pending 0x1: Cancellation pending	RW1TC	0x0
3	CR3	Cancellation Request 0x0: No cancellation pending 0x1: Cancellation pending	RW1TC	0x0
2	CR2	Cancellation Request 0x0: No cancellation pending 0x1: Cancellation pending	RW1TC	0x0
1	CR1	Cancellation Request 0x0: No cancellation pending 0x1: Cancellation pending	RW1TC	0x0
0	CR0	Cancellation Request 0x0: No cancellation pending 0x1: Cancellation pending	RW1TC	0x0

Table 18-728. Register Call Summary for Register MCAN_TXBCR

MCAN

- Software Initialization: [0]
- Transmit Cancellation: [1]
- MCAN Register Summary:
- MCAN Register Description: [4][5][6][7]

Table 18-729. MCAN_TXBTO

Address Offset	0x0000 1AD8
Physical Address	**Instance**
Description	Tx Buffer Transmission Occurred Signals successful transmissions, set when corresponding MCAN_TXBRP flag is cleared. Each Tx Buffer has its own Transmission Occurred bit. The bits are set when the corresponding MCAN_TXBRP bit is cleared after a successful transmission. The bits are reset when a new transmission is requested by writing a '1' to the corresponding bit of register the MCAN_TXBAR register.
Type	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TO31	TO30	TO29	TO28	TO27	TO26	TO25	TO24	TO23	TO22	TO21	TO20	TO19	TO18	TO17	TO16	TO15	TO14	TO13	TO12	TO11	TO10	TO9	TO8	TO7	TO6	TO5	TO4	TO3	TO2	TO1	TO0

Bits	Field Name	Description	Type	Reset
31	TO31	Transmission Occurred 0x0: No transmission occurred 0x1: Transmission occurred	R	0x0
30	TO30	Transmission Occurred 0x0: No transmission occurred 0x1: Transmission occurred	R	0x0
29	TO29	Transmission Occurred 0x0: No transmission occurred 0x1: Transmission occurred	R	0x0
28	TO28	Transmission Occurred 0x0: No transmission occurred 0x1: Transmission occurred	R	0x0
27	TO27	Transmission Occurred 0x0: No transmission occurred 0x1: Transmission occurred	R	0x0
26	TO26	Transmission Occurred 0x0: No transmission occurred 0x1: Transmission occurred	R	0x0
25	TO25	Transmission Occurred 0x0: No transmission occurred 0x1: Transmission occurred	R	0x0
24	TO24	Transmission Occurred 0x0: No transmission occurred 0x1: Transmission occurred	R	0x0
23	TO23	Transmission Occurred 0x0: No transmission occurred 0x1: Transmission occurred	R	0x0
22	TO22	Transmission Occurred 0x0: No transmission occurred 0x1: Transmission occurred	R	0x0
21	TO21	Transmission Occurred 0x0: No transmission occurred 0x1: Transmission occurred	R	0x0
20	TO20	Transmission Occurred 0x0: No transmission occurred 0x1: Transmission occurred	R	0x0

Bits	Field Name	Description	Type	Reset
19	TO19	Transmission Occurred 0x0: No transmission occurred 0x1: Transmission occurred	R	0x0
18	TO18	Transmission Occurred 0x0: No transmission occurred 0x1: Transmission occurred	R	0x0
17	TO17	Transmission Occurred 0x0: No transmission occurred 0x1: Transmission occurred	R	0x0
16	TO16	Transmission Occurred 0x0: No transmission occurred 0x1: Transmission occurred	R	0x0
15	TO15	Transmission Occurred 0x0: No transmission occurred 0x1: Transmission occurred	R	0x0
14	TO14	Transmission Occurred 0x0: No transmission occurred 0x1: Transmission occurred	R	0x0
13	TO13	Transmission Occurred 0x0: No transmission occurred 0x1: Transmission occurred	R	0x0
12	TO12	Transmission Occurred 0x0: No transmission occurred 0x1: Transmission occurred	R	0x0
11	TO11	Transmission Occurred 0x0: No transmission occurred 0x1: Transmission occurred	R	0x0
10	TO10	Transmission Occurred 0x0: No transmission occurred 0x1: Transmission occurred	R	0x0
9	TO9	Transmission Occurred 0x0: No transmission occurred 0x1: Transmission occurred	R	0x0
8	TO8	Transmission Occurred 0x0: No transmission occurred 0x1: Transmission occurred	R	0x0
7	TO7	Transmission Occurred 0x0: No transmission occurred 0x1: Transmission occurred	R	0x0
6	TO6	Transmission Occurred 0x0: No transmission occurred 0x1: Transmission occurred	R	0x0
5	TO5	Transmission Occurred 0x0: No transmission occurred 0x1: Transmission occurred	R	0x0
4	TO4	Transmission Occurred 0x0: No transmission occurred 0x1: Transmission occurred	R	0x0
3	TO3	Transmission Occurred 0x0: No transmission occurred 0x1: Transmission occurred	R	0x0

Bits	Field Name	Description	Type	Reset
2	TO2	Transmission Occurred 0x0: No transmission occurred 0x1: Transmission occurred	R	0x0
1	TO1	Transmission Occurred 0x0: No transmission occurred 0x1: Transmission occurred	R	0x0
0	TO0	Transmission Occurred 0x0: No transmission occurred 0x1: Transmission occurred	R	0x0

Table 18-730. Register Call Summary for Register MCAN_TXBTO

MCAN
- Software Initialization: [0]
- Disabled Automatic Retransmission (DAR) Mode: [1][2][3]
- Transmit Cancellation: [4]
- MCAN Register Summary:
- MCAN Register Description: [7]

Table 18-731. MCAN_TXBCF

Address Offset	0x0000 1ADC
Physical Address	**Instance**
Description	Tx Buffer Cancellation Finished Signals successful transmit cancellation, set when corresponding TXBRP flag is cleared after cancellation request. Each Tx Buffer has its own Cancellation Finished bit. The bits are set when the corresponding MCAN_TXBRP bit is cleared after a cancellation was requested via the MCAN_TXBCR register. In case the corresponding MCAN_TXBRP bit was not set at the point of cancellation, MCAN_TXBCF[n] CF bit is set immediately. The bits are reset when a new transmission is requested by writing a '1' to the corresponding bit of the MCAN_TXBAR register.
Type	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CF31	CF30	CF29	CF28	CF27	CF26	CF25	CF24	CF23	CF22	CF21	CF20	CF19	CF18	CF17	CF16	CF15	CF14	CF13	CF12	CF11	CF10	CF9	CF8	CF7	CF6	CF5	CF4	CF3	CF2	CF1	CF0

Bits	Field Name	Description	Type	Reset
31	CF31	Cancellation Finished 0x0: No transmit buffer cancellation 0x1: Transmit buffer cancellation finished	R	0x0
30	CF30	Cancellation Finished 0x0: No transmit buffer cancellation 0x1: Transmit buffer cancellation finished	R	0x0
29	CF29	Cancellation Finished 0x0: No transmit buffer cancellation 0x1: Transmit buffer cancellation finished	R	0x0
28	CF28	Cancellation Finished 0x0: No transmit buffer cancellation 0x1: Transmit buffer cancellation finished	R	0x0
27	CF27	Cancellation Finished 0x0: No transmit buffer cancellation 0x1: Transmit buffer cancellation finished	R	0x0

Bits	Field Name	Description	Type	Reset
26	CF26	Cancellation Finished 0x0: No transmit buffer cancellation 0x1: Transmit buffer cancellation finished	R	0x0
25	CF25	Cancellation Finished 0x0: No transmit buffer cancellation 0x1: Transmit buffer cancellation finished	R	0x0
24	CF24	Cancellation Finished 0x0: No transmit buffer cancellation 0x1: Transmit buffer cancellation finished	R	0x0
23	CF23	Cancellation Finished 0x0: No transmit buffer cancellation 0x1: Transmit buffer cancellation finished	R	0x0
22	CF22	Cancellation Finished 0x0: No transmit buffer cancellation 0x1: Transmit buffer cancellation finished	R	0x0
21	CF21	Cancellation Finished 0x0: No transmit buffer cancellation 0x1: Transmit buffer cancellation finished	R	0x0
20	CF20	Cancellation Finished 0x0: No transmit buffer cancellation 0x1: Transmit buffer cancellation finished	R	0x0
19	CF19	Cancellation Finished 0x0: No transmit buffer cancellation 0x1: Transmit buffer cancellation finished	R	0x0
18	CF18	Cancellation Finished 0x0: No transmit buffer cancellation 0x1: Transmit buffer cancellation finished	R	0x0
17	CF17	Cancellation Finished 0x0: No transmit buffer cancellation 0x1: Transmit buffer cancellation finished	R	0x0
16	CF16	Cancellation Finished 0x0: No transmit buffer cancellation 0x1: Transmit buffer cancellation finished	R	0x0
15	CF15	Cancellation Finished 0x0: No transmit buffer cancellation 0x1: Transmit buffer cancellation finished	R	0x0
14	CF14	Cancellation Finished 0x0: No transmit buffer cancellation 0x1: Transmit buffer cancellation finished	R	0x0
13	CF13	Cancellation Finished 0x0: No transmit buffer cancellation 0x1: Transmit buffer cancellation finished	R	0x0
12	CF12	Cancellation Finished 0x0: No transmit buffer cancellation 0x1: Transmit buffer cancellation finished	R	0x0
11	CF11	Cancellation Finished 0x0: No transmit buffer cancellation 0x1: Transmit buffer cancellation finished	R	0x0
10	CF10	Cancellation Finished 0x0: No transmit buffer cancellation 0x1: Transmit buffer cancellation finished	R	0x0

Bits	Field Name	Description	Type	Reset
9	CF9	Cancellation Finished 0x0: No transmit buffer cancellation 0x1: Transmit buffer cancellation finished	R	0x0
8	CF8	Cancellation Finished 0x0: No transmit buffer cancellation 0x1: Transmit buffer cancellation finished	R	0x0
7	CF7	Cancellation Finished 0x0: No transmit buffer cancellation 0x1: Transmit buffer cancellation finished	R	0x0
6	CF6	Cancellation Finished 0x0: No transmit buffer cancellation 0x1: Transmit buffer cancellation finished	R	0x0
5	CF5	Cancellation Finished 0x0: No transmit buffer cancellation 0x1: Transmit buffer cancellation finished	R	0x0
4	CF4	Cancellation Finished 0x0: No transmit buffer cancellation 0x1: Transmit buffer cancellation finished	R	0x0
3	CF3	Cancellation Finished 0x0: No transmit buffer cancellation 0x1: Transmit buffer cancellation finished	R	0x0
2	CF2	Cancellation Finished 0x0: No transmit buffer cancellation 0x1: Transmit buffer cancellation finished	R	0x0
1	CF1	Cancellation Finished 0x0: No transmit buffer cancellation 0x1: Transmit buffer cancellation finished	R	0x0
0	CF0	Cancellation Finished 0x0: No transmit buffer cancellation 0x1: Transmit buffer cancellation finished	R	0x0

Table 18-732. Register Call Summary for Register MCAN_TXBCF

MCAN

- Software Initialization: [0]
- Disabled Automatic Retransmission (DAR) Mode: [1][2][3]
- Transmit Cancellation: [4][5][6][7]
- MCAN Register Summary:
- MCAN Register Description: [10][11][12]

Table 18-733. MCAN_TXBTIE

Address Offset	0x0000 1AE0
Physical Address	**Instance**
Description	Tx Buffer Transmission Interrupt Enable Enable transmit interrupts for selected Tx buffers.
Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TIE31	TIE30	TIE29	TIE28	TIE27	TIE26	TIE25	TIE24	TIE23	TIE22	TIE21	TIE20	TIE19	TIE18	TIE17	TIE16	TIE15	TIE14	TIE13	TIE12	TIE11	TIE10	TIE9	TIE8	TIE7	TIE6	TIE5	TIE4	TIE3	TIE2	TIE1	TIE0

Bits	Field Name	Description	Type	Reset
31	TIE31	Transmission Interrupt Enable 0x0: Transmission interrupt disabled 0x1: Transmission interrupt enable	RW	0x0
30	TIE30	Transmission Interrupt Enable 0x0: Transmission interrupt disabled 0x1: Transmission interrupt enable	RW	0x0
29	TIE29	Transmission Interrupt Enable 0x0: Transmission interrupt disabled 0x1: Transmission interrupt enable	RW	0x0
28	TIE28	Transmission Interrupt Enable 0x0: Transmission interrupt disabled 0x1: Transmission interrupt enable	RW	0x0
27	TIE27	Transmission Interrupt Enable 0x0: Transmission interrupt disabled 0x1: Transmission interrupt enable	RW	0x0
26	TIE26	Transmission Interrupt Enable 0x0: Transmission interrupt disabled 0x1: Transmission interrupt enable	RW	0x0
25	TIE25	Transmission Interrupt Enable 0x0: Transmission interrupt disabled 0x1: Transmission interrupt enable	RW	0x0
24	TIE24	Transmission Interrupt Enable 0x0: Transmission interrupt disabled 0x1: Transmission interrupt enable	RW	0x0
23	TIE23	Transmission Interrupt Enable 0x0: Transmission interrupt disabled 0x1: Transmission interrupt enable	RW	0x0
22	TIE22	Transmission Interrupt Enable 0x0: Transmission interrupt disabled 0x1: Transmission interrupt enable	RW	0x0
21	TIE21	Transmission Interrupt Enable 0x0: Transmission interrupt disabled 0x1: Transmission interrupt enable	RW	0x0
20	TIE20	Transmission Interrupt Enable 0x0: Transmission interrupt disabled 0x1: Transmission interrupt enable	RW	0x0
19	TIE19	Transmission Interrupt Enable 0x0: Transmission interrupt disabled 0x1: Transmission interrupt enable	RW	0x0
18	TIE18	Transmission Interrupt Enable 0x0: Transmission interrupt disabled 0x1: Transmission interrupt enable	RW	0x0
17	TIE17	Transmission Interrupt Enable 0x0: Transmission interrupt disabled 0x1: Transmission interrupt enable	RW	0x0
16	TIE16	Transmission Interrupt Enable 0x0: Transmission interrupt disabled 0x1: Transmission interrupt enable	RW	0x0
15	TIE15	Transmission Interrupt Enable 0x0: Transmission interrupt disabled 0x1: Transmission interrupt enable	RW	0x0

Bits	Field Name	Description	Type	Reset
14	TIE14	Transmission Interrupt Enable 0x0: Transmission interrupt disabled 0x1: Transmission interrupt enable	RW	0x0
13	TIE13	Transmission Interrupt Enable 0x0: Transmission interrupt disabled 0x1: Transmission interrupt enable	RW	0x0
12	TIE12	Transmission Interrupt Enable 0x0: Transmission interrupt disabled 0x1: Transmission interrupt enable	RW	0x0
11	TIE11	Transmission Interrupt Enable 0x0: Transmission interrupt disabled 0x1: Transmission interrupt enable	RW	0x0
10	TIE10	Transmission Interrupt Enable 0x0: Transmission interrupt disabled 0x1: Transmission interrupt enable	RW	0x0
9	TIE9	Transmission Interrupt Enable 0x0: Transmission interrupt disabled 0x1: Transmission interrupt enable	RW	0x0
8	TIE8	Transmission Interrupt Enable 0x0: Transmission interrupt disabled 0x1: Transmission interrupt enable	RW	0x0
7	TIE7	Transmission Interrupt Enable 0x0: Transmission interrupt disabled 0x1: Transmission interrupt enable	RW	0x0
6	TIE6	Transmission Interrupt Enable 0x0: Transmission interrupt disabled 0x1: Transmission interrupt enable	RW	0x0
5	TIE5	Transmission Interrupt Enable 0x0: Transmission interrupt disabled 0x1: Transmission interrupt enable	RW	0x0
4	TIE4	Transmission Interrupt Enable 0x0: Transmission interrupt disabled 0x1: Transmission interrupt enable	RW	0x0
3	TIE3	Transmission Interrupt Enable 0x0: Transmission interrupt disabled 0x1: Transmission interrupt enable	RW	0x0
2	TIE2	Transmission Interrupt Enable 0x0: Transmission interrupt disabled 0x1: Transmission interrupt enable	RW	0x0
1	TIE1	Transmission Interrupt Enable 0x0: Transmission interrupt disabled 0x1: Transmission interrupt enable	RW	0x0
0	TIE0	Transmission Interrupt Enable 0x0: Transmission interrupt disabled 0x1: Transmission interrupt enable	RW	0x0

Table 18-734. Register Call Summary for Register MCAN_TXBTIE

MCAN

- MCAN Register Summary:

Table 18-735. MCAN_TXBCIE

Address Offset	0x0000 1AE4
Physical Address	**Instance**
Description	Tx Buffer Cancellation Finished Interrupt Enable Enable cancellation finished interrupts for selected Tx buffers.
Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CFIE31	CFIE30	CFIE29	CFIE28	CFIE27	CFIE26	CFIE25	CFIE24	CFIE23	CFIE22	CFIE21	CFIE20	CFIE19	CFIE18	CFIE17	CFIE16	CFIE15	CFIE14	CFIE13	CFIE12	CFIE11	CFIE10	CFIE9	CFIE8	CFIE7	CFIE6	CFIE5	CFIE4	CFIE3	CFIE2	CFIE1	CFIE0

Bits	Field Name	Description	Type	Reset
31	CFIE31	Cancellation Finished Interrupt Enable 0x0: Cancellation finished interrupt disabled 0x1: Cancellation finished interrupt enabled	RW	0x0
30	CFIE30	Cancellation Finished Interrupt Enable 0x0: Cancellation finished interrupt disabled 0x1: Cancellation finished interrupt enabled	RW	0x0
29	CFIE29	Cancellation Finished Interrupt Enable 0x0: Cancellation finished interrupt disabled 0x1: Cancellation finished interrupt enabled	RW	0x0
28	CFIE28	Cancellation Finished Interrupt Enable 0x0: Cancellation finished interrupt disabled 0x1: Cancellation finished interrupt enabled	RW	0x0
27	CFIE27	Cancellation Finished Interrupt Enable 0x0: Cancellation finished interrupt disabled 0x1: Cancellation finished interrupt enabled	RW	0x0
26	CFIE26	Cancellation Finished Interrupt Enable 0x0: Cancellation finished interrupt disabled 0x1: Cancellation finished interrupt enabled	RW	0x0
25	CFIE25	Cancellation Finished Interrupt Enable 0x0: Cancellation finished interrupt disabled 0x1: Cancellation finished interrupt enabled	RW	0x0
24	CFIE24	Cancellation Finished Interrupt Enable 0x0: Cancellation finished interrupt disabled 0x1: Cancellation finished interrupt enabled	RW	0x0
23	CFIE23	Cancellation Finished Interrupt Enable 0x0: Cancellation finished interrupt disabled 0x1: Cancellation finished interrupt enabled	RW	0x0
22	CFIE22	Cancellation Finished Interrupt Enable 0x0: Cancellation finished interrupt disabled 0x1: Cancellation finished interrupt enabled	RW	0x0
21	CFIE21	Cancellation Finished Interrupt Enable 0x0: Cancellation finished interrupt disabled 0x1: Cancellation finished interrupt enabled	RW	0x0
20	CFIE20	Cancellation Finished Interrupt Enable 0x0: Cancellation finished interrupt disabled 0x1: Cancellation finished interrupt enabled	RW	0x0
19	CFIE19	Cancellation Finished Interrupt Enable 0x0: Cancellation finished interrupt disabled 0x1: Cancellation finished interrupt enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
18	CFIE18	Cancellation Finished Interrupt Enable 0x0: Cancellation finished interrupt disabled 0x1: Cancellation finished interrupt enabled	RW	0x0
17	CFIE17	Cancellation Finished Interrupt Enable 0x0: Cancellation finished interrupt disabled 0x1: Cancellation finished interrupt enabled	RW	0x0
16	CFIE16	Cancellation Finished Interrupt Enable 0x0: Cancellation finished interrupt disabled 0x1: Cancellation finished interrupt enabled	RW	0x0
15	CFIE15	Cancellation Finished Interrupt Enable 0x0: Cancellation finished interrupt disabled 0x1: Cancellation finished interrupt enabled	RW	0x0
14	CFIE14	Cancellation Finished Interrupt Enable 0x0: Cancellation finished interrupt disabled 0x1: Cancellation finished interrupt enabled	RW	0x0
13	CFIE13	Cancellation Finished Interrupt Enable 0x0: Cancellation finished interrupt disabled 0x1: Cancellation finished interrupt enabled	RW	0x0
12	CFIE12	Cancellation Finished Interrupt Enable 0x0: Cancellation finished interrupt disabled 0x1: Cancellation finished interrupt enabled	RW	0x0
11	CFIE11	Cancellation Finished Interrupt Enable 0x0: Cancellation finished interrupt disabled 0x1: Cancellation finished interrupt enabled	RW	0x0
10	CFIE10	Cancellation Finished Interrupt Enable 0x0: Cancellation finished interrupt disabled 0x1: Cancellation finished interrupt enabled	RW	0x0
9	CFIE9	Cancellation Finished Interrupt Enable 0x0: Cancellation finished interrupt disabled 0x1: Cancellation finished interrupt enabled	RW	0x0
8	CFIE8	Cancellation Finished Interrupt Enable 0x0: Cancellation finished interrupt disabled 0x1: Cancellation finished interrupt enabled	RW	0x0
7	CFIE7	Cancellation Finished Interrupt Enable 0x0: Cancellation finished interrupt disabled 0x1: Cancellation finished interrupt enabled	RW	0x0
6	CFIE6	Cancellation Finished Interrupt Enable 0x0: Cancellation finished interrupt disabled 0x1: Cancellation finished interrupt enabled	RW	0x0
5	CFIE5	Cancellation Finished Interrupt Enable 0x0: Cancellation finished interrupt disabled 0x1: Cancellation finished interrupt enabled	RW	0x0
4	CFIE4	Cancellation Finished Interrupt Enable 0x0: Cancellation finished interrupt disabled 0x1: Cancellation finished interrupt enabled	RW	0x0
3	CFIE3	Cancellation Finished Interrupt Enable 0x0: Cancellation finished interrupt disabled 0x1: Cancellation finished interrupt enabled	RW	0x0
2	CFIE2	Cancellation Finished Interrupt Enable 0x0: Cancellation finished interrupt disabled 0x1: Cancellation finished interrupt enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
1	CFIE1	Cancellation Finished Interrupt Enable 0x0: Cancellation finished interrupt disabled 0x1: Cancellation finished interrupt enabled	RW	0x0
0	CFIE0	Cancellation Finished Interrupt Enable 0x0: Cancellation finished interrupt disabled 0x1: Cancellation finished interrupt enabled	RW	0x0

Table 18-736. Register Call Summary for Register MCAN_TXBCIE

MCAN
- MCAN Register Summary:

Table 18-737. MCAN_TXEFC

Address Offset	0x0000 1AF0
Physical Address	**Instance**
Description	Tx Event FIFO Configuration Tx event FIFO watermark, size and start address.
Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED		EFWM						RESERVED		EFS						EFSA														RESERVED	

Bits	Field Name	Description	Type	Reset
31:30	RESERVED	Reserved	R	0x0
29:24	EFWM	Event FIFO Watermark 0x0: Watermark interrupt disabled 0x1-0x20 (1-32): Level for Tx Event FIFO watermark interrupt (MCAN_IR[13] TEFW) > 0x20 (32): Watermark interrupt disabled	RW	0x0
23:22	RESERVED	Reserved	R	0x0
21:16	EFS	Event FIFO Size 0x0: Tx Event FIFO disabled 0x1-0x20 (1-32): Number of Tx Event FIFO elements > 0x20 (32): Values greater than 32 are interpreted as 32 The Tx Event FIFO elements are indexed from 0 to MCAN_TXEFC[21:16] EFS field - 1	RW	0x0
15:2	EFSA	Event FIFO Start Address Start address of Tx Event FIFO in Message RAM (32-bit word address, see Figure 18-148).	RW	0x0
1:0	RESERVED	Reserved	R	0x0

Table 18-738. Register Call Summary for Register MCAN_TXEFC

MCAN
- Tx Event Handling: [0][1]
- MCAN Register Summary:
- MCAN Register Description: [4]

Table 18-739. MCAN_TXEFS

Address Offset	0x0000 1AF4
Physical Address	**Instance**
Description	Tx Event FIFO Status Tx event FIFO element lost/full indication, put index, get index, and fill level.
Type	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED						TEFL	EFF	RESERVED			EFPI					RESERVED			EFGI					RESERVED		EFFL					

Bits	Field Name	Description	Type	Reset
31:26	RESERVED	Reserved	R	0x0
25	TEFL	This bit is a copy of interrupt flag MCAN_IR[15] TEFL. When the MCAN_IR[15] TEFL flag is reset, this bit is also reset. 0x0: No Tx Event FIFO element lost 0x1: Tx Event FIFO element lost, also set after write attempt to Tx Event FIFO of size zero.	R	0x0
24	EFF	Event FIFO Full 0x0: Tx Event FIFO not full 0x1: Tx Event FIFO full	R	0x0
23:21	RESERVED	Reserved	R	0x0
20:16	EFPI	Event FIFO Put Index Tx Event FIFO write index pointer, range 0 to 31.	R	0x0
15:13	RESERVED	Reserved	R	0x0
12:8	EFGI	Event FIFO Get Index Tx Event FIFO read index pointer, range 0 to 31.	R	0x0
7:6	RESERVED	Reserved	R	0x0
5:0	EFFL	Event FIFO Fill Level Number of elements stored in Tx Event FIFO, range 0 to 32.	R	0x0

Table 18-740. Register Call Summary for Register MCAN_TXEFS

MCAN

- Software Initialization: [0]
- Tx Event Handling: [1][2]
- Tx Event FIFO Element: [3]
- MCAN Register Summary:
- MCAN Register Description: [6][7]

Table 18-741. MCAN_TXEFA

Address Offset	0x0000 1AF8
Physical Address	**Instance**
Description	Tx Event FIFO Acknowledge Tx event FIFO acknowledge last index of read elements, updates get index and fill level.
Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								EFAI							

Bits	Field Name	Description	Type	Reset
31:5	RESERVED	Reserved	R	0x0
4:0	EFAI	After the Host CPU has read an element or a sequence of elements from the Tx Event FIFO it has to write the index of the last element read from Tx Event FIFO to the MCAN_TXEFA[4:0] EFAI field. This will set the Tx Event FIFO Get Index MCAN_TXEFS[12:8] EFGI field to the MCAN_TXEFA[4:0] EFAI field + 1 and update the Event FIFO Fill Level MCAN_TXEFS[5:0] EFFL field.	RW	0x0

Table 18-742. Register Call Summary for Register MCAN_TXEFA

MCAN

- FIFO Acknowledge Handling: [0]
- MCAN Register Summary:
- MCAN Register Description: [3][4]

Table 18-743. MCANSS_ECC_AGGR_REVISION

Address Offset	0x0000 1C00
Physical Address	**Instance**
Description	Aggregator Revision Register Revision parameters.
Type	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCHEME		BU		MODULE_ID												REVRTL				REVMAJ				CUSTOM		REVMIN					

Bits	Field Name	Description	Type	Reset
31:30	SCHEME	Scheme	R	0x1
29:28	BU	Business Unit	R	0x2
27:16	MODULE_ID	Module ID	R	0x6A0
15:11	REVRTL	RTL version	R	0x1
10:8	REVMAJ	Major version	R	0x3
7:6	CUSTOM	Custom version	R	0x0
5:0	REVMIN	Minor version	R	0x0

Table 18-744. Register Call Summary for Register MCANSS_ECC_AGGR_REVISION

MCAN

- ECC Aggregator: [0]
- MCAN Register Summary:

Table 18-745. MCANSS_ECC_VECTOR

Address Offset	0x0000 1C08
Physical Address	**Instance**
Description	ECC Vector Register ECC Vector Register.
Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								RD_SVBUS_DONE	RD_SVBUS_ADDRESS							RD_SVBUS	RESERVED							ECC_VECTOR							

Bits	Field Name	Description	Type	Reset
31:25	RESERVED	Reserved	R	0x0
24	RD_SVBUS_DONE	Status to indicate if read is complete	R	0x0
23:16	RD_SVBUS_ADDRESS	Read address	RW	0x0
15	RD_SVBUS	Write 1 to trigger a read	RW	0x0
14:11	RESERVED	Reserved	R	0x0
10:0	ECC_VECTOR	Value written to select the corresponding ECC RAM for control or status	RW	0x0

Table 18-746. Register Call Summary for Register MCANSS_ECC_VECTOR

MCAN

- ECC Aggregator: [0][1][2][3][4][5][6][7][8][9]
- MCAN Register Summary:

Table 18-747. MCANSS_ECC_MISC_STATUS

Address Offset	0x0000 1C0C
Physical Address	**Instance**
Description	Misc Status Misc Status.
Type	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								NUM_RAMS							

Bits	Field Name	Description	Type	Reset
31:11	RESERVED	Reserved	R	0x0
10:0	NUM_RAMS	Indicates the number of RAMS serviced by the ECC aggregator	R	0x1

Table 18-748. Register Call Summary for Register MCANSS_ECC_MISC_STATUS

MCAN

- ECC Aggregator: [0]
- MCAN Register Summary:

Table 18-749. MCANSS_ECC_WRAP_REVISION

Address Offset	0x0000 1C10
Physical Address	**Instance**
Description	ECC Wrapper Revision Register Revision parameters.
Type	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SCHEME				BU				MODULE_ID								REVRTL					REVMAJ			CUSTOM		REVMIN					

Bits	Field Name	Description	Type	Reset
31:30	SCHEME	Scheme	R	0x1
29:28	BU	Business Unit	R	0x2
27:16	MODULE_ID	Module ID	R	0x6A4
15:11	REVRTL	RTL version	R	0x1
10:8	REVMAJ	Major version	R	0x1
7:6	CUSTOM	Custom version	R	0x0
5:0	REVMIN	Minor version	R	0x0

Table 18-750. Register Call Summary for Register MCANSS_ECC_WRAP_REVISION

MCAN

- ECC Aggregator: [0]
- MCAN Register Summary:

Table 18-751. MCANSS_ECC_CONTROL

Address Offset	0x0000 1C14
Physical Address	**Instance**
Description	ECC Control ECC Control Register.
Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								RESERVED	ERROR_ONCE	FORCE_N_ROW	FORCE_DED	FORCE_SEC	ENABLE_RMW	ECC_CHECK	ECC_ENABLE

Bits	Field Name	Description	Type	Reset
31:7	RESERVED	Reserved	R	0x0
6	ERROR_ONCE	Force Error only once	RW	0x0
5	FORCE_N_ROW	Force Error on any RAM read	RW	0x0
4	FORCE_DED	Force Double Bit Error	RW	0x0
3	FORCE_SEC	Force Single Bit Error	RW	0x0
2	ENABLE_RMW	Enable RMW	RW	0x1
1	ECC_CHECK	Enable ECC check	RW	0x1
0	ECC_ENABLE	Enable ECC	RW	0x1

Table 18-752. Register Call Summary for Register MCANSS_ECC_CONTROL

MCAN

- ECC Aggregator: [0]
- MCAN Register Summary:

Table 18-753. MCANSS_ECC_ERR_CTRL1

Address Offset	0x0000 1C18		
Physical Address		**Instance**	
Description	ECC Error Control1 Register ECC Error Control1 Register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ECC_BIT1																ECC_ROW															

Bits	Field Name	Description	Type	Reset
31:16	ECC_BIT1	Data bit that needs to be flipped when FORCE_SEC is set	RW	0x0
15:0	ECC_ROW	Row address where single or double-bit error needs to be applied. This is ignored if FORCE_N_ROW is set.	RW	0x0

Table 18-754. Register Call Summary for Register MCANSS_ECC_ERR_CTRL1

MCAN

- ECC Aggregator: [0]
- MCAN Register Summary:

Table 18-755. MCANSS_ECC_ERR_CTRL2

Address Offset	0x0000 1C1C		
Physical Address		**Instance**	
Description	ECC Error Control2 Register ECC Error Control2 Register.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																ECC_BIT2															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Reserved	R	0x0
15:0	ECC_BIT2	Data bit that needs to be flipped if double bit error needs to be forced	RW	0x0

Table 18-756. Register Call Summary for Register MCANSS_ECC_ERR_CTRL2

MCAN

- ECC Aggregator: [0]
- MCAN Register Summary:

Table 18-757. MCANSS_ECC_ERR_STAT1

Address Offset	0x0000 1C20
Physical Address	**Instance**
Description	ECC Error Status1 Register ECC Error Status1 Register.
Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ECC_ROW																RESERVED						CLR_ECC_DED	CLR_ECC_SEC	RESERVED						ECC_DED	ECC_SEC

Bits	Field Name	Description	Type	Reset
31:16	ECC_ROW	Row address where the single or double-bit error has occurred	R	0x0
15:10	RESERVED	Reserved	R	0x0
9	CLR_ECC_DED	Clear Double Bit Error Status	RW1TC	0x0
8	CLR_ECC_SEC	Clear Single Bit Error Status	RW1TC	0x0
7:2	RESERVED	Reserved	R	0x0
1	ECC_DED	Level Double Bit Error Status	RW1TS	0x0
0	ECC_SEC	Level Single Bit Error Status	RW1TS	0x0

Table 18-758. Register Call Summary for Register MCANSS_ECC_ERR_STAT1

MCAN
- ECC Aggregator: [0][1][2][3][4][5]
- MCAN Register Summary:

Table 18-759. MCANSS_ECC_ERR_STAT2

Address Offset	0x0000 1C24
Physical Address	**Instance**
Description	ECC Error Status2 Register ECC Error Status2 Register.
Type	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ECC_BIT2																ECC_BIT1															

Bits	Field Name	Description	Type	Reset
31:16	ECC_BIT2	Data bit that corresponds to the double-bit error	R	0x0
15:0	ECC_BIT1	Data bit that corresponds to the single-bit error	R	0x0

Table 18-760. Register Call Summary for Register MCANSS_ECC_ERR_STAT2

MCAN
- ECC Aggregator: [0][1][2]
- MCAN Register Summary:

Table 18-761. MCANSS_ECC_SEC_EOI_REG

Address Offset	0x0000 1C3C
Physical Address	**Instance**
Description	EOI Register EOI Register.
Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															EOI_WR

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved	R	0x0
0	EOI_WR	EOI Register	RW	0x0

Table 18-762. Register Call Summary for Register MCANSS_ECC_SEC_EOI_REG

MCAN
- ECC Aggregator: [0][1]
- MCAN Register Summary:

Table 18-763. MCANSS_ECC_SEC_STATUS_REG0

Address Offset	0x0000 1C40
Physical Address	**Instance**
Description	Interrupt Status Register 0 Interrupt Status Register 0.
Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															MSGMEM_PEND

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved	R	0x0
0	MSGMEM_PEND	Interrupt Pending Status for MSGMEM_PEND	RW1TS	0x0

Table 18-764. Register Call Summary for Register MCANSS_ECC_SEC_STATUS_REG0

MCAN
- ECC Aggregator: [0]
- MCAN Register Summary:

Table 18-765. MCANSS_ECC_SEC_ENABLE_SET_REG0

Address Offset	0x0000 1C80
Physical Address	**Instance**
Description	Interrupt Enable Set Register 0 Interrupt Enable Set Register 0.
Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															MSGMEM_ENABLE_SET

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved	R	0x0
0	MSGMEM_ENABLE_SET	Interrupt Enable Set Register for MSGMEM_PEND	RW1TS	0x0

Table 18-766. Register Call Summary for Register MCANSS_ECC_SEC_ENABLE_SET_REG0

MCAN

- ECC Aggregator: [0][1]
- MCAN Register Summary:

Table 18-767. MCANSS_ECC_SEC_ENABLE_CLR_REG0

Address Offset	0x0000 1CC0
Physical Address	**Instance**
Description	Interrupt Enable Clear Register 0 Interrupt Enable Clear Register 0.
Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															MSGMEM_ENABLE_CLR

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved	R	0x0
0	MSGMEM_ENABLE_CLR	Interrupt Enable Clear Register for MSGMEM_PEND	RW1TC	0x0

Table 18-768. Register Call Summary for Register MCANSS_ECC_SEC_ENABLE_CLR_REG0

MCAN
- ECC Aggregator: [0]
- MCAN Register Summary:

Table 18-769. MCANSS_ECC_DED_EOI_REG

Address Offset	0x0000 1D3C
Physical Address	**Instance**
Description	EOI Register EOI Register.
Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															EOI_WR

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved	R	0x0
0	EOI_WR	EOI Register	RW	0x0

Table 18-770. Register Call Summary for Register MCANSS_ECC_DED_EOI_REG

MCAN
- ECC Aggregator: [0][1]
- MCAN Register Summary:

Table 18-771. MCANSS_ECC_DED_STATUS_REG0

Address Offset	0x0000 1D40
Physical Address	**Instance**
Description	Interrupt Status Register 0 Interrupt Status Register 0.
Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															MSGMEM_PEND

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved	R	0x0
0	MSGMEM_PEND	Interrupt Pending Status for MSGMEM_PEND	RW1TS	0x0

Table 18-772. Register Call Summary for Register MCANSS_ECC_DED_STATUS_REG0

MCAN
- ECC Aggregator: [0]
- MCAN Register Summary:

Table 18-773. MCANSS_ECC_DED_ENABLE_SET_REG0

Address Offset	0x0000 1D80
Physical Address	**Instance**
Description	Interrupt Enable Set Register 0 Interrupt Enable Set Register 0.
Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															MSGMEM_ENABLE_SET

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved	R	0x0
0	MSGMEM_ENABLE_SET	Interrupt Enable Set Register for MSGMEM_PEND	RW1TS	0x0

Table 18-774. Register Call Summary for Register MCANSS_ECC_DED_ENABLE_SET_REG0

MCAN

- ECC Aggregator: [0][1]
- MCAN Register Summary:

Table 18-775. MCANSS_ECC_DED_ENABLE_CLR_REG0

Address Offset	0x0000 1DC0
Physical Address	**Instance**
Description	Interrupt Enable Clear Register 0 Interrupt Enable Clear Register 0.
Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															MSGMEM_ENABLE_CLR

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved	R	0x0
0	MSGMEM_ENABLE_CLR	Interrupt Enable Clear Register for MSGMEM_PEND	RW1TC	0x0

Table 18-776. Register Call Summary for Register MCANSS_ECC_DED_ENABLE_CLR_REG0

MCAN

- ECC Aggregator: [0]
- MCAN Register Summary:

18.8 Gigabit Ethernet Switch (GMAC_SW)

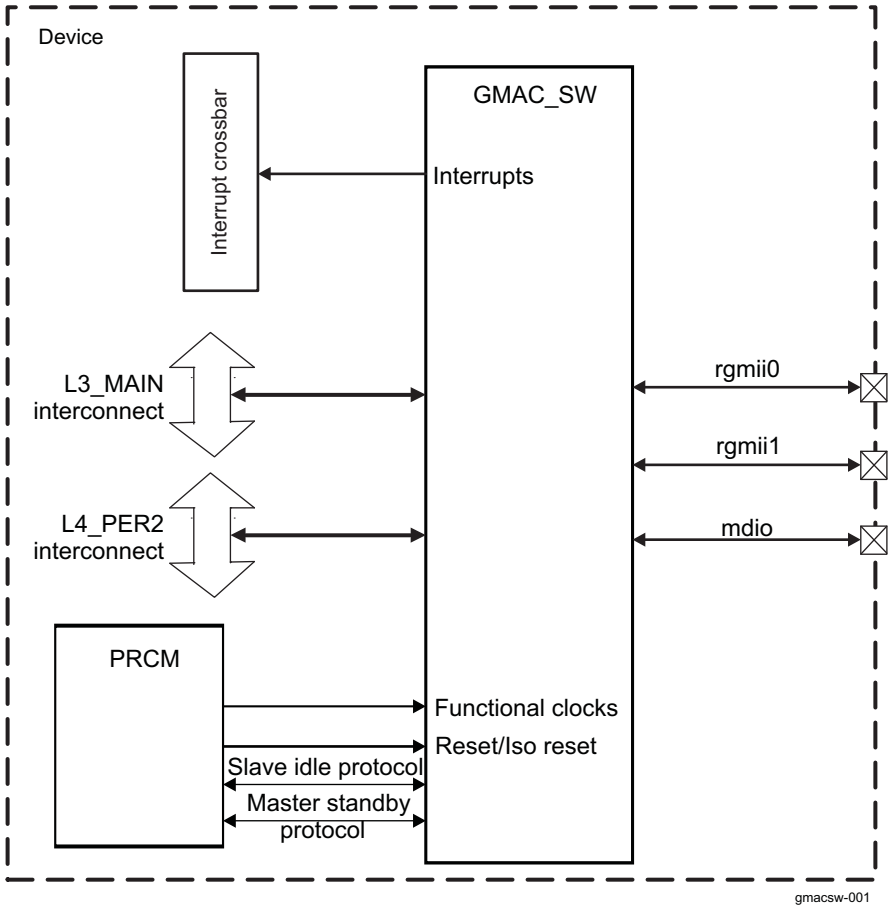
This chapter describes the three-port gigabit switch ethernet subsystem in the device.

18.8.1 GMAC_SW Overview

The three-port gigabit ethernet switch subsystem (GMAC_SW) provides ethernet packet communication and can be configured as an ethernet switch. It provides the reduced gigabit media independent interface (RGMII), and the management data input output (MDIO) for physical layer device (PHY) management.

Figure 18-154 shows the GMAC_SW subsystem overview.

Figure 18-154. GMAC_SW Overview



18.8.1.1 Features

The GMAC_SW subsystem provides the following features:

- Two Ethernet ports (port 1 and port 2) with RGMII interfaces plus internal Communications Port Programming Interface (CPPI 3.1) on port 0
- Synchronous 10/100/1000 Mbit operation
- Wire rate switching (802.1d)
- Non-blocking switch fabric
- Flexible logical FIFO-based packet buffer structure
- Four priority level Quality Of Service (QOS) support (802.1p)
- CPPI 3.1 compliant DMA controllers
- Support for Audio/Video Bridging (P802.1Qav/D6.0)

- Support for IEEE 1588 Clock Synchronization (2008 Annex D and Annex F)
  - Timing FIFO and time stamping logic embedded in the subsystem
- Device Level Ring (DLR) Support
- Energy Efficient Ethernet (EEE) support (802.3az)
- Flow Control Support (802.3x)
- Address Lookup Engine (ALE)
  - 1024 total address entries plus VLANs
  - Wire rate lookup
  - Host controlled time-based aging
  - Multiple spanning tree support (spanning tree per VLAN)
  - L2 address lock and L2 filtering support
  - MAC authentication (802.1x)
  - Receive-based or destination-based multicast and broadcast rate limits
  - MAC address blocking
  - Source port locking
  - OUI (Vendor ID) host accept/deny feature
  - Remapping of priority level of VLAN or ports
- VLAN support
  - 802.1Q compliant
    - Auto add port VLAN for untagged frames on ingress
    - Auto VLAN removal on egress and auto pad to minimum frame size
- Ethernet Statistics:
  - EtherStats and 802.3Stats Remote network Monitoring (RMON) statistics gathering (shared)
  - Programmable statistics interrupt mask when a statistic is above one half its 32-bit value
- Flow Control Support (802.3x)
- Digital loopback and FIFO loopback modes supported
- Maximum frame size 2016 bytes (2020 with VLAN)
- 8k (2048 × 32) internal CPPI buffer descriptor memory
- Management Data Input/Output (MDIO) module for PHY Management
- Programmable interrupt control with selected interrupt pacing
- Emulation support
- Programmable Transmit Inter Packet Gap (IPG)
- Reset isolation (switch function remains active even in case of all device resets except for POR pin reset and ICEPICK cold reset)
- Full duplex mode supported in 10/100/1000 Mbps. Half-duplex mode supported only in 10/100 Mbps.
- IEEE 802.3 gigabit Ethernet conformant

Terminology:

AVB— Audio Video Bridging

AVBTP— Audio Video Bridging Trasport Protocol

BMCA— Best Master Clock Algorithm

CFI— Canonical Format Indicator

CPPI— Communications Port Programming Interface

DLR— Device Level Ring

DSCP— Differentiated Services Code Point

EEE— Energy Efficient Ethernet

EMAC— Ethernet Media Access Control

EOP— End of Packet

EOQ— End of Queue

IPG— Inter-Packet Gap

LPI— Low Power Indicator

MDIO— Management Data Input/Output

MOF— Middle of Frame

PTP— Precision Time Protocol

RMON— Remote Monitoring

RTCP— RTP Control Protocol

RTP— Real-time Transport Protocol

SCR— Switched Central Resource

SRP— Stream Reservation Protocol

TOS— Type of Service

VLAN— Virtual Local Area Network

18.8.2 GMAC_SW Environment

18.8.2.1 RGMII Interface

Figure 18-155 shows a device with integrated CPSW and MDIO interfaced via a RGMII connection in a typical system. The individual CPSW and MDIO signals for the RGMII interface are summarized in Table 18-777.

Figure 18-155. RGMII Interface Typical Application

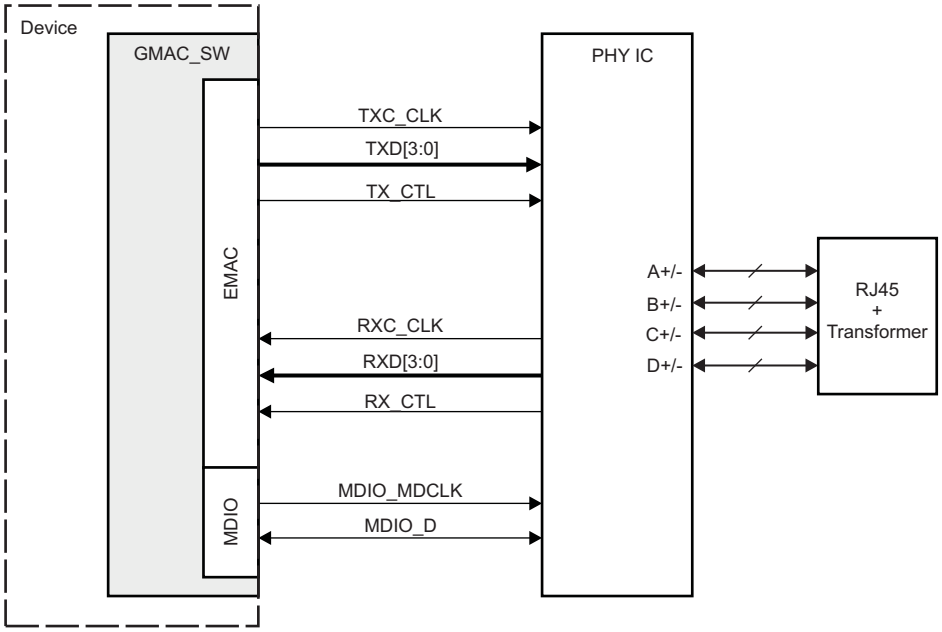


Table 18-777. RGMII I/O Description

Signal	Device Pin(s)	I/O[1]	Description
TXD[3:0]	rgmii0_txd[3:0] rgmii1_txd[3:0]	O	The transmit data pins are a collection of 4 bits of data. TXD0 is the least-significant bit (LSB). The signals are valid only when TX_CTL is asserted.
TX_CTL	rgmii0_rxctl rgmii1_rxctl	O	Transmit Control/enable. The transmit enable signal indicates that the TXD pins are generating data for use by the PHY.
TXC_CLK	rgmii0_txc rgmii1_txc	O	The transmit reference clock. The clock is 2.5 MHz at 10 Mbps operation, 25 MHz at 100 Mbps operation, and 125 MHz at 1000 Mbps of operation.
RXD[3:0]	rgmii0_rxd[3:0] rgmii1_rxd[3:0]	I	The receive data pins are a collection of 4 bits of data. RXD0 is the least-significant bit (LSB). The signals are valid only when RX_CTL is asserted
RX_CTL	rgmii0_rxctl rgmii1_rxctl	I	The receive data valid/control signal indicates that the RXD pins are nibble data for use by the GMAC_SW.
RXC_CLK	rgmii0_rxc rgmii1_rxc	I	The receive clock is a continuous clock that provides the timing reference for receive operations. The clock is generated by the PHY and is 2.5 MHz at 10 Mbps operation, 25 MHz at 100 Mbps operation, 125 MHz at 1000 Mbps of operation.
MDIO_MDCLK	mdio_mclk	O	Management data clock (MDIO_MCLK). The MDIO data clock is sourced by the MDIO module on the system. It is used to synchronize MDIO data access operations done on the MDIO pin.
MDIO_D	mdio_d	I/O	MDIO data pin drives PHY management data into and out of the PHY by way of an access frame consisting of start of frame, read/write indication, PHY address, register address, and data bit cycles. The MDIO_D pin acts as an output for all but the data bit cycles at which time it is an input for read operations.

(1) I = Input; O = Output

NOTE: The path from a module pin to device pad(s) is defined at the device I/O logic level. The control module registers assign the specific function to the device pads. For more information on control module settings, see Section 13.4.6.1, *Pad Configuration Registers* in *Chapter 13, Control Module*.

18.8.3 GMAC_SW Integration

Figure 18-156 shows the integration of the GMAC_SW module in the device.

Figure 18-156. GMAC_SW Integration

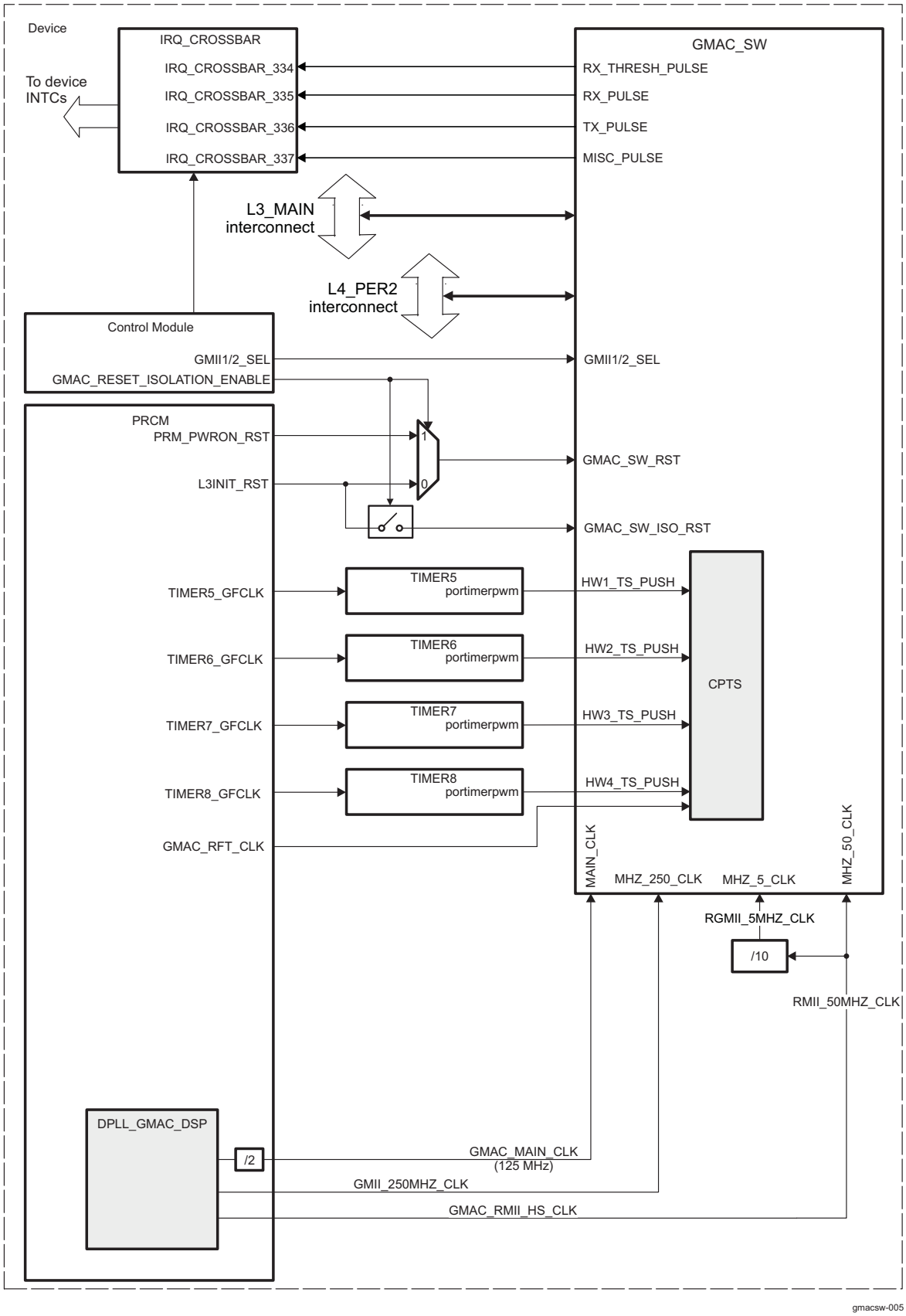


Table 18-778 through Table 18-780 summarize the integration of the GMAC_SW module in the device.

Table 18-778. GMAC_SW Integration Attributes

Module Instance	Attributes		
	Power Domain	Wake-Up Capability	Interconnect
GMAC_SW	PD_COREAON	Yes	L3_MAIN L4_PER2

Table 18-779. GMAC_SW Clocks and Resets

Clocks				
Module Instance	**Destination Signal Name**	**Source Signal Name**	**Source**	**Description**
GMAC_SW	MAIN_CLK	GMAC_MAIN_CLK	PRCM	Interface clock for the GMAC_SW module (125 MHz)
	MHZ_5_CLK	RGMII_5MHZ_CLK	PRCM	5-MHz RGMII clock
	MHZ_50_CLK	RMII_50MHZ_CLK	PRCM	50-MHz RGMII clock
	MHZ_250_CLK	GMII_250MHZ_CLK	PRCM	250-MHz RGMII clock
	CPTS_RFT_CLK	GMAC_RFT_CLK	PRCM	IEEE 1588 clock
Resets				
Module Instance	**Destination Signal Name**	**Source Signal Name**	**Source**	**Description**
GMAC_SW	GMAC_SW_RST	L3INIT_RST/PRM_PWR ON_RST	PRCM	GMAC_SW main reset
	GMAC_SW_ISO_RST	tied off/L3INIT_RST	PRCM	GMAC_SW isolate reset (must be enabled in Control module)

Table 18-780. GMAC_SW Hardware Requests

Interrupt Requests				
Module Instance	**Source Signal Name**	**IRQ_CROSSBAR Input**	**Default Mapping**	**Description**
GMAC_SW	RX_THRESH_PULSE	IRQ_CROSSBAR_334	-	Receive threshold interrupt
	RX_PULSE	IRQ_CROSSBAR_335	-	Receive packet completion interrupt
	TX_PULSE	IRQ_CROSSBAR_336	-	Transmit packet completion interrupt
	MISC_PULSE	IRQ_CROSSBAR_337	-	Miscellaneous interrupt (SPF2_PEND, SPF1_PEND, EVNT_PEND,STAT_PEND, HOST_PEND, MDIO_LINKINT, MDIO_USERINT)

NOTE: GMAC_SW has no default IRQ mappings through the IRQ_CROSSBAR. For GMAC_SW, the IRQ_CROSSBAR module must be configured prior to unmask interrupts in the interrupt controller(s).
For more information about the IRQ_CROSSBAR module, see Section 13.4.6.4, *IRQ_CROSSBAR Module Functional Description*, in Chapter 13, *Control Module*.
For more information about the device interrupt controllers, see Chapter 12, *Interrupt Controllers*.

NOTE: For the description of the interrupt source, see Section 18.8.4.5, *Interrupt Functionality*.

18.8.4 GMAC_SW Functional Description

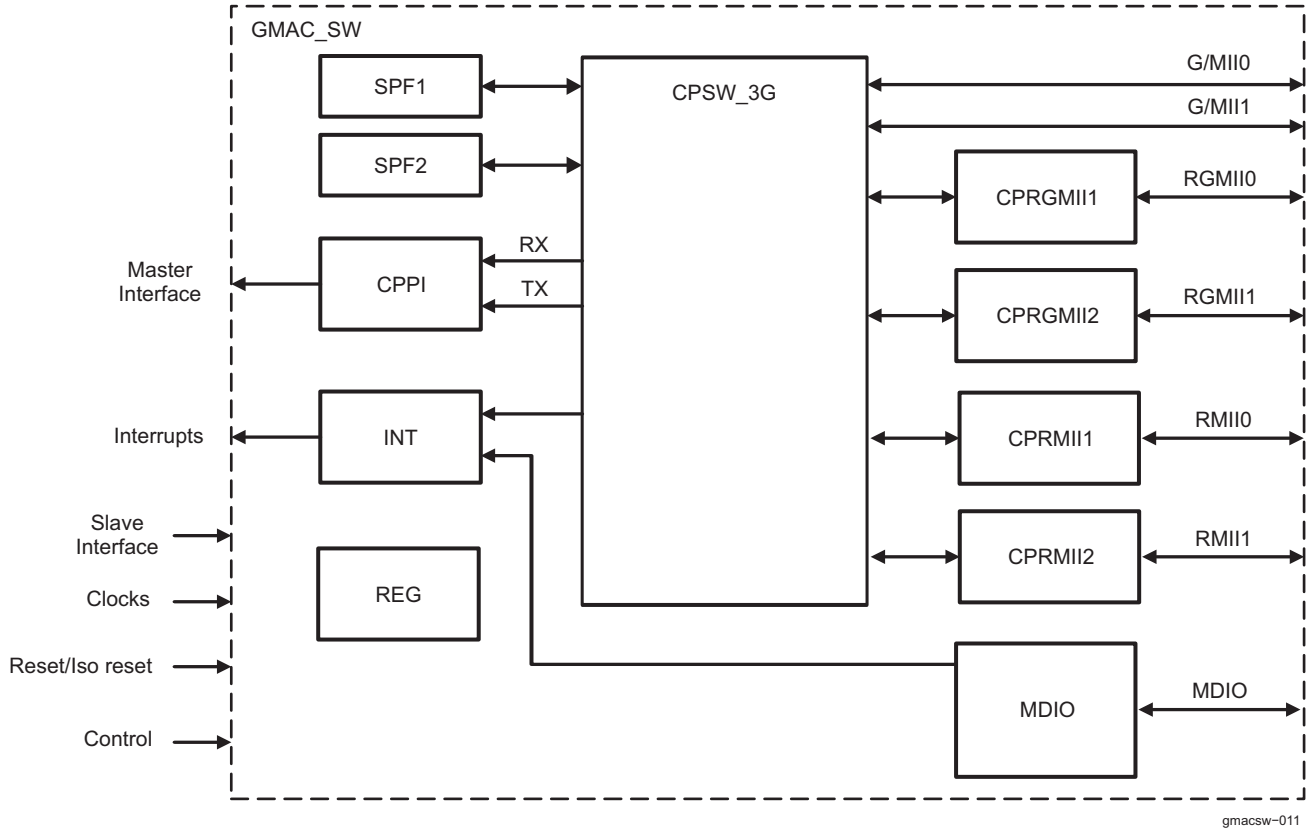
The 3-port switch (GMAC_SW) Ethernet subsystem modules are compliant to the IEEE Std 802.3 Specification.GMAC_SW is shown in Figure 18-157.

18.8.4.1 Functional Block Diagram

The GMAC_SW subsystem consists of:

- CPSW_3G which contains two G/MII interfaces (not pinned out in this device, used to connect to CPRGMII0/1)
- Two RGMII interface modules
- Two RMII interface modules (RMII0/1 are not pinned out in this device)
- One MDIO interface module
- One Interrupt Controller module
- One CPPI interface
- Local CPPI memory of size 8 KiB
- Two Static Packet Filters

Figure 18-157. GMAC_SW Top Level Block Diagram



18.8.4.2 GMAC_SW Ports

Ethernet Subsystem has three Ports. Port 0 is the Host port (internal to the Subsystem). Ports 1 and 2 are the external ports connected to RGMII.

Naming conventions followed in this chapter:

- Port0 is referred to the Host Port
- Port1 is referred to the RGMII0
- Port2 is referred to the RGMII1

The MII mode selection register bitfields (GMII1_SEL and GMII2_SEL) in the control module must be set to 0x2 (RGMII). See Section 13.5, *Control Module Register Manual* for details.

18.8.4.3 Clocking

18.8.4.3.1 Subsystem Clocking

GMAC_SW clocking summary is shown in Section 18.8.3, *GMAC_SW Integration*.

18.8.4.3.2 Interface Clocking

Data is transmitted and received with respect to the reference clocks of the interface pins.

18.8.4.3.2.1 RGMII Interface Clocking

RGMII_RXC, RGMII_TXC frequencies are:

- 2.5 MHz at 10 Mbps
- 25 MHz at 100 Mbps
- 125 MHz at 1000 Mbps

18.8.4.3.2.2 MDIO Clocking

The MDIO clock is based on a divide-down of the interface (MAIN_CLK) clock, running at 125 MHz.The application software or driver must control the divide-down value.

See the MDIO_CONTROL register for configuring the Clock Divider (CLKDIV) value.

18.8.4.4 Software IDLE

The submodule software idle register bits enable CPSW_3G operation to be completely or partially suspended by software control. There are three CPSW_3G submodules that contain software idle register bits (CPGMAC_SL1, CPGMAC_SL2, and CPDMA). Each of the three submodules may be individually commanded to enter the idle state. The idle state is entered at packet boundaries, and no further packet operations will occur on an idled submodule until the idle command is removed. The CPSW_3G module enters the idle state when all three submodules are commanded to enter and have entered the idle state. Idle status is determined by reading or polling the three submodule idle bits. The CPSW_3G is in the idle state when all three submodules are in the idle state. The CPSW_SOFT_IDLE[0] SOFT_IDLE bit may be set if desired after the submodules are in the idle state. The CPSW SOFT_IDLE bit causes packets to not be transferred from one FIFO to another FIFO internal to the switch.

18.8.4.5 Interrupt Functionality

GMAC_SW Ethernet Subsystem has four Interrupt outputs:

- RX_PULSE - Receive Interrupt
- TX_PULSE - Transmit Interrupt
- RX_THRESH_PULSE - Receive Threshold Interrupt
- MISC_PULSE - Miscellaneous Interrupt.

18.8.4.5.1 Receive Packet Completion Pulse Interrupt (RX_PULSE)

The RX_PULSE interrupt is a pulse interrupt selected from the GMAC_SW RX_PEND[7:0] interrupts. The receive DMA controller has eight channels with each channel having a corresponding (RX_PEND[7:0]).

The following steps will enable the receive packet completion interrupt:

1. Enable the required channel interrupts of the DMA engine by setting 1 to the appropriate bit in the CPDMA_RX_INTMASK_SET register.
2. The receive completion interrupt(s) to be routed to RX_PULSE is selected by setting one or more bits in the receive interrupt enable register (WR_C0_RX_EN) . The masked interrupt status can be read in

the address location of RX_STAT bit in the WR_C0_RX_STAT register.

When the GMAC_SW completes a packet reception, the subsystem issues an interrupt to the host processor by writing the packet's last buffer descriptor address to the appropriate channel queue's receive completion pointer located in the state RAM block. The interrupt is generated by the write when enabled by the interrupt mask, regardless of the value written.

Upon interrupt reception, the software processes one or more packets from the buffer chain and then acknowledges one or more interrupt(s) by writing the address of the last buffer descriptor processed to the queue's associated receive completion pointer (RX *n*_CP) in the receive DMA state RAM.

Upon reception of an interrupt, software should perform the following:

1. Read the WR_C0_RX_STAT[7:0] RX_STAT bit address location to determine which channel(s) caused the interrupt.
2. Process received packets for the interrupting channel(s).
3. Write the GMAC_SW completion pointer(s) (RX *n*_CP). The data written by the host (buffer descriptor address of the last processed buffer) is compared to the data in the register written by the subsystem (address of last buffer descriptor used by the subsystem). If the two values are not equal (which means that the GMAC_SW has received more packets than the software has processed), the receive packet completion interrupt signal remains asserted. If the two values are equal (which means that the host has processed all packets that the system has received), the pending interrupt is de-asserted. The value that the GMAC_SW is expecting is found by reading the receive channel *n* completion pointer register (RX *n*_CP).
4. Write the value 1h to the CPDMA_EOI_VECTOR register.

To disable the interrupt:

1. The eight channel interrupts may be individually disabled by writing to 1 the appropriate bit in the CPDMA_RX_INTMASK_CLEAR register.
2. The receive completion pulse interrupt could be disabled by clearing to 0 all the bits in the WR_C0_RX_EN register.

The software could still poll for the CPDMA_RX_INTSTAT_RAW and CPDMA_RX_INTSTAT_MASKED registers if the corresponding interrupts are enabled.

18.8.4.5.2 Transmit Packet Completion Pulse Interrupt (TX_PULSE)

The TX_PULSE interrupt is a pulse interrupt selected from the GMAC_SW TX_PEND[7:0] interrupts. The transmit DMA controller has eight channels with each channel having a corresponding (TX_PEND[7:0]).

To enable the transmit packet completion interrupt:

1. Enable the required channel interrupts of the DMA engine by setting 1 to the appropriate bit in the CPDMA_TX_INTMASK_SET register.
2. The transmit completion interrupt(s) to be routed to TX_PULSE is selected by setting one or more bits in the transmit interrupt enable register WR_C0_TX_EN .The masked interrupt status can be read in the address location of TX_STAT bit in the WR_C0_TX_STAT register.

When the GMAC_SW completes the transmission of a packet, the GMAC_SW subsystem issues an interrupt to the host processor by writing the packet's last buffer descriptor address to the appropriate channel queue's transmit completion pointer located in the state RAM block. The interrupt is generated by the write when enabled by the interrupt mask, regardless of the value written.

Upon reception of an interrupt, software should perform the following:

1. Read the TX_STAT bit address location to determine which channel(s) caused the interrupt
2. Process received packets for the interrupting channel(s).
3. Write the GMAC_SW completion pointer(s) (TX *n*_CP). The data written by the host (buffer descriptor address of the last processed buffer) is compared to the data in the register written by the GMAC_SW (address of last buffer descriptor used by the GMAC_SW). If the two values are not equal (which means that the GMAC_SW has transmitted more packets than the software has processed), the transmit packet completion interrupt signal remains asserted. If the two values are equal (which means that the host has processed all packets that the subsystem has transferred), the pending interrupt is

cleared. The value that the GMAC_SW is expecting is found by reading the transmit channel *n*completion pointer register (TX *n*_CP).

4. Write the 2h to the CPDMA_EOI_VECTOR register.

To disable the interrupt:

1. The eight channel interrupts may be individually disabled by writing to 1 the appropriate bit in the CPDMA_TX_INTMASK_CLEAR register.
2. The receive completion pulse interrupt could be disabled by clearing to 0 all the bits in the WR_C0_TX_EN register.

The software could still poll for the CPDMA_TX_INTSTAT_RAW and CPDMA_TX_INTSTAT_MASKED registers, if the corresponding interrupts are enabled.

18.8.4.5.3 Receive Threshold Pulse Interrupt (RX_THRESH_PULSE)

The RX_THRESH_PULSE interrupt is an immediate (non-paced) pulse interrupt selected from the CPSW_3G RX_THRESH_PEND[7:0] interrupts. The receive DMA controller has eight channels with each channel having a corresponding threshold pulse interrupt (RX_THRESH_PEND[7:0]).

To enable the receive threshold pulse Interrupt:

1. Enable the required channel interrupts of the DMA engine by setting 1 to the appropriate bit in the CPDMA_RX_INTMASK_SET register.
2. The receive threshold interrupt(s) to be routed to RX_THRESH_PULSE is selected by setting one or more bits in the receive threshold interrupt enable register WR_C0_RX_THRESH_EN. The masked interrupt status can be read in the address location of RX_THRESH_STAT bit in the WR_C0_RX_THRESH_STAT register.

The RX_THRESH_PULSE is asserted when enabled when the channel's associated free buffer count RX *n*_FREEBUFFER is less than or equal to the corresponding RX *n*_PENDTHRESH register.

Upon reception of an interrupt, software should perform the following:

1. Read the RX_THRESH_STAT bit address location to determine which channel(s) caused the interrupt.
2. Process the received packets in order to add more buffers to any channel that is below the threshold value.
3. Write the CPSW_3G completion pointer(s).
4. Write the value 0h to the CPDMA_EOI_VECTOR register.

The threshold pulse interrupt is an immediate interrupt intended to indicate that software should immediately process packets to preclude an overrun condition from occurring for the particular channels.

To disable the interrupt:

1. The eight channel receive threshold interrupts may be individually disabled by writing to 1 the appropriate bit in the CPDMA_RX_INTMASK_CLEAR register.
2. The receive threshold pulse interrupt could be disabled by clearing to 0 the corresponding bits in the WR_C0_RX_THRESH_EN register.

The software could still poll for the CPDMA_RX_INTSTAT_RAW and CPDMA_RX_INTSTAT_MASKED registers, if the corresponding interrupts are enabled.

18.8.4.5.4 Miscellaneous Pulse Interrupt (MISC_PULSE)

The MISC_PULSE interrupt is an immediate pulse interrupt selected from the miscellaneous interrupts (SPF2_PEND, SPF1_PEND, EVNT_PEND,STAT_PEND, HOST_PEND, MDIO_LINKINT, MDIO_USERINT).

The miscellaneous interrupt(s) is selected by setting one or more bits in the miscellaneous interrupt enable register (WR_C0_MISC_EN).

Upon reception of an interrupt, software should perform the following:

- Read the MISC_STAT bit address location to determine the source of the interrupt.
- Process the interrupt.

- Write the value 3h to the CPDMA_EOI_VECTOR register.

NOTE: The WR_C0_MISC_STAT register's MDIO_LINKINT and MDIO_USERINT bitfields represent the Port0/Phy0 status. MDIO_LINKINT[1] and MDIO_USERINT[1] are not provided in WR_C0_MISC_STAT register. As such, MDIO Link and User interrupts can only be generated for Port0. For Port1, software must poll the status, visible in the MDIO_LINKINTMASKED or MDIO_USERINTMASKED registers.

18.8.4.5.4.1 EVNT_PEND(CPTS_PEND) Interrupt

See Section 18.8.4.10, *Common Platform Time Sync (CPTS)* for more details on this interrupt.

18.8.4.5.4.2 Statistics Interrupt

The statistics level interrupt (STAT_PEND) will be asserted, if enabled when any statistics value is greater than or equal to 8000 0000h. The statistics interrupt is cleared by writing to decrement all statistics values greater than 8000 0000h (such that their new values are less than 8000 0000h). The raw and masked statistics interrupt status may be read by reading the CPDMA_TX_INTSTAT_RAW and CPDMA_TX_INTSTAT_MASKED registers, respectively.

The Statistics interrupt is enabled by setting to 1 the STAT_INT_MASK bit in the CPDMA_DMA_INTMASK_SET register

18.8.4.5.4.3 Host Error interrupt

The host error interrupt (HOST_PEND) will be asserted, if enabled when a host error is detected during transmit or receive CPDMA transactions. The host error interrupt is intended for software debug, and is cleared by a warm reset or a system reset. The raw and masked host interrupt status can be read by reading the CPDMA_DMA_INTSTAT_RAW and CPDMA_DMA_INTSTAT_MASKED registers, respectively.

The transmit host error conditions are:

- SOP error
- OWNERSHIP bit not set in SOP buffer
- next buffer descriptor pointer without EOP cleared to 0
- buffer pointer cleared to 0
- buffer length cleared to 0
- packet length error

The receive host error conditions are:

- OWNERSHIP bit not set in input buffer
- Zero buffer pointer
- Zero buffer Length on non-SOP descriptor
- SOP buffer length not greater than offset

The HOST_PEND is enabled by setting to 1 the HOST_ERR_INTMASK in the CPDMA_DMA_INTMASK_SET register. The host error interrupt is disabled by setting to 1 the appropriate bit in the CPDMA_DMA_INTMASK_CLEAR register.

18.8.4.5.4.4 MDIO Interrupts

MDIO_LINKINT is set if there is a change in the link state of the PHY corresponding to the address in the PHYADR_MON field of the MDIO_USERPHYSEL*n* register and the corresponding LINKINT_ENABLE bit is set. The MDIO_LINKINT event is also captured in the MDIO_LINKINTMASKED register. When the GO bit in the MDIO_USERACCESS*n* register transitions from 1 to 0, indicating the completion of a user access, and the corresponding USERINTMASKSET bit in the MDIO_USERINTMASKSET register is set, the MDIO_USERINT signal is asserted. The MDIO_USERINT event is also captured in the MDIO_USERINTMASKED register.

18.8.4.5.5 Interrupt Pacing

RX_PULSE and TX_PULSE interrupts can be paced. The RX_THRESH_PULSE and MISC_PULSE interrupts are not paced. The Interrupt pacing feature limits the number of interrupts that occur during a given period of time. For heavily loaded systems in which interrupts can occur at a very high rate (for example, 148,800 packets per second for Ethernet), the performance benefit is significant due to minimizing the overhead associated with servicing each interrupt. Interrupt pacing increases the processor cache hit ratio by minimizing the number of times that large interrupt service routines are moved to and from the processor instruction cache.

Each RX_PULSE and TX_PULSE interrupt contains an interrupt pacing sub-block (six total). Each sub-block is disabled by default allowing the selected interrupt inputs to pass through unaffected. The interrupt pacing module counts the number of interrupts that occur over a 1 ms interval of time. At the end of each 1 ms interval, the current number of interrupts is compared with a target number of interrupts (specified by the associated maximum number of interrupts register). Based on the results of the comparison, the length of time during which interrupts are blocked is dynamically adjusted. The 1 ms interval is derived from a 4 µs pulse that is created from a prescale counter whose value is set in the INT_PRESCALE field in the WR_INT_CONTROL register. The INT_PRESCALE value should be written with the number of ICLK periods in 4 µs. The pacing timer determines the interval during which interrupts are blocked and decrements every 4 µs. It is reloaded each time a zero count is reached. The value loaded into the pacing timer is calculated by hardware every 1 ms according to the following algorithm:

```
if (intr_count > 2*intr_max)
    pace_timer = 255;
  else if (intr_count > 1.5*intr_max)
    pace_timer = last_pace_timer*2 + 1;
  else if (intr_count > 1.0*intr_max)
    pace_timer = last_pace_timer + 1;
  else if (intr_count > 0.5*intr_max)
    pace_timer = last_pace_timer - 1;
  else if (intr_count != 0)
    pace_timer = last_pace_timer/2;
else
pace_timer = 0;
```

If the rate of interrupt inputs is much less than the target interrupt rate specified in the associated maximum interrupts register, then the interrupt is not blocked. If the interrupt rate is greater than the target rate, the interrupt will be "paced" at the rate specified in the interrupt maximum register. The WR_C0_RX_IMAX/WR_C0_TX_IMAX register should be written with a value between 2 and 63 inclusive, indicating the target number of interrupts per millisecond.

18.8.4.6 Reset Isolation

Reset isolation for the GMAC_SW allows the switch function to remain active in during all device resets except for POR pin reset and ICEPICK COLD reset. Packet traffic to/from the GMAC_SW host will be flushed/dropped, but the ethernet switch will remain operational for all traffic between external devices on the switch even though the device is undergoing a device reset. Pin mux configuration for ethernet related I/O and reference clocks needed by the GMAC_SW to be active is controlled by a protected control module bit.

18.8.4.6.1 Reset Isolation Functional Description

The device has two modes of operation concerning the reset of the GMAC_SW Ethernet switch. The mode is controlled by the GMAC_RESET_ISOLATION_ENABLE bit in the Control Module. This bit defaults to 0. Any modification of this bit first requires writing an unlock pattern to lock register in device control module. After modification, the bit should again be locked by writing appropriate value to the lock register. Writes to the GMAC_RESET_ISOLATION_ENABLE bit and to the lock register must all be supervisor mode writes.

GMAC_RESET_ISOLATION_ENABLE = 0 (disabled)

1. This is the default state of the bit after control module reset.
2. Upon any device level resets, the entire GMAC_SW, DPLL_GMAC_DSP, L3/L4 interconnect, control module (including all pin mux control and the GMAC_RESET_ISOLATION_ENABLE bit itself) are

immediately reset.

GMAC_RESET_ISOLATION_ENABLE = 1 (enabled)

1. This mode is selected when the GMAC_RESET_ISOLATION_ENABLE bit is set to 1 by software.
2. Upon any device reset source other than porz pin or ICEPICK cold (that is, this includes software global cold, any watchdog reset, warm resetn pin, ICEPICK warm, software global warm or security violation), the following is true:
    1. The CPSW_3GSS_R is put into "isolate" mode and non-switch related portions of the subsystem are reset.
    2. The 50 MHz and 125 MHz reference clocks to the GMAC_SW Ethernet Subsystem remain active throughout the entire reset condition.
    3. The control for pin multiplexing for all of the signals maintain their current configuration throughout the entire reset condition.
    4. The reset-isolated logic inside GMAC_SW Ethernet Subsystem maintains the switch functionality
3. Upon any cold reset sources, the entire GMAC_SW Ethernet Subsystem, DPLL_GMAC_DSP, control module (including all pin mux control and the GMAC_RESET_ISOLATION_ENABLE bit itself) are reset.

For more details on the register configuration, see the Control Module CTRL_CORE_CONTROL_IO_2 register in Section 13.5*, Control Module Register Manual*.

18.8.4.7 Software Reset

The CPSW_3G software reset register (CPSW_SOFT_RESET), CPSW_3GSS software reset register (WR_SOFT_RESET) and the three submodule software reset registers enable the CPSW_3GSS to be reset by software.

There are three CPSW_3G submodules that contain software reset registers (CPGMAC_SL1, CPGMAC_SL2 (SL_SOFT_RESET), and CPDMA (CPDMA_SOFT_RESET)). Each of the three submodules may be individually commanded to be reset by software.

For the CPDMA, the reset state is entered at packet boundaries, at which time the CPDMA reset occurs. The CPGMAC_SL soft reset is immediate. Submodule reset status is determined by reading or polling the submodule reset bit. If the submodule reset bit is read as a one, then the reset process has not yet completed. The submodule soft reset process could take up to 2ms each. The reset has completed if the submodule reset bit is read as a zero.

After all three submodules (in any order) have been reset and a read of each submodule reset bit indicates that the reset process is complete, the CPSW_3G software reset register bit may be written to complete the CPSW_3G module software reset operation. The CPSW_3G software reset bit controls the reset of the FIFO's, the statistics submodule, and the address lookup engine (ALE). The CPSW_3G software reset is immediate and will be indicated by reading a zero from the soft reset bit.

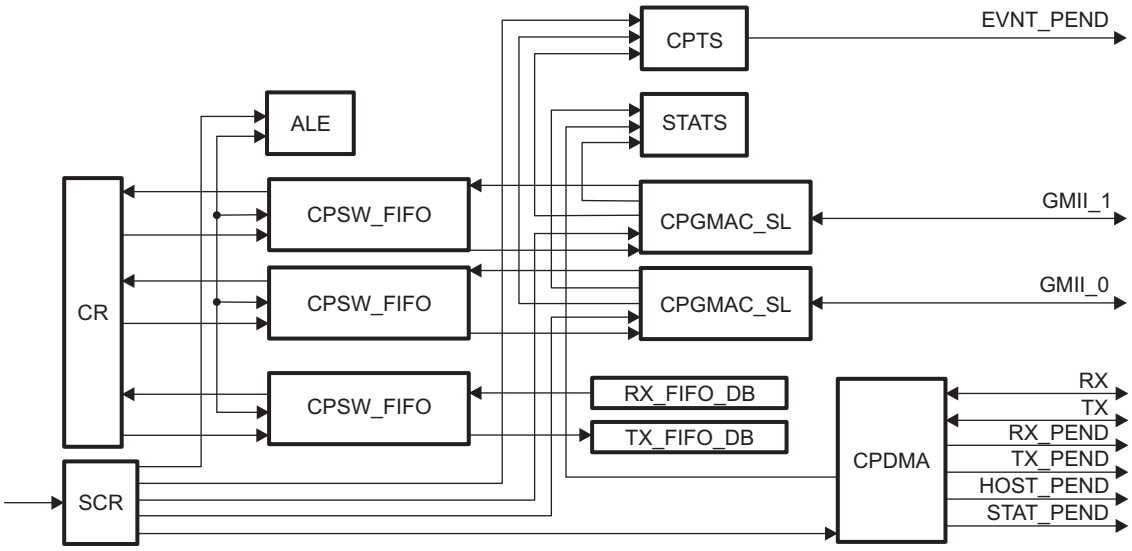
The CPSW_3GSS software reset bit controls the reset of the INT, REGS, and CPPI. The CPSW_3GSS software reset is immediate and will be indicated by reading a zero from the soft reset bit.

18.8.4.8 CPSW_3G

The CPSW_3G G/MII interfaces are compliant to the IEEE Std 802.3 Specification.

The CPSW_3G contains two CPGMAC_SL interfaces (ports 1 and 2), one CPPI interface Host Port (port 0), Common Platform Time Sync (CPTS), ALE Engine and CPDMA. A top-level block diagram of the CPSW_3G is shown in Figure 18-158.

Figure 18-158. CPSW_3G Block Diagram



18.8.4.8.1 CPDMA RX and TX Interfaces

The CPDMA submodule is a CPPI compliant packet DMA transfer controller. The CPPI interface is port 0.

18.8.4.8.1.1 Functional Operation

After reset, initialization, and configuration the host may initiate transmit operations. Transmit operations are initiated by host writes to the appropriate transmit channel head descriptor pointer contained in the STATERAM block. The transmit DMA controller then fetches the first packet in the packet chain from memory in accordance with CPPI protocol. The DMA controller writes the packet into the external transmit FIFO in 64-byte bursts (maximum).

Receive operations are initiated by host writes to the appropriate receive channel head descriptor pointer after host initialization and configuration. The receive DMA controller writes the receive packet data to external memory in accordance with CPPI protocol. For a detailed description of buffer descriptors, see Section 18.8.4.11, *CPPI Buffer Descriptors*.

18.8.4.8.1.2 Receive DMA Interface

The receive DMA is an eight channel CPPI compliant interface. Each channel has a single queue for frame reception.

18.8.4.8.1.2.1 Receive DMA Host Configuration

To configure the RX DMA for operation the software must perform the following:

1. Initialize the receive addresses.
2. Initialize the RX_HDP registers to 0.
3. Enable the desired receive interrupts in the CPDMA_RX_INTMASK_SET register.
4. Write the CPDMA_RX_BUFFER_OFFSET register value.
5. Setup the receive channel(s) buffer descriptors in host memory as required by CPPI
6. Enable the RX DMA controller by setting the RX_EN bit in the CPDMA_RX_CONTROL register.

18.8.4.8.1.2.2 Receive Channel Teardown

The host commands a receive channel teardown by writing the channel number to the CPDMA_RX_TEARDOWN register. When a teardown command is issued to an enabled receive channel the following will occur:

- Any current frame in reception will complete normally.
- The teardown_complete bit will be set in the next buffer descriptor in the chain (if there is one).
- The channel head descriptor pointer will be cleared to 0.
- A receive interrupt for the channel will be issued to the host.
- The software should acknowledge a teardown interrupt with a FFFF FFFCh Acknowledge value.

Channel teardown may be commanded on any channel at any time. The host is informed of the teardown completion by a set teardown complete buffer descriptor bit. The port does not clear any channel enables due to a teardown command. A teardown command to an inactive channel issues an interrupt that software should acknowledge with a FFFF FFFCh acknowledge value (note that there is no buffer descriptor in this case). Software may read the interrupt acknowledge location to determine if the interrupt was due to a commanded teardown. The read value will be FFFF FFFCh if the interrupt was due to a teardown command.

18.8.4.8.1.3 Transmit DMA Interface

The tranmsit DMA is an eight channel CPPI compliant interface. Priority between the eight queues may be either fixed or round robin as selected by the TX_PTYPE bit in the CPDMA_DMACONTROL register. If the priority type is fixed, then channel 7 has the highest priority and channel 0 has the lowest priority. Round robin priority proceeds from channel 0 to channel 7.

18.8.4.8.1.3.1 Transmit DMA Host Configuration

To configure the TX DMA for operation the software must do the following:

1. Initialize the TX_HDP registers to 0.
2. Enable the desired transmit interrupts in the CPDMA_TX_INTMASK_SET register.
3. Setup the transmit channel(s) buffer descriptors in host memory as defined in CPPI.
4. Configure and enable the transmit operation as desired in the CPDMA_TX_CONTROL register.
5. Write the appropriate TX_HDP registers with the appropriate values to start transmit operations.

18.8.4.8.1.3.2 Transmit Channel Teardown

The host commands a transmit channel teardown by writing the channel number to the CPDMA_TX_TEARDOWN register. When a teardown command is issued to an enabled transmit channel the following will occur:

- Any frame currently in transmission will complete normally
- The teardown complete bit will be set in the next sop buffer descriptor (if there is one).
- The channel head descriptor pointer will be cleared to 0.
- An interrupt will be issued to inform the host of the channel teardown.
- The software should acknowledge a teardown interrupt with a FFFF FFFCh acknowledge value

Channel teardown may be commanded on any channel at any time. The host is informed of the teardown completion by the set teardown complete buffer descriptor bit. The port does not clear any channel enables due to a teardown command. A teardown command to an inactive channel issues an interrupt that software should acknowledge with a FFFF FFFCh acknowledge value (note that there is no buffer descriptor in this case). Software may read the interrupt acknowledge location to determine if the interrupt was due to a commanded teardown. The read value will be FFFF FFFCh if the interrupt was due to a teardown command.

18.8.4.8.1.4 Transmit Rate Limiting

Transmit operations can be configured to rate limit the transmit data for each transmit priority. Rate limiting is enabled for a channel when the TX_RLIM bit associated with that channel is set in the CPDMA_DMACONTROL register. Rate limited channels must be the highest priority channels. For example, if two rate limited channels are required then TX_RLIM should be set to 11000000b with the MSB corresponding to channel 7. When any channels are configured to be rate-limiting, the priority type must be fixed for transmit. Round-robin priority type is not allowed when rate-limiting. Each of the eight

transmit priorities has an associated register to control the rate at which the priority is allowed to send data (TX_PRI(0..7)_RATE) when the channel is rate-limiting. Each priority has a send count (PRI(0..7)_SEND_CNT) and an idle count (PRI(0..7)_IDLE_CNT). The transfer rate includes the inter-packet gap (12 bytes) and the preamble (8 bytes). The rate in Mbits/second that each priority is allowed to send is controlled by the equation:

Priority Transfer rate in Mbit/s = ((PRI_IDLE_CNT/(PRI_IDLE_CNT + PRI_SEND_CNT)) × frequency × 32

Where *frequency* is the CPDMA interface clock (MAIN_CLK) frequency.

18.8.4.8.1.5 Command IDLE

The CMD_IDLE bit in the CPDMA_DMACONTROL register allows CPDMA operation to be suspended. When the idle state is commanded, the CPDMA will stop processing receive and transmit frames at the next frame boundary. Any frame currently in reception or transmission will be completed normally without suspension. For transmission, any complete or partial frame in the tx cell FIFO will be transmitted. For receive, frames that are detected by the CPDMA after the suspend state is entered are ignored. No statistics will be kept for ignored frames. Commanded idle is similar in operation to emulation control and clock stop.

18.8.4.8.2 Address Lookup Engine (ALE)

The address lookup engine (ALE) processes all received packets to determine which port(s) if any that the packet should the forwarded to. The ALE uses the incoming packet received port number, destination address, source address, length/type, and VLAN information to determine how the packet should be forwarded. The ALE outputs the port mask to the switch fabric that indicates the port(s) the packet should be forwarded to. The ALE is enabled when the ENABLE_ALE bit in the ALE_CONTROL register is set. All packets are dropped when the ENABLE_ALE bit is cleared to 0.

In normal operation, the CPGMAC_SL modules are configured to issue an abort, instead of an end of packet, at the end of a packet that contains an error (runt, frag, oversize, jabber, crc, alignment, code etc.) or at the end of a mac control packet. However, when the SL_MACCONTROL configuration bit(s) RX_CEF_EN, RX_CSF_EN, or RX_CMF_EN are set, error frames, short frames or mac control frames have a normal end of packet instead of an abort at the end of the packet. When the ALE receives a packet that contains errors (due to a set header error bit), or a mac control frame and does not receive an abort, the packet will be forwarded only to the host port (port 0). No ALE learning occurs on packets with errors or mac control frames. Learning is based on source address and lookup is based on destination address.

The ALE may be configured to operate in bypass mode by setting the ALE_BYPASS bit in the ALE_CONTROL register. When in bypass mode, all CPGMAC_SL received packets are forwarded only to the host port (port 0). Packets from the two ports can be on separate RX DMA channels by configuring the P0_CPDMA_RX_CH_MAP register. In bypass mode, the ALE processes host port transmit packets the same as in normal mode. In general, packets would be directed by the host in bypass mode.

The ALE may be configured to operate in OUI deny mode by setting the ENABLE_OUI_DENY bit in the ALE_CONTROL register. When in OUI deny mode, a packet with a non-matching OUI source address will be dropped unless the destination address matches a multicast table entry with the super bit set. Broadcast packets will be dropped unless the broadcast address is entered into the table with the super bit set. Unicast packets will be dropped unless the unicast address is in the table with block and secure both set (supervisory unicast packet).

Multicast supervisory packets are designated by the super bit in the table entry. Unicast supervisory packets are indicated when block and secure are both set. Supervisory packets are not dropped due to rate limiting, OUI, or VLAN processing.

18.8.4.8.2.1 Address Table Entry

The ALE table contains 1024 entries. Each table entry represents a free entry, an address, a VLAN, an address/VLAN pair, or an OUI address. Software should ensure that there are not double address entries in the table. The double entry used would be indeterminate. Reserved table bits must be written with zeroes.

Source Address learning occurs for packets with a unicast, multicast or broadcast destination address and a unicast or multicast (including broadcast) source address. Multicast source addresses have the group bit (bit 40) cleared before ALE processing begins, changing the multicast source address to a unicast source address. A multicast address of all ones is the broadcast address which may be added to the table. A learned unicast source address is added to the table with the following control bits:

Table 18-781. Learned Address Control Bits

Bit(s)	Value
unicast_type	11
Block	0
Secure	0

If a received packet has a source address that is equal to the destination address then the following occurs:

- The address is learned if the address is not found in the table.
- The address is updated if the address is found.
- The packet is dropped.

Table Entry Type

00 - Free Entry

01 - Address Entry : unicast or multicast determined by destination **address bit 40.**

10 - VLAN entry

11 - VLAN Address Entry : unicast or multicast determined by **address bit 40.**

18.8.4.8.2.1.1 Free Table Entry

Table 18-782. Free (Unused) Address Table Entry Bit Values

71:62	61:60	59:0
Reserved	ENTRY_TYPE(00)	Reserved

18.8.4.8.2.1.2 Multicast Address Table Entry

Table 18-783. Multicast Address Table Entry Bit Values

71:69	68:66	65	64	63:62	61:60	59:48	47:0
Reserved	PORT_MASK	SUPER	Reserved	MCAST_FWD_STATE	ENTRY_TYPE(01)	Reserved	MULTICAST_ADDRESS

Supervisory Packet (SUPER)

When set, this field indicates that the packet with a matching multicast destination address is a supervisory packet.

0: Non-supervisory packet

1: Supervisory packet

Port Mask(2:0) (PORT_MASK)

This 3-bit field is the port bit mask that is returned with a found multicast destination address. There may be multiple bits set indicating that the multicast packet may be forwarded to multiple ports (but not the receiving port).

Multicast Forward State (MCAST_FWD_STATE)

Indicates the port state(s) required for the received port on a destination address lookup in order for the multicast packet to be forwarded to the transmit port(s). A transmit port must be in the Forwarding state in order to forward the packet. If the transmit PORT_MASK has multiple set bits then each forward decision is independent of the other transmit port(s) forward decision.

00 - Forwarding

01 - Blocking/Forwarding/Learning

10 - Forwarding/Learning

11 - Forwarding

The forward state test returns a true value if both the RX and TX ports are in the required state.

Table Entry Type (ENTRY_TYPE)

Address entry type. Unicast or multicast determined by address bit 40.

01: Address entry. Unicast or multicast determined by address bit 40.

Packet Address (MULTICAST_ADDRESS)

This is the 48-bit packet MAC address. For an OUI address, only the upper 24-bits of the address are used in the source or destination address lookup. Otherwise, all 48-bits are used in the lookup.

18.8.4.8.2.1.3 VLAN/Multicast Address Table Entry

Table 18-784. VLAN/Multicast Address Table Entry Bit Values

71:69	68:66	65	64	63:62	61:60	59:48	47:0
Reseved	PORT_MASK	SUPER	Reserved	MCAST_FWD_STATE	ENTRY_TYPE(11)	VLAN_ID	MULTICAST_ADDRESS

Supervisory Packet (SUPER)

When set, this field indicates that the packet with a matching multicast destination address is a supervisory packet.

0: Non-supervisory packet

1: Supervisory packet

Port Mask(2:0) (PORT_MASK)

This 3-bit field is the port bit mask that is returned with a found multicast destination address. There may be multiple bits set indicating that the multicast packet may be forwarded to multiple ports (but not the receiving port).

Multicast Forward State (MCAST_FWD_STATE)

Indicates the port state(s) required for the received port on a destination address lookup in order for the multicast packet to be forwarded to the transmit port(s). A transmit port must be in the Forwarding state in order to forward the packet. If the transmit PORT_MASk has multiple set bits then each forward decision is independent of the other transmit port(s) forward decision.

00 - Forwarding

01 - Blocking/Forwarding/Learning

10 - Forwarding/Learning

11 - Forwarding

The forward state test returns a true value if both the RX and TX ports are in the required state.

Table Entry Type (ENTRY_TYPE)

Address entry type. Unicast or multicast determined by address bit 40.

11: VLAN address entry. Unicast or multicast determined by address bit 40.

VLAN ID (VLAN_ID)

The unique identifier for VLAN identification. This is the 12-bit VLAN ID.

Packet Address (MULTICAST_ADDRESS)

This is the 48-bit packet MAC address. For an OUI address, only the upper 24-bits of the address are used in the source or destination address lookup. Otherwise, all 48-bits are used in the lookup.

18.8.4.8.2.1.4 Unicast Address Table Entry

Table 18-785. Unicast Address Table Entry Bit Values

71:68	67:66	65	64	63:62	61:60	59:48	47:0
Reseved	PORT_NUMBER	BLOCK	SECURE	UNICAST_TYPE (00) or (X1)	ENTRY_TYPE(01)	Reserved	UNICAST_ADDRESS

Port Number (PORT_NUMBER)

This field indicates the port number (not port mask) that the packet with a unicast destination address may be forwarded to. Packets with unicast destination addresses are forwarded only to a single port (but not the receiving port).

Block (BLOCK)

The block bit indicates that a packet with a matching source or destination address should be dropped (block the address).

0 - Address is not blocked.

1 - Drop a packet with a matching source or destination address (secure

must be zero)

If block and secure are both set, then they no longer mean block and secure. When both are set, the block and secure bits indicate that the packet is a unicast supervisory (super) packet and they determine the unicast forward state test criteria. If both bits are set then the packet is forwarded if the receive port is in the Forwarding/Blocking/Learning state. If both bits are not set then the packet is forwarded if the receive port is in the Forwarding state.

Secure (SECURE)

This bit indicates that a packet with a matching source address should be dropped if the received port number is not equal to the table entry port_number.

0 - Received port number is a don't care.

1 - Drop the packet if the received port is not the secure port for the source

address and do not update the address (block must be zero)

Unicast Type (UNICAST_TYPE)

This field indicates the type of unicast address the table entry contains.

00 - Unicast address that is not ageable.

01 - Ageable unicast address that has not been touched.

10 - OUI address - lower 24-bits are don't cares (not ageable).

11 - Ageable unicast address that has been touched.

Table Entry Type (ENTRY_TYPE)

Address entry. Unicast or multicast determined by address bit 40.

01: Address entry. Unicast or multicast determined by address bit 40.

Packet Address (UNICAST_ADDRESS)

This is the 48-bit packet MAC address. All 48-bits are used in the lookup.

18.8.4.8.2.1.5 OUI Unicast Address Table Entry

Table 18-786. OUI Unicast Address Table Entry Bit Values

71:64	63:62	61:60	59:48	47:24	23:0
Reserved	UNICAST_TYPE(10)	ENTRY_TYPE(01)	Reserved	UNICAST_OUI	Reserved

Unicast Type (UNICAST_TYPE)

This field indicates the type of unicast address the table entry contains.

00 - Unicast address that is not ageable.

01 - Ageable unicast address that has not been touched.

10 - OUI address - lower 24-bits are don't cares (not ageable).

11 - Ageable unicast address that has been touched.

Table Entry Type (ENTRY_TYPE)

Address entry. Unicast or multicast determined by address bit 40.

01: Address entry. Unicast or multicast determined by address bit 40.

Packet Address (UNICAST_OUI)

For an OUI address, only the upper 24-bits of the address are used in the source or destination address lookup.

18.8.4.8.2.1.6 VLAN/Unicast Address Table Entry

Table 18-787. Unicast Address Table Entry Bit Values

71:68	67:66	65	64	63:62	61:60	59:48	47:0
Reseved	PORT_NUMBER	BLOCK	SECURE	UNICAST_TYPE (00) or (X1)	ENTRY_TYPE (11)	VLAN_ID	UNICAST_ADDRESS

Port Number (PORT_NUMBER)

This field indicates the port number (not port mask) that the packet with a unicast destination address may be forwarded to. Packets with unicast destination addresses are forwarded only to a single port (but not the receiving port).]

Block (BLOCK)

The block bit indicates that a packet with a matching source or destination address should be dropped (block the address).

0 - Address is not blocked.

1 - Drop a packet with a matching source or destination address (secure must be zero)

If block and secure are both set, then they no longer mean block and secure. When both are set, the block and secure bits indicate that the packet is a unicast supervisory (super) packet and they determine the unicast forward state test criteria. If both bits are set then the packet is forwarded if the receive port is in the Forwarding/Blocking/Learning state. If both bits are not set then the packet is forwarded if the receive port is in the Forwarding state.

Secure (SECURE)

This bit indicates that a packet with a matching source address should be dropped if the received port number is not equal to the table entry PORT_NUMBER.

0 - Received port number is a don't care.

1 - Drop the packet if the received port is not the secure port for the source

address and do not update the address (block must be zero)

Unicast Type (UNICAST_TYPE)

This field indicates the type of unicast address the table entry contains.

00 - Unicast address that is not ageable.

01 - Ageable unicast address that has not been touched.

10 - OUI address - lower 24-bits are don't cares (not ageable).

11 - Ageable unicast address that has been touched.

Table Entry Type (ENTRY_TYPE)

Address entry. Unicast or multicast determined by address bit 40.

11: VLAN address entry. Unicast or multicast determined by address bit 40.

VLAN ID (VLAN_ID)

The unique identifier for VLAN identification. This is the 12-bit VLAN ID.

Packet Address (UNICAST_ADDRESS)

This is the 48-bit packet MAC address. All 48-bits are used in the lookup.

18.8.4.8.2.1.7 VLAN Table Entry

Table 18-788. VLAN Table Entry

71:62	61:60	59:48	47:27	26:24	23:19	18:16	15:11	10:8	7:3	2:0
Reserved	ENTRY_TYPE(10)	VLAN_ID	Reserved	FORCE_UNTAGGED_EGRESS	Reserved	REG_MCAST_FLOOD_MASK	Reserved	UNREG_MCAST_FLOOD_MASK	Reserved	VLAN_MEMBER_LIST

Table Entry Type (ENTRY_TYPE)

10: VLAN entry

VLAN ID (VLAN_ID)

The unique identifier for VLAN identification. This is the 12-bit VLAN ID.

Force Untagged Packet Egress (FORCE_UNTAGGED_EGRESS)

This field causes the packet VLAN tag to be removed on egress (except on port 0).

Registered Multicast Flood Mask (REG_MCAST_FLOOD_MASK)

Mask [port 2-1-0] used for multicast when the multicast address is found.

Unregistered Multicast Flood Mask (UNREG_MCAST_FLOOD_MASK)

Mask [port 2-1-0] used for multicast when the multicast address is not found.

VLAN Member List (VLAN_MEMBER_LIST)

This 3-bit field indicates which port(s) are members of the associated VLAN.

18.8.4.8.2.2 Packet Forwarding Processes

There are four processes that an incoming received packet may go through to determine packet forwarding. The processes are Ingress Filtering, VLAN_Aware Lookup, VLAN_Unaware Lookup, and Egress.

Packet processing begins in the Ingress Filtering process. Each port has an associated packet forwarding state that can be one of four values (Disabled, Blocked, Learning, or Forwarding). The default state for all ports is Disabled. The host sets the packet forwarding state for each port.

In the packet ingress process (receive packet process), there is a forward state test for unicast destination addresses and a forward state test for multicast addresses. The multicast forward state test indicates the port states required for the receiving port in order for the multicast packet to be forwarded to the transmit port(s). A transmit port must be in the Forwarding state for the packet to be forwarded for transmission. The mcast_fwd_state indicates the required port state for the receiving port as indicated in the preceding table. The unicast forward state test indicates the port state required for the receiving port in order to forward the unicast packet. The transmit port must be in the Forwarding state in order to forward the packet. The block and secure bits determine the unicast forward state test criteria. If both bits are set then the packet is forwarded if the receive port is in the Forwarding/Blocking/Learning state. If both bits are not set then the packet is forwarded if the receive port is in the Forwarding state. The transmit port must be in the Forwarding state regardless. The forward state test used in the ingress process is determined by the destination address packet type (multicast/unicast).

In general, packets received with errors are dropped by the address lookup engine without learning, updating, or touching the address. The error condition and the abort are indicated by the CPGMAC_SL to the ALE. Packets with errors may be passed to the host (not aborted) by a CPGMAC_SL port, if the port has the RX_CMF_EN, RX_CEF_EN, or RX_CSF_EN bit(s) set. Error packets that are passed to the host by the CPGMAC_SL are considered to be bypass packets by the ALE and are sent only to the host. Error packets do not learn, update, or touch addresses regardless of whether they are aborted or sent to the host. Packets with errors received by the host are forwarded as normal.

The following control bits are in the SL_MACCONTROL register:

- [22] RX_CEF_EN - enables frames that are fragments, long, jabber, CRC, code, and alignment errors to be forwarded
- [23] RX_CSF_EN - enables short frames to be forwarded
- [24] RX_CMF_EN - enables MAC control frames to be forwarded.

18.8.4.8.2.3 Learning Process

The learning process is applied to each receive packet that is not aborted. The learning process is a concurrent process with the packet forwarding process.

18.8.4.8.2.4 VLAN Aware Mode

The CPSW_3G is in VLAN aware mode when the VLAN_AWARE bit is set in the CPSW_CONTROL register. In VLAN aware mode, ports 0 receive packets (out of the CPSW_3G) may or may not be VLAN encapsulated depending on the RX_VLAN_ENCAP bit in the CPSW_CONTROL register. Port 0 receive packet data is never modified. VLAN is not removed regardless of the force untagged egress bit for Port 0. VLAN encapsulated receive packets have a 32-bit VLAN header encapsulation word added to the packet data. VLAN encapsulated packets are specified by a set rx_vlan_encap bit in the packet buffer descriptor.

Port 0 transmit packets are never VLAN encapsulated (encapsulation is not allowed).

In VLAN aware mode, transmitted packet data is changed depending on the packet type (pkt_type), packet priority (pkt_pri), and VLAN information.

18.8.4.8.2.5 VLAN Unaware Mode

The CPSW_3G is in VLAN unaware mode when the VLAN_AWARE bit is cleared to 0 in the CPSW_CONTROL register. Port 0 receive packets (out of the CPSW_3G) may or may not be VLAN encapsulated depending on the RX_VLAN_ENCAP bit in the CPSW_CONTROL register. Port 0 transmit packets are never VLAN encapsulated.

18.8.4.8.3 Packet Priority Handling

Packets are received on three ports, two are CPGMAC_SL Ethernet ports and the third port is the CPPI host port. Received packets have a received packet priority (0 to 7, with 7 being the highest priority).

The received packet priority is the port priority for untagged packets, and the actual packet priority for priority tagged and VLAN tagged packets. The received packet priority is mapped through the receive ports associated packet priority to header packet priority mapping register to obtain the header packet priority (the CPDMA RX and TX nomenclature is reversed from the CPGMAC_SL nomenclature).

The header packet priority is mapped through the header priority to switch priority mapping register to obtain the hardware switch priority (0 to 3, with 3 being the highest priority). The header packet priority is then used as the actual transmit packet priority if the VLAN information is to be sent on egress.

18.8.4.8.4 FIFO Memory Control

Each of the three CPSW_3G ports has an identical associated FIFO. Each FIFO contains a single logical receive queue and four logical transmit queues (priority 0 through 3). Each FIFO memory contains 20,480 bytes (20k) total organized as 2560 by 64-bit words contained in a single memory instance. The FIFO memory is used for the associated port transmit and receive queues. The TX_MAX_BLKS field in the FIFOs associated Px_MAX_BLKS register determines the maximum number of 1k FIFO memory blocks to be allocated to the four logical transmit queues (transmit total). The RX_MAX_BLKS field in the FIFO's associated Px_MAX_BLKS register determines the maximum number of 1k memory blocks to be allocated to the logical receive queue. The TX_MAX_BLKS value plus the RX_MAX_BLKS value must sum to 20 (the total number of blocks in the FIFO). If the sum were less than 20, then some memory blocks would be unused. The default is 17 (decimal) transmit blocks and three receive blocks. The FIFOs follow the naming convention of the Ethernet ports. Host Port is Port0 and External Ports are Port1 and Port2.

18.8.4.8.5 FIFO Transmit Queue Control

There are four transmit queues in each transmit FIFO. Software has some flexibility in determining how packets are loaded into the queues and on how packet priorities are selected for transmission (how packets are removed and transmitted from queues). All ports on the switch have identical FIFO's. For the purposes of the below the transmit FIFO is switch egress even though the port 0 transmit FIFO is connected to the CPDMA receive (also switch egress). The CPDMA nomenclature is reversed from the CPGMAC_SL nomenclature due to legacy reasons.

18.8.4.8.5.1 Normal Priority Mode

When operating in normal mode, lower priority frames are dropped before higher priority frames. The intention is to give preference to higher priority frames. Priority 3 is the highest priority and is allowed to fill the FIFO. Priority 2 will drop packets if the packet is going to take space in the last 2k available. Priority 1 will drop packets if the packet is going to take space in the last 4k available. Priority 0 will drop packets if the packet is going to take space in the last 6k available. If fewer than 4 priorities are to be implemented then the priorities should be mapped such that the highest priorities are used. For example, if two priorities are going to be used then all packets should be mapped to priorities 3 and 2 and priorities 1 and 0 should be unused. Priority escalation may be used in normal priority mode if desired. Normal priority mode is configured as described below:

- Select normal priority mode by setting TX_IN_SEL = 00 for all ports (default value in P0/1/2_TX_IN_CTL)
- Configure priority mapping to use only the highest priorities if less than 4 priorities are used. Refer to Section 18.8.4.8.3, *Packet Priority Handling*.

18.8.4.8.5.2 Dual MAC Mode

When operating in dual MAC mode the intention is to transfer packets between ports 0 and 1 and ports 0 and 2, but not between ports 1 and 2. Each CPGMAC_SL appears as a single MAC with no bridging between MAC's. Each CPGMAC_SL has at least one unique (not the same) mac address.

Dual MAC mode is configured as described below:

- Set the ALE_VLAN_AWARE bit in the ALE_CONTROL register. This bit configures the ALE to process

in VLAN aware mode. The CPSW_3G VLAN aware bit (VLAN_AWARE in CPSW_CONTROL) determines how packets VLAN's are processed on CPGMAC_SL egress and does not affect how the ALE processes packets or the packet destination. The CPSW_3G VLAN aware bit may be set or not as required (must be set, if VLAN's are to exit the switch).

- Configure the Port 1 to Port 0 VLAN
  - Add a VLAN Table Entry with ports 0 and 1 as members (clear the flood masks).
  - Add a VLAN/Unicast Address Table Entry with the Port1/0 VLAN and a port number of 0. Packets received on port 1 with this unicast address will be sent only to port 0 (egress). If multiple mac addresses are desired for this port then multiple entries of this type may be configured.
- Configure the Port 2 to Port 0 VLAN
  - Add a VLAN Table Entry with ports 0 and 2 as members (clear the flood masks).
  - Add a VLAN/Unicast Address Table Entry with the Port2/0 VLAN and a port number of 0. Packets received on port 2 with this unicast address will be sent only to port 0 (egress). If multiple mac addresses are desired for this port then multiple entries of this type may be configured.
- Packets from the host (port 0) to ports 1 and 2 should be directed. If directed packets are not desired then VLAN with addresses can be added for both destination ports.
- Select the dual mac mode on the port 0 FIFO by setting TX_IN_SEL = 01 in P0_TX_IN_CTL. The intention of this mode is to allow packets from both ethernet ports to be written into the FIFO without one port starving the other port.
- The priority levels may be configured such that packets received on port 1 egress on one CPDMA RX channel while packets received on port 2 egress on a different CPDMA RX channel.

18.8.4.8.5.3 Rate Limit Mode

Rate-limit mode is intended to allow some CPDMA transmit (switch ingress) channels and some CPGMAC_SL FIFO priorities (switch egress) to be rate-limited. Non rate-limited traffic (bulk traffic) is allowed on non rate-limited channels and FIFO priorities. The bulk traffic does not impact the rate-limited traffic. Rate-limited traffic must be configured to be sent to rate-limited queues (via packet priority handling). The allocated rates for rate-limited traffic must not be oversubscribed. For example, if port 1 is sending 15% rate limited traffic to port 2 priority 3, and port 0 is also sending 10% rate-limited traffic to port 2 priority 3, then the port 2 priority 3 egress rate must be configured to be 25% plus a percent or two for margin. The switch must be configured to allow some percentage of non rate-limited traffic. Non rate-limited traffic must be configured to be sent to non rate-limited queues. No packets from the host should be dropped, but non rate-limited traffic received on an ethernet port can be dropped. Rate-limited mode is configured as shown below:

1. Set TX_IN_SEL = 10 in P1/2_TX_IN_CTL to enable ports 1 and 2 transmit FIFO inputs to be configured for rate-limiting queues. Enabling a queue to be rate-limiting with this field affects only the packet being loaded into the FIFO, it does not configure the transmit for queue shaping.
2. Configure the number of rate-limited queues for port 1 and 2 transmit FIFO's by setting the TX_RATE_EN field in P1/2_TX_IN_CTL. Rate limited queues must be the highest number. For example, if there are two rate limited queues then 1100 would be written to this field for priorities 3 and 2. This field enables the FIFO to allow rate-limited traffic into rate-limited queues while discriminating against non rate-limited queues.
3. Set P1_PRIN_SHAPE_EN and P2_PRIN_SHAPE_EN in the CPSW_PTYPE register. These bits determine which queues actually shape the output data stream. In general, the same priorities that are set in TX_RATE_EN are set in these bits as well, but the FIFO input and output enable bits are separate to allow rate-limiting from the host to non shaped channels if desired. When queue shaping is not enabled for a queue then packets are selected for egress based on priority. When queue shaping is enabled then packets are selected for egress based on queue percentages. If shaping is required on a single queue then it must be priority 3 (priorities 2, 1 and 0 are strict priority). If shaping is required on two queues then it must be on priorities 2 and 3 (priorities 1 and 0 are strict priority). If shaping is required on three queues then it must be priorities 3, 2, and 1 (priority 0 would then get the leftovers). Priority shaping follows the requirements in the IEEE P802.1Qav/D6.0 specification. Priority shaping is not compatible with priority escalation (escalation must be disabled).
4. P0_TX_IN_CTL[17:16] TX_IN_SEL should be set to 00, so Port 0 egress (CPDMA RX) is not rate-limited.

5. The CPDMA is configured for rate-limited transmit (switch ingress) channels by setting the highest bits of the TX_RLIM field in the CPDMA_DMACONTROL register. If there are two rate-limited channels, then TX_RLIM = 11000000 (the rate limited channels must be the highest priorities). Also, the TX_PTYPE bit in the CPDMA_DMACONTROL register must be set (fixed priority mode). Rate-limited channels must go to rate-limited FIFO queues, and the FIFO queue rate must not be oversubscribed.

18.8.4.8.6 Audio Video Bridging

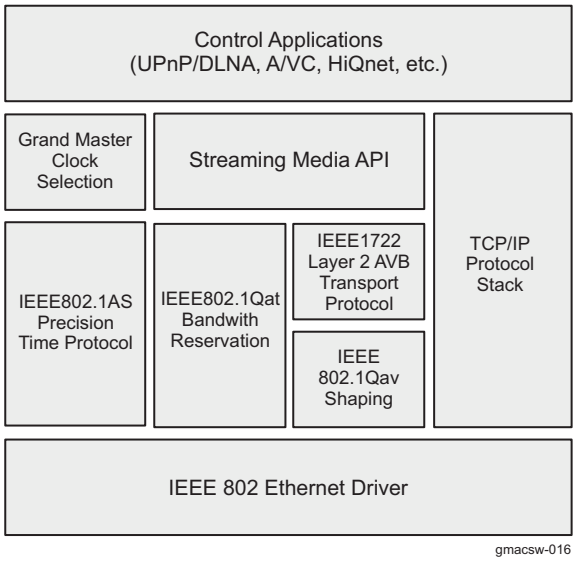
Audio Video Bridging is an ongoing project of IEEE 802.1 concerned with enabling low-latency streaming of time-sensitive audiovisual data over networks. Devices are designated as talkers (transmitters), bridges, or listeners (receivers). It is suggested that the maximum latency could be 2 ms over 7 hops for Class A devices and 20 ms over 7 hops for Class B devices. A hop is essentially a single local area network stage in the journey of a packet. Every time a bridge is encountered between one network section and another a hop is involved. One of the performance goals is that AVB streams will not use more than 75 percent of a link's bandwidth, leaving the remaining capacity for non-AVB streams.

The goal of developing AVB is simply--extend Ethernet's data-networking capabilities to the realm of reliable real-time audio/video networking.

An "Audio Video Bridging" network is one that implements a set of protocols being developed by the IEEE 802.1 Audio/Video Bridging Task Group. There are four primary differences between the proposed Audio Video Bridging architecture and existing 802 architectures (from now on the term "AVB" will be used instead of "Audio Video Bridging"):

1. Precise synchronization - IEEE 802.1AS: "*Timing and Synchronization for Time-Sensitive Applications in Bridged Local Area Networks.* "a.k.a Precision Time Protocol (PTP).
2. Traffic shaping for media streams - IEEE 802.1Qav: "*Virtual Bridged Local Area Networks:* Forwarding and Queuing for Time-Sensitive Streams."
3. Admission controls - IEEE 802.1Qat: "*Virtual Bridged Local Area Networks - Amendment 9:* Stream Reservation Protocol (SRP)."
4. Identification of non-participating devices - IEEE 802.1BA: "Audio/Video Bridging (AVB) Systems"

Figure 18-159. The Network Static with AVB



The following sections describe the media transport protocols that work within the AVB framework.

18.8.4.8.6.1 IEEE 802.1AS: Timing and Synchronization for Time-Sensitive Applications in Bridged Local

Area Networks (Precision Time Protocol (PTP))

The protocol defined by 802.1AS automatically selects a device to be the master clock, and then distributes this clock throughout the bridged LAN / IP subnet to all other network devices using link-specific transmit/receive time-stamping. However, we only use a two-step solution only on transmit. That is, we do not modify a packet with the timestamp on the way out. The timestamp packet is sent out and then a separate message with the timestamp is sent by the host afterward. Receive can be one or two step.

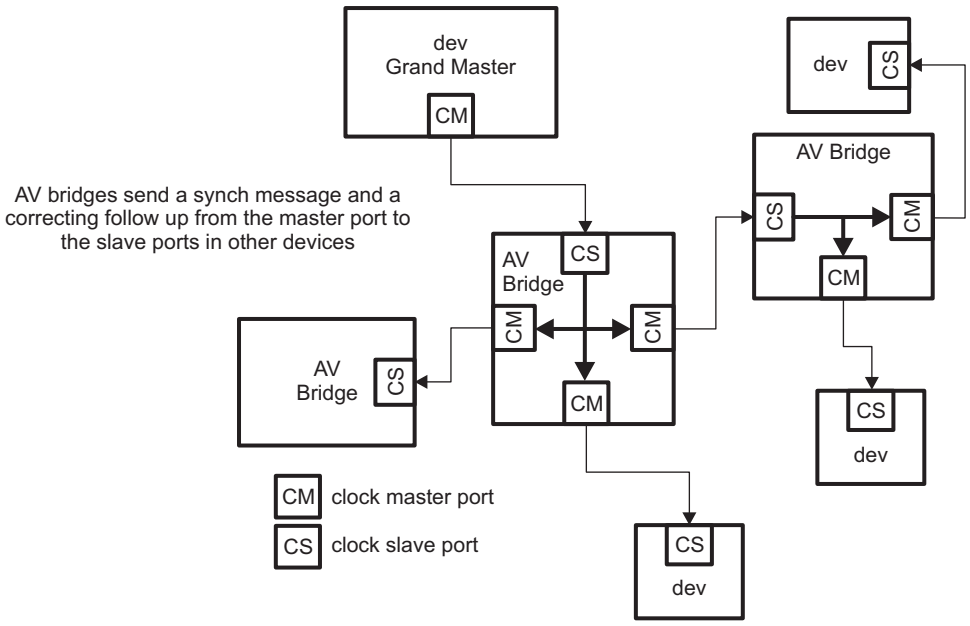
NOTE: The 802.1AS-distributed clock is not used as a media clock. Rather, the shared 802.1AS clock reference is used to regenerate the media clock at the listener/renderer. Such a reference removes the need to force the latency of the network to be constant, or compute long running averages in order to estimate the actual media rate of the transmitter in the presence of substantial network jitter. IEEE 802.1AS is based on the ratified IEEE 1588 standard.

Based on IEEE 1588:2002,Â PTP devices exchange standard Ethernet messages that synchronize network nodes to a common time reference by defining clock master selection and negotiation algorithms, link delay measurement and compensation, and clock rate matching and adjustment mechanisms.

Designed as a simplified profile of IEEE 1588, a primary difference between 1588 and IEEE 802.1AS is that PTP is a layer 2--in other words, a non-IP routable protocol. Like IEEE 1588, PTP defines an automatic method for negotiating the network clock master, the Best Master Clock Algorithm (BMCA). PTP nodes can be assigned one of eight priority levels, presumably based on clock quality. BMCA defines the underlying negotiation and signaling mechanism whose purpose is to identify the AVB LAN Grandmaster. Once a Grandmaster has been selected, synchronization automatically begins.

At the core of 802.1AS synchronization is time-stamping. In short, during PTP message ingress/egress from the 802.1AS-capable MAC, the PTP Ether type triggers the sampling of the value of a local real-time counter (RTC). Slave nodes compare the value of their RTC against the PTP Grandmaster and, by use of link delay measurement and compensation techniques, match their RTC value to the time of the AVB LAN PTP domain. After network time throughout the AVB LAN has converged, periodic SYNC and FOLLOW_UP messages provide the information that enables the PTP rate matching adjustment algorithms. The result is all PTP nodes are then synchronized to the same "Wall Clock" time. PTP assures 1-µs accuracy over seven network hops.

Figure 18-160. AVB Network & PTP Clock Entities

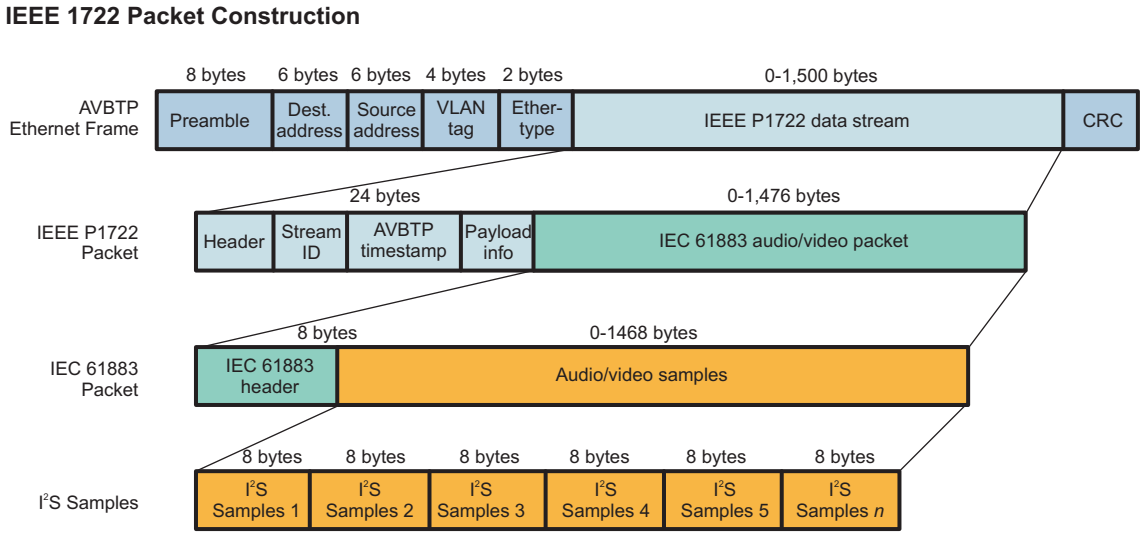


The media transport protocols that work within the AVB framework are:

18.8.4.8.6.1.1 IEEE 1722: "Layer 2 Transport Protocol for Time-Sensitive Streams"

AVBTP or 1722 sits above the IEEE 802.1 AVB plumbing and below the application layer. It acts as the conduit between an Ethernet MAC and a streaming application. AVBTP abstracts the underlying network transmission channel to enable the virtual connection of distributed audio and video CODECs over reliable Ethernet networks. A complete AVBTP Ethernet packet is shown in Figure 18-161 and illustrates how IEC 61883-6 AM824 uncompressed audio samples are encapsulated in an Ethernet frame.

Figure 18-161. IEEE 1722 Packets



1722 or AVBTP Presentation Time and Synchronization:

Synchronization in an AVB network starts with the Precision Time Protocol but ends with synchronized media clocks. PTP is responsible for synchronizing all nodes in an AVB network to identical wall clock time; not for synchronizing media clocks. In other words, PTP does not actually transport synchronized media clocks but instead provides a low-level building block crucial for managing a distributed media synchronization system.

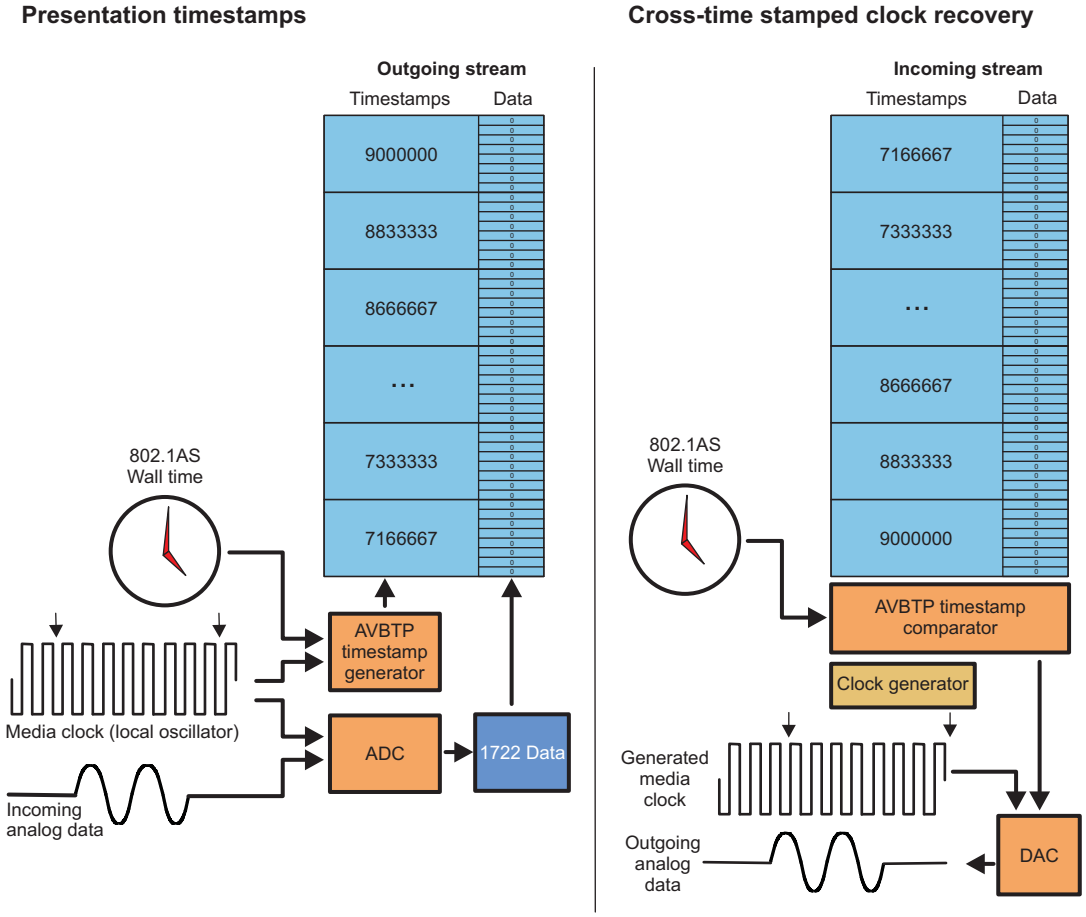
A crucial benefit of this approach is coexistence of multiple, independent media clock domains on an AVB network. Unrelated audio and video streams can simultaneously exist in the same LAN.

AVBTP assumes that AVB node media clocks are clocked by free-running oscillators. It is also assumed that the node's internal concept of wall clock time has been synchronized to the PTP Grandmaster. AVBTP media clock sources embed "AVBTP Presentation Timestamps" in AVBTP streaming packets. Figure 18-162 illustrates the relationship between PTP network time and AVBTP Presentation Timestamps.

18.8.4.8.6.1.2 IEEE 1733: Extends RTCP for RTP Streaming over AVB-supported Networks

This standard specifies the protocol, data encapsulations, connection management and presentation time procedures used to ensure interoperability between audio and video based end stations that use standard networking services provided by all IEEE 802 networks meeting QoS requirements for time-sensitive applications by leveraging the Real-time Transport Protocol (RTP) family of protocols and IEEE 802.1 Audio/Video Bridging (AVB) protocols.

Figure 18-162. Cross Time Stamping and Presentation Timestamps



18.8.4.8.6.2 IEEE 802.1Qav: "Virtual Bridged Local Area Networks: Forwarding and Queuing for Time-Sensitive Streams"

This standard allows bridges to provide guarantees for time-sensitive (that is, bounded latency and delivery variation), loss-sensitive real-time audio video (AV) data transmission (AV traffic). It specifies per priority ingress metering, priority regeneration, and timing-aware queue draining algorithms. This standard uses the timing derived from IEEE 802.1AS. Virtual Local Area Network (VLAN) tag encoded priority values are allocated, in aggregate, to segregate frames among controlled and non-controlled queues, allowing simultaneous support of both AV traffic and other bridged traffic over and between wired and wireless Local Area Networks (LANs).

Such a guarantee in bandwidth is provided by two functional entities:

- A registration protocol, which registers the service and its maximum network utilization with a device or switch (IEEE 802.1Qat: "Virtual Bridged Local Area Networks - Amendment 9: Stream Reservation Protocol (SRP)")
- A hardware bandwidth management service.
  - Receive policing and
  - Transmit rate control.

End Station Behavior

In order for an end station to successfully participate in the transmission and reception of time-sensitive streams, it is necessary for their behavior to be compatible with the operation of the forwarding and queuing mechanisms employed in bridges.

The requirements for end stations that participate as "talkers" i.e., sources of time-sensitive streams are different from the requirements that apply to "listeners", the destination station(s) for the streams.

Talker Behavior

In order for Talker-originated data streams to make use of the credit-based shaper behavior in Bridges, it is a requirement for a Talker to use the priorities that the Bridges in the network recognize as being associated with SR classes exclusively for transmitting stream data.

It is also necessary for the Talker and the Bridges in the path to the Listener(s), to have a common view of the bandwidth required in order to transmit the Talker's streams, and for that bandwidth to be reserved along the path to the Listener(s). This latter requirement can be met by means of stream reservation mechanisms, such as defined in SRP, or by other management means.

End stations that are Talkers shall exhibit transmission behavior for frames that are part of "time-sensitive streams" that is consistent with the operation of the credit-based shaper algorithm, both in terms of the way they transmit frames that are part of an individual data stream, and in terms of the way they transmit stream data frames from a Port.

In effect, the queuing model for a Talker Port (and a Listener port), and for given priorities, can be considered to look like Figure 18-163.

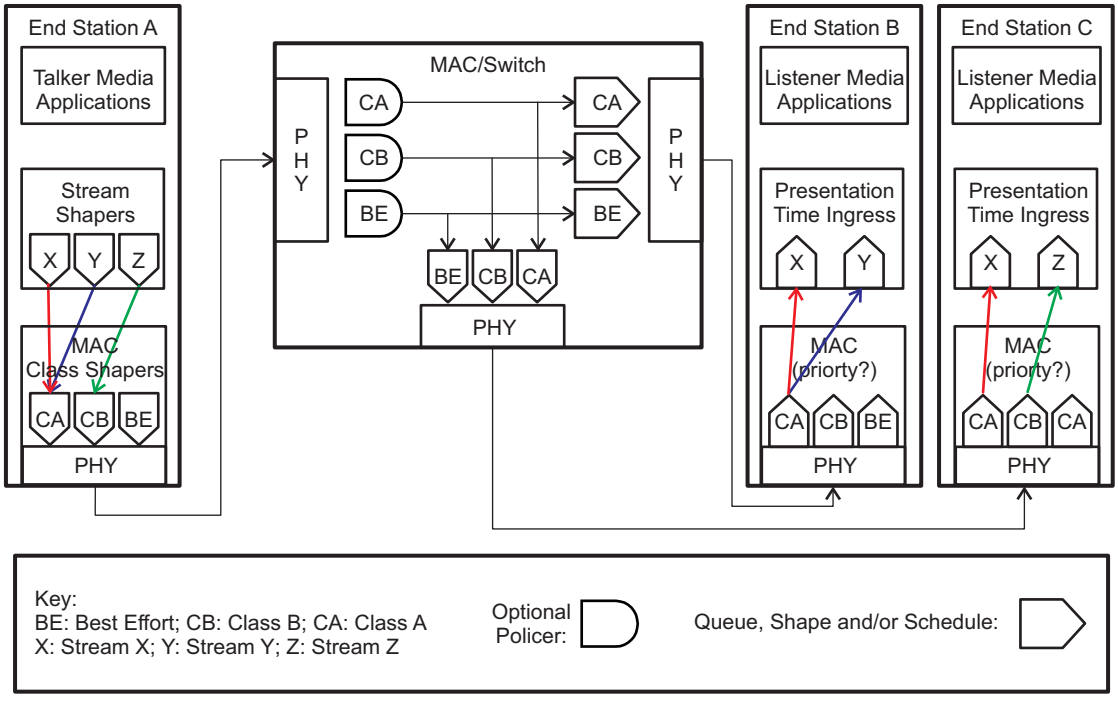
Listener Behavior

The primary requirement for a listener station is that it is capable of buffering the amount of data that could be transmitted for a stream during a time period equivalent to the accumulated maximum jitter that could be experienced by stream data frames in transmission between Talker and Listener.

From the point of view of the specification of the forwarding and queuing requirements for time-sensitive streams, it is assumed that the listener will assess the buffering required for a stream as part of the stream bandwidth reservation mechanisms employed by the implementation.

The credit-based shaper's operation details are beyond the scope of this document.

Figure 18-163. AV Stream Queuing/Policing



18.8.4.8.6.2.1 Configuring the Device for 802.1Qav Operation:

There is no dedicated register-set to be configured for the time-sensitive stream handling. The list of functional features of CPSW_3G that will have to be configured are:

- DESCRIPTORS and CHANNEL CONFIGURATIONS:
  - CPPI TX and RX descriptors
  - VLAN and Priority tags

Table 18-789. Example of TX Configuration

TX DMA CHANNEL	Packet Priority	Switch Queue Priority
7	7	3
6	5	2
5	3	1
4	1	0

Table 18-790. Example of RX Configuration

RX DMA CHANNEL	Packet Priority	Switch Queue Priority
0	7	0
0	5	0
0	3	0
0	1	0

- ALE Configuration:
  - ALE in VLAN-ware mode, Non-ALE in bypass mode.

Table 18-791. Example of Rate-limit Configurations

Register	Value	Description
CPSW_PTYPE	0x0006 0000	For Port1 -- 2 highest priority channels.
P1_TX_IN_CTL[23:20] TX_RATE_EN	0b1100	2 highest priority channels are rate limited
P1_TX_IN_CTL[17:16] TX_IN_SEL	0b10	Rate limit mode
P1_SEND_PERCENT	0x14 3E00	20% PRI7, 62% PRI5
CPDMA_DMACONTROL	0xC001	Chan7, Chan6 are Rate Limited. Round-robin selection of DMA channel.
CPDMA_TX_PRI7_RATE	0x1 0013	200 Mbps
CPDMA_TX_PRI6_RATE	0x1 0005	~600 Mbps

18.8.4.8.7 Ethernet MAC Sliver (CPGMAC_SL)

The CPGMAC_SL peripheral shall be compliant to the IEEE Std 802.3 Specification. Half-duplex mode is supported in 10/100 Mbps mode, but not in 1000 Mbps (gigabit) mode.

Features:

- Synchronous 10/100/1000 Mbit operation
- G/MII Interface
- Hardware Error handling including CRC
- Full-Duplex Gigabit operation (half-duplex gigabit is not supported)
- EtherStats and 802.3Stats RMON statistics gathering support for external statistics collection module
- Transmit CRC generation selectable on a per channel basis
- Emulation Support
- VLAN Aware Mode Support
- Hardware flow control

- Programmable Inter Packet Gap (IPG).

18.8.4.8.7.1 RGMII Interface

The CPRGMII peripheral is compliant to the RGMII specification document.

18.8.4.8.7.1.1 RGMII Features

- Supports 1000/100/10 Mbps speed
- Internal TXC delay on transmit is always enabled
- Internal TXC delay on transmit can be enabled or disabled using bits [26] RGMII2_ID_MODE_N and [25] RGMII1_ID_MODE_N of the CTRL_CORE_SMA_SW_1 register.
- MII mode is not supported

18.8.4.8.7.1.2 RGMII Receive (RX)

The CPRGMII receive (RX) interface converts the source synchronous DDR input data from the external RGMII PHY into the required G/MII (CPGMAC) signals.

18.8.4.8.7.1.3 In-Band Mode of Operation

The CPRGMII is operating in the in-band mode of operation when the EXT_EN bit of the SL_MACCONTROL register is set to 1. The link status, duplexity, and speed are determined from the RGMII input data stream RXD[3:0] when RX_CTL is deasserted, as defined in the RGMII specification. The PHY might need to be configured beforehand to output in-band data. The in-band data is indicated as shown in Table 18-792.

Table 18-792. In-Band Data

RXD3	RXD[2:1]		RXD0
Duplex status:	Link Speed:	RXC_CLK Speed:	Link Status:
0: half-duplex	00: 10-Mbps mode	2.5 MHz	0: Link is down
1: full-duplex	01: 100-Mbps mode	25 MHz	1: Link is up
	10: 1000-Mbps mode	125 MHz	
	11: reserved	reserved	

18.8.4.8.7.1.4 Forced Mode of Operation

The CPRGMII is operating in the forced mode of operation when the EXT_EN bit of the SL_MACCONTROL register is set to 0. In the forced mode of operation, the in-band data is ignored if present. The link status is forced high, and the duplexity and speed are determined from the SL_MACCONTROL[0] FULLDUPLEX and [7] GIG bits. If GIG = 1, then operation is gigabit mode. If the GIG is 0, the operation is 100 Mbps mode.

18.8.4.8.7.1.5 RGMII Transmit (TX)

The CPRGMII transmit (TX) interface converts the CPGMAC G/MII input data into the DDR RGMII format. The DDR data is then output to the external PHY.

The CPGMAC does not source the transmit error (TXERR) signal. Any transmit frame from the CPGMAC with an error (underrun) will be indicated as an error by an error CRC. Transmit error is assumed to be de-asserted at all times and is not an input into the CPRGMII module.

The TXD[7:0] data bus uses only the lower nibble. The CPRGMII will output the lower nibble twice in 10/100 mode to avoid unnecessary signal switching.

Packets will be precluded from transmission through the CPRGMII module for 4096 transmit clocks after the rising edge of RGMII_LINK. Packet transmission will begin on the first TX_CTL rising edge after the 4096 transmit clock count has expired.

18.8.4.8.7.2 Frame Classification

Received frames are proper (good) frames if they are between 64 and RX_MAXLEN in length (inclusive) and contain no errors (code/align/CRC).

Received frames are long frames if their frame count exceeds the value in the SL_RX_MAXLEN register. The SL_RX_MAXLEN register reset (default) value is 1518 (decimal). Long received frames are either oversized or jabber frames. Long frames with no errors are oversized frames. Long frames with CRC, code, or alignment errors are jabber frames.

Received frames are short frames if their frame count is less than 64 bytes. Short frames that contain no errors are undersized frames. Short frames with CRC, code, or alignment errors are fragment frames.

A received long packet will always contain RX_MAXLEN number of bytes transferred to memory (if RX_CEF_EN = 1). An example with RX_MAXLEN = 1518 is:

- If the frame length is 1518, then the packet is not a long packet and there will be 1518 bytes transferred to memory.
- If the frame length is 1519, there will be 1518 bytes transferred to memory. The last three bytes will be the first three CRC bytes.
- If the frame length is 1520, there will be 1518 bytes transferred to memory. The last two bytes will be the first two CRC bytes.
- If the frame length is 1521, there will be 1518 bytes transferred to memory. The last byte will be the first CRC byte.

If the frame length is 1522, there will be 1518 bytes transferred to memory. The last byte will be the last data byte.

18.8.4.8.8 Embedded Memories

Table 18-793. Embedded Memories

Memory Type Description	Number of Instances	
Single-port 2560 × 64 RAM	3	(Packet FIFO's)
Single-port 64-word × 1152-bit RAM	1	(ALE)
Single-port 2048-word × 32-bit RAM	1	(CPPI)

18.8.4.8.9 Flow Control

There are two types of switch flow control: CPPI port flow control and Ethernet port flow control. The CPPI and Ethernet port naming conventions for data flow into and out of the switch are reversed. For the CPPI port (port 0), transmit operations move packets from external memory into the switch and then out to either or both Ethernet transmit ports (ports 1 and 2). CPPI receive operations move packets that were received on either or both Ethernet receive ports to external memory.

18.8.4.8.9.1 CPPI Port Flow Control

The CPPI port has flow control available for transmit (switch ingress). CPPI receive operations (switch egress) do not require flow control. CPPI Transmit flow control is initiated when enabled and triggered. CPPI transmit flow control is enabled by setting the P0_FLOW_EN bit in the CPSW_FLOW_CONTROL register. CPPI transmit flow control is enabled by default on reset because host packets should not be dropped in any mode of operation.

18.8.4.8.9.2 Ethernet Port Flow Control

The Ethernet ports have flow control available for transmit and receive. Transmit flow control stops the Ethernet port from transmitting packets to the wire (switch egress) in response to a received pause frame. Transmit flow control does not depend on FIFO usage.

The ethernet ports have flow control available for receive operations (packet ingress). Ethernet port receive flow control is initiated when enabled and triggered. Packets received on an ethernet port can be sent to the other ethernet port or the CPPI port (or both). Each destination port can trigger the receive ethernet port flow control. An ethernet destination port triggers another ethernet receive flow control when the destination port is full.

When a packet is received on an ethernet port interface with enabled flow control the below occurs:

- The packet will be sent to all ports that currently have room to take the entire packet.
- The packet will be retried until successful to all ports that indicate they don't have room for the packet.

The flow control trigger to the CPGMAC_SL will be asserted until the packet has been sent, and there is room in the logical receive FIFO for packet runout from another flow control trigger (RX_PKT_CNT = 0). Ethernet port receive flow control is disabled by default on reset. Ethernet port receive flow control requires that the RX_FLOW_EN bit in the associated CPGMAC_SL be set to 1. When receive flow control is enabled on a port, the port's associated FIFO block allocation must be adjusted. The port RX allocation must increase from the default three blocks to accommodate the flow control runout. A corresponding decrease in the TX block allocation is required. If a sending port ignores a pause frame then packets may overrun on receive (and be dropped) but will not be dropped on transmit. If flow control is disabled for G/MII ports, then any packets that are dropped are dropped on transmit and not on receive.

18.8.4.8.9.2.1 Receive Flow Control

When enabled and triggered, receive flow control is initiated to limit the CPGMAC_SL from further frame reception. Half-duplex mode receive flow control is collision based while full duplex mode issues 802.3X pause frames. In either case, receive flow control prevents frame reception by issuing the flow control appropriate for the current mode of operation. Receive flow control is enabled by the RX_FLOW_EN bit in the SL_MACCONTROL register. Receive flow control is triggered (when enabled) when the RX_FLOW_TRIGGER input is asserted. The CPGMAC_SL is configured for collision or IEEE 802.3X flow control via the FULLDUPLEX bit in the SL_MACCONTROL register.

18.8.4.8.9.2.1.1 Collision Based Receive Buffer Flow Control

Collision-based receive buffer flow control provides a means of preventing frame reception when the port is operating in half-duplex mode (FULLDUPLEX is cleared in SL_MACCONTROL). When receive flow control is enabled and triggered, the port will generate collisions for received frames. The jam sequence transmitted will be the twelve byte sequence C3.C3.C3.C3.C3.C3.C3.C3.C3.C3.C3.C3 (hex). The jam sequence will begin no later than approximately as the source address starts to be received. Note that these forced collisions will not be limited to a maximum of 16 consecutive collisions, and are independent of the normal back-off algorithm. Receive flow control does not depend on the value of the incoming frame destination address. A collision will be generated for any incoming packet, regardless of the destination address.

18.8.4.8.9.2.1.2 IEEE 802.3X Based Receive Flow Control

IEEE 802.3x based receive flow control provides a means of preventing frame reception when the port is operating in full-duplex mode (FULLDUPLEX is set in SL_MACCONTROL). When receive flow control is enabled and triggered, the port will transmit a pause frame to request that the sending station stop transmitting for the period indicated within the transmitted pause frame.

The CPGMAC_SL will transmit a pause frame to the reserved multicast address at the first available opportunity (immediately if currently idle, or following the completion of the frame currently being transmitted). The pause frame will contain the maximum possible value for the pause time (FFFFh). The MAC will count the receive pause frame time (decrements FF00h down to 0) and retransmit an outgoing pause frame if the count reaches zero. When the flow control request is removed, the MAC will transmit a pause frame with a zero pause time to cancel the pause request.

Note that transmitted pause frames are only a request to the other end station to stop transmitting. Frames that are received during the pause interval will be received normally (provided the RX FIFO is not full).

Pause frames will be transmitted if enabled and triggered regardless of whether or not the port is observing the pause time period from an incoming pause frame.

The CPGMAC_SL will transmit pause frames as:

- The 48-bit reserved multicast destination address 01.80.C2.00.00.01.
- The 48-bit source address - from SL_SA[47:0] input.
- The 16-bit length/type field containing the value 88.08
- The 16-bit pause opcode equal to 00.01
- The 16-bit pause time value FF.FF. A pause-quantum is 512 bit-times. Pause frames sent to cancel a pause request will have a pause time value of 00.00.
- Zero padding to 64-byte data length (The CPGMAC_SL will transmit only 64 byte pause frames).
- The 32-bit frame-check sequence (CRC word).

All quantities above are hexadecimal and are transmitted most-significant byte first. The least-significant bit is transferred first in each byte.

If RX_FLOW_EN is cleared to 0 while the pause time is nonzero, then the pause time will be cleared to 0 and a zero count pause frame will be sent.

18.8.4.8.9.2.2 Transmit Flow Control

Incoming pause frames are acted upon, when enabled, to prevent the CPGMAC_SL from transmitting any further frames. Incoming pause frames are only acted upon when the FULLDUPLEX and TX_FLOW_EN bits in the SL_MACCONTROL register are set. Pause frames are not acted upon in half-duplex mode. Pause frame action will be taken if enabled, but normally the frame will be filtered and not transferred to memory. MAC control frames will be transferred to memory if the RX_CMF_EN (copy MAC frames) bit in the SL_MACCONTROL register is set. The TX_FLOW_EN and FULLDUPLEX bits effect whether or not MAC control frames are acted upon, but they have no effect upon whether or not MAC control frames are transferred to memory or filtered.

Pause frames are a subset of MAC Control Frames with an opcode field = 0001h. Incoming pause frames will only be acted upon by the port if:

- TX_FLOW_EN is set in SL_MACCONTROL register, and
- the frame's length is 64 to SL_RX_MAXLEN bytes inclusive, and
- the frame contains no CRC error or align/code errors.

The pause time value from valid frames will be extracted from the two bytes following the opcode. The pause time will be loaded into the port's transmit pause timer and the transmit pause time period will begin.

If a valid pause frame is received during the transmit pause time period of a previous transmit pause frame then:

- if the destination address is not equal to the reserved multicast address or any enabled or disabled unicast address, then the transmit pause timer will immediately expire, or
- if the new pause time value is zero then the transmit pause timer will immediately expire, else
- the port transmit pause timer will immediately be set to the new pause frame pause time value. (Any remaining pause time from the previous pause frame will be discarded).

If TX_FLOW_EN in SL_MACCONTROL register is cleared, then the pause-timer will immediately expire.

The port will not start the transmission of a new data frame any sooner than 512-bit times after a pause frame with a non-zero pause time has finished being received (MRXDV going inactive). No transmission will begin until the pause timer has expired (the port may transmit pause frames in order to initiate outgoing flow control). Any frame already in transmission when a pause frame is received will be completed and unaffected.

Incoming pause frames consist of the below:

- A 48-bit destination address equal to:
  - The reserved multicast destination address 01.80.C2.00.00.01
  - , or the CPGMAC_SL SL_SA [47:0] input.
- The 48-bit source address of the transmitting device.
- The 16-bit length/type field containing the value 88.08
- The 16-bit pause opcode equal to 00.01
- The 16-bit pause_time. A pause-quantum is 512 bit-times.
- Padding to 64-byte data length.
- The 32-bit frame-check sequence (CRC word).

All quantities above are hexadecimal and are transmitted most-significant byte first. The least-significant bit is transferred first in each byte.

The padding is required to make up the frame to a minimum of 64 bytes. The standard allows pause frames longer than 64 bytes to be discarded or interpreted as valid pause frames. The CPGMAC_SL will recognize any pause frame between 64 bytes and RX_MAXLEN bytes in length.

18.8.4.8.10 Short Gap

The port 1 (and port 2) transmit inter-packet gap (IPG) may be shortened by eight bit times when enabled and triggered. The TX_SHORT_GAP_EN bit in the SL_MACCONTROL register enables the gap to be shortened when triggered. The condition is triggered when the port 1 (port 2) transmit FIFO has a user defined number of FIFO blocks used. The port 1 transmit FIFO blocks used determines if the port 1 gap is shortened, and the port 2 transmit FIFO blocks used determines if the port 2 gap is shortened. The CPSW_GAP_THRESH register value determines the port 1 short gap threshold, and the CPSW_GAP_THRESH register value determines the port 2 short gap threshold.

18.8.4.8.11 Switch Latency

The CPSW_3G is a store and forward switch. The switch latency is defined as the amount of time between the end of packet reception of the received packet to the start of the output packet transmit.

Table 18-794. Switch Latency

Mode	Latency
Gig (1000)	880 ns
100	1.3 µs
10	6.5 µs

18.8.4.8.12 Emulation Control

The emulation control input (EMUSUSP) and submodule emulation control registers allow CPSW_3G operation to be completely or partially suspended. There are three CPSW_3G submodules that contain emulation control registers (CPGMAC_SL1, CPGMAC_SL2, and CPDMA). The submodule emulation control registers must be accessed to facilitate CPSW_3G emulation control. The CPSW_3G module enters the emulation suspend state if all three submodules are configured for emulation suspend and the emulation suspend input is asserted. A partial emulation suspend state is entered if one or two submodules is configured for emulation suspend and the emulation suspend input is asserted. Emulation suspend occurs at packet boundaries. The emulation control feature is implemented for compatibility with other peripherals.

CPGMAC_SL Emulation Control

The emulation control input (TBEMUSUP) and register bits (SOFT and FREE bits in the SL_EMCONTROL register) allow CPGMAC_SL operation to be suspended. When the emulation suspend state is entered, the CPGMAC_SL will stop processing receive and transmit frames at the next frame boundary. Any frame currently in reception or transmission will be completed normally without suspension. For receive, frames that are detected by the CPGMAC_SL after the suspend state is entered are ignored. Emulation control is implemented for compatibility with other peripherals.

CPDMA Emulation Control

The emulation control input (TBEMUSUP) and register bits (SOFT and FREE bits in the CPDMA_EMCONTROL register) allow CPDMA operation to be suspended. When the emulation suspend state is entered, the CPDMA will stop processing receive and transmit frames at the next frame boundary. Any frame currently in reception or transmission will be completed normally without suspension. For transmission, any complete or partial frame in the tx cell FIFO will be transmitted. For receive, frames that are detected by the CPDMA after the suspend state is entered are ignored. No statistics will be kept for ignored frames. Emulation control is implemented for compatibility with other peripherals

Table 18-795 shows the operations of the emulation control input and register bits.

Table 18-795. Emulation Control Input

EMUSUSP	SOFT	FREE	Description
0	X	X	Normal Operation
1	0	0	Normal Operation
1	1	0	Emulation Suspend
1	X	1	Normal Operation

18.8.4.8.13 FIFO Loopback

FIFO loopback mode is entered when the FIFO_LOOPBACK bit in the CPSW_CONTROL register is set. FIFO loopback mode causes packets received on a port to be turned around and transmitted back on the same port. Port 0 receive is fixed on channel 0 in FIFO loopback mode. The RXSOFOVERRUN statistic is incremented for each packet sent in FIFO loopback mode. Packets sent in with errors are returned with errors (they are not dropped). FIFO loopback is intended as a simple mechanism for test purposes. FIFO loopback should be performed in fullduplex mode only.

18.8.4.8.14 Device Level Ring (DLR) Support

Device Level Ring (DLR) support is enabled by setting the DLR_EN bit in the CPSW_CONTROL register. When enabled, incoming received DLR packets are detected and sent to queue 3 (highest priority) of the egress port(s). If the host port is the egress port for a DLR packet then the packet is sent on the CPDMA Rx channel selected by the P0_DLR_CPDMA_CH field in the P0_CONTROL register. The supervisor node MAC address feature is supported with the dlr_unicast bit in the unicast address table entry. When set, the dlr_unicast bit causes a packet with the matching destination address to be flooded to the vlan_member_list minus the receive port and minus the host port (the port_number field in the unicast address table entry is a don't care). Matching dlr_unicast packets are flooded regardless of whether the packet is a DLR packet or not. The EN_P0_UNI_FLOOD bit in the ALE_CONTROL register has no effect on DLR unicast packets. Packets are determined to be DLR packets as shown below:

1. DLR is enabled (DLR_EN is set in the CPSW_CONTROL).
2. One of the sequences below are true.
    1. The first packet ltype matches CPSW_VLAN_LTYPE[15:0] VLAN_LTYPE1 and Px_CONTROL[20] Px_VLAN_LTYPE1_EN is set and the second packet ltype matches CPSW_DLR_LTYPE[15:0] DLR_LTYPE.
    2. The first packet ltype matches CPSW_VLAN_LTYPE[31:16] VLAN_LTYPE2 and Px_CONTROL[21] Px_VLAN_LTYPE2_EN is set and the second packet ltype matches CPSW_DLR_LTYPE[15:0] DLR_LTYPE.
    3. The first packet ltype matches CPSW_VLAN_LTYPE[15:0] VLAN_LTYPE1 and Px_CONTROL[20] Px_VLAN_LTYPE1_EN is set and the second packet ltype matches CPSW_VLAN_LTYPE[31:16] VLAN_LTYPE2 and Px_CONTROL[21] Px_VLAN_LTYPE2_EN is set and the third packet ltype

matches CPSW_DLR_LTYPE[15:0] DLR_LTYPE.

18.8.4.8.15 Energy Efficient Ethernet Support (802.3az)

NOTE: CM_GMAC_CLKSTCTRL[1:0]CLKTRCTRL bit field must not be programmed for SW_SLEEP or HW_AUTO.

Energy Efficient Ethernet (EEE) allows the PRCM to turn off the module clock during inactive periods as determined by network and host traffic. The module can then be awakened by host queued transmit packet(s) or by a port's external Ethernet PHY. EEE operations are configured as shown below:

1. The 10-bit EEE clock pre-scale value is written to the CPSW_EEE_PRESCALE register. The pre-scaler is used to clock all EEE-related counters
2. The port Idle to LPI count values (Px_IDLE2LPI) are written with the desired values
3. The port LPI to Wake count values (Px_LPI2WAKE) are written with the desired values
4. The EEE_EN bit is set in the switch CPSW_CONTROL register
5. Set the WR_CONTROL[8] SS_EEE_EN bit to enable energy efficient operations at subsystem level.

EEE operation can begin after configuration. The host allows (through PRCM) the CPSW_3G to enter a low power state by asserting the CLKSTOP_REQ signal. There are no requirements on host queues or traffic in order for the host to assert or de-assert CLKSTOP_REQ to the CPSW_3G.

Each ethernet port has a transmit and a receive LPI (low power indicate) state. The receive LPI state is entered when the port's corresponding PHY indicates the LPI state via the CPSW_3G GMII interface. The PHY indicates LPI by asserting MRXER with a MRXD[7:0] value of 0x01 while MRXDV is deasserted (inter-packet gap). The Ethernet transmit port indicates LPI after the Px_IDLE2LPI value has been counted (the transmit port has gone idle for the configured amount of time). If another packet is received for transmit during the count then the count is restarted. When the transmit port has been idle for the Idle to LPI time, the transmit port enters the LPI state and indicates LPI to the associated PHY. The LPI is indicated to the external PHY by an asserted MTXER with a MTXD[7:0] while MTXEN is deasserted (inter-packet gap). The CPDMA LPI state includes transmit and receive. The CPDMA LPI state is entered when the CPDMA transmit and receive have both been idle for the Idle to LPI time (P0_IDLE2LPI). The Idle to LPI time value for all ports must be large relative to the switch latency to ensure that the count is not able to complete between successive packets.

NOTE: The procedure above is described for the GMII interfaces at the CPSW_3G boundary. External PHY signaling has the following conditions:

- RGMII is a DDR interface. TXEN and TXER are the sampled values of TX_CTL at the rising and the falling TXC_CLK edges, respectively. RXDV and RXER are the sampled values of RX_CTL at the rising and the falling RXC_CLK edges, respectively

When all transmit and receive ports are in the LPI state (CPSW LPI state), the CLKSTOP_ACK signal is asserted, and the PRCM is allowed to stop the module clock. When CLKSTOP_ACK is asserted, the clock may be turned on and off as desired by the host. The host is allowed to restart the clock, perform slave read/write operations to the CPSW_3G memory address space, and then turn off the clock again while CLKSTOP_ACK is asserted.

The software can remove and disable from re-entering the CPSW LPI state by restarting the module clock and then de-asserting CLKSTOP_REQ. The host may queue transmit packets at any time including without regard to the CPSW_3G LPI state (the clock must be restarted in order to write the CPSW_3G slave address space as described above). Host writes to transmit head descriptor pointers will cause the CLKSTOP_WAKEUP signal to be asserted if the CPSW_3G is in the low power state (if CLKSTOP_ACK is asserted).

The external Ethernet PHY's can also wakeup the PRCM by removing the Ethernet receive LPI indication. If the module is in the CPSW Idle state with CLKSTOP_ACK asserted and the receive LPI indication is removed, the CLKSTOP_WAKEUP signal will be asynchronously asserted. On wakeup, the PRCM restarts the clock and de-assert the CLKSTOP_REQ signal. The CLKSTOP_WAKEUP signal will be synchronously deasserted with CLKSTOP_ACK. Upon the deassertion of CLKSTOP_REQ, the Ethernet ports will count the Px_LPI2WAKE time for each port at which time the port is available for transmit. Upon the de-assertion of CLKSTOP_REQ, the CPDMA transmit will count the P0_LPI2WAKE count at which time the CPDMA will begin to send any packets that the host has queued (switch ingress). The wait time on CPDMA transmit is included to preclude the host from filling up the Ethernet port transmit FIFO's while the Ethernet ports are in the LPI to wake time. There is no LPI to wake time on CPDMA receive (switch egress).

18.8.4.8.16 CPSW_3G Network Statistics

The CPSW_3G has a set of statistics that record events associated with frame traffic on selected switch ports. The statistics values are cleared to zero 38 clocks after the rising edge of GMAC_RST. When one or more port enable (Pn_STAT_EN) bits in the CPSW_STAT_PORT_EN register are set, all statistics registers are write to decrement. The value written will be subtracted from the register value with the result being stored in the register. If a value greater than the statistics value is written, then zero will be written to the register (writing 0xFFFF FFFF clears a statistics location). When all port enable bits are cleared to zero, all statistics registers are read/write (normal write direct, so writing 0x0000 0000 clears a statistics location). All write accesses must be 32-bit accesses. In the below statistics descriptions, "the port" refers to any enabled port (with a corresponding set Pn_STAT_EN bit).

The statistics interrupt (STAT_PEND) will be issued if enabled when any statistics value is greater than or equal to 0x8000 0000. The statistics interrupt is removed by writing to decrement any statistics value greater than 0x8000 0000. The statistics are mapped into internal memory space and are 32-bits wide. All statistics rollover from 0xFFFF FFFF to 0x0000 0000.

See Section 18.8.6.5, *STATS Registers* for description of every network statistic.

Table 18-796 and Table 18-797 summarize network statistics.

Table 18-796. Rx Statistics Summary

Rx Statistic	Frame /Oct	Rx/Rx +Tx	Frame Type					Frame Size (bytes)								Event				
			MAC control		Data[1]															
			Pause frame	Non-pause [3]	Multi cast	Broad cast	Uni cast	<64	64	65-127	128-255	256-511	512-1023	1024-rx_maxlen	>rx_maxlen	Flow Coll.[2]	CRC Error	Align/ Code	Overrun	Addr. Disc.
Good Rx Frames	F	R	(y|[4]	y|	y|	y|	y)	n	(y|	y|	y|	y|	y|	y)	n	-[5]	n	n	-	n
Broadcast Rx Frames	F	R	(%|[6]	%|	n	y)	n	n	(y|	y|	y|	y|	y|	y)	n	-	n	n	-	n
Multicast Rx Frames	F	R	(%|	%|	y)	n	n	n	(y|	y|	y|	y|	y|	y)	n	-	n	n	-	n
Pause Rx Frames	F	R	y	n	n	n	n	n	(y|	y|	y|	y|	y|	y)	n	-	n	n	-	-
Rx CRC Errors	F	R	(y|	y|	y|	y|	y)	n	(y|	y|	y|	y|	y|	y)	n	-	y	n	-	n
Rx Align/Code Errors	F	R	(y|	y|	y|	y|	y)	n	(y|	y|	y|	y|	y|	y)	n	-	-	y	-	n
Oversized Rx Frames	F	R	(y|	y|	y|	y|	y)	n	n	n	n	n	n	n	y	-	n	n	-	n
Rx Jabbers	F	R	(y|	y|	y|	y|	y)	n	n	n	n	n	n	n	y	-	(y|	y)	-	n
Undersized Rx Frames	F	R	n	n	(y|	y|	y)	y	n	n	n	n	n	n	n	-	n	n	-	n
Rx Fragments	F	R	n	n	(y|	y|	y)	y^[7]	n	n	n	n	n	n	n	-	(y|	y)	-	-
Rx Overruns[8]	F	R	(y|	y|	y|	y|	y)	(y|	y|	y|	y|	y|	y|	y|	y)	-	-	-	y	n
64octet Frames	F	R+T[9]	(y|	y|	y|	y|	y)	n	y	n	n	n	n	n	n	-	-	-	-	n
65-127octet Frames	F	R+T	(y|	y|	y|	y|	y)	n	n	y	n	n	n	n	n	-	-	-	-	n
128-255octet Frames	F	R+T	(y|	y|	y|	y|	y)	n	n	n	y	n	n	n	n	-	-	-	-	n
256-511octet Frames	F	R+T	(y|	y|	y|	y|	y)	n	n	n	n	y	n	n	n	-	-	-	-	n
512-1023octet Frames	F	R+T	(y|	y|	y|	y|	y)	n	n	n	n	n	y	n	n	-	-	-	-	n
1024-UPoctet Frames	F	R+T	(y|	y|	y|	y|	y)	n	n	n	n	n	n	y	n	-	-	-	-	n
Rx Octets	O	R	(y|	y|	y|	y|	y)	n	(y|	y|	y|	y|	y|	y)	n	-	n	n	-	n
Net Octets	O	R+T	(y|	y|	y|	y|	y)	(y|	y|	y|	y|	y|	y|	y|	y|	y)	-	-	-	-

(1) The multicast, broadcast and unicast columns in the table refer to non-MAC Control/non-pause frames (i.e. data frames).

(2) Flow coll. are half-duplex collisions forced by the MAC to achieve flow-control. A collision will be forced during the first 8 bytes so should not show in frame fragments. Some of the '-'s in this column might in reality be 'n's.

(3) The non-pause column refers to all MAC control frames (for example, frames with length/type=88.08) with opcodes other than 0x0001. The pauseframe column refers to MAC frames with the opcode=0x0001.

(4) "AND" is assumed horizontally across the table between all conditions which form the statistic (marked y or n) except where (y|y), meaning "OR" is indicated. Parentheses are significant.

(5) "-" indicates conditions which are ignored in the formations of the statistic.

(6) "%" If either a MAC control frame or pause frame has a multicast or broadcast destination address then the appropriate statistics will be updated.

(7) "y^" Frame fragments are not counted if less than 8 bytes.

(8) The rx_overruns stat is for RX_MOF_OVERRUNS and RX_SOF_OVERRUNS added together.

(9) Statistics marked "R+T" are formed by summing the Rx and Tx statistics, each of which is formed independently.

Table 18-797. Tx Statistics Summary

Tx Statistic[1]	Frame/Oct	Tx/Rx+Tx	Frame Type					Frame Size (bytes)							Event									
			MAC control[2]		Data											Collision Type									
			Pause-MAC	Any-CPU	Multicast	Broadcast	Unicast	64	65-127	128-255	256-511	512-1023	1024-1535	>1535	CRC Error	Flow[3]	1	2-15	16	Late	No Carrier	Queued	Deferred	Underrun
Good Tx Frames	F	T	(y|[4]	y|	y|	y|	y)	(y|	y|	y|	y|	y|	y|	y)	-[5]	-	-	-	n	n	n	-	-	n
Broadcast Tx Frames	F	T	n	(%|[6]	n	y)	n	(y|	y|	y|	y|	y|	y|	y)	-	-	-	-	n	n	n	-	-	n
Multicast Tx Frames	F	T	(y|	%|	y)	n	n	(y|	y|	y|	y|	y|	y|	y)	-	-	-	-	n	n	n	-	-	n
Pause Tx Frames	F	T	y	n	n	n	n	y	n	n	n	n	n	n	-	-	-	-	-	-	-	-	-	-
Collisions	F	T	n	(y|	y|	y|	y)	(y|	y|	y|	y|	y|	y|	y)	-	(+[7]	+	+	+	+)	n	-	-	-
Single Collision Tx Frames	F	T	n	(y|	y|	y|	y)	(y|	y|	y|	y|	y|	y|	y)	-	-	y	n	n	n	n	-	-	-
Multiple Collision Tx Frames	F	T	n	(y|	y|	y|	y)	(y|	y|	y|	y|	y|	y|	y)	-	-	n	y	n	n	n	-	-	-
Excessive Collisions	F	T	n	(y|	y|	y|	y)	(y|	y|	y|	y|	y|	y|	y)	-	-	n	n	y	n	n	-	-	-
Late Collisions	F	T	n	(y|	y|	y|	y)	n	(y|	y|	y|	y|	y|	y)	-	-	-	-	-	y	-	-	-	-
Deferred Tx Frames	F	T	n	(y|	y|	y|	y)	(y|	y|	y|	y|	y|	y|	y)	-	-	n	n	n	n	n	-	y	n
Carrier Sense Errors	F	T	(y|	y|	y|	y|	y)	(y|	y|	y|	y|	y|	y|	y)	-	-	-	-	-	-	y	-	-	-
64octet Frames	F	R+T[8]	(y|	y|	y|	y|	y)	y	n	n	n	n	n	n	-	-	-	-	n	n	n	-	-	-
65-127octet Frames	F	R+T	(y|	y|	y|	y|	y)	n	y	n	n	n	n	n	-	-	-	-	n	n	n	-	-	-
128-255octet Frames	F	R+T	(y|	y|	y|	y|	y)	n	n	y	n	n	n	n	-	-	-	-	n	n	n	-	-	-

(1) When the transmit Tx FIFO is drained due to the MAC being disabled or link being lost, then the frames being purged will not appear in the Tx statistics.
(2) Pause (MAC) frames are issued in the MAC as perfect (no CRC error) 64 byte frames in full duplex only, so they cannot collide.
(3) The flow collision type is for half-duplex collisions forced by the MAC to achieve flow control. Some of the '-'s in this column might in reality be 'n's. To prevent double-counting, Net Octets are unaffected by the jam sequence – the 'received' bytes, however, are counted. (See Table 18-796.)
(4) "AND" is assumed horizontally across the table between all conditions which form the statistic (marked y or n) except where (y|y), meaning "OR" is indicated. Parentheses are significant.
(5) "-" indicates conditions which are ignored in the formations of the statistic.
(6) "%" If a CPU sourced MAC control frame has a multicast or broadcast destination address then the appropriate statistics will be updated.
(7) "+" indicates collisions which are "summed" (i.e. every collision is counted in the Collisions statistic). Jam sequences used for halfduplex flow control are also counted.
(8) Statistics marked "R+T" are formed by summing the Rx and Tx statistics, each of which is formed independently.

Table 18-797. Tx Statistics Summary (continued)

Tx Statistic[1]	Frame/Oct	Tx/Rx+Tx	Frame Type					Frame Size (bytes)							Event										
			MAC control[2]		Data											Collision Type									
			Pause-MAC	Any-CPU	Multicast	Broadcast	Unicast	64	65-127	128-255	256-511	512-1023	1024-1535	>1535	CRC Error	Flow[3]	1	2-15	16	Late	No Carrier	Queued	Deferred	Underrun
256-511octet Frames	F	R+T	(y|	y|	y|	y|	y)	n	n	n	y	n	n	n	-	-	-	-	n	n	n	-	-	-
512-1023octet Frames	F	R+T	(y|	y|	y|	y|	y)	n	n	n	n	y	n	n	-	-	-	-	n	n	n	-	-	-
1024-UPoctet Frames	F	R+T	(y|	y|	y|	y|	y)	n	n	n	n	n	y	y	-	-	-	-	n	n	n	-	-	-
Tx Octets	O	T	(y|	y|	y|	y|	y)	(y|	y|	y|	y|	y|	y|	y)	-	-	-	-	n	n	n	-	-	n
Net Octets	O	R+T	(y|	y|	y|	y|	y)	(y|	y|	y|	y|	y|	y|	y)	-	-	$[9]	$	$	$	$	-	-	-

[9] "$" Every byte written on the wire during each retry attempt is also counted in addition to frames which experience no collisions or carrier loss.

18.8.4.9 Static Packet Filter (SPF)

18.8.4.9.1 SPF Overview

The Static Packet Filter (SPF) provides protection against some of the most common forms of Denial of Service (DoS) attacks that exploit the Layer 3 and Layer 4 functions of the network stack running on a system. The DoS attacks that are mitigated by the SPF module are:

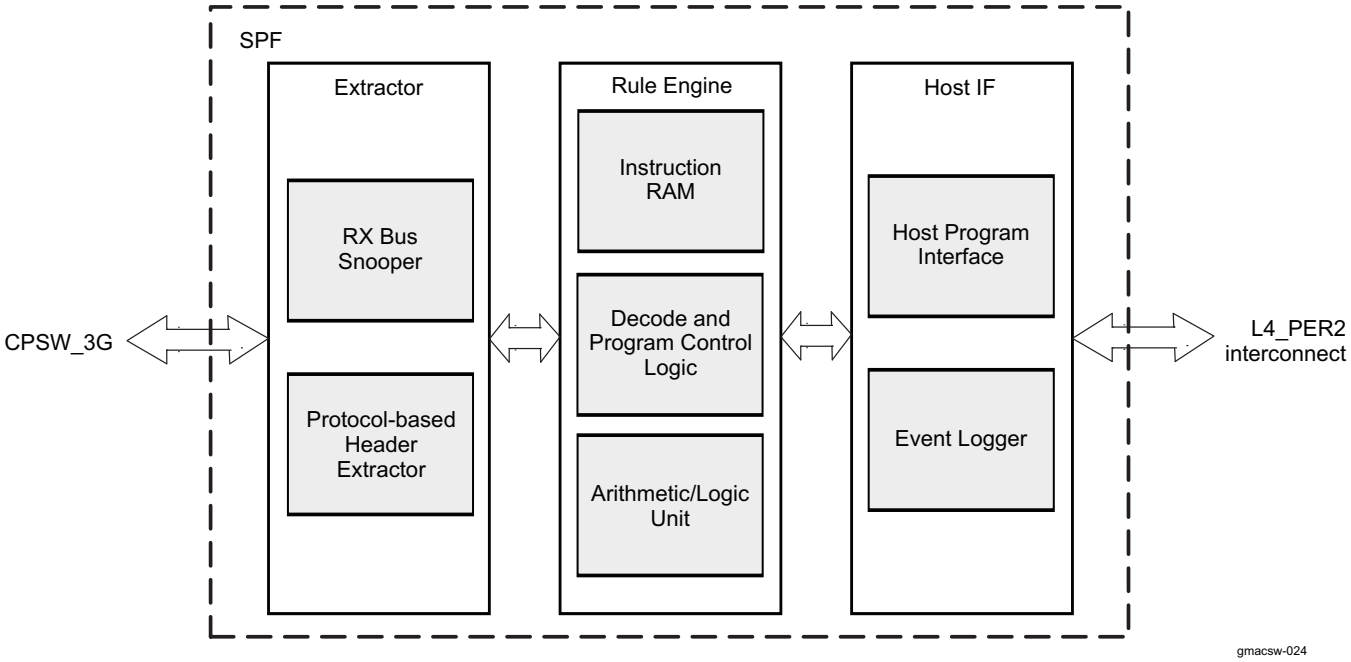
- ARP Flood
- Fraggle
- Illegal Fragment Offset
- Illegal TCP Options
- Jolt2
- Jolt
- Land
- Micro Fragment
- Null Scan
- Ping Flood
- Short ICMP Packet
- Smurf
- SSPing
- SYN Fragment
- TCP SYN Flood
- Xmas Scan

18.8.4.9.2 SPF Functional Description

18.8.4.9.2.1 SPF Block Diagram

Figure 18-164 shows the block diagram of the Static Packet Filter module. There are three main components of the static packet filter – packet header extractor to decode packet formats, rule engine to check for malformed protocol header fields and a host interface to provide control and configuration access to a host processor and to keep a log of filtered packets in system memory.

Figure 18-164. SPF Block Diagram



The packet parser interfaces to the three-port Ethernet switch (CPSW_3G). The CPSW_3G signals are snooped for receive traffic and the packet contents are provided to the packet header extractor. The parser can decode various network packet formats and determine the location of the header corresponding to each of the specified protocols and store the values in internal registers. The packet parser can decode VLAN, PPPoE, IP, TCP, UDP and ICMP frame formats and determine the location of each protocol header in a frame. The location of each of these headers is used by the rule engine for performing checks against preprogrammed conditions.

The rule engine monitors information in the header fields and upon detection of an abnormal combination of values in these packet header fields, issues an instruction to drop the packet to the external RFIFO interface. The rule engine can also monitor the receive rate of a particular class of packets and can limit the number of packets that actually pass through the system. The operation of rule engine is programmable and that gives flexibility to perform a range of different checks on the contents of packet to determine whether or not it should be accepted.

The event logger captures the activity in the packet filter. In addition, based upon the settings programmed by the host software, the event logger writes detailed information about any frames that have been dropped thus far. This information is written to a part of the system memory as configured by the host software.

18.8.4.9.2.2 Interrupts

The Static Packet Filter has one interrupt that is used to inform the host about excessive number of logged packet drops. The addresses of the instructions (or drop codes) in the instruction memory that cause packet to be dropped are associated with a threshold. When the threshold is met, the packet drop is logged. Each time a packet drop is logged, it is counted. When the number of logged records reaches the value specified by the SPF_INTCNT register, the host is interrupted. The threshold settings can be controlled by host software to limit the frequency of interrupts. Writing a zero to SPF_INTCNT disables the interrupt in this scenario. Whenever interrupt is enabled and is triggered, it can be cleared by writing one to either Raw register or Masked register.

Interrupts are controlled by the following registers:

1. Raw register SPF_INT_RAW: Holds the Raw state of interrupt, that is, without mask. The hardware interrupt is latched in this register and is only cleared when a one is written to either Raw register or Masked register. Writing a zero has no effect on this register.
2. Masked register SPF_INT_MASKED: This is the actual interrupt given to the system. It is the result of

bitwise AND operation of Raw and Mask registers. Once the interrupt is sensed by the system, this register should be cleared by writing one to this register or to Raw register during ISR execution. Writing zero has no effect on this register

3. Mask set register SPF_MASK_SET: Writing one to this register enables the interrupt. By default, the interrupt is disabled and needs to be enabled by writing a one to this register. Writing a zero has no effect on this register. Writing to this register also set clear register.
4. Mask clear register SPF_MASK_CLR: Writing one to this register disables interrupt. Writing a zero has no effect on this register.

18.8.4.9.2.3 Protocol Header Extractor

The extractor module has a protocol aware state machine. It decodes the header field in the current protocol header to determine the encapsulating protocol type and extracts various parameters used for software logging. The extractor module also provides the offset to octets in the packet, where Layer 3 and Layer 4 protocol headers start, to the rule engine. These protocol headers correspond to the location of IP and TCP/UDP/ICMP headers from the beginning of the packet.

Ethernet packets with VLAN, PPPoE, IP, IP Options, and ICMP/TCP/UDP protocol are supported by the extractor. When unknown protocols are detected, the extractor skips any additional packet processing. Some examples of how packet parsing is done by the Extractor are given below.

- Ethernet – VLAN - PPPoE – IP – IP Options –TCP/UDP/ICMP – Payload
- Ethernet – VLAN – PPPoE – IP – IP options – Unknown – Payload
- Ethernet – VLAN – PPPoE – IP – TCP/UDP/ICMP – Payload
- Ethernet – VLAN – PPPoE – IP – Unknown – Payload
- Ethernet – VLAN – PPPoE – Fragmented IP – Unknown – Payload
- Ethernet – VLAN – IP – IP Options –TCP/UDP/ICMP – Payload
- Ethernet – VLAN – IP – IP options – Unknown – Payload
- Ethernet – VLAN – IP – TCP/UDP/ICMP – Payload
- Ethernet – VLAN – IP – Unknown – Payload
- Ethernet – VLAN – Fragmented IP – Unknown – Payload
- Ethernet – IP – IP Options –TCP/UDP/ICMP – Payload
- Ethernet – IP – IP options – Unknown – Payload
- Ethernet – IP – TCP/UDP/ICMP – Payload
- Ethernet – IP – Unknown – Payload
- Ethernet – Fragmented IP – Unknown – Payload
- Ethernet – Unknown

The flow of Extractor state machine flow is as follows.

1. Wait for start of packet and proceed to step 2 if start of packet detected.
2. Read Ethernet header Length/Type field (at offset 13). Go to step 3 if next header is VLAN, to step 4 if it is PPP header, to step 5 if it is IP header and to step 8 otherwise.
3. Allow VLAN header to pass and read the Length/Type field. Go to step 4 if next header is PPP, to step 5 if it is IP header and to step 8 otherwise.
4. Read the protocol type in PPP header. Go to step 5 if protocol is IP and to step 8 for unknown protocol
5. Extract source IP address and destination IP address from the IP header. Determine if the packet has IP options or if the packet is fragmented. Go to step 6 if the packet has IP options and to step 8 if the packet is fragmented. If there are no options and the next protocol header is TCP/UDP, go to step 7. Otherwise, skip processing and go to step 8 because the packet has unrecognized protocol header
6. Skip IP options and determines the next protocol header. Go to step 7 if next header is TCP/UDP/ICMP. In case the packet contains an IP fragment or if the protocol in the next header is unknown, skip processing and go to step 8.
7. The packet has TCP/UDP/ICMP header. Extract TCP/UDP source and destination port numbers for logging or ICMP type and code fields and go to step 8.

8. Wait for end of frame and go to step 1 at end of frame.

In case the packet is aborted (indicated by address 0x10 on VBUSP with request and write ready asserted), the extractor state machine flushes its current state, goes back to the idle state and waits for next packet.

The Extractor provides information to rule engine about location of Layer 3 and Layer 4 protocols. When this information is provided to rule engine, the Base Register 1 and Base register 2 are loaded with Layer 3 start offset and Layer 4 start offset respectively. Until the extractor completes decoding of these protocol headers and sends the offset values to rule engine, the rule engine does not execute any instruction that operates on fields in these protocol headers.

Extractor module extracts IP protocol, Source IP address, Destination IP address, TCP/UDP source port, TCP/UDP destination port and ICMP type/code from incoming packet and provides this information to the Host logger module for software reporting. In case IP header or TCP/UDP header does not contain the extractable fields then the corresponding field is logged as zero.

In case packet has unknown Layer 4 protocol or IP packet is fragmented such that the packet does not contain TCP/UDP/ICMP header then TCP/UDP source/destination ports or ICMP type/code fields are logged as zeros.

18.8.4.9.2.4 Programmable Rule Engine

The rule engine is a micro-coded machine that is programmed by the host software. The rule engine is programmed to evaluate various expressions involving the header fields of different protocols that the incoming packet belongs to. These expressions are coded into the instructions that are stored in an internal RAM and are used to check whether a packet is mal-formed or is potentially a DoS attack packet. The rule engine executes the instructions for each incoming packet and makes a decision about accepting or rejecting the packet.

Since each packet can have multiple distinct protocol headers, the location of the headers can be different from packet to packet. The rule engine gets information about the location of the protocol headers from the packet header extractor. The packet header extractor can decode several different protocol types. The octet number at which a particular protocol header is stored is loaded into SPF Base Registers. An application may require SPF to filter packets based on protocols that are not decoded by the extractor. In such cases, SPF allows bypassing the header extractor and then the rule engine can be programmed to analyze packets and figure out the location of each header.

The Rule Engine instructions are programmed before the SPF module is enabled. The instructions cannot be overwritten while the rule engine is processing packets. To modify the contents of code RAM during operation, SPF must be disabled temporarily and then new instructions can be loaded.

Once SPF is enabled, the rule engine starts to fetch instructions one at a time. For each instruction, the operands are obtained from either the packet octets that are being received, from internal registers or from the immediate values inside the instruction itself. If the octet that is needed for execution has not yet been received, then the execution stalls until the required octet is received. In case the operand specified in the instruction refers to a packet octet that has already gone by and is not available in the packet buffer, then the execution stalls until the end of packet. An instruction is executed only when all the required operands are available. Based on the instruction execution results, the packet may immediately be dropped or the results of the evaluated expressions may be stored for future use. In addition, the current instruction can also cause the rule engine to skip a specified number of instructions (immediately following the current instruction) and resume from another location in the instruction RAM.

The rule engine operates on multiple operands and performs multiple tasks in each clock cycle. In each cycle, it can perform one arithmetic and/or logical operation on two pairs of operands. The operands are masked with a 32-bit mask that is generated from the information provided in the instruction. The mask allows for operations that involve variable sized operands. Each operation generates a 32-bit number and a flag bit. The 32-bit number is typically either the sum or the output of bit-wise logical operation. The flag is a single bit result of a comparison operation. The result of each operation can be saved in the internal registers if a specified condition is satisfied. Similarly, depending on the result of the operation, the rule

engine can jump to another location in the instruction memory. The program can instruct the rule engine to either perform two save operations, two conditional jumps or one jump and one save. The conditions must be mutually exclusive to prevent unspecified operation. In addition to providing an alternate path for execution, the instruction can cause the rule engine to accept or reject the current packet and exit from the program until the next packet is received.

When the execution of an instruction does not result in a decision to accept or reject the packet, the rule engine progresses to the next instruction. Instructions are executed until a final decision is made. If a packet is aborted in the middle at the network interface, the rule engine aborts execution and clears all base registers, program counter, octet counter and packet buffer. It is then ready to process next packet.

From a hardware perspective, the rule engine contains a buffer to store packet octets, several internal registers for temporary storage, instruction decoding logic and control circuitry. A description of each of the hardware resources in the rule engine follows.

18.8.4.9.2.4.1 Internal Registers

Several different types of registers are provided in the rule engine. The description of each of theese is provided below.

- Base Registers (B0-B4): There are four hardware base registers (B1-B4) that store the location of various protocol headers. The B0 register is not a hardware register and it always means a reference to the first octet in a packet. Each base register is eight bits wide and is used to reference octets in a packet. The base registers can point to any octet in the packet upto 255 octets. These registers are readable and writable by the rule engine. In addition, base registers B1 and B2 are loaded with data provided by the extractor when the rule engine is not running in extractor bypass mode (SPF_CONTROL[2] SPF_EXT_BYPASS). The rule engine can write to base registers irrespective of whether extractor bypass is enabled. When an operand needs to use contents of B1-B4 registers, the instruction only executes if the specified base register was loaded at least once after the beginning of current packet. If the specified base register was not loaded during the current packet, then the operand referenced by this base register cannot be extracted from the packet and will cause the rule engine to stall.
- Constant Registers (C0-C7): The rule engine can refer to any of eight 32-bit constant values. These are programmed by the host (SPF_CONSTj, j = 0 to 7) and the rule engine only has read access to these registers. Each of these registers can be changed at any time by the host software. However, changing the value of these registers is not recommended as unpredictable behavior may occur if contents are changed while the rule engine is executing instructions that use the C0-C7 values.
- General Purpose 32-bit registers (R0-R7): There are eight 32-bit registers that can be read/written by the rule engine. These are general purpose registers and can be used as temporary storage. In addition, when the rule engine is required to provide logging information, R4-R7 are used to store information that will be written to memory.
- Rate limit registers (L0-L3): Four 8-bit rate limit registers are used by rule engine to count specific types of packets that are to be rate limited. These registers are counters that are loaded with programmed threshold values (SPF_RATELIMi, i = 0 to 3) at the end of time interval determined by the clock pre-scaler (SPF_PRESCALE)
- General Purpose 1-bit registers (T0-T3): Four 1-bit registers are provided to store comparison results by the rule engine. These registers can be used to store 1-bit output from ALUs. In addition, the logical OR and logical AND of the flags can also be stored in T0-T3 registers. The 32-bit ALU results may also be stored in T0-T3 registers but only the LSB will be stored.

The rule engine instructions refer to these internal registers during execution. More details about instruction encoding are in subsequent sections.

18.8.4.9.2.4.2 Packet Buffer

When the rule engine is processing a packet, it stored a snapshot of latest 32 octets in an internal buffer. This buffer allows evaluation of expressions on packet octets even after the packet octets have gone by on the external receive VBUSP network interface.

At the start of packet, the first eight octets received are loaded into the first eight locations of the packet buffer. When the next eight octets arrive, these are loaded into the first eight locations and the existing octets are moved to the following eight locations. This process is continued until the packet buffer is full. When packet buffer is full and additional octets arrive, the oldest eight octets are shifted out and are no longer available to the rule engine. Thus, the packet buffer provides a snapshot of most recent octets received.

When the rule engine encounters instructions that reference packet octets which have not yet been received then the rule engine stalls until the required octets are available. Similarly, instructions which require octets that have already been shifted out of the packet buffer will cause rule engine to stall. And since these octets will never be available, the rule engine will not execute any further instructions for the current packet. It will return to the first instruction when the packet ends.

Figure 18-165. Packet Octets as Stored in the Packet Buffer

RFIFO VBUSP	63:56	55:48	47:40	39:32	31:24	23:16	15:8	7:0
Clock 0	8	7	6	5	4	3	2	1
Clock 8	16	15	14	13	12	11	10	9
	8	7	6	5	4	3	2	1
Clock 16	24	23	22	21	20	19	18	17
	16	15	14	13	12	11	10	9
	8	7	6	5	4	3	2	1
Clock 24	32	31	30	29	28	27	26	25
	24	23	22	21	20	19	18	17
	16	15	14	13	12	11	10	9
	8	7	6	5	4	3	2	1
Clock 40	48	47	46	45	44	43	42	41
	40	39	38	37	36	35	34	33
	32	31	30	29	28	27	26	25
	24	23	22	21	20	19	18	17

gmacsw-025

The contents of a packet, when used as one or more operands in an instruction, are always fetched from the packet buffer. The Base registers (B0-B4) in conjunction with an offset and bits are used to specify which octets are to be used as operands.

18.8.4.9.2.5 Intrusion Event Logger

The activity of the static packet filter is logged by the intrusion event logger. Based on the configuration, the event logger writes specific fields of the incoming packet that violate a particular rule to the system memory for software diagnostics.

The logging module allows controlling the frequency at which events are logged to the memory. A set of nine counters are provided and, of these, eight can be mapped to any address of the rule engine's instruction RAM using map registers. This mapping logically associates the log information with the instruction that caused the packet to be dropped. The 9th counter is not associated with any particular instruction and it can be used to count the packets that were dropped at instructions not mapped to any of the other eight counters.

For each of the nine counters, there is a register to store the minimum number of attacks that must be detected before any information is logged to memory. Only when this threshold is met and logging is enabled, the host interface module writes log data in system memory. The memory area used to log is specified by the host in the SPF_LOG_BEGIN address and SPF_LOG_END address registers.

The event logger continues to write data to memory until it runs out of space and the log overwrite is disabled. With each update of the log, the SPF_LOG_HWPTR is updated. This information can be used by the host to determine how much data has been logged. The software in turn keeps the SPF_LOG_SWPTR updated to inform the SPF about the next address from where information will be read by the host software. It is required that software program correct value of SPF_LOG_END address to enable hardware to determine roll-over location. The SPF considers all space before the SPF_LOG_SWPTR as available for logging. In case SPF_CONTROL[9] SPF_LOGOW_EN control bit is set, SPF ignores SPF_LOG_SWPTR and logs data irrespective of software read log status. Note that the memory space allocated for logging is a multiple of four 32-bit words and end address SPF_LOG_END should be loaded with byte address of the next byte following the last log entry. For example if 16 bytes space is allocated for SPF logging from address 0x1400_0000, then SPF_LOG_BEGIN should be 0x1400_0000 and SPF_LOG_END should be 0x1400_0010.

The host software must set SPF_CONTROL[8] SPF_LOG_EN bit to activate logging. The setting of SPF_LOG_EN bit has no effect on the threshold based counters and they are always active. In addition to the log counters, SPF has one master drop count register (SPF_DROPCNT) which tracks the total number of packets dropped thus far. This counter does not roll over and must be cleared for re-run by the host processor once it reaches the maximum value.

The format in which packet information is written in the memory by the event logger is shown in Table 18-798 and Table 18-799.

Table 18-798. Format for TCP/UDP Packets

Drop Code	-	-	Protocol
Source IP Address			
Destination IP Address			
Source TCP/UDP Port		Destination TCP/UDP Port	

Table 18-799. Format for ICMP Packets

Drop Code	-	-	Protocol
Source IP Address			
Destination IP Address			
Type	Code	Checksum	

Each entry logged to memory has a drop code associated with it. The drop code is the address in instruction memory that actually triggered the drop. Up to eight drop codes can be monitored in this manner. In addition to drop code, the protocol, IP addresses and source/destination ports associated with the dropped packet are recorded.

Log data can be supplied by either Extractor module or by Rule engine. The SPF_CONTROL[3] SPF_RULE_LOG bit must be set to use log data from Rule Engine and cleared to use data supplied by the Extractor.

When logging is done through the Rule Engine, contents of internal registers R4-R7 are written to memory. The format of packet information stored can be programmed in the rule engine except for the drop code field which is static and cannot be changed. The format in which rule engine information is written in the memory by the event logger is shown in Table 18-800.

Table 18-800. Rule Engine Format

Drop Code	Register 4[23:0]
Register 5[31:0]	
Register 6[31:0]	

Table 18-800. Rule Engine Format (continued)

Register 7[31:0]

The rule engine programming must ensure that these registers contain all required information that is to be recorded before the packet drop instruction is executed. As soon as the drop instruction is executed, the logging module starts to send data from the registers R4, R5, R6 and R7 to the system memory and is not possible to modify the contents of these registers.

18.8.4.9.2.6 Rate Limiter

The static packet filter provides a way to limit the rate of specific types of incoming packets. The SPF module provides four rate limit registers (L0-L3) that are used in conjunction with a clock prescaler (SPF_PRESCALE) and rule engine instructions. The clock prescaler is a clock divider and every count down to zero and the subsequent roll-over indicates end of a time interval. At every such event, each of the four rate limit registers is loaded with a preset value (SPF_RATELIMi, i= 0 to 3) that is programmable by the host processor. When packets are received, the rule engine can identify packets of particular types and execute a limit operation. The limit operation specifies a condition (described in Table 18-818) and a limit register. If the condition evaluates to true, the packet is dropped if the rate limit register is zero. If it is non-zero, then the rate limit register is decremented by one.

Thus, by using the prescaler and the limit registers, it is possible to control the rate of specific type of packets. By controlling the value of prescale counter (common to all rate limit registers) and the rate limit thresholds, the granularity of the rate can be modified. A lower value of prescale counter will cause frequent reloads of the rate limit registers and will allow rate control over small time intervals. Keeping the prescale counter at extremely small values may cause the rate limiter to be ineffective because the limit registers will be reloaded too frequently to count down to zero. Conversely, a higher value of prescale counter will cause less frequent reloads of the rate limit registers and the rate control will be over longer time intervals. The reload value of each 8-bit rate limit register and the prescale register are programmed by the host processor before SPF is enabled. A reload value of 0xFF can be used to disable any of the rate limiters at run-time without changing the firmware. Both the prescale and limit registers can be modified during run-time.

18.8.4.9.2.7 Rule Engine Instruction Set Architecture

18.8.4.9.2.7.1 Instruction Format

The design supports 64 deep instruction memory. Each of the instructions is 78 bit wide and is stored in a RAM internal to the SPF module. Each instruction can have up to four operands, two arithmetic/logical operations and two conditional jump/save actions based on the outcome of the operations.

The operand field is 52 bits wide and it can have up to four operands. The arithmetic and logical operation codes are four bits each. The operation codes are nine bits each.

Table 18-801. Instruction Format

Bits	Field Name	Description
77:26	OPERAND	Up to four operands are specified in this field. The source of the operands can be the packet that is being received at the network interface, one of the internal register values or an 8/16/32 bit operand specified within the instruction.
25:22	FUNCTION0	This field specifies a single or dual operand Arithmetic/Logic Function. This function is applied to one or both of first two operands specified in the instruction.
21:18	FUNCTION1	This field also specifies a single or dual operand Arithmetic/Logic Function. This function is applied to one or both of the third and fourth operands specified in the instruction.

Table 18-801. Instruction Format (continued)

Bits	Field Name	Description
17:9	OPERATION0	A Save/Jump/Limit/Nop operation is specified in this function. Save operation code includes the source and destination information. The save source is the output of functions and the destination is one of the internal registers where data should be saved. Jump operation code has information about a condition and a destination. The jump occurs when the condition, which is based on the result of function0 and function1, is true. The Jump destination controls the flow of instruction execution. Jump to location 1 results in the packet being dropped. Rule engine goes back to initial instruction and waits for next packet. The event logger writes information to memory. Jump to location zero results in the packet being accepted. The rule engine goes back to initial instruction and waits for next packet. Limit operation code has information about a condition and a rate limit register. The Limit operation either causes the packet to be dropped or results in the specified rate limit register to be decremented by one.
8:0	OPERATION1	This field specifies a second save/jump operation. The format for this field is same as for Operation0.

18.8.4.9.2.7.2 Operand Field

The operand field specifies up to four operands. Each operand is obtained from the incoming packet data, from an internal register or as an immediate value specified in the instruction itself. In addition, there is a bit mask associated with each operand. The mask values are encoded in the instruction itself and the mask for each operand is applied before it is used by the Arithmetic and Logic Unit.

The operands are input to the two 32-bit Arithmetic and Logic units. For all calculations, a mask is used with each operand. The mask is a 32-bit number that is generated from a 5-bit code provided in the instruction or from an immediate value in the instruction. The 5-bit mask code specifies how many bits (from the LSB side) will be input to the ALU. An immediate mask must be used when the mask is not a continuous sequence of 1's. Whenever an immediate mask value is specified, the mask value specified by the 'Bits' fields is ignored.

Each operand is at least 13-bits long and the encoding for each operand types is shown in Table 18-802 to Table 18-809.

Table 18-802. Packet Data Operand

Description	This type of operand is derived from the packet itself. The encoding specifies a number of bits (up to 32) to be extracted from a location in the packet indicated by the selected base register and offset. The format of 13-bit operand code is shown below. Internally, a 32-bit number is obtained from the packet. Then, a mask is created from the bits[4:0] field and bitwise ANDed with the 32- bits extracted from the packet.

12	11	10	9	8	7	6	5	4	3	2	1	0
Base[2:0]			Offset[4:0]					Bits[4:0]				

Bits	Field Name	Description
12:10	Base[2:0]	The Base field selects one of the Base registers B0, B1, B2, B3 or B4. The operand comprises of (Bits+1) bits extracted from the packet. The offset from where the operand octets are extracted is determined by the sum of the value of specified base register (Base 0 to Base 4) and the specified offset. Base 0 is start of packet.
9:5	Offset[4:0]	The value of the selected base register plus the specified offset is the octet location in the packet from where the specified number of bits will be picked.
4:0	Bits[4:0]	The number of bits to be used as operand is specified by Bits+1. When Bits[4:0] is zero, only one bit is used as operand. When Bits[4:0] is 31, then 32 bits are used as operand.

Table 18-803. Constant Operand

Description	One of the eight constants programmed by the host software can be used as operand. The bits field specifies the mask to be applied to the selected 32-bit constant.

12	11	10	9	8	7	6	5	4	3	2	1	0
1	0	1	0	0	C[2:0]			Bits[4:0]				

Bits	Field Name	Description
7:5	C[2:0]	This field selects one of the constants (C0-C7).
4:0	Bits[4:0]	The number of bits to be used as operand is specified by Bits+1. When Bits[4:0] is zero, the least significant bit is used as operand. When Bits[4:0] is 2, then two LSBs are used as operand and so on.

Table 18-804. 32-bit Register Operand

Description	One of eight 32-bit registers is used as operand and bits[4:0] field specifies the bitmask to be applied to that register.

12	11	10	9	8	7	6	5	4	3	2	1	0
1	0	1	0	1	R[2:0]			Bits[4:0]				

Bits	Field Name	Description
7:5	R[2:0]	This field selects one of the 32-bits registers (R0-R7).
4:0	Bits[4:0]	The number of bits to be used as operand is specified by Bits+1. When Bits[4:0] is zero, the least significant bit is used as operand. When Bits[4:0] is 2, then two LSBs are used as operand and so on.

Table 18-805. 1-bit Register Operand

Description	One of the four 1-bit registers is used as operand.

12	11	10	9	8	7	6	5	4	3	2	1	0
1	0	1	1	0	0	T[1:0]		0	0	0	0	0

Bits	Field Name	Description
6:5	T[1:0]	This field selects one of the four 1-bits registers (T0-T3).

Table 18-806. Base Register Operand

Description	One of the base registers is used. The 2-bit codes are 00, 01, 10 and 11 for B1 to B4 respectively.

12	11	10	9	8	7	6	5	4	3	2	1	0
1	0	1	1	1	1	B[1:0]		0	0	Bits[2:0]		

Bits	Field Name	Description
6:5	B[1:0]	This field selects one of the four base registers (B1-B4). Note that B0 is a virtual register which is always zero and specifies the start of packet.
2:0	Bits[2:0]	The number of bits to be used as operand is specified by Bits+1. When Bits[4:0] is zero, the least significant bit is used as operand. When Bits[4:0] is 2, then two LSBs are used as operand and so on.

Table 18-807. End of Packet Operand

Description	The End-of-Packet (EOP) operand can be used to detect the end of current packet. Use of this operand will stall the CPU until the end of packet is reached. The end of packet operand will always be equal to one when the CPU detects end of packet. It will be zero until packet data is being received and will also stall the CPU. The recommended use of this operand is in the last instruction where a check may be made for size of current packet. Note that this operand only tracks the end of a complete successfully received packet by the RFIFO. The EOP operand does not detect abort of a packet.

12	11	10	9	8	7	6	5	4	3	2	1	0
1	0	1	1	0	0	1	0	Bits[4:0]				

Bits	Field Name	Description
4:0	Bits[4:0]	The number of bits specifies the mask. However, these bits are inconsequential for EOP operand. Irrespective of the mask specified by bits[4:0], the EOP operand is always equal to 0x1 in the execution cycle of the instruction it is used.

Table 18-808. Octet Count Operand

Description	The Octet count operand is used to determine the number of bytes that have been received by rule engine.

12	11	10	9	8	7	6	5	4	3	2	1	0
1	0	1	0	0	1	1	1	Bits[4:0]				

Bits	Field Name	Description
4:0	Bits[4:0]	The number of bits specifies the mask that is applied to octet counter value. The recommend value for this field is 0x1F.

Table 18-809. Immediate Data Operands

Description	The immediate data operands can be bigger than 13 bits as shown in the codes below. Each of such operands is directly sent to the ALU without applying any addition bit mask. The immediate data operands can be 8, 16 or 32 bits wide.

12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	1	0	0	Data[7:0]							

20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	1	0	1	Data[15:0]															

36	35	34	33	32	31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	1	1	0	Data[31:0]																																	

Table 18-810. Immediate Operand Masks

Description	The operand fields can also be used to specify bit masks. These are used when only specific bits from an operand are to be extracted. There is provision for 8-bit and 16-bit masks that can be encoded into an instruction's operand field.

12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	0	0	Mask[7:0]							

20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
1	1	0	0	1	Mask[15:0]															

When the operand field is all-ones it does not specify any operand and is considered zero.

The tables below list the combinations in which operands may be programmed in an instruction. In this list, Pkt/Reg operand is an operand consisting of octets from the packet or the contents of one of the internal registers; mask specifies the value that is used as a mask instead of the mask generated from the Bits field in the instruction; immediate operand (immediate[7:0], immediate[15:0] and immediate[31:0]) is an 8/16/32 bit number that is used as an operand. The format of Pkt/Reg operand is as shown previously. The decoding of operands for each ALU is dependent upon the type of operation specified in the ALU and the format of the operand field. When there are four operands in the operand field, the first and second operands are used in ALU0 and the rest by ALU1. When ALU0 needs only one operand, then the remaining operands are fed to ALU1. The order of decoding operands is designed to first provide operands to ALU0 from the more significant side of operand field and then to ALU1. If the number of operands specified in instruction does not match with the number of operands required by the ALUs, unspecified behavior can occur. For an instruction to execute, both ALUs must have valid data. If any of the ALUs uses a packet data operand which is not yet available then the rule engine stalls until data is available. During the stall phase, the instruction is decoded every cycle and all required operands must be available simultaneously for the instruction to be executed In addition, if an instruction refers to packet octets that are no longer available, the rule engine stalls until the end of packet.

Table 18-811. Operand Field With Four Operands

Pkt / Reg / Imm[7:0] Operand[12:0]	Pkt / Reg / Imm [7:0] Operand[12:0]	Pkt / Reg / Imm [7:0] Operand[12:0]	Pkt / Reg / Imm [7:0] Operand[12:0]

Table 18-812. Operand Field With Three Operands

Pkt / Reg / Imm[7:0] Operand[12:0]	Pkt / Reg / Imm [7:0] Operand[12:0]	Pkt / Reg / Imm [7:0] Operand[12:0]	Reserved
Pkt/Reg Operand[12:0]	11000 Mask[7:0]	Pkt / Reg / Imm [7:0] Operand[12:0]	Pkt / Reg / Imm [7:0] Operand[12:0]
Pkt / Reg / Imm[7:0] Operand[12:0]	Pkt/Reg Operand[12:0]	11000 Mask[7:0]	Pkt / Reg / Imm [7:0] Operand[12:0]
Pkt / Reg / Imm[7:0] Operand[12:0]	Pkt / Reg / Imm [7:0] Operand[12:0]	Pkt/Reg Operand[12:0]	11000 Mask[7:0]
11101 Immediate[15:0] 11111		Pkt / Reg / Imm[7:0] Operand[12:0]	Pkt / Reg / Imm[7:0] Operand[12:0]
Pkt / Reg / Imm[7:0] Operand[12:0]	11101 Immediate[15:0] 11111		Pkt / Reg / Imm[7:0] Operand[12:0]

Table 18-813. Operand Field With Two Operands

Pkt / Reg / Imm[7:0] Operand[12:0]	Pkt / Reg / Imm [7:0] Operand[12:0]	Reserved	Reserved
Pkt / Reg / Imm[7:0] Operand[12:0]	Pkt/Reg Operand[12:0]	11000 Mask[7:0]	Reserved
Pkt/Reg Operand[12:0]	11000 Mask[7:0]	Pkt / Reg / Imm[7:0] Operand[12:0]	Reserved

Pkt/Reg Operand[12:0]	11000 Mask[7:0]	Pkt/Reg Operand[12:0]	11000 Mask[7:0]
11101 Immediate[15:0] 11111		Pkt/Reg Operand[12:0]	11000 Mask[7:0]
Pkt / Reg / Imm[7:0] Operand[12:0]	11101 Immediate[15:0] 11111		Reserved
Pkt/Reg Operand[12:0]	11000 Mask[7:0]	11101 Immediate[15:0] 11111	
11101 Immediate[15:0] 11111		11101 Immediate[15:0] 11111	
11101 Immediate[15:0] 11111		Pkt / Reg / Imm[7:0] Operand[12:0]	Reserved
Pkt/Reg Operand[12:0]	11001 Mask[15:0] 11111		Pkt / Reg / Imm[7:0] Operand[12:0]
Pkt / Reg / Imm[7:0] Operand[12:0]	Pkt/Reg Operand[12:0]	11001 Mask[15:0] 11111	
Pkt/Reg Operand[12:0]	1101 Mask[15:0] 01 Immediate[15:0]		
Pkt / Reg / Imm[7:0] Operand[12:0]	11110 Immediate[31:0] 11		
11110 Immediate[31:0] 11			Pkt / Reg / Imm[7:0] Operand[12:0]

Table 18-814. Operand Field With One Operand

Pkt / Reg / Imm[7:0] Operand[12:0]	Reserved	Reserved	Reserved
Pkt/Reg Operand[12:0]	11000 Mask[7:0]	Reserved	Reserved
Pkt/Reg Operand[12:0]	11001 Mask[15:0] 11111		Reserved
11101 Immediate[15:0] 11111		Reserved	Reserved
11110 Immediate[31:0] 11			Reserved

The formats in which operands are specified in the operand field have certain restrictions:

- a mask cannot follow an immediate value
- a mask cannot follow a previous mask
- a reserved field should have all following fields reserved as well
- the first field cannot be a mask

18.8.4.9.2.7.3 Arithmetic/Logical Function Field

There can be two codes for arithmetic and logical functions in each instruction. Each of the operations can involve one or two operands.

The location of each function field in an instruction is listed in Table 18-815.

Table 18-815. Arithmetic/Logical Instruction Field

	Bits	Description
	23:20	Single or Dual operand Arithmetic/Logic Function (function 0)
	19:16	Single or Dual operand Arithmetic/Logic Function (function 1)

The bit codes for each arithmetic/logic function that can be performed on the operands of an instruction are listed in Table 18-816.

Table 18-816. Arithmetic/Logical Operation Codes

Code	Description	Operands Required	32-bit result	1-bit flag result
0x0	Less Than	2	0	Set if true
0x1	Less Than Equal To	2	0	Set if true
0x2	Greater Than	2	0	Set if true
0x3	Greater Than Equal To	2	0	Set if true
0x4	Equal to Zero	1	0	Set if true
0x5	Not Equal to Zero	1	0	Set if true
0x6	Equal to One	1	0	Set if true
0x7	Not Equal to One	1	0	Set if true
0x8	Equal	2	0	Set if true
0x9	Not Equal	2	0	Set if true
0xA	Add	2	Sum	Set if overflow
0xB	Subtract	2	Difference	Set if output is negative
0xC	Bitwise AND	2	Bitwise AND	0
0xD	Bitwise OR	2	Bitwise OR	0
0xE	Result is same as Operand	1	Same as operand	0
0xF	No Function (NOF)	0	0	0

The ALU0 and ALU1 are assigned function0 and function1 respectively. In case only one function is to be performed in an instruction, function0 field should be used to do that. An instruction with no function for ALU0 and a valid function for ALU1 will result in unspecified behavior.

The output of each function is in the form of a 32-bit number and a single bit value. The single bit value is either the carry bit from an add/subtract operation or the Boolean result from a comparison function. The single bit output from each function is referred as the flag0 or flag1 result. The 32-bit number is the outcome of arithmetic or a bitwise logical operation and it is referred to as the word0 or word1 result.

18.8.4.9.2.7.4 Operation Field

The operation field specifies a limit, save or a jump operation. All of these operations involve two arguments that are specified within the instruction.

In case of limit operation, one of the arguments is a condition and the other argument is a limit register. In case of save operation, one of the arguments is source data and the other argument is a destination where the data will be saved. In jump operation, the first argument is a condition that must be true for the jump to occur and the second argument is a destination to which the program control will be moved to. The destination argument in a jump operation also specifies specific destinations that lead to the current packet being immediately rejected or accepted.

The location of operation field in an instruction is shown in Table 18-817.

Table 18-817. Operation Fields

Bits	Description
17:9	Limit/Save/Jump Operation (operation 0) For limit operation, [17:16] is the operation field, [15:13] is the condition field and [12:9] is a limit register. For save operation, [17:16] is the operation field, [15:13] is the source field, and [12:9] is the destination field. For jump operation, [17:16] is the operation field, [15:13] is the condition field, and [12:9] is the destination field
8:0	Limit/Save/Jump Operation (operation 1) For limit operation, [8:7] is the operation field, [6:4] is the condition field and [3:0] is a limit register. For save operation, [8:7] is the operation field, [6:4] is the source field, and [3:0] is the destination field. For jump operation, [8:7] is the operation field, [6:4] is the condition field, and [3:0] is the destination field.

Table 18-818. Limit Operation

Description	In a limit operation, the condition argument specifies a condition that must be true for the specified limit register to be decremented if non-zero or the packet to be dropped if the limit register is zero.

8	7	6	5	4	3	2	1	0
0	0	Condition Code			Limit Register			

Bits	Field Name	Description
6:4	Condition Code	It is a 3-bit code that specifies the condition that must be true for the operation to be executed. 0x0: Single bit flag (flag0 from function 0) 0x1: Single bit flag (flag1 from function 1) 0x2: Logical OR of flags (flag0 | flag1) 0x3: Logical NOR of flags (!(flag0 | flag1)) 0x4: Logical AND of flags (flag0 & flag1) 0x5: Logical NAND of flags (!(flag0 & flag1)) 0x6: XOR of flags (flag0 ^ flag1) 0x7: Unconditional Decrement/Limit
3:0	Limit Register	It is a 4-bit code that specifies the limit register that is to be decremented or checked for zero value. 0x0: Rate limit register L0 0x1: Rate limit register L1 0x2: Rate limit register L2 0x3: Rate limit register L3 0x4-0xF: Reserved

Table 18-819. Save Operation

Description	The arguments for a save operation include a code for source data and a code for the destination. The source is either a single bit data or a 32-bit word. The single bit data is either a flag from the functions or a logical OR/AND of the flags. The 32-bit word is the result of the ALU arithmetic/logical functions.

8	7	6	5	4	3	2	1	0
0	1	Source Data Code			Destination Register Code			

Bits	Field Name	Description
6:4	Source Data Code	It is a 3-bit code that specifies the source data which is to be save in another register 0x0: Single bit flag (flag0 from function 0) 0x1: Single bit flag (flag1 from function 1) 0x2: Logical OR of flags (flag0 | flag1) 0x3: Logical AND of flags (flag0 & flag1) 0x4: Output of function (word0) 0x5: Output of function (word1) 0x6-0x7: Reserved
3:0	Destination Register Code	It is a 4-bit code that specified the register in which source data is to be stored 0x0-0x3: Single Bit register (T0-T3) 0x4-0x7: Base Register (B1-B3) 0x8-0xF: 32-bit Register (R0-R7)

For some instructions, the rule engine extracts the operand from the current packet. To determine the packet bits to be used as operands, it refers to one or more base registers. Whenever an instruction causes a change in value of any of the base registers, it cause the rule engine to wait for one clock cycle before executing the next instruction. The rule engine is a pipelined processor and change in the value of a base register causes the prefetched operand values to become stale and these need to be fetched again. The clock cycle inserted allows rule-engine to fetch the data again before the instruction using this data is executed. Only the instructions that use packet data will be delayed by a clock cycle if the previous instruction caused a change in the base register that is referenced in the current instruction.

Table 18-820. Jump Operation

Description	For jump operation, one of the operands is a condition that must evaluate to true for the jump operation to occur. The other operand is the destination to which the program counter will move to if the jump operation is executed. The condition in a jump operation is either one of the flags from the output of the arithmetic/logic unit or a logical AND/OR of the flags.

8	7	6	5	4	3	2	1	0
1	0	Condition Code			Destination Code			

Bits	Field Name	Description
6:4	Condition Code	Condition Code 0x0: Single bit flag (flag0 from function 0) 0x1: Single bit flag (flag1 from function 1) 0x2: Logical OR of flags (flag0 | flag1) 0x3: Logical NOR of flags (!(flag0 | flag1)) 0x4: Logical AND of flags (flag0 & flag1) 0x5: Logical NAND of flags (!(flag0 & flag1)) 0x6: XOR of flags (flag0 ^ flag1) 0x7: Unconditional Jump
3:0	Destination Code	Destination Code 0x0: Accept packet, return to instruction zero and wait for the next packet. 0x1: Reject packet and return to instruction zero. 0x2-0xF: Go to instruction at offset +2 ...+15

The jump operation results in a change of program execution flow. The jump destination determines the next instruction that will be executed by the rule engine. The destination is only specified as a positive offset to the current value of the program counter. The program counter can result in the rule engine skipping a given number of instructions instead of executing the next instruction. In addition, the destination field can instruct the rule engine to immediately reject or accept the current packet without any further checks. The rule engine then goes back to the first instruction and waits for a new packet to arrive before it executes any further instructions.

The rule engine is a pipelined processor. When it executes a jump operation, there is a delay of one clock cycle before the next instruction is executed.

The operations specified in the two operation fields are independent of each other. However, it is possible that conflicting operations are specified in the operation fields of an instruction. When conflicting operations are specified and conditions of both operations evaluate to true; then the execution flow is determined as described below:

1. If one of the ALUs has a limit/jump operation that will cause the packet to be dropped, then the packet will be dropped irrespective of what the other ALU indicates.
2. If case 1 above is false and one of the ALUs has a jump operation that will cause the packet to be accepted, then the packet will be accepted and the rule engine will return to idle state.
3. If case 1 and 2 above are false and one of the ALUs has a jump to offset with the respective condition true, the jump will be executed. If there is conflicting jump to offset information in the two ALUs, ALU0 is given priority.

Table 18-821. No Operation

Description	If there is no Save, Jump or Limit operation specified in the operation field, it is interpreted as a no operation. There is no change in execution flow, no change in any register values or rate limit registers when a no operation is encountered.

8	7	6	5	4	3	2	1	0
1	1	Reserved			Reserved			

It is expected that the rule engine will be programmed such that a decision to accept or discard the current packet is made before all instructions have executed. If the rule engine reaches the last instruction and it does not result in a decision to accept/reject the packet, then the execution flow does not stop and all instructions from the first instruction are executed again until the end of packet is reached.

18.8.4.9.3 Programming Guide

18.8.4.9.3.1 Initialization Routine

The packet filter must be initialized in order for it to operate as specified. A typical order in which the module can be initialized is shown below:

1. Initialize the firmware by programming the internal memory and verify it (SPF_INSTR_W2, SPF_INSTR_W1, SPF_INSTR_W0, SPF_INSTR_CTL)
2. Allocate memory for logs written by SPF and initialized the log space parameters (SPF_LOG_BEGIN, SPF_LOG_END, SPF_SW_PTR)
3. Program the log map registers to associate drop codes with the log thresholds. (SPF_LOG_MAP0, SPF_LOG_MAP1)
4. Program log thresholds (SPF_LOG_THRESHk, where k = 0 to 8)
5. Program constant registers if required by the firmware (SPF_CONSTj, where j = 0 to 7)
6. Program clock prescale counters and rate limit registers if required by the firmware (SPF_PRESCALE, SPF_RATELIMi , where i = 0 to 3)
7. Program the interrupt frequency control register and interrupt mask register to setup interrupts (SPF_INTCNT, SPF_MASK_SET)
8. Program the control registers to enable SPF filter and logger (SPF_CONTROL).

18.8.4.9.3.2 Interrupt Service Routine

The interrupt service routine in SPF should be designed to process the data logs generated by SPF. It should trigger a higher layer application software that can analyze the data logs and determine if any remedial measures are required based on the information in the logs. At the minimum, the interrupt service routine must reprogram the software pointer for the logging machine to be able to continue logging.

The typical tasks performed following an interrupt are listed in the pseudo code below.

```
SPF_ISR {
read log hardware pointer
read log software pointer
determine how many log entries are to be read
read each log entry starting with the software pointer
update software pointer to reflect the next unread entry
clear interrupt
}
```

18.8.4.9.3.3 Rule Engine Example Program

Here is an example of how the rule engine can be programmed to detect packets that resemble Denial of Service traffic. The pseudo code is shown below.

```
IDLE:
jump to ICMP if start_of_packet
```

```
ICMP:
if (protocol==ICMP)
limit_ICMP
if (fragmented packet)
drop and jump to IDLE
accept and jump to IDLE
else
jump to IP
IP:
if (source_ip==dest_ip)
drop and jump to IDLE
if (fragmented and (fragment_offset+ip_size)>2^16)
drop and jump to IDLE
accept the packet and jump to IDLE
```

18.8.4.10 Common Platform Time Sync (CPTS)

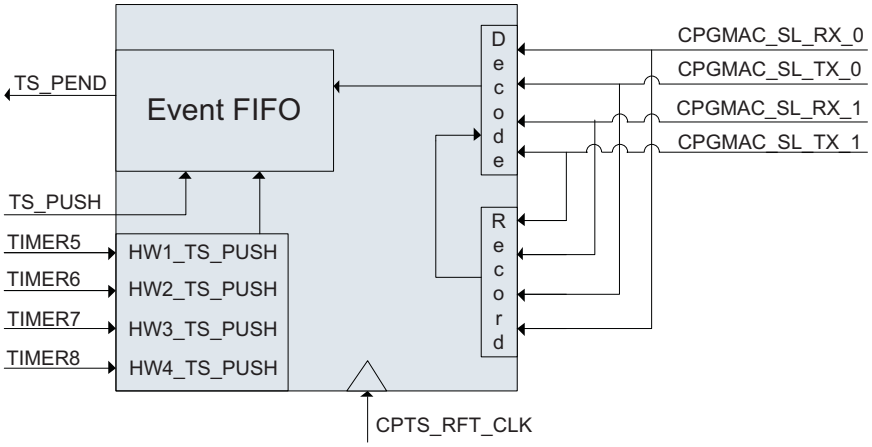
The Common Platform Time Sync (CPTS) module is used to facilitate host control of time sync operations. It enables compliance with the IEEE 1588-2008 standard for a precision clock synchronization protocol.

18.8.4.10.1 CPTS Architecture

Figure 18-166 shows the architecture of the CPTS module inside the GMAC_SW Ethernet Subsystem. Time stamp values for every packet transmitted or received on either port of the GMAC_SW are recorded. At the same time, each packet is decoded to determine if it is a valid time sync event. If so, an event is loaded into the Event FIFO for processing containing the recorded time stamp value when the packet was transmitted or received.

In addition, both hardware (HWx_TS_PUSH) and software (TS_PUSH) can be used to read the current time stamp value though the Event FIFO

Figure 18-166. CPTS Block Diagram



The reference clock used for the time stamp (CPTS_RFT_CLK) can be derived from several sources. See Chapter 3, *PRCM* for more details.

18.8.4.10.2 CPTS Initialization

The CPTS module should be configured as:

1. Reset the CPTS module.
2. Clear the CPTS_EN bit in the CPTS_CONTROL register.
3. Write the CLKSEL_RFT value in the CM_GMAC_GMAC_CLKCTRL register in the PRCM with the desired reference clock selection.
4. Set the CPTS_EN bit in the CPTS_CONTROL register.

5. If using interrupts and not polling, enable the interrupt by setting the TS_PEND_EN bit in the CPTS_INT_ENABLE register.

18.8.4.10.3 Time Stamp Value

The time stamp value is a 32-bit value that increments on each CPTS_RFT_CLK rising edge when CPTS_EN is set to 1. When CPTS_EN is cleared to 0, the time stamp value is reset to 0.

If more than 32-bits of time stamp are required by the application, the host software must maintain the necessary number of upper bits. The upper time stamp value should be incremented by the host when the rollover event is detected.

For test purposes, the time stamp can be written via the time stamp load function (CPTS_TS_LOAD_VAL and CPTS_TS_LOAD_EN registers).

18.8.4.10.4 Event FIFO

All time sync events are push onto the Event FIFO. There are 16 locations in the event FIFO with no overrun indication supported. Software must service the event FIFO in a timely manner to prevent FIFO overrun.

18.8.4.10.5 Time Sync Events

Time Sync events are 64-bit values that are pushed onto the event FIFO and read in two 32-bit reads. Two 32-bit registers, CPTS_EVENT_HIGH and CPTS_EVENT_LOW hold the data of a time sync event. There are five types of sync events:

- Time stamp push event
- Time stamp counter rollover event
- Time stamp counter half-rollover event
- Ethernet receive event
- Ethernet transmit event

18.8.4.10.5.1 Time Stamp Push Event

Software can obtain the current time stamp value (at the time of the write) by initiating a time stamp push event. The push event is initiated by setting the TS_PUSH bit of the CPTS_TS_PUSH register. The time stamp value is returned in the event, along with a time stamp push event code. Software should not push a second time stamp event on to the FIFO until the first time stamp value has been read from the event FIFO.

18.8.4.10.5.2 Time Stamp Counter Rollover Event

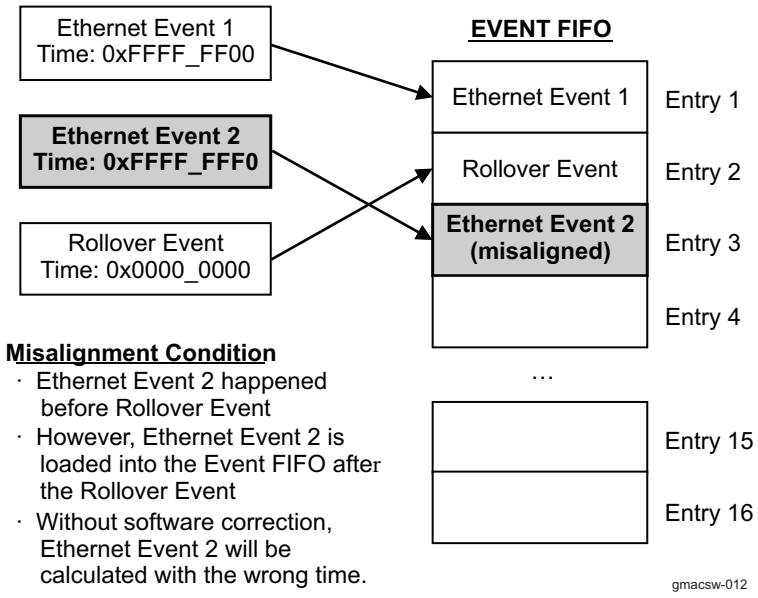
The CPTS module contains a 32-bit time stamp value. The counter upper bits are maintained by host software. The rollover event indicates to software that the time stamp counter has rolled over from FFFF FFFFh to 0000 0000h, and the software maintained upper count value should be incremented.

18.8.4.10.5.3 Time Stamp Counter Half-rollover Event

The CPTS includes a time stamp counter half-rollover event. The half-rollover event indicates to software that the time stamp value has incremented from 7FFF FFFFh to 8000 0000h. The half-rollover event is included to enable software to correct a misaligned event condition. The half-rollover event is included to enable software to determine the correct time for each event that contains a valid time stamp value, such as an Ethernet event. If an Ethernet event occurs around a counter rollover (full rollover), the rollover event could possibly be loaded into the event FIFO before the Ethernet event, even though the Ethernet event time was actually taken before the rollover. Figure 18-167 shows a misalignment condition.

Host software must detect and correct for misaligned event conditions. For every event after a rollover and before a half-rollover, software must examine the time stamp most significant bit. If bit 31 of the time stamp value is low (0000 0000h through 7FFF FFFFh), then the event time stamp was taken after the rollover and no correction is required. If the value is high (8000 0000h through FFFF FFFFh), the time stamp value was taken before the rollover and a misalignment is detected. The misaligned case indicates to software that it must subtract one from the upper count value stored in software to calculate the correct time for the misaligned event. The misaligned event occurs only on the rollover boundary and not on the half-rollover boundary. Software only needs to check for misalignment from a rollover event to a half-rollover event.

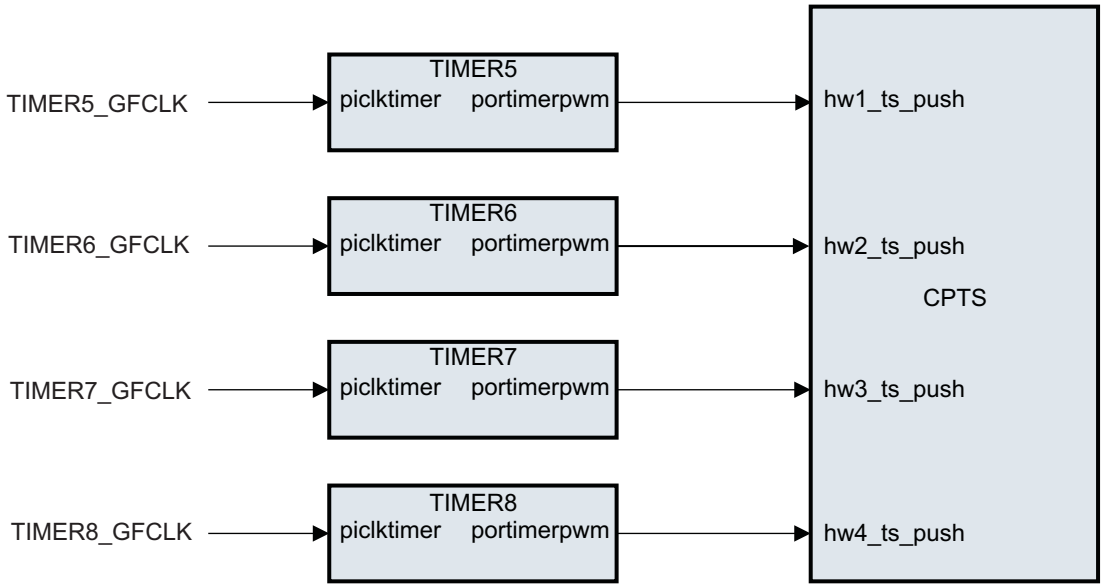
Figure 18-167. Event FIFO Misalignment Condition



18.8.4.10.5.4 Hardware Time Stamp Push Event

There are four hardware time stamp inputs (HW1/4_TS_PUSH) that can cause hardware time stamp push events to be loaded into the Event FIFO. Each hardware time stamp input is internally connected to the PORTIMERPWM output of each timer as shown in Figure 18-168.

Figure 18-168. HW1/4_TSP_PUSH Connection



The event is loaded into the event FIFO on the rising edge of the timer, and the PORT_NUMBER field in the CPTS_EVENT_HIGH register indicates the hardware time stamp input that caused the event.

Each hardware time stamp input must be asserted for at least 10 periods of the selected RCLK clock. Each input can be enabled or disabled by setting the respective bits in the CPTS_CONTROL register.

Hardware time stamps are intended to be an extremely low frequency signals, such that the event FIFO does not overrun. Software must keep up with the event FIFO and ensure that there is no overrun, or events will be lost.

18.8.4.10.5.5 Ethernet Port Events

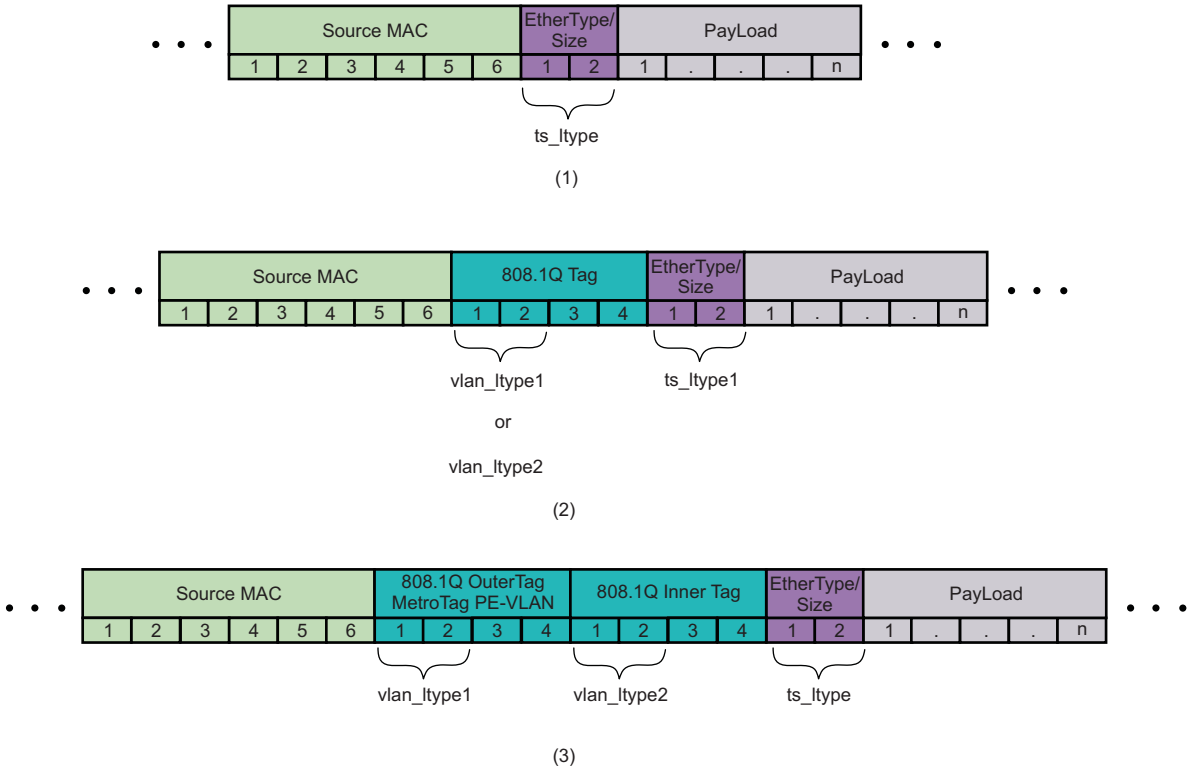
Packets transmitted or received on each Ethernet port can generate Ethernet Transmit Events or Ethernet Receive Events, respectively. The CPTS hardware will decode each packet to determine if it is a valid CPTS time sync event.

According to the IEEE 802.3 Ethernet standard, each Ethernet frame contains a 2-octet EtherType field to indicate which protocol is encapsulated in the PayLoad field, as shown in Figure 18-169. For standard time sync packets, this will contain the EtherType for the Precision Time Protocol (IEEE 1588), which is defined as 0x88F7. The CPTS hardware will compare this field to the TS_LTYPE1 field or the TS_LTYPE2 field (depending on which enable bit was set) in the CPSW_TS_LTYPE register, which should also be programmed to 88F7h.

When a virtual LAN is used, an additional 4-octet 802.1Q tag is inserted in the Ethernet frame before the EtherType field, as shown in Figure 18-169. To indicate to the CPTS hardware that a virtual LAN is in use, the Pn_TS_VLAN_LTYPE1_EN (or Pn_TS_VLAN_LTYPE2_EN) enable bit must be set in the Pn_CONTROL register. The EtherType for the 802.1Q tag is defined as 0x8100, and the CPTS hardware will compare this value to the VLAN_LTYPE1 (or VLAN_LTYPE2 depending on which enable bit was set) field in the CPSW_VLAN_LTYPE register, which should also be programmed to 0x8100.

When two stacked VLANs are used, two additional 4-octet 801.Q tags are inserted in the Ethernet frame before the EtherType field, as shown in Figure 18-169. In this case, both VLAN_LTYPE1 and VLAN_LTYPE2 must be enabled. The outer tag must match the value of the VLAN_LTYPE1 field, and the inner tag must match the value of the VLAN_LTYPE2 field.

Figure 18-169. Partial Ethernet-II Frames Showing Register Mapping of EtherTypes for a Simple Frame (1), a Single 1Q Tag Added (2), and Two 1Q Tags Added (3)



To enable transmit/receive event packets on a given Ethernet port, perform the following steps, where n is the port number:

1. Set the [1] Pn_TS_TX_EN or the [0] Pn_TS_RX_EN bit in the Pn_CONTROL register to enable transmit/receive event packets
2. Configure the register fields as follows:
    1. Set the Pn_CONTROL[2] Pn_TS_LTYPE1_EN or [3] Pn_TS_LTYPE2_EN bit to 1
    2. Set the CPSW_TS_LTYPE[15:0] TS_LTYPE1 or [31:16] TS_LTYPE2 (depending on which enable bit was set in 2(a)) field to 88F7h, which corresponds to the Precision Time Protocol (IEEE 1588) EtherType.
    3. Set the Pn_TS_SEQ_MTYPE[21:16] Pn_TS_SEQ_ID_OFFSET field to 1Eh, which is the sequence ID offset in the common message header given in the IEEE 1588 specification.
3. To enable support for VLAN tagging (IEEE 802.1Q):
    1. Set the [20] Pn_TS_VLAN_LTYPE1_EN or [21] Pn_TS_VLAN_LTYPE2_EN bit in the Pn_CONTROL register.
    2. Set the CPSW_VLAN_LTYPE[15:0] VLAN_LTYPE1 or [31:16] VLAN_LTYPE2 field (matching what was used in step 3a) to 8100h, which corresponds to the VLAN-tagged frame (IEEE 802.1Q) EtherType.
4. To enable support for up to two stacked VLANs (IEEE 802.1ad):
    1. Set the Pn_TS_VLAN_LTYPE1_EN and Pn_TS_VLAN_LTYPE2_EN bits in the Pn_CONTROL register.
    2. Set the VLAN_LTYPE1 field in the CPSW_VLAN_LTYPE register to match the EtherType of the outer tag.
    3. Set the VLAN_LTYPE2 field in the CPSW_VLAN_LTYPE register to 8100h, which corresponds to the VLAN-tagged frame (IEEE 802.1Q) EtherType of the inner tag.
5. Set the Pn_TS_SEQ_MTYPE[15:0] Pn_TS_MSG_TYPE_EN field to choose which types of time stamp messages will push events onto the Event FIFO. Table 9-20 lists the message types defined in the

IEEE 1588-2008 specification. Table 18-822 lists the message types defined in the IEEE 1588-2008 specification.

Table 18-822. Values of Message Type Field

Message Type	Value (hex)
Sync	0
Delay_Req	1
Pdelay_Req	2
Pdelay_Resp	3
Reserved	4-7
Follow_Up	8
Delay_Resp	9
Pdelay_Resp_Follow_Up	A
Announce	B
Signaling	C
Management	D
Reserved	E-F

Once a transmitted or received packet is determined to be a valid time sync packet, the Ethernet Transmit Event or Ethernet Receive Event is loaded onto the Event FIFO. The CPTS_EVENT_HIGH register contains the Message Type and Sequence ID values from the original time sync packet. The CPTS_EVENT_LOW register contains the time stamp value when the packet arrived at the corresponding port.

18.8.4.10.6 CPTS Interrupt Handling

When an event is push onto the Event FIFO, an interrupt can be generated to indicate to software that a time sync event occurred. The following steps should be taken to process time sync events using interrupts:

1. Enable the TS_PEND interrupt by setting the TS_PEND_EN bit of the CPTS_INT_ENABLE register.
2. Upon interrupt, read the CPTS_EVENT_LOW and CPTS_EVENT_HIGH registers values.
3. Set the EVENT_POP field (bit 0) of the CPTS_EVENT_POP register to pop the previously read value off of the event FIFO.
4. Process the interrupt as required by the application software.

Software has the option of processing more than a single event from the event FIFO in the interrupt service routine in the following way:

1. Enable the TS_PEND interrupt by setting the TS_PEND_EN bit of the CPTS_INT_ENABLE register.
2. Upon interrupt, read the CPTS_EVENT_LOW and CPTS_EVENT_HIGH registers values.
3. Set the EVENT_POP bit of the CPTS_EVENT_POP register to pop the previously read value off of the event FIFO.
4. Wait for an amount of time greater than four CPTS_RFT_CLK periods plus four MAIN_CLK periods.
5. Read the TS_PEND_RAW bit in the CPTS_INTSTAT_RAW register to determine if another valid event is in the event FIFO. If it is asserted, go to step 2; otherwise, go to step 6.
6. Process the interrupt(s) as required by the application software.

Software also has the option of disabling the interrupt and polling the TS_PEND_RAW bit of the CPTS_INTSTAT_RAW register to determine if a valid event is on the event FIFO.

18.8.4.11 CPPI Buffer Descriptors

The buffer descriptor is a central part of the GMAC_SW Ethernet Subsystem and is how the application software describes Ethernet packets to be sent and empty buffers to be filled with incoming packet data.

Host Software sends and receives network frames via the CPPI compliant host interface. The host interface includes module registers and host memory data structures. The host memory data structures are buffer descriptors and data buffers. Buffer descriptors are data structures that contain information about a single data buffer. Buffer descriptors may be linked together to describe frames or queues of frames for transmission of data and free buffer queues available for received data.

NOTE: The 8K bytes of Ethernet Subsystem CPPI RAM begin at address 0x4848 6000 and end at 0x4848 7FFF from the GMAC_SW perspective. The buffer descriptors programmed to access the CPPI RAM memory should use this address range.

18.8.4.11.1 TX Buffer Descriptors

A TX buffer descriptor (Table 18-823) is a contiguous block of four 32-bit data words aligned on a 32-bit word boundary.

Table 18-823. TX Buffer Descriptor Format

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Next Descriptor Pointer																															

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Buffer Pointer																															

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Buffer Offset																Buffer Length															

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOP	EOP	OWNER	EOQ	TDOWNCMPLT	PASSCRC	RESERVED										TO_PORT_EN	RESERVED	TO_PORT	RESERVED					Packet Length							

18.8.4.11.1.1 CPPI TX Data Word 0

Next Descriptor Pointer

The next descriptor pointer points to the 32-bit word aligned memory address of the next buffer descriptor in the transmit queue. This pointer is used to create a linked list of buffer descriptors. If the value of this pointer is zero, then the current buffer is the last buffer in the queue. The software application must set this value prior to adding the descriptor to the active transmit list. This pointer is not altered by the EMAC. The value of pNext should never be altered once the descriptor is in an active transmit queue, unless its current value is NULL. If the pNext pointer is initially NULL, and more packets need to be queued for transmit, the software application may alter this pointer to point to a newly appended descriptor. The EMAC will use the new pointer value and proceed to the next descriptor unless the pNext value has already been read. In this latter case, the transmitter will halt on the transmit channel in question, and the software application may restart it at that time. The software can detect this case by checking for an end of queue (EOQ) condition flag on the updated packet descriptor when it is returned by the EMAC

18.8.4.11.1.2 CPPI TX Data Word 1

Buffer Pointer

The byte aligned memory address of the buffer associated with the buffer descriptor. The host sets the buffer_pointer. The software application must set this value prior to adding the descriptor to the active transmit list. This pointer is not altered by the EMAC.

18.8.4.11.1.3 CPPI TX Data Word 2

Buffer Offset

Indicates how many unused bytes are at the start of the buffer. A value of 0000h indicates that no unused bytes are at the start of the buffer and that valid data begins on the first byte of the buffer. A value of 000Fh (decimal 15) indicates that the first 15 bytes of the buffer are to be ignored by the port and that valid buffer data starts on byte 16 of the buffer. The host sets the Buffer_Offset value (which may be zero to the buffer length minus 1). Valid only on SOP.

Buffer Length

Indicates how many valid data bytes are in the buffer. Unused or protocol specific bytes at the beginning of the buffer are not counted in the Buffer Length field. The host sets the Buffer_Length. The Buffer_Length must be greater than zero.

18.8.4.11.1.4 CPPI TX Data Word 3

Packet Length

Specifies the number of bytes in the entire packet. Offset bytes are not included. The sum of the Buffer_Length fields should equal the Packet_Length. Valid only on SOP. The packet length must be greater than zero. The packet data will be truncated to the packet length if the packet length is shorter than the sum of the packet buffer descriptor buffer lengths. A host error occurs if the packet length is greater than the sum of the packet buffer descriptor buffer lengths.

Start of Packet (SOP) Flag

When set, this flag indicates that the descriptor points to a packet buffer that is the start of a new packet. In the case of a single fragment packet, both the SOP and end of packet (EOP) flags are set. Otherwise,,the descriptor pointing to the last packet buffer for the packet sets the EOP flag. This bit is set by the software application and is not altered by the EMAC.

0 - Not start of packet buffer.

1 - Start of packet buffer.

End of Packet (EOP) Flag

When set, this flag indicates that the descriptor points to a packet buffer that is last for a given packet. In the case of a single fragment packet, both the start of packet (SOP) and EOP flags are set. Otherwise, the descriptor pointing to the last packet buffer for the packet sets the EOP flag. This bit is set by the software application and is not altered by the EMAC.

0 - Not end of packet buffer.

1 - End of packet buffer.

Ownership (OWNER) Flag

When set this flag indicates that all the descriptors for the given packet (from SOP to EOP) are currently owned by the EMAC. This flag is set by the software application on the SOP packet descriptor before adding the descriptor to the transmit descriptor queue. For a single fragment packet, the SOP, EOP, and OWNER flags are all set. The OWNER flag is cleared by the EMAC once it is finished with all the descriptors for the given packet. Note that this flag is valid on SOP descriptors only.

0 - The packet is owned by the host

1 - The packet is owned by the port

End of Queue (EOQ) Flag

When set, this flag indicates that the descriptor in question was the last descriptor in the transmit queue for a given transmit channel, and that the transmitter has halted. This flag is initially cleared by the software application prior to adding the descriptor to the transmit queue. This bit is set by the EMAC when the EMAC identifies that a descriptor is the last for a given packet (the EOP flag is set), and there are no more descriptors in the transmit list (next descriptor pointer is NULL).

The software application can use this bit to detect when the EMAC transmitter for the corresponding channel has halted. This is useful when the application appends additional packet descriptors to a transmit queue list that is already owned by the EMAC. Note that this flag is valid on EOP descriptors only.

0 - The TX queue has more packets to transfer.

1 - The Descriptor buffer is the last buffer in the last packet in the queue.

Teardown Complete (TDOWNCMPLT) Flag

This flag is used when a transmit queue is being torn down, or aborted, instead of allowing it to be transmitted. This would happen under device driver reset or shutdown conditions. The EMAC sets this bit in the SOP descriptor of each packet as it is aborted from transmission. Note that this flag is valid on SOP descriptors only. Also note that only the first packet in an unsent list has the TDOWNCMPLT flag set. Subsequent descriptors are not processed by the EMAC.

0 - The port has not completed the teardown process.

1 - The port has completed the commanded teardown process.

Pass CRC (PASSCRC) Flag

This flag is set by the software application in the SOP packet descriptor before it adds the descriptor to the transmit queue. Setting this bit indicates to the EMAC that the 4 byte Ethernet CRC is already present in the packet data, and that the EMAC should not generate its own version of the CRC. When the CRC flag is cleared, the EMAC generates and appends the 4-byte CRC. The buffer length and packet length fields do not include the CRC bytes. When the CRC flag is set, the 4-byte CRC is supplied by the software application and is already appended to the end of the packet data. The buffer length and packet length fields include the CRC bytes, as they are part of the valid packet data. Note that this flag is valid on SOP descriptors only.

0 - The CRC is not included with the packet data and packet length.

1 - The CRC is included with the packet data and packet length.

TO_PORT

Port number to send the directed packet to. This field is set by the host. This field is valid on SOP. Directed packets go to the directed port, but an ALE lookup is performed to determine untagged egress in VLAN_AWARE mode.

1 - Send the packet to port 1 if TO_PORT_EN is asserted.

2 - Send the packet to port 2 if TO_PORT_EN is asserted.

TO_PORT_ENABLE

Indicates when set that the packet is a directed packet to be sent to the TO_PORT field port number. This field is set by the host. The packet is sent to one port only (index not mask). This bit is valid on SOP.

0 - not a directed packet

1 - directed packet

18.8.4.11.2 RX Buffer Descriptors

A RX buffer descriptor (Table 18-824) is a contiguous block of four 32-bit data words aligned on a 32-bit word boundary.

Table 18-824. RX Buffer Descriptor Format

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Next Descriptor Pointer																															

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
Buffer Pointer																															

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED					Buffer Offset											RESERVED					Buffer Length										

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SOP	EOP	OWNER	EOQ	TDOWNCMPLT	PASSCRC	LONG	SHORT	MAC_CTL	OVERRUN	PKT_ERR	RX_VLAN_ENCAP	FROM_PORT				RESERVED					Packet Length										

18.8.4.11.2.1 CPPI RX Data Word 0

Next Descriptor Pointer

The 32-bit word aligned memory address of the next buffer descriptor in the RX queue. This is the mechanism used to reference the next buffer descriptor from the current buffer descriptor. If the value of this pointer is zero then the current buffer is the last buffer in the queue. The host sets the Next_Descriptor_Pointer.

18.8.4.11.2.2 CPPI RX Data Word 1

Buffer Pointer

The byte aligned memory address of the buffer associated with the buffer descriptor. The host sets the Buffer_Pointer.

18.8.4.11.2.3 CPPI RX Data Word 2

Buffer Offset

Indicates how many unused bytes are at the start of the buffer. A value of 0000h indicates that there are no unused bytes at the start of the buffer and that valid data begins on the first byte of the buffer. A value of 000Fh (decimal 15) indicates that the first 15 bytes of the buffer are to be ignored by the port and that valid buffer data starts on byte 16 of the buffer. The port writes the Buffer_Offset with the value from the CPDMA_RX_BUFFER_OFFSET register value. The host initializes the Buffer_Offset to zero for free buffers. The Buffer_Length must be greater than the CPDMARX_BUFFER_OFFSET register value. The buffer offset is valid only on SOP.

Buffer Length

Indicates how many valid data bytes are in the buffer. Unused or protocol specific bytes at the beginning of the buffer are not counted in the Buffer Length field. The host initializes the Buffer_Length, but the port may overwrite the host initiated value with the actual buffer length value on SOP and/or EOP buffer descriptors. SOP buffer length values will be overwritten if the packet size is less than the size of the buffer or if the offset is nonzero. EOP buffer length values will be overwritten if the entire buffer is not filled up with data. The Buffer_Length must be greater than zero.

18.8.4.11.2.4 CPPI RX Data Word 3

Packet Length

Specifies the number of bytes in the entire packet. Offset bytes are not included. The sum of the Buffer_Length fields should equal the Packet_Length. Valid only on SOP.

Start of Packet (SOP) Flag

When set, this flag indicates that the descriptor points to a packet buffer that is the start of a new packet. In the case of a single fragment packet, both the SOP and end of packet (EOP) flags are set. Otherwise, the descriptor pointing to the last packet buffer for the packet has the EOP flag set. This flag is initially cleared by the software application before adding the descriptor to the receive queue. This bit is set by the EMAC on SOP descriptors.

End of Packet (EOP) Flag

When set, this flag indicates that the descriptor points to a packet buffer that is last for a given packet. In the case of a single fragment packet, both the start of packet (SOP) and EOP flags are set. Otherwise, the descriptor pointing to the last packet buffer for the packet has the EOP flag set. This flag is initially cleared by the software application before adding the descriptor to the receive queue. This bit is set by the EMAC on EOP descriptors.

Ownership (OWNER) Flag

When set, this flag indicates that the descriptor is currently owned by the EMAC. This flag is set by the software application before adding the descriptor to the receive descriptor queue. This flag is cleared by the EMAC once it is finished with a given set of descriptors, associated with a received packet. The flag is updated by the EMAC on SOP descriptor only. So when the application identifies that the OWNER flag is cleared on an SOP descriptor, it may assume that all descriptors up to and including the first with the EOP flag set have been released by the EMAC. (Note that in the case of single buffer packets, the same descriptor will have both the SOP and EOP flags set.)

End of Queue (EOQ) Flag

When set, this flag indicates that the descriptor in question was the last descriptor in the receive queue for a given receive channel, and that the corresponding receiver channel has halted. This flag is initially cleared by the software application prior to adding the descriptor to the receive queue. This bit is set by the EMAC when the EMAC identifies that a descriptor is the last for a given packet received (also sets the EOP flag), and there are no more descriptors in the receive list (next descriptor pointer is NULL).The software application can use this bit to detect when the EMAC receiver for the corresponding channel has halted. This is useful when the application appends additional free buffer descriptors to an active receive queue. Note that this flag is valid on EOP descriptors only.

Teardown Complete (TDOWNCMPLT) Flag

This flag is used when a receive queue is being torn down, or aborted, instead of being filled with received data. This would happen under device driver reset or shutdown conditions. The EMAC sets this bit in the descriptor of the first free buffer when the tear down occurs. No additional queue processing is performed.

Pass CRC (PASSCRC) Flag

This flag is set by the EMAC in the SOP buffer descriptor if the received packet includes the 4-byte CRC. This flag should be cleared by the software application before submitting the descriptor to the receive queue.

Long (Jabber) Flag

This flag is set by the EMAC in the SOP buffer descriptor, if the received packet is a jabber frame and was not discarded because the RX_CEF_EN bit was set in the SL_MACCONTROL register. Jabber frames are frames that exceed the RXMAXLEN in length, and have CRC, code, or alignment errors.

Short (Fragment) Flag

This flag is set by the EMAC in the SOP buffer descriptor, if the received packet is only a packet fragment and was not discarded because the RX_CSF_EN bit was set in the SL_MACCONTROL register.

Control Flag

This flag is set by the EMAC in the SOP buffer descriptor, if the received packet is an EMAC control frame and was not discarded because the RX_CMF_EN bit was set in the SL_MACCONTROL register.

Overrun Flag

This flag is set by the EMAC in the SOP buffer descriptor, if the received packet was aborted due to a receive overrun.

Packet Error (PKT_ERR) Flag

Packet Contained Error on Ingress:

00 - no error

01 - CRC error on ingress

10 - Code error on ingress

11 - Align error on ingress

VLAN Encapsulated Packet (RX_VLAN_ENCAP)

Indicates when set that the packet data contains a 32-bit VLAN header word that is included in the packet byte count. This field is set by the port to be the value of the CPSW_CONTROL register RX_VLAN_ENCAP bit.

FROM_PORT

Indicates the port number that the packet was received on (ingress to the switch).

18.8.4.12 MDIO

The MII Management interface module implements the 802.3 serial management interface to interrogate and control two Ethernet PHYs simultaneously using a shared two-wire bus. Two user access registers (MDIO_USERACCESS0/MDIO_USERACCESS1) control and monitor up to two PHYs simultaneously.

18.8.4.12.1 MDIO Frame Formats

Table 18-825 shows the read format and Table 18-826 shows the write format of the 32-bit MII Management interface frames.

Table 18-825. MDIO Read Frame Format

Pre-amble	Start Delimiter	Operation Code	PHY Address	Register Address	Turnaround	Data
FFFF FFFFh	01	10	AAAAA	RRRRR	Z0	DDDD.DDDD.DDDD.DDDD

Table 18-826. MDIO Write Frame Format

Pre-amble	Start Delimiter	Operation Code	PHY Address	Register Address	Turnaround	Data
FFFF FFFFh	01	01	AAAAA	RRRRR	10	DDDD.DDDD.DDDD.DDDD

The default or idle state of the two wire serial interface is a logic one. All tri-state drivers should be disabled and the PHY's pull-up resistor will pull the MDIO line to a logic 1. Prior to initiating any other transaction, the station management entity shall send a preamble sequence of 32 contiguous logic 1 bits on the MDIO line with 32 corresponding cycles on MDCLK to provide the PHY with a pattern that it can use to establish synchronization. A PHY shall observe a sequence of 32 contiguous logic one bits on MDIO with 32 corresponding MDCLK cycles before it responds to any other transaction.

Preamble

The start of a frame is indicated by a preamble, which consists of a sequence of 32 contiguous bits all of which are a 1. This sequence provides the PHY a pattern to use to establish synchronization.

Start Delimiter

The preamble is followed by the start delimiter which is indicated by a 01 pattern. The pattern assures transitions from the default logic 1 state to logic 0, and back to logic 1.

Operation Code

The operation code for a read is 10, while the operation code for a write is a 01.

PHY Address

The PHY address is 5 bits allowing 32 unique values. The first bit transmitted is the MSB of the PHY address.

Register Address

The Register address is 5 bits allowing 32 registers to be addressed within each PHY. Refer to the 10/100 PHY address map for addresses of individual registers.

Turnaround

An idle bit time during which no device actively drives the MDIO signal shall be inserted between the register address field and the data field of a read frame in order to avoid contention. During a read frame, the PHY shall drive a zero bit onto MDIO for the first bit time following the idle bit and preceding the Data field. During a write frame, this field shall consist of a one bit followed by a zero bit.

Data

The Data field is 16 bits. The first bit transmitted and received is the MSB of the data word.

18.8.4.12.2 MDIO Functional Description

The MII Management I/F will remain idle until enabled by setting the ENABLE bit in the MDIO_CONTROL register. The MII Management I/F will then continuously poll the link status from within the Generic Status Register of all possible 32 PHY addresses in turn recording the results in the MDIO_LINK register. The LINKSEL bit in the MDIO_USERPHYSEL0/1 register determines the status input that is used. A change in the link status of the two PHYs being monitored will set the appropriate bit in the MDIO_LINKINTRAW register and the MDIO_LINKINTMASKED register, if enabled by the LINKINT_ENABLE bit in the MDIO_USERPHYSEL0/1 register.

The MDIO_ALIVE register is updated by the MII Management I/F module if the PHY acknowledged the read of the generic status register. In addition, any PHY register read transactions initiated by the host also cause the MDIO_ALIVE register to be updated.

At any time, the host can define a transaction for the MII Management interface module to undertake using the DATA, PHYADR, REGADR, and WRITE fields in a MDIO_USERACCESS0/1 register. When the host sets the GO bit in this register, the MII Management interface module will begin the transaction without any further intervention from the host. Upon completion, the MII Management interface will clear the GO bit and set the USERINTRAW bit in the MDIO_USERINTRAW register corresponding to the MDIO_USERACCESS0/1 register being used. The corresponding bit in the MDIO_USERINTMASKED register may also be set depending on the mask setting in the MDIO_USERINTMASKSET and MDIO_USERINTMASKCLR registers. A round-robin arbitration scheme is used to schedule transactions that may be queued by the host in different MDIO_USERACCESS0/1 registers. The host should check the status of the GO bit in the MDIO_USERACCESS0/1 register before initiating a new transaction to ensure that the previous transaction has completed. The host can use the ACK bit in the MDIO_USERACCESS0/1 register to determine the status of a read transaction.

It is necessary for software to use the MII Management interface module to setup the auto-negotiation parameters of each PHY attached to a MAC port, retrieve the negotiation results, and setup the SL_MACCONTROL register in the corresponding MAC.

18.8.5 GMAC_SW Programming Guide

18.8.5.1 Transmit Operation

After reset, the host must write zeroes to all TX DMA State head descriptor pointers. The TX port may then be enabled. To initiate packet transmission the host constructs transmit queues in memory (one or more packets for transmission) and then writes the appropriate TX DMA state head descriptor pointers. For each buffer added to a transmit queue, the host must initialize the TX buffer descriptor values as follows:

1. Write the Next Descriptor Pointer with the 32-bit aligned address of the next descriptor in the queue (zero if last descriptor)
2. Write the Buffer Pointer with the byte aligned address of the buffer data
3. Write the Buffer Length with the number of bytes in the buffer
4. Write the Buffer Offset with the number of bytes in the offset to the data (nonzero with SOP only)
5. Set the SOP, EOP, and Ownership bits as appropriate
6. Clear the End Of Queue bit

The port begins TX packet transmission on a given channel when the host writes the channel's TX queue head descriptor pointer with the address of the first buffer descriptor in the queue (nonzero value). Each channel may have one or more queues, so each channel may have one or more head descriptor pointers. The first buffer descriptor for each TX packet must have the Start of Packet (SOP) bit and the Ownership bit set to one by the host. The last buffer descriptor for each TX packet must have the End of Packet (EOP) bit set to one by the host. The port will transmit packets until all queued packets have been transmitted and the queue(s) are empty. When each packet transmission is complete, the port will clear the Ownership bit in the packet's SOP buffer descriptor and issue an interrupt to the host by writing the packet's last buffer descriptor address to the queue's TX DMA State Completion Pointer. The interrupt is generated by the write, regardless of the value written. When the last packet in a queue has been transmitted, the port sets the End Of Queue bit in the EOP buffer descriptor, clears the Ownership bit in the SOP Descriptor, zeroes the appropriate DMA state head descriptor pointer, and then issues a TX interrupt to the host by writing to the queue's associated TX completion pointer (address of the last buffer descriptor processed by the port). The port issues a maskable level interrupt (which may then be routed through external interrupt control logic to the host).

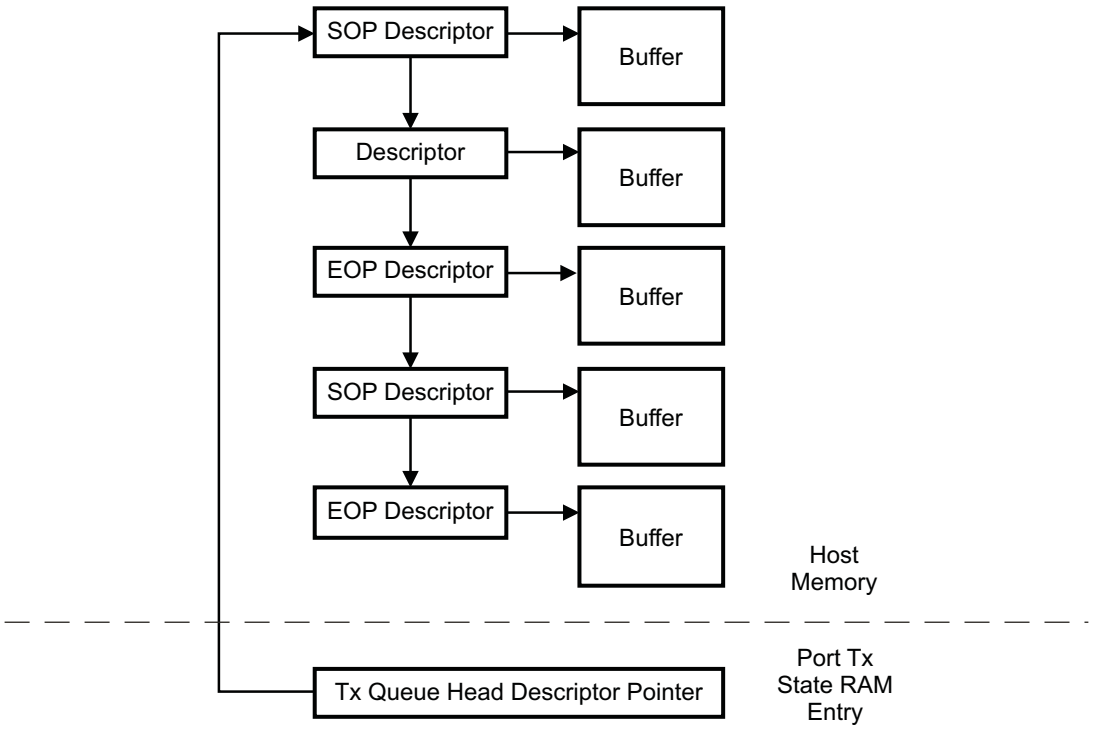
On interrupt from the port, the host processes the buffer queue, detecting transmitted packets by the status of the Ownership bit in the SOP buffer descriptor. If the Ownership bit is cleared to zero, then the packet has been transmitted and the host may reclaim the buffers associated with the packet. The host continues queue processing until the end of the queue or until a SOP buffer descriptor is read that contains a set Ownership bit indicating that the packet transmission is not complete. The host determines that all packets in the queue have been transmitted when the last packet in the queue has a cleared Ownership bit in the SOP buffer descriptor, the End of Queue bit is set in the last packet EOP buffer descriptor, and the Next Descriptor Pointer of the last packet EOP buffer descriptor is zero. The host acknowledges an interrupt by writing the address of the last buffer descriptor to the queue's associated TX Completion Pointer in the TX DMA State. If the host written buffer address value is different from the buffer address written by the port, then the level interrupt remains asserted. If the host written buffer address value is equal to the port written value, then the level interrupt is de-asserted. The port write to the completion pointer actually stores the value in the state register (RAM). The host written value is actually not written to the register location. The host written value is compared to the register contents (which was written by the port) and if the two values are equal, the interrupt is removed, otherwise the interrupt remains asserted. The host may process multiple packets previous to acknowledging an interrupt, or the host may acknowledge interrupts for every packet.

A mis-queued packet condition may occur when the host adds a packet to a queue for transmission as the port finishes transmitting the previous last packet in the queue. The mis-queued packet is detected by the host when queue processing detects a cleared Ownership bit in the SOP buffer descriptor, a set End of Queue bit in the EOP buffer descriptor, and a nonzero Next Descriptor Pointer in the EOP buffer descriptor. A mis-queued packet means that the port read the last EOP buffer descriptor before the host added the new last packet to the queue, so the port determined queue empty just before the last packet was added. The host corrects the mis-queued packet condition by initiating a new packet transfer for the mis-queued packet by writing the mis-queued packet's SOP buffer descriptor address to the appropriate DMA State TX Queue head Descriptor Pointer.

The host may add packets to the tail end of an active TX queue at any time by writing the Next Descriptor Pointer to the current last descriptor in the queue. If a TX queue is empty (inactive), the host may initiate packet transmission at any time by writing the appropriate TX DMA State head descriptor pointer. The host software should always check for and reinitiate transmission for mis-queued packets during queue processing on interrupt from the port. In order to preclude software underrun, the host should avoid adding buffers to an active queue for any TX packet that is not complete and ready for transmission.

The port determines that a packet is the last packet in the queue by detecting the End of Packet bit set with a zero Next Descriptor Pointer in the packet buffer descriptor. If the End of Packet bit is set and the Next Descriptor Pointer is nonzero, then the queue still contains one or more packets to be transmitted. If the EOP bit is set with a zero Next Descriptor Pointer, then the port will set the EOQ bit in the packet's EOP buffer descriptor and then zero the appropriate head descriptor pointer previous to interrupting the port (by writing the completion pointer) when the packet transmission is complete.

Figure 18-170. TX Queue Head Descriptor



18.8.5.2 Receive Operation

After reset, the host must write zeroes to all RX DMA State head descriptor pointers. The RX port may then be enabled. To initiate packet reception, the host constructs receive queues in memory and then writes the appropriate RX DMA state head descriptor pointer. For each RX buffer descriptor added to the queue, the host must initialize the RX buffer descriptor values as follows:

1. Write the Next Descriptor Pointer with the 32-bit aligned address of the next descriptor in the queue (zero if last descriptor)
2. Write the Buffer Pointer with the byte aligned address of the buffer data
3. Clear the Offset field
4. Write the Buffer Length with the number of bytes in the buffer
5. Clear the SOP, EOP, and EOQ bits
6. Set the Ownership bit

The host enables packet reception on a given channel by writing the address of the first buffer descriptor in the queue (nonzero value) to the channel's head descriptor pointer in the channel's RX DMA state. When packet reception begins on a given channel, the port fills each RX buffer with data in order starting with the first buffer and proceeding through the RX queue. If the Buffer Offset in the RX DMA State is nonzero, then the port will begin writing data after the offset number of bytes in the SOP buffer. The port performs the following operations at the end of each packet reception:

1. Overwrite the buffer length in the packet's EOP buffer descriptor with the number of bytes actually received in the packet's last buffer. The host initialized value is the buffer size. The overwritten value

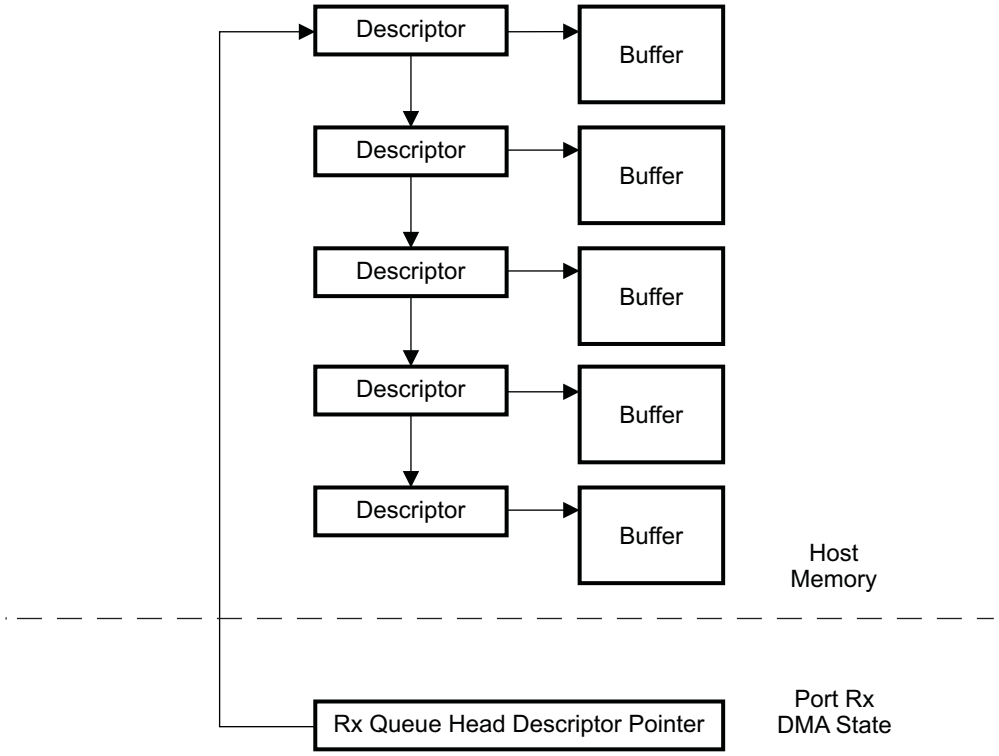
will be less than or equal to the host initialized value.

2. Set the EOP bit in the packet's EOP buffer descriptor.
3. Set the EOQ bit in the packet's EOP buffer descriptor if the current packet is the last packet in the queue.
4. Overwrite the packet's SOP buffer descriptor Buffer Offset with the RX DMA state value (the host initialized the buffer descriptor Buffer Offset value to zero). All non SOP buffer descriptors must have a zero Buffer Offset initialized by the host.
5. Overwrite the packet's SOP buffer descriptor buffer length with the number of valid data bytes in the buffer. If the buffer is filled up, the buffer length will be the buffer size minus buffer offset.
6. Set the SOP bit in the packet's SOP buffer descriptor.
7. Write the SOP buffer descriptor Packet Length field.
8. Clear the Ownership bit in the packet's SOP buffer descriptor.
9. Issue an RX host interrupt by writing the address of the packet's last buffer descriptor to the queue's RX DMA State Completion Pointer. The interrupt is generated by the write to the RX DMA State Completion Pointer address location, regardless of the value written.

On interrupt the host processes the RX buffer queue detecting received packets by the status of the Ownership bit in each packet's SOP buffer descriptor. If the Ownership bit is cleared then the packet has been completely received and is available to be processed by the host. The host may continue RX queue processing until the end of the queue or until a buffer descriptor is read that contains a set Ownership bit indicating that the next packet's reception is not complete. The host determines that the RX queue is empty when the last packet in the queue has a cleared Ownership bit in the SOP buffer descriptor, a set End of Queue bit in the EOP buffer descriptor, and the Next Descriptor Pointer in the EOP buffer descriptor is zero.

A mis-queued buffer may occur when the host adds buffers to a queue as the port finishes the reception of the previous last packet in the queue. The mis-queued buffer is detected by the host when queue processing detects a cleared Ownership bit in the SOP buffer descriptor, a set End of Queue bit in the EOP buffer descriptor, and a nonzero Next Descriptor Pointer in the EOP buffer descriptor. A mis-queued buffer means that the port read the last EOP buffer descriptor before the host added buffer descriptor(s) to the queue, so the port determined queue empty just before the host added more buffer descriptor(s). In the transmit case, the packet transmission is delayed by the time required for the host to determine the condition and reinitiate the transaction, but the packet is not actually lost. In the receive case, receive overrun condition may occur in the mis-queued buffer case. If a new packet reception is begun during the time that the port has determined the end of queue condition, then the received packet will overrun (start of packet overrun). If the mis-queued buffer occurs during the middle of a packet reception then middle of packet overrun may occur. If the mis-queued buffer occurs after the last packet has completed, and is corrected before the next packet reception begins, then overrun will not occur. The host acts on the mis-queued buffer condition by writing the added buffer descriptor address to the appropriate RX DMA State Head Descriptor Pointer.

Figure 18-171. RX Queue Head Descriptor



18.8.5.3 MDIO Software Interface

18.8.5.3.1 Initializing the MDIO Module

The following steps are performed by the application software or device driver to initialize the MDIO device:

1. Configure the PREAMBLE and CLKDIV bits in the MDIO Control register (MDIO_CONTROL).
2. Enable the MDIO module by setting the ENABLE bit in MDIO_CONTROL.
3. The MDIO PHY alive status register (MDIO_ALIVE) can be read in polling fashion until a PHY connected to the system responded, and the MDIO PHY link status register (MDIO_LINK) can determine whether this PHY already has a link.
4. Setup the appropriate PHY addresses in the MDIO user PHY select register (MDIO_USERPHYSEL0/1), and set the LINKINTENB bit to enable a link change event interrupt if desirable.
5. If an interrupt on general MDIO register access is desired, set the corresponding bit in the MDIO user command complete interrupt mask set register (MDIO_USERINTMASKSET) to use the MDIO user access register (MDIO_USERACCESS0/1). Since only one PHY is used in this device, the application software can use one MDIO_USERACCESS0/1 to trigger a completion interrupt; the other MDIO_USERACCESS0/1 is not setup.

18.8.5.3.2 Writing Data To a PHY Register

The MDIO module includes a user access register (MDIO_USERACCESS0/1) to directly access a specified PHY device.To write a PHY register, perform the following:

1. Check to ensure that the GO bit in the MDIO user access register (MDIO_USERACCESS0/1) is cleared.
2. Write to the GO, WRITE, REGADR, PHYADR, and DATA bits in MDIO_USERACCESS0/1 corresponding to the PHY and PHY register you want to write.

3. The write operation to the PHY is scheduled and completed by the MDIO module. Completion of the write operation can be determined by polling the GO bit in MDIO_USERACCESS0/1 for a 0.
4. Completion of the operation sets the corresponding USERINTRAW bit (0 or 1) in the MDIO user command complete interrupt register (MDIO_USERINTRAW) corresponding to MDIO_USERACCESS0/1 used. If interrupts have been enabled on this bit using the MDIO user command complete interrupt mask set register (MDIO_USERINTMASKSET), then the bit is also set in the MDIO user command complete interrupt register (MDIO_USERINTMASKED) and an interrupt is triggered on the host processor.

18.8.5.3.3 Reading Data From a PHY Register

The MDIO module includes a user access register (MDIO_USERACCESS0/1) to directly access a specified PHY device. To read a PHY register, perform the following:

1. Check to ensure that the GO bit in the MDIO user access register (MDIO_USERACCESS*n*) is cleared.
2. Write to the GO, REGADR, and PHYADR bits in MDIO_USERACCESS0/1 corresponding to the PHY and PHY register you want to read.
3. The read data value is available in the DATA bits in MDIO_USERACCESS0/1 after the module completes the read operation on the serial bus. Completion of the read operation can be determined by polling the GO and ACK bits in MDIO_USERACCESS0/1. After the GO bit has cleared, the ACK bit is set on a successful read.
4. Completion of the operation sets the corresponding USERINTRAW bit (0 or 1) in the MDIO user command complete interrupt register (MDIO_USERINTRAW) corresponding to MDIO_USERACCESS0/1 used. If interrupts have been enabled on this bit using the MDIO user command complete interrupt mask set register (MDIO_USERINTMASKSET), then the bit is also set in the MDIO user command complete interrupt register (MDIO_USERINTMASKED) and an interrupt is triggered on the host processor.

18.8.5.4 Initialization and Configuration of CPSW

To configure the GMAC_SW Ethernet Subsystem for operation, the host must perform the following:

1. Select the Interface (RGMII) Mode
2. Configure pads (PIN muxing), as per the interface selected.
3. Enable the GMAC_SW Ethernet Subsystem Clocks
4. Configure the PRCM registers CM_GMAC_CLKSTCTRL and CM_GMAC_GMAC_CLKCTRL to enable power and clocks to GMAC_SW Ethernet Subsystem.
5. Apply Soft Reset to GMAC_SW Subsystem, CPSW_3G, CPGMAC_SL1/2, and CPDMA
6. Initialize the HDPs (Header Description Pointer) and CPs (Completion Pointer) to NULL
7. Configure the Interrupts
8. Configure the CPSW_CONTROL register
9. Configure the CPSW_STAT_PORT_EN register
10. Configure the ALE
11. Configure the MDIO
12. Configure the CPDMA receive DMA controller
13. Configure the CPDMA transmit DMA controller
14. Configure the CPPI TX and RX Descriptors
15. Configure CPGMAC_SL1 and CPGMAC_SL2, as per the desired mode of operations.
16. Start up RX and TX DMA (Write to HDP of RX and TX)
17. Wait for the completion of Transfer (HDP cleared to 0)

18.8.6 GMAC_SW Register Manual

18.8.6.1 GMAC_SW Instance Summary

Table 18-827. GMAC_SW Instance Summary

Module Name	Module Base Address	Size
SS	0x4848 4000	80 Bytes
PORT	0x4848 4100	1792 Bytes
CPDMA	0x4848 4800	256 Bytes
STATS	0x4848 4900	128 Bytes
STATERAM	0x4848 4A00	288 Bytes
CPTS	0x4848 4C00	240 Bytes
ALE	0x4848 4D00	88 Bytes
SL1	0x4848 4D80	64 Bytes
SL2	0x4848 4DC0	64 Bytes
MDIO	0x4848 5000	192 Bytes
WR	0x4848 5200	352 Bytes
SPF1	0x4848 5C00	512 Bytes
SPF2	0x4848 5E00	512 Bytes

NOTE: CPPI RAM address space starts at 0x4848 6000 and is 8 KiB deep. For details about CPPI RAM, see Section 18.8.4.11, *CPPI Buffer Descriptors.*.

18.8.6.2 SS Registers

18.8.6.2.1 SS Register Summary

Table 18-828. SS Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	SS Physical Address
CPSW_ID_VER	R	32	0x0000 0000	0x4848 4000
CPSW_CONTROL	RW	32	0x0000 0004	0x4848 4004
CPSW_SOFT_RESET	RW	32	0x0000 0008	0x4848 4008
CPSW_STAT_PORT_EN	RW	32	0x0000 000C	0x4848 400C
CPSW_PTYPE	RW	32	0x0000 0010	0x4848 4010
CPSW_SOFT_IDLE	RW	32	0x0000 0014	0x4848 4014
CPSW_THRU_RATE	RW	32	0x0000 0018	0x4848 4018
CPSW_GAP_THRESH	RW	32	0x0000 001C	0x4848 401C
CPSW_TX_START_WDS	RW	32	0x0000 0020	0x4848 4020
CPSW_FLOW_CONTROL	RW	32	0x0000 0024	0x4848 4024
CPSW_VLAN_LTYPE	RW	32	0x0000 0028	0x4848 4028
CPSW_TS_LTYPE	RW	32	0x0000 002C	0x4848 402C
CPSW_DLR_LTYPE	RW	32	0x0000 0030	0x4848 4030
CPSW_EEE_PRESCALE	RW	32	0x0000 0034	0x4848 4034

18.8.6.2.2 SS Register Description

Table 18-829. CPSW_ID_VER

Address Offset	0x0000 0000		
Physical Address	0x4848 4000	**Instance**	SS
Description	CPSW_3G ID version register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REVISION																															

Bits	Field Name	Description	Type	Reset
31:0	REVISION	CPSW_3G Revision Value	R	0x-

Table 18-830. Register Call Summary for Register CPSW_ID_VER

Gigabit Ethernet Switch (GMAC_SW)
- SS Register Summary: [0]

Table 18-831. CPSW_CONTROL

Address Offset	0x0000 0004		
Physical Address	0x4848 4004	**Instance**	SS
Description	Switch control register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																											EEE_EN	DLR_EN	RX_VLAN_ENCAP	VLAN_AWARE	FIFO_LOOPBACK

Bits	Field Name	Description	Type	Reset
31:5	RESERVED		R	0x0
4	EEE_EN	EEE (Energy Efficient Ethernet) enable 0 – EEE is disabled. 1 – EEE is enabled	RW	0x0
3	DLR_EN	DLR enable 0 - DLR is disabled. DLR packets will not be moved to queue priority 3 and will not be separated out onto dlr_cpdma_ch. 1 - DLR is disabled. DLR packets be moved to destination port transmit queue priority 3 and will be separated out onto dlr_cpdma_ch when packet is to egress on port 0.	RW	0x0
2	RX_VLAN_ENCAP	Port 0 VLAN Encapsulation (egress): 0 - Port 0 receive packets (from CPSW_3G) are not VLAN encapsulated. 1 - Port 0 receive packets (from CPSW_3G) are VLAN encapsulated.	RW	0x0
1	VLAN_AWARE	VLAN Aware Mode: 0 - CPSW_3G is in the VLAN unaware mode. 1 - CPSW_3G is in the VLAN aware mode.	RW	0x0

Bits	Field Name	Description	Type	Reset
0	FIFO_LOOPBACK	FIFO Loopback Mode 0 - Loopback is disabled 1 - FIFO Loopback mode enabled. Each packet received is turned around and sent out on the same port's transmit path. Port 2 receive is fixed on channel zero. The RXSOFOVERRUN statistic will increment for every packet sent in FIFO loopback mode.	RW	0x0

Table 18-832. Register Call Summary for Register CPSW_CONTROL

Gigabit Ethernet Switch (GMAC_SW)

- Address Lookup Engine (ALE): [0][1][2][3]
- FIFO Transmit Queue Control: [4]
- FIFO Loopback: [5]
- Device Level Ring (DLR) Support: [6][7]
- Energy Efficient Ethernet Support (802.3az): [8]
- RX Buffer Descriptors: [9]
- Initialization and Configuration of CPSW: [10]
- SS Register Summary: [11]

Table 18-833. CPSW_SOFT_RESET

Address Offset	0x0000 0008		
Physical Address	0x4848 4008	**Instance**	SS
Description	Soft reset register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															SOFT_RESET

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	SOFT_RESET	Software reset - Writing a one to this bit causes the 3G logic (INT, REGS, CPPI, and SPF modules) to be reset. After writing a one to this bit, it may be polled to determine if the reset has occurred. If a one is read, the reset has not yet occurred. If a zero is read then reset has occurred.	RW	0x0

Table 18-834. Register Call Summary for Register CPSW_SOFT_RESET

Gigabit Ethernet Switch (GMAC_SW)

- Software Reset: [0]
- SS Register Summary: [1]

Table 18-835. CPSW_STAT_PORT_EN

Address Offset	0x0000 000C		
Physical Address	0x4848 400C	**Instance**	SS
Description	Statistics port enable register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																													P2_STAT_EN	P1_STAT_EN	P0_STAT_EN

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0
2	P2_STAT_EN	Port 2 (GMII2 and Port 2 FIFO) Statistics Enable 0 - Port 2 statistics are not enabled. 1 - Port 2 statistics are enabled.	RW	0x0
1	P1_STAT_EN	Port 1 (GMII1 and Port 1 FIFO) Statistics Enable 0 - Port 1 statistics are not enabled. 1 - Port 1 statistics are enabled.	RW	0x0
0	P0_STAT_EN	Port 0 Statistics Enable 0 - Port 0 statistics are not enabled 1 - Port 0 statistics are enabled. FIFO overruns (SOFOVERRUNS) are the only port 0 statistics that are enabled to be kept.	RW	0x0

Table 18-836. Register Call Summary for Register CPSW_STAT_PORT_EN

Gigabit Ethernet Switch (GMAC_SW)

- CPSW_3G Network Statistics: [0]
- Initialization and Configuration of CPSW: [1]
- SS Register Summary: [2]

Table 18-837. CPSW_PTYPE

Address Offset	0x0000 0010		
Physical Address	0x4848 4010	**Instance**	SS
Description	Transmit priority type register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED										P2_PRI3_SHAPE_EN	P2_PRI2_SHAPE_EN	P2_PRI1_SHAPE_EN	P1_PRI3_SHAPE_EN	P1_PRI2_SHAPE_EN	P1_PRI1_SHAPE_EN	RESERVED					P2_PTYPE_ESC	P1_PTYPE_ESC	P0_PTYPE_ESC	RESERVED				ESC_PRI_LD_VAL			

Bits	Field Name	Description	Type	Reset
31:22	RESERVED		R	0x0
21	P2_PRI3_SHAPE_EN	Port 2 Queue Priority 3 Transmit Shape Enable - If there is only one shaping queue then it must be priority 3.	RW	0x0

Bits	Field Name	Description	Type	Reset
20	P2_PRI2_SHAPE_EN	Port 2 Queue Priority 2 Transmit Shape Enable - If there are two shaping queues then they must be priorities 3 and 2.	RW	0x0
19	P2_PRI1_SHAPE_EN	Port 2 Queue Priority 1 Transmit Shape Enable - If there are three shaping queues all three bits should be set.	RW	0x0
18	P1_PRI3_SHAPE_EN	Port 1 Queue Priority 3 Transmit Shape Enable - If there is only one shaping queue then it must be priority 3.	RW	0x0
17	P1_PRI2_SHAPE_EN	Port 1 Queue Priority 2 Transmit Shape Enable- If there are two shaping queues then they must be priorities 3 and 2.	RW	0x0
16	P1_PRI1_SHAPE_EN	Port 1 Queue Priority 1 Transmit Shape Enable- If there are three shaping queues all three bits should be set.	RW	0x0
15:11	RESERVED		R	0x0
10	P2_PTYPE_ESC	Port 2 Priority Type Escalate - 0 - Port 2 priority type fixed 1 - Port 2 priority type escalate Escalate should not be used with queue shaping.	RW	0x0
9	P1_PTYPE_ESC	Port 1 Priority Type Escalate - 0 - Port 1 priority type fixed 1 - Port 1 priority type escalate Escalate should not be used with queue shaping.	RW	0x0
8	P0_PTYPE_ESC	Port 0 Priority Type Escalate - 0 - Port 0 priority type fixed 1 - Port 0 priority type escalate Escalate should not be used with queue shaping.	RW	0x0
7:5	RESERVED		R	0x0
4:0	ESC_PRI_LD_VAL	Escalate Priority Load Value When a port is in escalate priority, this is the number of higher priority packets sent before the next lower priority is allowed to send a packet. Escalate priority allows lower priority packets to be sent at a fixed rate relative to the next higher priority.	RW	0x0

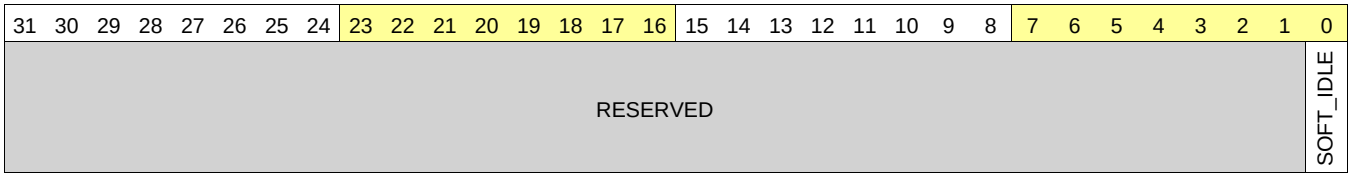
Table 18-838. Register Call Summary for Register CPSW_PTYPE

Gigabit Ethernet Switch (GMAC_SW)

- FIFO Transmit Queue Control: [0]
- Audio Video Bridging: [1]
- SS Register Summary: [2]
- PORT Register Description: [3][4][5][6][7][8]

Table 18-839. CPSW_SOFT_IDLE

Address Offset	0x0000 0014		
Physical Address	0x4848 4014	**Instance**	SS
Description	Software idle		
Type	RW		



Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	SOFT_IDLE	Software Idle - Setting this bit causes the switch fabric to stop forwarding packets at the next start of packet.	RW	0x0

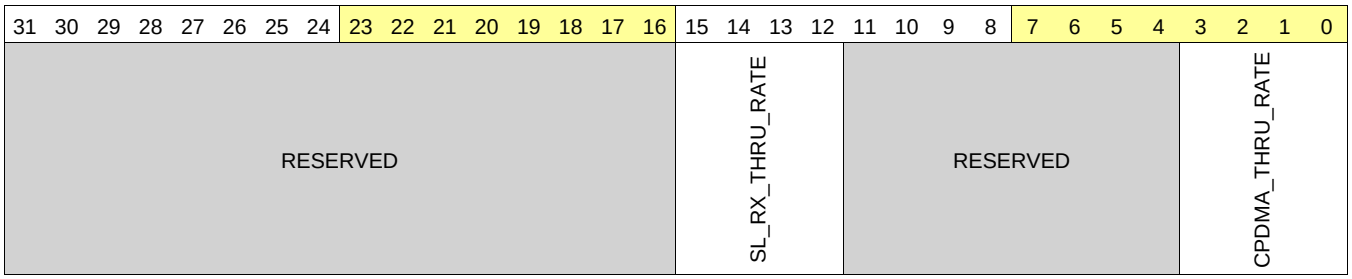
Table 18-840. Register Call Summary for Register CPSW_SOFT_IDLE

Gigabit Ethernet Switch (GMAC_SW)

- Software IDLE: [0]
- SS Register Summary: [1]

Table 18-841. CPSW_THRU_RATE

Address Offset	0x0000 0018		
Physical Address	0x4848 4018	**Instance**	SS
Description	Throughput rate		
Type	RW		



Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0
15:12	SL_RX_THRU_RATE	CPGMAC_SL Switch FIFO receive through rate. This register value is the maximum throughput of the ethernet ports to the crossbar SCR. The default is one 8-byte word for every 3 MAIN_CLK periods maximum.	RW	0x3
11:4	RESERVED		R	0x0
3:0	CPDMA_THRU_RATE	CPDMA Switch FIFO receive through rate. This register value is the maximum throughput of the CPDMA host port to the crossbar SCR. The default is one 8-byte word for every 3 MAIN_CLK periods maximum.	RW	0x3

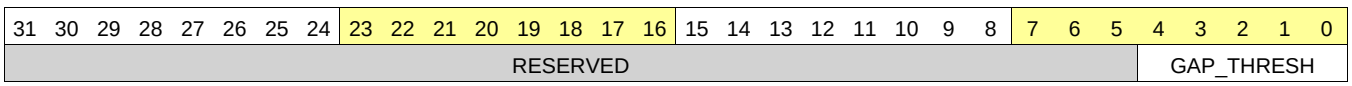
Table 18-842. Register Call Summary for Register CPSW_THRU_RATE

Gigabit Ethernet Switch (GMAC_SW)

- SS Register Summary: [0]

Table 18-843. CPSW_GAP_THRESH

Address Offset	0x0000 001C		
Physical Address	0x4848 401C	**Instance**	SS
Description	CPGMAC_SL short gap threshold		
Type	RW		



Bits	Field Name	Description	Type	Reset
31:5	RESERVED		R	0x0
4:0	GAP_THRESH	CPGMAC_SL Short Gap Threshold - This is the CPGMAC_SL associated FIFO transmit block usage value for triggering TX_SHORT_GAP.	RW	0xB

Table 18-844. Register Call Summary for Register CPSW_GAP_THRESH

Gigabit Ethernet Switch (GMAC_SW)

- Short Gap: [0][1]
- SS Register Summary: [2]

Table 18-845. CPSW_TX_START_WDS

Address Offset	0x0000 0020		
Physical Address	0x4848 4020	**Instance**	SS
Description	Transmit start words		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																					TX_START_WDS										

Bits	Field Name	Description	Type	Reset
31:11	RESERVED		R	0x0
10:0	TX_START_WDS	FIFO Packet Transmit (egress) Start Words. This value is the number of required packet words in the transmit FIFO before the packet egress will begin. This value is non-zero to preclude underrun. Decimal 32 is the recommended value. It should not be increased unnecessairly to prevent adding to the switch latency.	RW	0x20

Table 18-846. Register Call Summary for Register CPSW_TX_START_WDS

Gigabit Ethernet Switch (GMAC_SW)

- SS Register Summary: [0]

Table 18-847. CPSW_FLOW_CONTROL

Address Offset	0x0000 0024		
Physical Address	0x4848 4024	**Instance**	SS
Description	Flow control		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																													P2_FLOW_EN	P1_FLOW_EN	P0_FLOW_EN

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0
2	P2_FLOW_EN	Port 2 Receive flow control enable	RW	0x0
1	P1_FLOW_EN	Port 1 Receive flow control enable	RW	0x0
0	P0_FLOW_EN	Port 0 Receive flow control enable	RW	0x1

Table 18-848. Register Call Summary for Register CPSW_FLOW_CONTROL

Gigabit Ethernet Switch (GMAC_SW)

- Flow Control: [0]
- SS Register Summary: [1]

Table 18-849. CPSW_VLAN_LTYPE

Address Offset	0x0000 0028		
Physical Address	0x4848 4028	**Instance**	SS
Description	LTYPE1 and LTYPE 2 Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VLAN_LTYPE2																VLAN_LTYPE1															

Bits	Field Name	Description	Type	Reset
31:16	VLAN_LTYPE2	Time Sync VLAN LTYPE2 This VLAN LTYPE value is used for tx and rx. This is the inner VLAN if both are present.	RW	0x8100
15:0	VLAN_LTYPE1	Time Sync VLAN LTYPE1 This VLAN LTYPE value is used for tx and rx. This is the outer VLAN if both are present.	RW	0x8100

Table 18-850. Register Call Summary for Register CPSW_VLAN_LTYPE

Gigabit Ethernet Switch (GMAC_SW)

- Device Level Ring (DLR) Support: [0][1][2][3]
- Time Sync Events: [4][5][6][7]
- SS Register Summary: [8]

Table 18-851. CPSW_TS_LTYPE

Address Offset	0x0000 002C		
Physical Address	0x4848 402C	**Instance**	SS
Description	VLAN_LTYPE1 and VLAN_LTYPE2 Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TS_LTYPE2																TS_LTYPE1															

Bits	Field Name	Description	Type	Reset
31:16	TS_LTYPE2	Time Sync LTYPE2 This is an Ethertype value to match for tx and rx time sync packets.	RW	0x0
15:0	TS_LTYPE1	Time Sync LTYPE1 This is an ethertype value to match for tx and rx time sync packets.	RW	0x0

Table 18-852. Register Call Summary for Register CPSW_TS_LTYPE

Gigabit Ethernet Switch (GMAC_SW)

- Time Sync Events: [0][1]
- SS Register Summary: [2]

Table 18-853. CPSW_DLR_LTYPE

Address Offset	0x0000 0030		
Physical Address	0x4848 4030	**Instance**	SS
Description	DLR LTYPE register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																DLR_LTYPE															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0
15:0	DLR_LTYPE	DLR LTYPE. This is the ethertype value to match for DLR packets.	RW	0x80E1

Table 18-854. Register Call Summary for Register CPSW_DLR_LTYPE

Gigabit Ethernet Switch (GMAC_SW)

- Device Level Ring (DLR) Support: [0][1][2]
- SS Register Summary: [3]

Table 18-855. CPSW_EEE_PRESCALE

Address Offset	0x0000 0034		
Physical Address	0x4848 4034	**Instance**	SS
Description	EEE Pre-scale Counter Load Value Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																				EEE_PRESCALE											

Bits	Field Name	Description	Type	Reset
31:12	RESERVED		R	0x0
11:0	EEE_PRESCALE	Energy Efficient Ethernet Pre-scale count load value – This value is loaded into the EEE pre-scale counter each time the pre-scale count decrements to zero. The EEE counters are enabled to decrement each time the pre-scale counter reaches zero (and the EEE counters are enabled to count time). If this value is zero then the EEE counters decrement on every clock. If this value is 0x001 then the counters decrement on every other clock (and so on).	RW	0x0

Table 18-856. Register Call Summary for Register CPSW_EEE_PRESCALE

Gigabit Ethernet Switch (GMAC_SW)

- Energy Efficient Ethernet Support (802.3az): [0]
- SS Register Summary: [1]

18.8.6.3 PORT Registers

18.8.6.3.1 PORT Register Summary

Table 18-857. PORT Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	PORT Physical Address
P0_CONTROL	RW	32	0x0000 0000	0x4848 4100
P0_MAX_BLKS	RW	32	0x0000 0008	0x4848 4108
P0_BLK_CNT	RW	32	0x0000 000C	0x4848 410C
P0_TX_IN_CTL	RW	32	0x0000 0010	0x4848 4110
P0_PORT_VLAN	RW	32	0x0000 0014	0x4848 4114
P0_TX_PRI_MAP	RW	32	0x0000 0018	0x4848 4118
P0_CPDMA_TX_PRI_MAP	RW	32	0x0000 001C	0x4848 411C
P0_CPDMA_RX_CH_MAP	RW	32	0x0000 0020	0x4848 4120
P0_RX_DSCP_PRI_MAP0	RW	32	0x0000 0030	0x4848 4130
P0_RX_DSCP_PRI_MAP1	RW	32	0x0000 0034	0x4848 4134
P0_RX_DSCP_PRI_MAP2	RW	32	0x0000 0038	0x4848 4138
P0_RX_DSCP_PRI_MAP3	RW	32	0x0000 003C	0x4848 413C
P0_RX_DSCP_PRI_MAP4	RW	32	0x0000 0040	0x4848 4140
P0_RX_DSCP_PRI_MAP5	RW	32	0x0000 0044	0x4848 4144
P0_RX_DSCP_PRI_MAP6	RW	32	0x0000 0048	0x4848 4148
P0_RX_DSCP_PRI_MAP7	RW	32	0x0000 004C	0x4848 414C
P0_IDLE2LPI	RW	32	0x0000 0050	0x4848 4150
P0_LPI2WAKE	RW	32	0x0000 0054	0x4848 4154
P1_CONTROL	RW	32	0x0000 0100	0x4848 4200
P1_MAX_BLKS	RW	32	0x0000 0108	0x4848 4208
P1_BLK_CNT	RW	32	0x0000 010C	0x4848 420C
P1_TX_IN_CTL	RW	32	0x0000 0110	0x4848 4210
P1_PORT_VLAN	RW	32	0x0000 0114	0x4848 4214
P1_TX_PRI_MAP	RW	32	0x0000 0118	0x4848 4218
P1_TS_SEQ_MTYPE	RW	32	0x0000 011C	0x4848 421C
P1_SA_LO	RW	32	0x0000 0120	0x4848 4220
P1_SA_HI	RW	32	0x0000 0124	0x4848 4224
P1_SEND_PERCENT	RW	32	0x0000 0128	0x4848 4228
P1_RX_DSCP_PRI_MAP0	RW	32	0x0000 0130	0x4848 4230
P1_RX_DSCP_PRI_MAP1	RW	32	0x0000 0134	0x4848 4234
P1_RX_DSCP_PRI_MAP2	RW	32	0x0000 0138	0x4848 4238
P1_RX_DSCP_PRI_MAP3	RW	32	0x0000 013C	0x4848 423C
P1_RX_DSCP_PRI_MAP4	RW	32	0x0000 0140	0x4848 4240
P1_RX_DSCP_PRI_MAP5	RW	32	0x0000 0144	0x4848 4244
P1_RX_DSCP_PRI_MAP6	RW	32	0x0000 0148	0x4848 4248
P1_RX_DSCP_PRI_MAP7	RW	32	0x0000 014C	0x4848 424C
P1_IDLE2LPI	RW	32	0x0000 0150	0x4848 4250
P1_LPI2WAKE	RW	32	0x0000 0154	0x4848 4254
P2_CONTROL	RW	32	0x0000 0200	0x4848 4300
P2_MAX_BLKS	RW	32	0x0000 0208	0x4848 4308
P2_BLK_CNT	RW	32	0x0000 020C	0x4848 430C
P2_TX_IN_CTL	RW	32	0x0000 0210	0x4848 4310
P2_PORT_VLAN	RW	32	0x0000 0214	0x4848 4314

Table 18-857. PORT Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	PORT Physical Address
P2_TX_PRI_MAP	RW	32	0x0000 0218	0x4848 4318
P2_TS_SEQ_MTYPE	RW	32	0x0000 021C	0x4848 431C
P2_SA_LO	RW	32	0x0000 0220	0x4848 4320
P2_SA_HI	RW	32	0x0000 0224	0x4848 4324
P2_SEND_PERCENT	RW	32	0x0000 0228	0x4848 4328
P2_RX_DSCP_PRI_MAP0	RW	32	0x0000 0230	0x4848 4330
P2_RX_DSCP_PRI_MAP1	RW	32	0x0000 0234	0x4848 4334
P2_RX_DSCP_PRI_MAP2	RW	32	0x0000 0238	0x4848 4338
P2_RX_DSCP_PRI_MAP3	RW	32	0x0000 023C	0x4848 433C
P2_RX_DSCP_PRI_MAP4	RW	32	0x0000 0240	0x4848 4340
P2_RX_DSCP_PRI_MAP5	RW	32	0x0000 0244	0x4848 4344
P2_RX_DSCP_PRI_MAP6	RW	32	0x0000 0248	0x4848 4348
P2_RX_DSCP_PRI_MAP7	RW	32	0x0000 024C	0x4848 434C
P2_IDLE2LPI	RW	32	0x0000 0250	0x4848 4350
P2_LPI2WAKE	RW	32	0x0000 0254	0x4848 4354

18.8.6.3.2 PORT Register Description

Table 18-858. P0_CONTROL

Address Offset	0x0000 0000		
Physical Address	0x4848 4100	**Instance**	PORT
Description	CPSW PORT 0 control register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	P0_DLR_CPDMA_CH			RESERVED			P0_PASS_PRI_TAGGED	RESERVED		P0_VLAN_LTYPE2_EN	P0_VLAN_LTYPE1_EN	RESERVED			P0_DSCP_PRI_EN	RESERVED															

Bits	Field Name	Description	Type	Reset
31	RESERVED		R	0x0
30:28	P0_DLR_CPDMA_CH	Port 0 DLR CPDMA Channel This field indicates the CPDMA channel that DLR packets will be received on.	RW	0x0
27:25	RESERVED		R	0x0
24	P0_PASS_PRI_TAGGED	Port 0 Pass Priority Tagged 0 - Priority tagged packets have the zero VID replaced with the input port P0_PORT_VLAN [11:0] 1 - Priority tagged packets are processed unchanged.	RW	0x0
23:22	RESERVED		R	0x0
21	P0_VLAN_LTYPE2_EN	Port 0 VLAN LTYPE 2 enable 0 - disabled 1 - enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
20	P0_VLAN_LTYPE1_EN	Port 0 VLAN LTYPE 1 enable 0 - disabled 1 - enabled	RW	0x0
19:17	RESERVED		R	0x0
16	P0_DSCP_PRI_EN	Port 0 DSCP Priority Enable 0 - DSCP priority disabled 1 - DSCP priority enabled. All non-tagged IPV4 packets have their received packet priority determined by mapping the 6 TOS bits through the port DSCP priority mapping registers.	RW	0x0
15:0	RESERVED		RW	0x0

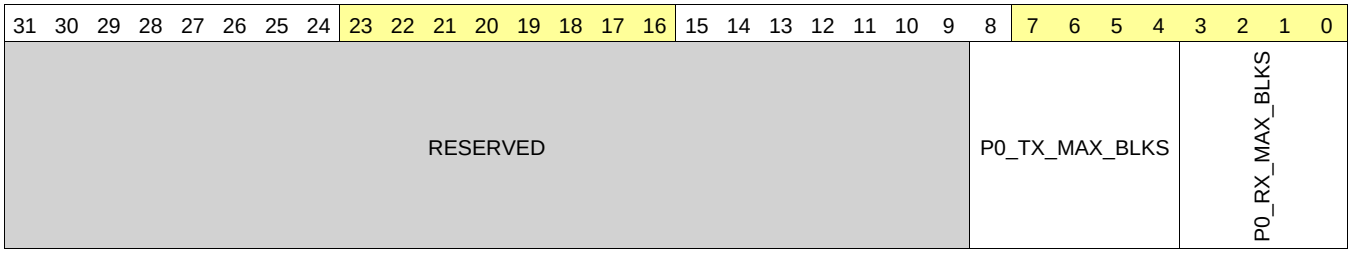
Table 18-859. Register Call Summary for Register P0_CONTROL

Gigabit Ethernet Switch (GMAC_SW)

- Device Level Ring (DLR) Support: [0]
- PORT Register Summary: [1]

Table 18-860. P0_MAX_BLKS

Address Offset	0x0000 0008		
Physical Address	0x4848 4108	**Instance**	PORT
Description	CPSW PORT 0 maximum FIFO blocks register		
Type	RW		



Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8:4	P0_TX_MAX_BLKS	Transmit FIFO Maximum Blocks - This value is the maximum number of 1k memory blocks that may be allocated to the FIFO's logical transmit priority queues. 0x10 is the recommended value of P0_TX_MAX_BLKS. Port 0 should remain in flow control mode. 0xE is the minimum value P0_TX_MAX_BLKS.	RW	0x10
3:0	P0_RX_MAX_BLKS	Receive FIFO Maximum Blocks - This value is the maximum number of 1k memory blocks that may be allocated to the FIFO's logical receive queue. 0x4 is the recommended value. 0x3 is the minimum value P0_RX_MAX_BLKS and 0x6 is the maximum value.	RW	0x4

Table 18-861. Register Call Summary for Register P0_MAX_BLKS

Gigabit Ethernet Switch (GMAC_SW)

- PORT Register Summary: [0]

Table 18-862. P0_BLK_CNT

Address Offset	0x0000 000C		
Physical Address	0x4848 410C	**Instance**	PORT
Description	CPSW PORT 0 FIFO block usage count (read only)		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								P0_TX_BLK_CNT							
																												P0_RX_BLK_CNT			

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8:4	P0_TX_BLK_CNT	Port 0 Transmit Block Count Usage - This value is the number of blocks allocated to the FIFO logical transmit queues.	R	0x4
3:0	P0_RX_BLK_CNT	Port 0 Receive Block Count Usage - This value is the number of blocks allocated to the FIFO logical receive queues.	R	0x1

Table 18-863. Register Call Summary for Register P0_BLK_CNT

Gigabit Ethernet Switch (GMAC_SW)

- PORT Register Summary: [0]

Table 18-864. P0_TX_IN_CTL

Address Offset	0x0000 0010		
Physical Address	0x4848 4110	**Instance**	PORT
Description	CPSW PORT 0 transmit FIFO control		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								TX_RATE_EN				RESERVED		TX_IN_SEL		TX_BLKS_REM				RESERVED		TX_PRI_WDS									

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:20	TX_RATE_EN	Transmit FIFO Input Rate Enable	RW	0x0
19:18	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
17:16	TX_IN_SEL	Transmit FIFO Input Queue Type Select 00 - Normal priority mode 01 - Dual MAC mode 10 - Rate Limit mode 11 - reserved Note that Dual MAC mode is not compatible with escalation or shaping because dual MAC mode forces round robin priority on FIFO egress. Rate-limiting and shaping are still available for Port 1 and Port 2 when Port 0 is set in dual MAC mode.	RW	0x0
15:12	TX_BLKS_REM	Transmit FIFO Input Blocks to subtract in dual MAC mode	RW	0x4
11:10	RESERVED		R	0x0
9:0	TX_PRI_WDS	Transmit FIFO Words in queue	RW	0xC0

Table 18-865. Register Call Summary for Register P0_TX_IN_CTL

Gigabit Ethernet Switch (GMAC_SW)
- FIFO Transmit Queue Control: [0][1]
- PORT Register Summary: [2]

Table 18-866. P0_PORT_VLAN

Address Offset	0x0000 0014		
Physical Address	0x4848 4114	**Instance**	PORT
Description	CPSW PORT 0 VLAN register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																PORT_PRI			PORT_CFI	PORT_VID											

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0
15:13	PORT_PRI	Port VLAN Priority (7 is highest priority)	RW	0x0
12	PORT_CFI	Port CFI bit	RW	0x0
11:0	PORT_VID	Port VLAN ID	RW	0x0

Table 18-867. Register Call Summary for Register P0_PORT_VLAN

Gigabit Ethernet Switch (GMAC_SW)
- PORT Register Summary: [0]
- PORT Register Description: [1]

Table 18-868. P0_TX_PRI_MAP

Address Offset	0x0000 0018		
Physical Address	0x4848 4118	**Instance**	PORT
Description	CPSW PORT 0 TX header priority to switch priority mapping register		
Type	RW		

31 30	29 28	27 26	25 24	23 22	21 20	19 18	17 16	15 14	13 12	11 10	9 8	7 6	5 4	3 2	1 0
RESERVED	PRI7	RESERVED	PRI6	RESERVED	PRI5	RESERVED	PRI4	RESERVED	PRI3	RESERVED	PRI2	RESERVED	PRI1	RESERVED	PRI0

Bits	Field Name	Description	Type	Reset
31:30	RESERVED		R	0x0
29:28	PRI7	Priority 7 - A packet header priority of 0x7 is given this switch queue priority.	RW	0x3
27:26	RESERVED		R	0x0
25:24	PRI6	Priority 6 - A packet header priority of 0x6 is given this switch queue priority.	RW	0x3
23:22	RESERVED		R	0x0
21:20	PRI5	Priority 5 - A packet header priority of 0x5 is given this switch queue priority.	RW	0x2
19:18	RESERVED		R	0x0
17:16	PRI4	Priority 4 - A packet header priority of 0x4 is given this switch queue priority.	RW	0x2
15:14	RESERVED		R	0x0
13:12	PRI3	Priority 3 - A packet header priority of 0x3 is given this switch queue priority.	RW	0x1
11:10	RESERVED		R	0x0
9:8	PRI2	Priority 2 - A packet header priority of 0x2 is given this switch queue priority.	RW	0x0
7:6	RESERVED		R	0x0
5:4	PRI1	Priority 1 - A packet header priority of 0x1 is given this switch queue priority.	RW	0x0
3:2	RESERVED		R	0x0
1:0	PRI0	Priority 0 - A packet header priority of 0x0 is given this switch queue priority.	RW	0x1

Table 18-869. Register Call Summary for Register P0_TX_PRI_MAP

Gigabit Ethernet Switch (GMAC_SW)

- PORT Register Summary: [0]

Table 18-870. P0_CPDMA_TX_PRI_MAP

Address Offset	0x0000 001C		
Physical Address	0x4848 411C	**Instance**	PORT
Description	CPSW CPDMA TX (PORT 0 RX) packet priority to header priority		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	PRI7			RESERVED	PRI6			RESERVED	PRI5			RESERVED	PRI4			RESERVED	PRI3			RESERVED	PRI2			RESERVED	PRI1			RESERVED	PRI0		

Bits	Field Name	Description	Type	Reset
31	RESERVED		R	0x0
30:28	PRI7	Priority 7 - A packet pri of 0x7 is mapped (changed) to this header packet priority.	RW	0x7
27	RESERVED		R	0x0
26:24	PRI6	Priority 6 - A packet pri of 0x6 is mapped (changed) to this header packet priority.	RW	0x6
23	RESERVED		R	0x0
22:20	PRI5	Priority 5 - A packet pri of 0x5 is mapped (changed) to this header packet priority.	RW	0x5
19	RESERVED		R	0x0
18:16	PRI4	Priority 4 - A packet pri of 0x4 is mapped (changed) to this header packet priority.	RW	0x4
15	RESERVED		R	0x0
14:12	PRI3	Priority 3 - A packet pri of 0x3 is mapped (changed) to this header packet priority.	RW	0x3
11	RESERVED		R	0x0
10:8	PRI2	Priority 2 - A packet pri of 0x2 is mapped (changed) to this header packet priority.	RW	0x2
7	RESERVED		R	0x0
6:4	PRI1	Priority 1 - A packet pri of 0x1 is mapped (changed) to this header packet priority.	RW	0x1
3	RESERVED		R	0x0
2:0	PRI0	Priority 0 - A packet pri of 0x0 is mapped (changed) to this header packet priority.	RW	0x0

Table 18-871. Register Call Summary for Register P0_CPDMA_TX_PRI_MAP

Gigabit Ethernet Switch (GMAC_SW)

- PORT Register Summary: [0]

Table 18-872. P0_CPDMA_RX_CH_MAP

Address Offset	0x0000 0020		
Physical Address	0x4848 4120	**Instance**	PORT
Description	CPSW CPDMA RX (PORT 0 TX) switch priority to DMA channel		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	P2_PRI3			RESERVED	P2_PRI2			RESERVED	P2_PRI1			RESERVED	P2_PRI0			RESERVED	P1_PRI3			RESERVED	P1_PRI2			RESERVED	P1_PRI1			RESERVED	P1_PRI0		

Bits	Field Name	Description	Type	Reset
31	RESERVED		R	0x0
30:28	P2_PRI3	Port 2 Priority 3 packets go to this CPDMA Rx Channel	RW	0x0
27	RESERVED		R	0x0
26:24	P2_PRI2	Port 2 Priority 2 packets go to this CPDMA Rx Channel	RW	0x0
23	RESERVED		R	0x0
22:20	P2_PRI1	Port 2 Priority 1 packets go to this CPDMA Rx Channel	RW	0x0
19	RESERVED		R	0x0
18:16	P2_PRI0	Port 2 Priority 0 packets go to this CPDMA Rx Channel	RW	0x0
15	RESERVED		R	0x0
14:12	P1_PRI3	Port 1 Priority 3 packets go to this CPDMA Rx Channel	RW	0x0
11	RESERVED		R	0x0
10:8	P1_PRI2	Port 1 Priority 2 packets go to this CPDMA Rx Channel	RW	0x0
7	RESERVED		R	0x0
6:4	P1_PRI1	Port 1 Priority 1 packets go to this CPDMA Rx Channel	RW	0x0
3	RESERVED		R	0x0
2:0	P1_PRI0	Port 1 Priority 0 packets go to this CPDMA Rx Channel	RW	0x0

Table 18-873. Register Call Summary for Register P0_CPDMA_RX_CH_MAP

Gigabit Ethernet Switch (GMAC_SW)

- Address Lookup Engine (ALE): [0]
- PORT Register Summary: [1]

Table 18-874. P0_RX_DSCP_PRI_MAP0

Address Offset	0x0000 0030		
Physical Address	0x4848 4130	**Instance**	PORT
Description	CPSW PORT 0 RX DSCP priority to RX packet mapping reg 0		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	PRI7			RESERVED	PRI6			RESERVED	PRI5			RESERVED	PRI4			RESERVED	PRI3			RESERVED	PRI2			RESERVED	PRI1			RESERVED	PRI0		

Bits	Field Name	Description	Type	Reset
31	RESERVED		R	0x0
30:28	PRI7	Priority 7 - A packet TOS of 0d7 is mapped to this received packet priority.	RW	0x0
27	RESERVED		R	0x0
26:24	PRI6	Priority 6 - A packet TOS of 0d6 is mapped to this received packet priority.	RW	0x0
23	RESERVED		R	0x0
22:20	PRI5	Priority 5 - A packet TOS of 0d5 is mapped to this received packet priority.	RW	0x0
19	RESERVED		R	0x0
18:16	PRI4	Priority 4 - A packet TOS of 0d4 is mapped to this received packet priority.	RW	0x0
15	RESERVED		R	0x0
14:12	PRI3	Priority 3 - A packet TOS of 0d3 is mapped to this received packet priority.	RW	0x0
11	RESERVED		R	0x0
10:8	PRI2	Priority 2 - A packet TOS of 0d2 is mapped to this received packet priority.	RW	0x0
7	RESERVED		R	0x0
6:4	PRI1	Priority 1 - A packet TOS of 0d1 is mapped to this received packet priority.	RW	0x0
3	RESERVED		R	0x0
2:0	PRI0	Priority 0 - A packet TOS of 0d0 is mapped to this received packet priority.	RW	0x0

Table 18-875. Register Call Summary for Register P0_RX_DSCP_PRI_MAP0

Gigabit Ethernet Switch (GMAC_SW)
- PORT Register Summary: [0]

Table 18-876. P0_RX_DSCP_PRI_MAP1

Address Offset	0x0000 0034		
Physical Address	0x4848 4134	**Instance**	PORT
Description	CPSW PORT 0 RX DSCP priority to RX packet mapping reg 1		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	PRI15			RESERVED	PRI14			RESERVED	PRI13			RESERVED	PRI12			RESERVED	PRI11			RESERVED	PRI10			RESERVED	PRI9			RESERVED	PRI8		

Bits	Field Name	Description	Type	Reset
31	RESERVED		R	0x0
30:28	PRI15	Priority 15 - A packet TOS of 0d15 is mapped to this received packet priority.	RW	0x0
27	RESERVED		R	0x0
26:24	PRI14	Priority 14 - A packet TOS of 0d14 is mapped to this received packet priority.	RW	0x0
23	RESERVED		R	0x0
22:20	PRI13	Priority 13 - A packet TOS of 0d13 is mapped to this received packet priority.	RW	0x0
19	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
18:16	PRI12	Priority 12 - A packet TOS of 0d12 is mapped to this received packet priority.	RW	0x0
15	RESERVED		R	0x0
14:12	PRI11	Priority 11 - A packet TOS of 0d11 is mapped to this received packet priority.	RW	0x0
11	RESERVED		R	0x0
10:8	PRI10	Priority 10 - A packet TOS of 0d10 is mapped to this received packet priority.	RW	0x0
7	RESERVED		R	0x0
6:4	PRI9	Priority 9 - A packet TOS of 0d9 is mapped to this received packet priority.	RW	0x0
3	RESERVED		R	0x0
2:0	PRI8	Priority 8 - A packet TOS of 0d8 is mapped to this received packet priority.	RW	0x0

Table 18-877. Register Call Summary for Register P0_RX_DSCP_PRI_MAP1

Gigabit Ethernet Switch (GMAC_SW)

- PORT Register Summary: [0]

Table 18-878. P0_RX_DSCP_PRI_MAP2

Address Offset	0x0000 0038		
Physical Address	0x4848 4138	**Instance**	PORT
Description	CPSW PORT 0 RX DSCP priority to RX packet mapping reg 2		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	PRI23			RESERVED	PRI22			RESERVED	PRI21			RESERVED	PRI20			RESERVED	PRI19			RESERVED	PRI18			RESERVED	PRI17			RESERVED	PRI16		

Bits	Field Name	Description	Type	Reset
31	RESERVED		R	0x0
30:28	PRI23	Priority 23 - A packet TOS of 0d23 is mapped to this received packet priority.	RW	0x0
27	RESERVED		R	0x0
26:24	PRI22	Priority 22 - A packet TOS of 0d22 is mapped to this received packet priority.	RW	0x0
23	RESERVED		R	0x0
22:20	PRI21	Priority 21 - A packet TOS of 0d21 is mapped to this received packet priority.	RW	0x0
19	RESERVED		R	0x0
18:16	PRI20	Priority 20 - A packet TOS of 0d20 is mapped to this received packet priority.	RW	0x0
15	RESERVED		R	0x0
14:12	PRI19	Priority 19 - A packet TOS of 0d19 is mapped to this received packet priority.	RW	0x0
11	RESERVED		R	0x0
10:8	PRI18	Priority 18 - A packet TOS of 0d18 is mapped to this received packet priority.	RW	0x0
7	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
6:4	PRI17	Priority 17 - A packet TOS of 0d17 is mapped to this received packet priority.	RW	0x0
3	RESERVED		R	0x0
2:0	PRI16	Priority 16 - A packet TOS of 0d16 is mapped to this received packet priority.	RW	0x0

Table 18-879. Register Call Summary for Register P0_RX_DSCP_PRI_MAP2

Gigabit Ethernet Switch (GMAC_SW)
- PORT Register Summary: [0]

Table 18-880. P0_RX_DSCP_PRI_MAP3

Address Offset	0x0000 003C		
Physical Address	0x4848 413C	**Instance**	PORT
Description	CPSW PORT 0 RX DSCP priority to RX packet mapping reg 3		
Type	RW		

31	30 29 28	27	26 25 24	23	22 21 20	19	18 17 16	15	14 13 12	11	10 9 8	7	6 5 4	3	2 1 0
RESERVED	PRI31	RESERVED	PRI30	RESERVED	PRI29	RESERVED	PRI28	RESERVED	PRI27	RESERVED	PRI26	RESERVED	PRI25	RESERVED	PRI24

Bits	Field Name	Description	Type	Reset
31	RESERVED		R	0x0
30:28	PRI31	Priority 31 - A packet TOS of 0d31 is mapped to this received packet priority.	RW	0x0
27	RESERVED		R	0x0
26:24	PRI30	Priority 30 - A packet TOS of 0d30 is mapped to this received packet priority.	RW	0x0
23	RESERVED		R	0x0
22:20	PRI29	Priority 29 - A packet TOS of 0d39 is mapped to this received packet priority.	RW	0x0
19	RESERVED		R	0x0
18:16	PRI28	Priority 28 - A packet TOS of 0d28 is mapped to this received packet priority.	RW	0x0
15	RESERVED		R	0x0
14:12	PRI27	Priority 27 - A packet TOS of 0d27 is mapped to this received packet priority.	RW	0x0
11	RESERVED		R	0x0
10:8	PRI26	Priority 26 - A packet TOS of 0d26 is mapped to this received packet priority.	RW	0x0
7	RESERVED		R	0x0
6:4	PRI25	Priority 25 - A packet TOS of 0d25 is mapped to this received packet priority.	RW	0x0
3	RESERVED		R	0x0
2:0	PRI24	Priority 24 - A packet TOS of 0d24 is mapped to this received packet priority.	RW	0x0

Table 18-881. Register Call Summary for Register P0_RX_DSCP_PRI_MAP3

Gigabit Ethernet Switch (GMAC_SW)
- PORT Register Summary: [0]

Table 18-882. P0_RX_DSCP_PRI_MAP4

Address Offset	0x0000 0040		
Physical Address	0x4848 4140	**Instance**	PORT
Description	CPSW PORT 0 RX DSCP priority to RX packet mapping reg 4		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	PRI39			RESERVED	PRI38			RESERVED	PRI37			RESERVED	PRI36			RESERVED	PRI35			RESERVED	PRI34			RESERVED	PRI33			RESERVED	PRI32		

Bits	Field Name	Description	Type	Reset
31	RESERVED		R	0x0
30:28	PRI39	Priority 39 - A packet TOS of 0d39 is mapped to this received packet priority.	RW	0x0
27	RESERVED		R	0x0
26:24	PRI38	Priority 38 - A packet TOS of 0d38 is mapped to this received packet priority.	RW	0x0
23	RESERVED		R	0x0
22:20	PRI37	Priority 37 - A packet TOS of 0d37 is mapped to this received packet priority.	RW	0x0
19	RESERVED		R	0x0
18:16	PRI36	Priority 36 - A packet TOS of 0d36 is mapped to this received packet priority.	RW	0x0
15	RESERVED		R	0x0
14:12	PRI35	Priority 35 - A packet TOS of 0d35 is mapped to this received packet priority.	RW	0x0
11	RESERVED		R	0x0
10:8	PRI34	Priority 34 - A packet TOS of 0d34 is mapped to this received packet priority.	RW	0x0
7	RESERVED		R	0x0
6:4	PRI33	Priority 33 - A packet TOS of 0d33 is mapped to this received packet priority.	RW	0x0
3	RESERVED		R	0x0
2:0	PRI32	Priority 32 - A packet TOS of 0d32 is mapped to this received packet priority.	RW	0x0

Table 18-883. Register Call Summary for Register P0_RX_DSCP_PRI_MAP4

Gigabit Ethernet Switch (GMAC_SW)

- PORT Register Summary: [0]

Table 18-884. P0_RX_DSCP_PRI_MAP5

Address Offset	0x0000 0044		
Physical Address	0x4848 4144	**Instance**	PORT
Description	CPSW PORT 0 RX DSCP priority to RX packet mapping reg 5		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	PRI47			RESERVED	PRI46			RESERVED	PRI45			RESERVED	PRI44			RESERVED	PRI43			RESERVED	PRI42			RESERVED	PRI41			RESERVED	PRI40		

Bits	Field Name	Description	Type	Reset
31	RESERVED		R	0x0
30:28	PRI47	Priority 47 - A packet TOS of 0d47 is mapped to this received packet priority.	RW	0x0
27	RESERVED		R	0x0
26:24	PRI46	Priority 46 - A packet TOS of 0d46 is mapped to this received packet priority.	RW	0x0
23	RESERVED		R	0x0
22:20	PRI45	Priority 45 - A packet TOS of 0d45 is mapped to this received packet priority.	RW	0x0
19	RESERVED		R	0x0
18:16	PRI44	Priority 44 - A packet TOS of 0d44 is mapped to this received packet priority.	RW	0x0
15	RESERVED		R	0x0
14:12	PRI43	Priority 43 - A packet TOS of 0d43 is mapped to this received packet priority.	RW	0x0
11	RESERVED		R	0x0
10:8	PRI42	Priority 42 - A packet TOS of 0d42 is mapped to this received packet priority.	RW	0x0
7	RESERVED		R	0x0
6:4	PRI41	Priority 41 - A packet TOS of 0d41 is mapped to this received packet priority.	RW	0x0
3	RESERVED		R	0x0
2:0	PRI40	Priority 40 - A packet TOS of 0d40 is mapped to this received packet priority.	RW	0x0

Table 18-885. Register Call Summary for Register P0_RX_DSCP_PRI_MAP5

Gigabit Ethernet Switch (GMAC_SW)

- PORT Register Summary: [0]

Table 18-886. P0_RX_DSCP_PRI_MAP6

Address Offset	0x0000 0048		
Physical Address	0x4848 4148	**Instance**	PORT
Description	CPSW PORT 0 RX DSCP priority to RX packet mapping reg 6		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	PRI55			RESERVED	PRI54			RESERVED	PRI53			RESERVED	PRI52			RESERVED	PRI51			RESERVED	PRI50			RESERVED	PRI49			RESERVED	PRI48		

Bits	Field Name	Description	Type	Reset
31	RESERVED		R	0x0
30:28	PRI55	Priority 55 - A packet TOS of 0d55 is mapped to this received packet priority.	RW	0x0
27	RESERVED		R	0x0
26:24	PRI54	Priority 54 - A packet TOS of 0d54 is mapped to this received packet priority.	RW	0x0
23	RESERVED		R	0x0
22:20	PRI53	Priority 53 - A packet TOS of 0d53 is mapped to this received packet priority.	RW	0x0
19	RESERVED		R	0x0
18:16	PRI52	Priority 52 - A packet TOS of 0d52 is mapped to this received packet priority.	RW	0x0
15	RESERVED		R	0x0
14:12	PRI51	Priority 51 - A packet TOS of 0d51 is mapped to this received packet priority.	RW	0x0
11	RESERVED		R	0x0
10:8	PRI50	Priority 50 - A packet TOS of 0d50 is mapped to this received packet priority.	RW	0x0
7	RESERVED		R	0x0
6:4	PRI49	Priority 49 - A packet TOS of 0d49 is mapped to this received packet priority.	RW	0x0
3	RESERVED		R	0x0
2:0	PRI48	Priority 48 - A packet TOS of 0d48 is mapped to this received packet priority.	RW	0x0

Table 18-887. Register Call Summary for Register P0_RX_DSCP_PRI_MAP6

Gigabit Ethernet Switch (GMAC_SW)

- PORT Register Summary: [0]

Table 18-888. P0_RX_DSCP_PRI_MAP7

Address Offset	0x0000 004C		
Physical Address	0x4848 414C	**Instance**	PORT
Description	CPSW PORT 0 RX DSCP priority to RX packet mapping reg 7		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	PRI63			RESERVED	PRI62			RESERVED	PRI61			RESERVED	PRI60			RESERVED	PRI59			RESERVED	PRI58			RESERVED	PRI57			RESERVED	PRI56		

Bits	Field Name	Description	Type	Reset
31	RESERVED		R	0x0
30:28	PRI63	Priority 63 - A packet TOS of 0d63 is mapped to this received packet priority.	RW	0x0
27	RESERVED		R	0x0
26:24	PRI62	Priority 62 - A packet TOS of 0d62 is mapped to this received packet priority.	RW	0x0
23	RESERVED		R	0x0
22:20	PRI61	Priority 61 - A packet TOS of 0d61 is mapped to this received packet priority.	RW	0x0
19	RESERVED		R	0x0
18:16	PRI60	Priority 60 - A packet TOS of 0d60 is mapped to this received packet priority.	RW	0x0
15	RESERVED		R	0x0
14:12	PRI59	Priority 59 - A packet TOS of 0d59 is mapped to this received packet priority.	RW	0x0
11	RESERVED		R	0x0
10:8	PRI58	Priority 58 - A packet TOS of 0d58 is mapped to this received packet priority.	RW	0x0
7	RESERVED		R	0x0
6:4	PRI57	Priority 57 - A packet TOS of 0d57 is mapped to this received packet priority.	RW	0x0
3	RESERVED		R	0x0
2:0	PRI56	Priority 56 - A packet TOS of 0d56 is mapped to this received packet priority.	RW	0x0

Table 18-889. Register Call Summary for Register P0_RX_DSCP_PRI_MAP7

Gigabit Ethernet Switch (GMAC_SW)

- PORT Register Summary: [0]

Table 18-890. P0_IDLE2LPI

Address Offset	0x0000 0050		
Physical Address	0x4848 4150	**Instance**	PORT
Description	Port 0 EEE Idle to LPI Counter Load Value Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								P0_IDLE2LPI																							

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19:0	P0_IDLE2LPI	Port 0 EEE Idle to LPI counter load value - After CLKSTOP_REQ is asserted, this value is loaded into the port 0 idle to LPI counter on each clock that the port 0 transmit is not idle. Port 0 enters the transmit LPI state when this counter decrements to zero. This counter decrements each time the EEE prescale counter decrements to zero.	RW	0x0

Table 18-891. Register Call Summary for Register P0_IDLE2LPI

Gigabit Ethernet Switch (GMAC_SW)

- Energy Efficient Ethernet Support (802.3az): [0]
- PORT Register Summary: [1]
- PORT Register Description: [2]

Table 18-892. P0_LPI2WAKE

Address Offset	0x0000 0054		
Physical Address	0x4848 4154	**Instance**	PORT
Description	Port 0 EEE LPI to Wake Counter Load Value Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								P0_LPI2WAKE																							

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19:0	P0_LPI2WAKE	Port 0 EEE LPI to wake counter load value – When the port is in the transmit LPI state and the CLKSTOP_REQ signal is deasserted, this value is loaded into the port 0 LPI to wake counter. Transmit packet operations may begin (resume) when the LPI to wake count decrements to zero (on the pre-scale count). This is the time that the CPDMA transmit must wait before transmit (switch ingress) packet operations begin (resume after wakeup).	RW	0x0

Table 18-893. Register Call Summary for Register P0_LPI2WAKE

Gigabit Ethernet Switch (GMAC_SW)

- Energy Efficient Ethernet Support (802.3az): [0]
- PORT Register Summary: [1]
- PORT Register Description: [2]

Table 18-894. P1_CONTROL

Address Offset	0x0000 0100		
Physical Address	0x4848 4200	**Instance**	PORT
Description	CPSW PORT 1 control register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED						P1_TX_CLKSTOP_EN	P1_PASS_PRI_TAGGED	RESERVED		P1_VLAN_LTYPE2_EN	P1_VLAN_LTYPE1_EN	RESERVED			P1_DSCP_PRI_EN	P1_TS_107	P1_TS_320	P1_TS_319	P1_TS_132	P1_TS_131	P1_TS_130	P1_TS_129	P1_TS_TTL_NONZERO	P1_TS_UNI_EN	P1_TS_ANNEX_F_EN	P1_TS_ANNEX_E_EN	P1_TS_ANNEX_D_EN	P1_TS_LTYPE2_EN	P1_TS_LTYPE1_EN	P1_TS_TX_EN	P1_TS_RX_EN

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	P1_TX_CLKSTOP_EN	Port 1 Transmit clockstop enable 0 – RGMII transmit clockstop not enabled 1 – RGMII transmit clockstop enabled. The transmit clock will be stopped after the LPI state is entered (and indicated to the CPRGMII) and the P1_Idle2LPI time is counted (counter value reused). The P1_Idle2LPI counter value must be greater than 9 transmit clocks (slowest clock)	RW	0x0
24	P1_PASS_PRI_TAGGED	Port 1 Pass Priority Tagged 0 - Priority tagged packets have the zero VID replaced with the input port P1_PORT_VLAN [11:0] 1 - Priority tagged packets are processed unchanged.	RW	0x0
23:22	RESERVED		R	0x0
21	P1_VLAN_LTYPE2_EN	Port 1 VLAN LTYPE 2 enable 0 - disabled 1 - VLAN LTYPE2 enabled on transmit and receive	RW	0x0
20	P1_VLAN_LTYPE1_EN	Port 1 VLAN LTYPE 1 enable 0 - disabled 1 - VLAN LTYPE1 enabled on transmit and receive	RW	0x0
19:17	RESERVED		R	0x0
16	P1_DSCP_PRI_EN	Port 1 DSCP Priority Enable 0 - DSCP priority disabled 1 - DSCP priority enabled. All non-tagged IPV4 packets have their received packet priority determined by mapping the 6 TOS bits through the port DSCP priority mapping registers.	RW	0x0
15	P1_TS_107	Port 1 Time Sync Destination IP Address 107 enable 0 – disabled 1 – destination IP address (dec) 224.0.0.107 is enabled.	RW	0x0
14	P1_TS_320	Port 1 Time Sync Destination Port Number 320 enable 0 - disabled 1 - Annex D (UDP/IPv4) time sync packet destination port number 320 (decimal) is enabled.	RW	0x0
13	P1_TS_319	Port 1 Time Sync Destination Port Number 319 enable 0 - disabled 1 - Annex D (UDP/IPv4) time sync packet destination port number 319 (decimal) is enabled.	RW	0x0
12	P1_TS_132	Port 1 Time Sync Destination IP Address 132 enable 0 - disabled 1 - Annex D (UDP/IPv4) time sync packet destination IP address number 132 (decimal) is enabled.	RW	0x0

Bits	Field Name	Description	Type	Reset
11	P1_TS_131	Port 1 Time Sync Destination IP Address 131 enable 0 - disabled 1 - Annex D (UDP/IPv4) time sync packet destination IP address number 131 (decimal) is enabled.	RW	0x0
10	P1_TS_130	Port 1 Time Sync Destination IP Address 130 enable 0 - disabled 1 - Annex D (UDP/IPv4) time sync packet destination IP address number 130 (decimal) is enabled.	RW	0x0
9	P1_TS_129	Port 1 Time Sync Destination IP Address 129 enable 0 - disabled 1 - Annex D (UDP/IPv4) time sync packet destination IP address number 129 (decimal) is enabled.	RW	0x0
8	P1_TS_TTL_NONZERO	Port 1 Time Sync Time To Live Non-zero enable. 0 = TTL must be zero. 1 = TTL may be any value.	RW	0x0
7	P1_TS_UNI_EN	Port 1 Time Sync Unicast Enable 0 – Unicast disabled 1 – Unicast enabled	RW	0x0
6	P1_TS_ANNEX_F_EN	Port 1 Time Sync Annex F enable 0 – Annex F disabled 1 – Annex F enabled	RW	0x0
5	P1_TS_ANNEX_E_EN	Port 1 Time Sync Annex E enable 0 – Annex E disabled 1 – Annex E enabled	RW	0x0
4	P1_TS_ANNEX_D_EN	Port 1 Time Sync Annex D enable 0 - Annex D disabled 1 - Annex D enabled	RW	0x0
3	P1_TS_LTYPE2_EN	Port 1 Time Sync LTYPE 2 enable 0 - disabled 1 - enabled	RW	0x0
2	P1_TS_LTYPE1_EN	Port 1 Time Sync LTYPE 1 enable 0 - disabled 1 - enabled	RW	0x0
1	P1_TS_TX_EN	Port 1 Time Sync Transmit Enable 0 - disabled 1 - enabled	RW	0x0
0	P1_TS_RX_EN	Port 1 Time Sync Receive Enable 0 - Port 1 Receive Time Sync disabled 1 - Port 1 Receive Time Sync enabled	RW	0x0

Table 18-895. Register Call Summary for Register P1_CONTROL

Gigabit Ethernet Switch (GMAC_SW)
- PORT Register Summary: [0]

Table 18-896. P1_MAX_BLKS

Address Offset	0x0000 0108		
Physical Address	0x4848 4208	**Instance**	PORT
Description	CPSW PORT 1 maximum FIFO blocks register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								P1_TX_MAX_BLKS				P1_RX_MAX_BLKS			

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8:4	P1_TX_MAX_BLKS	Transmit FIFO Maximum Blocks - This value is the maximum number of 1k memory blocks that may be allocated to the FIFO's logical transmit priority queues. 0x11 is the recommended value of P1_TX_MAX_BLKS unless the port is in fullduplex flow control mode. In flow control mode, the P1_RX_MAX_BLKS will need to increase in order to accept the required run out in fullduplex mode. This value will need to decrease by the amount of increase in P1_RX_MAX_BLKS. 0xE is the minimum value for P1_TX_MAX_BLKS.	RW	0x11
3:0	P1_RX_MAX_BLKS	Receive FIFO Maximum Blocks - This value is the maximum number of 1k memory blocks that may be allocated to the FIFO's logical receive queue. This value must be greater than or equal to 0x3. It should be increased In fullduplex flow control mode to 0x5 or 0x6 depending on the required runout space. The P1_TX_MAX_BLKS value must be decreased by the amount of increase in P1_RX_MAX_BLKS. 0x6 is the maximum value for P1_RX_MAX_BLKS.	RW	0x3

Table 18-897. Register Call Summary for Register P1_MAX_BLKS

Gigabit Ethernet Switch (GMAC_SW)

- PORT Register Summary: [0]

Table 18-898. P1_BLK_CNT

Address Offset	0x0000 010C		
Physical Address	0x4848 420C	**Instance**	PORT
Description	CPSW PORT 1 FIFO block usage count (read only)		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								P1_TX_BLK_CNT				P1_RX_BLK_CNT			

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8:4	P1_TX_BLK_CNT	Port 1 Transmit Block Count Usage - This value is the number of blocks allocated to the FIFO logical transmit queues.	R	0x4
3:0	P1_RX_BLK_CNT	Port 1 Receive Block Count Usage - This value is the number of blocks allocated to the FIFO logical receive queues.	R	0x1

Table 18-899. Register Call Summary for Register P1_BLK_CNT

Gigabit Ethernet Switch (GMAC_SW)

- PORT Register Summary: [0]

Table 18-900. P1_TX_IN_CTL

Address Offset	0x0000 0110		
Physical Address	0x4848 4210	**Instance**	PORT
Description	CPSW PORT 1 transmit FIFO control		
Type	RW		

31 30 29 28	27 26 25 24	23 22 21 20	19 18	17 16	15 14 13 12	11 10	9 8	7 6 5 4 3 2 1 0
RESERVED	HOST_BLKS_REM	TX_RATE_EN	RESERVED	TX_IN_SEL	TX_BLKS_REM	RESERVED		TX_PRI_WDS

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:24	HOST_BLKS_REM	Transmit FIFO Blocks that must be free before a non rate-limited CPDMA channel can begin sending a packet to the FIFO.	RW	0x8
23:20	TX_RATE_EN	Transmit FIFO Input Rate Enable	RW	0x0
19:18	RESERVED		R	0x0
17:16	TX_IN_SEL	Transmit FIFO Input Queue Type Select 0x0 - Normal priority mode 0x1 - reserved 0x2 - Rate Limit mode 0x3 - reserved	RW	0x0
15:12	TX_BLKS_REM	Transmit FIFO Input blocks to subtract on non rate-limited traffic in rate limit mode.	RW	0x4
11:10	RESERVED		R	0x0
9:0	TX_PRI_WDS	Transmit FIFO Words in queue	RW	0xc0

Table 18-901. Register Call Summary for Register P1_TX_IN_CTL

Gigabit Ethernet Switch (GMAC_SW)

- Audio Video Bridging: [0][1]
- PORT Register Summary: [2]

Table 18-902. P1_PORT_VLAN

Address Offset	0x0000 0114		
Physical Address	0x4848 4214	**Instance**	PORT
Description	CPSW PORT 1 VLAN register		
Type	RW		

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13	12	11 10 9 8	7 6 5 4 3 2 1 0
RESERVED		PORT_PRI	PORT_CFI	PORT_VID	

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0
15:13	PORT_PRI	Port VLAN Priority (7 is highest priority)	RW	0x0
12	PORT_CFI	Port CFI bit	RW	0x0

Bits	Field Name	Description	Type	Reset
11:0	PORT_VID	Port VLAN ID	RW	0x0

Table 18-903. Register Call Summary for Register P1_PORT_VLAN

Gigabit Ethernet Switch (GMAC_SW)

- PORT Register Summary: [0]
- PORT Register Description: [1]

Table 18-904. P1_TX_PRI_MAP

Address Offset	0x0000 0118		
Physical Address	0x4848 4218	**Instance**	PORT
Description	CPSW PORT 1 TX header priority to switch priority mapping register		
Type	RW		

31 30	29 28	27 26	25 24	23 22	21 20	19 18	17 16	15 14	13 12	11 10	9 8	7 6	5 4	3 2	1 0
RESERVED	PRI7	RESERVED	PRI6	RESERVED	PRI5	RESERVED	PRI4	RESERVED	PRI3	RESERVED	PRI2	RESERVED	PRI1	RESERVED	PRI0

Bits	Field Name	Description	Type	Reset
31:30	RESERVED		R	0x0
29:28	PRI7	Priority 7 - A packet header priority of 0x7 is given this switch queue priority	RW	0x3
27:26	RESERVED		R	0x0
25:24	PRI6	Priority 6 - A packet header priority of 0x6 is given this switch queue priority	RW	0x3
23:22	RESERVED		R	0x0
21:20	PRI5	Priority 5 - A packet header priority of 0x5 is given this switch queue priority	RW	0x2
19:18	RESERVED		R	0x0
17:16	PRI4	Priority 4 - A packet header priority of 0x4 is given this switch queue priority	RW	0x2
15:14	RESERVED		R	0x0
13:12	PRI3	Priority 3 - A packet header priority of 0x3 is given this switch queue priority	RW	0x1
11:10	RESERVED		R	0x0
9:8	PRI2	Priority 2 - A packet header priority of 0x2 is given this switch queue priority	RW	0x0
7:6	RESERVED		R	0x0
5:4	PRI1	Priority 1 - A packet header priority of 0x1 is given this switch queue priority	RW	0x0
3:2	RESERVED		R	0x0
1:0	PRI0	Priority 0 - A packet header priority of 0x0 is given this switch queue priority	RW	0x1

Table 18-905. Register Call Summary for Register P1_TX_PRI_MAP

Gigabit Ethernet Switch (GMAC_SW)

- PORT Register Summary: [0]

Table 18-906. P1_TS_SEQ_MTYPE

Address Offset	0x0000 011C		
Physical Address	0x4848 421C	**Instance**	PORT
Description	CPSW PORT 1 time sync sequence ID offset and message type.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								P1_TS_SEQ_ID_OFFSET								P1_TS_MSG_TYPE_EN															

Bits	Field Name	Description	Type	Reset
31:22	RESERVED		R	0x0
21:16	P1_TS_SEQ_ID_OFFSET	Port 1 Time Sync Sequence ID Offset This is the number of octets that the sequence ID is offset in the tx and rx time sync message header. The minimum value is 6.	RW	0x1E
15:0	P1_TS_MSG_TYPE_EN	Port 1 Time Sync Message Type Enable - Each bit in this field enables the corresponding message type in receive and transmit time sync messages (Bit 0 enables message type 0 etc.).	RW	0x0

Table 18-907. Register Call Summary for Register P1_TS_SEQ_MTYPE

Gigabit Ethernet Switch (GMAC_SW)

- PORT Register Summary: [0]

Table 18-908. P1_SA_LO

Address Offset	0x0000 0120		
Physical Address	0x4848 4220	**Instance**	PORT
Description	CPSW CPGMAC_SL1 source address low register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																MACSRCADDR_7_0								MACSRCADDR_15_8							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0
15:8	MACSRCADDR_7_0	Source Address Lower 8 bits (byte 0)	RW	0x0
7:0	MACSRCADDR_15_8	Source Address bits 15:8 (byte 1)	RW	0x0

Table 18-909. Register Call Summary for Register P1_SA_LO

Gigabit Ethernet Switch (GMAC_SW)

- PORT Register Summary: [0]

Table 18-910. P1_SA_HI

Address Offset	0x0000 0124		
Physical Address	0x4848 4224	**Instance**	PORT
Description	CPSW CPGMAC_SL1 source address high register		
Type	RW		

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14 13 12 11 10 9 8	7 6 5 4 3 2 1 0
MACSRCADDR_23_16	MACSRCADDR_31_24	MACSRCADDR_39_32	MACSRCADDR_47_40

Bits	Field Name	Description	Type	Reset
31:24	MACSRCADDR_23_16	Source Address bits 23:16 (byte 2)	RW	0x0
23:16	MACSRCADDR_31_24	Source Address bits 31:24 (byte 3)	RW	0x0
15:8	MACSRCADDR_39_32	Source Address bits 39:32 (byte 4)	RW	0x0
7:0	MACSRCADDR_47_40	Source Address bits 47:40 (byte 5)	RW	0x0

Table 18-911. Register Call Summary for Register P1_SA_HI

Gigabit Ethernet Switch (GMAC_SW)
- PORT Register Summary: [0]

Table 18-912. P1_SEND_PERCENT

Address Offset	0x0000 0128		
Physical Address	0x4848 4228	**Instance**	PORT
Description	CPSW PORT 1 transmit queue send percentages		
Type	RW		

31 30 29 28 27 26 25 24	23 22 21 20 19 18 17 16	15	14 13 12 11 10 9 8	7	6 5 4 3 2 1 0
RESERVED	PRI3_SEND_PERCENT	RESERVED	PRI2_SEND_PERCENT	RESERVED	PRI1_SEND_PERCENT

Bits	Field Name	Description	Type	Reset
31:23	RESERVED		R	0x0
22:16	PRI3_SEND_PERCENT	Priority 3 Transmit Percentage - This percentage value is sent from FIFO priority 3 (maximum) when CPSW_PTYPE[18] P1_PRI3_SHAPE_EN is set (queue shaping enabled). This is the percentage of the wire that packets from priority 3 receive (which includes interpacket gap and preamble bytes). If shaping is enabled on this queue then this value must be between zero and 0d100 (not inclusive).	RW	0x0
15	RESERVED		R	0x0
14:8	PRI2_SEND_PERCENT	Priority 2 Transmit Percentage - This percentage value is sent from FIFO priority 2 (maximum) when CPSW_PTYPE[17] P1_PRI2_SHAPE_EN is set (queue shaping enabled). This is the percentage of the wire that packets from priority 2 receive (which includes interpacket gap and preamble bytes). If shaping is enabled on this queue then this value must be between zero and 0d100 (not inclusive).	RW	0x0
7	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
6:0	PRI1_SEND_PERCENT	Priority 1 Transmit Percentage - This percentage value is sent from FIFO priority 1 (maximum) when the CPSW_PTYPE[16] P1_PRI1_SHAPE_EN is set (queue shaping enabled). This is the percentage of the wire that packets from priority 1 receive (which includes interpacket gap and preamble bytes). If shaping is enabled on this queue then this value must be between zero and 0d100 (not inclusive).	RW	0x0

Table 18-913. Register Call Summary for Register P1_SEND_PERCENT

Gigabit Ethernet Switch (GMAC_SW)
- Audio Video Bridging: [0]
- PORT Register Summary: [1]

Table 18-914. P1_RX_DSCP_PRI_MAP0

Address Offset	0x0000 0130		
Physical Address	0x4848 4230	**Instance**	PORT
Description	CPSW PORT 1 RX DSCP priority to RX packet mapping reg 0		
Type	RW		

31	30 29 28	27	26 25 24	23	22 21 20	19	18 17 16	15	14 13 12	11	10 9 8	7	6 5 4	3	2 1 0
RESERVED	PRI7	RESERVED	PRI6	RESERVED	PRI5	RESERVED	PRI4	RESERVED	PRI3	RESERVED	PRI2	RESERVED	PRI1	RESERVED	PRI0

Bits	Field Name	Description	Type	Reset
31	RESERVED		R	0x0
30:28	PRI7	Priority 7 - A packet TOS of 0d7 is mapped to this received packet priority.	RW	0x0
27	RESERVED		R	0x0
26:24	PRI6	Priority 6 - A packet TOS of 0d6 is mapped to this received packet priority.	RW	0x0
23	RESERVED		R	0x0
22:20	PRI5	Priority 5 - A packet TOS of 0d5 is mapped to this received packet priority.	RW	0x0
19	RESERVED		R	0x0
18:16	PRI4	Priority 4 - A packet TOS of 0d4 is mapped to this received packet priority.	RW	0x0
15	RESERVED		R	0x0
14:12	PRI3	Priority 3 - A packet TOS of 0d3 is mapped to this received packet priority.	RW	0x0
11	RESERVED		R	0x0
10:8	PRI2	Priority 2 - A packet TOS of 0d2 is mapped to this received packet priority.	RW	0x0
7	RESERVED		R	0x0
6:4	PRI1	Priority 1 - A packet TOS of 0d1 is mapped to this received packet priority.	RW	0x0
3	RESERVED		R	0x0
2:0	PRI0	Priority 0 - A packet TOS of 0d0 is mapped to this received packet priority.	RW	0x0

Table 18-915. Register Call Summary for Register P1_RX_DSCP_PRI_MAP0

Gigabit Ethernet Switch (GMAC_SW)

- PORT Register Summary: [0]

Table 18-916. P1_RX_DSCP_PRI_MAP1

Address Offset	0x0000 0134		
Physical Address	0x4848 4234	**Instance**	PORT
Description	CPSW PORT 1 RX DSCP priority to RX packet mapping reg 1		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	PRI15			RESERVED	PRI14			RESERVED	PRI13			RESERVED	PRI12			RESERVED	PRI11			RESERVED	PRI10			RESERVED	PRI9			RESERVED	PRI8		

Bits	Field Name	Description	Type	Reset
31	RESERVED		R	0x0
30:28	PRI15	Priority 15 - A packet TOS of 0d15 is mapped to this received packet priority.	RW	0x0
27	RESERVED		R	0x0
26:24	PRI14	Priority 14 - A packet TOS of 0d14 is mapped to this received packet priority.	RW	0x0
23	RESERVED		R	0x0
22:20	PRI13	Priority 13 - A packet TOS of 0d13 is mapped to this received packet priority.	RW	0x0
19	RESERVED		R	0x0
18:16	PRI12	Priority 12 - A packet TOS of 0d12 is mapped to this received packet priority.	RW	0x0
15	RESERVED		R	0x0
14:12	PRI11	Priority 11 - A packet TOS of 0d11 is mapped to this received packet priority.	RW	0x0
11	RESERVED		R	0x0
10:8	PRI10	Priority 10 - A packet TOS of 0d10 is mapped to this received packet priority.	RW	0x0
7	RESERVED		R	0x0
6:4	PRI9	Priority 9 - A packet TOS of 0d9 is mapped to this received packet priority.	RW	0x0
3	RESERVED		R	0x0
2:0	PRI8	Priority 8 - A packet TOS of 0d8 is mapped to this received packet priority.	RW	0x0

Table 18-917. Register Call Summary for Register P1_RX_DSCP_PRI_MAP1

Gigabit Ethernet Switch (GMAC_SW)

- PORT Register Summary: [0]

Table 18-918. P1_RX_DSCP_PRI_MAP2

Address Offset	0x0000 0138		
Physical Address	0x4848 4238	**Instance**	PORT
Description	CPSW PORT 1 RX DSCP priority to RX packet mapping reg 2		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	PRI23			RESERVED	PRI22			RESERVED	PRI21			RESERVED	PRI20			RESERVED	PRI19			RESERVED	PRI18			RESERVED	PRI17			RESERVED	PRI16		

Bits	Field Name	Description	Type	Reset
31	RESERVED		R	0x0
30:28	PRI23	Priority 23 - A packet TOS of 0d23 is mapped to this received packet priority.	RW	0x0
27	RESERVED		R	0x0
26:24	PRI22	Priority 22 - A packet TOS of 0d22 is mapped to this received packet priority.	RW	0x0
23	RESERVED		R	0x0
22:20	PRI21	Priority 21 - A packet TOS of 0d21 is mapped to this received packet priority.	RW	0x0
19	RESERVED		R	0x0
18:16	PRI20	Priority 20 - A packet TOS of 0d20 is mapped to this received packet priority.	RW	0x0
15	RESERVED		R	0x0
14:12	PRI19	Priority 19 - A packet TOS of 0d19 is mapped to this received packet priority.	RW	0x0
11	RESERVED		R	0x0
10:8	PRI18	Priority 18 - A packet TOS of 0d18 is mapped to this received packet priority.	RW	0x0
7	RESERVED		R	0x0
6:4	PRI17	Priority 17 - A packet TOS of 0d17 is mapped to this received packet priority.	RW	0x0
3	RESERVED		R	0x0
2:0	PRI16	Priority 16 - A packet TOS of 0d16 is mapped to this received packet priority.	RW	0x0

Table 18-919. Register Call Summary for Register P1_RX_DSCP_PRI_MAP2

Gigabit Ethernet Switch (GMAC_SW)

- PORT Register Summary: [0]

Table 18-920. P1_RX_DSCP_PRI_MAP3

Address Offset	0x0000 013C		
Physical Address	0x4848 423C	**Instance**	PORT
Description	CPSW PORT 1 RX DSCP priority to RX packet mapping reg 3		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	PRI31			RESERVED	PRI30			RESERVED	PRI29			RESERVED	PRI28			RESERVED	PRI27			RESERVED	PRI26			RESERVED	PRI25			RESERVED	PRI24		

Bits	Field Name	Description	Type	Reset
31	RESERVED		R	0x0
30:28	PRI31	Priority 31 - A packet TOS of 0d31 is mapped to this received packet priority.	RW	0x0
27	RESERVED		R	0x0
26:24	PRI30	Priority 30 - A packet TOS of 0d30 is mapped to this received packet priority.	RW	0x0
23	RESERVED		R	0x0
22:20	PRI29	Priority 29 - A packet TOS of 0d39 is mapped to this received packet priority.	RW	0x0
19	RESERVED		R	0x0
18:16	PRI28	Priority 28 - A packet TOS of 0d28 is mapped to this received packet priority.	RW	0x0
15	RESERVED		R	0x0
14:12	PRI27	Priority 27 - A packet TOS of 0d27 is mapped to this received packet priority.	RW	0x0
11	RESERVED		R	0x0
10:8	PRI26	Priority 26 - A packet TOS of 0d26 is mapped to this received packet priority.	RW	0x0
7	RESERVED		R	0x0
6:4	PRI25	Priority 25 - A packet TOS of 0d25 is mapped to this received packet priority.	RW	0x0
3	RESERVED		R	0x0
2:0	PRI24	Priority 24 - A packet TOS of 0d24 is mapped to this received packet priority.	RW	0x0

Table 18-921. Register Call Summary for Register P1_RX_DSCP_PRI_MAP3

Gigabit Ethernet Switch (GMAC_SW)

- PORT Register Summary: [0]

Table 18-922. P1_RX_DSCP_PRI_MAP4

Address Offset	0x0000 0140		
Physical Address	0x4848 4240	**Instance**	PORT
Description	CPSW PORT 1 RX DSCP priority to RX packet mapping reg 4		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	PRI39			RESERVED	PRI38			RESERVED	PRI37			RESERVED	PRI36			RESERVED	PRI35			RESERVED	PRI34			RESERVED	PRI33			RESERVED	PRI32		

Bits	Field Name	Description	Type	Reset
31	RESERVED		R	0x0
30:28	PRI39	Priority 39 - A packet TOS of 0d39 is mapped to this received packet priority.	RW	0x0
27	RESERVED		R	0x0
26:24	PRI38	Priority 38 - A packet TOS of 0d38 is mapped to this received packet priority.	RW	0x0
23	RESERVED		R	0x0
22:20	PRI37	Priority 37 - A packet TOS of 0d37 is mapped to this received packet priority.	RW	0x0
19	RESERVED		R	0x0
18:16	PRI36	Priority 36 - A packet TOS of 0d36 is mapped to this received packet priority.	RW	0x0
15	RESERVED		R	0x0
14:12	PRI35	Priority 35 - A packet TOS of 0d35 is mapped to this received packet priority.	RW	0x0
11	RESERVED		R	0x0
10:8	PRI34	Priority 34 - A packet TOS of 0d34 is mapped to this received packet priority.	RW	0x0
7	RESERVED		R	0x0
6:4	PRI33	Priority 33 - A packet TOS of 0d33 is mapped to this received packet priority.	RW	0x0
3	RESERVED		R	0x0
2:0	PRI32	Priority 32 - A packet TOS of 0d32 is mapped to this received packet priority.	RW	0x0

Table 18-923. Register Call Summary for Register P1_RX_DSCP_PRI_MAP4

Gigabit Ethernet Switch (GMAC_SW)

- PORT Register Summary: [0]

Table 18-924. P1_RX_DSCP_PRI_MAP5

Address Offset	0x0000 0144		
Physical Address	0x4848 4244	**Instance**	PORT
Description	CPSW PORT 1 RX DSCP priority to RX packet mapping reg 5		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	PRI47			RESERVED	PRI46			RESERVED	PRI45			RESERVED	PRI44			RESERVED	PRI43			RESERVED	PRI42			RESERVED	PRI41			RESERVED	PRI40		

Bits	Field Name	Description	Type	Reset
31	RESERVED		R	0x0
30:28	PRI47	Priority 47 - A packet TOS of 0d47 is mapped to this received packet priority.	RW	0x0
27	RESERVED		R	0x0
26:24	PRI46	Priority 46 - A packet TOS of 0d46 is mapped to this received packet priority.	RW	0x0
23	RESERVED		R	0x0
22:20	PRI45	Priority 45 - A packet TOS of 0d45 is mapped to this received packet priority.	RW	0x0
19	RESERVED		R	0x0
18:16	PRI44	Priority 44 - A packet TOS of 0d44 is mapped to this received packet priority.	RW	0x0
15	RESERVED		R	0x0
14:12	PRI43	Priority 43 - A packet TOS of 0d43 is mapped to this received packet priority.	RW	0x0
11	RESERVED		R	0x0
10:8	PRI42	Priority 42 - A packet TOS of 0d42 is mapped to this received packet priority.	RW	0x0
7	RESERVED		R	0x0
6:4	PRI41	Priority 41 - A packet TOS of 0d41 is mapped to this received packet priority.	RW	0x0
3	RESERVED		R	0x0
2:0	PRI40	Priority 40 - A packet TOS of 0d40 is mapped to this received packet priority.	RW	0x0

Table 18-925. Register Call Summary for Register P1_RX_DSCP_PRI_MAP5

Gigabit Ethernet Switch (GMAC_SW)

- PORT Register Summary: [0]

Table 18-926. P1_RX_DSCP_PRI_MAP6

Address Offset	0x0000 0148		
Physical Address	0x4848 4248	**Instance**	PORT
Description	CPSW PORT 1 RX DSCP priority to RX packet mapping reg 6		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	PRI55			RESERVED	PRI54			RESERVED	PRI53			RESERVED	PRI52			RESERVED	PRI51			RESERVED	PRI50			RESERVED	PRI49			RESERVED	PRI48		

Bits	Field Name	Description	Type	Reset
31	RESERVED		R	0x0
30:28	PRI55	Priority 55 - A packet TOS of 0d55 is mapped to this received packet priority.	RW	0x0
27	RESERVED		R	0x0
26:24	PRI54	Priority 54 - A packet TOS of 0d54 is mapped to this received packet priority.	RW	0x0
23	RESERVED		R	0x0
22:20	PRI53	Priority 53 - A packet TOS of 0d53 is mapped to this received packet priority.	RW	0x0
19	RESERVED		R	0x0
18:16	PRI52	Priority 52 - A packet TOS of 0d52 is mapped to this received packet priority.	RW	0x0
15	RESERVED		R	0x0
14:12	PRI51	Priority 51 - A packet TOS of 0d51 is mapped to this received packet priority.	RW	0x0
11	RESERVED		R	0x0
10:8	PRI50	Priority 50 - A packet TOS of 0d50 is mapped to this received packet priority.	RW	0x0
7	RESERVED		R	0x0
6:4	PRI49	Priority 49 - A packet TOS of 0d49 is mapped to this received packet priority.	RW	0x0
3	RESERVED		R	0x0
2:0	PRI48	Priority 48 - A packet TOS of 0d48 is mapped to this received packet priority.	RW	0x0

Table 18-927. Register Call Summary for Register P1_RX_DSCP_PRI_MAP6

Gigabit Ethernet Switch (GMAC_SW)

- PORT Register Summary: [0]

Table 18-928. P1_RX_DSCP_PRI_MAP7

Address Offset	0x0000 014C		
Physical Address	0x4848 424C	**Instance**	PORT
Description	CPSW PORT 1 RX DSCP priority to RX packet mapping reg 7		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	PRI63			RESERVED	PRI62			RESERVED	PRI61			RESERVED	PRI60			RESERVED	PRI59			RESERVED	PRI58			RESERVED	PRI57			RESERVED	PRI56		

Bits	Field Name	Description	Type	Reset
31	RESERVED		R	0x0
30:28	PRI63	Priority 63 - A packet TOS of 0d63 is mapped to this received packet priority.	RW	0x0
27	RESERVED		R	0x0
26:24	PRI62	Priority 62 - A packet TOS of 0d62 is mapped to this received packet priority.	RW	0x0
23	RESERVED		R	0x0
22:20	PRI61	Priority 61 - A packet TOS of 0d61 is mapped to this received packet priority.	RW	0x0
19	RESERVED		R	0x0
18:16	PRI60	Priority 60 - A packet TOS of 0d60 is mapped to this received packet priority.	RW	0x0
15	RESERVED		R	0x0
14:12	PRI59	Priority 59 - A packet TOS of 0d59 is mapped to this received packet priority.	RW	0x0
11	RESERVED		R	0x0
10:8	PRI58	Priority 58 - A packet TOS of 0d58 is mapped to this received packet priority.	RW	0x0
7	RESERVED		R	0x0
6:4	PRI57	Priority 57 - A packet TOS of 0d57 is mapped to this received packet priority.	RW	0x0
3	RESERVED		R	0x0
2:0	PRI56	Priority 56 - A packet TOS of 0d56 is mapped to this received packet priority.	RW	0x0

Table 18-929. Register Call Summary for Register P1_RX_DSCP_PRI_MAP7

Gigabit Ethernet Switch (GMAC_SW)

- PORT Register Summary: [0]

Table 18-930. P1_IDLE2LPI

Address Offset	0x0000 0150		
Physical Address	0x4848 4250	**Instance**	PORT
Description	Port 1 EEE Idle to LPI Counter Load Value Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								P1_IDLE2LPI																							

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19:0	P1_IDLE2LPI	Port 1 EEE Idle to LPI counter load value - After CLKSTOP_REQ is asserted, this value is loaded into the port 1 idle to LPI counter on each clock that the port 1 transmit is not idle. Port 0 enters the transmit LPI state when this counter decrements to zero. This counter decrements each time the EEE prescale counter decrements to zero.	RW	0x0

Table 18-931. Register Call Summary for Register P1_IDLE2LPI

Gigabit Ethernet Switch (GMAC_SW)

- PORT Register Summary: [0]
- PORT Register Description: [1]

Table 18-932. P1_LPI2WAKE

Address Offset	0x0000 0154		
Physical Address	0x4848 4254	**Instance**	PORT
Description	Port 1 EEE LPI to Wake Counter Load Value Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								P1_LPI2WAKE																							

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19:0	P1_LPI2WAKE	Port 1 EEE LPI to wake counter load value – When the port is in the transmit LPI state and the CLKSTOP_REQ signal is deasserted, this value is loaded into the port 1 LPI to wake counter. Transmit packet operations may begin (resume) when the LPI to wake count decrements to zero (on the pre-scale count). This is the time that the CPDMA transmit must wait before transmit (switch ingress) packet operations begin (resume after wakeup).	RW	0x0

Table 18-933. Register Call Summary for Register P1_LPI2WAKE

Gigabit Ethernet Switch (GMAC_SW)

- PORT Register Summary: [0]
- PORT Register Description: [1]

Table 18-934. P2_CONTROL

Address Offset	0x0000 0200		
Physical Address	0x4848 4300	**Instance**	PORT
Description	CPSW_3GF PORT 2 control register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED						P2_TX_CLKSTOP_EN	P2_PASS_PRI_TAGGED	RESERVED		P2_VLAN_LTYPE2_EN	P2_VLAN_LTYPE1_EN	RESERVED			P2_DSCP_PRI_EN	P2_TS_107	P2_TS_320	P2_TS_319	P2_TS_132	P2_TS_131	P2_TS_130	P2_TS_129	P2_TS_TTL_NONZERO	P2_TS_UNI_EN	P2_TS_ANNEX_F_EN	P2_TS_ANNEX_E_EN	P2_TS_ANNEX_D_EN	P2_TS_LTYPE2_EN	P2_TS_LTYPE1_EN	P2_TS_TX_EN	P2_TS_RX_EN

Bits	Field Name	Description	Type	Reset
31:26	RESERVED		R	0x0
25	P2_TX_CLKSTOP_EN	Port 2 Transmit clockstop enable 0 – RGMII transmit clockstop not enabled 1 – RGMII transmit clockstop enabled. The transmit clock will be stopped after the LPI state is entered (and indicated to the CPRGMII) and the P2_Idle2LPI time is counted (counter value reused). The P2_Idle2LPI counter value must be greater than 9 transmit clocks (slowest clock)	RW	0x0
24	P2_PASS_PRI_TAGGED	Port 2 Pass Priority Tagged 0 - Priority tagged packets have the zero VID replaced with the input port P2_PORT_VLAN [11:0] 1 - Priority tagged packets are processed unchanged.	RW	0x0
23:22	RESERVED		R	0x0
21	P2_VLAN_LTYPE2_EN	Port 2 VLAN LTYPE 2 enable 0 - disabled 1 - VLAN LTYPE2 enabled on transmit and receive	RW	0x0
20	P2_VLAN_LTYPE1_EN	Port 2 VLAN LTYPE 1 enable 0 - disabled 1 - VLAN LTYPE1 enabled on transmit and receive	RW	0x0
19:17	RESERVED		R	0x0
16	P2_DSCP_PRI_EN	Port 0 DSCP Priority Enable 0 - DSCP priority disabled 1 - DSCP priority enabled. All non-tagged IPV4 packets have their received packet priority determined by mapping the 6 TOS bits through the port DSCP priority mapping registers.	RW	0x0
15	P2_TS_107	Port 2 Time Sync Destination IP Address 107 enable 0 – disabled 1 – destination IP address (dec) 224.0.0.107 is enabled.	RW	0x0
14	P2_TS_320	Port 2 Time Sync Destination Port Number 320 enable 0 - disabled 1 - Annex D (UDP/IPv4) time sync packet destination port number 320 (decimal) is enabled.	RW	0x0
13	P2_TS_319	Port 2 Time Sync Destination Port Number 319 enable 0 - disabled 1 - Annex D (UDP/IPv4) time sync packet destination port number 319 (decimal) is enabled.	RW	0x0
12	P2_TS_132	Port 2 Time Sync Destination IP Address 132 enable 0 - disabled 1 - Annex D (UDP/IPv4) time sync packet destination IP address number 132 (decimal) is enabled.	RW	0x0

Bits	Field Name	Description	Type	Reset
11	P2_TS_131	Port 2 Time Sync Destination IP Address 131 enable 0 - disabled 1 - Annex D (UDP/IPv4) time sync packet destination IP address number 131 (decimal) is enabled.	RW	0x0
10	P2_TS_130	Port 2 Time Sync Destination IP Address 130 enable 0 - disabled 1 - Annex D (UDP/IPv4) time sync packet destination IP address number 130 (decimal) is enabled.	RW	0x0
9	P2_TS_129	Port 2 Time Sync Destination IP Address 129 enable 0 - disabled 1 - Annex D (UDP/IPv4) time sync packet destination IP address number 129 (decimal) is enabled.	RW	0x0
8	P2_TS_TTL_NONZERO	Port 2 Time Sync Time To Live Non-zero enable. 0 = TTL must be zero. 1 = TTL may be any value.	RW	0x0
7	P2_TS_UNI_EN	Port 2 Time Sync Unicast Enable 0 – Unicast disabled 1 – Unicast enabled	RW	0x0
6	P2_TS_ANNEX_F_EN	Port 2 Time Sync Annex F enable 0 – Annex F disabled 1 – Annex F enabled	RW	0X0
5	P2_TS_ANNEX_E_EN	Port 2 Time Sync Annex E enable 0 – Annex E disabled 1 – Annex E enabled	RW	0X0
4	P2_TS_ANNEX_D_EN	Port 2 Time Sync Annex D enable 0 - Annex D disabled 1 - Annex D enabled	RW	0x0
3	P2_TS_LTYPE2_EN	Port 2 Time Sync LTYPE 2 enable 0 - disabled 1 - enabled	RW	0x0
2	P2_TS_LTYPE1_EN	Port 2 Time Sync LTYPE 1 enable 0 - disabled 1 - enabled	RW	0x0
1	P2_TS_TX_EN	Port 2 Time Sync Transmit Enable 0 - disabled 1 - enabled	RW	0x0
0	P2_TS_RX_EN	Port 2 Time Sync Receive Enable 0 - Port 1 Receive Time Sync disabled 1 - Port 1 Receive Time Sync enabled	RW	0x0

Table 18-935. Register Call Summary for Register P2_CONTROL

Gigabit Ethernet Switch (GMAC_SW)
- PORT Register Summary: [0]

Table 18-936. P2_MAX_BLKS

Address Offset	0x0000 0208		
Physical Address	0x4848 4308	**Instance**	PORT
Description	CPSW PORT 2 maximum FIFO blocks register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								P2_TX_MAX_BLKS				P2_RX_MAX_BLKS			

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		RW	0x0
8:4	P2_TX_MAX_BLKS	Transmit FIFO Maximum Blocks - This value is the maximum number of 1k memory blocks that may be allocated to the FIFO's logical transmit priority queues. 0x11 is the recommended value of P2_TX_MAX_BLKS unless the port is in fullduplex flow control mode. In flow control mode, the P2_RX_MAX_BLKS will need to increase in order to accept the required run out in fullduplex mode. This value will need to decrease by the amount of increase in P2_RX_MAX_BLKS. 0xE is the minimum value P2_TX_MAX_BLKS.	RW	0x11
3:0	P2_RX_MAX_BLKS	Receive FIFO Maximum Blocks - This value is the maximum number of 1k memory blocks that may be allocated to the FIFO's logical receive queue. This value must be greater than or equal to 0x3. It should be increased In fullduplex flow control mode to 0x5 or 0x6 depending on the required runout space. The P2_TX_MAX_BLKS value must be decreased by the amount of increase in P2_RX_MAX_BLKS. 0x3 is the minimum value P2_RX_MAX_BLKS and 0x6 is the maximum value.	RW	0x3

Table 18-937. Register Call Summary for Register P2_MAX_BLKS

Gigabit Ethernet Switch (GMAC_SW)
- PORT Register Summary: [0]

Table 18-938. P2_BLK_CNT

Address Offset	0x0000 020C		
Physical Address	0x4848 430C	**Instance**	PORT
Description	CPSW PORT 2 FIFO block usage count (read only)		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								P2_TX_BLK_CNT						P2_RX_BLK_CNT	

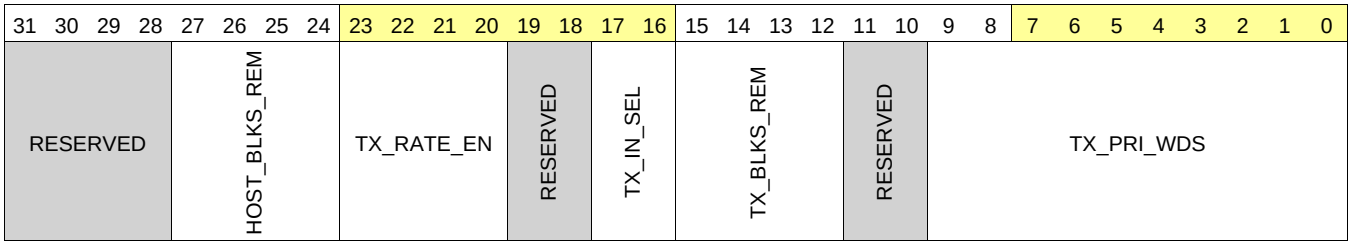
Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8:4	P2_TX_BLK_CNT	Port 2 Transmit Block Count Usage - This value is the number of blocks allocated to the FIFO logical transmit queues.	R	0x4
3:0	P2_RX_BLK_CNT	Port 2 Receive Block Count Usage - This value is the number of blocks allocated to the FIFO logical receive queues.	R	0x1

Table 18-939. Register Call Summary for Register P2_BLK_CNT

Gigabit Ethernet Switch (GMAC_SW)
- PORT Register Summary: [0]

Table 18-940. P2_TX_IN_CTL

Address Offset	0x0000 0210		
Physical Address	0x4848 4310	**Instance**	PORT
Description	CPSW PORT 2 transmit FIFO control		
Type	RW		



Bits	Field Name	Description	Type	Reset
31:28	RESERVED		RW	0x0
27:24	HOST_BLKS_REM	Transmit FIFO Blocks that must be free before a non rate-limited CPDMA channel can begin sending a packet to the FIFO.	RW	0x8
23:20	TX_RATE_EN	Transmit FIFO Input Rate Enable	RW	0x0
19:18	RESERVED		RW	0x0
17:16	TX_IN_SEL	Transmit FIFO Input Queue Type Select 0x0 - Normal priority mode 0x1 - reserved 0x2 - Rate Limit mode 0x3 - reserved	RW	0x0
15:12	TX_BLKS_REM	Transmit FIFO Input blocks to subtract on non rate-limited traffic in rate limit mode.	RW	0x4
11:10	RESERVED		RW	0x0
9:0	TX_PRI_WDS	Transmit FIFO Words in queue	RW	0xc0

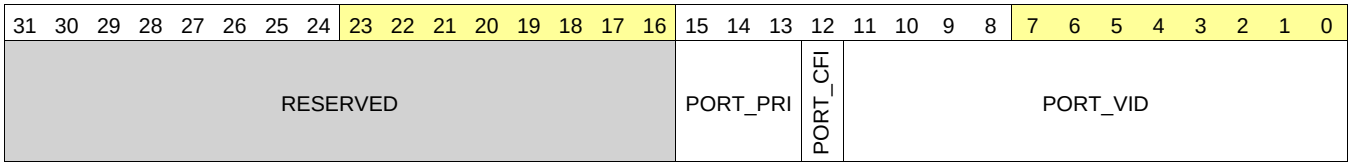
Table 18-941. Register Call Summary for Register P2_TX_IN_CTL

Gigabit Ethernet Switch (GMAC_SW)

- PORT Register Summary: [0]

Table 18-942. P2_PORT_VLAN

Address Offset	0x0000 0214		
Physical Address	0x4848 4314	**Instance**	PORT
Description	CPSW PORT 2 VLAN register		
Type	RW		



Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0
15:13	PORT_PRI	Port VLAN Priority (7 is highest priority)	RW	0x0
12	PORT_CFI	Port CFI bit	RW	0x0
11:0	PORT_VID	Port VLAN ID	RW	0x0

Table 18-943. Register Call Summary for Register P2_PORT_VLAN

Gigabit Ethernet Switch (GMAC_SW)

- PORT Register Summary: [0]
- PORT Register Description: [1]

Table 18-944. P2_TX_PRI_MAP

Address Offset	0x0000 0218		
Physical Address	0x4848 4318	**Instance**	PORT
Description	CPSW PORT 2 TX header priority to switch priority mapping register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED		PRI7		RESERVED		PRI6		RESERVED		PRI5		RESERVED		PRI4		RESERVED		PRI3		RESERVED		PRI2		RESERVED		PRI1		RESERVED		PRI0	

Bits	Field Name	Description	Type	Reset
31:30	RESERVED		R	0x0
29:28	PRI7	Priority 7 - A packet header priority of 0x7 is given this switch queue priority.	RW	0x3
27:26	RESERVED		R	0x0
25:24	PRI6	Priority 6 - A packet header priority of 0x6 is given this switch queue priority.	RW	0x3
23:22	RESERVED		R	0x0
21:20	PRI5	Priority 5 - A packet header priority of 0x5 is given this switch queue priority.	RW	0x2
19:18	RESERVED		R	0x0
17:16	PRI4	Priority 4 - A packet header priority of 0x4 is given this switch queue priority.	RW	0x2
15:14	RESERVED		R	0x0
13:12	PRI3	Priority 3 - A packet header priority of 0x3 is given this switch queue priority.	RW	0x1
11:10	RESERVED		R	0x0
9:8	PRI2	Priority 2 - A packet header priority of 0x2 is given this switch queue priority.	RW	0x0
7:6	RESERVED		R	0x0
5:4	PRI1	Priority 1 - A packet header priority of 0x1 is given this switch queue priority.	RW	0x0
3:2	RESERVED		R	0x0
1:0	PRI0	Priority 0 - A packet header priority of 0x0 is given this switch queue priority.	RW	0x1

Table 18-945. Register Call Summary for Register P2_TX_PRI_MAP

Gigabit Ethernet Switch (GMAC_SW)

- PORT Register Summary: [0]

Table 18-946. P2_TS_SEQ_MTYPE

Address Offset	0x0000 021C		
Physical Address	0x4848 431C	**Instance**	PORT
Description	CPSW_3GF PORT 2 time sync sequence ID offset and message type.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								P2_TS_SEQ_ID_OFFSET								P2_TS_MSG_TYPE_EN															

Bits	Field Name	Description	Type	Reset
31:22	RESERVED		R	0x0
21:16	P2_TS_SEQ_ID_OFFSET	Port 2 Time Sync Sequence ID Offset This is the number of octets that the sequence ID is offset in the tx and rx time sync message header. The minimum value is 6.	RW	0x1E
15:0	P2_TS_MSG_TYPE_EN	Port 2 Time Sync Message Type Enable - Each bit in this field enables the corresponding message type in receive and transmit time sync messages (Bit 0 enables message type 0 etc.).	RW	0x0

Table 18-947. Register Call Summary for Register P2_TS_SEQ_MTYPE

Gigabit Ethernet Switch (GMAC_SW)

- PORT Register Summary: [0]

Table 18-948. P2_SA_LO

Address Offset	0x0000 0220		
Physical Address	0x4848 4320	**Instance**	PORT
Description	CPSW CPGMAC_SL2 source address low register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																MACSRCADDR_7_0								MACSRCADDR_15_8							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0
15:8	MACSRCADDR_7_0	Source Address Lower 8 bits (byte 0)	RW	0x0
7:0	MACSRCADDR_15_8	Source Address bits 15:8 (byte 1)	RW	0x0

Table 18-949. Register Call Summary for Register P2_SA_LO

Gigabit Ethernet Switch (GMAC_SW)

- PORT Register Summary: [0]

Table 18-950. P2_SA_HI

Address Offset	0x0000 0224		
Physical Address	0x4848 4324	**Instance**	PORT
Description	CPSW CPGMAC_SL2 source address high register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
MACSRCADDR_23_16								MACSRCADDR_31_23								MACSRCADDR_39_32								MACSRCADDR_47_40							

Bits	Field Name	Description	Type	Reset
31:24	MACSRCADDR_23_16	Source Address bits 23:16 (byte 2)	RW	0x0
23:16	MACSRCADDR_31_23	Source Address bits 31:23 (byte 3)	RW	0x0
15:8	MACSRCADDR_39_32	Source Address bits 39:32 (byte 4)	RW	0x0
7:0	MACSRCADDR_47_40	Source Address bits 47:40 (byte 5)	RW	0x0

Table 18-951. Register Call Summary for Register P2_SA_HI

Gigabit Ethernet Switch (GMAC_SW)

- PORT Register Summary: [0]

Table 18-952. P2_SEND_PERCENT

Address Offset	0x0000 0228		
Physical Address	0x4848 4328	**Instance**	PORT
Description	CPSW PORT 2 transmit queue send percentages		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								PRI3_SEND_PERCENT								RESERVED	PRI2_SEND_PERCENT							RESERVED	PRI1_SEND_PERCENT						

Bits	Field Name	Description	Type	Reset
31:23	RESERVED		R	0x0
22:16	PRI3_SEND_PERCENT	Priority 3 Transmit Percentage - This percentage value is sent from FIFO priority 3 (maximum) when the CPSW_PTYPE[21] P2_PRI3_SHAPE_EN is set (queue shaping enabled). This is the percentage of the wire that packets from priority 3 receive (which includes interpacket gap and preamble bytes). If shaping is enabled on this queue then this value must be between zero and 0d100 (not inclusive).	RW	0x0
15	RESERVED		R	0x0
14:8	PRI2_SEND_PERCENT	Priority 2 Transmit Percentage - This percentage value is sent from FIFO priority 2 (maximum) when the CPSW_PTYPE[20] P2_PRI2_SHAPE_EN is set (queue shaping enabled). This is the percentage of the wire that packets from priority 2 receive (which includes interpacket gap and preamble bytes). If shaping is enabled on this queue then this value must be between zero and 0d100 (not inclusive).	RW	0x0
7	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
6:0	PRI1_SEND_PERCENT	Priority 1 Transmit Percentage - This percentage value is sent from FIFO priority 1 (maximum) when the CPSW_PTYPE[19] P2_PRI1_SHAPE_EN is set (queue shaping enabled). This is the percentage of the wire that packets from priority 1 receive (which includes interpacket gap and preamble bytes). If shaping is enabled on this queue then this value must be between zero and 0d100 (not inclusive).	RW	0x0

Table 18-953. Register Call Summary for Register P2_SEND_PERCENT

Gigabit Ethernet Switch (GMAC_SW)
- PORT Register Summary: [0]

Table 18-954. P2_RX_DSCP_PRI_MAP0

Address Offset	0x0000 0230		
Physical Address	0x4848 4330	**Instance**	PORT
Description	CPSW PORT 2 RX DSCP priority to RX packet mapping reg 0		
Type	RW		

31	30 29 28	27	26 25 24	23	22 21 20	19	18 17 16	15	14 13 12	11	10 9 8	7	6 5 4	3	2 1 0
RESERVED	PRI7	RESERVED	PRI6	RESERVED	PRI5	RESERVED	PRI4	RESERVED	PRI3	RESERVED	PRI2	RESERVED	PRI1	RESERVED	PRI0

Bits	Field Name	Description	Type	Reset
31	RESERVED		R	0x0
30:28	PRI7	Priority 7 - A packet TOS of 0d7 is mapped to this received packet priority.	RW	0x0
27	RESERVED		R	0x0
26:24	PRI6	Priority 6 - A packet TOS of 0d6 is mapped to this received packet priority.	RW	0x0
23	RESERVED		R	0x0
22:20	PRI5	Priority 5 - A packet TOS of 0d5 is mapped to this received packet priority.	RW	0x0
19	RESERVED		R	0x0
18:16	PRI4	Priority 4 - A packet TOS of 0d4 is mapped to this received packet priority.	RW	0x0
15	RESERVED		R	0x0
14:12	PRI3	Priority 3 - A packet TOS of 0d3 is mapped to this received packet priority.	RW	0x0
11	RESERVED		R	0x0
10:8	PRI2	Priority 2 - A packet TOS of 0d2 is mapped to this received packet priority.	RW	0x0
7	RESERVED		R	0x0
6:4	PRI1	Priority 1 - A packet TOS of 0d1 is mapped to this received packet priority.	RW	0x0
3	RESERVED		R	0x0
2:0	PRI0	Priority 0 - A packet TOS of 0d0 is mapped to this received packet priority.	RW	0x0

Table 18-955. Register Call Summary for Register P2_RX_DSCP_PRI_MAP0

Gigabit Ethernet Switch (GMAC_SW)

- PORT Register Summary: [0]

Table 18-956. P2_RX_DSCP_PRI_MAP1

Address Offset	0x0000 0234		
Physical Address	0x4848 4334	**Instance**	PORT
Description	CPSW PORT 2 RX DSCP priority to RX packet mapping reg 1		
Type	RW		

31	30 29 28	27	26 25 24	23	22 21 20	19	18 17 16	15	14 13 12	11	10 9 8	7	6 5 4	3	2 1 0
RESERVED	PRI15	RESERVED	PRI14	RESERVED	PRI13	RESERVED	PRI12	RESERVED	PRI11	RESERVED	PRI10	RESERVED	PRI9	RESERVED	PRI8

Bits	Field Name	Description	Type	Reset
31	RESERVED		R	0x0
30:28	PRI15	Priority 15 - A packet TOS of 0d15 is mapped to this received packet priority.	RW	0x0
27	RESERVED		R	0x0
26:24	PRI14	Priority 14 - A packet TOS of 0d14 is mapped to this received packet priority.	RW	0x0
23	RESERVED		R	0x0
22:20	PRI13	Priority 13 - A packet TOS of 0d13 is mapped to this received packet priority.	RW	0x0
19	RESERVED		R	0x0
18:16	PRI12	Priority 12 - A packet TOS of 0d12 is mapped to this received packet priority.	RW	0x0
15	RESERVED		R	0x0
14:12	PRI11	Priority 11 - A packet TOS of 0d11 is mapped to this received packet priority.	RW	0x0
11	RESERVED		R	0x0
10:8	PRI10	Priority 10 - A packet TOS of 0d10 is mapped to this received packet priority.	RW	0x0
7	RESERVED		R	0x0
6:4	PRI9	Priority 9 - A packet TOS of 0d9 is mapped to this received packet priority.	RW	0x0
3	RESERVED		R	0x0
2:0	PRI8	Priority 8 - A packet TOS of 0d8 is mapped to this received packet priority.	RW	0x0

Table 18-957. Register Call Summary for Register P2_RX_DSCP_PRI_MAP1

Gigabit Ethernet Switch (GMAC_SW)

- PORT Register Summary: [0]

Table 18-958. P2_RX_DSCP_PRI_MAP2

Address Offset	0x0000 0238		
Physical Address	0x4848 4338	**Instance**	PORT
Description	CPSW PORT 2 RX DSCP priority to RX packet mapping reg 2		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	PRI23			RESERVED	PRI22			RESERVED	PRI21			RESERVED	PRI20			RESERVED	PRI19			RESERVED	PRI18			RESERVED	PRI17			RESERVED	PRI16		

Bits	Field Name	Description	Type	Reset
31	RESERVED		R	0x0
30:28	PRI23	Priority 23 - A packet TOS of 0d23 is mapped to this received packet priority.	RW	0x0
27	RESERVED		R	0x0
26:24	PRI22	Priority 22 - A packet TOS of 0d22 is mapped to this received packet priority.	RW	0x0
23	RESERVED		R	0x0
22:20	PRI21	Priority 21 - A packet TOS of 0d21 is mapped to this received packet priority.	RW	0x0
19	RESERVED		R	0x0
18:16	PRI20	Priority 20 - A packet TOS of 0d20 is mapped to this received packet priority.	RW	0x0
15	RESERVED		R	0x0
14:12	PRI19	Priority 19 - A packet TOS of 0d19 is mapped to this received packet priority.	RW	0x0
11	RESERVED		R	0x0
10:8	PRI18	Priority 18 - A packet TOS of 0d18 is mapped to this received packet priority.	RW	0x0
7	RESERVED		R	0x0
6:4	PRI17	Priority 17 - A packet TOS of 0d17 is mapped to this received packet priority.	RW	0x0
3	RESERVED		R	0x0
2:0	PRI16	Priority 16 - A packet TOS of 0d16 is mapped to this received packet priority.	RW	0x0

Table 18-959. Register Call Summary for Register P2_RX_DSCP_PRI_MAP2

Gigabit Ethernet Switch (GMAC_SW)

- PORT Register Summary: [0]

Table 18-960. P2_RX_DSCP_PRI_MAP3

Address Offset	0x0000 023C		
Physical Address	0x4848 433C	**Instance**	PORT
Description	CPSW PORT 2 RX DSCP priority to RX packet mapping reg 3		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	PRI31			RESERVED	PRI30			RESERVED	PRI29			RESERVED	PRI28			RESERVED	PRI27			RESERVED	PRI26			RESERVED	PRI25			RESERVED	PRI24		

Bits	Field Name	Description	Type	Reset
31	RESERVED		R	0x0
30:28	PRI31	Priority 31 - A packet TOS of 0d31 is mapped to this received packet priority.	RW	0x0
27	RESERVED		R	0x0
26:24	PRI30	Priority 30 - A packet TOS of 0d30 is mapped to this received packet priority.	RW	0x0
23	RESERVED		R	0x0
22:20	PRI29	Priority 29 - A packet TOS of 0d39 is mapped to this received packet priority.	RW	0x0
19	RESERVED		R	0x0
18:16	PRI28	Priority 28 - A packet TOS of 0d28 is mapped to this received packet priority.	RW	0x0
15	RESERVED		R	0x0
14:12	PRI27	Priority 27 - A packet TOS of 0d27 is mapped to this received packet priority.	RW	0x0
11	RESERVED		R	0x0
10:8	PRI26	Priority 26 - A packet TOS of 0d26 is mapped to this received packet priority.	RW	0x0
7	RESERVED		R	0x0
6:4	PRI25	Priority 25 - A packet TOS of 0d25 is mapped to this received packet priority.	RW	0x0
3	RESERVED		R	0x0
2:0	PRI24	Priority 24 - A packet TOS of 0d24 is mapped to this received packet priority.	RW	0x0

Table 18-961. Register Call Summary for Register P2_RX_DSCP_PRI_MAP3

Gigabit Ethernet Switch (GMAC_SW)

- PORT Register Summary: [0]

Table 18-962. P2_RX_DSCP_PRI_MAP4

Address Offset	0x0000 0240		
Physical Address	0x4848 4340	**Instance**	PORT
Description	CPSW PORT 2 RX DSCP priority to RX packet mapping reg 4		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	PRI39			RESERVED	PRI38			RESERVED	PRI37			RESERVED	PRI36			RESERVED	PRI35			RESERVED	PRI34			RESERVED	PRI33			RESERVED	PRI32		

Bits	Field Name	Description	Type	Reset
31	RESERVED		R	0x0
30:28	PRI39	Priority 39 - A packet TOS of 0d39 is mapped to this received packet priority.	RW	0x0
27	RESERVED		R	0x0
26:24	PRI38	Priority 38 - A packet TOS of 0d38 is mapped to this received packet priority.	RW	0x0
23	RESERVED		R	0x0
22:20	PRI37	Priority 37 - A packet TOS of 0d37 is mapped to this received packet priority.	RW	0x0
19	RESERVED		R	0x0
18:16	PRI36	Priority 36 - A packet TOS of 0d36 is mapped to this received packet priority.	RW	0x0
15	RESERVED		R	0x0
14:12	PRI35	Priority 35 - A packet TOS of 0d35 is mapped to this received packet priority.	RW	0x0
11	RESERVED		R	0x0
10:8	PRI34	Priority 34 - A packet TOS of 0d34 is mapped to this received packet priority.	RW	0x0
7	RESERVED		R	0x0
6:4	PRI33	Priority 33 - A packet TOS of 0d33 is mapped to this received packet priority.	RW	0x0
3	RESERVED		R	0x0
2:0	PRI32	Priority 32 - A packet TOS of 0d32 is mapped to this received packet priority.	RW	0x0

Table 18-963. Register Call Summary for Register P2_RX_DSCP_PRI_MAP4

Gigabit Ethernet Switch (GMAC_SW)

- PORT Register Summary: [0]

Table 18-964. P2_RX_DSCP_PRI_MAP5

Address Offset	0x0000 0244		
Physical Address	0x4848 4344	**Instance**	PORT
Description	CPSW PORT 2 RX DSCP priority to RX packet mapping reg 5		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	PRI47			RESERVED	PRI46			RESERVED	PRI45			RESERVED	PRI44			RESERVED	PRI43			RESERVED	PRI42			RESERVED	PRI41			RESERVED	PRI40		

Bits	Field Name	Description	Type	Reset
31	RESERVED		R	0x0
30:28	PRI47	Priority 47 - A packet TOS of 0d47 is mapped to this received packet priority.	RW	0x0
27	RESERVED		R	0x0
26:24	PRI46	Priority 46 - A packet TOS of 0d46 is mapped to this received packet priority.	RW	0x0
23	RESERVED		R	0x0
22:20	PRI45	Priority 45 - A packet TOS of 0d45 is mapped to this received packet priority.	RW	0x0
19	RESERVED		R	0x0
18:16	PRI44	Priority 44 - A packet TOS of 0d44 is mapped to this received packet priority.	RW	0x0
15	RESERVED		R	0x0
14:12	PRI43	Priority 43 - A packet TOS of 0d43 is mapped to this received packet priority.	RW	0x0
11	RESERVED		R	0x0
10:8	PRI42	Priority 42 - A packet TOS of 0d42 is mapped to this received packet priority.	RW	0x0
7	RESERVED		R	0x0
6:4	PRI41	Priority 41 - A packet TOS of 0d41 is mapped to this received packet priority.	RW	0x0
3	RESERVED		R	0x0
2:0	PRI40	Priority 40 - A packet TOS of 0d40 is mapped to this received packet priority.	RW	0x0

Table 18-965. Register Call Summary for Register P2_RX_DSCP_PRI_MAP5

Gigabit Ethernet Switch (GMAC_SW)

- PORT Register Summary: [0]

Table 18-966. P2_RX_DSCP_PRI_MAP6

Address Offset	0x0000 0248		
Physical Address	0x4848 4348	**Instance**	PORT
Description	CPSW PORT 2 RX DSCP priority to RX packet mapping reg 6		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	PRI55			RESERVED	PRI54			RESERVED	PRI53			RESERVED	PRI52			RESERVED	PRI51			RESERVED	PRI50			RESERVED	PRI49			RESERVED	PRI48		

Bits	Field Name	Description	Type	Reset
31	RESERVED		R	0x0
30:28	PRI55	Priority 55 - A packet TOS of 0d55 is mapped to this received packet priority.	RW	0x0
27	RESERVED		R	0x0
26:24	PRI54	Priority 54 - A packet TOS of 0d54 is mapped to this received packet priority.	RW	0x0
23	RESERVED		R	0x0
22:20	PRI53	Priority 53 - A packet TOS of 0d53 is mapped to this received packet priority.	RW	0x0
19	RESERVED		R	0x0
18:16	PRI52	Priority 52 - A packet TOS of 0d52 is mapped to this received packet priority.	RW	0x0
15	RESERVED		R	0x0
14:12	PRI51	Priority 51 - A packet TOS of 0d51 is mapped to this received packet priority.	RW	0x0
11	RESERVED		R	0x0
10:8	PRI50	Priority 50 - A packet TOS of 0d50 is mapped to this received packet priority.	RW	0x0
7	RESERVED		R	0x0
6:4	PRI49	Priority 49 - A packet TOS of 0d49 is mapped to this received packet priority.	RW	0x0
3	RESERVED		R	0x0
2:0	PRI48	Priority 48 - A packet TOS of 0d48 is mapped to this received packet priority.	RW	0x0

Table 18-967. Register Call Summary for Register P2_RX_DSCP_PRI_MAP6

Gigabit Ethernet Switch (GMAC_SW)

- PORT Register Summary: [0]

Table 18-968. P2_RX_DSCP_PRI_MAP7

Address Offset	0x0000 024C		
Physical Address	0x4848 434C	**Instance**	PORT
Description	CPSW PORT 2 RX DSCP priority to RX packet mapping reg 7		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	PRI63			RESERVED	PRI62			RESERVED	PRI61			RESERVED	PRI60			RESERVED	PRI59			RESERVED	PRI58			RESERVED	PRI57			RESERVED	PRI56		

Bits	Field Name	Description	Type	Reset
31	RESERVED		R	0x0
30:28	PRI63	Priority 63 - A packet TOS of 0d63 is mapped to this received packet priority.	RW	0x0
27	RESERVED		R	0x0
26:24	PRI62	Priority 62 - A packet TOS of 0d62 is mapped to this received packet priority.	RW	0x0
23	RESERVED		R	0x0
22:20	PRI61	Priority 61 - A packet TOS of 0d61 is mapped to this received packet priority.	RW	0x0
19	RESERVED		R	0x0
18:16	PRI60	Priority 60 - A packet TOS of 0d60 is mapped to this received packet priority.	RW	0x0
15	RESERVED		R	0x0
14:12	PRI59	Priority 59 - A packet TOS of 0d59 is mapped to this received packet priority.	RW	0x0
11	RESERVED		R	0x0
10:8	PRI58	Priority 58 - A packet TOS of 0d58 is mapped to this received packet priority.	RW	0x0
7	RESERVED		R	0x0
6:4	PRI57	Priority 57 - A packet TOS of 0d57 is mapped to this received packet priority.	RW	0x0
3	RESERVED		R	0x0
2:0	PRI56	Priority 56 - A packet TOS of 0d56 is mapped to this received packet priority.	RW	0x0

Table 18-969. Register Call Summary for Register P2_RX_DSCP_PRI_MAP7

Gigabit Ethernet Switch (GMAC_SW)

- PORT Register Summary: [0]

Table 18-970. P2_IDLE2LPI

Address Offset	0x0000 0250		
Physical Address	0x4848 4350	**Instance**	PORT
Description	Port 2 EEE Idle to LPI Counter Load Value Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								P2_IDLE2LPI																							

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19:0	P2_IDLE2LPI	Port 2 EEE Idle to LPI counter load value - After CLKSTOP_REQ is asserted, this value is loaded into the port 2 idle to LPI counter on each clock that the port 2 transmit is not idle. Port 2 enters the transmit LPI state when this counter decrements to zero. This counter decrements each time the EEE prescale counter decrements to zero.	RW	0x0

Table 18-971. Register Call Summary for Register P2_IDLE2LPI

Gigabit Ethernet Switch (GMAC_SW)

- PORT Register Summary: [0]
- PORT Register Description: [1]

Table 18-972. P2_LPI2WAKE

Address Offset	0x0000 0254		
Physical Address	0x4848 4354	**Instance**	PORT
Description	Port 2 EEE LPI to Wake Counter Load Value Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								P2_LPI2WAKE																							

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19:0	P2_LPI2WAKE	Port 2 EEE LPI to wake counter load value – When the port is in the transmit LPI state and the CLKSTOP_REQ signal is deasserted, this value is loaded into the port 2 LPI to wake counter. Transmit packet operations may begin (resume) when the LPI to wake count decrements to zero (on the pre-scale count). This is the time that the CPDMA transmit must wait before transmit (switch ingress) packet operations begin (resume after wakeup).	RW	0x0

Table 18-973. Register Call Summary for Register P2_LPI2WAKE

Gigabit Ethernet Switch (GMAC_SW)

- PORT Register Summary: [0]

18.8.6.4 CPDMA registers

18.8.6.4.1 CPDMA Register Summary

Table 18-974. CPDMA Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	CPDMA Physical Address
CPDMA_TX_IDVER	R	32	0x0000 0000	0x4848 4800
CPDMA_TX_CONTROL	RW	32	0x0000 0004	0x4848 4804
CPDMA_TX_TEARDOWN	RW	32	0x0000 0008	0x4848 4808
CPDMA_RX_IDVER	R	32	0x0000 0010	0x4848 4810
CPDMA_RX_CONTROL	RW	32	0x0000 0014	0x4848 4814
CPDMA_RX_TEARDOWN	RW	32	0x0000 0018	0x4848 4818
CPDMA_SOFT_RESET	RW	32	0x0000 001C	0x4848 481C
CPDMA_DMACONTROL	RW	32	0x0000 0020	0x4848 4820
CPDMA_DMASTATUS	R	32	0x0000 0024	0x4848 4824
CPDMA_RX_BUFFER_OFFSET	RW	32	0x0000 0028	0x4848 4828
CPDMA_EMCONTROL	RW	32	0x0000 002C	0x4848 482C
CPDMA_TX_PRI0_RATE	RW	32	0x0000 0030	0x4848 4830
CPDMA_TX_PRI1_RATE	RW	32	0x0000 0034	0x4848 4834
CPDMA_TX_PRI2_RATE	RW	32	0x0000 0038	0x4848 4838
CPDMA_TX_PRI3_RATE	RW	32	0x0000 003C	0x4848 483C
CPDMA_TX_PRI4_RATE	RW	32	0x0000 0040	0x4848 4840
CPDMA_TX_PRI5_RATE	RW	32	0x0000 0044	0x4848 4844
CPDMA_TX_PRI6_RATE	RW	32	0x0000 0048	0x4848 4848
CPDMA_TX_PRI7_RATE	RW	32	0x0000 004C	0x4848 484C
CPDMA_TX_INTSTAT_RAW	R	32	0x0000 0080	0x4848 4880
CPDMA_TX_INTSTAT_MASKED	R	32	0x0000 0084	0x4848 4884
CPDMA_TX_INTMASK_SET	W	32	0x0000 0088	0x4848 4888
CPDMA_TX_INTMASK_CLEAR	W	32	0x0000 008C	0x4848 488C
CPDMA_IN_VECTOR	R	32	0x0000 0090	0x4848 4890
CPDMA_EOI_VECTOR	RW	32	0x0000 0094	0x4848 4894
CPDMA_RX_INTSTAT_RAW	R	32	0x0000 00A0	0x4848 48A0
CPDMA_RX_INTSTAT_MASKED	R	32	0x0000 00A4	0x4848 48A4
CPDMA_RX_INTMASK_SET	RW	32	0x0000 00A8	0x4848 48A8
CPDMA_RX_INTMASK_CLEAR	RW	32	0x0000 00AC	0x4848 48AC
CPDMA_DMA_INTSTAT_RAW	R	32	0x0000 00B0	0x4848 48B0
CPDMA_DMA_INTSTAT_MASKED	R	32	0x0000 00B4	0x4848 48B4
CPDMA_DMA_INTMASK_SET	W	32	0x0000 00B8	0x4848 48B8
CPDMA_DMA_INTMASK_CLEAR	RW	32	0x0000 00BC	0x4848 48BC
CPDMA_RX0_PENDTHRESH	RW	32	0x0000 00C0	0x4848 48C0
CPDMA_RX1_PENDTHRESH	RW	32	0x0000 00C4	0x4848 48C4
CPDMA_RX2_PENDTHRESH	RW	32	0x0000 00C8	0x4848 48C8
CPDMA_RX3_PENDTHRESH	RW	32	0x0000 00CC	0x4848 48CC
CPDMA_RX4_PENDTHRESH	RW	32	0x0000 00D0	0x4848 48D0
CPDMA_RX5_PENDTHRESH	RW	32	0x0000 00D4	0x4848 48D4
CPDMA_RX6_PENDTHRESH	RW	32	0x0000 00D8	0x4848 48D8
CPDMA_RX7_PENDTHRESH	RW	32	0x0000 00DC	0x4848 48DC
CPDMA_RX0_FREEBUFFER	W	32	0x0000 00E0	0x4848 48E0
CPDMA_RX1_FREEBUFFER	W	32	0x0000 00E4	0x4848 48E4

Table 18-974. CPDMA Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	CPDMA Physical Address
CPDMA_RX2_FREEBUFFER	W	32	0x0000 00E8	0x4848 48E8
CPDMA_RX3_FREEBUFFER	W	32	0x0000 00EC	0x4848 48EC
CPDMA_RX4_FREEBUFFER	W	32	0x0000 00F0	0x4848 48F0
CPDMA_RX5_FREEBUFFER	W	32	0x0000 00F4	0x4848 48F4
CPDMA_RX6_FREEBUFFER	W	32	0x0000 00F8	0x4848 48F8
CPDMA_RX7_FREEBUFFER	W	32	0x0000 00FC	0x4848 48FC

18.8.6.4.2 CPDMA Register Description

Table 18-975. CPDMA_TX_IDVER

Address Offset	0x0000 0000		
Physical Address	0x4848 4800	**Instance**	CPDMA
Description	CPDMA_REGS TX revision register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REVISION																															

Bits	Field Name	Description	Type	Reset
31:0	REVISION	CPDMA TX Revision Value	R	0x-

Table 18-976. Register Call Summary for Register CPDMA_TX_IDVER

Gigabit Ethernet Switch (GMAC_SW)
- CPDMA Register Summary: [0]

Table 18-977. CPDMA_TX_CONTROL

Address Offset	0x0000 0004		
Physical Address	0x4848 4804	**Instance**	CPDMA
Description	CPDMA_REGS TX control register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															TX_EN

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	TX_EN	TX Enable 0 - Disabled 1 - Enabled	RW	0x0

Table 18-978. Register Call Summary for Register CPDMA_TX_CONTROL

Gigabit Ethernet Switch (GMAC_SW)
- CPDMA RX and TX Interfaces: [0]
- CPDMA Register Summary: [1]

Table 18-979. CPDMA_TX_TEARDOWN

Address Offset	0x0000 0008		
Physical Address	0x4848 4808	**Instance**	CPDMA
Description	CPDMA_REGS TX teardown register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TX_TDN_RDY	RESERVED																							TX_TDN_CH							

Bits	Field Name	Description	Type	Reset
31	TX_TDN_RDY	Tx Teardown Ready - read as zero, but is always assumed to be one (unused).	R	0x0
30:3	RESERVED		R	0x0
2:0	TX_TDN_CH	Tx Teardown Channel - Transmit channel teardown is commanded by writing the encoded value of the transmit channel to be torn down. The teardown register is read as zero.	RW	0x0

Table 18-980. Register Call Summary for Register CPDMA_TX_TEARDOWN

Gigabit Ethernet Switch (GMAC_SW)

- CPDMA RX and TX Interfaces: [0]
- CPDMA Register Summary: [1]

Table 18-981. CPDMA_RX_IDVER

Address Offset	0x0000 0010		
Physical Address	0x4848 4810	**Instance**	CPDMA
Description	CPDMA_REGS RX revision register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REVISION																															

Bits	Field Name	Description	Type	Reset
31:0	REVISION	RX Revision Value	R	0x-

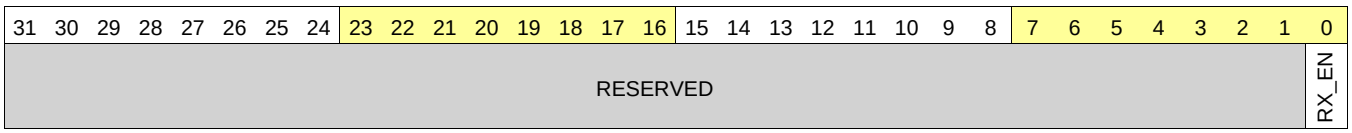
Table 18-982. Register Call Summary for Register CPDMA_RX_IDVER

Gigabit Ethernet Switch (GMAC_SW)

- CPDMA Register Summary: [0]

Table 18-983. CPDMA_RX_CONTROL

Address Offset	0x0000 0014		
Physical Address	0x4848 4814	**Instance**	CPDMA
Description	CPDMA_REGS RX control register		
Type	RW		



Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	RX_EN	RX DMA Enable 0 - Disabled 1 - Enabled	RW	0x0

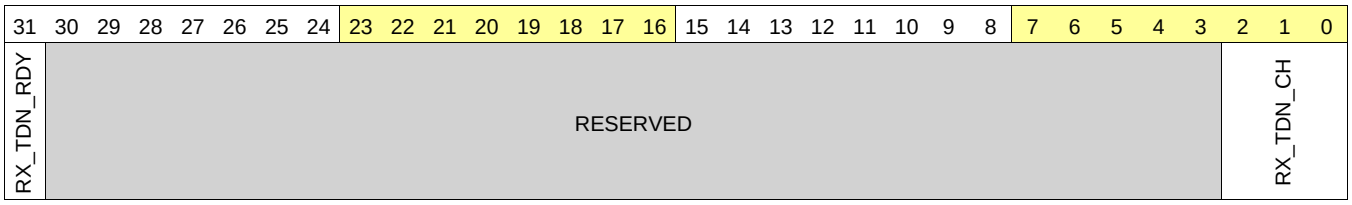
Table 18-984. Register Call Summary for Register CPDMA_RX_CONTROL

Gigabit Ethernet Switch (GMAC_SW)

- CPDMA RX and TX Interfaces: [0]
- CPDMA Register Summary: [1]

Table 18-985. CPDMA_RX_TEARDOWN

Address Offset	0x0000 0018		
Physical Address	0x4848 4818	**Instance**	CPDMA
Description	CPDMA_REGS RX teardown register		
Type	RW		



Bits	Field Name	Description	Type	Reset
31	RX_TDN_RDY	Teardown Ready - read as zero, but is always assumed to be one (unused).	R	0x0
30:3	RESERVED		R	0x0
2:0	RX_TDN_CH	Rx Teardown Channel -Receive channel teardown is commanded by writing the encoded value of the receive channel to be torn down. The teardown register is read as zero.	RW	0x0

Table 18-986. Register Call Summary for Register CPDMA_RX_TEARDOWN

Gigabit Ethernet Switch (GMAC_SW)

- CPDMA RX and TX Interfaces: [0]
- CPDMA Register Summary: [1]

Table 18-987. CPDMA_SOFT_RESET

Address Offset	0x0000 001C		
Physical Address	0x4848 481C	**Instance**	CPDMA
Description	CPDMA_REGS soft reset register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															SOFT_RESET

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	SOFT_RESET	Software reset - Writing a one to this bit causes the CPDMA logic to be reset. Software reset occurs when the RX and TX DMA Controllers are in an idle state to avoid locking up the VBUSP bus. After writing a one to this bit, it may be polled to determine if the reset has occurred. If a one is read, the reset has not yet occurred. If a zero is read then reset has occurred.	RW	0x0

Table 18-988. Register Call Summary for Register CPDMA_SOFT_RESET

Gigabit Ethernet Switch (GMAC_SW)

- Software Reset: [0]
- CPDMA Register Summary: [1]

Table 18-989. CPDMA_DMACONTROL

Address Offset	0x0000 0020		
Physical Address	0x4848 4820	**Instance**	CPDMA
Description	CPDMA_REGS CPDMA control register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																TX_RLIM								RESERVED			RX_CEF	CMD_IDLE	RX_OFFLEN_BLOCK	RX_OWNERSHIP	TX_PTYPE

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0
15:8	TX_RLIM	Transmit Rate Limit Channel Bus 00000000 - no rate-limited channels 10000000 - channel 7 is rate-limited 11000000 - channels 7 downto 6 are rate-limited 11100000 - channels 7 downto 5 are rate-limited 11110000 - channels 7 downto 4 are rate-limited 11111000 - channels 7 downto 3 are rate-limited 11111100 - channels 7 downto 2 are rate-limited 11111110 - channels 7 downto 1 are rate-limited 11111111 - channels 7 downto 0 are rate-limited all others invalid - this bus must be set MSB towards LSB. TX_PTYPE must be set if any TX_RLIM bit is set for fixed priority.	RW	0x0
7:5	RESERVED		R	0x0
4	RX_CEF	RX Copy Error Frames Enable - Enables DMA overrun frames to be transferred to memory (up to the point of overrun). The overrun error bit will be set in the frame EOP buffer descriptor. Overrun frame data will be filtered when RX_CEF is not set. Frames coming from the receive FIFO with other error bits set are not effected by this bit. 0 - Frames containing overrun errors are filtered. 1 - Frames containing overrun errors are transferred to memory.	RW	0x0
3	CMD_IDLE	Command Idle 0 - Idle not commanded 1 - Idle Commanded (read IDLE in CPDMA_DMASTATUS)	RW	0x0
2	RX_OFFLEN_BLOCK	Receive Offset/Length word write block. 0 - Do not block the DMA writes to the receive buffer descriptor offset/buffer length word. The offset/buffer length word is written as specified in CPPI 3.0. 1 - Block all CPDMA DMA controller writes to the receive buffer descriptor offset/buffer length words during CPPI packet processing. when this bit is set, the CPDMA will never write the third word to any receive buffer descriptor.	RW	0x0
1	RX_OWNERSHIP	Receive Ownership Write Bit Value. 0 - The CPDMA writes the receive ownership bit to zero at the end of packet processing as specified in CPPI 3.0. 1 - The CPDMA writes the receive ownership bit to one at the end of packet processing. Users who do not use the ownership mechanism can use this mode to preclude the necessity of software having to set this bit each time the buffer descriptor is used.	RW	0x0
0	TX_PTYPE	Transmit Queue Priority Type 0 - The queue uses a round robin scheme to select the next channel for transmission. 1 - The queue uses a fixed (channel 7 highest priority) priority scheme to select the next channel for transmission	RW	0x0

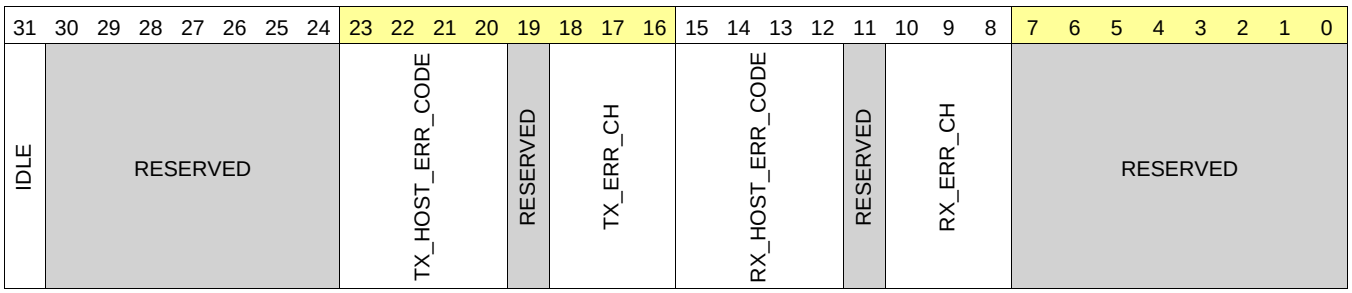
Table 18-990. Register Call Summary for Register CPDMA_DMACONTROL

Gigabit Ethernet Switch (GMAC_SW)

- CPDMA RX and TX Interfaces: [0][1][2]
- FIFO Transmit Queue Control: [3][4]
- Audio Video Bridging: [5]
- CPDMA Register Summary: [6]

Table 18-991. CPDMA_DMASTATUS

Address Offset	0x0000 0024		
Physical Address	0x4848 4824	**Instance**	CPDMA
Description	CPDMA_REGS CPDMA status register		
Type	R		



Bits	Field Name	Description	Type	Reset
31	IDLE	Idle Status Bit - Indicates when set that the CPDMA is not transferring a packet on transmit or receive.	R	0x0
30:24	RESERVED		R	0x0
23:20	TX_HOST_ERR_CODE	TX Host Error Code - This field is set to indicate CPDMA detected TX DMA related host errors. The host should read this field after a HOST_ERR_INT to determine the error. Host error Interrupts require hardware reset in order to recover. A zero packet length is an error, but it is not detected. 0x0 - No error 0x1 - SOP error. 0x2 - Ownership bit not set in SOP buffer. 0x3 - Zero Next Buffer Descriptor Pointer Without EOP 0x4 - Zero Buffer Pointer. 0x5 - Zero Buffer Length 0x6 - Packet Length Error (sum of buffers is less than packet length) 0x7 - 1xF - reserved	R	0x0
19	RESERVED		R	0x0
18:16	TX_ERR_CH	TX Host Error Channel - This field indicates which TX channel (if applicable) the host error occurred on. This field is cleared to zero on a host read.	R	0x0
15:12	RX_HOST_ERR_CODE	RX Host Error Code - This field is set to indicate CPDMA detected RX DMA related host errors. The host should read this field after a HOST_ERR_INT to determine the error. Host error Interrupts require hardware reset in order to recover. 0x0 - No error 0x1 - reserved 0x2 - Ownership bit not set in input buffer. 0x3 - reserved 0x4 - Zero Buffer Pointer. 0x5 - Zero buffer length on non-SOP descriptor 0x6 - SOP buffer length not greater than offset 0x7 - 1xF - reserved	R	0x0
11	RESERVED		R	0x0
10:8	RX_ERR_CH	RX Host Error Channel - This field indicates which RX channel the host error occurred on. This field is cleared to zero on a host read.	R	0x0
7:0	RESERVED		R	0x0

Table 18-992. Register Call Summary for Register CPDMA_DMASTATUS

Gigabit Ethernet Switch (GMAC_SW)

- CPDMA Register Summary: [0]
- CPDMA Register Description: [1]

Table 18-993. CPDMA_RX_BUFFER_OFFSET

Address Offset	0x0000 0028		
Physical Address	0x4848 4828	**Instance**	CPDMA
Description	CPDMA_REGS receive buffer offset		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RX_BUFFER_OFFSET															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0
15:0	RX_BUFFER_OFFSET	Receive Buffer Offset Value - The RX_BUFFER_OFFSET will be written by the port into each frame SOP buffer descriptor buffer_offset field. The frame data will begin after the rx_buffer_offset value of bytes. A value of 0x0000 indicates that there are no unused bytes at the beginning of the data and that valid data begins on the first byte of the buffer. A value of 0x000F (decimal 15) indicates that the first 15 bytes of the buffer are to be ignored by the port and that valid buffer data starts on byte 16 of the buffer. This value is used for all channels.	RW	0x0

Table 18-994. Register Call Summary for Register CPDMA_RX_BUFFER_OFFSET

Gigabit Ethernet Switch (GMAC_SW)

- CPDMA RX and TX Interfaces: [0]
- RX Buffer Descriptors: [1]
- CPDMA Register Summary: [2]

Table 18-995. CPDMA_EMCONTROL

Address Offset	0x0000 002C		
Physical Address	0x4848 482C	**Instance**	CPDMA
Description	CPDMA_REGS emulation control		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														SOFT	FREE

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1	SOFT	Emulation Soft Bit	RW	0x0
0	FREE	Emulation Free Bit	RW	0x0

Table 18-996. Register Call Summary for Register CPDMA_EMCONTROL

Gigabit Ethernet Switch (GMAC_SW)

- Emulation Control: [0]
- CPDMA Register Summary: [1]

Table 18-997. CPDMA_TX_PRI0_RATE

Address Offset	0x0000 0030		
Physical Address	0x4848 4830	**Instance**	CPDMA
Description	CPDMA_REGS transmit (ingress) priority 0 rate		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								PRIN_IDLE_CNT								RESERVED								PRIN_SEND_CNT							

Bits	Field Name	Description	Type	Reset
31:30	RESERVED		R	0x0
29:16	PRIN_IDLE_CNT	Priority (7:0) idle count	RW	0x0
15:14	RESERVED		R	0x0
13:0	PRIN_SEND_CNT	Priority (7:0) send count	RW	0x0

Table 18-998. Register Call Summary for Register CPDMA_TX_PRI0_RATE

Gigabit Ethernet Switch (GMAC_SW)

- CPDMA Register Summary: [0]

Table 18-999. CPDMA_TX_PRI1_RATE

Address Offset	0x0000 0034		
Physical Address	0x4848 4834	**Instance**	CPDMA
Description	CPDMA_REGS transmit (ingress) priority 1 rate		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								PRIN_IDLE_CNT								RESERVED								PRIN_SEND_CNT							

Bits	Field Name	Description	Type	Reset
31:30	RESERVED		R	0x0
29:16	PRIN_IDLE_CNT	Priority (7:0) idle count	RW	0x0
15:14	RESERVED		R	0x0
13:0	PRIN_SEND_CNT	Priority (7:0) send count	RW	0x0

Table 18-1000. Register Call Summary for Register CPDMA_TX_PRI1_RATE

Gigabit Ethernet Switch (GMAC_SW)

- CPDMA Register Summary: [0]

Table 18-1001. CPDMA_TX_PRI2_RATE

Address Offset	0x0000 0038		
Physical Address	0x4848 4838	**Instance**	CPDMA
Description	CPDMA_REGS transmit (ingress) priority 2 rate		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								PRIN_IDLE_CNT								RESERVED								PRIN_SEND_CNT							

Bits	Field Name	Description	Type	Reset
31:30	RESERVED		R	0x0
29:16	PRIN_IDLE_CNT	Priority (7:0) idle count	RW	0x0
15:14	RESERVED		R	0x0
13:0	PRIN_SEND_CNT	Priority (7:0) send count	RW	0x0

Table 18-1002. Register Call Summary for Register CPDMA_TX_PRI2_RATE

Gigabit Ethernet Switch (GMAC_SW)
- CPDMA Register Summary: [0]

Table 18-1003. CPDMA_TX_PRI3_RATE

Address Offset	0x0000 003C		
Physical Address	0x4848 483C	**Instance**	CPDMA
Description	CPDMA_REGS transmit (ingress) priority 3 rate		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								PRIN_IDLE_CNT								RESERVED								PRIN_SEND_CNT							

Bits	Field Name	Description	Type	Reset
31:30	RESERVED		R	0x0
29:16	PRIN_IDLE_CNT	Priority (7:0) idle count	RW	0x0
15:14	RESERVED		R	0x0
13:0	PRIN_SEND_CNT	Priority (7:0) send count	RW	0x0

Table 18-1004. Register Call Summary for Register CPDMA_TX_PRI3_RATE

Gigabit Ethernet Switch (GMAC_SW)
- CPDMA Register Summary: [0]

Table 18-1005. CPDMA_TX_PRI4_RATE

Address Offset	0x0000 0040		
Physical Address	0x4848 4840	**Instance**	CPDMA
Description	CPDMA_REGS transmit (ingress) priority 4 rate		
Type	RW		

31 30	29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14	13 12 11 10 9 8	7 6 5 4 3 2 1 0
RESERVED		PRIN_IDLE_CNT	RESERVED		PRIN_SEND_CNT

Bits	Field Name	Description	Type	Reset
31:30	RESERVED		R	0x0
29:16	PRIN_IDLE_CNT	Priority (7:0) idle count	RW	0x0
15:14	RESERVED		R	0x0
13:0	PRIN_SEND_CNT	Priority (7:0) send count	RW	0x0

Table 18-1006. Register Call Summary for Register CPDMA_TX_PRI4_RATE

Gigabit Ethernet Switch (GMAC_SW)
- CPDMA Register Summary: [0]

Table 18-1007. CPDMA_TX_PRI5_RATE

Address Offset	0x0000 0044		
Physical Address	0x4848 4844	**Instance**	CPDMA
Description	CPDMA_REGS transmit (ingress) priority 5 rate		
Type	RW		

31 30	29 28 27 26 25 24	23 22 21 20 19 18 17 16	15 14	13 12 11 10 9 8	7 6 5 4 3 2 1 0
RESERVED		PRIN_IDLE_CNT	RESERVED		PRIN_SEND_CNT

Bits	Field Name	Description	Type	Reset
31:30	RESERVED		R	0x0
29:16	PRIN_IDLE_CNT	Priority (7:0) idle count	RW	0x0
15:14	RESERVED		R	0x0
13:0	PRIN_SEND_CNT	Priority (7:0) send count	RW	0x0

Table 18-1008. Register Call Summary for Register CPDMA_TX_PRI5_RATE

Gigabit Ethernet Switch (GMAC_SW)
- CPDMA Register Summary: [0]

Table 18-1009. CPDMA_TX_PRI6_RATE

Address Offset	0x0000 0048		
Physical Address	0x4848 4848	**Instance**	CPDMA
Description	CPDMA_REGS TRANSMIT (INGRESS) PRIORITY 6 RATE		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								PRIN_IDLE_CNT								RESERVED								PRIN_SEND_CNT							

Bits	Field Name	Description	Type	Reset
31:30	RESERVED		R	0x0
29:16	PRIN_IDLE_CNT	Priority (7:0) idle count	RW	0x0
15:14	RESERVED		R	0x0
13:0	PRIN_SEND_CNT	Priority (7:0) send count	RW	0x0

Table 18-1010. Register Call Summary for Register CPDMA_TX_PRI6_RATE

Gigabit Ethernet Switch (GMAC_SW)

- Audio Video Bridging: [0]
- CPDMA Register Summary: [1]

Table 18-1011. CPDMA_TX_PRI7_RATE

Address Offset	0x0000 004C		
Physical Address	0x4848 484C	**Instance**	CPDMA
Description	CPDMA_REGS transmit (ingress) priority 7 rate		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								PRIN_IDLE_CNT								RESERVED								PRIN_SEND_CNT							

Bits	Field Name	Description	Type	Reset
31:30	RESERVED		R	0x0
29:16	PRIN_IDLE_CNT	Priority (7:0) idle count	RW	0x0
15:14	RESERVED		R	0x0
13:0	PRIN_SEND_CNT	Priority (7:0) send count	RW	0x0

Table 18-1012. Register Call Summary for Register CPDMA_TX_PRI7_RATE

Gigabit Ethernet Switch (GMAC_SW)

- Audio Video Bridging: [0]
- CPDMA Register Summary: [1]

Table 18-1013. CPDMA_TX_INTSTAT_RAW

Address Offset	0x0000 0080		
Physical Address	0x4848 4880	**Instance**	CPDMA
Description	CPDMA_INT TX interrupt status register (raw value)		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								TX7_PEND	TX6_PEND	TX5_PEND	TX4_PEND	TX3_PEND	TX2_PEND	TX1_PEND	TX0_PEND

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x0
7	TX7_PEND	TX7_PEND raw int read (before mask).	R	0x0
6	TX6_PEND	TX6_PEND raw int read (before mask).	R	0x0
5	TX5_PEND	TX5_PEND raw int read (before mask).	R	0x0
4	TX4_PEND	TX4_PEND raw int read (before mask).	R	0x0
3	TX3_PEND	TX3_PEND raw int read (before mask).	R	0x0
2	TX2_PEND	TX2_PEND raw int read (before mask).	R	0x0
1	TX1_PEND	TX1_PEND raw int read (before mask).	R	0x0
0	TX0_PEND	TX0_PEND raw int read (before mask).	R	0x0

Table 18-1014. Register Call Summary for Register CPDMA_TX_INTSTAT_RAW

Gigabit Ethernet Switch (GMAC_SW)

- Transmit Packet Completion Pulse Interrupt (TX_PULSE): [0]
- Miscellaneous Pulse Interrupt (MISC_PULSE): [1][2]
- CPDMA Register Summary: [3]

Table 18-1015. CPDMA_TX_INTSTAT_MASKED

Address Offset	0x0000 0084		
Physical Address	0x4848 4884	**Instance**	CPDMA
Description	CPDMA_INT TX interrupt status register (masked value)		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								TX7_PEND	TX6_PEND	TX5_PEND	TX4_PEND	TX3_PEND	TX2_PEND	TX1_PEND	TX0_PEND

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x0
7	TX7_PEND	TX7_PEND masked interrupt read.	R	0x0
6	TX6_PEND	TX6_PEND masked interrupt read.	R	0x0
5	TX5_PEND	TX5_PEND masked interrupt read.	R	0x0
4	TX4_PEND	TX4_PEND masked interrupt read.	R	0x0
3	TX3_PEND	TX3_PEND masked interrupt read.	R	0x0
2	TX2_PEND	TX2_PEND masked interrupt read.	R	0x0
1	TX1_PEND	TX1_PEND masked interrupt read.	R	0x0

Bits	Field Name	Description	Type	Reset
0	TX0_PEND	TX0_PEND masked interrupt read.	R	0x0

Table 18-1016. Register Call Summary for Register CPDMA_TX_INTSTAT_MASKED

Gigabit Ethernet Switch (GMAC_SW)

- Transmit Packet Completion Pulse Interrupt (TX_PULSE): [0]
- Miscellaneous Pulse Interrupt (MISC_PULSE): [1][2]
- CPDMA Register Summary: [3]

Table 18-1017. CPDMA_TX_INTMASK_SET

Address Offset	0x0000 0088		
Physical Address	0x4848 4888	**Instance**	CPDMA
Description	CPDMA_INT TX interrupt mask set register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								TX7_MASK	TX6_MASK	TX5_MASK	TX4_MASK	TX3_MASK	TX2_MASK	TX1_MASK	TX0_MASK

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x0
7	TX7_MASK	TX Channel 7 Mask - Write one to enable interrupt.	RW	0x0
6	TX6_MASK	TX Channel 6 Mask - Write one to enable interrupt.	RW	0x0
5	TX5_MASK	TX Channel 5 Mask - Write one to enable interrupt.	RW	0x0
4	TX4_MASK	TX Channel 4 Mask - Write one to enable interrupt.	RW	0x0
3	TX3_MASK	TX Channel 3 Mask - Write one to enable interrupt.	RW	0x0
2	TX2_MASK	TX Channel 2 Mask - Write one to enable interrupt.	RW	0x0
1	TX1_MASK	TX Channel 1 Mask - Write one to enable interrupt.	RW	0x0
0	TX0_MASK	TX Channel 0 Mask - Write one to enable interrupt.	RW	0x0

Table 18-1018. Register Call Summary for Register CPDMA_TX_INTMASK_SET

Gigabit Ethernet Switch (GMAC_SW)

- Transmit Packet Completion Pulse Interrupt (TX_PULSE): [0]
- CPDMA RX and TX Interfaces: [1]
- CPDMA Register Summary: [2]

Table 18-1019. CPDMA_TX_INTMASK_CLEAR

Address Offset	0x0000 008C		
Physical Address	0x4848 488C	**Instance**	CPDMA
Description	CPDMA_INT TX Interrupt mask clear register		
Type	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								TX7_MASK	TX6_MASK	TX5_MASK	TX4_MASK	TX3_MASK	TX2_MASK	TX1_MASK	TX0_MASK

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x0
7	TX7_MASK	TX Channel 7 Mask - Write one to disable interrupt.	RW	0x0
6	TX6_MASK	TX Channel 6 Mask - Write one to disable interrupt.	RW	0x0
5	TX5_MASK	TX Channel 5 Mask - Write one to disable interrupt.	RW	0x0
4	TX4_MASK	TX Channel 4 Mask - Write one to disable interrupt.	RW	0x0
3	TX3_MASK	TX Channel 3 Mask - Write one to disable interrupt.	RW	0x0
2	TX2_MASK	TX Channel 2 Mask - Write one to disable interrupt.	RW	0x0
1	TX1_MASK	TX Channel 1 Mask - Write one to disable interrupt.	RW	0x0
0	TX0_MASK	TX Channel 0 Mask - Write one to disable interrupt.	RW	0x0

Table 18-1020. Register Call Summary for Register CPDMA_TX_INTMASK_CLEAR

Gigabit Ethernet Switch (GMAC_SW)
- Transmit Packet Completion Pulse Interrupt (TX_PULSE): [0]
- CPDMA Register Summary: [1]

Table 18-1021. CPDMA_IN_VECTOR

Address Offset	0x0000 0090		
Physical Address	0x4848 4890	**Instance**	CPDMA
Description	CPDMA_INT input vector (read only)		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DMA_IN_VECTOR																															

Bits	Field Name	Description	Type	Reset
31:0	DMA_IN_VECTOR	DMA Input Vector - The value of DMA_IN_VECTOR is reset to zero, but will change to the IN_VECTOR bus value one clock after reset is deasserted. Thereafter, this value will change to a new IN_VECTOR value one clock after the IN_VECTOR value changes.	R	0x0

Table 18-1022. Register Call Summary for Register CPDMA_IN_VECTOR

Gigabit Ethernet Switch (GMAC_SW)
- CPDMA Register Summary: [0]

Table 18-1023. CPDMA_EOI_VECTOR

Address Offset	0x0000 0094		
Physical Address	0x4848 4894	**Instance**	CPDMA
Description	CPDMA_INT end of interrupt vector		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														DMA_EOI_VECTOR	

Bits	Field Name	Description	Type	Reset
31:5	RESERVED		R	0x0
4:0	DMA_EOI_VECTOR	DMA End of Interrupt Vector - The EOI_VECTOR(4:0) pins reflect the value written to this location one MAIN_CLK cycle after a write to this location. The EOI_WR signal is asserted for a single clock cycle after a latency of two MAIN_CLK cycles when a write is performed to this location.	RW	0x0

Table 18-1024. Register Call Summary for Register CPDMA_EOI_VECTOR

Gigabit Ethernet Switch (GMAC_SW)

- Receive Packet Completion Pulse Interrupt (RX_PULSE): [0]
- Transmit Packet Completion Pulse Interrupt (TX_PULSE): [1]
- Receive Threshold Pulse Interrupt (RX_THRESH_PULSE): [2]
- Miscellaneous Pulse Interrupt (MISC_PULSE): [3]
- CPDMA Register Summary: [4]

Table 18-1025. CPDMA_RX_INTSTAT_RAW

Address Offset	0x0000 00A0		
Physical Address	0x4848 48A0	**Instance**	CPDMA
Description	CPDMA_INT RX Interrupt status register (raw value)		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RX7_THRESH_PEND	RX6_THRESH_PEND	RX5_THRESH_PEND	RX4_THRESH_PEND	RX3_THRESH_PEND	RX2_THRESH_PEND	RX1_THRESH_PEND	RX0_THRESH_PEND	RX7_PEND	RX6_PEND	RX5_PEND	RX4_PEND	RX3_PEND	RX2_PEND	RX1_PEND	RX0_PEND

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0
15	RX7_THRESH_PEND	RX7_THRESH_PEND raw int read (before mask).	R	0x0
14	RX6_THRESH_PEND	RX6_THRESH_PEND raw int read (before mask).	R	0x0
13	RX5_THRESH_PEND	RX5_THRESH_PEND raw int read (before mask).	R	0x0
12	RX4_THRESH_PEND	RX4_THRESH_PEND raw int read (before mask).	R	0x0
11	RX3_THRESH_PEND	RX3_THRESH_PEND raw int read (before mask).	R	0x0
10	RX2_THRESH_PEND	RX2_THRESH_PEND raw int read (before mask).	R	0x0
9	RX1_THRESH_PEND	RX1_THRESH_PEND raw int read (before mask).	R	0x0
8	RX0_THRESH_PEND	RX0_THRESH_PEND raw int read (before mask).	R	0x0
7	RX7_PEND	RX7_PEND raw int read (before mask).	R	0x0
6	RX6_PEND	RX6_PEND raw int read (before mask).	R	0x0
5	RX5_PEND	RX5_PEND raw int read (before mask).	R	0x0
4	RX4_PEND	RX4_PEND raw int read (before mask).	R	0x0
3	RX3_PEND	RX3_PEND raw int read (before mask).	R	0x0
2	RX2_PEND	RX2_PEND raw int read (before mask).	R	0x0
1	RX1_PEND	RX1_PEND raw int read (before mask).	R	0x0
0	RX0_PEND	RX0_PEND raw int read (before mask).	R	0x0

Table 18-1026. Register Call Summary for Register CPDMA_RX_INTSTAT_RAW

Gigabit Ethernet Switch (GMAC_SW)

- Receive Packet Completion Pulse Interrupt (RX_PULSE): [0]
- Receive Threshold Pulse Interrupt (RX_THRESH_PULSE): [1]
- CPDMA Register Summary: [2]

Table 18-1027. CPDMA_RX_INTSTAT_MASKED

Address Offset	0x0000 00A4		
Physical Address	0x4848 48A4	**Instance**	CPDMA
Description	CPDMA_INT RX interrupt status register (masked value)		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RX7_THRESH_PEND	RX6_THRESH_PEND	RX5_THRESH_PEND	RX4_THRESH_PEND	RX3_THRESH_PEND	RX2_THRESH_PEND	RX1_THRESH_PEND	RX0_THRESH_PEND	RX7_PEND	RX6_PEND	RX5_PEND	RX4_PEND	RX3_PEND	RX2_PEND	RX1_PEND	RX0_PEND

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0
15	RX7_THRESH_PEND	RX7_THRESH_PEND masked int read.	R	0x0
14	RX6_THRESH_PEND	RX6_THRESH_PEND masked int read.	R	0x0
13	RX5_THRESH_PEND	RX5_THRESH_PEND masked int read.	R	0x0
12	RX4_THRESH_PEND	RX4_THRESH_PEND masked int read.	R	0x0
11	RX3_THRESH_PEND	RX3_THRESH_PEND masked int read.	R	0x0
10	RX2_THRESH_PEND	RX2_THRESH_PEND masked int read.	R	0x0
9	RX1_THRESH_PEND	RX1_THRESH_PEND masked int read.	R	0x0
8	RX0_THRESH_PEND	RX0_THRESH_PEND masked int read.	R	0x0
7	RX7_PEND	RX7_PEND masked int read.	R	0x0
6	RX6_PEND	RX6_PEND masked int read.	R	0x0
5	RX5_PEND	RX5_PEND masked int read.	R	0x0
4	RX4_PEND	RX4_PEND masked int read.	R	0x0
3	RX3_PEND	RX3_PEND masked int read.	R	0x0
2	RX2_PEND	RX2_PEND masked int read.	R	0x0
1	RX1_PEND	RX1_PEND masked int read.	R	0x0
0	RX0_PEND	RX0_PEND masked int read.	R	0x0

Table 18-1028. Register Call Summary for Register CPDMA_RX_INTSTAT_MASKED

Gigabit Ethernet Switch (GMAC_SW)

- Receive Packet Completion Pulse Interrupt (RX_PULSE): [0]
- Receive Threshold Pulse Interrupt (RX_THRESH_PULSE): [1]
- CPDMA Register Summary: [2]

Table 18-1029. CPDMA_RX_INTMASK_SET

Address Offset	0x0000 00A8		
Physical Address	0x4848 48A8	**Instance**	CPDMA
Description	CPDMA_INT RX interrupt mask set register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RX7_THRESH_PEND_MASK	RX6_THRESH_PEND_MASK	RX5_THRESH_PEND_MASK	RX4_THRESH_PEND_MASK	RX3_THRESH_PEND_MASK	RX2_THRESH_PEND_MASK	RX1_THRESH_PEND_MASK	RX0_THRESH_PEND_MASK	RX7_PEND_MASK	RX6_PEND_MASK	RX5_PEND_MASK	RX4_PEND_MASK	RX3_PEND_MASK	RX2_PEND_MASK	RX1_PEND_MASK	RX0_PEND_MASK

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0
15	RX7_THRESH_PEND_MASK	RX Channel 7 Threshold Pending Int. Mask - Write one to enable Int.	RW	0x0
14	RX6_THRESH_PEND_MASK	RX Channel 6 Threshold Pending Int. Mask - Write one to enable Int.	RW	0x0
13	RX5_THRESH_PEND_MASK	RX Channel 5 Threshold Pending Int. Mask - Write one to enable Int.	RW	0x0
12	RX4_THRESH_PEND_MASK	RX Channel 4 Threshold Pending Int. Mask - Write one to enable Int.	RW	0x0
11	RX3_THRESH_PEND_MASK	RX Channel 3 Threshold Pending Int. Mask - Write one to enable Int.	RW	0x0
10	RX2_THRESH_PEND_MASK	RX Channel 2 Threshold Pending Int. Mask - Write one to enable Int.	RW	0x0
9	RX1_THRESH_PEND_MASK	RX Channel 1 Threshold Pending Int. Mask - Write one to enable Int.	RW	0x0
8	RX0_THRESH_PEND_MASK	RX Channel 0 Threshold Pending Int. Mask - Write one to enable Int.	RW	0x0
7	RX7_PEND_MASK	RX Channel 7 Pending Int. Mask - Write one to enable Int.	RW	0x0
6	RX6_PEND_MASK	RX Channel 6 Pending Int. Mask - Write one to enable Int.	RW	0x0
5	RX5_PEND_MASK	RX Channel 5 Pending Int. Mask - Write one to enable Int.	RW	0x0
4	RX4_PEND_MASK	RX Channel 4 Pending Int. Mask - Write one to enable Int.	RW	0x0
3	RX3_PEND_MASK	RX Channel 3 Pending Int. Mask - Write one to enable Int.	RW	0x0
2	RX2_PEND_MASK	RX Channel 2 Pending Int. Mask - Write one to enable Int.	RW	0x0
1	RX1_PEND_MASK	RX Channel 1 Pending Int. Mask - Write one to enable Int.	RW	0x0
0	RX0_PEND_MASK	RX Channel 0 Pending Int. Mask - Write one to enable Int.	RW	0x0

Table 18-1030. Register Call Summary for Register CPDMA_RX_INTMASK_SET

Gigabit Ethernet Switch (GMAC_SW)

- Receive Packet Completion Pulse Interrupt (RX_PULSE): [0]
- Receive Threshold Pulse Interrupt (RX_THRESH_PULSE): [1]
- CPDMA RX and TX Interfaces: [2]
- CPDMA Register Summary: [3]

Table 18-1031. CPDMA_RX_INTMASK_CLEAR

Address Offset	0x0000 00AC		
Physical Address	0x4848 48AC	**Instance**	CPDMA
Description	CPDMA_INT RX interrupt mask clear register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RX7_THRESH_PEND_MASK	RX6_THRESH_PEND_MASK	RX5_THRESH_PEND_MASK	RX4_THRESH_PEND_MASK	RX3_THRESH_PEND_MASK	RX2_THRESH_PEND_MASK	RX1_THRESH_PEND_MASK	RX0_THRESH_PEND_MASK	RX7_PEND_MASK	RX6_PEND_MASK	RX5_PEND_MASK	RX4_PEND_MASK	RX3_PEND_MASK	RX2_PEND_MASK	RX1_PEND_MASK	RX0_PEND_MASK

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0
15	RX7_THRESH_PEND_MASK	RX Channel 7 Threshold Pending Int. Mask - Write one to disable Int.	RW	0x0
14	RX6_THRESH_PEND_MASK	RX Channel 6 Threshold Pending Int. Mask - Write one to disable Int.	RW	0x0
13	RX5_THRESH_PEND_MASK	RX Channel 5 Threshold Pending Int. Mask - Write one to disable Int.	RW	0x0
12	RX4_THRESH_PEND_MASK	RX Channel 4 Threshold Pending Int. Mask - Write one to disable Int.	RW	0x0
11	RX3_THRESH_PEND_MASK	RX Channel 3 Threshold Pending Int. Mask - Write one to disable Int.	RW	0x0
10	RX2_THRESH_PEND_MASK	RX Channel 2 Threshold Pending Int. Mask - Write one to disable Int.	RW	0x0
9	RX1_THRESH_PEND_MASK	RX Channel 1 Threshold Pending Int. Mask - Write one to disable Int.	RW	0x0
8	RX0_THRESH_PEND_MASK	RX Channel 0 Threshold Pending Int. Mask - Write one to disable Int.	RW	0x0
7	RX7_PEND_MASK	RX Channel 7 Pending Int. Mask - Write one to disable Int.	RW	0x0
6	RX6_PEND_MASK	RX Channel 6 Pending Int. Mask - Write one to disable Int.	RW	0x0
5	RX5_PEND_MASK	RX Channel 5 Pending Int. Mask - Write one to disable Int.	RW	0x0
4	RX4_PEND_MASK	RX Channel 4 Pending Int. Mask - Write one to disable Int.	RW	0x0
3	RX3_PEND_MASK	RX Channel 3 Pending Int. Mask - Write one to disable Int.	RW	0x0
2	RX2_PEND_MASK	RX Channel 2 Pending Int. Mask - Write one to disable Int.	RW	0x0

Bits	Field Name	Description	Type	Reset
1	RX1_PEND_MASK	RX Channel 1 Pending Int. Mask - Write one to disable Int.	RW	0x0
0	RX0_PEND_MASK	RX Channel 0 Pending Int. Mask - Write one to disable Int.	RW	0x0

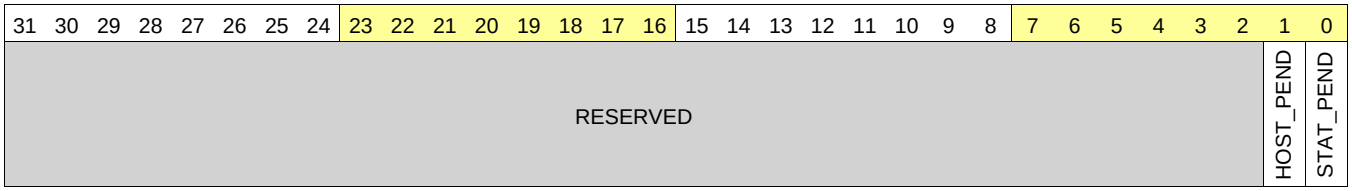
Table 18-1032. Register Call Summary for Register CPDMA_RX_INTMASK_CLEAR

Gigabit Ethernet Switch (GMAC_SW)

- Receive Packet Completion Pulse Interrupt (RX_PULSE): [0]
- Receive Threshold Pulse Interrupt (RX_THRESH_PULSE): [1]
- CPDMA Register Summary: [2]

Table 18-1033. CPDMA_DMA_INTSTAT_RAW

Address Offset	0x0000 00B0		
Physical Address	0x4848 48B0	**Instance**	CPDMA
Description	CPDMA_INT DMA interrupt status register (raw value)		
Type	R		



Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1	HOST_PEND	Host Pending Interrupt - raw int read (before mask).	R	0x0
0	STAT_PEND	Statistics Pending Interrupt - raw int read (before mask).	R	0x0

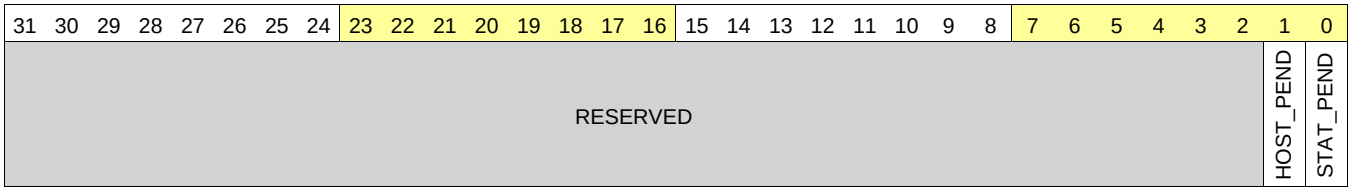
Table 18-1034. Register Call Summary for Register CPDMA_DMA_INTSTAT_RAW

Gigabit Ethernet Switch (GMAC_SW)

- CPDMA Register Summary: [0]

Table 18-1035. CPDMA_DMA_INTSTAT_MASKED

Address Offset	0x0000 00B4		
Physical Address	0x4848 48B4	**Instance**	CPDMA
Description	CPDMA_INT DMA interrupt status register (masked value)		
Type	R		



Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1	HOST_PEND	Host Pending Interrupt - masked interrupt read.	R	0x0
0	STAT_PEND	Statistics Pending Interrupt - masked interrupt read.	R	0x0

Table 18-1036. Register Call Summary for Register CPDMA_DMA_INTSTAT_MASKED

Gigabit Ethernet Switch (GMAC_SW)

- CPDMA Register Summary: [0]

Table 18-1037. CPDMA_DMA_INTMASK_SET

Address Offset	0x0000 00B8		
Physical Address	0x4848 48B8	**Instance**	CPDMA
Description	CPDMA_INT DMA interrupt mask set register		
Type	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														HOST_ERR_INT_MASK	STAT_INT_MASK

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1	HOST_ERR_INT_MASK	Host Error Interrupt Mask - Write one to enable interrupt.	W	0x0
0	STAT_INT_MASK	Statistics Interrupt Mask - Write one to enable interrupt.	R	0x0

Table 18-1038. Register Call Summary for Register CPDMA_DMA_INTMASK_SET

Gigabit Ethernet Switch (GMAC_SW)

- Miscellaneous Pulse Interrupt (MISC_PULSE): [0][1]
- CPDMA Register Summary: [2]

Table 18-1039. CPDMA_DMA_INTMASK_CLEAR

Address Offset	0x0000 00BC		
Physical Address	0x4848 48BC	**Instance**	CPDMA
Description	CPDMA_INT DMA interrupt mask clear register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														HOST_ERR_INT_MASK	STAT_INT_MASK

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1	HOST_ERR_INT_MASK	Host Error Interrupt Mask - Write one to disable interrupt.	RW	0x0
0	STAT_INT_MASK	Statistics Interrupt Mask - Write one to disable interrupt.	RW	0x0

Table 18-1040. Register Call Summary for Register CPDMA_DMA_INTMASK_CLEAR

Gigabit Ethernet Switch (GMAC_SW)

- Miscellaneous Pulse Interrupt (MISC_PULSE): [0]
- CPDMA Register Summary: [1]

Table 18-1041. CPDMA_RX0_PENDTHRESH

Address Offset	0x0000 00C0		
Physical Address	0x4848 48C0	**Instance**	CPDMA
Description	CPDMA_INT receive threshold pending register channel 0		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								RX_PENDTHRESH							

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x0
7:0	RX_PENDTHRESH	Rx Flow Threshold - This field contains the threshold value for issuing receive threshold pending interrupts (when enabled).	RW	0x0

Table 18-1042. Register Call Summary for Register CPDMA_RX0_PENDTHRESH

Gigabit Ethernet Switch (GMAC_SW)

- CPDMA Register Summary: [0]
- CPDMA Register Description: [1]

Table 18-1043. CPDMA_RX1_PENDTHRESH

Address Offset	0x0000 00C4		
Physical Address	0x4848 48C4	**Instance**	CPDMA
Description	CPDMA_INT receive threshold pending register channel 1		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								RX_PENDTHRESH							

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x0
7:0	RX_PENDTHRESH	Rx Flow Threshold - This field contains the threshold value for issuing receive threshold pending interrupts (when enabled).	RW	0x0

Table 18-1044. Register Call Summary for Register CPDMA_RX1_PENDTHRESH

Gigabit Ethernet Switch (GMAC_SW)

- CPDMA Register Summary: [0]
- CPDMA Register Description: [1]

Table 18-1045. CPDMA_RX2_PENDTHRESH

Address Offset	0x0000 00C8		
Physical Address	0x4848 48C8	**Instance**	CPDMA
Description	CPDMA_INT receive threshold pending register channel 2		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								RX_PENDTHRESH							

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x0
7:0	RX_PENDTHRESH	Rx Flow Threshold - This field contains the threshold value for issuing receive threshold pending interrupts (when enabled).	RW	0x0

Table 18-1046. Register Call Summary for Register CPDMA_RX2_PENDTHRESH

Gigabit Ethernet Switch (GMAC_SW)

- CPDMA Register Summary: [0]
- CPDMA Register Description: [1]

Table 18-1047. CPDMA_RX3_PENDTHRESH

Address Offset	0x0000 00CC		
Physical Address	0x4848 48CC	**Instance**	CPDMA
Description	CPDMA_INT receive threshold pending register channel 3		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								RX_PENDTHRESH							

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x0
7:0	RX_PENDTHRESH	Rx Flow Threshold - This field contains the threshold value for issuing receive threshold pending interrupts (when enabled).	RW	0x0

Table 18-1048. Register Call Summary for Register CPDMA_RX3_PENDTHRESH

Gigabit Ethernet Switch (GMAC_SW)

- CPDMA Register Summary: [0]
- CPDMA Register Description: [1]

Table 18-1049. CPDMA_RX4_PENDTHRESH

Address Offset	0x0000 00D0		
Physical Address	0x4848 48D0	**Instance**	CPDMA
Description	CPDMA_INT receive threshold pending register channel 4		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								RX_PENDTHRESH							

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x0
7:0	RX_PENDTHRESH	Rx Flow Threshold - This field contains the threshold value for issuing receive threshold pending interrupts (when enabled).	RW	0x0

Table 18-1050. Register Call Summary for Register CPDMA_RX4_PENDTHRESH

Gigabit Ethernet Switch (GMAC_SW)

- CPDMA Register Summary: [0]
- CPDMA Register Description: [1]

Table 18-1051. CPDMA_RX5_PENDTHRESH

Address Offset	0x0000 00D4		
Physical Address	0x4848 48D4	**Instance**	CPDMA
Description	CPDMA_INT receive threshold pending register channel 5		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								RX_PENDTHRESH							

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x0
7:0	RX_PENDTHRESH	Rx Flow Threshold - This field contains the threshold value for issuing receive threshold pending interrupts (when enabled).	RW	0x0

Table 18-1052. Register Call Summary for Register CPDMA_RX5_PENDTHRESH

Gigabit Ethernet Switch (GMAC_SW)

- CPDMA Register Summary: [0]
- CPDMA Register Description: [1]

Table 18-1053. CPDMA_RX6_PENDTHRESH

Address Offset	0x0000 00D8		
Physical Address	0x4848 48D8	**Instance**	CPDMA
Description	CPDMA_INT receive threshold pending register channel 6		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								RX_PENDTHRESH							

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x0
7:0	RX_PENDTHRESH	Rx Flow Threshold - This field contains the threshold value for issuing receive threshold pending interrupts (when enabled).	RW	0x0

Table 18-1054. Register Call Summary for Register CPDMA_RX6_PENDTHRESH

Gigabit Ethernet Switch (GMAC_SW)

- CPDMA Register Summary: [0]
- CPDMA Register Description: [1]

Table 18-1055. CPDMA_RX7_PENDTHRESH

Address Offset	0x0000 00DC		
Physical Address	0x4848 48DC	**Instance**	CPDMA
Description	CPDMA_INT receive threshold pending register channel 7		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								RX_PENDTHRESH							

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x0
7:0	RX_PENDTHRESH	Rx Flow Threshold - This field contains the threshold value for issuing receive threshold pending interrupts (when enabled).	RW	0x0

Table 18-1056. Register Call Summary for Register CPDMA_RX7_PENDTHRESH

Gigabit Ethernet Switch (GMAC_SW)

- CPDMA Register Summary: [0]
- CPDMA Register Description: [1]

Table 18-1057. CPDMA_RX0_FREEBUFFER

Address Offset	0x0000 00E0		
Physical Address	0x4848 48E0	**Instance**	CPDMA
Description	CPDMA_INT receive free buffer register channel 0		
Type	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RX_FREEBUFFER															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0
15:0	RX_FREEBUFFER	Rx Free Buffer Count - This field contains the count of free buffers available. The CPDMA_RX0_PENDTHRESH[7:0] RX_PENDTHRESH value is compared with this field to determine if the receive threshold pending interrupt should be asseted (if enabled). This is a write to increment field. This field rolls over to zero on overflow. If receive threshold pending interrupts are used, the host must initialize this field to the number of available buffers (one register per channel). The port decrements (by the number of buffers in the received frame) the associated channel register for each received frame. This is a write to increment field. The host must write this field with the number of buffers that have been freed due to host processing.	W	0x0

Table 18-1058. Register Call Summary for Register CPDMA_RX0_FREEBUFFER

Gigabit Ethernet Switch (GMAC_SW)

- CPDMA Register Summary: [0]

Table 18-1059. CPDMA_RX1_FREEBUFFER

Address Offset	0x0000 00E4		
Physical Address	0x4848 48E4	**Instance**	CPDMA
Description	CPDMA_INT receive free buffer register channel 1		
Type	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RX_FREEBUFFER															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0
15:0	RX_FREEBUFFER	Rx Free Buffer Count - This field contains the count of free buffers available. The CPDMA_RX1_PENDTHRESH[7:0] RX_PENDTHRESH value is compared with this field to determine if the receive threshold pending interrupt should be asseted (if enabled). This is a write to increment field. This field rolls over to zero on overflow. If receive threshold pending interrupts are used, the host must initialize this field to the number of available buffers (one register per channel). The port decrements (by the number of buffers in the received frame) the associated channel register for each received frame. This is a write to increment field. The host must write this field with the number of buffers that have been freed due to host processing.	W	0x0

Table 18-1060. Register Call Summary for Register CPDMA_RX1_FREEBUFFER

Gigabit Ethernet Switch (GMAC_SW)

- CPDMA Register Summary: [0]

Table 18-1061. CPDMA_RX2_FREEBUFFER

Address Offset	0x0000 00E8		
Physical Address	0x4848 48E8	**Instance**	CPDMA
Description	CPDMA_INT receive free buffer register channel 2		
Type	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RX_FREEBUFFER															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0
15:0	RX_FREEBUFFER	Rx Free Buffer Count - This field contains the count of free buffers available. The CPDMA_RX2_PENDTHRESH[7:0] RX_PENDTHRESH value is compared with this field to determine if the receive threshold pending interrupt should be asseted (if enabled). This is a write to increment field. This field rolls over to zero on overflow. If receive threshold pending interrupts are used, the host must initialize this field to the number of available buffers (one register per channel). The port decrements (by the number of buffers in the received frame) the associated channel register for each received frame. This is a write to increment field. The host must write this field with the number of buffers that have been freed due to host processing.	W	0x0

Table 18-1062. Register Call Summary for Register CPDMA_RX2_FREEBUFFER

Gigabit Ethernet Switch (GMAC_SW)

- CPDMA Register Summary: [0]

Table 18-1063. CPDMA_RX3_FREEBUFFER

Address Offset	0x0000 00EC		
Physical Address	0x4848 48EC	**Instance**	CPDMA
Description	CPDMA_INT receive free buffer register channel 3		
Type	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RX_FREEBUFFER															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0
15:0	RX_FREEBUFFER	Rx Free Buffer Count - This field contains the count of free buffers available. The CPDMA_RX3_PENDTHRESH[7:0] RX_PENDTHRESH value is compared with this field to determine if the receive threshold pending interrupt should be asseted (if enabled). This is a write to increment field. This field rolls over to zero on overflow. If receive threshold pending interrupts are used, the host must initialize this field to the number of available buffers (one register per channel). The port decrements (by the number of buffers in the received frame) the associated channel register for each received frame. This is a write to increment field. The host must write this field with the number of buffers that have been freed due to host processing.	W	0x0

Table 18-1064. Register Call Summary for Register CPDMA_RX3_FREEBUFFER

Gigabit Ethernet Switch (GMAC_SW)
- CPDMA Register Summary: [0]

Table 18-1065. CPDMA_RX4_FREEBUFFER

Address Offset	0x0000 00F0		
Physical Address	0x4848 48F0	**Instance**	CPDMA
Description	CPDMA_INT receive free buffer register channel 4		
Type	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RX_FREEBUFFER															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0
15:0	RX_FREEBUFFER	Rx Free Buffer Count - This field contains the count of free buffers available. The CPDMA_RX4_PENDTHRESH[7:0] RX_PENDTHRESH value is compared with this field to determine if the receive threshold pending interrupt should be asseted (if enabled). This is a write to increment field. This field rolls over to zero on overflow. If receive threshold pending interrupts are used, the host must initialize this field to the number of available buffers (one register per channel). The port decrements (by the number of buffers in the received frame) the associated channel register for each received frame. This is a write to increment field. The host must write this field with the number of buffers that have been freed due to host processing.	W	0x0

Table 18-1066. Register Call Summary for Register CPDMA_RX4_FREEBUFFER

Gigabit Ethernet Switch (GMAC_SW)
- CPDMA Register Summary: [0]

Table 18-1067. CPDMA_RX5_FREEBUFFER

Address Offset	0x0000 00F4		
Physical Address	0x4848 48F4	**Instance**	CPDMA
Description	CPDMA_INT receive free buffer register channel 5		
Type	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RX_FREEBUFFER															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0
15:0	RX_FREEBUFFER	Rx Free Buffer Count - This field contains the count of free buffers available. The CPDMA_RX5_PENDTHRESH[7:0] RX_PENDTHRESH value is compared with this field to determine if the receive threshold pending interrupt should be asseted (if enabled). This is a write to increment field. This field rolls over to zero on overflow. If receive threshold pending interrupts are used, the host must initialize this field to the number of available buffers (one register per channel). The port decrements (by the number of buffers in the received frame) the associated channel register for each received frame. This is a write to increment field. The host must write this field with the number of buffers that have been freed due to host processing.	W	0x0

Table 18-1068. Register Call Summary for Register CPDMA_RX5_FREEBUFFER

Gigabit Ethernet Switch (GMAC_SW)

- CPDMA Register Summary: [0]

Table 18-1069. CPDMA_RX6_FREEBUFFER

Address Offset	0x0000 00F8		
Physical Address	0x4848 48F8	**Instance**	CPDMA
Description	CPDMA_INT receive free buffer register channel 6		
Type	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RX_FREEBUFFER															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0
15:0	RX_FREEBUFFER	Rx Free Buffer Count - This field contains the count of free buffers available. The CPDMA_RX6_PENDTHRESH[7:0] RX_PENDTHRESH value is compared with this field to determine if the receive threshold pending interrupt should be asseted (if enabled). This is a write to increment field. This field rolls over to zero on overflow. If receive threshold pending interrupts are used, the host must initialize this field to the number of available buffers (one register per channel). The port decrements (by the number of buffers in the received frame) the associated channel register for each received frame. This is a write to increment field. The host must write this field with the number of buffers that have been freed due to host processing.	W	0x0

Table 18-1070. Register Call Summary for Register CPDMA_RX6_FREEBUFFER

Gigabit Ethernet Switch (GMAC_SW)
- CPDMA Register Summary: [0]

Table 18-1071. CPDMA_RX7_FREEBUFFER

Address Offset	0x0000 00FC		
Physical Address	0x4848 48FC	**Instance**	CPDMA
Description	CPDMA_INT receive free buffer register channel 7		
Type	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RX_FREEBUFFER															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0
15:0	RX_FREEBUFFER	Rx Free Buffer Count - This field contains the count of free buffers available. The CPDMA_RX7_PENDTHRESH[7:0] RX_PENDTHRESH value is compared with this field to determine if the receive threshold pending interrupt should be asseted (if enabled). This is a write to increment field. This field rolls over to zero on overflow. If receive threshold pending interrupts are used, the host must initialize this field to the number of available buffers (one register per channel). The port decrements (by the number of buffers in the received frame) the associated channel register for each received frame. This is a write to increment field. The host must write this field with the number of buffers that have been freed due to host processing.	W	0x0

Table 18-1072. Register Call Summary for Register CPDMA_RX7_FREEBUFFER

Gigabit Ethernet Switch (GMAC_SW)
- CPDMA Register Summary: [0]

18.8.6.5 STATS Registers

18.8.6.5.1 STATS Register Summary

Table 18-1073. STATS Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	STATS Physical Address
GOOD_RX_FRAMES	RW	32	0x0000 0000	0x4848 4900
BROADCAST_RX_FRAMES	RW	32	0x0000 0004	0x4848 4904
MULTICAST_RX_FRAMES	RW	32	0x0000 0008	0x4848 4908
PAUSE_RX_FRAMES	RW	32	0x0000 000C	0x4848 490C
RX_CRC_ERRORS	RW	32	0x0000 0010	0x4848 4910
RX_ALIGN_CODE_ERRORS	RW	32	0x0000 0014	0x4848 4914
OVERSIZE_RX_FRAMES	RW	32	0x0000 0018	0x4848 4918
RX_JABBERS	RW	32	0x0000 001C	0x4848 491C
UNDERSIZE_RX_FRAMES	RW	32	0x0000 0020	0x4848 4920
RX_FRAGMENTS	RW	32	0x0000 0024	0x4848 4924
RX_OCTETS	RW	32	0x0000 0030	0x4848 4930

Table 18-1073. STATS Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	STATS Physical Address
GOOD_TX_FRAMES	RW	32	0x0000 0034	0x4848 4934
BROADCAST_TX_FRAMES	RW	32	0x0000 0038	0x4848 4938
MULTICAST_TX_FRAMES	RW	32	0x0000 003C	0x4848 493C
PAUSE_TX_FRAMES	RW	32	0x0000 0040	0x4848 4940
DEFERRED_TX_FRAMES	RW	32	0x0000 0044	0x4848 4944
COLLISIONS	RW	32	0x0000 0048	0x4848 4948
SINGLE_COLLISION_TX_FRAMES	RW	32	0x0000 004C	0x4848 494C
MULTIPLE_COLLISION_TX_FRAMES	RW	32	0x0000 0050	0x4848 4950
EXCESSIVE_COLLISIONS	RW	32	0x0000 0054	0x4848 4954
LATE_COLLISIONS	RW	32	0x0000 0058	0x4848 4958
TX_UNDERRUN	RW	32	0x0000 005C	0x4848 495C
CARRIER_SENSE_ERRORS	RW	32	0x0000 0060	0x4848 4960
TX_OCTETS	RW	32	0x0000 0064	0x4848 4964
RX_TX_64_OCTET_FRAMES	RW	32	0x0000 0068	0x4848 4968
RX_TX_65_127_OCTET_FRAMES	RW	32	0x0000 006C	0x4848 496C
RX_TX_128_255_OCTET_FRAMES	RW	32	0x0000 0070	0x4848 4970
RX_TX_256_511_OCTET_FRAMES	RW	32	0x0000 0074	0x4848 4974
RX_TX_512_1023_OCTET_FRAMES	RW	32	0x0000 0078	0x4848 4978
RX_TX_1024_UP_OCTET_FRAMES	RW	32	0x0000 007C	0x4848 497C
NET_OCTETS	RW	32	0x0000 0080	0x4848 4980
RX_START_OF_FRAME_OVERRUNS	RW	32	0x0000 0084	0x4848 4984
RX_MIDDLE_OF_FRAME_OVERRUNS	RW	32	0x0000 0088	0x4848 4988
RX_DMA_OVERRUNS	RW	32	0x0000 008C	0x4848 498C

18.8.6.5.2 STATS Register Description

Table 18-1074. GOOD_RX_FRAMES

Address Offset	0x0000 0000		
Physical Address	0x4848 4900	**Instance**	STATS
Description	The total number of good frames received on the port. A good frame is defined to be: - Any data or MAC control frame which matched a unicast, broadcast or multicast address, or matched due to promiscuous mode - Had a length of 64 to SL_RX_MAXLEN[13:0] RX_MAXLEN bytes inclusive - Had no CRC error, alignment error or code error. See the RX_ALIGN_CODE_ERRORS and RX_CRC_ERRORS statistic descriptions for definitions of alignment, code and CRC errors. Overruns have no effect upon this statistic.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VALUE																															

Bits	Field Name	Description	Type	Reset
31:0	VALUE	Statistic value	RW	0x0000 0000

Table 18-1075. Register Call Summary for Register GOOD_RX_FRAMES

Gigabit Ethernet Switch (GMAC_SW)

- STATS Register Summary: [0]

Table 18-1076. BROADCAST_RX_FRAMES

Address Offset	0x0000 0004		
Physical Address	0x4848 4904	**Instance**	STATS
Description	The total number of good broadcast frames received on the port. A good broadcast frame is defined to be: - Any data or MAC control frame which was destined for only address 0xFFFFFFFFFFFF - Had a length of SL_RX_MAXLEN[13:0] RX_MAXLEN bytes inclusive - Had no CRC error, alignment error or code error. See the RX_ALIGN_CODE_ERRORS and RX_CRC_ERRORS statistic descriptions for definitions of alignment, code and CRC errors. Overruns have no effect upon this statistic.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VALUE																															

Bits	Field Name	Description	Type	Reset
31:0	VALUE	Statistic value	RW	0x0000 0000

Table 18-1077. Register Call Summary for Register BROADCAST_RX_FRAMES

Gigabit Ethernet Switch (GMAC_SW)

- STATS Register Summary: [0]

Table 18-1078. MULTICAST_RX_FRAMES

Address Offset	0x0000 0008		
Physical Address	0x4848 4908	**Instance**	STATS
Description	The total number of good multicast frames received on the port. A good multicast frame is defined to be: - Any data or MAC control frame which was destined for any multicast address other than 0xFFFFFFFFFFFF - Had a length of SL_RX_MAXLEN[13:0] RX_MAXLEN bytes inclusive - Had no CRC error, alignment error or code error. See the RX_ALIGN_CODE_ERRORS and RX_CRC_ERRORS statistic descriptions for definitions of alignment, code and CRC errors. Overruns have no effect upon this statistic.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VALUE																															

Bits	Field Name	Description	Type	Reset
31:0	VALUE	Statistic value	RW	0x0000 0000

Table 18-1079. Register Call Summary for Register MULTICAST_RX_FRAMES

Gigabit Ethernet Switch (GMAC_SW)

- STATS Register Summary: [0]

Table 18-1080. PAUSE_RX_FRAMES

Address Offset	0x0000 000C		
Physical Address	0x4848 490C	**Instance**	STATS
Description	The total number of IEEE 802.3X pause frames received by the port (whether acted upon or not). Such a frame: - Contained any unicast, broadcast, or multicast address - Contained the length/type field value 88.08 (hex) and the opcode 0x0001 - Was of length 64 to SL_RX_MAXLEN[13:0] RX_MAXLEN bytes inclusive - Had no CRC error, alignment error or code error - Pause-frames had been enabled on the port (SL_MACCONTROL[4] TX_FLOW_EN = 1). The port could have been in either half or full-duplex mode. See the RX_ALIGN_CODE_ERRORS and RX_CRC_ERRORS statistic descriptions for definitions of alignment, code and CRC errors. Overruns have no effect upon this statistic.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VALUE																															

Bits	Field Name	Description	Type	Reset
31:0	VALUE	Statistic value	RW	0x0000 0000

Table 18-1081. Register Call Summary for Register PAUSE_RX_FRAMES

Gigabit Ethernet Switch (GMAC_SW)

- STATS Register Summary: [0]

Table 18-1082. RX_CRC_ERRORS

Address Offset	0x0000 0010		
Physical Address	0x4848 4910	**Instance**	STATS
Description	The total number of frames received on the port that experienced a CRC error. Such a frame: - Was any data or MAC control frame which matched a unicast, broadcast or multicast address, or matched due to promiscuous mode - Was of length 64 to SL_RX_MAXLEN[13:0] RX_MAXLEN bytes inclusive - Had no code/align error, - Had a CRC error Overruns have no effect upon this statistic. A CRC error is defined to be: - A frame containing an even number of nibbles - Failing the Frame Check Sequence test		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VALUE																															

Bits	Field Name	Description	Type	Reset
31:0	VALUE	Statistic value	RW	0x0000 0000

Table 18-1083. Register Call Summary for Register RX_CRC_ERRORS

Gigabit Ethernet Switch (GMAC_SW)

- STATS Register Summary: [0]
- STATS Register Description: [1][2][3][4][5][6][7][8][9][10]

Table 18-1084. RX_ALIGN_CODE_ERRORS

Address Offset	0x0000 0014		
Physical Address	0x4848 4914	**Instance**	STATS
Description	The total number of frames received on the port that experienced an alignment error or code error. Such a frame: - Was any data or MAC control frame which matched a unicast, broadcast or multicast address, or matched due to promiscuous mode - Was of length 64 to SL_RX_MAXLEN[13:0] RX_MAXLEN bytes inclusive - Had either an alignment error or a code error Overruns have no effect upon this statistic. An alignment error is defined to be: - A frame containing an odd number of nibbles - Failing the Frame Check Sequence test if the final nibble is ignored A code error is defined to be a frame which has been discarded because the port's MRXER pin driven with a one for at least one bit-time's duration at any point during the frame's reception. Note: RFC 1757 etherStatsCRCAlignErrors Ref. 1.5 can be calculated by summing RX_ALIGN_CODE_ERRORS and RX_CRC_ERRORS.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VALUE																															

Bits	Field Name	Description	Type	Reset
31:0	VALUE	Statistic value	RW	0x0000 0000

Table 18-1085. Register Call Summary for Register RX_ALIGN_CODE_ERRORS

Gigabit Ethernet Switch (GMAC_SW)

- STATS Register Summary: [0]
- STATS Register Description: [1][2][3][4][5][6][7][8][9][10]

Table 18-1086. OVERSIZE_RX_FRAMES

Address Offset	0x0000 0018		
Physical Address	0x4848 4918	**Instance**	STATS
Description	The total number of oversized frames received on the port. An oversized frame is defined to be: - Was any data or MAC control frame which matched a unicast, broadcast or multicast address, or matched due to promiscuous mode - Was greater than SL_RX_MAXLEN[13:0] RX_MAXLEN in bytes - Had no CRC error, alignment error or code error See the RX_ALIGN_CODE_ERRORS and RX_CRC_ERRORS statistic descriptions for definitions of alignment, code and CRC errors. Overruns have no effect upon this statistic.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VALUE																															

Bits	Field Name	Description	Type	Reset
31:0	VALUE	Statistic value	RW	0x0000 0000

Table 18-1087. Register Call Summary for Register OVERSIZE_RX_FRAMES

Gigabit Ethernet Switch (GMAC_SW)

- STATS Register Summary: [0]

Table 18-1088. RX_JABBERS

Address Offset	0x0000 001C		
Physical Address	0x4848 491C	**Instance**	STATS
Description	The total number of jabber frames received on the port. A jabber frame: - Was any data or MAC control frame which matched a unicast, broadcast or multicast address, or matched due to promiscuous mode - Was greater than SL_RX_MAXLEN[13:0] RX_MAXLEN in bytes - Had no CRC error, alignment error or code error See the RX_ALIGN_CODE_ERRORS and RX_CRC_ERRORS statistic descriptions for definitions of alignment, code and CRC errors. Overruns have no effect upon this statistic.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VALUE																															

Bits	Field Name	Description	Type	Reset
31:0	VALUE	Statistic value	RW	0x0000 0000

Table 18-1089. Register Call Summary for Register RX_JABBERS

Gigabit Ethernet Switch (GMAC_SW)

- STATS Register Summary: [0]

Table 18-1090. UNDERSIZE_RX_FRAMES

Address Offset	0x0000 0020		
Physical Address	0x4848 4920	**Instance**	STATS
Description	The total number of undersized frames received on the port. An undersized frame is defined to be: - Was any data frame which matched a unicast, broadcast or multicast address, or matched due to promiscuous mode - Was less than 64 octets long - Had no CRC error, alignment error or code error See the RX_ALIGN_CODE_ERRORS and RX_CRC_ERRORS statistic descriptions for definitions of alignment, code and CRC errors. Overruns have no effect upon this statistic.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VALUE																															

Bits	Field Name	Description	Type	Reset
31:0	VALUE	Statistic value	RW	0x0000 0000

Table 18-1091. Register Call Summary for Register UNDERSIZE_RX_FRAMES

Gigabit Ethernet Switch (GMAC_SW)

- STATS Register Summary: [0]

Table 18-1092. RX_FRAGMENTS

Address Offset	0x0000 0024		
Physical Address	0x4848 4924	**Instance**	STATS
Description	The total number of frame fragments received on the port. A frame fragment is defined to be: - Any data frame (address matching does not matter) - Less than 64 bytes long - Having a CRC error, an alignment error, or a code error - Not the result of a collision caused by half duplex, collision based flow control See the RX_ALIGN_CODE_ERRORS and RX_CRC_ERRORS statistic descriptions for definitions of alignment, code and CRC errors. Overruns have no effect upon this statistic.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VALUE																															

Bits	Field Name	Description	Type	Reset
31:0	VALUE	Statistic value	RW	0x0000 0000

Table 18-1093. Register Call Summary for Register RX_FRAGMENTS

Gigabit Ethernet Switch (GMAC_SW)

- STATS Register Summary: [0]

Table 18-1094. RX_OCTETS

Address Offset	0x0000 0030		
Physical Address	0x4848 4930	**Instance**	STATS
Description	The total number of bytes in all good frames received on the port. A good frame is defined to be: - Any data or MAC control frame which matched a unicast, broadcast or multicast address, or matched due to promiscuous mode - Of length 64 to SL_RX_MAXLEN[13:0] RX_MAXLEN bytes inclusive - Had no CRC error, alignment error or code error See the RX_ALIGN_CODE_ERRORS and RX_CRC_ERRORS statistic descriptions for definitions of alignment, code and CRC errors. Overruns have no effect upon this statistic.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VALUE																															

Bits	Field Name	Description	Type	Reset
31:0	VALUE	Statistic value	RW	0x0000 0000

Table 18-1095. Register Call Summary for Register RX_OCTETS

Gigabit Ethernet Switch (GMAC_SW)

- STATS Register Summary: [0]

Table 18-1096. GOOD_TX_FRAMES

Address Offset	0x0000 0034		
Physical Address	0x4848 4934	**Instance**	STATS
Description	The total number of good frames received on the port. A good frame is defined to be: - Any data or MAC control frame which matched a unicast, broadcast or multicast address, or matched due to promiscuous mode - Any length - Had no late or excessive collisions, no carrier loss and no underrun		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VALUE																															

Bits	Field Name	Description	Type	Reset
31:0	VALUE	Statistic value	RW	0x0000 0000

Table 18-1097. Register Call Summary for Register GOOD_TX_FRAMES

Gigabit Ethernet Switch (GMAC_SW)

- STATS Register Summary: [0]

Table 18-1098. BROADCAST_TX_FRAMES

Address Offset	0x0000 0038		
Physical Address	0x4848 4938	**Instance**	STATS
Description	The total number of good broadcast frames received on the port. A good broadcast frame is defined to be: - Any data or MAC control frame which was destined for only address 0xFFFFFFFFFFFF - Any length - Had no late or excessive collisions, no carrier loss and no underrun		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VALUE																															

Bits	Field Name	Description	Type	Reset
31:0	VALUE	Statistic value	RW	0x0000 0000

Table 18-1099. Register Call Summary for Register BROADCAST_TX_FRAMES

Gigabit Ethernet Switch (GMAC_SW)

- STATS Register Summary: [0]

Table 18-1100. MULTICAST_TX_FRAMES

Address Offset	0x0000 003C		
Physical Address	0x4848 493C	**Instance**	STATS
Description	The total number of good multicast frames received on the port. A good multicast frame is defined to be: - Any data or MAC control frame which was destined for any multicast address other than 0xFFFFFFFFFFFF - Any length - Had no late or excessive collisions, no carrier loss and no underrun		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VALUE																															

Bits	Field Name	Description	Type	Reset
31:0	VALUE	Statistic value	RW	0x0000 0000

Table 18-1101. Register Call Summary for Register MULTICAST_TX_FRAMES

Gigabit Ethernet Switch (GMAC_SW)

- STATS Register Summary: [0]
- STATS Register Description: [1]

Table 18-1102. PAUSE_TX_FRAMES

Address Offset	0x0000 0040		
Physical Address	0x4848 4940	**Instance**	STATS
Description	This statistic indicates the number of IEEE 802.3X pause frames transmitted by the port. Pause frames cannot underrun or contain a CRC error because they are created in the transmitting MAC, so these error conditions have no effect upon the statistic. Pause frames sent by software will not be included in this count. Since pause frames are only transmitted in full duplex carrier loss and collisions have no effect upon this statistic. Transmitted pause frames are always 64 byte multicast frames so will appear in the MULTICAST_TX_FRAMES and RX_TX_64_OCTET_FRAMES statistics.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VALUE																															

Bits	Field Name	Description	Type	Reset
31:0	VALUE	Statistic value	RW	0x0000 0000

Table 18-1103. Register Call Summary for Register PAUSE_TX_FRAMES

Gigabit Ethernet Switch (GMAC_SW)

- STATS Register Summary: [0]

Table 18-1104. DEFERRED_TX_FRAMES

Address Offset	0x0000 0044		
Physical Address	0x4848 4944	**Instance**	STATS
Description	The total number of frames transmitted on the port that first experienced deferment. Such a frame: - Was any data or MAC control frame destined for any unicast, broadcast or multicast address - Was any size - Had no carrier loss and no underrun - Experienced no collisions before being successfully transmitted - Found the medium busy when transmission was first attempted, so had to wait. CRC errors have no effect upon this statistic.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VALUE																															

Bits	Field Name	Description	Type	Reset
31:0	VALUE	Statistic value	RW	0x0000 0000

Table 18-1105. Register Call Summary for Register DEFERRED_TX_FRAMES

Gigabit Ethernet Switch (GMAC_SW)

- STATS Register Summary: [0]

Table 18-1106. COLLISIONS

Address Offset	0x0000 0048		
Physical Address	0x4848 4948	**Instance**	STATS
Description	This statistic records the total number of times that the port experienced a collision. Collisions occur under two circumstances. 1. When a transmit data or MAC control frame: - Was destined for any unicast, broadcast or multicast address - Was any size - Had no carrier loss and no underrun - Experienced a collision. A jam sequence is sent for every non-late collision, so this statistic will increment on each occasion if a frame experiences multiple collisions (and increments on late collisions) CRC errors have no effect upon this statistic. 2. When the port is in half-duplex mode, flow control is active, and a frame reception begins.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VALUE																															

Bits	Field Name	Description	Type	Reset
31:0	VALUE	Statistic value	RW	0x0000 0000

Table 18-1107. Register Call Summary for Register COLLISIONS

Gigabit Ethernet Switch (GMAC_SW)

- STATS Register Summary: [0]

Table 18-1108. SINGLE_COLLISION_TX_FRAMES

Address Offset	0x0000 004C		
Physical Address	0x4848 494C	**Instance**	STATS
Description	The total number of frames transmitted on the port that experienced exactly one collision. Such a frame: - Was any data or MAC control frame destined for any unicast, broadcast or multicast address - Was any size - Had no carrier loss and no underrun - Experienced one collision before successful transmission. The collision was not late. CRC errors have no effect upon this statistic.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VALUE																															

Bits	Field Name	Description	Type	Reset
31:0	VALUE	Statistic value	RW	0x0000 0000

Table 18-1109. Register Call Summary for Register SINGLE_COLLISION_TX_FRAMES

Gigabit Ethernet Switch (GMAC_SW)

- STATS Register Summary: [0]

Table 18-1110. MULTIPLE_COLLISION_TX_FRAMES

Address Offset	0x0000 0050		
Physical Address	0x4848 4950	**Instance**	STATS
Description	The total number of frames transmitted on the port that experienced multiple collisions. Such a frame: - Was any data or MAC control frame destined for any unicast, broadcast or multicast address - Was any size - Had no carrier loss and no underrun - Experienced 2 to 15 collisions before being successfully transmitted. None of the collisions were late. CRC errors have no effect upon this statistic.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VALUE																															

Bits	Field Name	Description	Type	Reset
31:0	VALUE	Statistic value	RW	0x0000 0000

Table 18-1111. Register Call Summary for Register MULTIPLE_COLLISION_TX_FRAMES

Gigabit Ethernet Switch (GMAC_SW)

- STATS Register Summary: [0]

Table 18-1112. EXCESSIVE_COLLISIONS

Address Offset	0x0000 0054		
Physical Address	0x4848 4954	**Instance**	STATS
Description	The total number of frames for which transmission was abandoned due to excessive collisions. Such a frame: - Was any data or MAC control frame destined for any unicast, broadcast or multicast address - Was any size - Had no carrier loss and no underrun - Experienced 16 collisions before abandoning all attempts at transmitting the frame. None of the collisions were late. CRC errors have no effect upon this statistic.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VALUE																															

Bits	Field Name	Description	Type	Reset
31:0	VALUE	Statistic value	RW	0x0000 0000

Table 18-1113. Register Call Summary for Register EXCESSIVE_COLLISIONS

Gigabit Ethernet Switch (GMAC_SW)

- STATS Register Summary: [0]

Table 18-1114. LATE_COLLISIONS

Address Offset	0x0000 0058		
Physical Address	0x4848 4958	**Instance**	STATS
Description	The total number of frames on the port for which transmission was abandoned because they experienced a late collision. Such a frame: - Was any data or MAC control frame destined for any unicast, broadcast or multicast address - Was any size - Experienced a collision later than 512 bit-times into the transmission. There may have been up to 15 previous (non-late) collisions which had previously required the transmission to be re-attempted. The Late Collisions statistic dominates over the single, multiple and excessive Collisions statistics - if a late collision occurs the frame will not be counted in any of these other three statistics. CRC errors have no effect upon this statistic.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VALUE																															

Bits	Field Name	Description	Type	Reset
31:0	VALUE	Statistic value	RW	0x0000 0000

Table 18-1115. Register Call Summary for Register LATE_COLLISIONS

Gigabit Ethernet Switch (GMAC_SW)

- STATS Register Summary: [0]

Table 18-1116. TX_UNDERRUN

Address Offset	0x0000 005C		
Physical Address	0x4848 495C	**Instance**	STATS
Description	There should be no transmitted frames that experience underrun.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VALUE																															

Bits	Field Name	Description	Type	Reset
31:0	VALUE	Statistic value	RW	0x0000 0000

Table 18-1117. Register Call Summary for Register TX_UNDERRUN

Gigabit Ethernet Switch (GMAC_SW)

- STATS Register Summary: [0]

Table 18-1118. CARRIER_SENSE_ERRORS

Address Offset	0x0000 0060		
Physical Address	0x4848 4960	**Instance**	STATS
Description	The total number of frames received on the port that had a CPDMA middle of frame (MOF) overrun. MOF overrun frame is defined to be: - Was any data or MAC control frame destined for any unicast, broadcast or multicast address - Was any size - The carrier sense condition was lost or never asserted when transmitting the frame (the frame is not retransmitted). This is a transmit only statistic. Carrier Sense is a don't care for received frames. Transmit frames with carrier sense errors are sent until completion and are not aborted. CRC errors have no effect upon this statistic.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VALUE																															

Bits	Field Name	Description	Type	Reset
31:0	VALUE	Statistic value	RW	0x0000 0000

Table 18-1119. Register Call Summary for Register CARRIER_SENSE_ERRORS

Gigabit Ethernet Switch (GMAC_SW)

- STATS Register Summary: [0]

Table 18-1120. TX_OCTETS

Address Offset	0x0000 0064		
Physical Address	0x4848 4964	**Instance**	STATS
Description	The total number of bytes in all good frames transmitted on the port. A good frame is defined to be: - Any data or MAC control frame which was destined for any unicast, broadcast or multicast address - Was any size - Had no late or excessive collisions, no carrier loss and no underrun.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VALUE																															

Bits	Field Name	Description	Type	Reset
31:0	VALUE	Statistic value	RW	0x0000 0000

Table 18-1121. Register Call Summary for Register TX_OCTETS

Gigabit Ethernet Switch (GMAC_SW)

- STATS Register Summary: [0]

Table 18-1122. RX_TX_64_OCTET_FRAMES

Address Offset	0x0000 0068		
Physical Address	0x4848 4968	**Instance**	STATS
Description	The total number of 64-byte frames received and transmitted on the port. Such a frame is defined to be: - Any data or MAC control frame which was destined for any unicast, broadcast or multicast address - Did not experience late collisions, excessive collisions, or carrier sense error - Was exactly 64 bytes long. (If the frame was being transmitted and experienced carrier loss that resulted in a frame of this size being transmitted, then the frame will be recorded in this statistic). CRC errors, code/align errors and overruns do not affect the recording of frames in this statistic.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VALUE																															

Bits	Field Name	Description	Type	Reset
31:0	VALUE	Statistic value	RW	0x0000 0000

Table 18-1123. Register Call Summary for Register RX_TX_64_OCTET_FRAMES

Gigabit Ethernet Switch (GMAC_SW)

- STATS Register Summary: [0]
- STATS Register Description: [1]

Table 18-1124. RX_TX_65_127_OCTET_FRAMES

Address Offset	0x0000 006C		
Physical Address	0x4848 496C	**Instance**	STATS
Description	The total number of frames of size 65 to 127 bytes received and transmitted on the port. Such a frame is defined to be: - Any data or MAC control frame which was destined for any unicast, broadcast or multicast address - Did not experience late collisions, excessive collisions, or carrier sense error - Was 65 to 127 bytes long CRC errors, code/align errors, underruns and overruns do not affect the recording of frames in this statistic.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VALUE																															

Bits	Field Name	Description	Type	Reset
31:0	VALUE	Statistic value	RW	0x0000 0000

Table 18-1125. Register Call Summary for Register RX_TX_65_127_OCTET_FRAMES

Gigabit Ethernet Switch (GMAC_SW)

- STATS Register Summary: [0]

Table 18-1126. RX_TX_128_255_OCTET_FRAMES

Address Offset	0x0000 0070		
Physical Address	0x4848 4970	**Instance**	STATS
Description	The total number of frames of size 128 to 255 bytes received and transmitted on the port. Such a frame is defined to be: - Any data or MAC control frame which was destined for any unicast, broadcast or multicast address - Did not experience late collisions, excessive collisions, or carrier sense error - Was 128 to 255 bytes long CRC errors, code/align errors, underruns and overruns do not affect the recording of frames in this statistic.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VALUE																															

Bits	Field Name	Description	Type	Reset
31:0	VALUE	Statistic value	RW	0x0000 0000

Table 18-1127. Register Call Summary for Register RX_TX_128_255_OCTET_FRAMES

Gigabit Ethernet Switch (GMAC_SW)

- STATS Register Summary: [0]

Table 18-1128. RX_TX_256_511_OCTET_FRAMES

Address Offset	0x0000 0074		
Physical Address	0x4848 4974	**Instance**	STATS
Description	The total number of frames of size 256 to 511 bytes received and transmitted on the port. Such a frame is defined to be: - Any data or MAC control frame which was destined for any unicast, broadcast or multicast address - Did not experience late collisions, excessive collisions, or carrier sense error - Was 256 to 511 bytes long CRC errors, code/align errors, underruns and overruns do not affect the recording of frames in this statistic.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VALUE																															

Bits	Field Name	Description	Type	Reset
31:0	VALUE	Statistic value	RW	0x0000 0000

Table 18-1129. Register Call Summary for Register RX_TX_256_511_OCTET_FRAMES

Gigabit Ethernet Switch (GMAC_SW)

- STATS Register Summary: [0]

Table 18-1130. RX_TX_512_1023_OCTET_FRAMES

Address Offset	0x0000 0078		
Physical Address	0x4848 4978	**Instance**	STATS
Description	The total number of frames of size 512 to 1023 bytes received and transmitted on the port. Such a frame is defined to be: - Any data or MAC control frame which was destined for any unicast, broadcast or multicast address - Did not experience late collisions, excessive collisions, or carrier sense error - Was 512 to 1023 bytes long CRC errors, code/align errors, underruns and overruns do not affect the recording of frames in this statistic.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VALUE																															

Bits	Field Name	Description	Type	Reset
31:0	VALUE	Statistic value	RW	0x0000 0000

Table 18-1131. Register Call Summary for Register RX_TX_512_1023_OCTET_FRAMES

Gigabit Ethernet Switch (GMAC_SW)

- STATS Register Summary: [0]

Table 18-1132. RX_TX_1024_UP_OCTET_FRAMES

Address Offset	0x0000 007C		
Physical Address	0x4848 497C	**Instance**	STATS
Description	The total number of frames of size 1024 to SL_RX_MAXLEN[13:0] RX_MAXLEN bytes for receive or 1024 up for transmit on the port. Such a frame is defined to be: - Any data or MAC control frame which was destined for any unicast, broadcast or multicast address - Did not experience late collisions, excessive collisions, or carrier sense error - Was 1024 to SL_RX_MAXLEN[13:0] RX_MAXLEN bytes long on receive, or any size on transmit CRC errors, code/align errors, underruns and overruns do not affect the recording of frames in this statistic.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VALUE																															

Bits	Field Name	Description	Type	Reset
31:0	VALUE	Statistic value	RW	0x0000 0000

Table 18-1133. Register Call Summary for Register RX_TX_1024_UP_OCTET_FRAMES

Gigabit Ethernet Switch (GMAC_SW)

- STATS Register Summary: [0]

Table 18-1134. NET_OCTETS

Address Offset	0x0000 0080		
Physical Address	0x4848 4980	**Instance**	STATS
Description	The total number of bytes of frame data received and transmitted on the port. Each frame counted: - was any data or MAC control frame destined for any unicast, broadcast or multicast address (address match does not matter) - Any length (including less than 64 bytes and greater than SL_RX_MAXLEN[13:0] RX_MAXLEN bytes) Also counted in this statistic is: - Every byte transmitted before a carrier- loss was experienced - Every byte transmitted before each collision was experienced, (i.e. multiple retries are counted each time) - Every byte received if the port is in half-duplex mode until a jam sequence was transmitted to initiate flow control. (The jam sequence was not counted to prevent double-counting) Error conditions such as alignment errors, CRC errors, code errors, overruns and underruns do not affect the recording of bytes by this statistic. The objective of this statistic is to give a reasonable indication of ethernet utilization		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VALUE																															

Bits	Field Name	Description	Type	Reset
31:0	VALUE	Statistic value	RW	0x0000 0000

Table 18-1135. Register Call Summary for Register NET_OCTETS

Gigabit Ethernet Switch (GMAC_SW)

- STATS Register Summary: [0]

Table 18-1136. RX_START_OF_FRAME_OVERRUNS

Address Offset	0x0000 0084		
Physical Address	0x4848 4984	**Instance**	STATS
Description	The total number of frames received on the port that had a CPDMA start of frame (SOF) overrun or were dropped by due to FIFO resource limitations, or were dropped by the SPF. SOF overrun frame is defined to be: - Any data or MAC control frame which matched a unicast, broadcast or multicast address, or matched due to promiscuous mode - Any length (including less than 64 bytes and greater than SL_RX_MAXLEN[13:0] RX_MAXLEN bytes) - The CPDMA had a start of frame overrun or the packet was dropped due to FIFO resource limitations		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VALUE																															

Bits	Field Name	Description	Type	Reset
31:0	VALUE	Statistic value	RW	0x0000 0000

Table 18-1137. Register Call Summary for Register RX_START_OF_FRAME_OVERRUNS

Gigabit Ethernet Switch (GMAC_SW)

- STATS Register Summary: [0]

Table 18-1138. RX_MIDDLE_OF_FRAME_OVERRUNS

Address Offset	0x0000 0088		
Physical Address	0x4848 4988	**Instance**	STATS
Description	The total number of frames received on the port that had a CPDMA middle of frame (MOF) overrun. MOF overrun frame is defined to be: - Any data or MAC control frame which matched a unicast, broadcast or multicast address, or matched due to promiscuous mode - Any length (including less than 64 bytes and greater than SL_RX_MAXLEN[13:0] RX_MAXLEN bytes) - The CPDMA had a middle of frame overrun		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VALUE																															

Bits	Field Name	Description	Type	Reset
31:0	VALUE	Statistic value	RW	0x0000 0000

Table 18-1139. Register Call Summary for Register RX_MIDDLE_OF_FRAME_OVERRUNS

Gigabit Ethernet Switch (GMAC_SW)

- STATS Register Summary: [0]

Table 18-1140. RX_DMA_OVERRUNS

Address Offset	0x0000 008C		
Physical Address	0x4848 498C	**Instance**	STATS
Description	The total number of frames received on the port that had either a DMA start of frame (SOF) overrun or a DMA MOF overrun. An Rx DMA overrun frame is defined to be: - Any data or MAC control frame which matched a unicast, broadcast or multicast address, or matched due to promiscuous mode - Any length (including less than 64 bytes and greater than SL_RX_MAXLEN[13:0] RX_MAXLEN bytes) - The CPGMAC_SL was unable to receive it because it did not have the DMA buffer resources to receive it (zero head descriptor pointer at the start or during the middle of the frame reception) CRC errors, alignment errors and code errors have no effect upon this statistic.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VALUE																															

Bits	Field Name	Description	Type	Reset
31:0	VALUE	Statistic value	RW	0x0000 0000

Table 18-1141. Register Call Summary for Register RX_DMA_OVERRUNS

Gigabit Ethernet Switch (GMAC_SW)

- STATS Register Summary: [0]

18.8.6.6 STATERAM Registers

18.8.6.6.1 STATERAM Register Summary

Table 18-1142. STATERAM Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	STATERAM Physical Address
TX0_HDP	RW	32	0x0000 0000	0x4848 4A00
TX1_HDP	RW	32	0x0000 0004	0x4848 4A04
TX2_HDP	RW	32	0x0000 0008	0x4848 4A08
TX3_HDP	RW	32	0x0000 000C	0x4848 4A0C
TX4_HDP	RW	32	0x0000 0010	0x4848 4A10
TX5_HDP	RW	32	0x0000 0014	0x4848 4A14
TX6_HDP	RW	32	0x0000 0018	0x4848 4A18
TX7_HDP	RW	32	0x0000 001C	0x4848 4A1C
RX0_HDP	RW	32	0x0000 0020	0x4848 4A20
RX1_HDP	RW	32	0x0000 0024	0x4848 4A24
RX2_HDP	RW	32	0x0000 0028	0x4848 4A28
RX3_HDP	RW	32	0x0000 002C	0x4848 4A2C
RX4_HDP	RW	32	0x0000 0030	0x4848 4A30
RX5_HDP	RW	32	0x0000 0034	0x4848 4A34
RX6_HDP	RW	32	0x0000 0038	0x4848 4A38
RX7_HDP	RW	32	0x0000 003C	0x4848 4A3C
TX0_CP	RW	32	0x0000 0040	0x4848 4A40
TX1_CP	RW	32	0x0000 0044	0x4848 4A44
TX2_CP	RW	32	0x0000 0048	0x4848 4A48
TX3_CP	RW	32	0x0000 004C	0x4848 4A4C
TX4_CP	RW	32	0x0000 0050	0x4848 4A50

Table 18-1142. STATERAM Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	STATERAM Physical Address
TX5_CP	RW	32	0x0000 0054	0x4848 4A54
TX6_CP	RW	32	0x0000 0058	0x4848 4A58
TX7_CP	RW	32	0x0000 005C	0x4848 4A5C
RX0_CP	RW	32	0x0000 0060	0x4848 4A60
RX1_CP	RW	32	0x0000 0064	0x4848 4A64
RX2_CP	RW	32	0x0000 0068	0x4848 4A68
RX3_CP	RW	32	0x0000 006C	0x4848 4A6C
RX4_CP	RW	32	0x0000 0070	0x4848 4A70
RX5_CP	RW	32	0x0000 0074	0x4848 4A74
RX6_CP	RW	32	0x0000 0078	0x4848 4A78
RX7_CP	RW	32	0x0000 007C	0x4848 4A7C

18.8.6.6.2 STATERAM Register Description

Table 18-1143. TX0_HDP

Address Offset	0x0000 0000		
Physical Address	0x4848 4A00	**Instance**	STATERAM
Description	CPDMA_STATERAM TX channel 0 head descriptor pointer		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TX_HDP																															

Bits	Field Name	Description	Type	Reset
31:0	TX_HDP	TX Channel (0..7) DMA Head Descriptor Pointer - Writing a TX DMA Buffer Descriptor address to a head pointer location initiates TX DMA operations in the queue for the selected channel. Writing to these locations when they are non-zero is an error (except at reset). Host software must initialize these locations to zero on reset.	RW	0x0

Table 18-1144. Register Call Summary for Register TX0_HDP

Gigabit Ethernet Switch (GMAC_SW)
- STATERAM Register Summary: [0]

Table 18-1145. TX1_HDP

Address Offset	0x0000 0004		
Physical Address	0x4848 4A04	**Instance**	STATERAM
Description	CPDMA_STATERAM TX channel 1 head descriptor pointer		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TX_HDP																															

Bits	Field Name	Description	Type	Reset
31:0	TX_HDP	TX Channel (0..7) DMA Head Descriptor Pointer - Writing a TX DMA Buffer Descriptor address to a head pointer location initiates TX DMA operations in the queue for the selected channel. Writing to these locations when they are non-zero is an error (except at reset). Host software must initialize these locations to zero on reset.	RW	0x0

Table 18-1146. Register Call Summary for Register TX1_HDP

Gigabit Ethernet Switch (GMAC_SW)
- STATERAM Register Summary: [0]

Table 18-1147. TX2_HDP

Address Offset	0x0000 0008		
Physical Address	0x4848 4A08	**Instance**	STATERAM
Description	CPDMA_STATERAM TX channel 2 head descriptor pointer		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TX_HDP																															

Bits	Field Name	Description	Type	Reset
31:0	TX_HDP	TX Channel (0..7) DMA Head Descriptor Pointer - Writing a TX DMA Buffer Descriptor address to a head pointer location initiates TX DMA operations in the queue for the selected channel. Writing to these locations when they are non-zero is an error (except at reset). Host software must initialize these locations to zero on reset.	RW	0x0

Table 18-1148. Register Call Summary for Register TX2_HDP

Gigabit Ethernet Switch (GMAC_SW)
- STATERAM Register Summary: [0]

Table 18-1149. TX3_HDP

Address Offset	0x0000 000C		
Physical Address	0x4848 4A0C	**Instance**	STATERAM
Description	CPDMA_STATERAM TX channel 3 head descriptor pointer		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TX_HDP																															

Bits	Field Name	Description	Type	Reset
31:0	TX_HDP	TX Channel (0..7) DMA Head Descriptor Pointer - Writing a TX DMA Buffer Descriptor address to a head pointer location initiates TX DMA operations in the queue for the selected channel. Writing to these locations when they are non-zero is an error (except at reset). Host software must initialize these locations to zero on reset.	RW	0x0

Table 18-1150. Register Call Summary for Register TX3_HDP

Gigabit Ethernet Switch (GMAC_SW)

- STATERAM Register Summary: [0]

Table 18-1151. TX4_HDP

Address Offset	0x0000 0010		
Physical Address	0x4848 4A10	**Instance**	STATERAM
Description	CPDMA_STATERAM TX channel 4 head descriptor pointer		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TX_HDP																															

Bits	Field Name	Description	Type	Reset
31:0	TX_HDP	TX Channel (0..7) DMA Head Descriptor Pointer - Writing a TX DMA Buffer Descriptor address to a head pointer location initiates TX DMA operations in the queue for the selected channel. Writing to these locations when they are non-zero is an error (except at reset). Host software must initialize these locations to zero on reset.	RW	0x0

Table 18-1152. Register Call Summary for Register TX4_HDP

Gigabit Ethernet Switch (GMAC_SW)

- STATERAM Register Summary: [0]

Table 18-1153. TX5_HDP

Address Offset	0x0000 0014		
Physical Address	0x4848 4A14	**Instance**	STATERAM
Description	CPDMA_STATERAM TX channel 5 head descriptor pointer		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TX_HDP																															

Bits	Field Name	Description	Type	Reset
31:0	TX_HDP	TX Channel (0..7) DMA Head Descriptor Pointer - Writing a TX DMA Buffer Descriptor address to a head pointer location initiates TX DMA operations in the queue for the selected channel. Writing to these locations when they are non-zero is an error (except at reset). Host software must initialize these locations to zero on reset.	RW	0x0

Table 18-1154. Register Call Summary for Register TX5_HDP

Gigabit Ethernet Switch (GMAC_SW)

- STATERAM Register Summary: [0]

Table 18-1155. TX6_HDP

Address Offset	0x0000 0018		
Physical Address	0x4848 4A18	**Instance**	STATERAM
Description	CPDMA_STATERAM TX channel 6 head descriptor pointer		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TX_HDP																															

Bits	Field Name	Description	Type	Reset
31:0	TX_HDP	TX Channel (0..7) DMA Head Descriptor Pointer - Writing a TX DMA Buffer Descriptor address to a head pointer location initiates TX DMA operations in the queue for the selected channel. Writing to these locations when they are non-zero is an error (except at reset). Host software must initialize these locations to zero on reset.	RW	0x0

Table 18-1156. Register Call Summary for Register TX6_HDP

Gigabit Ethernet Switch (GMAC_SW)

- STATERAM Register Summary: [0]

Table 18-1157. TX7_HDP

Address Offset	0x0000 001C		
Physical Address	0x4848 4A1C	**Instance**	STATERAM
Description	CPDMA_STATERAM TX channel 7 head descriptor pointer		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TX_HDP																															

Bits	Field Name	Description	Type	Reset
31:0	TX_HDP	TX Channel (0..7) DMA Head Descriptor Pointer - Writing a TX DMA Buffer Descriptor address to a head pointer location initiates TX DMA operations in the queue for the selected channel. Writing to these locations when they are non-zero is an error (except at reset). Host software must initialize these locations to zero on reset.	RW	0x0

Table 18-1158. Register Call Summary for Register TX7_HDP

Gigabit Ethernet Switch (GMAC_SW)

- STATERAM Register Summary: [0]

Table 18-1159. RX0_HDP

Address Offset	0x0000 0020		
Physical Address	0x4848 4A20	**Instance**	STATERAM
Description	CPDMA_STATERAM RX 0 channel 0 head descriptor pointer		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RX_HDP																															

Bits	Field Name	Description	Type	Reset
31:0	RX_HDP	RX DMA Head Descriptor Pointer - Writing an RX DMA Buffer Descriptor address to this location allows RX DMA operations in the selected channel when a channel frame is received. Writing to these locations when they are non-zero is an error (except at reset). Host software must initialize these locations to zero on reset.	RW	0x0

Table 18-1160. Register Call Summary for Register RX0_HDP

Gigabit Ethernet Switch (GMAC_SW)

- STATERAM Register Summary: [0]

Table 18-1161. RX1_HDP

Address Offset	0x0000 0024		
Physical Address	0x4848 4A24	**Instance**	STATERAM
Description	CPDMA_STATERAM RX 1 channel 1 head descriptor pointer		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RX_HDP																															

Bits	Field Name	Description	Type	Reset
31:0	RX_HDP	RX DMA Head Descriptor Pointer - Writing an RX DMA Buffer Descriptor address to this location allows RX DMA operations in the selected channel when a channel frame is received. Writing to these locations when they are non-zero is an error (except at reset). Host software must initialize these locations to zero on reset.	RW	0x0

Table 18-1162. Register Call Summary for Register RX1_HDP

Gigabit Ethernet Switch (GMAC_SW)

- STATERAM Register Summary: [0]

Table 18-1163. RX2_HDP

Address Offset	0x0000 0028		
Physical Address	0x4848 4A28	**Instance**	STATERAM
Description	CPDMA_STATERAM RX 2 channel 2 head descriptor pointer		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RX_HDP																															

Bits	Field Name	Description	Type	Reset
31:0	RX_HDP	RX DMA Head Descriptor Pointer - Writing an RX DMA Buffer Descriptor address to this location allows RX DMA operations in the selected channel when a channel frame is received. Writing to these locations when they are non-zero is an error (except at reset). Host software must initialize these locations to zero on reset.	RW	0x0

Table 18-1164. Register Call Summary for Register RX2_HDP

Gigabit Ethernet Switch (GMAC_SW)

- STATERAM Register Summary: [0]

Table 18-1165. RX3_HDP

Address Offset	0x0000 002C		
Physical Address	0x4848 4A2C	**Instance**	STATERAM
Description	CPDMA_STATERAM RX 3 channel 3 head descriptor pointer		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RX_HDP																															

Bits	Field Name	Description	Type	Reset
31:0	RX_HDP	RX DMA Head Descriptor Pointer - Writing an RX DMA Buffer Descriptor address to this location allows RX DMA operations in the selected channel when a channel frame is received. Writing to these locations when they are non-zero is an error (except at reset). Host software must initialize these locations to zero on reset.	RW	0x0

Table 18-1166. Register Call Summary for Register RX3_HDP

Gigabit Ethernet Switch (GMAC_SW)

- STATERAM Register Summary: [0]

Table 18-1167. RX4_HDP

Address Offset	0x0000 0030		
Physical Address	0x4848 4A30	**Instance**	STATERAM
Description	CPDMA_STATERAM RX 4 channel 4 head descriptor pointer		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RX_HDP																															

Bits	Field Name	Description	Type	Reset
31:0	RX_HDP	RX DMA Head Descriptor Pointer - Writing an RX DMA Buffer Descriptor address to this location allows RX DMA operations in the selected channel when a channel frame is received. Writing to these locations when they are non-zero is an error (except at reset). Host software must initialize these locations to zero on reset.	RW	0x0

Table 18-1168. Register Call Summary for Register RX4_HDP

Gigabit Ethernet Switch (GMAC_SW)

- STATERAM Register Summary: [0]

Table 18-1169. RX5_HDP

Address Offset	0x0000 0034		
Physical Address	0x4848 4A34	**Instance**	STATERAM
Description	CPDMA_STATERAM RX 5 channel 5 head descriptor pointer		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RX_HDP																															

Bits	Field Name	Description	Type	Reset
31:0	RX_HDP	RX DMA Head Descriptor Pointer - Writing an RX DMA Buffer Descriptor address to this location allows RX DMA operations in the selected channel when a channel frame is received. Writing to these locations when they are non-zero is an error (except at reset). Host software must initialize these locations to zero on reset.	RW	0x0

Table 18-1170. Register Call Summary for Register RX5_HDP

Gigabit Ethernet Switch (GMAC_SW)

- STATERAM Register Summary: [0]

Table 18-1171. RX6_HDP

Address Offset	0x0000 0038		
Physical Address	0x4848 4A38	**Instance**	STATERAM
Description	CPDMA_STATERAM RX 6 channel 6 head desc pointer		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RX_HDP																															

Bits	Field Name	Description	Type	Reset
31:0	RX_HDP	RX DMA Head Descriptor Pointer - Writing an RX DMA Buffer Descriptor address to this location allows RX DMA operations in the selected channel when a channel frame is received. Writing to these locations when they are non-zero is an error (except at reset). Host software must initialize these locations to zero on reset.	RW	0x0

Table 18-1172. Register Call Summary for Register RX6_HDP

Gigabit Ethernet Switch (GMAC_SW)

- STATERAM Register Summary: [0]

Table 18-1173. RX7_HDP

Address Offset	0x0000 003C		
Physical Address	0x4848 4A3C	**Instance**	STATERAM
Description	CPDMA_STATERAM RX 7 channel 7 head desc pointer		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RX_HDP																															

Bits	Field Name	Description	Type	Reset
31:0	RX_HDP	RX DMA Head Descriptor Pointer - Writing an RX DMA Buffer Descriptor address to this location allows RX DMA operations in the selected channel when a channel frame is received. Writing to these locations when they are non-zero is an error (except at reset). Host software must initialize these locations to zero on reset.	RW	0x0

Table 18-1174. Register Call Summary for Register RX7_HDP

Gigabit Ethernet Switch (GMAC_SW)

- STATERAM Register Summary: [0]

Table 18-1175. TX0_CP

Address Offset	0x0000 0040		
Physical Address	0x4848 4A40	**Instance**	STATERAM
Description	CPDMA_STATERAM TX channel 0 completion pointer register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TX_CP																															

Bits	Field Name	Description	Type	Reset
31:0	TX_CP	Tx Completion Pointer Register - This register is written by the host with the buffer descriptor address for the last buffer processed by the host during interrupt processing. The port uses the value written to determine if the interrupt should be deasserted.	RW	0x0

Table 18-1176. Register Call Summary for Register TX0_CP

Gigabit Ethernet Switch (GMAC_SW)

- STATERAM Register Summary: [0]

Table 18-1177. TX1_CP

Address Offset	0x0000 0044		
Physical Address	0x4848 4A44	**Instance**	STATERAM
Description	CPDMA_STATERAM TX channel 1 completion pointer register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TX_CP																															

Bits	Field Name	Description	Type	Reset
31:0	TX_CP	Tx Completion Pointer Register - This register is written by the host with the buffer descriptor address for the last buffer processed by the host during interrupt processing. The port uses the value written to determine if the interrupt should be deasserted.	RW	0x0

Table 18-1178. Register Call Summary for Register TX1_CP

Gigabit Ethernet Switch (GMAC_SW)

- STATERAM Register Summary: [0]

Table 18-1179. TX2_CP

Address Offset	0x0000 0048		
Physical Address	0x4848 4A48	**Instance**	STATERAM
Description	CPDMA_STATERAM TX channel 2 completion pointer register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TX_CP																															

Bits	Field Name	Description	Type	Reset
31:0	TX_CP	Tx Completion Pointer Register - This register is written by the host with the buffer descriptor address for the last buffer processed by the host during interrupt processing. The port uses the value written to determine if the interrupt should be deasserted.	RW	0x0

Table 18-1180. Register Call Summary for Register TX2_CP

Gigabit Ethernet Switch (GMAC_SW)
- STATERAM Register Summary: [0]

Table 18-1181. TX3_CP

Address Offset	0x0000 004C		
Physical Address	0x4848 4A4C	**Instance**	STATERAM
Description	CPDMA_STATERAM TX channel 3 completion pointer register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TX_CP																															

Bits	Field Name	Description	Type	Reset
31:0	TX_CP	Tx Completion Pointer Register - This register is written by the host with the buffer descriptor address for the last buffer processed by the host during interrupt processing. The port uses the value written to determine if the interrupt should be deasserted.	RW	0x0

Table 18-1182. Register Call Summary for Register TX3_CP

Gigabit Ethernet Switch (GMAC_SW)
- STATERAM Register Summary: [0]

Table 18-1183. TX4_CP

Address Offset	0x0000 0050		
Physical Address	0x4848 4A50	**Instance**	STATERAM
Description	CPDMA_STATERAM TX channel 4 completion pointer register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TX_CP																															

Bits	Field Name	Description	Type	Reset
31:0	TX_CP	Tx Completion Pointer Register - This register is written by the host with the buffer descriptor address for the last buffer processed by the host during interrupt processing. The port uses the value written to determine if the interrupt should be deasserted.	RW	0x0

Table 18-1184. Register Call Summary for Register TX4_CP

Gigabit Ethernet Switch (GMAC_SW)
- STATERAM Register Summary: [0]

Table 18-1185. TX5_CP

Address Offset	0x0000 0054		
Physical Address	0x4848 4A54	**Instance**	STATERAM
Description	CPDMA_STATERAM TX channel 5 completion pointer register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TX_CP																															

Bits	Field Name	Description	Type	Reset
31:0	TX_CP	Tx Completion Pointer Register - This register is written by the host with the buffer descriptor address for the last buffer processed by the host during interrupt processing. The port uses the value written to determine if the interrupt should be deasserted.	RW	0x0

Table 18-1186. Register Call Summary for Register TX5_CP

Gigabit Ethernet Switch (GMAC_SW)

- STATERAM Register Summary: [0]

Table 18-1187. TX6_CP

Address Offset	0x0000 0058		
Physical Address	0x4848 4A58	**Instance**	STATERAM
Description	CPDMA_STATERAM TX channel 6 completion pointer register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TX_CP																															

Bits	Field Name	Description	Type	Reset
31:0	TX_CP	Tx Completion Pointer Register - This register is written by the host with the buffer descriptor address for the last buffer processed by the host during interrupt processing. The port uses the value written to determine if the interrupt should be deasserted.	RW	0x0

Table 18-1188. Register Call Summary for Register TX6_CP

Gigabit Ethernet Switch (GMAC_SW)

- STATERAM Register Summary: [0]

Table 18-1189. TX7_CP

Address Offset	0x0000 005C		
Physical Address	0x4848 4A5C	**Instance**	STATERAM
Description	CPDMA_STATERAM TX channel 7 completion pointer register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TX_CP																															

Bits	Field Name	Description	Type	Reset
31:0	TX_CP	Tx Completion Pointer Register - This register is written by the host with the buffer descriptor address for the last buffer processed by the host during interrupt processing. The port uses the value written to determine if the interrupt should be deasserted.	RW	0x0

Table 18-1190. Register Call Summary for Register TX7_CP

Gigabit Ethernet Switch (GMAC_SW)
- STATERAM Register Summary: [0]

Table 18-1191. RX0_CP

Address Offset	0x0000 0060		
Physical Address	0x4848 4A60	**Instance**	STATERAM
Description	CPDMA_STATERAM RX channel 0 completion pointer register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RX_CP																															

Bits	Field Name	Description	Type	Reset
31:0	RX_CP	Rx Completion Pointer Register - This register is written by the host with the buffer descriptor address for the last buffer processed by the host during interrupt processing. The port uses the value written to determine if the interrupt should be deasserted. Note: The value read is the completion pointer (interrupt acknowledge) value that was written by the CPDMA DMA controller (port). The value written to this register by the host is compared with the value that the port wrote to determine if the interrupt should remain asserted. The value written is not actually stored in the location. The interrupt is deasserted if the two values are equal.	RW	0x0

Table 18-1192. Register Call Summary for Register RX0_CP

Gigabit Ethernet Switch (GMAC_SW)
- STATERAM Register Summary: [0]

Table 18-1193. RX1_CP

Address Offset	0x0000 0064		
Physical Address	0x4848 4A64	**Instance**	STATERAM
Description	CPDMA_STATERAM RX channel 1 completion pointer register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RX_CP																															

Bits	Field Name	Description	Type	Reset
31:0	RX_CP	Rx Completion Pointer Register - This register is written by the host with the buffer descriptor address for the last buffer processed by the host during interrupt processing. The port uses the value written to determine if the interrupt should be deasserted. Note: The value read is the completion pointer (interrupt acknowledge) value that was written by the CPDMA DMA controller (port). The value written to this register by the host is compared with the value that the port wrote to determine if the interrupt should remain asserted. The value written is not actually stored in the location. The interrupt is deasserted if the two values are equal.	RW	0x0

Table 18-1194. Register Call Summary for Register RX1_CP

Gigabit Ethernet Switch (GMAC_SW)
- STATERAM Register Summary: [0]

Table 18-1195. RX2_CP

Address Offset	0x0000 0068		
Physical Address	0x4848 4A68	**Instance**	STATERAM
Description	CPDMA_STATERAM RX channel 2 completion pointer register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RX_CP																															

Bits	Field Name	Description	Type	Reset
31:0	RX_CP	Rx Completion Pointer Register - This register is written by the host with the buffer descriptor address for the last buffer processed by the host during interrupt processing. The port uses the value written to determine if the interrupt should be deasserted. Note: The value read is the completion pointer (interrupt acknowledge) value that was written by the CPDMA DMA controller (port). The value written to this register by the host is compared with the value that the port wrote to determine if the interrupt should remain asserted. The value written is not actually stored in the location. The interrupt is deasserted if the two values are equal.	RW	0x0

Table 18-1196. Register Call Summary for Register RX2_CP

Gigabit Ethernet Switch (GMAC_SW)
- STATERAM Register Summary: [0]

Table 18-1197. RX3_CP

Address Offset	0x0000 006C		
Physical Address	0x4848 4A6C	**Instance**	STATERAM
Description	CPDMA_STATERAM RX channel 3 completion pointer register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RX_CP																															

Bits	Field Name	Description	Type	Reset
31:0	RX_CP	Rx Completion Pointer Register - This register is written by the host with the buffer descriptor address for the last buffer processed by the host during interrupt processing. The port uses the value written to determine if the interrupt should be deasserted. Note: The value read is the completion pointer (interrupt acknowledge) value that was written by the CPDMA DMA controller (port). The value written to this register by the host is compared with the value that the port wrote to determine if the interrupt should remain asserted. The value written is not actually stored in the location. The interrupt is deasserted if the two values are equal.	RW	0x0

Table 18-1198. Register Call Summary for Register RX3_CP

Gigabit Ethernet Switch (GMAC_SW)
- STATERAM Register Summary: [0]

Table 18-1199. RX4_CP

Address Offset	0x0000 0070		
Physical Address	0x4848 4A70	**Instance**	STATERAM
Description	CPDMA_STATERAM RX channel 4 completion pointer register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RX_CP																															

Bits	Field Name	Description	Type	Reset
31:0	RX_CP	Rx Completion Pointer Register - This register is written by the host with the buffer descriptor address for the last buffer processed by the host during interrupt processing. The port uses the value written to determine if the interrupt should be deasserted. Note: The value read is the completion pointer (interrupt acknowledge) value that was written by the CPDMA DMA controller (port). The value written to this register by the host is compared with the value that the port wrote to determine if the interrupt should remain asserted. The value written is not actually stored in the location. The interrupt is deasserted if the two values are equal.	RW	0x0

Table 18-1200. Register Call Summary for Register RX4_CP

Gigabit Ethernet Switch (GMAC_SW)
- STATERAM Register Summary: [0]

Table 18-1201. RX5_CP

Address Offset	0x0000 0074		
Physical Address	0x4848 4A74	**Instance**	STATERAM
Description	CPDMA_STATERAM RX channel 5 completion pointer register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RX_CP																															

Bits	Field Name	Description	Type	Reset
31:0	RX_CP	Rx Completion Pointer Register - This register is written by the host with the buffer descriptor address for the last buffer processed by the host during interrupt processing. The port uses the value written to determine if the interrupt should be deasserted. Note: The value read is the completion pointer (interrupt acknowledge) value that was written by the CPDMA DMA controller (port). The value written to this register by the host is compared with the value that the port wrote to determine if the interrupt should remain asserted. The value written is not actually stored in the location. The interrupt is deasserted if the two values are equal.	RW	0x0

Table 18-1202. Register Call Summary for Register RX5_CP

Gigabit Ethernet Switch (GMAC_SW)
- STATERAM Register Summary: [0]

Table 18-1203. RX6_CP

Address Offset	0x0000 0078		
Physical Address	0x4848 4A78	**Instance**	STATERAM
Description	CPDMA_STATERAM RX channel 6 completion pointer register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RX_CP																															

Bits	Field Name	Description	Type	Reset
31:0	RX_CP	Rx Completion Pointer Register - This register is written by the host with the buffer descriptor address for the last buffer processed by the host during interrupt processing. The port uses the value written to determine if the interrupt should be deasserted. Note: The value read is the completion pointer (interrupt acknowledge) value that was written by the CPDMA DMA controller (port). The value written to this register by the host is compared with the value that the port wrote to determine if the interrupt should remain asserted. The value written is not actually stored in the location. The interrupt is deasserted if the two values are equal.	RW	0x0

Table 18-1204. Register Call Summary for Register RX6_CP

Gigabit Ethernet Switch (GMAC_SW)
- STATERAM Register Summary: [0]

Table 18-1205. RX7_CP

Address Offset	0x0000 007C		
Physical Address	0x4848 4A7C	**Instance**	STATERAM
Description	CPDMA_STATERAM RX channel 7 completion pointer register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RX_CP																															

Bits	Field Name	Description	Type	Reset
31:0	RX_CP	Rx Completion Pointer Register - This register is written by the host with the buffer descriptor address for the last buffer processed by the host during interrupt processing. The port uses the value written to determine if the interrupt should be deasserted. Note: The value read is the completion pointer (interrupt acknowledge) value that was written by the CPDMA DMA controller (port). The value written to this register by the host is compared with the value that the port wrote to determine if the interrupt should remain asserted. The value written is not actually stored in the location. The interrupt is deasserted if the two values are equal.	RW	0x0

Table 18-1206. Register Call Summary for Register RX7_CP

Gigabit Ethernet Switch (GMAC_SW)
- STATERAM Register Summary: [0]

18.8.6.7 CPTS registers

18.8.6.7.1 CPTS Register Summary

Table 18-1207. CPTS Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	CPTS Physical Address
CPTS_IDVER	R	32	0x0000 0000	0x4848 4C00
CPTS_CONTROL	RW	32	0x0000 0004	0x4848 4C04
CPTS_TS_PUSH	W	32	0x0000 000C	0x4848 4C0C
CPTS_TS_LOAD_VAL	RW	32	0x0000 0010	0x4848 4C10
CPTS_TS_LOAD_EN	W	32	0x0000 0014	0x4848 4C14
CPTS_INTSTAT_RAW	RW	32	0x0000 0020	0x4848 4C20
CPTS_INTSTAT_MASKED	R	32	0x0000 0024	0x4848 4C24
CPTS_INT_ENABLE	RW	32	0x0000 0028	0x4848 4C28
CPTS_EVENT_POP	W	32	0x0000 0030	0x4848 4C30
CPTS_EVENT_LOW	R	32	0x0000 0034	0x4848 4C34
CPTS_EVENT_HIGH	R	32	0x0000 0038	0x4848 4C38

18.8.6.7.2 CPTS Register Description

Table 18-1208. CPTS_IDVER

Address Offset	0x0000 0000		
Physical Address	0x4848 4C00	**Instance**	CPTS
Description	CPTS revision		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REVISION																															

Bits	Field Name	Description	Type	Reset
31:0	REVISION	CPTS revision value	R	0x-

Table 18-1209. Register Call Summary for Register CPTS_IDVER

Gigabit Ethernet Switch (GMAC_SW)

- CPTS Register Summary: [0]

Table 18-1210. CPTS_CONTROL

Address Offset	0x0000 0004		
Physical Address	0x4848 4C04	**Instance**	CPTS
Description	Time sync control register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																				HW4_TS_PUSH_EN	HW3_TS_PUSH_EN	HW2_TS_PUSH_EN	HW1_TS_PUSH_EN	RESERVED						INT_TEST	CPTS_EN

Bits	Field Name	Description	Type	Reset
31:12	RESERVED		R	0x0
11	HW4_TS_PUSH_EN	Hardware push 4 enable	RW	0x0
10	HW3_TS_PUSH_EN	Hardware push 3 enable	RW	0x0
9	HW2_TS_PUSH_EN	Hardware push 2 enable	RW	0x0
8	HW1_TS_PUSH_EN	Hardware push 1 enable	RW	0x0
7:2	RESERVED		R	0x0
1	INT_TEST	Interrupt Test - When set, this bit allows the raw interrupt to be written to facilitate interrupt test.	RW	0x0
0	CPTS_EN	Time Sync Enable - When disabled (cleared to zero), the RCLK domain is held in reset. 0 - Time Sync Disabled 1 - Time Sync Enabled	RW	0x0

Table 18-1211. Register Call Summary for Register CPTS_CONTROL

Gigabit Ethernet Switch (GMAC_SW)

- CPTS Initialization: [0][1]
- Time Sync Events: [2]
- CPTS Register Summary: [3]
- CPTS Register Description: [4]

Table 18-1212. CPTS_TS_PUSH

Address Offset	0x0000 000C		
Physical Address	0x4848 4C0C	**Instance**	CPTS
Description	Time stamp event push register		
Type	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															TS_PUSH

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	TS_PUSH	Time stamp event push - When a logic high is written to this bit a time stamp event is pushed onto the event FIFO. The time stamp value is the time of the write of this register, not the time of the event read. The time stamp value can then be read on interrupt via the event registers. Software should not push a second time stamp event onto the event FIFO until the first time stamp value has been read from the event FIFO (there should be only one time stamp event in the event FIFO at any given time). This bit is write only and always reads zero.	W	0x0

Table 18-1213. Register Call Summary for Register CPTS_TS_PUSH

Gigabit Ethernet Switch (GMAC_SW)

- Time Sync Events: [0]
- CPTS Register Summary: [1]

Table 18-1214. CPTS_TS_LOAD_VAL

Address Offset	0x0000 0010		
Physical Address	0x4848 4C10	**Instance**	CPTS
Description	Time stamp load value register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TS_LOAD_VAL																															

Bits	Field Name	Description	Type	Reset
31:0	TS_LOAD_VAL	Time Stamp Load Value - Writing the CPTS_TS_LOAD_EN[0] TS_LOAD_EN bit causes the value contained in this register to be written into the time stamp. The time stamp value is read by initiating a time stamp push event, not by reading this register. When reading this register, the value read is not the time stamp, but is the value that was last written to this register.	RW	0x0

Table 18-1215. Register Call Summary for Register CPTS_TS_LOAD_VAL

Gigabit Ethernet Switch (GMAC_SW)

- Time Stamp Value: [0]
- CPTS Register Summary: [1]
- CPTS Register Description: [2]

Table 18-1216. CPTS_TS_LOAD_EN

Address Offset	0x0000 0014		
Physical Address	0x4848 4C14	**Instance**	CPTS
Description	Time stamp load enable register		
Type	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															TS_LOAD_EN

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	TS_LOAD_EN	Time Stamp Load - Writing a one to this bit enables the time stamp value to be written via the CPTS_TS_LOAD_VAL register. This feature is included for test purposes. This bit is write only.	W	0x0

Table 18-1217. Register Call Summary for Register CPTS_TS_LOAD_EN

Gigabit Ethernet Switch (GMAC_SW)

- Time Stamp Value: [0]
- CPTS Register Summary: [1]
- CPTS Register Description: [2]

Table 18-1218. CPTS_INTSTAT_RAW

Address Offset	0x0000 0020		
Physical Address	0x4848 4C20	**Instance**	CPTS
Description	Time sync interrupt status raw register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															TS_PEND_RAW

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	TS_PEND_RAW	TS_PEND_RAW int read (before enable). Writable when CPTS_CONTROL[1] INT_TEST = 1. A one in this bit indicates that there is one or more events in the event FIFO.	RW	0x0

Table 18-1219. Register Call Summary for Register CPTS_INTSTAT_RAW

Gigabit Ethernet Switch (GMAC_SW)

- CPTS Interrupt Handling: [0][1]
- CPTS Register Summary: [2]

Table 18-1220. CPTS_INTSTAT_MASKED

Address Offset	0x0000 0024		
Physical Address	0x4848 4C24	**Instance**	CPTS
Description	Time sync interrupt status masked register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															TS_PEND

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	TS_PEND	TS_PEND masked interrupt read (after enable).	R	0x0

Table 18-1221. Register Call Summary for Register CPTS_INTSTAT_MASKED

Gigabit Ethernet Switch (GMAC_SW)
- CPTS Register Summary: [0]

Table 18-1222. CPTS_INT_ENABLE

Address Offset	0x0000 0028		
Physical Address	0x4848 4C28	**Instance**	CPTS
Description	Time sync interrupt enable register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															TS_PEND_EN

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	TS_PEND_EN	TS_PEND masked interrupt enable.	RW	0x0

Table 18-1223. Register Call Summary for Register CPTS_INT_ENABLE

Gigabit Ethernet Switch (GMAC_SW)
- CPTS Initialization: [0]
- CPTS Interrupt Handling: [1][2]
- CPTS Register Summary: [3]

Table 18-1224. CPTS_EVENT_POP

Address Offset	0x0000 0030		
Physical Address	0x4848 4C30	**Instance**	CPTS
Description	Event interrupt pop register		
Type	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															EVENT_POP

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	EVENT_POP	Event Pop - When a logic high is written to this bit an event is popped off the event FIFO. The event FIFO pop occurs as part of the interrupt process after the event has been read in the CPTS_EVENT_LOW and CPTS_EVENT_HIGH registers. Popping an event discards the event and causes the next event, if any, to be moved to the top of the FIFO ready to be read by software on interrupt.	W	0x0

Table 18-1225. Register Call Summary for Register CPTS_EVENT_POP

Gigabit Ethernet Switch (GMAC_SW)
- CPTS Interrupt Handling: [0][1]
- CPTS Register Summary: [2]

Table 18-1226. CPTS_EVENT_LOW

Address Offset	0x0000 0034		
Physical Address	0x4848 4C34	**Instance**	CPTS
Description	Lower 32-bits of the event value		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TIME_STAMP																															

Bits	Field Name	Description	Type	Reset
31:0	TIME_STAMP	Time Stamp - The timestamp is valid for transmit, receive, and time stamp push event types. The timestamp value is not valid for counter roll event types.	R	0x0

Table 18-1227. Register Call Summary for Register CPTS_EVENT_LOW

Gigabit Ethernet Switch (GMAC_SW)
- Time Sync Events: [0][1]
- CPTS Interrupt Handling: [2][3]
- CPTS Register Summary: [4]
- CPTS Register Description: [5]

Table 18-1228. CPTS_EVENT_HIGH

Address Offset	0x0000 0038		
Physical Address	0x4848 4C38	**Instance**	CPTS
Description	Upper 32-bits of the event value		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								PORT_NUMBER				EVENT_TYPE				MESSAGE_TYPE				SEQUENCE_ID											

Bits	Field Name	Description	Type	Reset
31:29	RESERVED		R	0x0
28:24	PORT_NUMBER	Port Number - indicates the port number of an ethernet event or the hardware push pin number (1 to 4).	R	0x0
23:20	EVENT_TYPE	Time Sync Event Type 0x0 - Time Stamp Push Event 0x1 - Time Stamp Rollover Event 0x2 - Time Stamp Half Rollover Event 0x3 - Hardware Time Stamp Push Event 0x4 - Ethernet Receive Event 0x5 - Ethernet Transmit Event	R	0x0
19:16	MESSAGE_TYPE	Message type - The message type value that was contained in an ethernet transmit or receive time sync packet. This field is valid only for ethernet transmit or receive events.	R	0x0
15:0	SEQUENCE_ID	Sequence ID - The 16-bit sequence id is the value that was contained in an ethernet transmit or receivetime sync packet. This field is valid only for ethernet transmit or receive events.	R	0x0

Table 18-1229. Register Call Summary for Register CPTS_EVENT_HIGH

Gigabit Ethernet Switch (GMAC_SW)

- Time Sync Events: [0][1][2]
- CPTS Interrupt Handling: [3][4]
- CPTS Register Summary: [5]
- CPTS Register Description: [6]

18.8.6.8 ALE registers

18.8.6.8.1 ALE Register Summary

Table 18-1230. ALE Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	ALE Physical Address
ALE_IDVER	R	32	0x0000 0000	0x4848 4D00
ALE_CONTROL	RW	32	0x0000 0008	0x4848 4D08
ALE_PRESCALE	RW	32	0x0000 0010	0x4848 4D10
ALE_UNKNOWN_VLAN	RW	32	0x0000 0018	0x4848 4D18
ALE_TBLCTL	RW	32	0x0000 0020	0x4848 4D20
ALE_TBLW2	RW	32	0x0000 0034	0x4848 4D34
ALE_TBLW1	RW	32	0x0000 0038	0x4848 4D38
ALE_TBLW0	RW	32	0x0000 003C	0x4848 4D3C
ALE_PORTCTL0	RW	32	0x0000 0040	0x4848 4D40
ALE_PORTCTL1	RW	32	0x0000 0044	0x4848 4D44
ALE_PORTCTL2	RW	32	0x0000 0048	0x4848 4D48
ALE_PORTCTL3	RW	32	0x0000 004C	0x4848 4D4C
ALE_PORTCTL4	RW	32	0x0000 0050	0x4848 4D50
ALE_PORTCTL5	RW	32	0x0000 0054	0x4848 4D54

18.8.6.8.2 ALE Register Description

Table 18-1231. ALE_IDVER

Address Offset	0x0000 0000		
Physical Address	0x4848 4D00	**Instance**	ALE
Description	ADDRESS LOOKUP ENGINE revision register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REVISION																															

Bits	Field Name	Description	Type	Reset
31:0	REVISION	ALE Revision Value	R	0x-

Table 18-1232. Register Call Summary for Register ALE_IDVER

Gigabit Ethernet Switch (GMAC_SW)

- ALE Register Summary: [0]

Table 18-1233. ALE_CONTROL

Address Offset	0x0000 0008		
Physical Address	0x4848 4D08	**Instance**	ALE
Description	Address lookup engine control register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ENABLE_ALE	CLEAR_TABLE	AGE_OUT_NOW	RESERVED																				EN_P0_UNI_FLOOD	LEARN_NO_VID	EN_VID0_MODE	ENABLE_OUI_DENY	BYPASS	RATE_LIMIT_TX	VLAN_AWARE	ENABLE_AUTH_MODE	ENABLE_RATE_LIMIT

Bits	Field Name	Description	Type	Reset
31	ENABLE_ALE	Enable ALE - 0 - Drop all packets 1 - Enable ALE packet processing	RW	0x0
30	CLEAR_TABLE	Clear ALE address table - Setting this bit causes the ALE hardware to write all table bit values to zero. Software must perform a clear table operation as part of the ALE setup/configuration process. Setting this bit causes all ALE accesses to be held up for 64 clocks while the clear is performed. Access to all ALE registers will be blocked (wait states) until the 64 clocks have completed. This bit cannot be read as one because the read is blocked until the clear table is completed at which time this bit is cleared to zero.	RW	0x0
29	AGE_OUT_NOW	Age Out Address Table Now - Setting this bit causes the ALE hardware to remove (free up) any ageable table entry that does not have a set touch bit. This bit is cleared when the age out process has completed. This bit may be read. The age out process takes 4096 clocks best case (no ale packet processing during ageout) and 66550 clocks absolute worst case.	RW	0x0

Bits	Field Name	Description	Type	Reset
28:9	RESERVED		R	0x0
8	EN_P0_UNI_FLOOD	Enable Port 0 (Host Port) unicast flood 0 - do not flood unknown unicast packets to host port (p0) 1 - flood unknown unicast packets to host port (p0)	RW	0x0
7	LEARN_NO_VID	Learn No VID - 0 - VID is learned with the source address 1 - VID is not learned with the source address (source address is not tied to VID).	RW	0x0
6	EN_VID0_MODE	Enable VLAN ID = 0 Mode 0 - Process the packet with VID = PORT_VLAN[11:0]. 1 - Process the packet with VID = 0.	RW	0x0
5	ENABLE_OUI_DENY	Enable OUI Deny Mode - When set this bit indicates that a packet with a non OUI table entry matching source address will be dropped to the host unless the destination address matches a multicast table entry with the super bit set.	RW	0x0
4	BYPASS	ALE Bypass - When set, all packets received on ports 0 and 1 are sent to the host (only to the host).	RW	0x0
3	RATE_LIMIT_TX	Rate Limit Transmit mode - 0 - Broadcast and multicast rate limit counters are received port based 1 - Broadcast and multicast rate limit counters are transmit port based.	RW	0x0
2	VLAN_AWARE	ALE VLAN Aware - Determines what is done if VLAN not found. 0 - Flood if VLAN not found 1 - Drop packet if VLAN not found	RW	0x0
1	ENABLE_AUTH_MODE	Enable MAC Authorization Mode - Mac authorization mode requires that all table entries be made by the host software. There are no learned address in authorization mode and the packet will be dropped if the source address is not found (and the destination address is not a multicast address with the super table entry bit set). 0 - The ALE is not in MAC authorization mode 1 - The ALE is in MAC authorization mode	RW	0x0
0	ENABLE_RATE_LIMIT	Enable Broadcast and Multicast Rate Limit 0 - Broadcast/Multicast rates not limited 1 - Broadcast/Multicast packet reception limited to the port control register rate limit fields.	RW	0x0

Table 18-1234. Register Call Summary for Register ALE_CONTROL

Gigabit Ethernet Switch (GMAC_SW)

- Address Lookup Engine (ALE): [0][1][2]
- FIFO Transmit Queue Control: [3]
- Device Level Ring (DLR) Support: [4]
- ALE Register Summary: [5]

Table 18-1235. ALE_PRESCALE

Address Offset	0x0000 0010		
Physical Address	0x4848 4D10	**Instance**	ALE
Description	Address lookup engine prescale register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								PRESCALE																							

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19:0	PRESCALE	ALE Prescale Register - The input clock is divided by this value for use in the multicast/broadcast rate limiters. The minimum operating value is 0x10. The prescaler is off when the value is zero.	RW	0x0

Table 18-1236. Register Call Summary for Register ALE_PRESCALE

Gigabit Ethernet Switch (GMAC_SW)
- ALE Register Summary: [0]

Table 18-1237. ALE_UNKNOWN_VLAN

Address Offset	0x0000 0018		
Physical Address	0x4848 4D18	**Instance**	ALE
Description	Address lookup engine unknown vlan register		
Type	RW		

31 30	29 28 27 26 25 24	23 22	21 20 19 18 17 16	15 14	13 12 11 10 9 8	7 6	5 4 3 2 1 0
RESERVED	UNKNOWN_FORCE_UNTAGGED_EGRESS	RESERVED	UNKNOWN_REG_MCAST_FLOOD_MASK	RESERVED	UNKNOWN_MCAST_FLOOD_MASK	RESERVED	UNKNOWN_VLAN_MEMBER_LIST

Bits	Field Name	Description	Type	Reset
31:30	RESERVED		R	0x0
29:24	UNKNOWN_FORCE_UNTAGGED_EGRESS	Unknown VLAN Force Untagged Egress.	RW	0x0
23:22	RESERVED		R	0x0
21:16	UNKNOWN_REG_MCAST_FLOOD_MASK	Unknown VLAN Registered Multicast Flood Mask	RW	0x0
15:14	RESERVED		R	0x0
13:8	UNKNOWN_MCAST_FLOOD_MASK	Unknown VLAN Multicast Flood Mask	RW	0x0
7:6	RESERVED		R	0x0
5:0	UNKNOWN_VLAN_MEMBER_LIST	Unknown VLAN Member List	RW	0x0

Table 18-1238. Register Call Summary for Register ALE_UNKNOWN_VLAN

Gigabit Ethernet Switch (GMAC_SW)
- ALE Register Summary: [0]

Table 18-1239. ALE_TBLCTL

Address Offset	0x0000 0020		
Physical Address	0x4848 4D20	**Instance**	ALE
Description	Address lookup engine table control		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WRITE_RDZ	RESERVED																							ENTRY_POINTER							

Bits	Field Name	Description	Type	Reset
31	WRITE_RDZ	Write Bit - This bit is always read as zero. Writing a 1 to this bit causes the three table word register values to be written to the entry_pointer location in the address table. Writing a 0 to this bit causes the three table word register values to be loaded from the entry_pointer location in the address table so that they may be subsequently read. A read of any ALE address location will be stalled until the read or write has completed.	RW	0x0
30:10	RESERVED		R	0x0
9:0	ENTRY_POINTER	Table Entry Pointer - The entry_pointer contains the table entry value that will be read/written with accesses to the table word registers.	RW	0x0

Table 18-1240. Register Call Summary for Register ALE_TBLCTL

Gigabit Ethernet Switch (GMAC_SW)

- ALE Register Summary: [0]

Table 18-1241. ALE_TBLW2

Address Offset	0x0000 0034		
Physical Address	0x4848 4D34	**Instance**	ALE
Description	Address lookup engine table word 2 register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								ENTRY71_64							

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x0
7:0	ENTRY71_64	Table entry bits 71:64	RW	0x0

Table 18-1242. Register Call Summary for Register ALE_TBLW2

Gigabit Ethernet Switch (GMAC_SW)

- ALE Register Summary: [0]

Table 18-1243. ALE_TBLW1

Address Offset	0x0000 0038		
Physical Address	0x4848 4D38	**Instance**	ALE
Description	Address lookup engine table word 1 register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ENTRY63_32																															

Bits	Field Name	Description	Type	Reset
31:0	ENTRY63_32	Table entry bits 63:32	RW	0x0

Table 18-1244. Register Call Summary for Register ALE_TBLW1

Gigabit Ethernet Switch (GMAC_SW)

- ALE Register Summary: [0]

Table 18-1245. ALE_TBLW0

Address Offset	0x0000 003C		
Physical Address	0x4848 4D3C	**Instance**	ALE
Description	Address lookup engine table word 0 register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ENTRY31_0																															

Bits	Field Name	Description	Type	Reset
31:0	ENTRY31_0	Table entry bits 31:0	RW	0x0

Table 18-1246. Register Call Summary for Register ALE_TBLW0

Gigabit Ethernet Switch (GMAC_SW)

- ALE Register Summary: [0]

Table 18-1247. ALE_PORTCTL0

Address Offset	0x0000 0040		
Physical Address	0x4848 4D40	**Instance**	ALE
Description	Address lookup engine port 0 control register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BCAST_LIMIT								MCAST_LIMIT								RESERVED										NO_SA_UPDATE	NO_LEARN	VID_INGRESS_CHECK	DROP_UNTAGGED	PORT_STATE			

Bits	Field Name	Description	Type	Reset
31:24	BCAST_LIMIT	Broadcast Packet Rate Limit - Each prescale pulse loads this field into the port broadcast rate limit counter. The port counters are decremented with each packet received or transmitted depending on whether the mode is transmit or receive. If the counters decrement to zero, then further packets are rate limited until the next prescale pulse. Broadcast rate limiting is enabled by a non-zero value in this field	RW	0x0
23:16	MCAST_LIMIT	Multicast Packet Rate Limit - Each prescale pulse loads this field into the port multicast rate limit counter. The port counters are decremented with each packet received or transmitted depending on whether the mode is transmit or receive. If the counters decrement to zero, then further packets are rate limited until the next prescale pulse. Multicast rate limiting is enabled by a non-zero value in this field.	RW	0x0
15:6	RESERVED		R	0x0
5	NO_SA_UPDATE	No Souce Address Update - When set the port is disabled from updating the source port number in an ALE table entry.	RW	0x0
4	NO_LEARN	No Learn Mode - When set the port is disabled from learning an address.	RW	0x0
3	VID_INGRESS_CHECK	VLAN ID Ingress Check - If VLAN not found then drop the packet. Packets with an unknown (default) VLAN will be dropped.	RW	0x0
2	DROP_UNTAGGED	Drop Untagged Packets - Drop non-VLAN tagged ingress packets.	RW	0x0
1:0	PORT_STATE	Port State 0x0 - Disabled 0x1 - Blocked 0x2 - Learn 0x3 - Forward	RW	0x0

Table 18-1248. Register Call Summary for Register ALE_PORTCTL0

Gigabit Ethernet Switch (GMAC_SW)
- ALE Register Summary: [0]

Table 18-1249. ALE_PORTCTL1

Address Offset	0x0000 0044		
Physical Address	0x4848 4D44	**Instance**	ALE
Description	Address lookup engine port 1 control register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BCAST_LIMIT								MCAST_LIMIT								RESERVED										NO_SA_UPDATE	NO_LEARN	VID_INGRESS_CHECK	DROP_UNTAGGED	PORT_STATE			

Bits	Field Name	Description	Type	Reset
31:24	BCAST_LIMIT	Broadcast Packet Rate Limit - Each prescale pulse loads this field into the port broadcast rate limit counter. The port counters are decremented with each packet received or transmitted depending on whether the mode is transmit or receive. If the counters decrement to zero, then further packets are rate limited until the next prescale pulse. Broadcast rate limiting is enabled by a non-zero value in this field	RW	0x0
23:16	MCAST_LIMIT	Multicast Packet Rate Limit - Each prescale pulse loads this field into the port multicast rate limit counter. The port counters are decremented with each packet received or transmitted depending on whether the mode is transmit or receive. If the counters decrement to zero, then further packets are rate limited until the next prescale pulse. Multicast rate limiting is enabled by a non-zero value in this field.	RW	0x0
15:6	RESERVED		R	0x0
5	NO_SA_UPDATE	No Souce Address Update - When set the port is disabled from updating the source port number in an ALE table entry.	RW	0x0
4	NO_LEARN	No Learn Mode - When set the port is disabled from learning an address.	RW	0x0
3	VID_INGRESS_CHECK	VLAN ID Ingress Check - If VLAN not found then drop the packet. Packets with an unknown (default) VLAN will be dropped.	RW	0x0
2	DROP_UNTAGGED	Drop Untagged Packets - Drop non-VLAN tagged ingress packets.	RW	0x0
1:0	PORT_STATE	Port State 0x0 - Disabled 0x1 - Blocked 0x2 - Learn 0x3 - Forward	RW	0x0

Table 18-1250. Register Call Summary for Register ALE_PORTCTL1

Gigabit Ethernet Switch (GMAC_SW)

- ALE Register Summary: [0]

Table 18-1251. ALE_PORTCTL2

Address Offset	0x0000 0048		
Physical Address	0x4848 4D48	**Instance**	ALE
Description	Address lookup engine port 2 control register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BCAST_LIMIT								MCAST_LIMIT								RESERVED										NO_SA_UPDATE	NO_LEARN	VID_INGRESS_CHECK	DROP_UNTAGGED	PORT_STATE		

Bits	Field Name	Description	Type	Reset
31:24	BCAST_LIMIT	Broadcast Packet Rate Limit - Each prescale pulse loads this field into the port broadcast rate limit counter. The port counters are decremented with each packet received or transmitted depending on whether the mode is transmit or receive. If the counters decrement to zero, then further packets are rate limited until the next prescale pulse. Broadcast rate limiting is enabled by a non-zero value in this field	RW	0x0
23:16	MCAST_LIMIT	Multicast Packet Rate Limit - Each prescale pulse loads this field into the port multicast rate limit counter. The port counters are decremented with each packet received or transmitted depending on whether the mode is transmit or receive. If the counters decrement to zero, then further packets are rate limited until the next prescale pulse. Multicast rate limiting is enabled by a non-zero value in this field.	RW	0x0
15:6	RESERVED		R	0x0
5	NO_SA_UPDATE	No Souce Address Update - When set the port is disabled from updating the source port number in an ALE table entry.	RW	0x0
4	NO_LEARN	No Learn Mode - When set the port is disabled from learning an address.	RW	0x0
3	VID_INGRESS_CHECK	VLAN ID Ingress Check - If VLAN not found then drop the packet. Packets with an unknown (default) VLAN will be dropped.	RW	0x0
2	DROP_UNTAGGED	Drop Untagged Packets - Drop non-VLAN tagged ingress packets.	RW	0x0
1:0	PORT_STATE	Port State 0x0 - Disabled 0x1 - Blocked 0x2 - Learn 0x3 - Forward	RW	0x0

Table 18-1252. Register Call Summary for Register ALE_PORTCTL2

Gigabit Ethernet Switch (GMAC_SW)
- ALE Register Summary: [0]

Table 18-1253. ALE_PORTCTL3

Address Offset	0x0000 004C		
Physical Address	0x4848 4D4C	**Instance**	ALE
Description	Address lookup engine port 3 control register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BCAST_LIMIT								MCAST_LIMIT								RESERVED										NO_SA_UPDATE	NO_LEARN	VID_INGRESS_CHECK	DROP_UNTAGGED	PORT_STATE			

Bits	Field Name	Description	Type	Reset
31:24	BCAST_LIMIT	Broadcast Packet Rate Limit - Each prescale pulse loads this field into the port broadcast rate limit counter. The port counters are decremented with each packet received or transmitted depending on whether the mode is transmit or receive. If the counters decrement to zero, then further packets are rate limited until the next prescale pulse. Broadcast rate limiting is enabled by a non-zero value in this field	RW	0x0
23:16	MCAST_LIMIT	Multicast Packet Rate Limit - Each prescale pulse loads this field into the port multicast rate limit counter. The port counters are decremented with each packet received or transmitted depending on whether the mode is transmit or receive. If the counters decrement to zero, then further packets are rate limited until the next prescale pulse. Multicast rate limiting is enabled by a non-zero value in this field.	RW	0x0
15:6	RESERVED		R	0x0
5	NO_SA_UPDATE	No Souce Address Update - When set the port is disabled from updating the source port number in an ALE table entry.	RW	0x0
4	NO_LEARN	No Learn Mode - When set the port is disabled from learning an address.	RW	0x0
3	VID_INGRESS_CHECK	VLAN ID Ingress Check - If VLAN not found then drop the packet. Packets with an unknown (default) VLAN will be dropped.	RW	0x0
2	DROP_UNTAGGED	Drop Untagged Packets - Drop non-VLAN tagged ingress packets.	RW	0x0
1:0	PORT_STATE	Port State 0x0 - Disabled 0x1 - Blocked 0x2 - Learn 0x3 - Forward	RW	0x0

Table 18-1254. Register Call Summary for Register ALE_PORTCTL3

Gigabit Ethernet Switch (GMAC_SW)

- ALE Register Summary: [0]

Table 18-1255. ALE_PORTCTL4

Address Offset	0x0000 0050		
Physical Address	0x4848 4D50	**Instance**	ALE
Description	Address lookup engine port 4 control register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BCAST_LIMIT								MCAST_LIMIT								RESERVED										NO_SA_UPDATE	NO_LEARN	VID_INGRESS_CHECK	DROP_UNTAGGED	PORT_STATE			

Bits	Field Name	Description	Type	Reset
31:24	BCAST_LIMIT	Broadcast Packet Rate Limit - Each prescale pulse loads this field into the port broadcast rate limit counter. The port counters are decremented with each packet received or transmitted depending on whether the mode is transmit or receive. If the counters decrement to zero, then further packets are rate limited until the next prescale pulse. Broadcast rate limiting is enabled by a non-zero value in this field	RW	0x0
23:16	MCAST_LIMIT	Multicast Packet Rate Limit - Each prescale pulse loads this field into the port multicast rate limit counter. The port counters are decremented with each packet received or transmitted depending on whether the mode is transmit or receive. If the counters decrement to zero, then further packets are rate limited until the next prescale pulse. Multicast rate limiting is enabled by a non-zero value in this field.	RW	0x0
15:6	RESERVED		R	0x0
5	NO_SA_UPDATE	No Souce Address Update - When set the port is disabled from updating the source port number in an ALE table entry.	RW	0x0
4	NO_LEARN	No Learn Mode - When set the port is disabled from learning an address.	RW	0x0
3	VID_INGRESS_CHECK	VLAN ID Ingress Check - If VLAN not found then drop the packet. Packets with an unknown (default) VLAN will be dropped.	RW	0x0
2	DROP_UNTAGGED	Drop Untagged Packets - Drop non-VLAN tagged ingress packets.	RW	0x0
1:0	PORT_STATE	Port State 0x0 - Disabled 0x1 - Blocked 0x2 - Learn 0x3 - Forward	RW	0x0

Table 18-1256. Register Call Summary for Register ALE_PORTCTL4

Gigabit Ethernet Switch (GMAC_SW)

- ALE Register Summary: [0]

Table 18-1257. ALE_PORTCTL5

Address Offset	0x0000 0054		
Physical Address	0x4848 4D54	**Instance**	ALE
Description	Address lookup engine port 5 control register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
BCAST_LIMIT								MCAST_LIMIT								RESERVED										NO_SA_UPDATE	NO_LEARN	VID_INGRESS_CHECK	DROP_UNTAGGED	PORT_STATE			

Bits	Field Name	Description	Type	Reset
31:24	BCAST_LIMIT	Broadcast Packet Rate Limit - Each prescale pulse loads this field into the port broadcast rate limit counter. The port counters are decremented with each packet received or transmitted depending on whether the mode is transmit or receive. If the counters decrement to zero, then further packets are rate limited until the next prescale pulse. Broadcast rate limiting is enabled by a non-zero value in this field	RW	0x0
23:16	MCAST_LIMIT	Multicast Packet Rate Limit - Each prescale pulse loads this field into the port multicast rate limit counter. The port counters are decremented with each packet received or transmitted depending on whether the mode is transmit or receive. If the counters decrement to zero, then further packets are rate limited until the next prescale pulse. Multicast rate limiting is enabled by a non-zero value in this field.	RW	0x0
15:6	RESERVED		R	0x0
5	NO_SA_UPDATE	No Souce Address Update - When set the port is disabled from updating the source port number in an ALE table entry.	RW	0x0
4	NO_LEARN	No Learn Mode - When set the port is disabled from learning an address.	RW	0x0
3	VID_INGRESS_CHECK	VLAN ID Ingress Check - If VLAN not found then drop the packet. Packets with an unknown (default) VLAN will be dropped.	RW	0x0
2	DROP_UNTAGGED	Drop Untagged Packets - Drop non-VLAN tagged ingress packets.	RW	0x0
1:0	PORT_STATE	Port State 0x0 - Disabled 0x1 - Blocked 0x2 - Learn 0x3 - Forward	RW	0x0

Table 18-1258. Register Call Summary for Register ALE_PORTCTL5

Gigabit Ethernet Switch (GMAC_SW)
- ALE Register Summary: [0]

18.8.6.9 SL registers

18.8.6.9.1 SL Register Summary

Table 18-1259. SL Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	SL1 Physical Address	SL2 Physical Address
SL_IDVER	R	32	0x0000 0000	0x4848 4D80	0x4848 4DC0
SL_MACCONTROL	RW	32	0x0000 0004	0x4848 4D84	0x4848 4DC4
SL_MACSTATUS	R	32	0x0000 0008	0x4848 4D88	0x4848 4DC8
SL_SOFT_RESET	RW	32	0x0000 000C	0x4848 4D8C	0x4848 4DCC
SL_RX_MAXLEN	RW	32	0x0000 0010	0x4848 4D90	0x4848 4DD0
SL_BOFFTEST	RW	32	0x0000 0014	0x4848 4D94	0x4848 4DD4
SL_RX_PAUSE	R	32	0x0000 0018	0x4848 4D98	0x4848 4DD8
SL_TX_PAUSE	R	32	0x0000 001C	0x4848 4D9C	0x4848 4DDC
SL_EMCONTROL	RW	32	0x0000 0020	0x4848 4DA0	0x4848 4DE0
SL_RX_PRI_MAP	RW	32	0x0000 0024	0x4848 4DA4	0x4848 4DE4
SL_TX_GAP	RW	32	0x0000 0028	0x4848 4DA8	0x4848 4DE8

18.8.6.9.2 SL Register Description

Table 18-1260. SL_IDVER

Address Offset	0x0000 0000		
Physical Address	0x4848 4D80 0x4848 4DC0	**Instance**	SL1 SL2
Description	CPGMAC_SL revision register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REVISION																															

Bits	Field Name	Description	Type	Reset
31:0	REVISION	CPGMAC_SL revision Value	R	0x-

Table 18-1261. Register Call Summary for Register SL_IDVER

Gigabit Ethernet Switch (GMAC_SW)

- SL Register Summary: [0]

Table 18-1262. SL_MACCONTROL

Address Offset	0x0000 0004		
Physical Address	0x4848 4D84 0x4848 4DC4	**Instance**	SL1 SL2
Description	CPGMAC_SL MAC control register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								RX_CMF_EN	RX_CSF_EN	RX_CEF_EN	TX_SHORT_GAP_LIM_EN	RESERVED		EXT_EN	GIG_FORCE	IFCTL_B	IFCTL_A	RESERVED			CMD_IDLE	TX_SHORT_GAP_EN	RESERVED	GIG	TX_PACE	GMII_EN	TX_FLOW_EN	RX_FLOW_EN	MTEST	LOOPBACK	FULLDUPLEX

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24	RX_CMF_EN	RX Copy MAC Control Frames Enable - Enables MAC control frames to be transferred to memory. MAC control frames are normally acted upon (if enabled), but not copied to memory. MAC control frames that are pause frames will be acted upon if enabled in the SL_MACCONTROL register, regardless of the value of RX_CMF_EN. Frames transferred to memory due to RX_CMF_EN will have the control bit set in their EOP buffer descriptor. 0 - MAC control frames are filtered (but acted upon if enabled). 1 - MAC control frames are transferred to memory.	RW	0x0

Bits	Field Name	Description	Type	Reset
23	RX_CSF_EN	RX Copy Short Frames Enable - Enables frames or fragments shorter than 64 bytes to be copied to memory. Frames transferred to memory due to RX_CSF_EN will have the fragment or undersized bit set in their receive footer. Fragments are short frames that contain CRC/align/code errors and undersized are short frames without errors. 0 - Short frames are filtered 1 - Short frames are transferred to memory.	RW	0x0
22	RX_CEF_EN	RX Copy Error Frames Enable - Enables frames containing errors to be transferred to memory. The appropriate error bit will be set in the frame receive footer. Frames containing errors will be filtered when RX_CEF _EN is not set. 0 - Frames containing errors are filtered. 1 - Frames containing errors are transferred to memory.	RW	0x0
21	TX_SHORT_GAP_LIM_EN	Transmit Short Gap Limit Enable When set this bit limits the number of short gap packets transmitted to 100ppm. Each time a short gap packet is sent, a counter is loaded with 10,000 and decremented on each wireside clock. Another short gap packet will not be sent out until the counter decrements to zero. This mode is included to preclude the host from filling up the FIFO and sending every packet out with short gap which would violate the maximum number of packets per second allowed.	RW	0x0
20:19	RESERVED		R	0x0
18	EXT_EN	Control Enable - Enables the fullduplex and gigabit mode to be selected from the FULLDUPLEX_IN and GIG_IN input signals and not from the FULLDUPLEX and GIG bits in this register. The FULLDUPLEX_MODE bit reflects the actual fullduplex mode selected 0 - Use this setting for RMII/GMII mode . 1 - Use this setting for RGMII mode	RW	0x0
17	GIG_FORCE	Gigabit Mode Force - This bit is used to force the CPGMAC_SL into gigabit mode if the input GMII_MTCLK has been stopped by the PHY.	RW	0x0
16	IFCTL_B	Interface Control B (NOT FUNCTIONAL) 0 - 10Mbps operation 1 - 100Mbps operation	RW	0x0
15	IFCTL_A	Interface Control A 0 - 10Mbps operation 1 - 100Mbps operation	RW	0x0
14:12	RESERVED		R	0x0
11	CMD_IDLE	Command Idle 0 - Idle not commanded 1 - Idle Commanded (read IDLE in SL_MACSTATUS)	RW	0x0
10	TX_SHORT_GAP_EN	Transmit Short Gap Enable 0 - Transmit with a short IPG is disabled 1 - Transmit with a short IPG (when TX_SHORT_GAP input is asserted) is enabled.	RW	0x0
9:8	RESERVED		R	0x0
7	GIG	Gigabit Mode - 0 - 10/100 mode 1 - Gigabit mode (full duplex only) The GIG_OUT output is the value of this bit.	RW	0x0
6	TX_PACE	Transmit Pacing Enable 0 - Transmit Pacing Disabled 1 - Transmit Pacing Enabled	RW	0x0
5	GMII_EN	GMII Enable - 0 - GMII RX and TX held in reset. 1 - GMII RX and TX released from reset.	RW	0x0

Bits	Field Name	Description	Type	Reset
4	TX_FLOW_EN	Transmit Flow Control Enable - Determines if incoming pause frames are acted upon in full-duplex mode. Incoming pause frames are not acted upon in half-duplex mode regardless of this bit setting. The RX_MBP_Enable bits determine whether or not received pause frames are transferred to memory. 0 - Transmit Flow Control Disabled. Full-duplex mode - Incoming pause frames are not acted upon. 1 - Transmit Flow Control Enabled . Full-duplex mode - Incoming pause frames are acted upon.	RW	0x0
3	RX_FLOW_EN	Receive Flow Control Enable - 0 - Receive Flow Control Disabled Half-duplex mode - No flow control generated collisions are sent. Full-duplex mode - No outgoing pause frames are sent. 1 - Receive Flow Control Enabled Half-duplex mode - Collisions are initiated when receive flow control is triggered. Full-duplex mode - Outgoing pause frames are sent when receive flow control is triggered.	RW	0x0
2	MTEST	Manufacturing Test mode - This bit must be set to allow writes to the SL_BOFFTEST and SL_RX_PAUSE/SL_TX_PAUSE registers.	RW	0x0
1	LOOPBACK	Loop Back Mode - Loopback mode forces internal fullduplex mode regardless of whether the FULLDUPLEX bit is set or not. The LOOPBACK bit should be changed only when GMII_EN is deasserted. 0 - Not looped back 1 - Loop Back Mode enabled	RW	0x0
0	FULLDUPLEX	Full Duplex mode - Gigabit mode forces fullduplex mode regardless of whether the FULLDUPLEX bit is set or not. The FULLDUPLEX_OUT output is the value of this register bit 0 - half duplex mode 1 - full duplex mode	RW	0x0

Table 18-1263. Register Call Summary for Register SL_MACCONTROL

Gigabit Ethernet Switch (GMAC_SW)

- Address Lookup Engine (ALE): [1][2]
- Ethernet MAC Sliver (CPGMAC_SL): [5][6][7]
- Flow Control: [8][9][10][11][12][13][14][15]
- Short Gap: [16]
- RX Buffer Descriptors: [17][18][19]
- MDIO Functional Description: [20]
- STATS Register Description: [21]
- SL Register Summary: [22]
- SL Register Description: [23]

Table 18-1264. SL_MACSTATUS

Address Offset	0x0000 0008		
Physical Address	0x4848 4D88 0x4848 4DC8	**Instance**	SL1 SL2
Description	CPGMAC_SL MAC status register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IDLE	RESERVED																										EXT_GIG	EXT_FULLDUPLEX	RESERVED	RX_FLOW_ACT	TX_FLOW_ACT

Bits	Field Name	Description	Type	Reset
31	IDLE	CPGMAC_SL IDLE - The CPGMAC_SL is in the idle state (valid after an idle command) 0 - The CPGMAC_SL is not in the idle state. 1 - The CPGMAC_SL is in the idle state.	R	0x1
30:5	RESERVED		R	0x0
4	EXT_GIG	External GIG - This is the value of the EXT_GIG input bit.	R	0x0
3	EXT_FULLDUPLEX	External Fullduplex - This is the value of the EXT_FULLDUPLEX input bit.	R	0x0
2	RESERVED		R	0x0
1	RX_FLOW_ACT	Receive Flow Control Active - When asserted, indicates that receive flow control is enabled and triggered.	R	0x0
0	TX_FLOW_ACT	Transmit Flow Control Active - When asserted, this bit indicates that the pause time period is being observed for a received pause frame. No new transmissions will begin while this bit is asserted except for the transmission of pause frames. Any transmission in progress when this bit is asserted will complete.	R	0x0

Table 18-1265. Register Call Summary for Register SL_MACSTATUS

Gigabit Ethernet Switch (GMAC_SW)

- SL Register Summary: [0]
- SL Register Description: [1]

Table 18-1266. SL_SOFT_RESET

Address Offset	0x0000 000C		
Physical Address	0x4848 4D8C 0x4848 4DCC	**Instance**	SL1 SL2
Description	CPGMAC_SL soft reset register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															SOFT_RESET

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	SOFT_RESET	Software reset - Writing a one to this bit causes the CPGMAC_SL logic to be reset. After writing a one to this bit, it may be polled to determine if the reset has occurred. If a one is read, the reset has not yet occurred. If a zero is read then reset has occurred.	RW	0x0

Table 18-1267. Register Call Summary for Register SL_SOFT_RESET

Gigabit Ethernet Switch (GMAC_SW)

- Software Reset: [0]
- SL Register Summary: [1]

Table 18-1268. SL_RX_MAXLEN

Address Offset	0x0000 0010		
Physical Address	0x4848 4D90 0x4848 4DD0	**Instance**	SL1 SL2
Description	CPGMAC_SL RX Maximum length register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RX_MAXLEN															

Bits	Field Name	Description	Type	Reset
31:14	RESERVED		R	0x0
13:0	RX_MAXLEN	RX Maximum Frame Length - This field determines the maximum length of a received frame. The reset value is 1518 (dec). Frames with byte counts greater than rx_maxlen are long frames. Long frames with no errors are oversized frames. Long frames with CRC, code, or alignment error are jabber frames. The maximum value is 16,383.	RW	0x5EE

Table 18-1269. Register Call Summary for Register SL_RX_MAXLEN

Gigabit Ethernet Switch (GMAC_SW)

- Ethernet MAC Sliver (CPGMAC_SL): [0][1]
- Flow Control: [2]
- STATS Register Description: [3][4][5][6][7][8][9][10][11][12][13][14][15][16][17]
- SL Register Summary: [18]

Table 18-1270. SL_BOFFTEST

Address Offset	0x0000 0014		
Physical Address	0x4848 4D94 0x4848 4DD4	**Instance**	SL1 SL2
Description	CPGMAC_SL backoff test register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								PACEVAL								RNDNUM								COLL_COUNT								RESERVED								TX_BACKOFF							

Bits	Field Name	Description	Type	Reset
31	RESERVED		R	0x0
30:26	PACEVAL	Pacing Register Current Value. A non-zero value in this field indicates that transmit pacing is active. A transmit frame collision or deferral causes PACEVAL to loaded with decimal 31, good frame transmissions (with no collisions or deferrals) cause PACEVAL to be decremented down to zero. When PACEVAL is nonzero, the transmitter delays 4 IPGs between new frame transmissions after each successfully transmitted frame that had no deferrals or collisions. Transmit pacing helps reduce 'capture' effects improving overall network bandwidth.	RW	0x0
25:16	RNDNUM	Backoff Random Number Generator - This field allows the Backoff Random Number Generator to be read (or written in test mode only). This field can be written only when mtest has previously been set. Reading this field returns the generator's current value. The value is reset to zero and begins counting on the clock after the deassertion of reset.	RW	0x0
15:12	COLL_COUNT	Collision Count - The number of collisions the current frame has experienced.	R	0x0
11:10	RESERVED		R	0x0
9:0	TX_BACKOFF	Backoff Count - This field allows the current value of the backoff counter to be observed for test purposes. This field is loaded automatically according to the backoff algorithm, and is decremented by one for each slot time after the collision.	R	0x0

Table 18-1271. Register Call Summary for Register SL_BOFFTEST

Gigabit Ethernet Switch (GMAC_SW)

- SL Register Summary: [0]
- SL Register Description: [1]

Table 18-1272. SL_RX_PAUSE

Address Offset	0x0000 0018		
Physical Address	0x4848 4D98 0x4848 4DD8	**Instance**	SL1 SL2
Description	CPGMAC_SL receive pause timer register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																RX_PAUSETIMER															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0
15:0	RX_PAUSETIMER	RX Pause Timer Value - This field allows the contents of the receive pause timer to be observed (and written in test mode). The receive pause timer is loaded with 0xFF00 when the CPGMAC_SL sends an outgoing pause frame (with pause time of 0xFFFF). The receive pause timer is decremented at slot time intervals. If the receive pause timer decrements to zero, then another outgoing pause frame will be sent and the load/decrement process will be repeated.	R	0x0

Table 18-1273. Register Call Summary for Register SL_RX_PAUSE

Gigabit Ethernet Switch (GMAC_SW)

- SL Register Summary: [0]
- SL Register Description: [1]

Table 18-1274. SL_TX_PAUSE

Address Offset	0x0000 001C		
Physical Address	0x4848 4D9C 0x4848 4DDC	**Instance**	SL1 SL2
Description	CPGMAC_SL transmit pause timer register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																TX_PAUSETIMER															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0
15:0	TX_PAUSETIMER	TX Pause Timer Value - This field allows the contents of the transmit pause timer to be observed (and written in test mode). The transmit pause timer is loaded by a received (incoming) pause frame, and then decremented, at slottime intervals, down to zero at which time CPGMAC_SL transmit frames are again enabled.	R	0x0

Table 18-1275. Register Call Summary for Register SL_TX_PAUSE

Gigabit Ethernet Switch (GMAC_SW)

- SL Register Summary: [0]
- SL Register Description: [1]

Table 18-1276. SL_EMCONTROL

Address Offset	0x0000 0020		
Physical Address	0x4848 4DA0 0x4848 4DE0	**Instance**	SL1 SL2
Description	CPGMAC_SL emulation control register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														SOFT	FREE

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1	SOFT	Emulation Soft Bit. Emulation soft bit. This bit is used in conjunction with FREE bit to determine the emulation suspend mode. This bit has no effect if FREE = 1.	RW	0x0
0	FREE	Emulation Free Bit. Emulation free bit. This bit is used in conjunction with SOFT bit to determine the emulation suspend mode.	RW	0x0

Table 18-1277. Register Call Summary for Register SL_EMCONTROL

Gigabit Ethernet Switch (GMAC_SW)

- Emulation Control: [0]
- SL Register Summary: [1]

Table 18-1278. SL_RX_PRI_MAP

Address Offset	0x0000 0024		
Physical Address	0x4848 4DA4 0x4848 4DE4	**Instance**	SL1 SL2
Description	CPGMAC_SL RX packet priority to header priority mapping register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED	PRI7			RESERVED	PRI6			RESERVED	PRI5			RESERVED	PRI4			RESERVED	PRI3			RESERVED	PRI2			RESERVED	PRI1			RESERVED	PRI0		

Bits	Field Name	Description	Type	Reset
31	RESERVED		R	0x0
30:28	PRI7	Priority 7 - A packet priority of 0x7 is mapped (changed) to this value.	RW	0x7
27	RESERVED		R	0x0
26:24	PRI6	Priority 6 - A packet priority of 0x6 is mapped (changed) to this value.	RW	0x6
23	RESERVED		R	0x0
22:20	PRI5	Priority 5 - A packet priority of 0x5 is mapped (changed) to this value.	RW	0x5
19	RESERVED		R	0x0
18:16	PRI4	Priority 4 - A packet priority of 0x4 is mapped (changed) to this value.	RW	0x4
15	RESERVED		R	0x0
14:12	PRI3	Priority 3 - A packet priority of 0x3 is mapped (changed) to this value.	RW	0x3
11	RESERVED		R	0x0
10:8	PRI2	Priority 2 - A packet priority of 0x2 is mapped (changed) to this value.	RW	0x2
7	RESERVED		R	0x0
6:4	PRI1	Priority 1 - A packet priority of 0x1 is mapped (changed) to this value.	RW	0x1
3	RESERVED		R	0x0
2:0	PRI0	Priority 0 - A packet priority of 0x0 is mapped (changed) to this value.	RW	0x0

Table 18-1279. Register Call Summary for Register SL_RX_PRI_MAP

Gigabit Ethernet Switch (GMAC_SW)

- SL Register Summary: [0]

Table 18-1280. SL_TX_GAP

Address Offset	0x0000 0028		
Physical Address	0x4848 4DA8 0x4848 4DE8	**Instance**	SL1 SL2
Description	Transmit inter-packet gap register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								TX_GAP							

Bits	Field Name	Description	Type	Reset
31:9	RESERVED		R	0x0
8:0	TX_GAP	Transmit Inter-Packet Gap	RW	0xC

Table 18-1281. Register Call Summary for Register SL_TX_GAP

Gigabit Ethernet Switch (GMAC_SW)

- SL Register Summary: [1]

18.8.6.10 MDIO registers

18.8.6.10.1 MDIO Register Summary

Table 18-1282. MDIO Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	MDIO Physical Address
MDIO_VER	RW	32	0x0000 0000	0x4848 5000
MDIO_CONTROL	RW	32	0x0000 0004	0x4848 5004
MDIO_ALIVE	RW	32	0x0000 0008	0x4848 5008
MDIO_LINK	R	32	0x0000 000C	0x4848 500C
MDIO_LINKINTRAW	RW	32	0x0000 0010	0x4848 5010
MDIO_LINKINTMASKED	RW	32	0x0000 0014	0x4848 5014
MDIO_USERINTRAW	RW	32	0x0000 0020	0x4848 5020
MDIO_USERINTMASKED	RW	32	0x0000 0024	0x4848 5024
MDIO_USERINTMASKSET	RW	32	0x0000 0028	0x4848 5028
MDIO_USERINTMASKCLR	RW	32	0x0000 002C	0x4848 502C
MDIO_USERACCESS0	RW	32	0x0000 0080	0x4848 5080
MDIO_USERPHYSEL0	RW	32	0x0000 0084	0x4848 5084
MDIO_USERACCESS1	RW	32	0x0000 0088	0x4848 5088
MDIO_USERPHYSEL1	RW	32	0x0000 008C	0x4848 508C

18.8.6.10.2 MDIO Register Description

Table 18-1283. MDIO_VER

Address Offset	0x0000 0000		
Physical Address	0x4848 5000	**Instance**	MDIO
Description	MDIO Revision		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REVISION																															

Bits	Field Name	Description	Type	Reset
31:0	REVISION	MDIO revision value	RW	0x-

Table 18-1284. Register Call Summary for Register MDIO_VER

Gigabit Ethernet Switch (GMAC_SW)
- MDIO Register Summary: [0]

Table 18-1285. MDIO_CONTROL

Address Offset	0x0000 0004		
Physical Address	0x4848 5004	**Instance**	MDIO
Description	MDIO Control register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
IDLE	ENABLE	RESERVED	HIGHEST_USER_CHANNEL					RESERVED			PREAMBLE	FAULT	FAULTENB	INTTESTENB	RESERVED	CLKDIV																

Bits	Field Name	Description	Type	Reset
31	IDLE	MDIO state machine IDLE. Set to 1 when the state machine is in the idle state. 0: State machine is not in idle state. 1: State machine is in idle state.	R	0x0
30	ENABLE	Enable control. If the MDIO state machine is active at the time it is disabled, it will complete the current operation before halting and setting the IDLE bit. If using byte access, the ENABLE bit has to be the last bit written in this register. 0: Disables the MDIO state machine. 1: Enable the MDIO state machine.	RW	0x0
29	RESERVED		R	0x0
28:24	HIGHEST_USER_CHANNEL	Highest user channel. This field specifies the highest user access channel that is available in the module and is currently set to 1. This implies that the MDIO_USERACCESS1 register is the highest available user access channel.	R	0x0

Bits	Field Name	Description	Type	Reset
23:21	RESERVED		R	0x0
20	PREAMBLE	Preamble disable. 0: Standard MDIO preamble is used. 1: Disables this device from sending MDIO frame preambles.	RW	0x0
19	FAULT	Fault indicator. This bit is set to 1 if the MDIO pins fail to read back what the device is driving onto them. This indicates a physical layer fault and the module state machine is reset. Writing a 1 to it clears this bit. 0: No failure. 1: Physical layer fault; the MDIO state machine is reset.	RW	0x0
18	FAULTENB	Fault detect enable. This bit has to be set to 1 to enable the physical layer fault detection. 0: Disables the physical layer fault detection. 1: Enables the physical layer fault detection.	RW	0x0
17	INTTESTENB	Interrupt test enable. This bit can be set to 1 to enable the host to set the USERINT and LINKINT bits for test purposes. 0: Interrupt bits are not set. 1: Enables the host to set the USERINT and LINKINT bits for test purposes.	RW	0x0
16	RESERVED		R	0x0
15:0	CLKDIV	Clock divider. This field specifies the division ratio between ICLK and the frequency of MDCLK. MDCLK is disabled when CLKDIV is set to 0. MDCLK frequency = ICLK frequency/(CLKDIV+1).	RW	0x0

Table 18-1286. Register Call Summary for Register MDIO_CONTROL

Gigabit Ethernet Switch (GMAC_SW)

- Interface Clocking: [0]
- MDIO Functional Description: [1]
- Initializing the MDIO Module: [2][3]
- MDIO Register Summary: [4]
- MDIO Register Description: [5][6][7]

Table 18-1287. MDIO_ALIVE

Address Offset	0x0000 0008		
Physical Address	0x4848 5008	**Instance**	MDIO
Description	PHY Alive Status Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ALIVE																															

Bits	Field Name	Description	Type	Reset
31:0	ALIVE	MDIO alive. Each of the 32 bits of this register is set if the most recent access to the PHY with address corresponding to the register bit number was acknowledged by the PHY, the bit is reset if the PHY fails to acknowledge the access. Both the user and polling accesses to a PHY will cause the corresponding alive bit to be updated. The alive bits are only meant to be used to give an indication of the presence or not of a PHY with the corresponding address. Writing a 1 to any bit will clear it, writing a 0 has no effect.	RW	0x0

Table 18-1288. Register Call Summary for Register MDIO_ALIVE

Gigabit Ethernet Switch (GMAC_SW)

- MDIO Functional Description: [0][1]
- Initializing the MDIO Module: [2]
- MDIO Register Summary: [3]

Table 18-1289. MDIO_LINK

Address Offset	0x0000 000C		
Physical Address	0x4848 500C	**Instance**	MDIO
Description	PHY Link Status		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LINK																															

Bits	Field Name	Description	Type	Reset
31:0	LINK	MDIO link state. This register is updated after a read of the Generic Status Register of a PHY. The bit is set if the PHY with the corresponding address has link and the PHY acknowledges the read transaction. The bit is reset if the PHY indicates it does not have link or fails to acknowledge the read transaction. Writes to the register have no effect. In addition, the status of the two PHYs specified in the MDIO_USERPHYSEL registers can be determined using the MLINK input pins (NOT PINNED OUT). This is determined by the LINKSEL bit in the MDIO_USERPHYSEL register.	R	0x0

Table 18-1290. Register Call Summary for Register MDIO_LINK

Gigabit Ethernet Switch (GMAC_SW)

- Miscellaneous Pulse Interrupt (MISC_PULSE): [0][1][2][3]
- MDIO Functional Description: [4]
- Initializing the MDIO Module: [5]
- MDIO Register Summary: [6]

Table 18-1291. MDIO_LINKINTRAW

Address Offset	0x0000 0010		
Physical Address	0x4848 5010	**Instance**	MDIO
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														LINKINTRAW	

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1:0	LINKINTRAW	MDIO link change event, raw value.	RW	0x0

Table 18-1292. Register Call Summary for Register MDIO_LINKINTRAW

Gigabit Ethernet Switch (GMAC_SW)

- MDIO Functional Description: [0]
- MDIO Register Summary: [1]

Table 18-1293. MDIO_LINKINTMASKED

Address Offset	0x0000 0014		
Physical Address	0x4848 5014	**Instance**	MDIO
Description	MDIO Link Status Change Interrupt Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														LINKINTMASKED	

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1:0	LINKINTMASKED	MDIO link change interrupt, masked value. When asserted 1, a bit indicates that there was an MDIO link change event (i.e. change in the MDIO Link register) corresponding to the PHY address in the MDIO_USERPHYSEL register and the corresponding LINKINTENB bit was set. LINKINTMASKED[0] and LINKINTMASKED[1] correspond to MDIO_USERPHYSEL0 and MDIO_USERPHYSEL1, respectively. Writing a 1 will clear the interrupt and writing 0 has no effect. If the INTTESTENB bit in the MDIO_CONTROL register is set, the host may set the LINKINT bits to a 1. This mode may be used for test purposes.	RW	0x0

Table 18-1294. Register Call Summary for Register MDIO_LINKINTMASKED

Gigabit Ethernet Switch (GMAC_SW)

- Miscellaneous Pulse Interrupt (MISC_PULSE): [0][1]
- MDIO Functional Description: [2]
- MDIO Register Summary: [3]

Table 18-1295. MDIO_USERINTRAW

Address Offset	0x0000 0020		
Physical Address	0x4848 5020	**Instance**	MDIO
Description	MDIO User Command Complete Interrupt		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														USERINTRAW	

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1:0	USERINTRAW	Raw value of MDIO user command complete event for the MDIO_USERACCESS1 and MDIO_USERACCESS0 register, respectively. When asserted 1, a bit indicates that the previously scheduled PHY read or write command using that particular MDIO_USERACCESS register has completed. Writing a 1 will clear the event and writing 0 has no effect. If the INTTESTENB bit in the MDIO_CONTROL register is set, the host may set the USERINTRAW bits to a 1. This mode may be used for test purposes.	RW	0x0

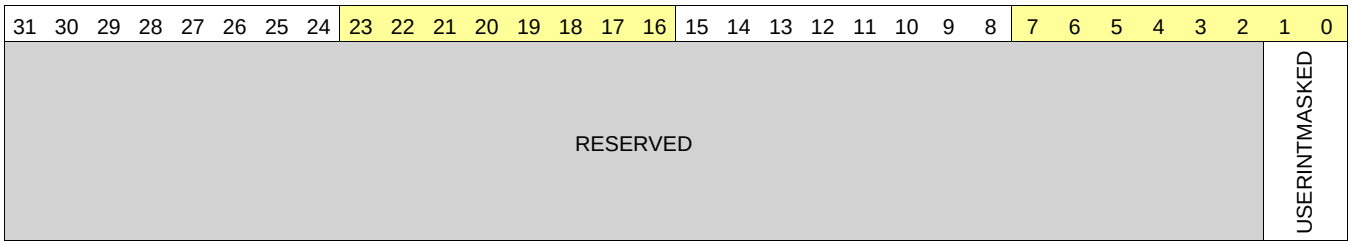
Table 18-1296. Register Call Summary for Register MDIO_USERINTRAW

Gigabit Ethernet Switch (GMAC_SW)

- MDIO Functional Description: [0]
- Writing Data To a PHY Register: [1]
- Reading Data From a PHY Register: [2]
- MDIO Register Summary: [3]

Table 18-1297. MDIO_USERINTMASKED

Address Offset	0x0000 0024		
Physical Address	0x4848 5024	**Instance**	MDIO
Description	MDIO User Command Complete Interrupt		
Type	RW		



Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1:0	USERINTMASKED	Masked value of MDIO user command complete interrupt for the MDIO_USERACCESS1 and MDIO_USERACCESS0 register, respectively. When asserted 1, a bit indicates that the previously scheduled PHY read or write command using that particular MDIO_USERACCESS register has completed and the corresponding USERINTMASKSET bit is set to 1. Writing a 1 will clear the interrupt and writing 0 has no effect. If the INTTESTENB bit in the MDIO_CONTROL register is set, the host may set the USERINTMASKED bits to a 1. This mode may be used for test purposes.	RW	0x0

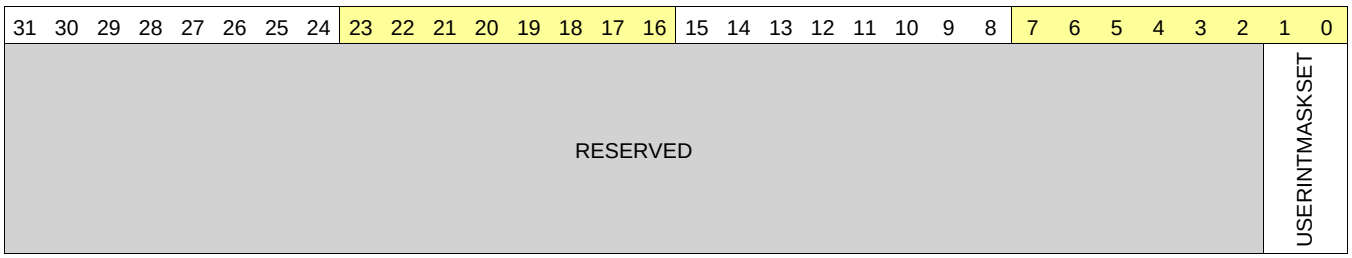
Table 18-1298. Register Call Summary for Register MDIO_USERINTMASKED

Gigabit Ethernet Switch (GMAC_SW)

- Miscellaneous Pulse Interrupt (MISC_PULSE): [0][1]
- MDIO Functional Description: [2]
- Writing Data To a PHY Register: [3]
- Reading Data From a PHY Register: [4]
- MDIO Register Summary: [5]

Table 18-1299. MDIO_USERINTMASKSET

Address Offset	0x0000 0028		
Physical Address	0x4848 5028	**Instance**	MDIO
Description	MDIO User Command Complete Interrupt Mask Set		
Type	RW		



Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1:0	USERINTMASKSET	MDIO user interrupt mask set for USERINTMASKED[1:0], respectively. Writing a bit to 1 will enable MDIO user command complete interrupts for that particular MDIO_USERACCESS register. MDIO user interrupt for a particular MDIO_USERACCESS register is disabled if the corresponding bit is 0. Writing a 0 to this register has no effect.	RW	0x0

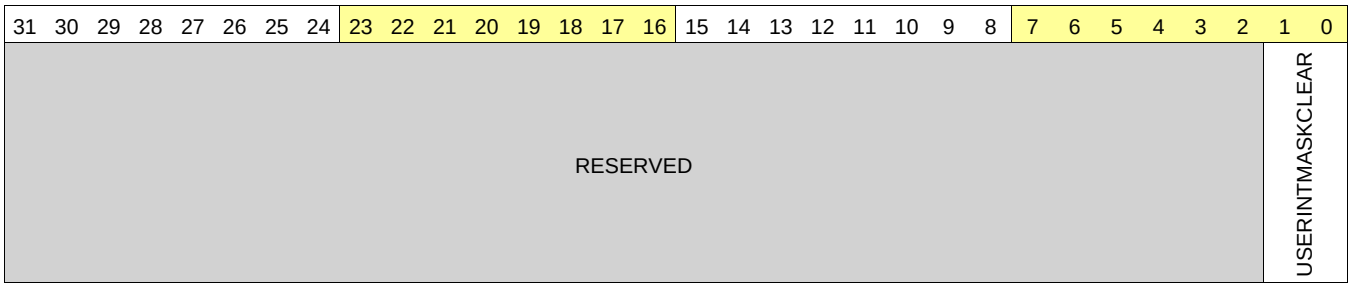
Table 18-1300. Register Call Summary for Register MDIO_USERINTMASKSET

Gigabit Ethernet Switch (GMAC_SW)

- Miscellaneous Pulse Interrupt (MISC_PULSE): [0]
- MDIO Functional Description: [1]
- Initializing the MDIO Module: [2]
- Writing Data To a PHY Register: [3]
- Reading Data From a PHY Register: [4]
- MDIO Register Summary: [5]

Table 18-1301. MDIO_USERINTMASKCLR

Address Offset	0x0000 002C		
Physical Address	0x4848 502C	**Instance**	MDIO
Description	MDIO User Command Complete Interrupt Mask Clear		
Type	RW		



Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1:0	USERINTMASKCLEAR	MDIO user command complete interrupt mask clear for USERINTMASKED[1:0], respectively. Writing a bit to 1 will disable further user command complete interrupts for that particular MDIO_USERACCESS register. Writing a 0 to this register has no effect.	RW	0x0

Table 18-1302. Register Call Summary for Register MDIO_USERINTMASKCLR

Gigabit Ethernet Switch (GMAC_SW)

- MDIO Functional Description: [0]
- MDIO Register Summary: [1]

Table 18-1303. MDIO_USERACCESS0

Address Offset	0x0000 0080		
Physical Address	0x4848 5080	**Instance**	MDIO
Description	MDIO_User_Access		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GO	WRITE	ACK	RESERVED					REGADR								PHYADR								DATA							

Bits	Field Name	Description	Type	Reset
31	GO	Go. Writing a 1 to this bit causes the MDIO state machine to perform an MDIO access when it is convenient for it to do so, this is not an instantaneous process. Writing a 0 to this bit has no effect. This bit is write able only if the MDIO state machine is enabled. This bit will self clear when the requested access has been completed. Any writes to the MDIO_USERACCESS0 register are blocked when the GO bit is 1. If byte access is being used, the GO bit should be written last.	RW	0x0
30	WRITE	Write enable. Setting this bit to a 1 causes the MDIO transaction to be a register write, otherwise it is a register read.	RW	0x0
29	ACK	Acknowledge. This bit is set if the PHY acknowledged the read transaction.	RW	0x0
28:26	RESERVED		R	0x0
25:21	REGADR	Register address. Specifies the PHY register to be accessed for this transaction.	RW	0x0
20:16	PHYADR	PHY address. Specifies the PHY to be accesses for this transaction.	RW	0x0
15:0	DATA	User data. The data value read from or to be written to the specified PHY register.	RW	0x0

Table 18-1304. Register Call Summary for Register MDIO_USERACCESS0

Gigabit Ethernet Switch (GMAC_SW)

- MDIO: [0]
- MDIO Functional Description: [1][2][3][4][5]
- Initializing the MDIO Module: [6][7][8]
- Writing Data To a PHY Register: [9][10][11][12][13]
- Reading Data From a PHY Register: [14][15][16][17][18]
- MDIO Register Summary: [19]
- MDIO Register Description: [20][21][22]

Table 18-1305. MDIO_USERPHYSEL0

Address Offset	0x0000 0084		
Physical Address	0x4848 5084	**Instance**	MDIO
Description	MDIO User PHY Select		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								LINKSEL	LINKINTENB	RESERVED	PHYADDRMON				

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x0
7	LINKSEL	Link status determination select. Set to 1 to determine link status using the MLINK pin (NOT PINNED OUT). Default value is 0 which implies that the link status is determined by the MDIO state machine.	RW	0x0
6	LINKINTENB	Link change interrupt enable. Set to 1 to enable link change status interrupts for PHY address specified in PHYADDRMON. Link change interrupts are disabled if this bit is set to 0. 0: Link change interrupts are disabled. 1: Link change status interrupts for PHY address specified in PHYADDRMON bits are enabled.	RW	0x0
5	RESERVED		R	0x0
4:0	PHYADDRMON	PHY address whose link status is to be monitored.	RW	0x0

Table 18-1306. Register Call Summary for Register MDIO_USERPHYSEL0

Gigabit Ethernet Switch (GMAC_SW)

- MDIO Functional Description: [0][1]
- Initializing the MDIO Module: [2]
- MDIO Register Summary: [3]
- MDIO Register Description: [4]

Table 18-1307. MDIO_USERACCESS1

Address Offset	0x0000 0088		
Physical Address	0x4848 5088	**Instance**	MDIO
Description	MDIO User Access		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
GO		WRITE		ACK		RESERVED		REGADR					PHYADR			DATA															

Bits	Field Name	Description	Type	Reset
31	GO	Go. Writing a 1 to this bit causes the MDIO state machine to perform an MDIO access when it is convenient for it to do so, this is not an instantaneous process. Writing a 0 to this bit has no effect. This bit is write able only if the MDIO state machine is enabled. This bit will self clear when the requested access has been completed. Any writes to the MDIO_USERACCESS1 register are blocked when the GO bit is 1. If byte access is being used, the GO bit should be written last.	RW	0x0
30	WRITE	Write enable. Setting this bit to a 1 causes the MDIO transaction to be a register write, otherwise it is a register read.	RW	0x0
29	ACK	Acknowledge. This bit is set if the PHY acknowledged the read transaction.	RW	0x0
28:26	RESERVED		R	0x0
25:21	REGADR	Register address. Specifies the PHY register to be accessed for this transaction.	RW	0x0
20:16	PHYADR	PHY address. Specifies the PHY to be accesses for this transaction.	RW	0x0
15:0	DATA	User data. The data value read from or to be written to the specified PHY register.	RW	0x0

Table 18-1308. Register Call Summary for Register MDIO_USERACCESS1

Gigabit Ethernet Switch (GMAC_SW)

- MDIO: [0]
- MDIO Register Summary: [1]
- MDIO Register Description: [2][3][4][5]

Table 18-1309. MDIO_USERPHYSEL1

Address Offset	0x0000 008C		
Physical Address	0x4848 508C	**Instance**	MDIO
Description	MDIO User PHY Select		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								LINKSEL	LINKINTENB	RESERVED	PHYADDRMON				

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x0
7	LINKSEL	Link status determination select. Set to 1 to determine link status using the MLINK pin (NOT PINNED OUT). Default value is 0 which implies that the link status is determined by the MDIO state machine.	RW	0x0
6	LINKINTENB	Link change interrupt enable. Set to 1 to enable link change status interrupts for PHY address specified in PHYADDRMON. Link change interrupts are disabled if this bit is set to 0. 0: Link change interrupts are disabled. 1: Link change status interrupts for PHY address specified in PHYADDRMON bits are enabled.	RW	0x0
5	RESERVED		R	0x0
4:0	PHYADDRMON	PHY address whose link status is to be monitored.	RW	0x0

Table 18-1310. Register Call Summary for Register MDIO_USERPHYSEL1

Gigabit Ethernet Switch (GMAC_SW)

- MDIO Register Summary: [0]
- MDIO Register Description: [1]

18.8.6.11 WR registers

18.8.6.11.1 WR Register Summary

Table 18-1311. WR Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	WR Physical Address
WR_IDVER	R	32	0x0000 0000	0x4848 5200
WR_SOFT_RESET	RW	32	0x0000 0004	0x4848 5204
WR_CONTROL	RW	32	0x0000 0008	0x4848 5208
WR_INT_CONTROL	RW	32	0x0000 000C	0x4848 520C
WR_C0_RX_THRESH_EN	RW	32	0x0000 0010	0x4848 5210
WR_C0_RX_EN	RW	32	0x0000 0014	0x4848 5214
WR_C0_TX_EN	RW	32	0x0000 0018	0x4848 5218
WR_C0_MISC_EN	RW	32	0x0000 001C	0x4848 521C
WR_C0_RX_THRESH_STAT	R	32	0x0000 0040	0x4848 5240
WR_C0_RX_STAT	R	32	0x0000 0044	0x4848 5244
WR_C0_TX_STAT	R	32	0x0000 0048	0x4848 5248
WR_C0_MISC_STAT	R	32	0x0000 004C	0x4848 524C
WR_C0_RX_IMAX	RW	32	0x0000 0070	0x4848 5270
WR_C0_TX_IMAX	RW	32	0x0000 0074	0x4848 5274
WR_RGMII_CTL	R	32	0x0000 0088	0x4848 5288
WR_STATUS	R	32	0x0000 008C	0x4848 528C

18.8.6.11.2 WR Register Description

Table 18-1312. WR_IDVER

Address Offset	0x0000 0000		
Physical Address	0x4848 5200	**Instance**	WR
Description	Subsystem wrapper revision register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REVISION																															

Bits	Field Name	Description	Type	Reset
31:0	REVISION	Wrapper revision value	R	0x-

Table 18-1313. Register Call Summary for Register WR_IDVER

Gigabit Ethernet Switch (GMAC_SW)

- WR Register Summary: [0]

Table 18-1314. WR_SOFT_RESET

Address Offset	0x0000 0004		
Physical Address	0x4848 5204	**Instance**	WR
Description	Subsystem soft reset register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															SOFT_RESET

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	SOFT_RESET	Software reset - Writing a one to this bit causes the CPGMACSS_R logic to be reset (INT, REGS, CPPI). Software reset occurs on the clock following the register bit write.	RW	0x0

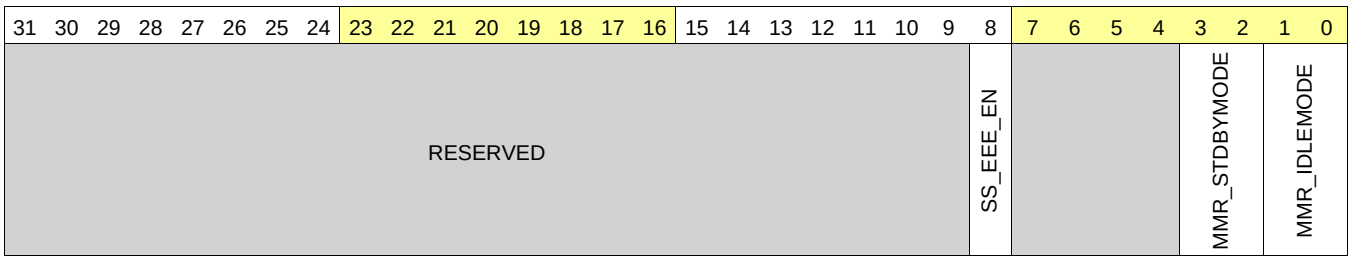
Table 18-1315. Register Call Summary for Register WR_SOFT_RESET

Gigabit Ethernet Switch (GMAC_SW)

- Software Reset: [0]
- WR Register Summary: [1]

Table 18-1316. WR_CONTROL

Address Offset	0x0000 0008		
Physical Address	0x4848 5208	**Instance**	WR
Description	Subsystem control register		
Type	RW		



Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reserved	R	0x0
8	SS_EEE_EN	Subsystem Energy Efficient Ethernet enable 0: EEE disabled 1: EEE enabled	RW	0x0
7:4	RESERVED	Reserved	R	0x0
3:2	MMR_STDBYMODE	Configuration of the local initiator state management mode. By definition, initiator may generate read/write transaction as long as it is out of STANDBY state. 0x0: Force-standby mode : Local initiator is unconditionally placed in standbystate. 0x1: No-standby mode : Local initiator is unconditionally placed out of standby state. 0x3: Reserved : Reserved. 0x2: Reserved : Reserved.	RW	0x0
1:0	MMR_IDLEMODE	Configuration of the local initiator state management mode. By definition, initiator may generate read/write transaction as long as it is out of IDLE state. 0x0: Force-idle mode : Local initiator is unconditionally placed in idle state. 0x1: No-idle mode : Local initiator is unconditionally placed out of idle state. 0x3: Reserved : Reserved. 0x2: Reserved : Reserved.	RW	0x0

Table 18-1317. Register Call Summary for Register WR_CONTROL

Gigabit Ethernet Switch (GMAC_SW)

- Energy Efficient Ethernet Support (802.3az): [0]
- WR Register Summary: [1]

Table 18-1318. WR_INT_CONTROL

Address Offset	0x0000 000C		
Physical Address	0x4848 520C	**Instance**	WR
Description	Subsystem interrupt control		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INT_TEST	RESERVED															INT_PACE_EN								RESERVED								INT_PRESCALE							

Bits	Field Name	Description	Type	Reset
31	INT_TEST	Interrupt Test - Test bit to the interrupt pacing blocks	RW	0x0
30:22	RESERVED		R	0x0
21:16	INT_PACE_EN	Interrupt Pacing Enable INT_PACE_EN[0] – Enables RX_PULSE Pacing (0 is pacing bypass) INT_PACE_EN[1] – Enables TX_PULSE Pacing (0 is pacing bypass)	RW	0x0
15:12	RESERVED		R	0x0
11:0	INT_PRESCALE	Interrupt Counter Prescaler - The number of MAIN_CLK periods in 4us.	R	0x0

Table 18-1319. Register Call Summary for Register WR_INT_CONTROL

Gigabit Ethernet Switch (GMAC_SW)

- Interrupt Pacing: [0]
- WR Register Summary: [1]

Table 18-1320. WR_C0_RX_THRESH_EN

Address Offset	0x0000 0010		
Physical Address	0x4848 5210	**Instance**	WR
Description	Subsystem core 0 receive threshold int enable register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								C0_RX_THRESH_EN							

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x0
7:0	C0_RX_THRESH_EN	Core 0 Receive Threshold Enable - Each bit in this register corresponds to the bit in the receive threshold interrupt that is enabled to generate an interrupt on RX_THRESH_PULSE.	RW	0x0

Table 18-1321. Register Call Summary for Register WR_C0_RX_THRESH_EN

Gigabit Ethernet Switch (GMAC_SW)

- Receive Threshold Pulse Interrupt (RX_THRESH_PULSE): [0][1]
- WR Register Summary: [2]

Table 18-1322. WR_C0_RX_EN

Address Offset	0x0000 0014		
Physical Address	0x4848 5214	**Instance**	WR
Description	Subsystem core 0 receive interrupt enable register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								C0_RX_EN							

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x0
7:0	C0_RX_EN	Core 0 Receive Enable - Each bit in this register corresponds to the bit in the rx interrupt that is enabled to generate an interrupt on RX_PULSE.	RW	0x0

Table 18-1323. Register Call Summary for Register WR_C0_RX_EN

Gigabit Ethernet Switch (GMAC_SW)

- Receive Packet Completion Pulse Interrupt (RX_PULSE): [0][1]
- WR Register Summary: [2]

Table 18-1324. WR_C0_TX_EN

Address Offset	0x0000 0018		
Physical Address	0x4848 5218	**Instance**	WR
Description	Subsystem core 0 transmit interrupt enable register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								C0_TX_EN							

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x0
7:0	C0_TX_EN	Core 0 Transmit Enable - Each bit in this register corresponds to the bit in the tx interrupt that is enabled to generate an interrupt on TX_PULSE.	RW	0x0

Table 18-1325. Register Call Summary for Register WR_C0_TX_EN

Gigabit Ethernet Switch (GMAC_SW)

- Transmit Packet Completion Pulse Interrupt (TX_PULSE): [0][1]
- WR Register Summary: [2]

Table 18-1326. WR_C0_MISC_EN

Address Offset	0x0000 001C		
Physical Address	0x4848 521C	**Instance**	WR
Description	Subsystem core 0 misc interrupt enable register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								C0_MISC_EN							

Bits	Field Name	Description	Type	Reset
31:5	RESERVED		R	0x0
4:0	C0_MISC_EN	Core 0 Misc Enable - Each bit in this register corresponds to the miscellaneous interrupt (SPF2_PEND, SPF1_PEND, EVNT_PEND, STAT_PEND, HOST_PEND, MDIO_LINKINT, MDIO_USERINT) that is enabled to generate an interrupt on MISC_PULSE.	RW	0x0

Table 18-1327. Register Call Summary for Register WR_C0_MISC_EN

Gigabit Ethernet Switch (GMAC_SW)

- Miscellaneous Pulse Interrupt (MISC_PULSE): [0]
- WR Register Summary: [1]

Table 18-1328. WR_C0_RX_THRESH_STAT

Address Offset	0x0000 0040		
Physical Address	0x4848 5240	**Instance**	WR
Description	Subsystem core 0 rx threshold masked int status register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								C0_RX_THRESH_STAT							

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x0
7:0	C0_RX_THRESH_STAT	Core 0 Receive Threshold Masked Interrupt Status - Each bit in this read only register corresponds to the bit in the receive threshold interrupt that is enabled and generating an interrupt on RX_THRESH_PULSE.	R	0x0

Table 18-1329. Register Call Summary for Register WR_C0_RX_THRESH_STAT

Gigabit Ethernet Switch (GMAC_SW)

- Receive Threshold Pulse Interrupt (RX_THRESH_PULSE): [0]
- WR Register Summary: [1]

Table 18-1330. WR_C0_RX_STAT

Address Offset	0x0000 0044		
Physical Address	0x4848 5244	**Instance**	WR
Description	Subsystem core 0 rx interrupt masked int status register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								C0_RX_STAT							

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x0
7:0	C0_RX_STAT	Core 0 Receive Masked Interrupt Status - Each bit in this read only register corresponds to the bit in the Rx interrupt that is enabled and generating an interrupt on RX_PULSE.	R	0x0

Table 18-1331. Register Call Summary for Register WR_C0_RX_STAT

Gigabit Ethernet Switch (GMAC_SW)

- Receive Packet Completion Pulse Interrupt (RX_PULSE): [0][1]
- WR Register Summary: [2]

Table 18-1332. WR_C0_TX_STAT

Address Offset	0x0000 0048		
Physical Address	0x4848 5248	**Instance**	WR
Description	Subsystem core 0 tx interrupt masked int status register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								C0_TX_STAT							

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x0
7:0	C0_TX_STAT	Core 0 Transmit Masked Interrupt Status - Each bit in this read only register corresponds to the bit in the Tx interrupt that is enabled and generating an interrupt on TX_PULSE .	R	0x0

Table 18-1333. Register Call Summary for Register WR_C0_TX_STAT

Gigabit Ethernet Switch (GMAC_SW)

- Transmit Packet Completion Pulse Interrupt (TX_PULSE): [0]
- WR Register Summary: [1]

Table 18-1334. WR_C0_MISC_STAT

Address Offset	0x0000 004C		
Physical Address	0x4848 524C	**Instance**	WR
Description	Subsystem core 0 misc interrupt masked int status register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								C0_MISC_STAT							

Bits	Field Name	Description	Type	Reset
31:5	RESERVED		R	0x0
4:0	C0_MISC_STAT	Core 0 Misc Masked Interrupt Status - Each bit in this register corresponds to the miscellaneous interrupt (SPF2_PEND, SPF1_PEND, EVNT_PEND, STAT_PEND, HOST_PEND, MDIO_LINKINT, MDIO_USERINT) that is enabled and generating an interrupt on MISC_PULSE .	R	0x0

Table 18-1335. Register Call Summary for Register WR_C0_MISC_STAT

Gigabit Ethernet Switch (GMAC_SW)

- Miscellaneous Pulse Interrupt (MISC_PULSE): [0][1]
- WR Register Summary: [2]

Table 18-1336. WR_C0_RX_IMAX

Address Offset	0x0000 0070		
Physical Address	0x4848 5270	**Instance**	WR
Description	Subsystem core 0 receive interrupts per millisecond		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																													C0_RX_IMAX		

Bits	Field Name	Description	Type	Reset
31:6	RESERVED		R	0x0
5:0	C0_RX_IMAX	Core 0 Receive Interrupts per Millisecond - The maximum number of interrupts per millisecond generated on RX_PULSE if pacing is enabled for this interrupt.	RW	0x0

Table 18-1337. Register Call Summary for Register WR_C0_RX_IMAX

Gigabit Ethernet Switch (GMAC_SW)

- Interrupt Pacing: [0]
- WR Register Summary: [1]

Table 18-1338. WR_C0_TX_IMAX

Address Offset	0x0000 0074		
Physical Address	0x4848 5274	**Instance**	WR
Description	Subsystem core 0 transmit interrupts per millisecond		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								C0_TX_IMAX							

Bits	Field Name	Description	Type	Reset
31:6	RESERVED		R	0x0
5:0	C0_TX_IMAX	Core 0 Transmit Interrupts per Millisecond - The maximum number of interrupts per millisecond generated on TX_PULSE if pacing is enabled for this interrupt.	RW	0x0

Table 18-1339. Register Call Summary for Register WR_C0_TX_IMAX

Gigabit Ethernet Switch (GMAC_SW)

- Interrupt Pacing: [0]
- WR Register Summary: [1]

Table 18-1340. WR_RGMII_CTL

Address Offset	0x0000 0088		
Physical Address	0x4848 5288	**Instance**	WR
Description	RGMII control signal register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								RGMII2_FULLDUPLEX	RGMII2_SPEED		RGMII2_LINK	RGMII1_FULLDUPLEX		RGMII1_SPEED	RGMII1_LINK

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x0
7	RGMII2_FULLDUPLEX	RGMII 2 Fullduplex - This is the CPRGMII fullduplex output signal. 0 - Half-duplex mode 1 - Full-duplex mode	R	0x0
6:5	RGMII2_SPEED	RGMII2 Speed - This is the CPRGMII speed output signal 0x0 - 10Mbps mode 0x1 - 100Mbps mode 0x2 - 1000Mbps (gig) mode 0x3 - reserved	R	0x0
4	RGMII2_LINK	RGMII2 Link Indicator - This is the CPRGMII link output signal 0 - RGMII2 link is down 1 - RGMII2 link is up	R	0x0

Bits	Field Name	Description	Type	Reset
3	RGMII1_FULLDUPLEX	RGMII1 Fullduplex - This is the CPRGMII fullduplex output signal. 0 - Half-duplex mode 1 - Full-duplex mode	R	0x0
2:1	RGMII1_SPEED	RGMII1 Speed - This is the CPRGMII speed output signal 0x0 - 10Mbps mode 0x1 - 100Mbps mode 0x2 - 1000Mbps (gig) mode 0x3 - reserved	R	0x0
0	RGMII1_LINK	RGMII1 Link Indicator - This is the CPRGMII link output signal 0 - RGMII1 link is down 1 - RGMII1 link is up	R	0x0

Table 18-1341. Register Call Summary for Register WR_RGMII_CTL

Gigabit Ethernet Switch (GMAC_SW)

- WR Register Summary: [0]

Table 18-1342. WR_STATUS

Address Offset	0x0000 008C		
Physical Address	0x4848 528C	**Instance**	WR
Description	Subsystem Status register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																													SPF2_CLKSTOP_ACK	SPF1_CLKSTOP_ACK	EEE_CLKSTOP_ACK

Bits	Field Name	Description	Type	Reset
31:3	RESERVED		R	0x0
2	SPF2_CLKSTOP_ACK	SPF2 Clockstop Acknowledge – When asserted the subsystem gated clock is not turned on due to SPF2.	R	0x0
1	SPF1_CLKSTOP_ACK	SPF1 Clockstop Acknowledge – When asserted the subsystem gated clock is not turned on due to SPF1.	R	0x0
0	EEE_CLKSTOP_ACK	CPSW_3G Clockstop Acknowledge – When asserted the subsystem gated clock is not turned on due to the CPSW_3G.	R	0x0

Table 18-1343. Register Call Summary for Register WR_STATUS

Gigabit Ethernet Switch (GMAC_SW)

- WR Register Summary: [0]

18.8.6.12 SPF Registers

18.8.6.12.1 SPF Register Summary

Table 18-1344. SPF Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	SPF1 Physical Address	SPF2 Physical Address
SPF_IDVER	R	32	0x0000 0000	0x4848 5C00	0x4848 5E00
SPF_STATUS	RW	32	0x0000 0004	0x4848 5C04	0x4848 5E04
SPF_CONTROL	RW	32	0x0000 0008	0x4848 5C08	0x4848 5E08
SPF_DROPCOUNT	R	32	0x0000 000C	0x4848 5C0C	0x4848 5E0C
SPF_SWRESET	RW	32	0x0000 0010	0x4848 5C10	0x4848 5E10
SPF_PRESCALE	RW	32	0x0000 0014	0x4848 5C14	0x4848 5E14
SPF_RATELIMi[1]	RW	32	0x0000 0018 + (i * 4)	0x4848 5C18 + (i * 4)	0x4848 5E18 + (i * 4)
SPF_CONSTj[2]	RW	32	0x0000 001C + (j * 4)	0x4848 5C1C + (j * 4)	0x4848 5E1C + (j * 4)
SPF_INSTRW2	RW	32	0x0000 0050	0x4848 5C50	0x4848 5E50
SPF_INSTRW1	RW	32	0x0000 0054	0x4848 5C54	0x4848 5E54
SPF_INSTRW0	RW	32	0x0000 0058	0x4848 5C58	0x4848 5E58
SPF_INSTR_CTL	RW	32	0x0000 005C	0x4848 5C5C	0x4848 5E5C
SPF_LOG_BEGIN	RW	32	0x0000 0060	0x4848 5C60	0x4848 5E60
SPF_LOG_END	RW	32	0x0000 0064	0x4848 5C64	0x4848 5E64
SPF_LOG_HWPTR	R	32	0x0000 0068	0x4848 5C68	0x4848 5E68
SPF_LOG_SWPTR	RW	32	0x0000 006C	0x4848 5C6C	0x4848 5E6C
SPF_LOG_MAP0	RW	32	0x0000 0070	0x4848 5C70	0x4848 5E70
SPF_LOG_MAP1	RW	32	0x0000 0074	0x4848 5C74	0x4848 5E74
SPF_LOG_THRESHk[3]	RW	32	0x0000 0078 + (k * 4)	0x4848 5C78 + (k * 4)	0x4848 5E78 + (k * 4)
SPF_INTCNT	RW	32	0x0000 009C	0x4848 5C9C	0x4848 5E9C
SPF_INT_RAW	RW	32	0x0000 00A0	0x4848 5CA0	0x4848 5EA0
SPF_INT_MASKED	RW	32	0x0000 00A4	0x4848 5CA4	0x4848 5EA4
SPF_MASK_SET	RW	32	0x0000 00A8	0x4848 5CA8	0x4848 5EA8
SPF_MASK_CLR	RW	32	0x0000 00AC	0x4848 5CAC	0x4848 5EAC

(1) i = 0 to 3
(2) j = 0 to 7
(3) k = 0 to 8

18.8.6.12.2 SPF Register Description

Table 18-1345. SPF_IDVER

Address Offset	0x0000 0000		
Physical Address	0x4848 5C00 0x4848 5E00	**Instance**	SPF1 SPF2
Description	SPF revision register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REVISION																															

Bits	Field Name	Description	Type	Reset
31:0	REVISION	SPF revision value	R	0x-

Table 18-1346. Register Call Summary for Register SPF_IDVER

Gigabit Ethernet Switch (GMAC_SW)
- SPF Register Summary: [0]

Table 18-1347. SPF_STATUS

Address Offset	0x0000 0004		
Physical Address	0x4848 5C04 0x4848 5E04	**Instance**	SPF1 SPF2
Description	Status register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															SPF_BUSY

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	SPF_BUSY	SPF is Busy/Idle, Busy Packet processing or logging in progress.	RW	0x0

Table 18-1348. Register Call Summary for Register SPF_STATUS

Gigabit Ethernet Switch (GMAC_SW)
- SPF Register Summary: [0]

Table 18-1349. SPF_CONTROL

Address Offset	0x0000 0008		
Physical Address	0x4848 5C08 0x4848 5E08	**Instance**	SPF1 SPF2
Description	SPF control register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						SPF_LOGOW_EN	SPF_LOG_EN	RESERVED				SPF_RULE_LOG	SPF_EXT_BYPASS	SPF_DROP	SPF_ENABLE

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9	SPF_LOGOW_EN	SPF Log Overwrite Enable. Setting this bit will cause SPF to overwrite previously logged data whether or not software has updated the software_working_pointer. Overwriting only occurs if there is new data but no space to write it in the space indicated by log_start_address and log_end_address.	RW	0x0
8	SPF_LOG_EN	SPF Log Enable. Setting this bit will allow SPF to log information about dropped packets to memory.	RW	0x0
7:4	RESERVED		R	0x0
3	SPF_RULE_LOG	SPF Rule Engine Log Enable. Setting this bit will allow SPF to log data from rule engine. The default is log data from extractor.	RW	0x0
2	SPF_EXT_BYPASS	SPF Extractor Bypass Enable. The extractor will not provide any offset information to rule engine if this bit is set. The rule engine must load each of the base registers it intends to use to determine if the packet should be discarded.	RW	0x0
1	SPF_DROP	SPF Drop Enable. This bit must be set to activate packet drops.	RW	0x0
0	SPF_ENABLE	SPF Enable. This bit must be set to enable any operation in SPF. The SPF instruction memory can only be accessed by host processor when the spf_enable is deasserted. Once spf_enable is set, writing a zero to this bit will only take effect when spf_busy signal is low. This ensures that spf stops only on packet boundaries.	RW	0x0

Table 18-1350. Register Call Summary for Register SPF_CONTROL

Gigabit Ethernet Switch (GMAC_SW)

- SPF Functional Description: [0][1][2][3]
- Programming Guide: [4]
- SPF Register Summary: [5]

Table 18-1351. SPF_DROPCOUNT

Address Offset	0x0000 000C		
Physical Address	0x4848 5C0C 0x4848 5E0C	**Instance**	SPF1 SPF2
Description	Drop Count Register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								SPF_DROPCNT																							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x0
23:0	SPF_DROPCNT	SPF Drop counter indicates the number of packets dropped so far. This counter does not roll over and must be cleared by writing 0x00FFFFFF.	R	0x0

Table 18-1352. Register Call Summary for Register SPF_DROPCOUNT

Gigabit Ethernet Switch (GMAC_SW)

- SPF Register Summary: [0]

Table 18-1353. SPF_SWRESET

Address Offset	0x0000 0010		
Physical Address	0x4848 5C10 0x4848 5E10	**Instance**	SPF1 SPF2
Description	Software Reset Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															SPF_SWRST

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	SPF_SWRST	SPF Software reset bit can be set to initiate a software reset. It stays high until the reset has not completed, this reset clears all registers to default value.	RW	0x0

Table 18-1354. Register Call Summary for Register SPF_SWRESET

Gigabit Ethernet Switch (GMAC_SW)

- SPF Register Summary: [0]

Table 18-1355. SPF_PRESCALE

Address Offset	0x0000 0014		
Physical Address	0x4848 5C14 0x4848 5E14	**Instance**	SPF1 SPF2
Description	Rate Limit Prescale Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								SPF_PRESCALE																							

Bits	Field Name	Description	Type	Reset
31:20	RESERVED		R	0x0
19:0	SPF_PRESCALE	The MAIN clock is divided by this value for use in Rate Limiters. It is used to create rolling time intervals for use in rate limiting feature.	RW	0x0

Table 18-1356. Register Call Summary for Register SPF_PRESCALE

Gigabit Ethernet Switch (GMAC_SW)

- SPF Functional Description: [0][1]
- Programming Guide: [2]
- SPF Register Summary: [3]
- SPF Register Description: [4][5]

Table 18-1357. SPF_RATELIMi

Address Offset	0x0000 0018 + (i * 4)	**Index**	i = 0 to 3
Physical Address	0x4848 5C18 + (i * 4) 0x4848 5E18 + (i * 4)	**Instance**	SPF1 SPF2
Description	Rate Limit Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								SPF_RATELIM							

Bits	Field Name	Description	Type	Reset
31:8	RESERVED		R	0x0
7:0	SPF_RATELIM	SPF Rate Limit Register. The number of packets corresponding to a filter that will be allowed per unit time interval. The filters are programmed in the rule engine and time interval is determined by the SPF_PRESCALE register.	RW	0x0

Table 18-1358. Register Call Summary for Register SPF_RATELIMi

Gigabit Ethernet Switch (GMAC_SW)

- SPF Functional Description: [0][1]
- Programming Guide: [2]
- SPF Register Summary: [3]

Table 18-1359. SPF_CONSTj

Address Offset	0x0000 0028 + (j * 4)	**Index**	j = 0 to 7
Physical Address	0x4848 5C1C + (j * 4) 0x4848 5E1C + (j * 4)	**Instance**	SPF1 SPF2
Description	Constant Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPF_CONST																															

Bits	Field Name	Description	Type	Reset
31:0	SPF_CONST	SPF Constant Register. The contents of this register are used as input to any instruction that references it.	RW	0x0

Table 18-1360. Register Call Summary for Register SPF_CONSTj

Gigabit Ethernet Switch (GMAC_SW)

- SPF Functional Description: [0]
- Programming Guide: [1]
- SPF Register Summary: [2]

Table 18-1361. SPF_INSTRW2

Address Offset	0x0000 0050		
Physical Address	0x4848 5C50 0x4848 5E50	**Instance**	SPF1 SPF2
Description	Instruction Word 2 Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																		SPF_INSTR_W2													

Bits	Field Name	Description	Type	Reset
31:14	RESERVED		R	0x0
13:0	SPF_INSTR_W2	SPF Rule Engine Instruction Word [75:64] is read from or written to this field.	RW	0x0

Table 18-1362. Register Call Summary for Register SPF_INSTRW2

Gigabit Ethernet Switch (GMAC_SW)

- SPF Register Summary: [0]

Table 18-1363. SPF_INSTRW1

Address Offset	0x0000 0054		
Physical Address	0x4848 5C54 0x4848 5E54	**Instance**	SPF1 SPF2
Description	Instruction Word 1 Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPF_INSTR_W1																															

Bits	Field Name	Description	Type	Reset
31:0	SPF_INSTR_W1	SPF Rule Engine Instruction Word [63:32] is read from or written to this field.	RW	0x0

Table 18-1364. Register Call Summary for Register SPF_INSTRW1

Gigabit Ethernet Switch (GMAC_SW)

- SPF Register Summary: [0]

Table 18-1365. SPF_INSTRW0

Address Offset	0x0000 0058		
Physical Address	0x4848 5C58 0x4848 5E58	**Instance**	SPF1 SPF2
Description	Instruction Word 0 Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPF_INSTR_W0																															

Bits	Field Name	Description	Type	Reset
31:0	SPF_INSTR_W0	SPF Rule Engine Instruction Word [31:0] is read from or written to this field.	RW	0x0

Table 18-1366. Register Call Summary for Register SPF_INSTRW0

Gigabit Ethernet Switch (GMAC_SW)

- SPF Register Summary: [0]

Table 18-1367. SPF_INSTR_CTL

Address Offset	0x0000 005C		
Physical Address	0x4848 5C5C 0x4848 5E5C	**Instance**	SPF1 SPF2
Description	Instruction Control Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPF_INSTR_WEN	SPF_INSTR_REN	RESERVED																						SPF_INSTR_PTR							

Bits	Field Name	Description	Type	Reset
31	SPF_INSTR_WEN	SPF Write enable bit specifies whether a write operation is to be performed. To read or write instructions, spf processing must be stopped. When the rule engine is processing instructions, the instruction memory cannot be accessed. This bit is set to perform a write and the data in the SPF_INSTR_W2, SPF_INSTR_W1 and SPF_INSTR_W0 registers is written to the instruction RAM at address specified in the SPF_INSTR_PTR field. This bit is always read as zero.	W	0x0
30	SPF_INSTR_REN	SPF Read enable bit specifies whether a read operation is to be performed. This bit is set to perform a read and read data is available in the SPF_INSTR_W2, SPF_INSTR_W1 and SPF_INSTR_W0 registers once read operation has completed. This bit is always read as zero.	W	0x0
29:6	RESERVED		R	0x0
5:0	SPF_INSTR_PTR	The address in the instruction memory that is to be accessed.	RW	0x0

Table 18-1368. Register Call Summary for Register SPF_INSTR_CTL

Gigabit Ethernet Switch (GMAC_SW)

- Programming Guide: [0]
- SPF Register Summary: [1]

Table 18-1369. SPF_LOG_BEGIN

Address Offset	0x0000 0060		
Physical Address	0x4848 5C60 0x4848 5E60	**Instance**	SPF1 SPF2
Description	Log Begin Address Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPF_LOG_BEGIN																															

Bits	Field Name	Description	Type	Reset
31:0	SPF_LOG_BEGIN	SPF starts to write log data to memory starting from address given in this field.	RW	0x0

Table 18-1370. Register Call Summary for Register SPF_LOG_BEGIN

Gigabit Ethernet Switch (GMAC_SW)

- SPF Functional Description: [0][1]
- Programming Guide: [2]
- SPF Register Summary: [3]
- SPF Register Description: [4][5][6]

Table 18-1371. SPF_LOG_END

Address Offset	0x0000 0064		
Physical Address	0x4848 5C64 0x4848 5E64	**Instance**	SPF1 SPF2
Description	Log End Address Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPF_LOG_END																															

Bits	Field Name	Description	Type	Reset
31:0	SPF_LOG_END	This register along with SPF_LOG_BEGIN register defines the memory range for writing log data, the range(SPF_LOG_END SPF_LOG_BEGIN) should be multiple of 4 words(32 bits), as this is a look ahead register therefore the value progammed should be next word address. (i.e. last word address + 4).	RW	0x00001000

Table 18-1372. Register Call Summary for Register SPF_LOG_END

Gigabit Ethernet Switch (GMAC_SW)

- SPF Functional Description: [0][1][2][3]
- Programming Guide: [4]
- SPF Register Summary: [5]
- SPF Register Description: [6][7][8]

Table 18-1373. SPF_LOG_HWPTR

Address Offset	0x0000 0068		
Physical Address	0x4848 5C68 0x4848 5E68	**Instance**	SPF1 SPF2
Description	Log Hardware Pointer Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPF_LOG_HWPTR																															

Bits	Field Name	Description	Type	Reset
31:0	SPF_LOG_HWPTR	This register indicated the address of next location in memory that the SPF will log information to.	RW	0x0

Table 18-1374. Register Call Summary for Register SPF_LOG_HWPTR

Gigabit Ethernet Switch (GMAC_SW)

- SPF Functional Description: [0]
- SPF Register Summary: [1]
- SPF Register Description: [2]

Table 18-1375. SPF_LOG_SWPTR

Address Offset	0x0000 006C		
Physical Address	0x4848 5C6C 0x4848 5E6C	**Instance**	SPF1 SPF2
Description	Log Software Pointer Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPF_LOG_SWPTR																															

Bits	Field Name	Description	Type	Reset
31:0	SPF_LOG_SWPTR	This register specifies the address where software shall do next read, software must inform SPF about memory roll over by writing SPF_LOG_END into this register.	RW	0x0

Table 18-1376. Register Call Summary for Register SPF_LOG_SWPTR

Gigabit Ethernet Switch (GMAC_SW)

- SPF Functional Description: [0][1][2]
- SPF Register Summary: [3]
- SPF Register Description: [4]

Table 18-1377. SPF_LOG_MAP0

Address Offset	0x0000 0070		
Physical Address	0x4848 5C70 0x4848 5E70	**Instance**	SPF1 SPF2
Description	Filter Code Map Register 0		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPF_LOGMAP3								SPF_LOGMAP2								SPF_LOGMAP1								SPF_LOGMAP0							

Bits	Field Name	Description	Type	Reset
31:24	SPF_LOGMAP3	Mapping of drop code 3 to log threshold 3	RW	0x0
23:16	SPF_LOGMAP2	Mapping of drop code 2 to log threshold 2	RW	0x0
15:8	SPF_LOGMAP1	Mapping of drop code 1 to log threshold 1	RW	0x0
7:0	SPF_LOGMAP0	Mapping of drop code 0 to log threshold 0	RW	0x0

Table 18-1378. Register Call Summary for Register SPF_LOG_MAP0

Gigabit Ethernet Switch (GMAC_SW)

- Programming Guide: [0]
- SPF Register Summary: [1]

Table 18-1379. SPF_LOG_MAP1

Address Offset	0x0000 0074		
Physical Address	0x4848 5C74 0x4848 5E74	**Instance**	SPF1 SPF2
Description	Filter Code Map Register 1		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPF_LOGMAP7								SPF_LOGMAP6								SPF_LOGMAP5								SPF_LOGMAP4							

Bits	Field Name	Description	Type	Reset
31:24	SPF_LOGMAP7	Mapping of drop code 7 to log threshold 7	RW	0x0
23:16	SPF_LOGMAP6	Mapping of drop code 6 to log threshold 6	RW	0x0
15:8	SPF_LOGMAP5	Mapping of drop code 5 to log threshold 5	RW	0x0
7:0	SPF_LOGMAP4	Mapping of drop code 4 to log threshold 4	RW	0x0

Table 18-1380. Register Call Summary for Register SPF_LOG_MAP1

Gigabit Ethernet Switch (GMAC_SW)

- Programming Guide: [0]
- SPF Register Summary: [1]

Table 18-1381. SPF_LOG_THRESHk

Address Offset	0x0000 0078 + (k * 4)	**Index**	k = 0 to 8
Physical Address	0x4848 5C78 + (k * 4) 0x4848 5E78 + (k * 4)	**Instance**	SPF1 SPF2
Description	Log Threshold and Count Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SPF_COUNT																SPF_THRESH															

Bits	Field Name	Description	Type	Reset
31:16	SPF_COUNT	Number of packets dropped for drop code k (8 is default)	R	0x0
15:0	SPF_THRESH	Number of packets to be dropped before logging starts	RW	0xA

Table 18-1382. Register Call Summary for Register SPF_LOG_THRESHk

Gigabit Ethernet Switch (GMAC_SW)

- Programming Guide: [0]
- SPF Register Summary: [1]

Table 18-1383. SPF_INTCNT

Address Offset	0x0000 009C		
Physical Address	0x4848 5C9C 0x4848 5E9C	**Instance**	SPF1 SPF2
Description	Interrupt Frequency Control Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								SPF_INTCNT							

Bits	Field Name	Description	Type	Reset
31:5	RESERVED		R	0x0
4:0	SPF_INTCNT	Number of time thresholds must be met before a drop interrupt is triggered.	RW	0x0

Table 18-1384. Register Call Summary for Register SPF_INTCNT

Gigabit Ethernet Switch (GMAC_SW)

- SPF Functional Description: [0][1]
- Programming Guide: [2]
- SPF Register Summary: [3]
- SPF Register Description: [4]

Table 18-1385. SPF_INT_RAW

Address Offset	0x0000 00A0		
Physical Address	0x4848 5CA0 0x4848 5EA0	**Instance**	SPF1 SPF2
Description	Raw Interrupt Status register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															SPF_INT_RAW

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	SPF_INT_RAW	Status of Raw interrupt signal	RW	0x0

Table 18-1386. Register Call Summary for Register SPF_INT_RAW

Gigabit Ethernet Switch (GMAC_SW)

- SPF Functional Description: [0]
- SPF Register Summary: [1]
- SPF Register Description: [2]

Table 18-1387. SPF_INT_MASKED

Address Offset	0x0000 00A4		
Physical Address	0x4848 5CA4 0x4848 5EA4	**Instance**	SPF1 SPF2
Description	Interrupt Status register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															SPF_INT_MASKED

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	SPF_INT_MASKED	Status of interrupt signal with mask	RW	0x0

Table 18-1388. Register Call Summary for Register SPF_INT_MASKED

Gigabit Ethernet Switch (GMAC_SW)

- SPF Functional Description: [0]
- SPF Register Summary: [1]
- SPF Register Description: [2]

Table 18-1389. SPF_MASK_SET

Address Offset	0x0000 00A8		
Physical Address	0x4848 5CA8 0x4848 5EA8	**Instance**	SPF1 SPF2
Description	Interrupt Mask Set Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															SPF_MASKSET

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	SPF_MASKSET	Write a 1 to this bit to enable the interrupt.	RW	0x0

Table 18-1390. Register Call Summary for Register SPF_MASK_SET

Gigabit Ethernet Switch (GMAC_SW)

- SPF Functional Description: [0]
- Programming Guide: [1]
- SPF Register Summary: [2]

Table 18-1391. SPF_MASK_CLR

Address Offset	0x0000 00AC		
Physical Address	0x4848 5CAC 0x4848 5EAC	**Instance**	SPF1 SPF2
Description	Interrupt Mask Clear Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															SPF_MASKCLR

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0
0	SPF_MASKCLR	Write a 1 to this bit to disable the interrupt.	RW	0x0

Table 18-1392. Register Call Summary for Register SPF_MASK_CLR

Gigabit Ethernet Switch (GMAC_SW)

- SPF Functional Description: [0]
- SPF Register Summary: [1]

SDIO Controller

This chapter describes the features and functions of the SDIO interface of the device.

Topic **Page**

19.1 SDIO Overview

The SDIO host controller provides an interface between a local host (LH) such as a microprocessor unit or digital signal processor and SDIO cards. It handles SDIO transactions with minimal LH intervention.

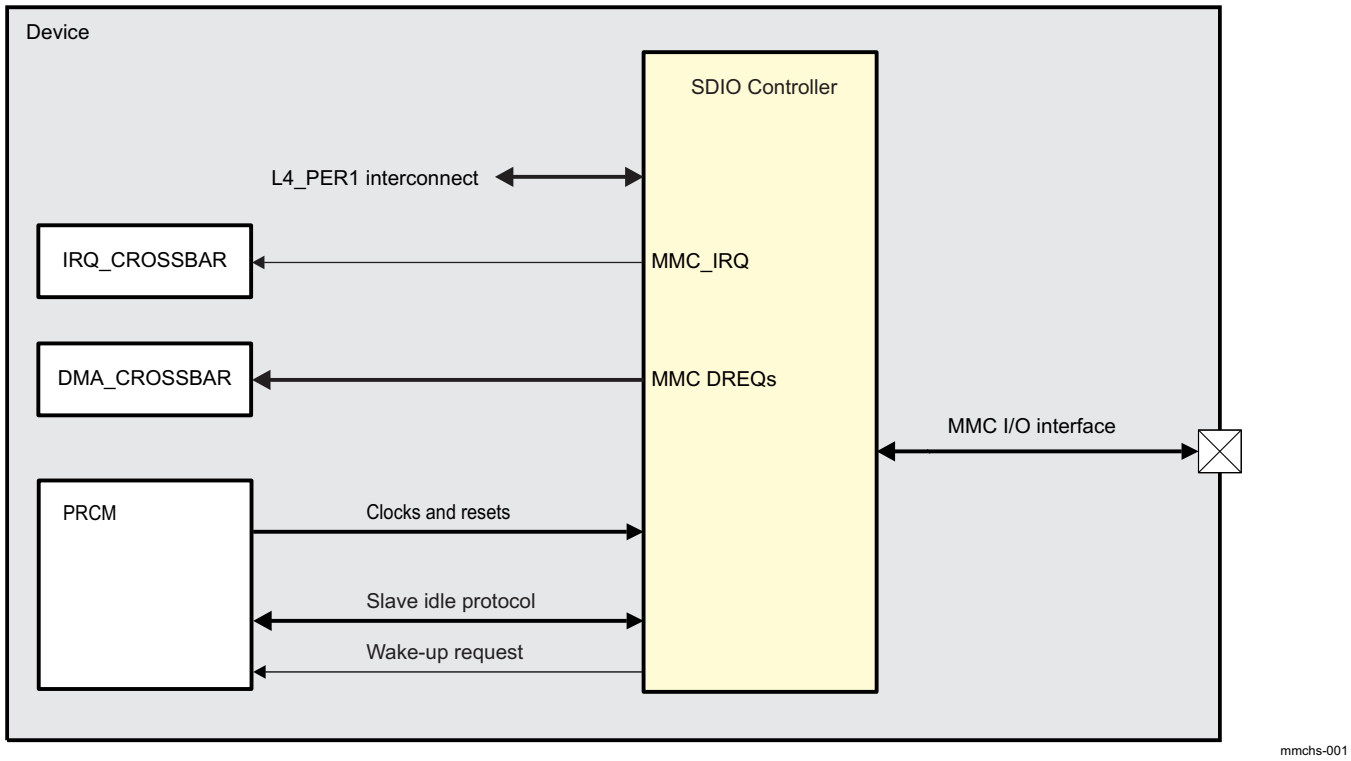
The SDIO host controller deals with SDIO protocol at transmission level, data packing, adding cyclic redundancy checks (CRCs), start/end bit, and checking for syntactical correctness.

The application interface can send every SDIO command and poll for the status of the adapter or wait for an interrupt request, which is sent back in case of exceptions or to warn of end of operation.

The application interface can read card responses or flag registers. It can also mask individual interrupt sources. All these operations can be performed by reading and writing control registers. The SDIO host controller also supports two slave DMA channels.

The SDIO controller is also referred to as MMC. Figure 19-1 shows an overview of the SDIO controller.

Figure 19-1. SDIO Overview



19.1.1 SDIO Features

This section describes the features supplied by the SDIO controller.

Compliance with standards:

- SDIO command/response sets and interrupt/read-wait suspend-resume operations as defined in the SD part E1 specification v3.00
- SD command/response sets as defined in the SD Physical Layer specification v3.01
- SD Host Controller Standard Specification sets as defined in the SD card specification Part A2 v3.00

Main features of the SDIO host controller:

- Flexible architecture allowing support for new command structure
- 32-bit wide access bus to maximize bus throughput
- Designed for low power
- Programmable clock generation

- L4 slave interface supports:
  - 32-bit data bus width
  - 8/16/32 bit access supported
  - 9-bit address bus width
  - Streaming burst supported only with burst length up to 7
  - WNP supported
- Built-in 1024-byte buffer for read or write
- Two DMA channels, one interrupt line
- Supported data transfer rates up to SDR25 mode
- The SDIO controller is connected to 1,8V/3.3V compatible I/Os to support 1,8V/3.3V signaling

The differences between the SDIO host controller and a standard SD host controller defined by the *SD Card Specification, Part A2, SD Host Controller Standard Specification*, v3.00 are:

- The clock divider in the SDIO host controller supports a wider range of frequency than specified in the *SD Memory Card Specifications*, v3.0. The SDIO host controller supports odd and even clock ratioes.
- The SDIO host controller supports configurable busy time-out.
- There is no external LED control.

19.2 SDIO Environment

Figure 19-2 shows the SDIO host controller environment.

Figure 19-2. SDIO Controller Environment

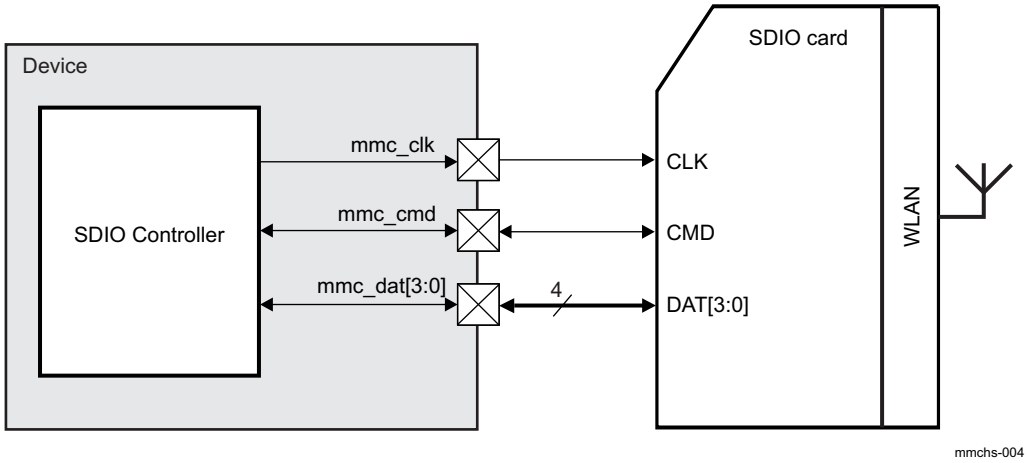


Table 19-1 describes the SDIO host controller I/O's.

Table 19-1. Description of SDIO host controller I/O's

Instance	Signal name	Direction	Description
MMC	mmc_clk	O	External clock for SDIO card
	mmc_cmd	I/O	Command line
	mmc_dat[3:0]	I/O	Data signals

NOTE: For the mmc_clk signal to work properly, the INPUTENABLE bit of the appropriate CTRL_CORE_PAD_x registers should be set to 0x1 by software. This is because the SDIO controller uses the input from the pad as loopback clock, which the controller can use for read capture.

19.2.1 Protocol and Data Format

The bus protocol between the SDIO host controller and the card is message-based. Each message is represented by one of the following parts:

- Command: A command starts an operation. The command is transferred serially from the SDIO host controller to the card on the CMD line.
- Response: A response is an answer to a command. The response is sent from the card to the SDIO host controller. It is transferred serially on the CMD line.
- Data: Data are transferred from the SDIO host controller to the card or from a card to the SDIO host controller using the data lines.
- Busy: The DAT[0] signal is maintained low by the card as far as it is programming the data received.
- CRC status: The CRC result is sent by the card through the DAT[0] line when executing a write transfer. In the case of transmission error, occurring on any of the active data lines, the card sends a negative CRC status on DAT[0]. In the case of successful transmission, over all active data lines, the card sends a positive CRC status on DAT[0] and starts the data programming procedure.

19.2.1.1 Protocol

There are specific commands for block-oriented operation. For information about commands and programming sequences, see the *SDIO Card Specification (Part E1)*.

Figure 19-3 and Figure 19-4 show how multiple block-oriented operations are defined. A multiple block-oriented operation sends a data block plus CRC bits. The transfer terminates when a stop command follows on the mmc_cmd line.

Figure 19-3. Multiple Block Read Operation

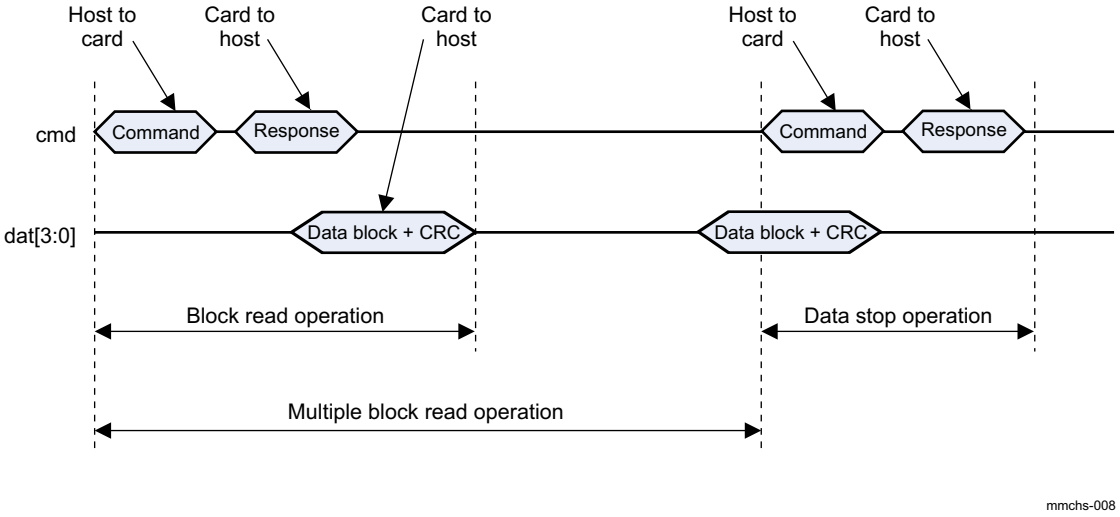
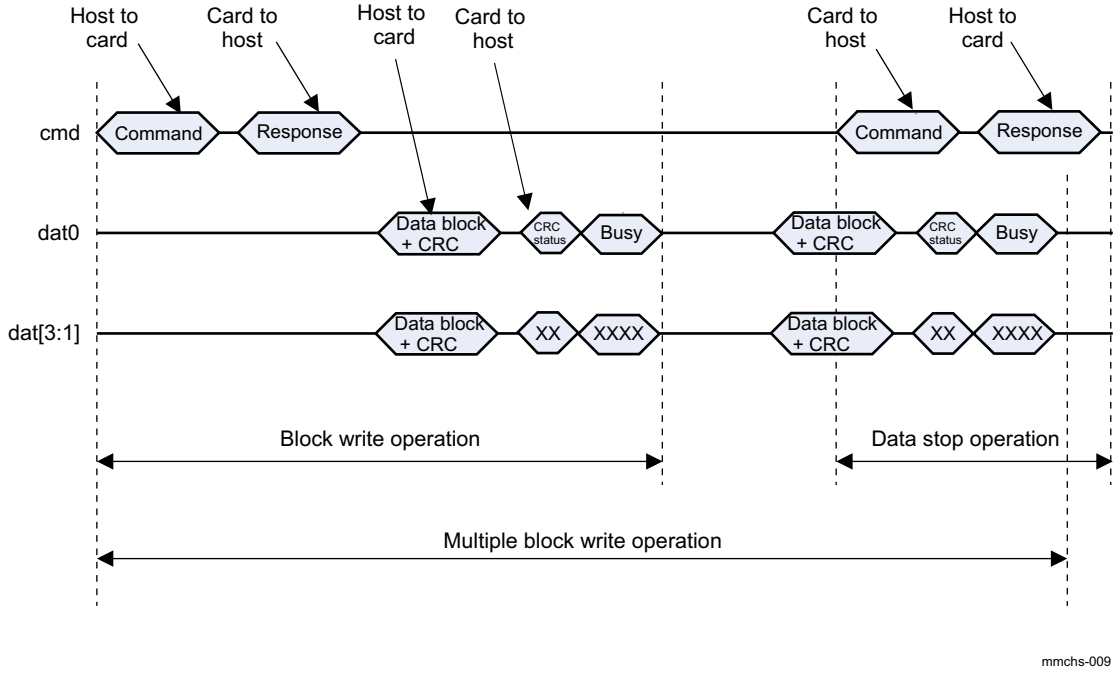


Figure 19-4. Multiple Block Write Operation With Card Busy Signal



NOTE:

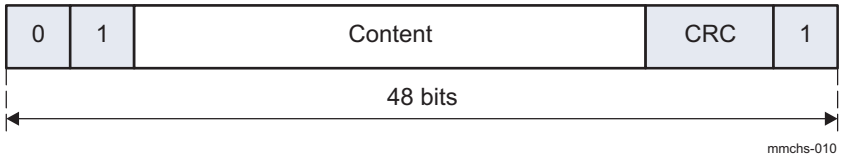
- The card busy signal is not always generated by the card; refer to Figure 19-3 and Figure 19-4, that show a particular case.
- Software must perform a software reset (set the MMCHS_SYSCTL[26] SRD bit to 0x1) after a data time-out to ensure that CLK is stopped.

19.2.1.2 Data Format

Coding Scheme for Command Token

Command tokens always start with 0 and end with 1. The second bit is a transmitter bit: 1 for a host command. The content is the command index (coded by 6 bits) and an argument (for example, an address), coded by 32 bits. The content is protected by 7-bit CRC checksum (see Figure 19-5).

Figure 19-5. Command Token Format



Coding Scheme for Response Token

Response tokens always start with 0 and end with 1. The second bit is a transmitter bit: 0 for a card response. The content is different for each type of response (R1, R2, R3, R4, and R5, R6, R7) and the content is protected by 7-bit CRC checksum (see Figure 19-6 and Figure 19-7). Depending on the type of commands sent to the card, the MMCHS_CMD register must be configured differently to avoid false CRC or index errors to be flagged on command response (see Table 19-2). For more information about response types, see the *SD Memory Card Specifications*, and *SDIO Card Specification*.

Figure 19-6. Response Token Format (R1, R3, R4, R5, R6, R7)

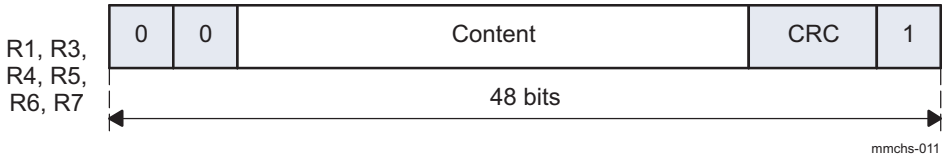


Figure 19-7. Response Token Format (R2)

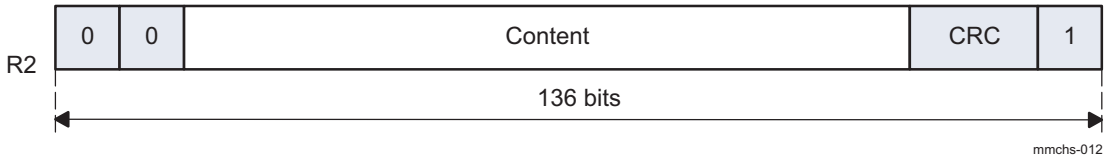


Table 19-2. Relationship Between Configuration and Name of Response Type

Response Type MMCHS_CMD[17:16] RSP_TYPE	Index Check Enable MMCHS_CMD[20] CICE	CRC Check Enable MMCHS_CMD[19] CCCE	Name of Response Type
00	0	0	No response
01	0	1	R2
10	0	0	R3 (R4 for SD cards)
10	1	1	R1, R6, R5, (R7 for SD cards)
11	1	1	R1b, R5b

Coding Scheme for Data Token

Data tokens always start with 0 and end with 1 (see Figure 19-8 and Figure 19-9).

Figure 19-8. Data Token Format for 1-Bit Transfers

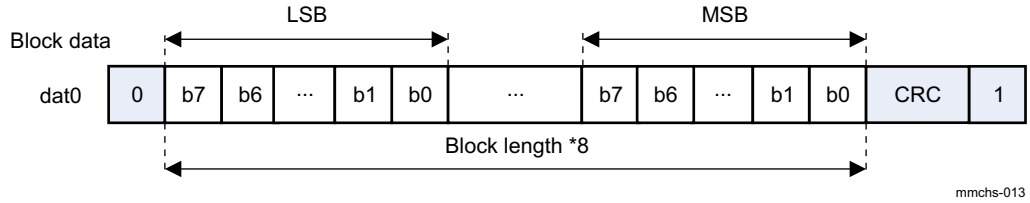
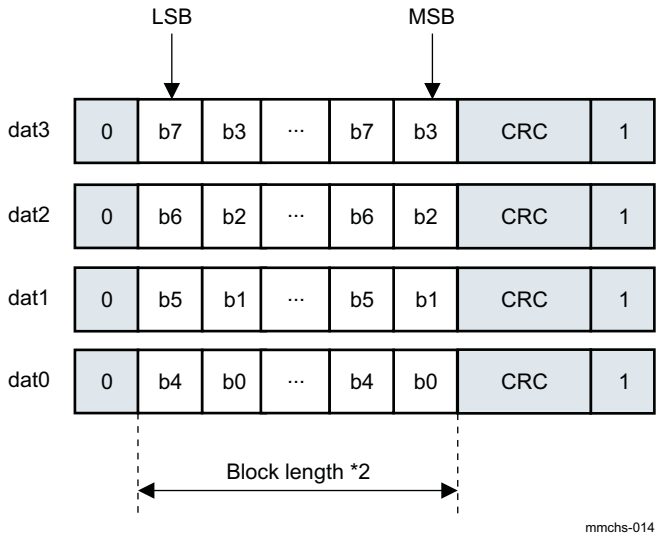


Figure 19-9. Data Token Format for 4-Bit Transfers

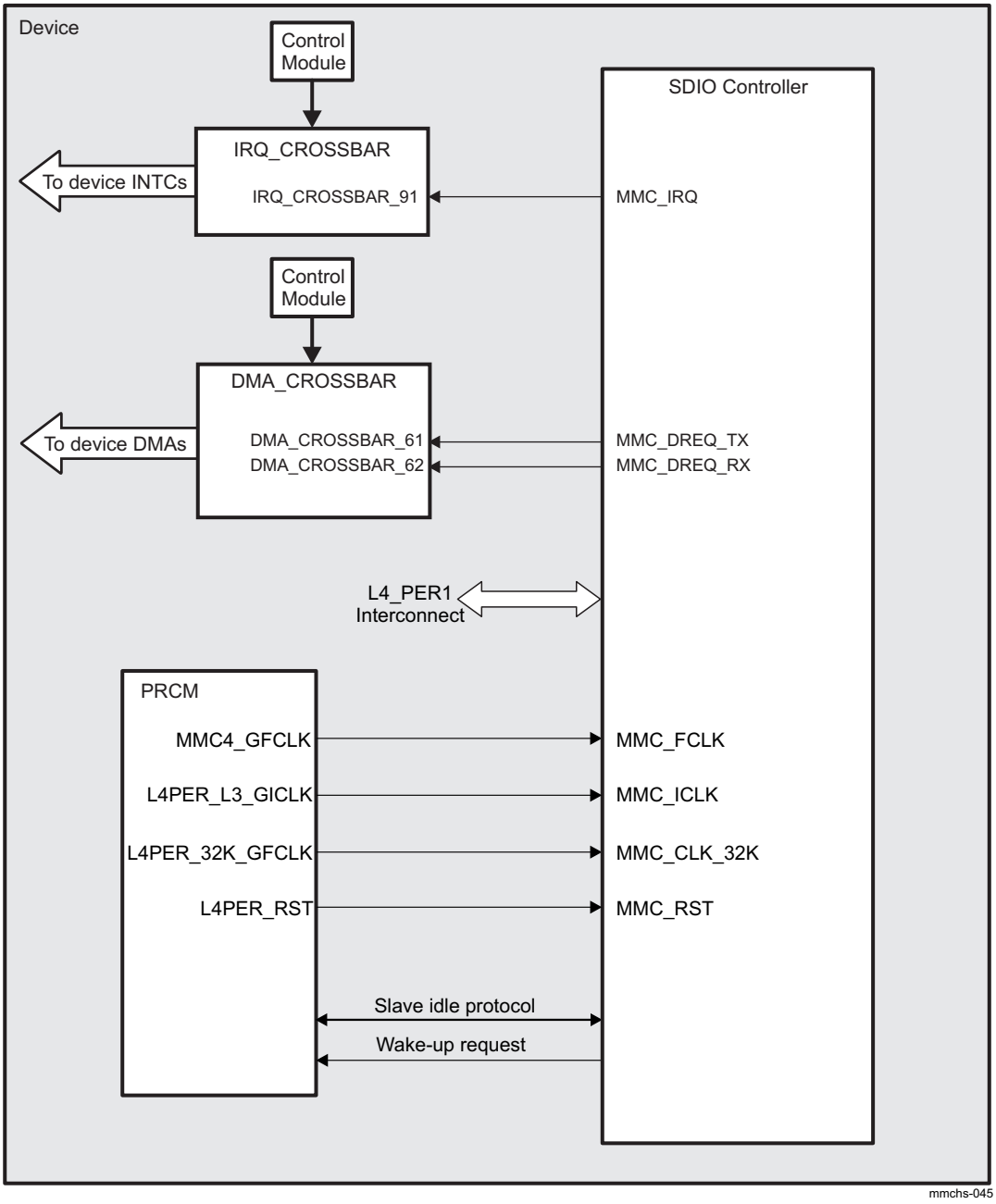


19.3 SDIO Integration

This section describes the integration of the SDIO controller in the device, including information about clocks, resets, and hardware requests.

Figure 19-10 shows the integration of the SDIO controller which is connected to the L4_PER1 interconnect and act as a slave only.

Figure 19-10. SDIO Integration



NOTE: For more information about the slave idle protocol and the wake-up request, see Section 3.1.1, *Device Power-Management Architecture Building Blocks*, in Chapter 3, *Power, Reset, and Clock Management*.

Table 19-3 through Table 19-5 summarize the integration of the module in the device.

Table 19-3. SDIO Controller Integration Attributes

Module Instance	Attributes	
	Power Domain	Interconnect
MMC	PD_COREAON	L4_PER1

Table 19-4. SDIO Controller Clocks and Resets

Clocks				
Module Instance	**Destination Signal Name**	**Source Signal Name**	**Source**	**Description**
MMC	MMC_ICLK	L4PER_L3_GICLK	PRCM	SDIO controller interface clock
	MMC_FCLK	MMC4_GFCLK	PRCM	SDIO controller functional clock
	MMC_CLK_32K	L4PER_32K_GFCLK	PRCM	SDIO controller debounce clock
Resets				
Module Instance	**Destination Signal Name**	**Source Signal Name**	**Source**	**Description**
MMC4	MMC_RST	L4PER_RST	PRCM	Reset to the SDIO controller

Table 19-5. SDIO Controller Hardware Requests

Interrupt Requests				
Module Instance	**Source Signal Name**	**Destination IRQ_CROSSBAR Input**	**Default Mapping**	**Description**
MMC	MMC_IRQ	IRQ_CROSSBAR_91	-	SDIO controller interrupt request. This IRQ source signal is not mapped by default to any device INTC.
DMA Requests				
Module Instance	**Source Signal Name**	**Destination DMA_CROSSBAR Input**	**Default Mapping**	**Description**
MMC	MMC_DREQ_TX	DMA_CROSSBAR_61	DMA_EDMA_DREQ_60	SDIO Controller DMA transmit request
	MMC_DREQ_RX	DMA_CROSSBAR_62	DMA_EDMA_DREQ_61	SDIO Controller DMA receive request

NOTE: The "**Default Mapping**" column in Table 19-5 *SDIO Controller Hardware Requests* shows the default mapping of module IRQ and DREQ source signals. These module IRQ and DREQ source signals can also be mapped to other lines of each device Interrupt or DMA controller through the IRQ_CROSSBAR and DMA_CROSSBAR modules, respectively.
For more information about the IRQ_CROSSBAR module, see Section 13.4.6.4, *IRQ_CROSSBAR Module Functional Description*, in Chapter 13, *Control Module*.
For more information about the DMA_CROSSBAR module, see Section 13.4.6.5, *DMA_CROSSBAR Module Functional Description*, in Chapter 13, *Control Module*.
For more information about the device interrupt controller, see Chapter 12, *Interrupt controller*.
For more information about the device EDMA module, see Chapter 11, *Enhanced DMA*.

NOTE:

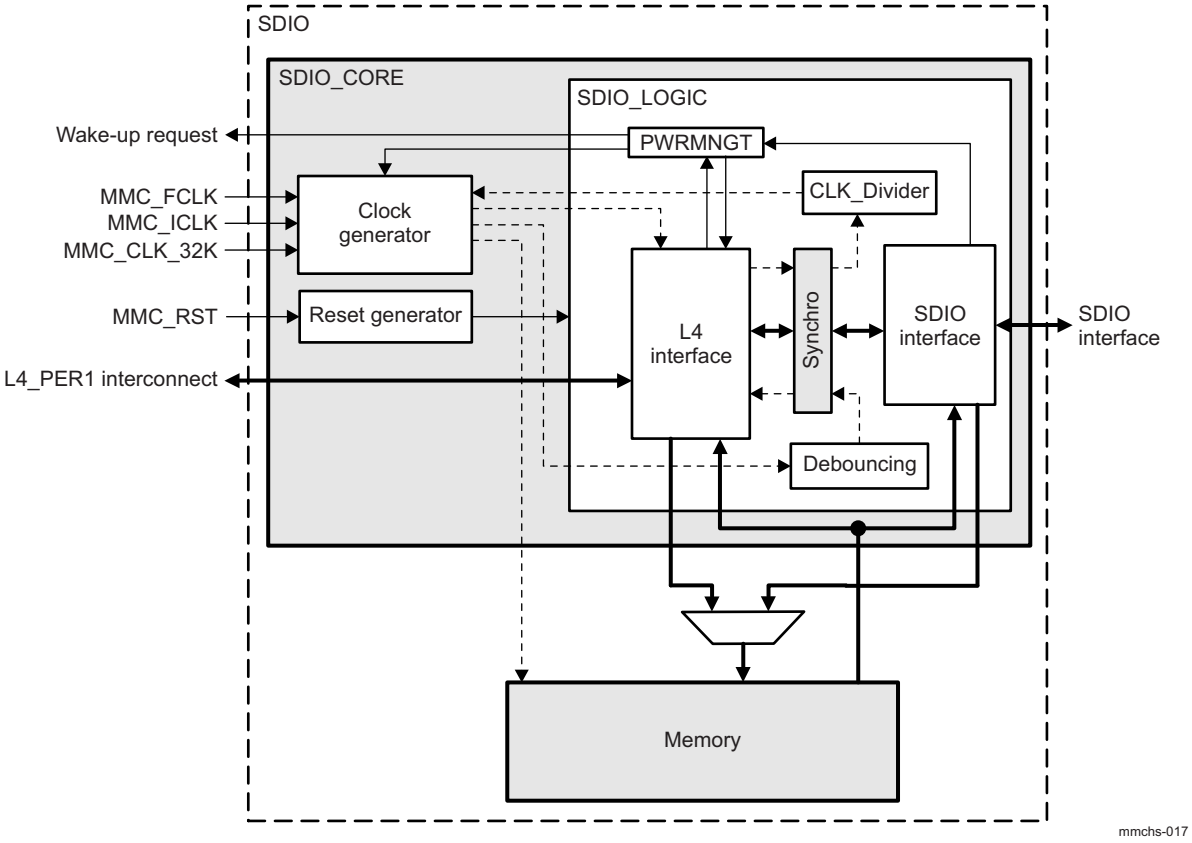
- For a description of the interrupt source, see Section 19.4.4, *Interrupt Requests*.
- For a description of the DMA source, see Section 19.4.5, *DMA Modes*.

19.4 SDIO Functional Description

19.4.1 Block Diagram

Figure 19-11 is a block diagram of the SDIO host controller.

Figure 19-11. SDIO Diagram



19.4.2 Resets

19.4.2.1 Hardware Reset

The module is reinitialized by the hardware (see Table 19-4 for more information about reset signals).

The MMCHS_SYSSTATUS[0] RESETDONE bit can be monitored by software to check whether the module is ready to use after a hardware reset.

NOTE: The functional clock (MMC_FCLK) and interface clock (MMC_ICLK) must be provided to the module to allow the RESETDONE status bit to be set.

The debounce clock (MMC_CLK_32K) must be active to reset the module correctly.

This hardware reset signal has a global reset action on the module. All configuration registers and all state-machines are reset in all clock domains.

19.4.2.2 Software Reset

The module is reinitialized by software through the MMCHS_SYSCONFIG[1] SOFTRESET bit. This bit has the same effect on the module logic as the hardware signal (MMC_RST), with the following exceptions:

- Debounce logic

- MMCHS_PSTATE, MMCHS_CAPA, and MMCHS_CUR_CAPA registers (see the corresponding register description)

The SOFTRESET bit is active high. The bit is automatically reinitialized to 0 by hardware. The MMCHS_SYSCTL[24] SRA bit has the same action on the design as the SOFTRESET bit.

The MMCHS_SYSSTATUS[0] RESETDONE bit can be monitored by software to check whether the module is ready to use after a software reset.

Moreover, two partial software reset bits are provided:

- MMCHS_SYSCTL[26] SRD
- MMCHS_SYSCTL[25] SRC

These two reset bits are useful to reinitialize data or command processes, respectively, in case of line conflict. When these bits are set to 1, a reset process is automatically released when the reset completes:

- The MMCHS_SYSCTL[26] SRD bit resets all finite state-machines (FSMs) and status management that handle data transfers on the interface and functional sides.
- The MMCHS_SYSCTL[25] SRC bit resets all FSMs and status management that handle command transfers on the interface and functional sides.

19.4.3 Power Management

The SDIO host controller can enter into different modes and save power:

- Normal mode
- Idle mode

The two modes are mutually exclusive (the module can be in normal mode or in idle mode). The SDIO host controller is compliant with the handshake protocol of the power, reset, and clock management (PRCM) module.

Normal Mode

The autogating of interface and functional clocks occurs when the following conditions are met:

- The MMCHS_SYSCONFIG[0] AUTOIDLE bit is set to 1.
- There is no transaction on the SDIO interface.

The autogating of interface and functional clocks stops when the following conditions are met:

- A register access occurs through the L4 interconnect.
- A wake-up event occurs (an interrupt from SDIO card).
- A transaction on the SDIO interface starts.

Idle Mode

The MMC_ICLK and MMC_FCLK clocks provided to the SDIO host controller are switched off upon a PRCM module request. They are switched back upon module request.

The SDIO host controller complies with the handshaking protocol of the PRCM module:

- IDLE request from the system power manager
- Idle acknowledgment from the SDIO host controller
- Wake-up request from the SDIO host controller

The idle acknowledgment varies according to the MMCHS_SYSCONFIG[4:3] SIDLEMODE bit field:

- 0x0: Force-idle mode. The SDIO host controller acknowledges the system power manager request unconditionally.
- 0x1: No-idle mode. The SDIO host controller ignores the system power manager request and behaves normally as if the request was not asserted.
- 0x2: Smart-idle mode. The SDIO host controller acknowledges the system power manager request according to its internal state.
- 0x3: Smart-idle wake-up-capable mode. The SDIO host controller acknowledges the system power manager request according to its internal state. However, the module may generate wake-up events

when it is in IDLE state (related to IRQ or DMA requests)

During the smart-idle mode period, the SDIO host controller acknowledges that the MMC_ICLK and MMC_FCLK clocks may be switched off, regardless of the value set in the MMCHS_SYSCONFIG[9:8] CLOCKACTIVITY bit field.

The debounce clock must never be switched off by the system power manager in order to detect wake-up in idle mode.

Transition From Normal Mode to Smart-Idle Mode

Smart-idle mode is enabled when the MMCHS_SYSCONFIG[4:3] SIDLEMODE bit field is set to 0x2 or 0x3.

The SDIO host controller goes into idle mode when the PRCM issues an IDLE request, according to its internal activity.

The SDIO host controller acknowledges the IDLE request from the PRCM after ensuring the following:

- The current multi- or single-block transfer is complete.
- Any interrupt or DMA request is asserted.
- There is no card interrupt on the DATA[1] signal.

As long as the SDIO controller does not acknowledge the IDLE request, if an event occurs, the SDIO host controller can still generate an interrupt or a DMA request. In this case, the module ignores the IDLE request from the PRCM module.

As soon as the SDIO controller acknowledges the IDLE request from the PRCM module:

- If smart-idle mode: The module does not assert any new interrupt or DMA request.
- If smart-idle wake-up-capable mode: The module may generate wake-up events related to an interrupt or DMA request.

Wake-Up Event in Smart-Idle Mode

The wake-up feature is enabled when both the MMCHS_SYSCONFIG[2] ENAWAKEUP bit and MMCHS_HCTL[24] IWE bits are set to 0x1.

The MMCHS_IE[8] CIRQ_ENABLE bit must also be set to 0x1 before going in idle mode. Setting it to 0x1 enables the wakeup event detection. The source of wakeup can be identified after idle mode exiting by reading register MMCHS_STAT

The wakeup is generated only in smart-idle mode, when the module is in idle mode.

Table 19-6 lists the supported cases in smart-idle mode.

Table 19-6. Smart-Idle Mode and Wake-Up Capabilities

Mode	MMC_ICLK Clock	MMC_FCLK Clock	Wake-Up Event
Card interrupt	May be switched off(1)	May be switched off(1)	The module sends an asynchronous wake-up request when a card interrupt on the DATA[1] signal is detected.

(1) The SDIO host controller assumes that both clocks may be switched off, regardless of the value set in the MMCHS_SYSCONFIG[9:8] CLOCKACTIVITY bit field.

Transition From Smart-Idle Mode to Normal Mode

The SDIO host controller detects the end of the idle period when the PRCM module deasserts the IDLE request.

For the wake-up event, there is a corresponding interrupt status in the MMCHS_STAT register. The SDIO host controller operates the conversion between the wake-up and interrupt (or DMA request) upon exit from smart-idle mode, if the associated enable bit is set in the MMCHS_ISE register.

Interrupts and wake-up events have independent enable and disable controls, accessible through the MMCHS_HCTL and MMCHS_ISE registers. The overall consistency must be ensured by software.

The MMCHS_STAT[8] CIRQ bit is updated with the event that caused the wake-up when theMMCHS_IE[8] CIRQ_ENABLE bit is enabled.

Then, the wake-up event at the origin of the transition from smart-idle mode to normal mode is converted into an interrupt or DMA request. (The MMCHS_STAT register is updated and the status of the interrupt signal is changed.)

When the IDLE request from the PRCM module is deasserted, the module switches back to normal mode. The module is fully operational.

Force-Idle Mode

Force-idle mode is enabled when the MMCHS_SYSCONFIG[4:3] SIDLEMODE bit field is set to 0x0.

Force-idle mode is an idle mode in which the SDIO host controller responds unconditionally to the IDLE request from the PRCM module. Moreover, in this mode, the SDIO host controller unconditionally deasserts interrupts and DMA request lines if they are asserted.

The transition from normal mode to force-idle mode does not affect the bits of the MMCHS_STAT register.

In force-idle mode, the interrupt and DMA request lines are deasserted. MMC_ICLK and MMC_FCLK can be switched off.

CAUTION

In force-idle mode, an IDLE request from the PRCM module during a command or a data transfer can lead to an unexpected and unpredictable result. When the module is idle, any access to the module generates an error as long as the MMC_ICLK clock is alive.

The module exits force-idle mode when the PRCM module deasserts the IDLE request. Then the module switches back to normal mode. The module is fully operational. Interrupt and DMA request lines are optionally asserted one clock cycle later.

Power Pad Control

The SDIO host controller has the ability to reduce the pad power leakage when no transfer is sent through the pad. According to the MMCHS_CON[15] PADEN there are two different pad power management modes: automatic and manual.

If MMCHS_CON[15] PADEN is set to 1, pads mmc_clk, mmc_cmd, mmc_dat[0], mmc_dat[2] and mmc_dat[3] are always powerd on.

If MMCHS_CON[15] PADEN is set to 0, the power for pads mmc_clk, mmc_cmd, mmc_dat[0], mmc_dat[2] and mmc_dat[3] is "ON" only when there is a transfer on going. This is automatically managed by an internal state machine of the SDIO host controller.

The DAT[1] pad active state is controlled through MMCHS_CON[11] CTPL in order to detect SDIO asynchronous interrupt when there is no transaction.

The delay between pad power "ON" and the command transmission is controlled through the MMCHS_PWCNT register which act as a programmable counter. It is also used to delay the pad power "OFF" after the end of transmission. By default this counter is reset which means that no additional delay is added. But there is approximately 6-7 clock cycles margin between pad power "ON" and real start of the command due to an internal state machine of the SDIO host controller.

NOTE: The MMCHS_PWCNT register is considered as static. No dynamic configuration during the transfer is supported. This results in an unpredictable behavior.

Local Power Management

Table 19-7 describes the power-management features available for the SDIO controller.

NOTE: For information about source clock gating and a description of the sleep/wake-up transitions, see Section 3.6, *Clock Management Functional Description*, in Chapter 3, *Power, Reset, and Clock Management*.

Table 19-7. Local Power-Management Features

Feature	Registers	Description
Clock auto gating	MMCHS_SYSCONFIG[0] AUTOIDLE	This bit allows a local power optimization inside the module by gating the MMC_ICLK clock upon the interface activity, or gating the MMC_FCLK clock upon the internal activity.
Slave-idle modes	MMCHS_SYSCONFIG[3:4] SIDLEMODE	Force-idle, No-idle, Smart-idle and Smart-idle wake-up-capable modes are available.
Clock activity	MMCHS_SYSCONFIG[8:9] CLOCKACTIVITY	For configuration details, see Table 19-8.
Global wake-up enable	MMCHS_SYSCONFIG[2] ENAWAKEUP	This bit enables the wake-up feature at the module level.
Wake-up sources enable	MMCHS_HCTL register	This register holds one active-high enable bit per event source that is able to generate a wake-up signal.

Table 19-8. Clock Activity Settings

CLOCKACTIVITY Values	Clock State When Module is in IDLE State		Features Available When Module is in IDLE State	Wake-Up Events
	MMC_ICLK	MMC_FCLK		
00	OFF	OFF	None	Card interrupt
10	OFF	ON	None	
01	ON	OFF	None	
11	ON	ON	All	

CAUTION

The PRCM module has no hardware means of reading CLOCKACTIVITY settings. Thus, software must ensure consistent programming between the CLOCKACTIVITY and SDIO clock PRCM control bits. For a description of the Clock activity feature, see Section 3.1.1.1.2, *Module-Level Clock Management*, in Chapter 3, *Power, Reset, and Clock Management*.

19.4.4 Interrupt Requests

Several internal module events can generate an interrupt. Each interrupt has a status bit, an interrupt enable bit, and a signal status enable:

- The status of each type of interrupt is automatically updated in the MMCHS_STAT register; it indicates which service is required.
- The interrupt status enable bits of the MMCHS_IE register enable or disable the automatic update of the MMCHS_STAT register on an event-by-event basis.
- The interrupt signal enable bits of the MMCHS_ISE register enable or disable the transmission of an interrupt request on the interrupt line MMC_IRQ on an event-by-event basis.

If an interrupt status is disabled in the MMCHS_IE register, then the corresponding interrupt request is not transmitted, and the value of the corresponding interrupt signal enable in the MMCHS_ISE register is ignored.

When an interrupt event occurs, the corresponding status bit is automatically set to 0x1 (the SDIO host controller updates the status bit) in the MMCHS_STAT register. If a mask is later applied on the interrupt in the MMCHS_ISE register, the interrupt request is deactivated.

When the interrupt source has not been serviced, if the interrupt status is cleared in the MMCHS_STAT register and the corresponding mask is removed from the MMCHS_ISE register, the interrupt status is not asserted again in the MMCHS_STAT register and the SDIO host controller does not transmit an interrupt request.

NOTE: If the buffer write ready (BWR) interrupt or the buffer read ready (BRR) only interrupt are not serviced and are cleared in the MMCHS_STAT register, and the corresponding mask is removed, then the SDIO host controller waits for the service of the interrupt without updating the status MMCHS_STAT register or transmitting an interrupt request.

Table 19-9 lists the event flags, and their mask, that can cause module interrupts.

Table 19-9. Events

Event Flag	Event Mask	Map to	Description
MMCHS_STAT[29] BADA	MMCHS_IE[29] BADA_ENABLE	MMC_IRQ	Bad access to data space. This bit is set automatically to indicate a bad access to buffer when not allowed: This bit is set during a read access to the data register (MMCHS_DATA) while buffer reads are not allowed (MMCHS_PSTATE[11] BRE = 0). This bit is set during a write access to the data register (MMCHS_DATA) while buffer writes are not allowed (MMCHS_PSTATE[10] BWE = 0).
MMCHS_STAT[28] CERR	MMCHS_IE[28] CERR_ENABLE	MMC_IRQ	Card error. This bit is set automatically when there is at least one error in a response of type R1, R1b, R6, R5, or R5b. Only bits referenced as type E (error) in the status field in the response can set a card status error. An error bit in the response is flagged only if the corresponding bit in the card status response error MMCHS_CSRE is set. There is no card error detection for the auto CMD12 command.
MMCHS_STAT[24] ACE	MMCHS_IE[24] ACE_ENABLE	MMC_IRQ	Auto CMD12 error. This bit is set automatically when one of the bits MMCHS_AC12[4:0] changes from 0 to 1.
MMCHS_STAT[22] DEB	MMCHS_IE[22] DEB_ENABLE	MMC_IRQ	Data end bit error. This bit is set automatically when detecting a 0 at the end bit position of read data on the DAT line or at the end position of the CRC status in write mode.
MMCHS_STAT[21] DCRC	MMCHS_IE[21] DCRC_ENABLE	MMC_IRQ	Data CRC error. This bit is set automatically when there is a CRC16 error in the data phase response following a block read command or if there is a 3-bit CRC status difference of a position 010 token during a block write command.
MMCHS_STAT[20] DTO	MMCHS_IE[20] DTO_ENABLE	MMC_IRQ	Data time-out error. This bit is set automatically according to the following conditions: • Busy time-out for R1b, R5b response type • Busy time-out after write CRC status • Write CRC status time-out • Read data time-out
MMCHS_STAT[19] CIE	MMCHS_IE[19] CIE_ENABLE	MMC_IRQ	Command index error. This bit is set automatically when the response index differs from the corresponding command index previously emitted. The check is enabled through the MMCHS_CMD[20] CICE bit.
MMCHS_STAT[18] CEB	MMCHS_IE[18] CEB_ENABLE	MMC_IRQ	Command end bit error. This bit is set automatically when detecting a 0 at the end bit position of a command response.

Table 19-9. Events (continued)

Event Flag	Event Mask	Map to	Description
MMCHS_STAT[17] CCRC	MMCHS_IE[17] CCRC_ENABLE	MMC_IRQ	Command CRC error. This bit is set automatically when a CRC7 error occurs in the command response. CRC check is enabled through the MMCHS_CMD[19] CCCE bit.
MMCHS_STAT[16] CTO	MMCHS_IE[16] CTO_ENABLE	MMC_IRQ	Command time-out error. This bit is set automatically when no response is received within 64 clock cycles from the end bit of the command. For commands that reply within five clock cycles, the time-out is still detected at 64 clock cycles.
MMCHS_STAT[15] ERRI	MMCHS_IE[15] ERRI_ENABLE	MMC_IRQ	Error interrupt. If any of the bits in the error interrupt status register (MMCHS_STAT[31:16]) are set, this bit is set to 1.
MMCHS_STAT[8] CIRQ	MMCHS_IE[8] CIRQ_ENABLE	MMC_IRQ	Card interrupt. The interrupt source is sampled during the interrupt cycle.
MMCHS_STAT[5] BRR	MMCHS_IE[5] BRR_ENABLE	MMC_IRQ	Buffer read ready. This bit is set automatically during a read operation to the card (see class 2 block-oriented read commands) when one block specified by the MMCHS_BLK[10:0] BLEN bit field is completely written in the buffer. It indicates that the memory card has filled out the buffer and that the LH must empty the buffer by reading it.
MMCHS_STAT[4] BWR	MMCHS_IE[4] BWR_ENABLE	MMC_IRQ	Buffer write ready. This bit is set automatically during a write operation to the card (see class 4 block-oriented write command) when the host can write a complete block as specified by the MMCHS_BLK[10:0] BLEN bit field. It indicates that the memory card has emptied one block from the buffer and that the LH can write one block of data into the buffer.
MMCHS_STAT[2] BGE	MMCHS_IE[2] BGE_ENABLE	MMC_IRQ	Block gap event. When a stop at the block gap is requested (MMCHS_HCTL[16] SBGR), this bit is automatically set when transaction is stopped at the block gap during a read or write operation.
MMCHS_STAT[1] TC	MMCHS_IE[1] TC_ENABLE	MMC_IRQ	Transfer completed. This bit is always set when a read/write transfer is complete or between two blocks when the transfer is stopped because of a stop at block gap request (MMCHS_HCTL[16] SBGR). • In read mode: This bit is automatically set when a read transfer completes (MMCHS_PSTATE[9] RTA). • In write mode: This bit is automatically set when the DAT line use completes (MMCHS_PSTATE[2] DLA).
MMCHS_STAT[0] CC	MMCHS_IE[0] CC_ENABLE	MMC_IRQ	Command complete. This bit is set when a 1-to-0 transition occurs in the register command inhibit (MMCHS_PSTATE[0] CMDI). If the command is a type for which no response is expected, then the command complete interrupt is generated at the end of the command. A command time-out error (MMCHS_STAT[16] CTO) has higher priority than command complete (MMCHS_STAT[0] CC). If a response is expected but none is received, then a command time-out error is detected and signaled, instead of the command complete interrupt.

NOTE: To send an interrupt request to the MMC_IRQ line, the mask/unmask bit must be set in the MMCHS_IE and MMCHS_ISE registers.

The SDIO host controller supports interrupt-driven operation and polling.

19.4.4.1 Interrupt-Driven Operation

An interrupt-enable bit must be set in the MMCHS_IE register to enable the module internal source of interrupt.

When an interrupt event occurs, the single interrupt line is asserted and the LH must:

1. Read the MMCHS_STAT register to identify which event occurred.
2. Write 1 into the corresponding bit of the MMCHS_STAT register to clear the interrupt status and release the interrupt line (if a read is done after this write, this returns 0).

NOTE: In the MMCHS_STAT register, the card interrupt (CIRQ) and error interrupt (ERRI) bits cannot be cleared.

The MMCHS_STAT[8] CIRQ status bit must be masked by disabling the MMCHS_IE[8] CIRQ_ENABLE bit (set to 0x0), and then the interrupt routine must clear the SDIO interrupt source in the SDIO card common control register (CCCR).

The MMCHS_STAT[15] ERRI bit is automatically cleared when all status bits in the MMCHS_STATregister (bits 31 through 16) are cleared.

19.4.4.2 Polling

When the interrupt capability of an event is disabled in the MMCHS_ISE register, the interrupt line is not asserted:

- Software can poll the status bit in the MMCHS_STAT register to detect when the corresponding event occurs.
- Writing 1 into the corresponding bit of the MMCHS_STAT register clears the interrupt status and does not affect the interrupt line state.

NOTE: See the previous note concerning clearing of the CIRQ and ERRI bits.

19.4.4.3 Asynchronous Interrupt

Asynchronous interrupt is defined in *SDIO Card Specification version 3.00, part E*. This interrupt is effective in 4-bit mode and is generated without SD clock. Asynchronous interrupt period is defined in the synchronous interrupt period after the last data block and until a next command is received.

Asynchronous interrupt period in a multiple block write operation.

If MMCHS_CAPA[29] AIS is set to 0, writing to MMCHS_AC12[30] AI_ENABLE is ignored. This bit is set to 0. A synchronous interrupt period starts two clocks after the last data block. If MMCHS_AC12[30] AI_ENABLE is set to 1, the asynchronous interrupt period starts four clocks after the start of the synchronous interrupt period. Four clocks after the start bit of the next command, the asynchronous interrupt period ends and goes back to the synchronous interrupt period.

19.4.5 DMA Modes

The SDIO controller supports DMA slave mode. It is selected by setting the MMCHS_CON[20] DMA_MNS bit to 0. In this case, the controller is slave on the DMA transaction managed by two separated requests (MMC_DMA_TX and MMC_DMA_RX).

19.4.5.1 Slave DMA Operations

The SDIO host controller can be interfaced with a DMA controller. At the system level, the advantage is to discharge the LH of the data transfers.

The DMA request is issued if the following conditions are met:

- The MMCHS_CMD[0] DE bit is set to 1 to trigger the initial DMA request (the write must be done when running the data transfer command).
- A command was emitted on the CMD line.
- There is enough space in the buffer of the SDIO host controller to write an entire block (BLEN writes).

19.4.5.1.1 DMA Receive Mode

In a DMA block read operation (single or multiple), the request signal MMC_DMA_RX is asserted to its active level when a complete block is written in the buffer. The block size transfer is specified in the MMCHS_BLK[10:0] BLEN bit field.

MMC_DMA_RX is deasserted to its inactive level when the a certain device DMA module reads one word from the buffer.

Only one request is sent per block; the DMA controller can make a 1-shot read access or several DMA bursts, in which case the DMA controller must manage the number of burst accesses, according to the BLEN bit field block size.

New DMA requests are internally masked if the DMA has not read exactly BLEN bytes and a new complete block is not ready. Because DMA accesses are 32-bit accesses, the number of DMA reads is Integer(BLEN / 4) + 1.

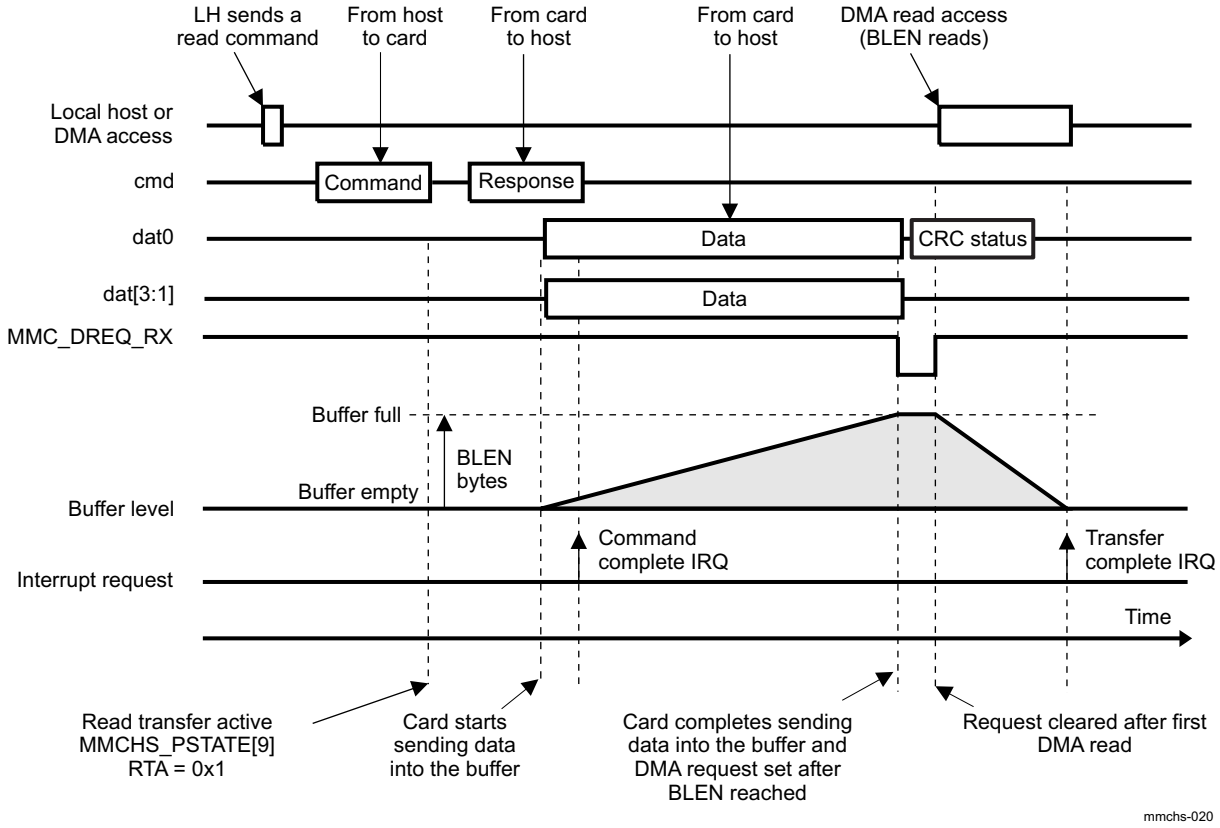
The receive buffer never overflows. In multiple block transfers for block sizes larger than 512 bytes, when the buffer becomes full, the mmc_clk clock signal (provided to the card) is momentarily stopped until a certain device DMA or another initiator performs a read access, which reads a complete block in the buffer.

To summarize:

- DMA transfer size = BLEN buffer size in one shot or by burst
- One DMA request per block

Figure 19-12 shows DMA receive mode.

Figure 19-12. DMA Receive Mode



19.4.5.1.2 DMA Transmit Mode

In a DMA block write operation (single or multiple), the request signal MMC_DMA_TX is asserted to its active level when a complete block is to be written to the buffer. The block size transfer is specified in the MMCHS_BLK[10:0] BLEN bit field.

MMC_DMA_TX is deasserted to its inactive level when a certain device DMA writes one word to the buffer.

Only one request is sent per block; the DMA controller can make a 1-shot write access or multiple write DMA bursts, in which case the DMA controller must manage the number of burst accesses, according to the BLEN bit field block size.

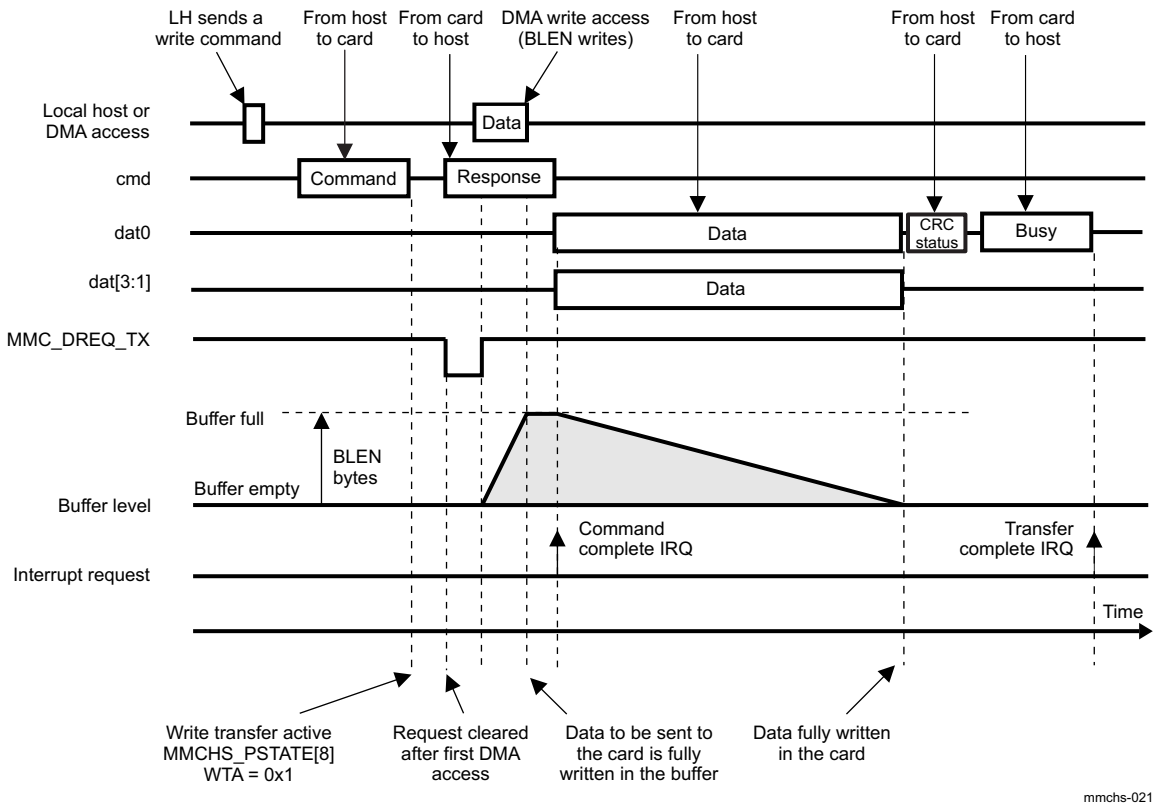
New DMA requests are internally masked if the DMA has not written exactly BLEN bytes (because DMA accesses are 32-bit accesses, the number of DMA reads is Integer(BLEN / 4) + 1) and if there is not enough memory space to write a complete block in the buffer.

To summarize:

- DMA transfer size = BLEN buffer size in one shot or by burst
- One DMA request per block

Figure 19-13 shows DMA transmit mode.

Figure 19-13. DMA Transmit Mode



19.4.6 Mode Selection

The SDIO host controller has been designed to be the most transparent with the type of the card.

The type of the card connected is differentiated by the software initialization procedure. Software identifies the type of card connected during software initialization. For each card type, there are corresponding commands. Some commands are not supported by all cards. For more information, see the *SD Memory Card Specifications* and *SDIO Card Specification, Part E1*.

19.4.7 Buffer Management

19.4.7.1 Data Buffer

The SDIO host controller uses a data buffer. This buffer transfers data from one data bus (interconnect) to another data bus (SDIO bus) and vice versa.

The buffer is the heart of the interface and ensures the transfer between the two interfaces (interconnect and the card).

To enhance performance, the data buffer is completed by a prefetch register and a post-write buffer that are not accessible by the host controller.

The read access time of the prefetch register is faster than that of the data buffer. The prefetch register allows data to be read from the data buffer at an increased speed by preloading data into the prefetch register.

The entry point of the prefetch buffer and the post-write buffer is the 32-bit MMCHS_DATA register. A write access to the MMCHS_DATA register followed by a read access from the MMCHS_DATA register corresponds to a write access to the post-write buffer followed by a read access to the prefetch buffer. As a consequence, it is normal that the data of the write access to the MMCHS_DATA register and the data of the read access to the MMCHS_DATA register are different.

The number of 32-bit accesses to the MMCHS_DATA register that are needed to read (or write) a data block with a size of the MMCHS_BLK[11:0] BLEN bit field is equal to the rounded up result of BLEN divided by 4.

The maximum block size supported by the host controller is hard-coded in the MMCHS_CAPA[17:16] MBL bit field and cannot be changed.

A read access to the MMCHS_DATA register is allowed only when the buffer read-enable status is set to 1 (the MMCHS_PSTATE[11] BRE bit); otherwise, a bad access (the MMCHS_STAT[29] BADA bit) is signaled.

A write access to the MMCHS_DATA register is allowed only when the buffer write-enable status is set to 1 (the MMCHS_PSTATE[10] BWE bit); otherwise, a bad access (the MMCHS_STAT[29] BADA bit) is signaled and the data are not written.

The data buffer has two modes of operation to store and read of the first and second portions of the data buffer:

- When the size of the data block to transfer is less than or equal to MEM_SIZE/2 (in double-buffering), two data transfers can occur at the same time from one data bus to the other data bus, and vice versa. The SDIO host controller uses the two portions of the data buffer in a ping-pong manner so that storing and reading the first and second portions of the data buffer are automatically interchanged from time to time. In this way, data can be read from one portion (for instance, through a DMA read access on the interconnect bus) while data (for instance, from the card) are being stored into the other portion, and vice versa. When BLEN is less than or equal to 0x200 (that is, less than or equal to 512 bytes), each of the two portions of the buffer that can be used have a size of BLEN (that is, 32 bits × BLEN divided by 4). No more than this total size of 2 × 32 bits × BLEN divided by 4 can be used.

CAUTION

The MMCHS_CMD[4] DDIR bit must be configured before a transfer to indicate the direction of the transfer.

Figure 19-14 and Figure 19-15 show the buffer management for a write and a read, respectively.

Figure 19-14. Buffer Management for a Write

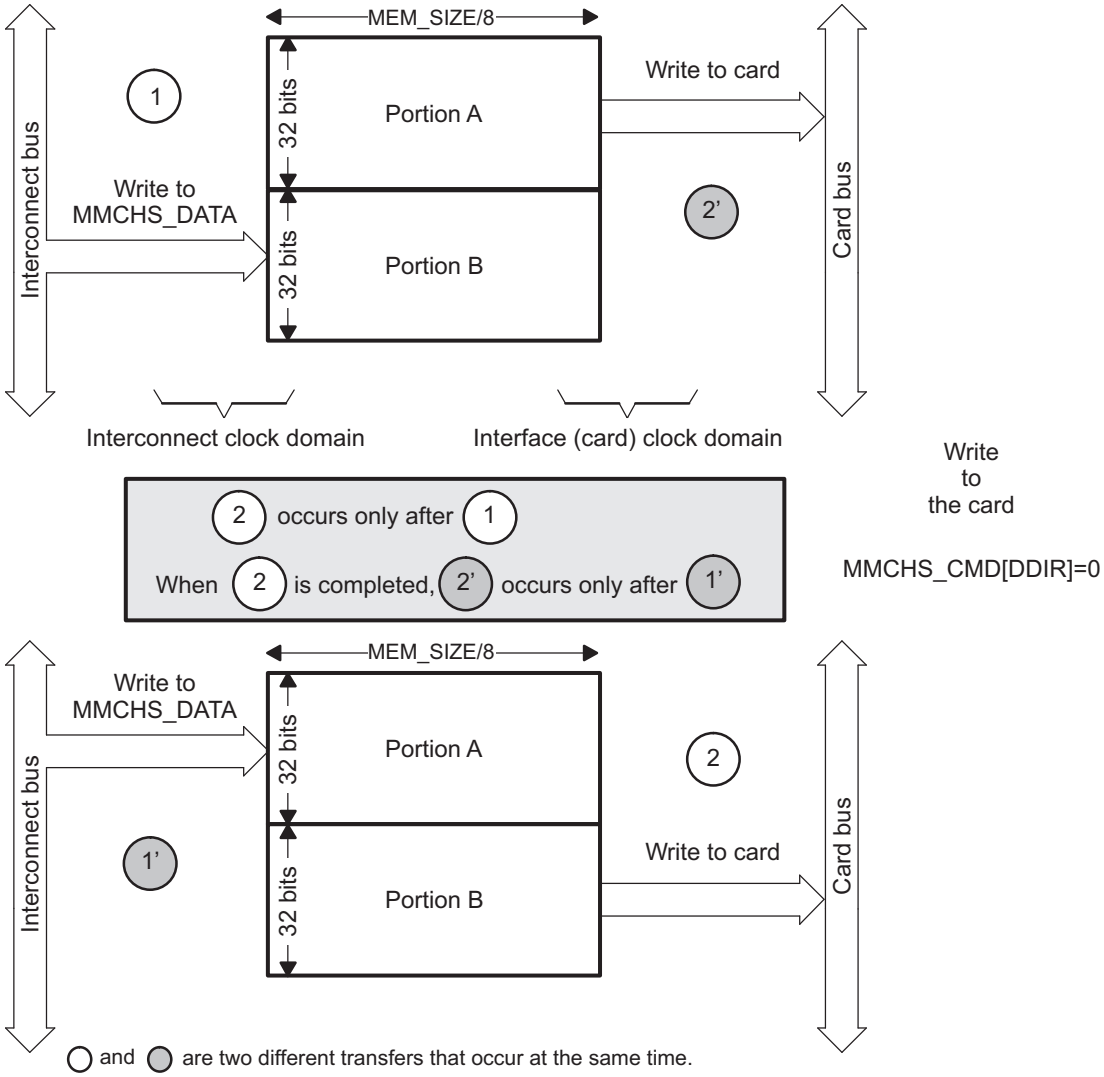
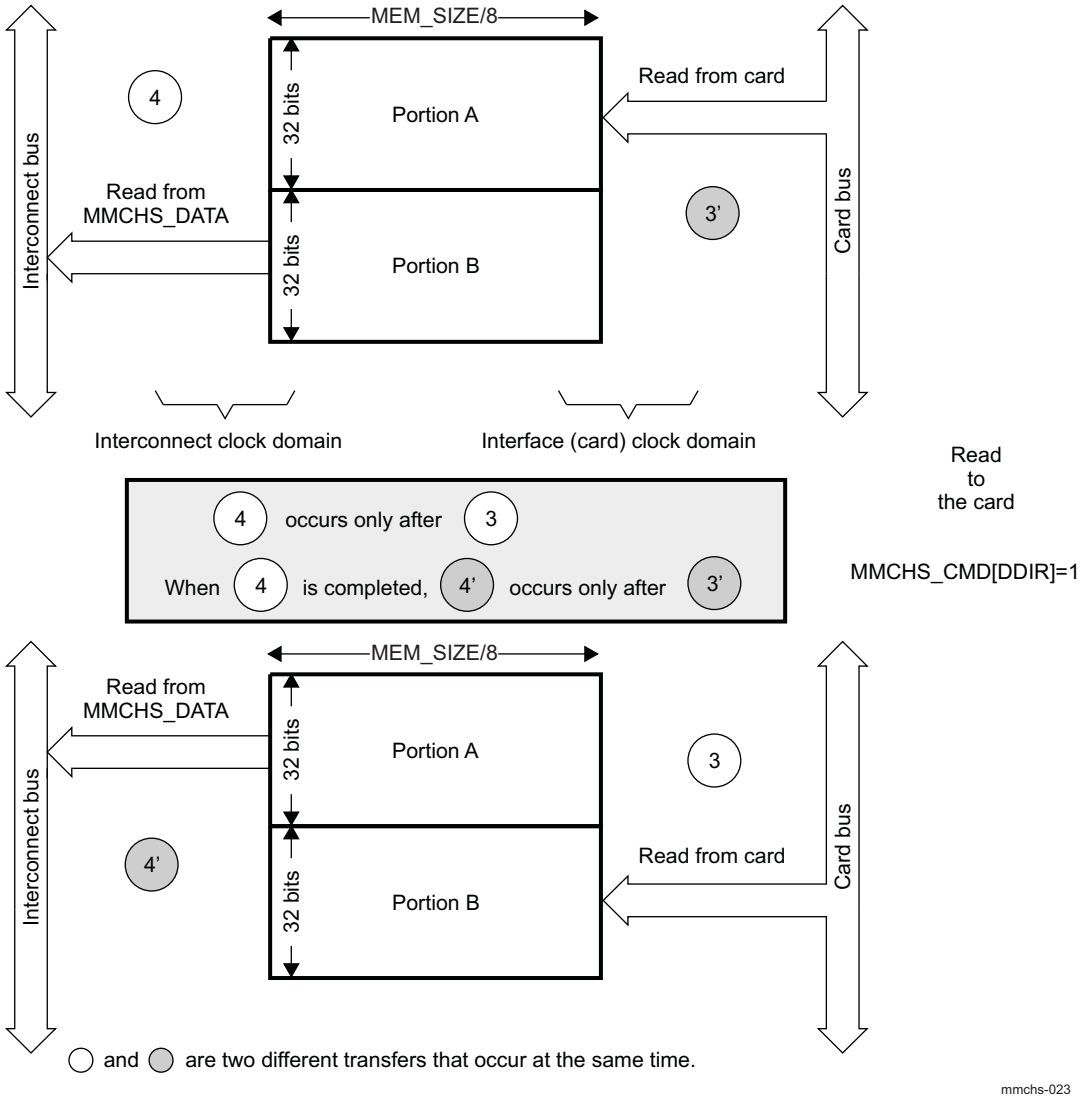


Figure 19-15. Buffer Management for a Read



- When the size of the data block to transfer is larger than MEM_SIZE / 2, only one data transfer at a time can occur from one data bus to the other data bus. The SDIO host controller uses the entire data buffer as a single portion.

  In this mode, a bad access (the MMCHS_STAT[29] BADA bit) is signaled when two data transfers occur at the same time from one data bus to the other data bus, and vice versa.

19.4.7.1.1 Memory Size, Block Length, and Buffer-Management Relationship

The maximum block length and buffer management that can be targeted by the system depend on the memory depth setting (see Table 19-10).

NOTE: Double-buffering is always the buffer management for large memory depth.

Table 19-10. Memory Size, BLEN, and Buffer Relationship

Memory Size (MMCHS_HL_HWINFO[5:2] MEM_SIZE in bytes)	512	1024
Maximum block length supported	512	1024

Table 19-10. Memory Size, BLEN, and Buffer Relationship (continued)

Memory Size (MMCHS_HL_HWINFO[5:2] MEM_SIZE in bytes)	512	1024
Double-buffering for maximum block length	N/A	BLEN <= 512
Single-buffering for block length	BLEN <= 512	512 < BLEN <= 1024

NOTE: For single-buffering management, throughput on the SDIO bus interface deteriorates in multiblock transfers, because the controller must wait for the filling or emptying of the buffer between each block transfer on the SDIO bus. The clock is maintained on write SDIO transfers (the MMCHS_CMD[4] DDIR bit is 0) and halted on read SDIO transfers (the MMCHS_CMD[4] DDIR bit is 1).

19.4.7.1.2 Data Buffer Status

The data buffer status is defined in the following interrupt status register and status register:

- Interrupt status registers:
  - MMCHS_STAT[29] BADA: Bad access to data space
  - MMCHS_STAT[5] BRR: Buffer read ready
  - MMCHS_STAT[4] BWR: Buffer write ready
- Status registers:
  - MMCHS_PSTATE[11] BRE: Buffer read enable
  - MMCHS_PSTATE[10] BWE: Buffer write enable

19.4.8 Transfer Process

The process of a transfer depends on the type of command. It can be with or without a response, and with or without data.

19.4.8.1 Different Types of Commands

Different types of commands are specific for the SD and SDIO cards. For more information, see the *SD Memory Card Specifications* and *SDIO Card Specification, Part E1*.

19.4.8.2 Different Types of Responses

Different types of responses are specific for the SD and SDIO cards. For more information, see the *SD Memory Card Specifications* and *SDIO Card Specification, Part E1*.

Table 19-11 describes how the SD/SDIO responses are stored in the MMCHS_RSPxx registers.

Table 19-11. SD/SDIO Responses in the MMCHS_RSPxx Registers

Type of Response	Response Field	Response Register
R1, R1b (normal response), R3, R4, R5, R5b, R6, R7	RESP[39:8](1)	MMCHS_RSP10[31:0]
R1b (Auto CMD12 response), R1	RESP[39:8](1)	MMCHS_RSP76[31:0]
R2	RESP[127:0](1)	MMCHS_RSP76[31:0] MMCHS_RSP54[31:0] MMCHS_RSP32[31:0] MMCHS_RSP10[31:0]

(1) RESP refers to the command response format described in the specifications mentioned.

When the host controller modifies part of the MMCHS_RSPxx registers, it preserves the unmodified bits.

The host controller stores the Auto CMD12 response in the MMCHS_RSP76[31:0] register because the host controller may execute multiple block data transfers on the data line concurrently with a command. This allows the host controller to avoid overwriting the response of Auto CMD12 with the command response stored in the MMCHS_RSP10 register, and vice versa.

19.4.9 Transfer or Command Status and Errors Reporting

Flags in the SDIO host controller show the status of communication with the card:

- A time-out (of a command, data, or response)
- A CRC error

Error conditions generate interrupts. For more information, see Table 19-12 and the register description.

Table 19-12. CC and TC Values Upon Error Detected

Error Hold in MMCHS_STAT		CC	TC	Comments
29	BADA			No dependency with CC or TC BADA is related to the MMCHS_DATA register accesses. Its assertion does not depend on the ongoing transfer.
28	CERR	1		CC is set upon CERR.
22	DEB		1	TC is set upon DEB.
21	DCRC		1	TC is set upon DCRC.
20	DTO			DTO and TC are mutually exclusive. DCRC and DEB cannot occur with DTO.
19	CIE	1		CC is set upon CIE.
18	CEB	1		CC is set upon CEB.
17	CCRC	1		CC can be set upon CCRC. See CTO comment.
16	CTO			CTO and CC are mutually exclusive. CIE, CEB, and CERR cannot occur with CTO. CTO can occur at the same time as CCRC: It indicates a command abort due to contention on the CMD line. In this case no CC appears.

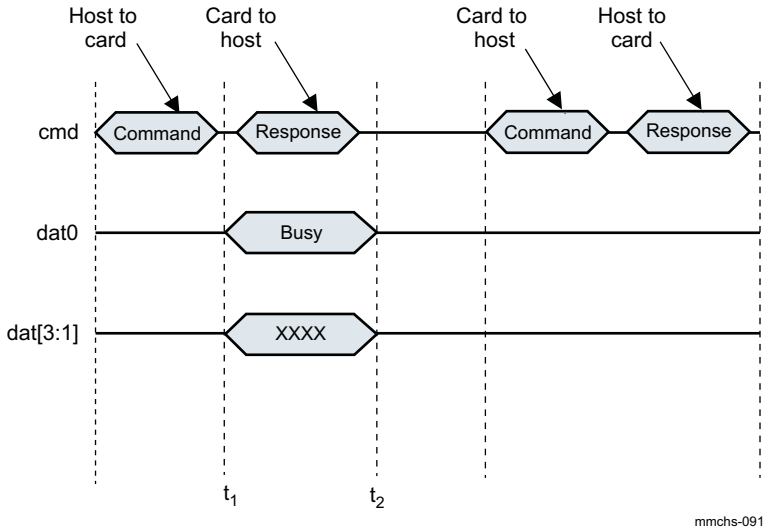
A MMCHS_STAT[20] DTO event can be asserted in the following conditions:

- Busy time-out for R1b, R5b response type
- Busy time-out after write CRC status
- Write CRC status time-out
- Read data time-out

19.4.9.1 Busy Time-Out for R1b, R5b Response Type

Figure 19-16 shows the DTO event condition asserted when there is a busy time-out for Rb1, R5b response.

Figure 19-16. Busy Time-Out for R1b, R5b Response Type



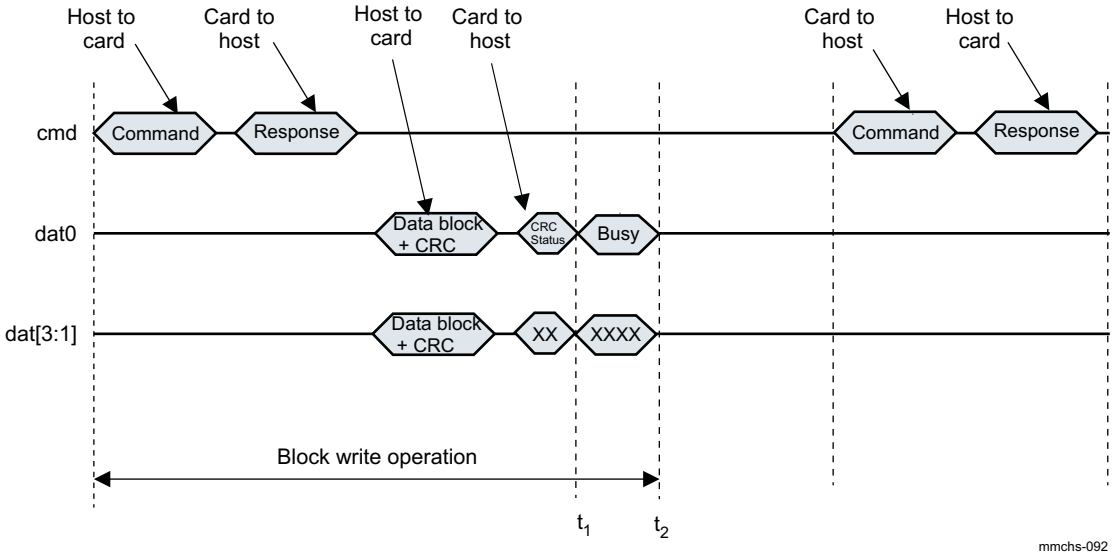
t_1 – Data time-out counter is loaded and starts after R1b, R5b response type.

t_2 – Data time-out counter stops and if it is 0, the MMCHS_STAT[20] DTO bit is generated.

19.4.9.2 Busy Time-Out After Write CRC Status

Figure 19-17 shows the DTO event condition asserted when there is a busy time-out after write CRC status.

Figure 19-17. Busy Time-Out After Write CRC Status



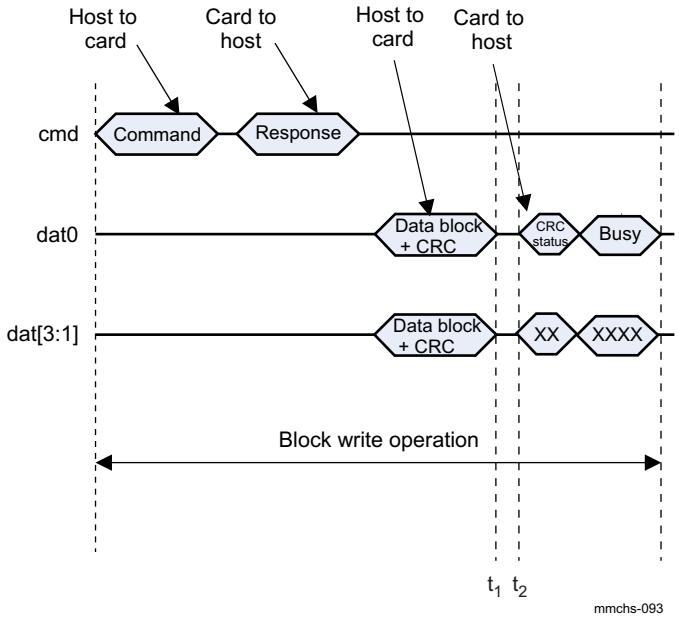
t_1 – Data time-out counter is loaded and starts after CRC status.

t_2 – Data time-out counter stops and if it is 0, the MMCHS_STAT[20] DTO bit is generated.

19.4.9.3 Write CRC Status Time-Out

Figure 19-18 shows the DTO event condition asserted when there is a write CRC status time-out.

Figure 19-18. Write CRC Status Time-Out



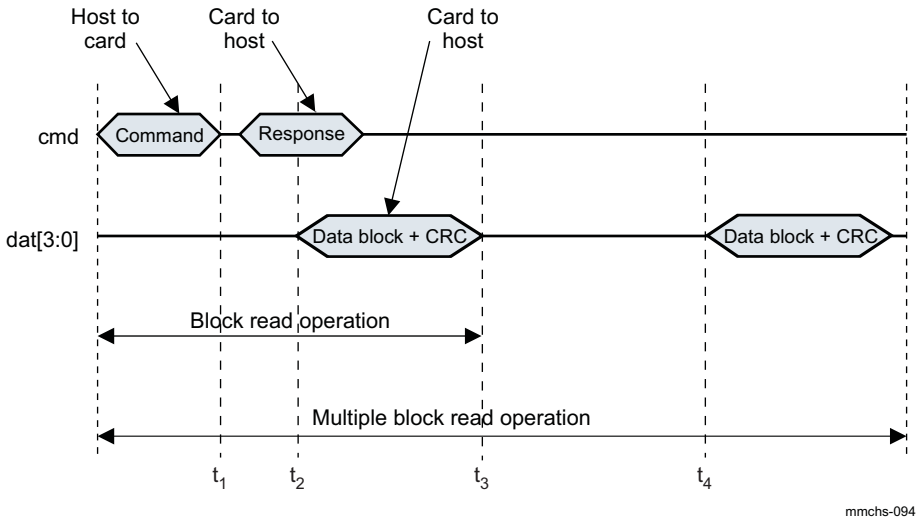
t_1 – Data time-out counter is loaded and starts after data block + CRC.

t_2 – Data time-out counter stops and if it is 0, the MMCHS_STAT[20] DTO bit is generated.

19.4.9.4 Read Data Time-Out

Figure 19-19 shows the DTO event condition asserted when there is a read data time-out.

Figure 19-19. Read Data Time-Out



t_1 – Data time-out counter is loaded and starts after command transmission.

t_2 – Data time-out counter stops and if it is 0, the MMCHS_STAT[20] DTO bit is generated.

t_3 – Data time-out counter is loaded and starts after data block + CRC transmission.

t_4 – Data time-out counter stops and if it is 0, the MMCHS_STAT[20] DTO bit is generated.

19.4.10 Transfer Stop

Whenever a transfer is initiated, the transmission can be stopped before it finishes. This could happen in case of multiple-block-oriented transfers (transfer length is known).

NOTE: Because the SDIO controller manages transfers based on a block granularity, the buffer accepts a block only if there is enough space to store it completely. Consequently, if a block is pending in the buffer, no command is sent to the card because the card clock will be shut off by the controller.

The SDIO controller includes two features that make a transfer stop more convenient and easier to manage:

- Auto CMD12 (for SD cards only):

  Auto CMD12 feature is enabled by setting the MMCHS_CMD[3:2] ACEN bit field to 0x1 (this setting is relevant for a SD transfer with a known number of blocks to transfer). When the Auto CMD12 feature is enabled, the SDIO controller automatically issues a CMD12 command when the expected number of blocks is exchanged.
- Stop at block gap:

  This feature is enabled by setting the MMCHS_HCTL[16] SBGR bit to 0x1. When enabled, this capability holds the transfer on until the end of a block boundary. If a stop transmission is needed, software can use this pause to send a CMD12 to the card.

NOTE: For SD cards, the stop-at-block-gap feature is not supported in read mode.

For SDIO cards, this setting can be supported in read mode if the card has read-wait capability.

Table 19-13 shows the common way to stop a transfer, indicating the command to send and the features to enable.

Table 19-13. SDIO Controller Transfer Stop Command Summary

		Write Transfer		Read Transfer	
		SD	SDIO	SD	SDIO
Single block		Transfer ends automatically. Wait TC.	Transfer ends automatically. Wait TC.	Transfer ends automatically. Wait TC.	Transfer ends automatically. Wait TC.
Multiblocks	Before the programmed block boundary	Send CMD12. Wait TC.	Send CMD52. Wait TC.	Send CMD12. Wait TC.	Send CMD52. Wait TC.
	Stop at the end of the transfer	Auto CMD12 active. Transfer ends automatically. Wait TC.	Set MMCHS_HCTL[16] SBGR bit to 0x1. Send CMD52. Wait TC.	Auto CMD12 active. Transfer ends automatically. Wait TC.	**If READ_WAIT supported** Stop at block gap. Wait TC.
					If READ_WAIT not supported Send CMD52. Wait TC.

NOTE: The SDIO controller sends the stop command to the card on a block boundary, regardless of when the command was written to the controller registers.

19.4.11 Output Signals Generation

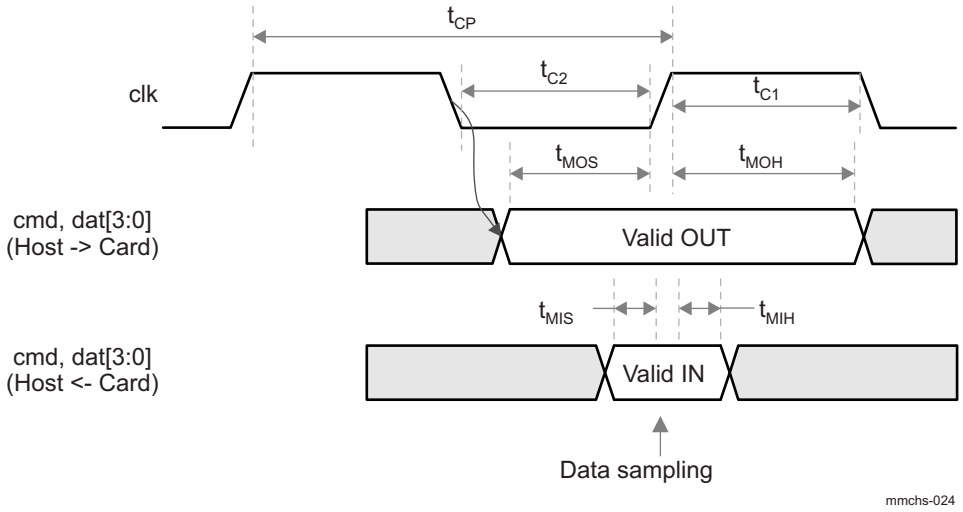
The SDIO output signals can be driven on the falling edge or rising edge, depending on the MMCHS_HCTL[2] HSPE bit.

19.4.11.1 Generation on Falling Edge of SDIO clock

The controller defaults to this mode to maximize hold timings. In this case, the MMCHS_HCTL[2] HSPE bit is set to 0.

Figure 19-20 shows the output signals of the module when generating from the falling edge of the SDIO clock.

Figure 19-20. Output Driven on Falling Edge

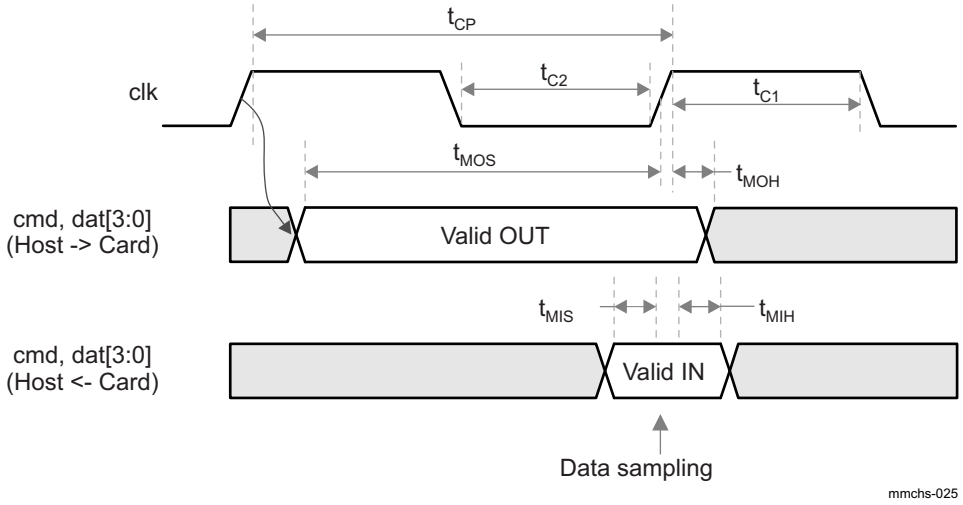


19.4.11.2 Generation on Rising Edge of SDIO clock

This mode is intended to increase setup timings. This feature is activated by setting the MMCHS_HCTL[2] HSPE bit to 1.

Figure 19-21 shows the output signals of the module when generating from the rising edge of the SDIO clock.

Figure 19-21. Output Driven on Rising Edge



19.4.12 Test Registers

Test registers are available to comply with the *SD Host Controller Specification*. This feature is useful to generate interrupts manually for driver debugging.

The force event register (MMCHS_FE) is used to control the error status and error interrupt status for Auto CMD12.

The system test register (MMCHS_SYSTEST) is used to control the signals that connect to I/O pins when the module is configured in the system test mode (the MMCHS_CON[4] MODE bit = 1) for boundary connectivity verification.

19.4.13 SDIO Hardware Status Features

Table 19-14 describes the SDIO hardware status features.

Table 19-14. SDIO Hardware Status Features

Feature	Type	Register/Bit Field	Description
Interrupt flags		See Section 19.4.4, *Interrupt Requests.*	
CMD line signal level	Status	MMCHS_PSTATE[24] CLEV	Indicates the level of the command line
DAT lines signal level	Status	MMCHS_PSTATE[23:20] DLEV	Indicates the level of the data lines
Buffer read enable	Status	MMCHS_PSTATE[11] BRE	Readable data exists in the buffer.
Buffer write enable	Status	MMCHS_PSTATE[10] BWE	Indicates whether there is enough space in the buffer to write BLEN bytes of data
Read transfer active	Status	MMCHS_PSTATE[9] RTA	Used to detect completion of a read transfer.
Write transfer active	Status	MMCHS_PSTATE[8] WTA	Indicates a write transfer active
Data line active	Status	MMCHS_PSTATE[2] DLA	Indicates whether the data lines are active
Command Inhibit (DAT lines)	Status	MMCHS_PSTATE[1] DATI	Indicates whether issuing of command using data lines is allowed. For example, commands with busy mechanism (that is, R1b response), data transfer commands.
Command inhibit (CMD line)	Status	MMCHS_PSTATE[0] CMDI	Indicates whether issuing of command using command line is allowed

Table 19-15 describes the SDIO preset value features.

Table 19-15. SDIO Preset Value Registers

Feature	Type	Register	Description
Preset value register	Status	MMCHS_PVINITSD	Preset Values for Initialization and Default Speed modes
Preset value register	Status	MMCHS_PVHSSDR12	Preset Values for High Speed and SDR12 speed modes
Preset value register	Status	MMCHS_PVSDR25SDR50	Preset Values for SDR25 speed mode

19.5 SDIO Programming Guide

19.5.1 Low-Level Programming Models

19.5.1.1 Global Initialization

19.5.1.1.1 Surrounding Modules Global Initialization

This section identifies the requirements for initializing the surrounding modules when the module must be used for the first time after a device reset. This initialization of surrounding modules is based on the integration and environment of the SDIO controller. For more information, see Section 19.3, *SDIO Integration*, and Section 19.2, *SDIO Environment*.Table 19-16 shows the global initialization of the surrounding modules.

Table 19-16. Global Initialization of Surrounding Modules

Surrounding Modules	Comments
PRCM	Module interface and functional clocks must be enabled. See Chapter 3, *Power, Reset, and Clock Management*.
Control module	Module-specific pad muxing and configuration must be set in the control module. See Chapter 13, *Control Module* .
DMA_CROSSBAR	DMA_CROSSBAR configuration must be done to allow module DREQs to be mapped to certain device DMA line. For more information see Section 13.4.6.5, *DMA_CROSSBAR Module Functional Description*, in Chapter 13, *Control Module*.
Device DMAs	Device DMAs configuration must be done to enable the module DMA channel requests.
IRQ_CROSSBAR	IRQ_CROSSBAR configuration must be done to allow module IRQs to be mapped to certain device INTC line. For more information see Section 13.4.6.4, *IRQ_CROSSBAR Module Functional Description*, in Chapter 13, *Control Module*
Device INTCs	Device INTCs must be configured to enable the interrupt request generation. For more information see Chapter 12 *Interrupt controller*

19.5.1.1.2 SDIO Host Controller Initialization Flow

Table 19-17 shows the SDIO controller general initialization steps.

Table 19-17. SDIO Controller Meta Initialization Steps

Step	Access Type	Register/Bit Field/Programming Model	Value
Initialize clocks.		See Section 19.5.1.1.2.1, *Enable Interface and Functional Clock for the SDIO Controller.*	
Software reset of the controller.		See Section 19.5.1.1.2.2, *SDIO Soft Reset Flow*.	
Set module hardware capabilities.		See Section 19.5.1.1.2.3,*Set SDIO Default Capabilities*.	
Set module idle and wake-up modes.		See Section 19.5.1.1.2.4, *Wake-Up Configuration*.	

19.5.1.1.2.1 Enable Interface and Functional Clock for the SDIO Controller

Before any access to the registers of the SDIO controller, its interface and functional clocks in the PRCM module registers must be enabled. See Section 3.6.4.1.4, *Clock Domain Module Attributes*, in Chapter 3, *Power, Reset, and Clock Management*.

19.5.1.1.2.2 SDIO Soft Reset Flow

Figure 19-22 shows the soft reset process of the SDIO controller.

Figure 19-22. SDIO Controller Software Reset Flow

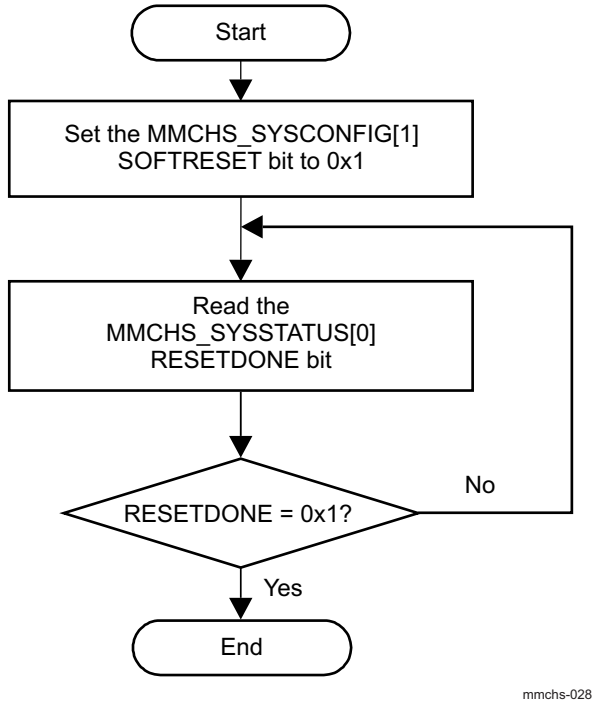


Table 19-18. Register Call Summary for Main Sequence – Software Reset Flow

Register Name	Register Name
MMCHS_SYSCONFIG	MMCHS_SYSSTATUS

19.5.1.1.2.3 Set SDIO Default Capabilities

Software must read capabilities and set (write) the MMCHS_CAPA[26:24] and MMCHS_CUR_CAPA[23:0] bit fields before the SDIO host driver is started.

19.5.1.1.2.4 Wake-Up Configuration

Table 19-19 describes the SDIO controller wake-up configuration.

Table 19-19. SDIO Controller Wake-Up Configuration

Step	Access Type	Register/Bit Field/Programming Model	Value
Configure wake-up bit (if necessary).	W	MMCHS_SYSCONFIG[2] ENAWAKEUP	0x1
Enable wake-up events on SD card interrupt (if necessary).	W	MMCHS_HCTL[24] IWE	0x1
SDIO card only: Enable card interrupt (if necessary).	W	MMCHS_IE[8] CIRQ_ENABLE	0x1

19.5.1.1.2.5 SDIO Host and Bus Configuration

Figure 19-23 shows the SDIO bus configuration process.

Figure 19-23. SDIO Controller Bus Configuration

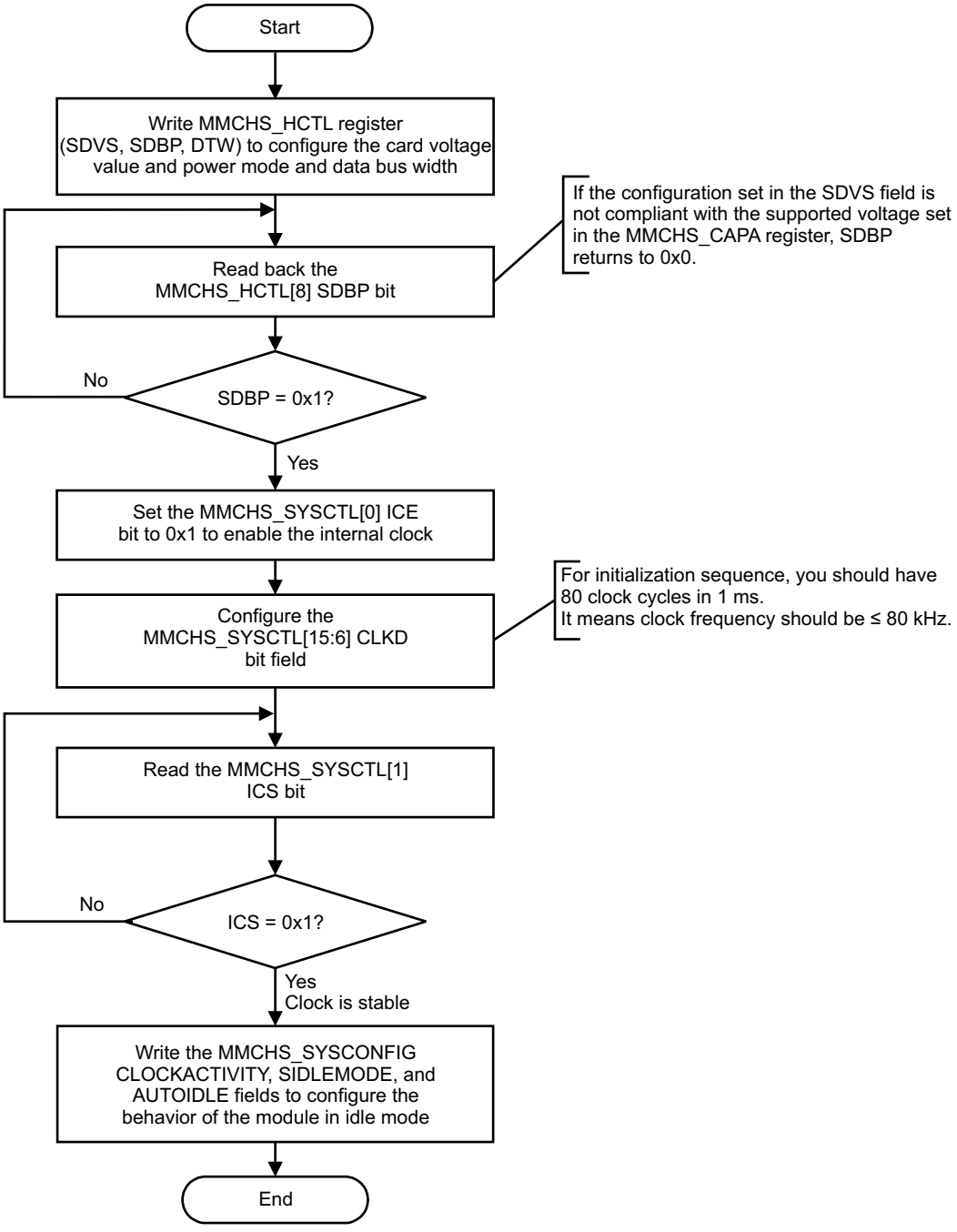


Table 19-20. Register Call Summary for Main Sequence – Bus Configuration

Register Name	Register Name
MMCHS_CON	MMCHS_HCTL
MMCHS_SYSCONFIG	MMCHS_SYSCTL

19.5.1.2 Operational Modes Configuration

19.5.1.2.1 Basic Operations for SDIO Host Controller

The SDIO host controller performs data transfers: data to card (referred to as write transfers) and data from card (referred to as read transfers).

The host controller requires transfers to run on a block-by-block basis rather than on a DMA burst size basis. A single DMA request (or block request interrupt) is signaled for each block. Pipelining is supported as long as the block size is less than one half of the memory buffer size.

19.5.1.2.1.1 Card Identification and Selection

Figure 19-24 and Figure 19-25 show the card identification and selection process.

Figure 19-24. SDIO Controller Card Identification and Selection – Part 1

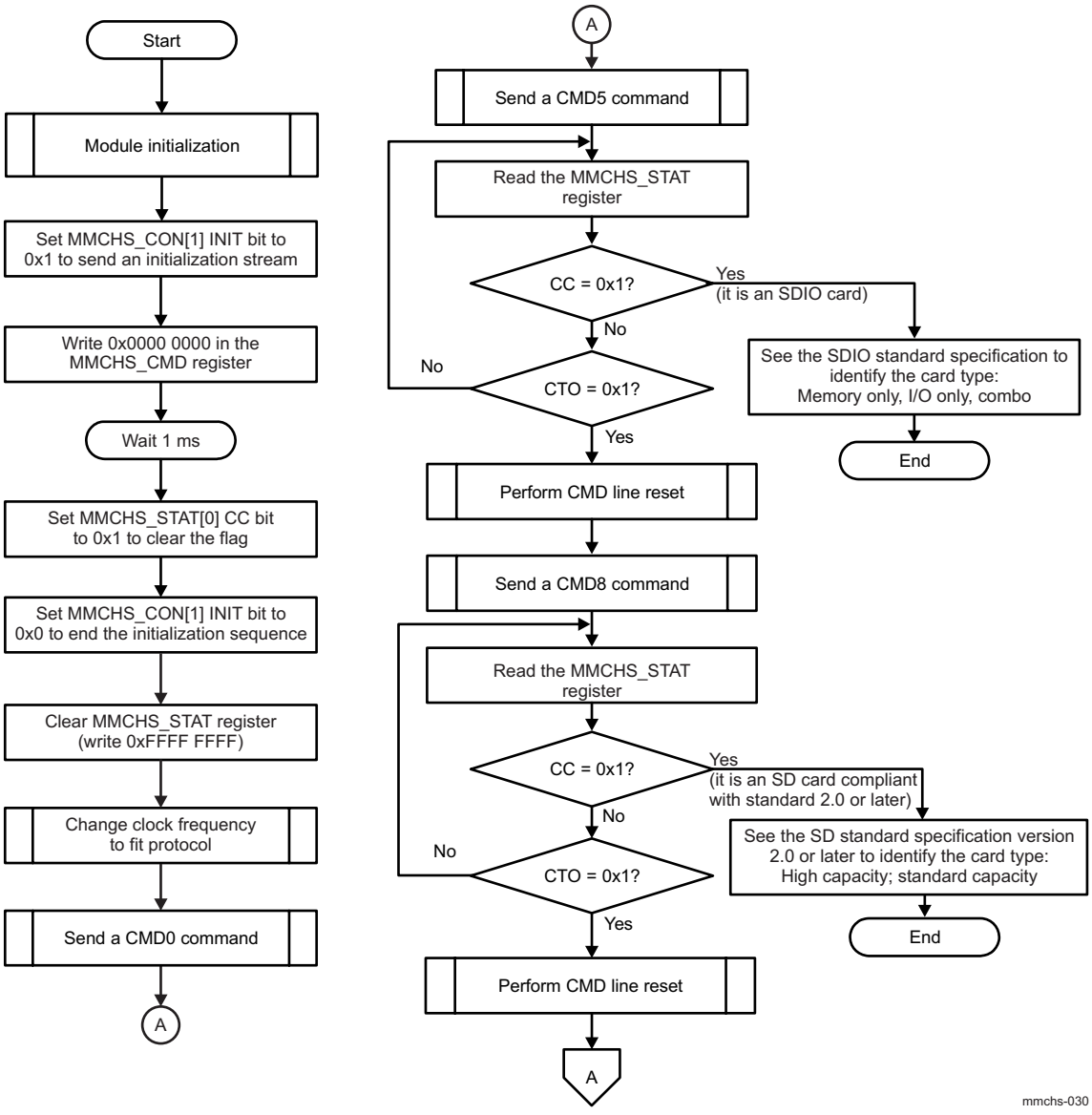


Figure 19-25. SDIO Controller Card Identification and Selection – Part 2

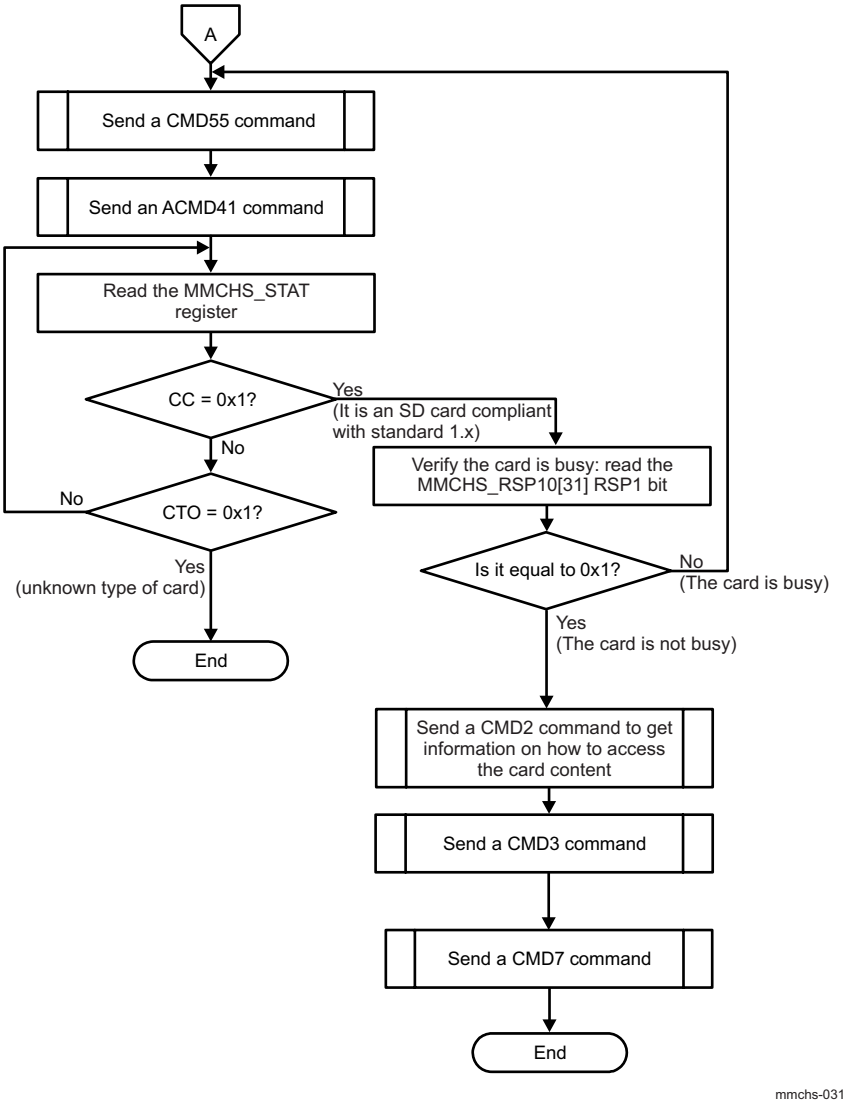


Table 19-21. Register Call Summary for Main Sequence – Card Identification and Selection

Register Name	Register Name	Register Name
MMCHS_CON	MMCHS_CMD	MMCHS_STAT
MMCHS_SYSCTL	MMCHS_RSP10	

Table 19-22 lists the subprocess call summary.

Table 19-22. Subprocess Call Summary for Main Sequence – Card Identification and Selection

Subprocess Name	Cross-Reference
Initialize module.	See Section 19.5.1.1.2 , *SDIO Host Controller Initialization Flow*.
Change clock frequency to fit protocol.	See Section 19.5.1.2.1.6.2, *SDIO Controller Clock Frequency Change*.
Send a command.	See Section 19.5.1.2.1.6.1, *Command Transfer Flow*.
Perform CMD line reset.	See Section 19.5.1.2.1.1.1, *CMD Line Reset Procedure*.

19.5.1.2.1.1 CMD Line Reset Procedure

Table 19-23 lists the CMD line reset.

Table 19-23. CMD Line Reset

Step	Access Type	Register/Bit Field/Programming Model	Value
Initiate CMD line reset.	W	MMCHS_SYSCTL[25] SRC	0x1
Poll the SRC bit until it is set to 0x1.	R	MMCHS_SYSCTL[25] SRC	= 0x1
Wait until the SRC bit returns to 0x0 (reset procedure is completed).	R	MMCHS_SYSCTL[25] SRC	= 0x0

19.5.1.2.1.2 Read/Write Transfer Flow in DMA Mode With Interrupt

Figure 19-26 shows the read and write protocol in DMA slave mode with interrupt signaling.

Figure 19-26. SDIO Controller Read/Write Transfer Flow in DMA Slave Mode With interrupt

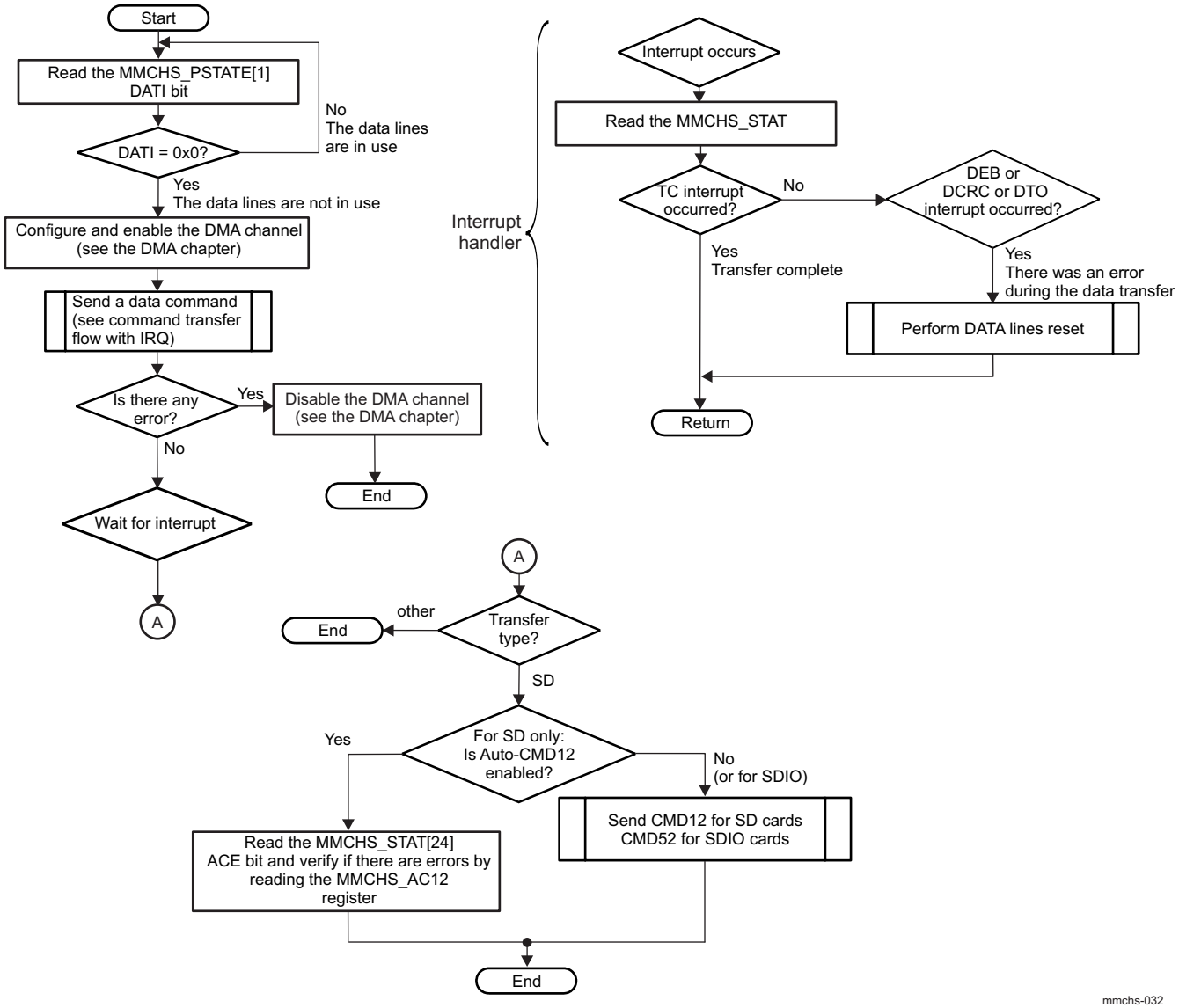


Table 19-24. Register Call Summary for Main Sequence – Read/Write Transfer Flow in DMA Mode With interrupt

Register Name	Register Name
MMCHS_PSTATE	MMCHS_STAT
MMCHS_SYSCTL	MMCHS_CMD

Table 19-25. Subprocess Call Summary for Main Sequence – SDIO Controller Read/Write Transfer Flow in DMA Mode With Interrupt

Subprocess Name	Cross-Reference
Send a data command.	See Figure 19-32.
Perform data lines reset.	See Section 19.5.1.2.1.2.1, *DATA Lines Reset Procedure*.

19.5.1.2.1.2.1 DATA Lines Reset Procedure

Table 19-26 describes the data lines reset.

Table 19-26. DATA Lines Reset

Step	Access Type	Register/Bit Field/Programming Model	Value
Initiate data lines reset.	W	MMCHS_SYSCTL[26] SRD	0x1
Poll the SRD bit until it is set to 0x1.	R	MMCHS_SYSCTL[26] SRD	= 0x1
Wait until the SRD bit returns to 0x0 (reset procedure is complete).	R	MMCHS_SYSCTL[26] SRD	= 0x0

19.5.1.2.1.3 Read/Write Transfer Flow in DMA Mode With Polling

Figure 19-27 shows the read and write protocol in DMA mode.

Figure 19-27. SDIO Controller Read/Write Transfer Flow in DMA Mode With Polling

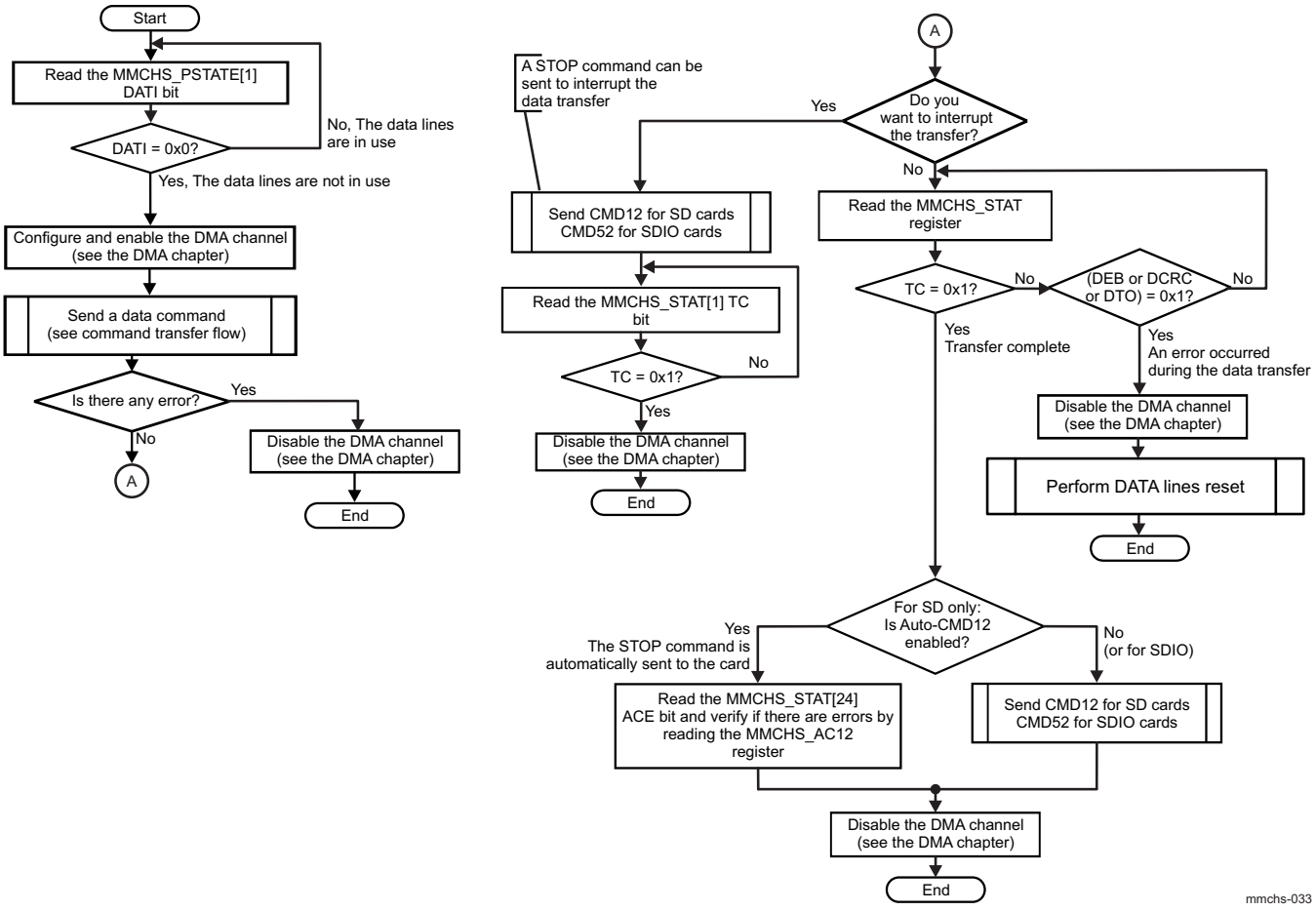


Table 19-27. Register Call Summary for Main Sequence – Read/Write Transfer Flow in DMA Mode With Polling

Register Name	Register Name	Register Name
MMCHS_PSTATE	MMCHS_STAT	MMCHS_SYSCTL
MMCHS_CMD	MMCHS_AC12	

Table 19-28. Subprocess Call Summary for Main Sequence – Read/Write Transfer Flow in DMA Mode With Polling

Subprocess Name	Cross-Reference
Send command.	See Section 19.5.1.2.1.6.1, *Command Transfer Flow*.
Perform data lines reset.	See Section 19.5.1.2.1.2.1, *DATA Lines Reset Procedure*.

19.5.1.2.1.4 Read/Write Transfer Flow Without DMA With Polling

Figure 19-28 shows a read/write transfer without using the DMA and with polling.

Figure 19-28. SDIO Controller Read/Write Transfer Flow Without DMA and With Polling

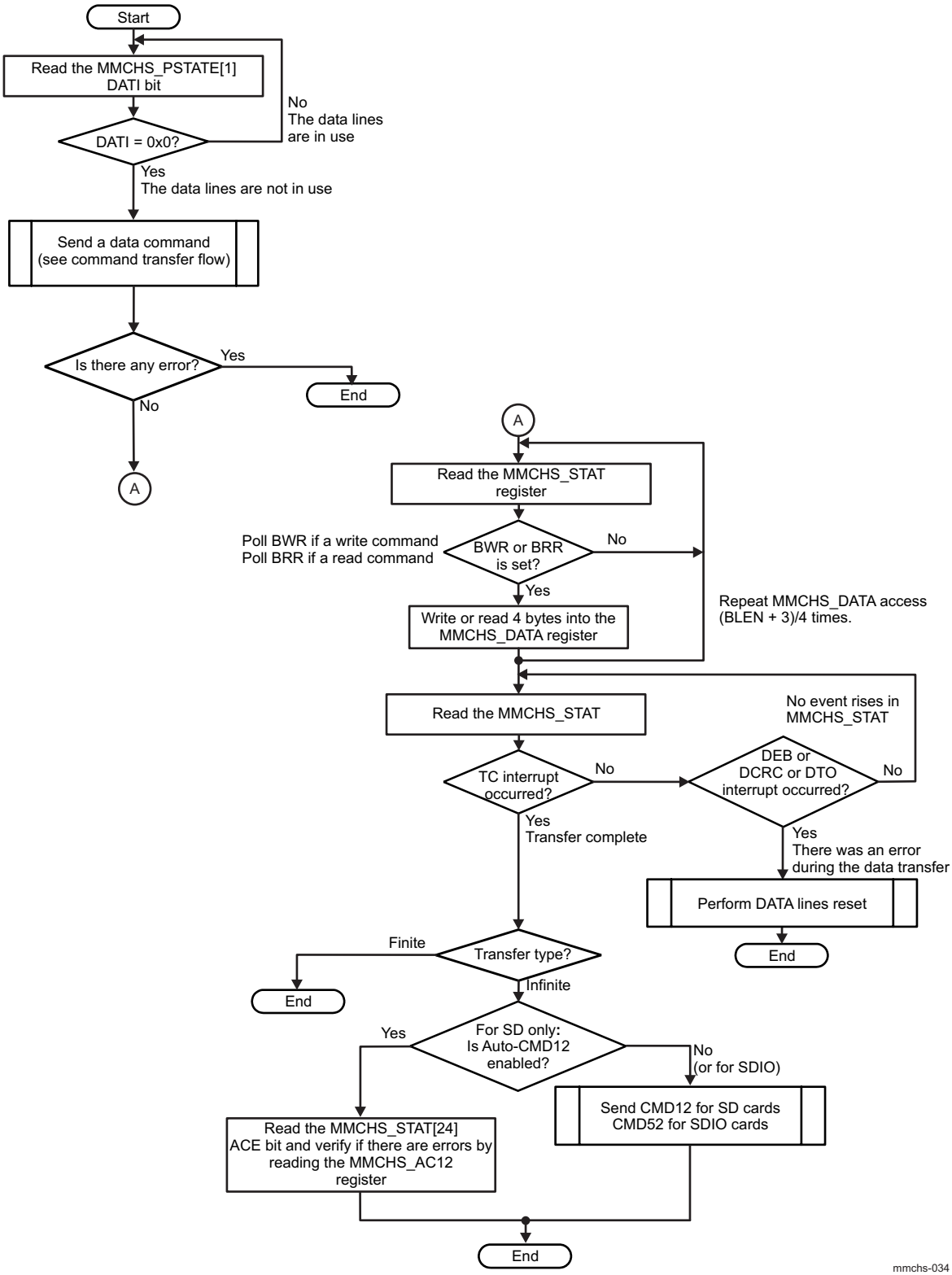


Table 19-29. Register Call Summary for Main Sequence – Read/Write Transfer Flow Without DMA With Polling

Register Name	Register Name	Register Name
MMCHS_PSTATE	MMCHS_DATA	MMCHS_STAT
MMCHS_SYSCTL	MMCHS_CMD	MMCHS_AC12

Table 19-30. Subprocess Call Summary for Main Sequence – Read/Write Transfer Flow Without DMA With Polling

Subprocess Name	Cross-Reference
Send data command.	See Section 19.5.1.2.1.6.1, *Command Transfer Flow*.
Perform data lines reset.	See Section 19.5.1.2.1.2.1, *DATA Lines Reset Procedure*.

19.5.1.2.1.5 Suspend-Resume Flow

The suspend-and-resume feature is supported only by SDIO cards.

19.5.1.2.1.5.1 Suspend Flow

Figure 19-29 shows the suspend flow for SDIO cards.

Figure 19-29. SDIO Controller Suspend Flow

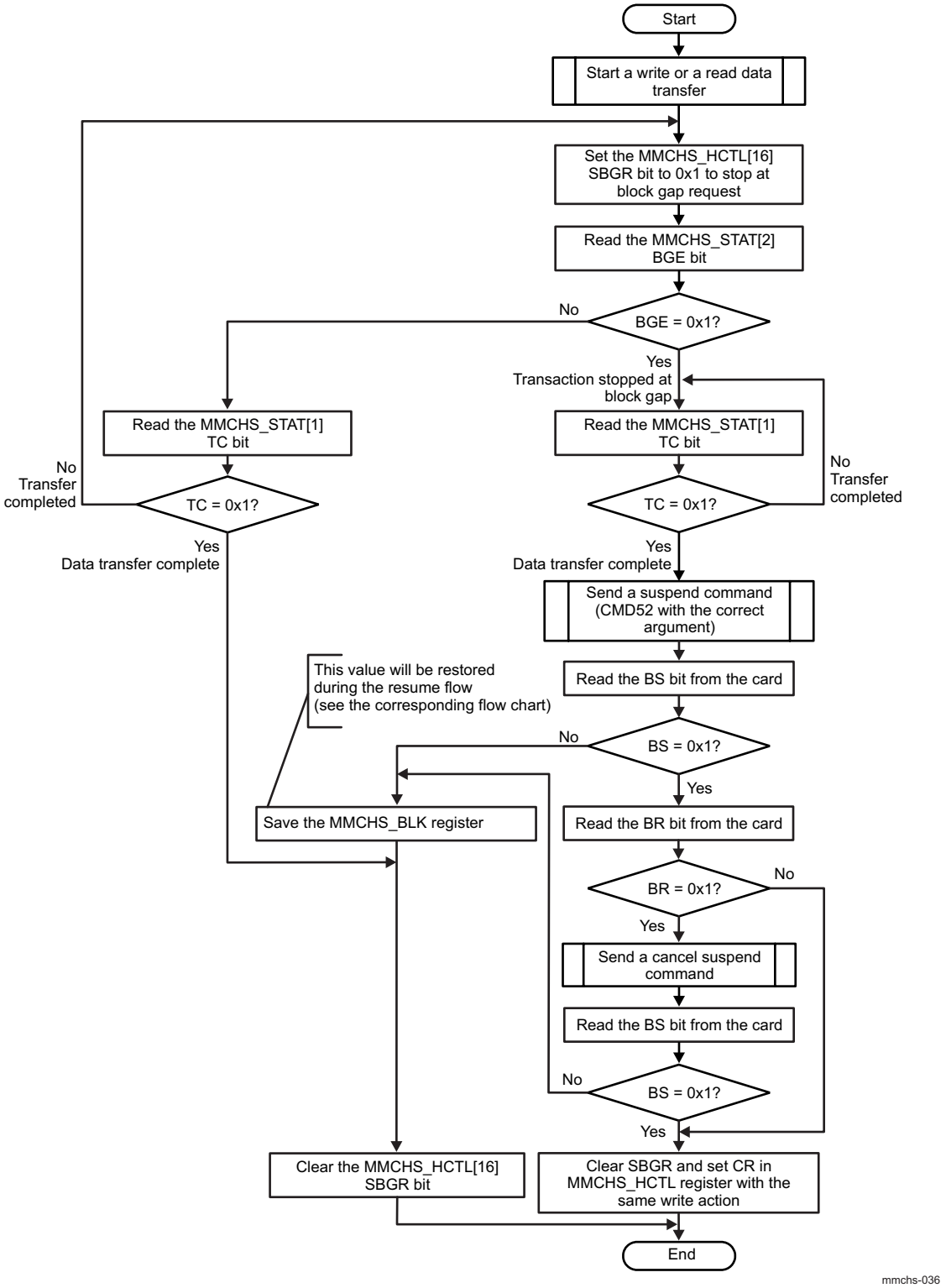


Table 19-31. Register Call Summary for Main Sequence – Suspend Flow

Register Name	Register Name	Register Name
MMCHS_HCTL	MMCHS_STAT	MMCHS_BLK

Table 19-32. Subprocess Call Summary for Main Sequence – Suspend Flow

Subprocess Name	Cross-Reference
Start a write or a read data transfer.	See Section 19.5.1.2.1, *Basic Operations for SDIO Host Controller*.
Send a suspend command (CMD52 with the correct argument).	See Section 19.5.1.2.1.6.1, *Command Transfer Flow*.
Send a cancel suspend command.	See Section 19.5.1.2.1.6.1, *Command Transfer Flow*.

19.5.1.2.1.5.2 Resume Flow

Figure 19-30 shows the resume flow for SDIO cards.

Figure 19-30. SDIO Controller Resume Flow

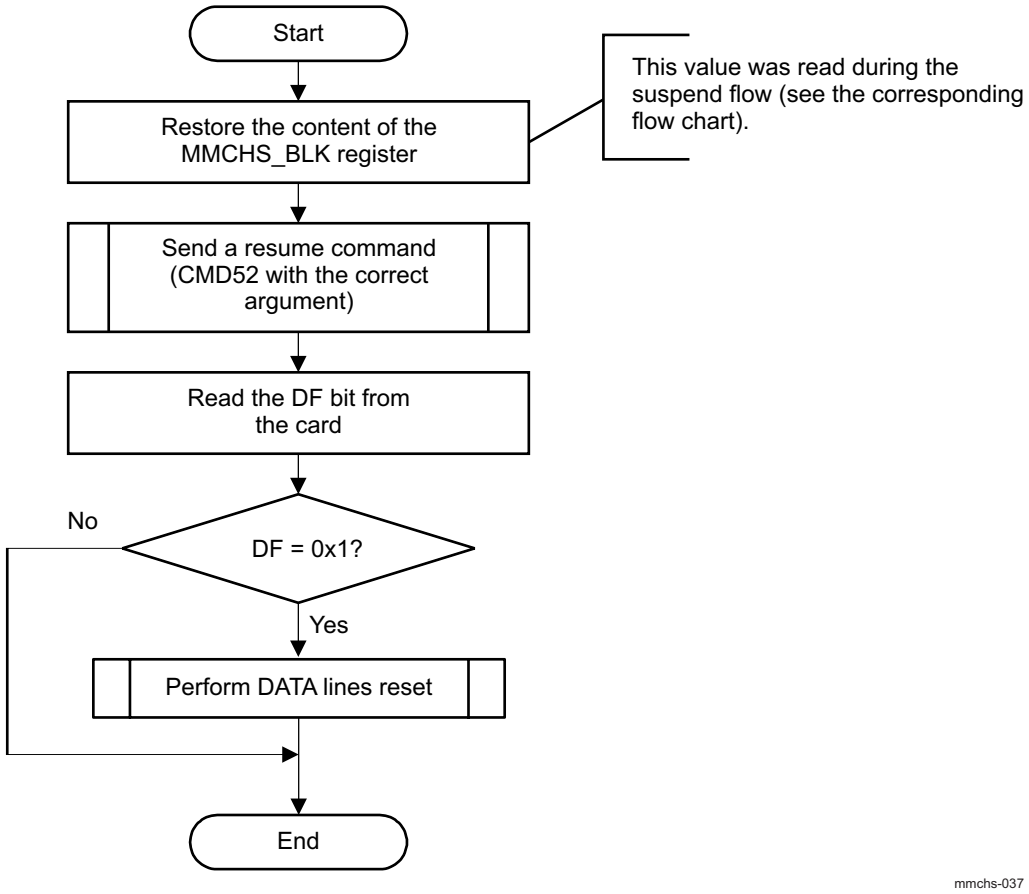


Table 19-33. Register Call Summary for Main Sequence - Resume Flow

Register Name	Register Name
MMCHS_BLK	MMCHS_SYSCTL

Table 19-34. Subprocess Call Summary for Main Sequence - Resume Flow

Subprocess Name	Cross-Reference
Send a resume command (CMD52 with the correct argument).	See Section 19.5.1.2.1.6.1, *Command Transfer Flow*.
Perform data lines reset.	See Section 19.5.1.2.1.2.1, *DATA Lines Reset Procedure*.

19.5.1.2.1.6 Basic Operations – Steps Detailed

19.5.1.2.1.6.1 Command Transfer Flow

Figure 19-31 shows how to send a command to the card using polling instead of interrupts for event signaling.

Figure 19-31. SDIO Controller Command Transfer Flow With Polling

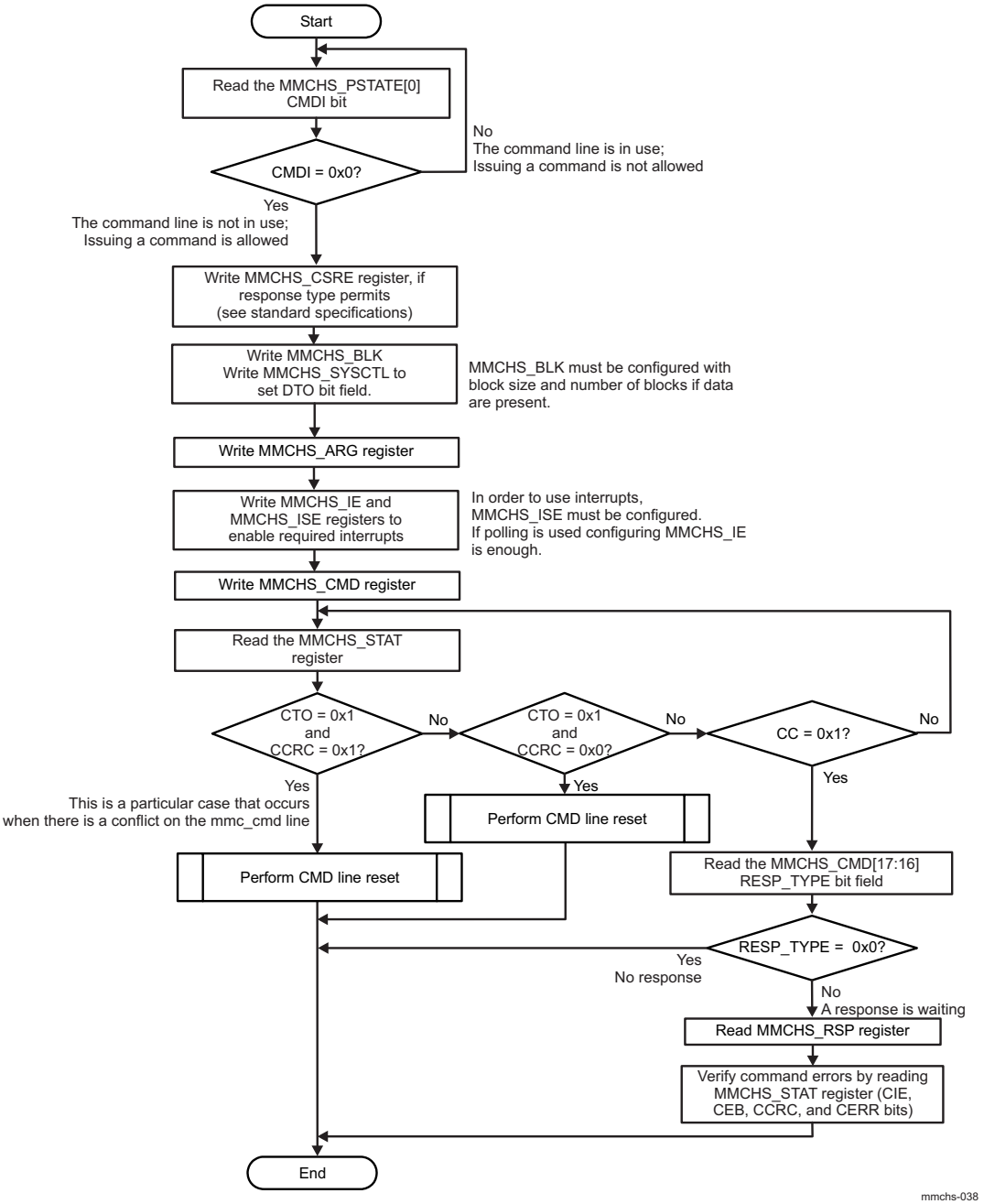


Table 19-35. Register Call Summary for Main Sequence – Command Transfer Flow With Polling

Register Name	Register Name	Register Name
MMCHS_PSTATE	MMCHS_CON	MMCHS_CSRE
MMCHS_STAT	MMCHS_BLK	MMCHS_SYSCTL

Table 19-35. Register Call Summary for Main Sequence – Command Transfer Flow With Polling (continued)

Register Name	Register Name	Register Name
MMCHS_ARG	MMCHS_IE	MMCHS_CMD
MMCHS_RSP10	MMCHS_RSP32	MMCHS_RSP54
MMCHS_RSP76		

Table 19-36. Subprocess Call Summary for Main Sequence – Command Transfer Flow With Polling

Subprocess Name	Cross-Reference
Perform CMD line reset.	See Section 19.5.1.2.1.1.1, *CMD Line Reset Procedure*.

Figure 19-32 shows how to send a command to the card using interrupts for event signaling.

Figure 19-32. SDIO Controller Command Transfer Flow With interrupts

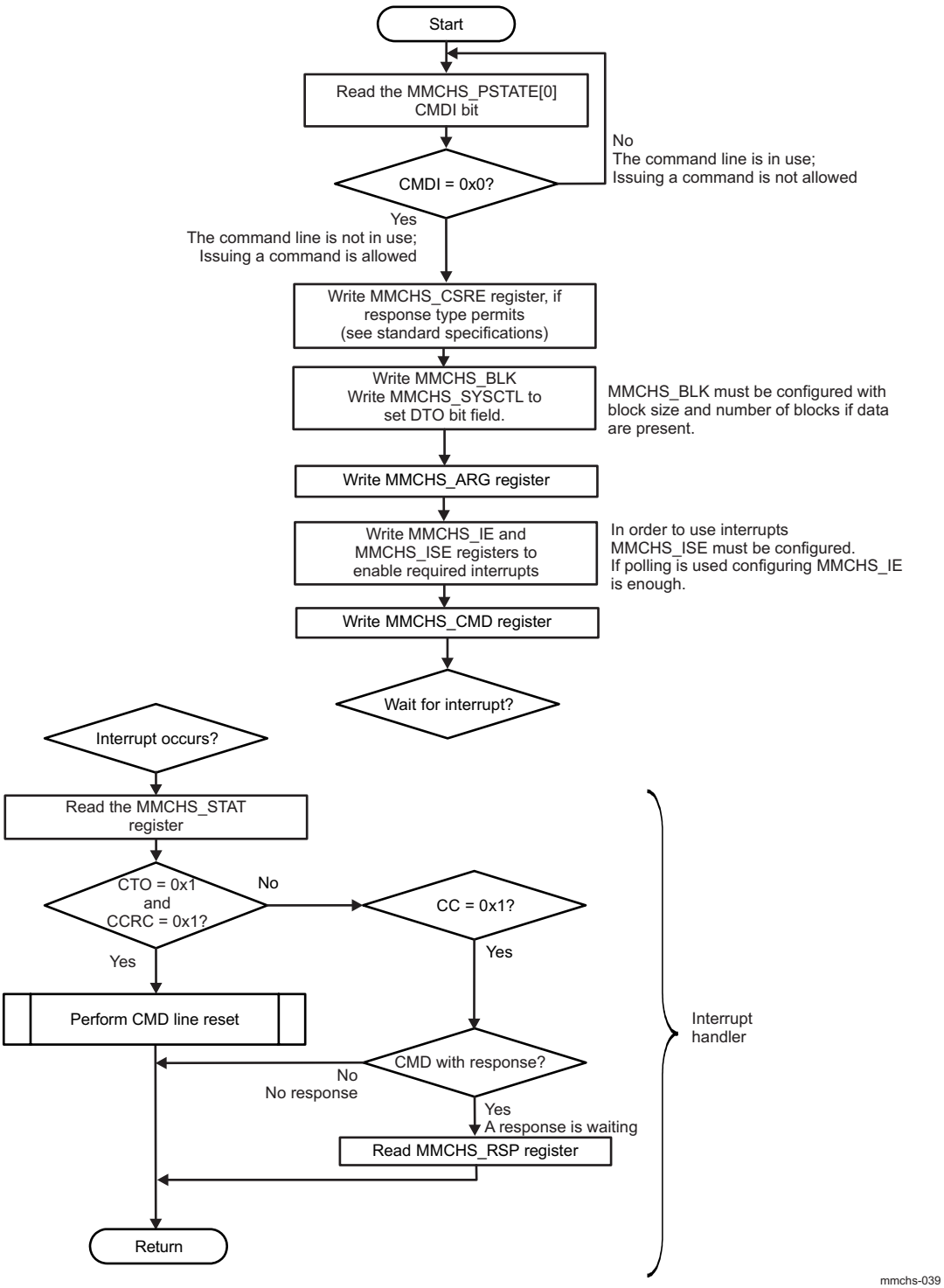


Table 19-37. Register Call Summary for Main Sequence – Command Transfer Flow With Interrupts

Register Name	Register Name	Register Name
MMCHS_PSTATE	MMCHS_CON	MMCHS_CSRE
MMCHS_STAT	MMCHS_BLK	MMCHS_SYSCTL

Table 19-37. Register Call Summary for Main Sequence – Command Transfer Flow With Interrupts (continued)

Register Name	Register Name	Register Name
MMCHS_ARG	MMCHS_IE	MMCHS_ISE
MMCHS_RSP10	MMCHS_RSP32	MMCHS_RSP54
MMCHS_RSP76	MMCHS_CMD	

Table 19-38. Subprocess Call Summary for Main Sequence – Command Transfer Flow With Interrupts

Subprocess Name	Cross-Reference
Perform CMD line reset.	See Section 19.5.1.2.1.1.1, *CMD Line Reset Procedure*.

19.5.1.2.1.6.2 SDIO Controller Clock Frequency Change

Figure 19-33 shows the different steps that allow changing the SDIO output clock frequency.

Figure 19-33. SDIO Controller Clock Frequency Change Flow

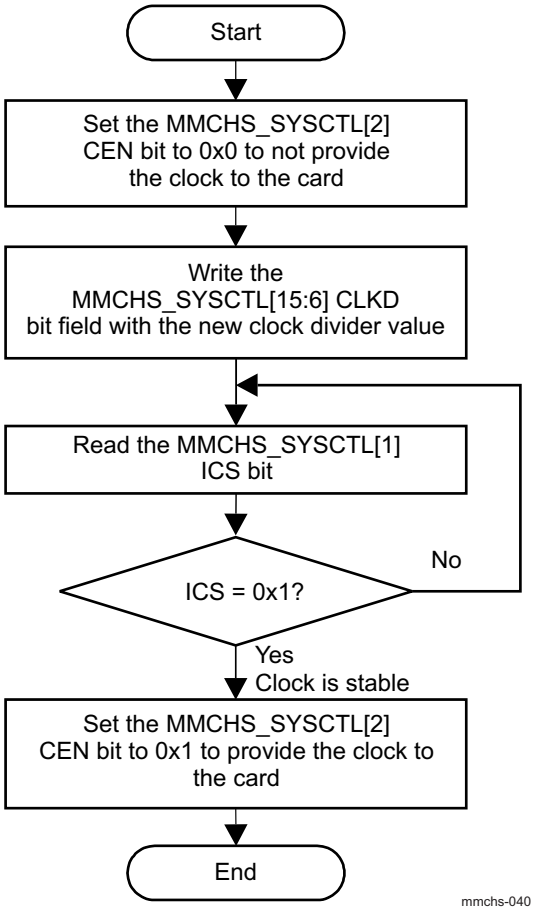


Table 19-39. Register Call Summary for Main Sequence – Clock Frequency Change Flow

Register Name
MMCHS_SYSCTL

19.6 SDIO Register Manual

19.6.1 SDIO Instance Summary

Table 19-40. SDIO Instance Summary

Module Name	Module Base Address	Size
MMC	0x480D 1000	4 KiB

19.6.2 SDIO Registers

19.6.2.1 SDIO Register Summary

Table 19-41 lists the SDIO registers.

Table 19-41. SDIO Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	MMC Base Address
MMCHS_HL_REV	R	32	0x0000 0000	0x480D 1000
MMCHS_HL_HWINFO	R	32	0x0000 0004	0x480D 1004
MMCHS_HL_SYSCONFIG	RW	32	0x0000 0010	0x480D 1010
MMCHS_SYSCONFIG	RW	32	0x0000 0110	0x480D 1110
MMCHS_SYSSTATUS	R	32	0x0000 0114	0x480D 1114
MMCHS_CSRE	RW	32	0x0000 0124	0x480D 1124
MMCHS_SYSTEST	RW	32	0x0000 0128	0x480D 1128
MMCHS_CON	RW	32	0x0000 012C	0x480D 112C
MMCHS_PWCNT	RW	32	0x0000 0130	0x480D 1130
RESERVED	R	32	0x0000 0134	0x480D 1134
MMCHS_SDMASA	RW	32	0x0000 0200	0x480D 1200
MMCHS_BLK	RW	32	0x0000 0204	0x480D 1204
MMCHS_ARG	RW	32	0x0000 0208	0x480D 1208
MMCHS_CMD	RW	32	0x0000 020C	0x480D 120C
MMCHS_RSP10	R	32	0x0000 0210	0x480D 1210
MMCHS_RSP32	R	32	0x0000 0214	0x480D 1214
MMCHS_RSP54	R	32	0x0000 0218	0x480D 1218
MMCHS_RSP76	R	32	0x0000 021C	0x480D 121C
MMCHS_DATA	RW	32	0x0000 0220	0x480D 1220
MMCHS_PSTATE	R	32	0x0000 0224	0x480D 1224
MMCHS_HCTL	RW	32	0x0000 0228	0x480D 1228
MMCHS_SYSCTL	RW	32	0x0000 022C	0x480D 122C
MMCHS_STAT	RW	32	0x0000 0230	0x480D 1230
MMCHS_IE	RW	32	0x0000 0234	0x480D 1234
MMCHS_ISE	RW	32	0x0000 0238	0x480D 1238
MMCHS_AC12	RW	32	0x0000 023C	0x480D 123C
MMCHS_CAPA	RW	32	0x0000 0240	0x480D 1240
MMCHS_CAPA2	R	32	0x0000 0244	0x480D 1244
MMCHS_CUR_CAPA	RW	32	0x0000 0248	0x480D 1248
MMCHS_FE	W	32	0x0000 0250	0x480D 1250
RESERVED	R	32	0x0000 0254	0x480D 1254
RESERVED	R	32	0x0000 0258	0x480D 1258
MMCHS_PVINITSD	R	32	0x0000 0260	0x480D 1260

Table 19-41. SDIO Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	MMC Base Address
MMCHS_PVHSSDR12	R	32	0x0000 0264	0x480D 1264
MMCHS_PVSDR25SDR50	R	32	0x0000 0268	0x480D 1268
RESERVED	R	32	0x0000 026C	0x480D 126C
MMCHS_REV	R	32	0x0000 02FC	0x480D 12FC

19.6.2.2 SDIO Register Description

Table 19-42. MMCHS_HL_REV

Address Offset	0x0000 0000		
Physical Address	0x480D 1000	**Instance**	MMC
Description	IP Revision Identifier (X.Y.R) Used by software to track features, bugs, and compatibility		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REVISION																															

Bits	Field Name	Description	Type	Reset
31:0	REVISION	IP Revision	R	TI internal data

Table 19-43. Register Call Summary for Register MMCHS_HL_REV

SDIO Register Manual
- SDIO Register Summary: [0]

Table 19-44. MMCHS_HL_HWINFO

Address Offset	0x0000 0004		
Physical Address	0x480D 1004	**Instance**	MMC
Description	Information about the IP module's hardware configuration.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								RETMODE	MEM_SIZE					MERGE_MEM	MADMA_EN

Bits	Field Name	Description	Type	Reset
31:7	RESERVED		R	0x-
6	RETMODE	Retention Mode generic parameter This bit field indicates whether the retention mode is supported using the pin PIRFFRET. Read 0x1: Retention mode enabled Read 0x0: Retention mode disabled	R	0x-

Bits	Field Name	Description	Type	Reset
5:2	MEM_SIZE	Memory size for FIFO buffer: Read 0x2: Memory of 1024 bytes, max block length is 1024 bytes Read 0x1: Memory of 512 bytes, max block length is 512 bytes Read 0x8: Memory of 4096 bytes, max block length is 2048 bytes Read 0x4: Memory of 2048 bytes, max block length is 2048 bytes	R	0x-
1	MERGE_MEM	Memory merged for FIFO buffer: This register defines the configuration of FIFO buffer architecture. If the bit is set STA and DFT shall support clock multiplexing and balancing. Read 0x1: A single memory is used with multiplexed addresses, data and clocks. Read 0x0: 2 memories instantiated, one per data transfer direction.	R	0x-
0	MADMA_EN	Master DMA enabled generic parameter: This register defines the configuration of the controller to know if it supports the master DMA management called ADMA. Read 0x1: Controller supports ADMA Read 0x0: No Master DMA (ADMA) management supported	R	0x-

Table 19-45. Register Call Summary for Register MMCHS_HL_HWINFO

SDIO Functional Description • Data Buffer: [0]
SDIO Register Manual • SDIO Register Summary: [1]

Table 19-46. MMCHS_HL_SYSCONFIG

Address Offset	0x0000 0010		
Physical Address	0x480D 1010	**Instance**	MMC
Description	Clock Management Configuration Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												IDLEMODE		FREEEMU	SOFTRESET

Bits	Field Name	Description	Type	Reset
31:4	RESERVED		R	0x2
3:2	IDLEMODE	Configuration of the local target state management mode. By definition, target can handle read/write transaction as long as it is out of IDLE state. 0x0: Force-idle mode: local target's idle state follows (acknowledges) the system's IDLE requests unconditionally, i.e. regardless of the IP module's internal requirements.Backup mode, for debug only. 0x1: No-idle mode: local target never enters idle state.Backup mode, for debug only. 0x3: Smart-idle wakeup-capable mode: local target's idle state eventually follows (acknowledges) the system's IDLE requests, depending on the IP module's internal requirements.IP module may generate (IRQ- or DMA-request-related) wakeup events when in idle state.Mode is only relevant if the appropriate IP module "swakeup" output(s) is (are) implemented. 0x2: Smart-idle mode: local target's idle state eventually follows (acknowledges) the system's IDLE requests, depending on the IP module's internal requirements.IP module shall not generate (IRQ- or DMA-request-related) wakeup events.	RW	0x2
1	FREEEMU	Sensitivity to emulation (debug) suspend input signal. Functionality NOT implemented in MMCHS. 0x0: IP module is sensitive to emulation suspend 0x1: IP module is not sensitive to emulation suspend	RW	0
0	SOFTRESET	Software reset. (Optional) Write 0x0: No action Write 0x1: Initiate software reset Read 0x1: Reset (software or other) ongoing Read 0x0: Reset done, no pending action	RW	0

Table 19-47. Register Call Summary for Register MMCHS_HL_SYSCONFIG

SDIO Register Manual

- SDIO Register Summary: [0]

Table 19-48. MMCHS_SYSCONFIG

Address Offset	0x0000 0110		
Physical Address	0x480D 1110	**Instance**	MMC
Description	System Configuration Register This register allows controlling various parameters of the Interconnect interface.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						CLOCKACTIVITY		RESERVED			SIDLEMODE		ENAWAKEUP	SOFTRESET	AUTOIDLE

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x0
9:8	CLOCKACTIVITY	Clocks activity during wake up mode period. Bit8: Interface clock Bit9: Functional clock 0x0: Interface and Functional clock may be switched off. 0x1: Interface clock is maintained. Functional clock may be switched-off. 0x3: Interface and Functional clocks are maintained. 0x2: Functional clock is maintained. Interface clock may be switched-off.	RW	0x0
7:5	RESERVED	This bit is initialized to zero, and writes to it are ignored. Reads return 0.	R	0x0
4:3	SIDLEMODE	Power management 0x0: If an IDLE request is detected, the SDIO controller acknowledges it unconditionally and goes in Inactive mode. Interrupt and DMA requests are unconditionally de-asserted. 0x1: If an IDLE request is detected, the request is ignored and the module keeps on behaving normally. 0x3: Smart-idle wakeup-capable mode: local target's idle state eventually follows (acknowledges) the system's IDLE requests, depending on the IP module's internal requirements.IP module may generate (IRQ- or DMA-request-related) wakeup events when in idle state.Mode is only relevant if the appropriate IP module "swakeup" output(s) is (are) implemented. 0x2: Smart-idle mode: local target's idle state eventually follows (acknowledges) the system's IDLE requests, depending on the IP module's internal requirements.IP module shall not generate (IRQ- or DMA-request-related) wakeup events.	RW	0x2
2	ENAWAKEUP	Wakeup feature control 0x0: Wakeup capability is disabled 0x1: Wakeup capability is enabled	RW	1
1	SOFTRESET	Software reset. The bit is automatically reset by the hardware. During reset, it always returns 0. Write 0x0: No effect. Write 0x1: Trigger a module reset. Read 0x1: The module is reset. Read 0x0: Normal mode	RW	0
0	AUTOIDLE	Internal Clock gating strategy 0x0: Clocks are free-running 0x1: Automatic clock gating strategy is applied, based on the Interconnect and SDIO interface activity	RW	1

Table 19-49. Register Call Summary for Register MMCHS_SYSCONFIG

SDIO Functional Description • Software Reset: [0] • Power Management: [1][2][3][4][5][6][7][8][9][10][11]
SDIO Programming Guide • Global Initialization: [12][13][14]
SDIO Register Manual • SDIO Register Summary: [15] • SDIO Register Description: [16][17][18]

Table 19-50. MMCHS_SYSSTATUS

Address Offset	0x0000 0114		
Physical Address	0x480D 1114	**Instance**	MMC
Description	System Status Register This register provides status information about the module excluding the interrupt status information		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															RESETDONE

Bits	Field Name	Description	Type	Reset
31:1	RESERVED		R	0x0000 0000
0	RESETDONE	Internal Reset Monitoring Note: the debounce clock , the system clock (Interface) and the functional clock shall be provided to the SDIO host controller to allow the internal reset monitoring. Read 0x1: Reset completed. Read 0x0: Internal module reset is on-going	R	0

Table 19-51. Register Call Summary for Register MMCHS_SYSSTATUS

SDIO Functional Description
- Hardware Reset: [0]
- Software Reset: [1]

SDIO Programming Guide
- Global Initialization: [2]

SDIO Register Manual
- SDIO Register Summary: [3]

Table 19-52. MMCHS_CSRE

Address Offset	0x0000 0124		
Physical Address	0x480D 1124	**Instance**	MMC
Description	Card Status Response Error This register enables the host controller to detect card status errors of response type R1, R1b for all cards and of R5, R5b and R6 response for cards types SD or SDIO. When a bit MMCHS_CSRE[i] is set to 1, if the corresponding bit at the same position in the response MMCHS_RSP0[i] is set to 1, the host controller indicates a card error (MMCHS_STAT[CERR]) interrupt status to avoid the host driver reading the response register (MMCHS_RSP10). Note: No automatic card error detection for autoCMD12 is implemented; the host system has to check autoCMD12 response register (MMCHS_RSP76) for possible card errors.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CSRE																															

Bits	Field Name	Description	Type	Reset
31:0	CSRE	Card status response error	RW	0x0000 0000

Table 19-53. Register Call Summary for Register MMCHS_CSRE

SDIO Functional Description • Interrupt Requests: [0]
SDIO Programming Guide • Operational Modes Configuration: [1][2]
SDIO Register Manual • SDIO Register Summary: [3] • SDIO Register Description: [4][5]

Table 19-54. MMCHS_SYSTEST

Address Offset	0x0000 0128		
Physical Address	0x480D 1128	**Instance**	MMC
Description	System Test Register This register is used to control the signals that connect to I/O pins when the module is configured in system test (SYSTEST) mode for boundary connectivity verification. Note: In SYSTEST mode, a write into MMCHS_CMD register will not start a transfer. The buffer behaves as a stack accessible only by the local host (push and pop operations). In this mode, the Transfer Block Size (MMCHS_BLK[BLEN]) and the Blocks count for current transfer (MMCHS_BLK[NBLK]) are needed to generate a Buffer write ready interrupt (MMCHS_STAT[BWR]) or a Buffer read ready interrupt (MMCHS_STAT[BRR]) and DMA requests if enabled.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																		WAKD	SSB	RESERVED				D3D	D2D	D1D	D0D	DDIR	CDAT	CDIR	MCKD

Bits	Field Name	Description	Type	Reset
31:14	RESERVED		R	0x0
13	WAKD	Wake request output signal data value Write 0x0: The pin SWAKEUP is driven low. Write 0x1: The pin SWAKEUP is driven high. Read 0x1: No action. Returns 1. Read 0x0: No action. Returns 0.	RW	0
12	SSB	Set status bit This bit must be cleared prior attempting to clear a status bit of the interrupt status register (MMCHS_STAT). Write 0x0: Clear this SSB bitfield. Writing 0 does not clear already set status bits; Write 0x1: Force to 1 all status bits of the interrupt status register (MMCHS_STAT) only if the corresponding bitfield in the Interrupt signal enable register (MMCHS_ISE) is set. Read 0x1: No action. Returns 1. Read 0x0: No action. Returns 0.	RW	0
11:8	RESERVED		R	0

Bits	Field Name	Description	Type	Reset
7	D3D	DAT3 input/output signal data value Write 0x0: If SYSTEST[DDIR] = 0 (output mode direction), the DAT3 line is driven low. If SYSTEST[DDIR] = 1 (input mode direction), no effect. Write 0x1: If SYSTEST[DDIR] = 0 (output mode direction), the DAT3 line is driven high. If SYSTEST[DDIR] = 1 (input mode direction), no effect. Read 0x1: If SYSTEST[DDIR] = 1 (input mode direction), returns the value on the DAT3 line (high) If SYSTEST[DDIR] = 0 (output mode direction), returns 1 Read 0x0: If SYSTEST[DDIR] = 1 (input mode direction), returns the value on the DAT3 line (low). If SYSTEST[DDIR] = 0 (output mode direction), returns 0	RW	0
6	D2D	DAT2 input/output signal data value Write 0x0: If SYSTEST[DDIR] = 0 (output mode direction), the DAT2 line is driven low. If SYSTEST[DDIR] = 1 (input mode direction), no effect. Write 0x1: If SYSTEST[DDIR] = 0 (output mode direction), the DAT2 line is driven high. If SYSTEST[DDIR] = 1 (input mode direction), no effect. Read 0x1: If SYSTEST[DDIR] = 1 (input mode direction), returns the value on the DAT2 line (high) If SYSTEST[DDIR] = 0 (output mode direction), returns 1 Read 0x0: If SYSTEST[DDIR] = 1 (input mode direction), returns the value on the DAT2 line (low). If SYSTEST[DDIR] = 0 (output mode direction), returns 0	RW	0
5	D1D	DAT1 input/output signal data value Write 0x0: If SYSTEST[DDIR] = 0 (output mode direction), the DAT1 line is driven low. If SYSTEST[DDIR] = 1 (input mode direction), no effect. Write 0x1: If SYSTEST[DDIR] = 0 (output mode direction), the DAT1 line is driven high. If SYSTEST[DDIR] = 1 (input mode direction), no effect. Read 0x1: If SYSTEST[DDIR] = 1 (input mode direction), returns the value on the DAT1 line (high) If SYSTEST[DDIR] = 0 (output mode direction), returns 1 Read 0x0: If SYSTEST[DDIR] = 1 (input mode direction), returns the value on the DAT1 line (low). If SYSTEST[DDIR] = 0 (output mode direction), returns 0	RW	0
4	D0D	DAT0 input/output signal data value Write 0x0: If SYSTEST[DDIR] = 0 (output mode direction), the DAT0 line is driven low. If SYSTEST[DDIR] = 1 (input mode direction), no effect. Write 0x1: If SYSTEST[DDIR] = 0 (output mode direction), the DAT0 line is driven high. If SYSTEST[DDIR] = 1 (input mode direction), no effect. Read 0x1: If SYSTEST[DDIR] = 1 (input mode direction), returns the value on the DAT0 line (high) If SYSTEST[DDIR] = 0 (output mode direction), returns 1 Read 0x0: If SYSTEST[DDIR] = 1 (input mode direction), returns the value on the DAT0 line (low). If SYSTEST[DDIR] = 0 (output mode direction), returns 0	RW	0
3	DDIR	Control of the DAT[7:0] pins direction. Write 0x0: The DAT lines are outputs (host to card) Write 0x1: The DAT lines are inputs (card to host) Read 0x1: No action. Returns 1. Read 0x0: No action. Returns 0.	RW	0

Bits	Field Name	Description	Type	Reset
2	CDAT	CMD input/output signal data value Write 0x0: If SYSTEST[CDIR] = 0 (output mode direction), the CMD line is driven low. If SYSTEST[CDIR] = 1 (input mode direction), no effect. Write 0x1: If SYSTEST[CDIR] = 0 (output mode direction), the CMD line is driven high. If SYSTEST[CDIR] = 1 (input mode direction), no effect. Read 0x1: If SYSTEST[CDIR] = 1 (input mode direction), returns the value on the CMD line (high) If SYSTEST[CDIR] = 0 (output mode direction), returns 1 Read 0x0: If SYSTEST[CDIR] = 1 (input mode direction), returns the value on the CMD line (low). If SYSTEST[CDIR] = 0 (output mode direction), returns 0	RW	0
1	CDIR	Control of the CMD pin direction. Write 0x0: The CMD line is an output (host to card) Write 0x1: The CMD line is an input (card to host) Read 0x1: No action. Returns 1. Read 0x0: No action. Returns 0.	RW	0
0	MCKD	SDIO clock output signal data value Write 0x0: The output clock is driven low. Write 0x1: The output clock is driven high. Read 0x1: No action. Returns 1. Read 0x0: No action. Returns 0.	RW	0

Table 19-55. Register Call Summary for Register MMCHS_SYSTEST

SDIO Functional Description • Test Registers: [0]
SDIO Register Manual • SDIO Register Summary: [1]

Table 19-56. MMCHS_CON

Address Offset	0x0000 012C		
Physical Address	0x480D 112C	**Instance**	MMC
Description	Configuration Register This register is used: - to select the functional mode or the SYSTEST mode for any card. - to send an initialization sequence to any card. - to enable the detection on DAT[1] of a card interrupt for SDIO cards only.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED										SDMA_LNE	DMA_MNS	DDR	RESERVED		CLKEXTFREE	PADEN	RESERVED			CTPL	RESERVED			DW8	MODE	RESERVED			INIT	RESERVED	

Bits	Field Name	Description	Type	Reset
31:22	RESERVED		R	0x000
21	SDMA_LNE	Slave DMA Level/Edge Request: The waveform of the DMA request can be configured either edge sensitive with early de-assertion on first access to MMCHS_DATA register or late de-assertion, request remains active until last allowed data written into MMCHS_DATA. 0x0: Slave DMA edge sensitive, Early DMA de-assertion 0x1: Slave DMA level sensitive, Late DMA de-assertion	RW	0
20	DMA_MNS	DMA Master or Slave selection: When this bit is set and the controller is configured to use the DMA, Interconnect master interface is used to get datas from system using ADMA2 procedure (direct access to the memory).This option is only available if generic parameter MADMA_EN is asserted to '1'. 0x0: The controller is slave on data transfers with system. 0x1: The controller is master on data exchange with system, controller must be configured as using DMA. **NOTE: The SDIO Controller does NOT support Master DMA operations. This bit must be kept to value of 0x0.**	RW	0
19	DDR	Dual Data Rate mode: When this register is set, the controller uses both clock edge to emit or receive data. Odd bytes are transmitted on falling edges and even bytes are transmitted on rise edges. It only applies on Data bytes and CRC, Start, end bits and CRC status are kept full cycle. This bit field is only meaningful and active for even clock divider ratio of MMCHS_SYSCTL[CLKD], it is insensitive to MMCHS_HCTL[HSPE] setting. 0x0: Standard mode : data are transmitted on a single edge depending on MMCHS_HCTRL[HSPE]. 0x1: Data Bytes and CRC are transmitted on both edge.**NOTE: The SDIO Controller does NOT support DDR mode. This bit must be kept to value of 0x0.**	RW	0
18:17	RESERVED		R	0x0
16	CLKEXTFREE	External clock free running: This register is used to maintain card clock out of transfer transaction to enable slave module for example to generate a synchronous interrupt on DAT[1]. The Clock will be maintain only if MMCHS_SYSCTL[CEN] is set. 0x0: External card clock is cut off outside active transaction period. 0x1: External card clock is maintain even out of active transaction period only if MMCHS_SYSCTL[CEN] is set.	RW	0
15	PADEN	Control Power for SDIO Lines: This register is only useful when SDIO PADs contain power saving mechanism to minimize its leakage power. It works as a GPIO that directly control the ACTIVE pin of PADs. Excepted for DAT[1], the signal is also combine outside the module with the dedicated power control MMCHS_CON[CTPL] bit. 0x0: ADPIDLE module pin is not forced, it is automatically generated by the SDIO controller FSMs. 0x1: ADPIDLE module pin is forced to active state.	RW	0
14:12	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
11	CTPL	Control Power for DAT[1] line SD cards: By default, this bit is set to 0 and the host controller automatically disables all the input buffers outside of a transaction to minimize the leakage current. SDIO cards: When this bit is set to 1, the host controller automatically disables all the input buffers except the buffer of DAT[1] outside of a transaction in order to detect asynchronous card interrupt on DAT[1] line and minimize the leakage current of the buffers. 0x0: Disable all the input buffers outside of a transaction. 0x1: Disable all the input buffers except the buffer of DAT[1] outside of a transaction.	RW	0
10:6	RESERVED			0x18
5	DW8	8-bit mode MMC select **For SD/SDIO cards, this bit must be set to 0.** 0x0: 1-bit or 4-bit Data width (DAT[0] used, SD cards) 0x1: 8-bit Data width. **NOTE: Not supported on this device.**	RW	0
4	MODE	Mode select This bit select between Functional mode and SYSTEST mode. 0x0: Functional mode. Transfers to SD/SDIO cards follow the card protocol. SDIO clock is enabled. SD transfers are operated under the control of the CMD register. 0x1: SYSTEST mode The signal pins are configured as general-purpose input/output and the 1024-byte buffer is configured as a stack memory accessible only by the local host or system DMA. The pins retain their default type (input, output or in-out). SYSTEST mode is operated under the control of the SYSTEST register.	RW	0
3:2	RESERVED		R	0x0
1	INIT	Send initialization stream All cards. When this bit is set to 1, and the card is idle, an initialization sequence is sent to the card. An initialization sequence consists of setting the CMD line to 1 during 80 clock cycles. The initialisation sequence is mandatory - but it is not required to do it through this bit - this bit makes it easier. Clock divider (MMCHS_SYSCTL[CLKD]) should be set to ensure that 80 clock periods are greater than 1ms. (section 6.4 in the SD card specification). Note: in this mode, there is no command sent to the card and no response is expected 0x0: The host does not send an initialization sequence. 0x1: The host sends an initialization sequence.	RW	0
0	RESERVED		R	0

Table 19-57. Register Call Summary for Register MMCHS_CON

SDIO Functional Description • Power Management: [0][1][2][3] • DMA Modes: [4] • Test Registers: [5]
SDIO Programming Guide • Global Initialization: [6] • Operational Modes Configuration: [7][8][9]

Table 19-57. Register Call Summary for Register MMCHS_CON (continued)

SDIO Register Manual • SDIO Register Summary: [10] • SDIO Register Description: [11][12][13]

Table 19-58. MMCHS_PWCNT

Address Offset	0x0000 0130		
Physical Address	0x480D 1130	**Instance**	MMC
Description	Power Counter Register This register is used to program a counter to delay command transfers after activating the PAD power, this value depends on PAD characteristics and voltage.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																PWRCNT															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0000
15:0	PWRCNT	Power counter register. This register is used to introduce a delay between the PAD ACTIVE pin assertion and the command issued. 0xFFFF: TCF x 65535 delay (card clock period) 0x0: No additional delay added 0x1: TCF delay (card clock period) 0xFFFE: TCF x 65534 delay (card clock period) 0x2: TCF x 2 delay (card clock period)	RW	0x0000

Table 19-59. Register Call Summary for Register MMCHS_PWCNT

SDIO Functional Description • Power Management: [0][1]
SDIO Register Manual • SDIO Register Summary: [2]

Table 19-60. MMCHS_SDMASA

Address Offset	0x0000 0200		
Physical Address	0x480D 1200	**Instance**	MMC
Description	This register contains the physical system memory address used for DMA transfers.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SDMA_ARG2																															

Bits	Field Name	Description	Type	Reset
31:0	SDMA_ARG2	This register contains the system memory address for a DMA transfer. When the Host Controller stops the DMA transfer, this register shall point to the system address of the next contiguous data position. It can be accessed only if no transaction is executing (i.e., after a transaction has stopped). Read operations during transfers may return an invalid value. The Host Driver shall initialize this register before starting a DMA transaction. After DMA has stopped, the next system address of the next contiguous data position can be read from this register. The DMA transfer waits at the every boundary specified by the Host DMA Buffer Boundary in the Block Size register. The Host Controller generates DMA Interrupt to request the Host Driver to update this register. The Host Driver sets the next system address of the next data position to this register. When the most upper byte of this register (003h) is written, the Host Controller restarts the DMA transfer. When restarting DMA by the Resume command or by setting Continue Request in the Block Gap Control register, the Host Controller shall start at the next contiguous address stored here in the DMA System Address register.	RW	0x0000 0000

Table 19-61. Register Call Summary for Register MMCHS_SDMASA

SDIO Register Manual

- SDIO Register Summary: [0]

Table 19-62. MMCHS_BLK

Address Offset	0x0000 0204		
Physical Address	0x480D 1204	**Instance**	MMC
Description	Transfer Length Configuration Register MMCHS_BLK[BLEN] is the block size register. MMCHS_BLK[NBLK] is the block count register. This register shall be used for any card.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
NBLK																RESERVED				BLEN											

Bits	Field Name	Description	Type	Reset
31:16	NBLK	Blocks count for current transfer This register is enabled when Block count Enable (MMCHS_CMD[BCE]) is set to 1 and is valid only for multiple block transfers. Setting the block count to 0 results no data blocks being transferred. Note: The host controller decrements the block count after each block transfer and stops when the count reaches zero. This register can be accessed only if no transaction is executing (i.e, after a transaction has stopped). Read operations during transfers may return an invalid value and write operation will be ignored. In suspend context, the number of blocks yet to be transferred can be determined by reading this register. When restoring transfer context prior to issuing a Resume command, The local host shall restore the previously saved block count. 0xFFFF: 65535 blocks 0x0: Stop count 0x1: 1 block 0x2: 2 blocks	RW	0x0000
15:12	RESERVED		R	0x0
11:0	BLEN	Transfer Block Size. This register specifies the block size for block data transfers. Read operations during transfers may return an invalid value, and write operations are ignored. When a CMD12 command is issued to stop the transfer, a read of the BLEN field after transfer completion (MMCHS_STAT[TC] set to 1) will not return the true byte number of data length while the stop occurs but the value written in this register before transfer is launched. 0x1: 1 byte block length 0x7FF: 2047 bytes block length 0x0: No data transfer 0x1FF: 511 bytes block length 0x800: 2048 bytes block length 0x2: 2 bytes block length 0x3: 3 bytes block length 0x200: 512 bytes block length	RW	0x000

Table 19-63. Register Call Summary for Register MMCHS_BLK

SDIO Functional Description • Interrupt Requests: [0][1] • Slave DMA Operations: [2][3] • Data Buffer: [4]
SDIO Programming Guide • Operational Modes Configuration: [5][6][7][8]
SDIO Register Manual • SDIO Register Summary: [9] • SDIO Register Description: [10][11][12][13][14][15][16][17][18]

Table 19-64. MMCHS_ARG

Address Offset	0x0000 0208		
Physical Address	0x480D 1208	**Instance**	MMC
Description	Command Argument Register This register contains command argument specified as bit 39-8 of Command-Format These registers must be initialized prior to sending the command itself to the card (write action into the register MMCHS_CMD register). Only exception is for a command index specifying stuff bits in arguments, making a write unnecessary.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
ARG																															

Bits	Field Name	Description	Type	Reset
31:0	ARG	Command argument bits [31:0]	RW	0x0000 0000

Table 19-65. Register Call Summary for Register MMCHS_ARG

SDIO Programming Guide
- Operational Modes Configuration: [0][1]

SDIO Register Manual
- SDIO Register Summary: [2]

Table 19-66. MMCHS_CMD

Address Offset	0x0000 020C		
Physical Address	0x480D 120C	**Instance**	MMC
Description	Command and Transfer Mode Register MMCHS_CMD[31:16] = the command register MMCHS_CMD[15:0] = the transfer mode. This register configures the data and command transfers. A write into the most significant byte send the command. A write into MMCHS_CMD[15:0] registers during data transfer has no effect. This register shall be used for any card. Note: In SYSTEST mode, a write into MMCHS_CMD register will not start a transfer.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								INDX						CMD_TYPE		DP	CICE	CCCE	RESERVED	RSP_TYPE		RESERVED								MSBS	DDIR	ACEN		BCE	DE

Bits	Field Name	Description	Type	Reset
31:30	RESERVED		R	0x0
29:24	INDX	Command index Binary encoded value from 0 to 63 specifying the command number send to card 0xD: CMD13 or ACMD13 0x33: CMD51 or ACMD51 0x3B: CMD59 or ACMD59 0x15: CMD21 or ACMD21 0x1E: CMD30 or ACMD30 0x8: CMD8 or ACMD8 0x5: CMD5 or ACMD5	RW	0x00

Bits	Field Name	Description	Type	Reset
		0x2E: CMD46 or ACMD46		
		0x1B: CMD27 or ACMD27		
		0x2C: CMD44 or ACMD44		
		0x36: CMD54 or ACMD54		
		0x2: CMD2 or ACMD2		
		0x3E: CMD62 or ACMD62		
		0x4: CMD4 or ACMD4		
		0x39: CMD57 or ACMD57		
		0x32: CMD50 or ACMD50		
		0x6: CMD6 or ACMD6		
		0x1: CMD1 or ACMD1		
		0x1D: CMD29 or ACMD29		
		0x3F: CMD63 or ACMD63		
		0x28: CMD40 or ACMD40		
		0x3A: CMD58 or ACMD58		
		0x24: CMD36 or ACMD36		
		0x0: CMD0 or ACMD0		
		0x2D: CMD45 or ACMD45		
		0x38: CMD56 or ACMD56		
		0x3C: CMD60 or ACMD60		
		0xB: CMD11 or ACMD11		
		0x3D: CMD61 or ACMD61		
		0x20: CMD32 or ACMD32		
		0x3: CMD3 or ACMD3		
		0x17: CMD23 or ACMD23		
		0x30: CMD48 or ACMD48		
		0x31: CMD49 or ACMD49		
		0x11: CMD17 or ACMD17		
		0x23: CMD35 or ACMD35		
		0x35: CMD53 or ACMD53		
		0x2F: CMD47 or ACMD47		
		0xA: CMD10 or ACMD10		
		0x9: CMD9 or ACMD9		
		0x10: CMD16 or ACMD16		
		0x26: CMD38 or ACMD38		
		0x21: CMD33 or ACMD33		
		0x25: CMD37 or ACMD37		
		0x12: CMD18 or ACMD18		
		0x13: CMD19 or ACMD19		
		0x2B: CMD43 or ACMD43		
		0x37: CMD55 or ACMD55		
		0x18: CMD24 or ACMD24		
		0x14: CMD20 or ACMD20		
		0xE: CMD14 or ACMD14		
		0x16: CMD22 or ACMD22		
		0x2A: CMD42 or ACMD42		
		0x1C: CMD28 or ACMD28		
		0x7: CMD7 or ACMD7		
		0x19: CMD25 or ACMD25		

Bits	Field Name	Description	Type	Reset
		0x1F: CMD31 or ACMD31 0x34: CMD52 or ACMD52 0x1A: CMD26 or ACMD26 0x29: CMD41 or ACMD41 0xF: CMD15 or ACMD15 0xC: CMD12 or ACMD12 0x27: CMD39 or ACMD39 0x22: CMD34 or ACMD34		
23:22	CMD_TYPE	Command type This register specifies three types of special command: Suspend, Resume and Abort. These bits shall be set to 00b for all other commands. 0x0: Others Commands 0x1: CMD52 for writing "Bus Suspend" in CCCR 0x3: Abort command CMD12, CMD52 for writing " I/O Abort" in CCCR 0x2: CMD52 for writing "Function Select" in CCCR	RW	0x0
21	DP	Data present select This register indicates that data is present and DAT line shall be used. It must be set to 0 in the following conditions: - command using only CMD line - command with no data transfer but using busy signal on DAT[0] - Resume command 0x0: Command with no data transfer 0x1: Command with data transfer	RW	0
20	CICE	Command Index check enable This bit must be set to 1 to enable index check on command response to compare the index field in the response against the index of the command. If the index is not the same in the response as in the command, it is reported as a command index error (MMCHS_STAT[CIE] set to1). **Note:** The register CICE cannot be configured for an Auto CMD12, then index check is automatically checked when this command is issued. 0x0: Index check disable 0x1: Index check enable	RW	0
19	CCCE	Command CRC check enable This bit must be set to 1 to enable CRC7 check on command response to protect the response against transmission errors on the bus. If an error is detected, it is reported as a command CRC error (MMCHS_STAT[CCRC] set to 1). **Note:** The register CCCE cannot be configured for an Auto CMD12, and then CRC check is automatically checked when this command is issued. 0x0: CRC7 check disable 0x1: CRC7 check enable	RW	0
18	RESERVED		R	0
17:16	RSP_TYPE	Response type This bits defines the response type of the command 0x0: No response	RW	0x0

Bits	Field Name	Description	Type	Reset
		0x1: Response Length 136 bits 0x3: Response Length 48 bits with busy after response 0x2: Response Length 48 bits		
15:6	RESERVED		R	0x000
5	MSBS	Multi/Single block select This bit must be set to 1 for data transfer in case of multi block command. For any others command this bit shall be set to 0. 0x0: Single block. If this bit is 0, it is not necessary to set the register MMCHS_BLK[NBLK]. 0x1: Multi block. When Block Count is disabled (MMCHS_CMD[BCE] is set to 0) in Multiple block transfers (MMCHS_CMD[MSBS] is set to 1), the module can perform infinite transfer.	RW	0
4	DDIR	Data transfer Direction Select This bit defines either data transfer will be a read or a write. 0x0: Data Write (host to card) 0x1: Data Read (card to host)	RW	0
3:2	ACEN	Auto CMD Enable - SD card only. This field determines use of auto command functions. There are two methods to stop Multiple-block read and write operation Auto CMD12 Enable When this field is set to 01b, the Host Controller issues CMD12 automatically when last block transfer is completed. Auto CMD12 error is indicated to the Auto CMD Error Status register (MMCHS_AC12). The Host Driver shall not set this bit if the command does not require CMD12. In particular, secure commands defined in the Part 3 File Security specification do not require CMD12. 0x0: Auto Command Disabled 0x1: Auto CMD12 enable or CCS detection enabled. 0x2: Reserved 0x3: Reserved	RW	0x0
1	BCE	Block Count Enable Multiple block transfers only. This bit is used to enable the block count register (MMCHS_BLK[NBLK]). When Block Count is disabled (MMCHS_CMD[BCE] is set to 0) in Multiple block transfers (MMCHS_CMD[MSBS] is set to 1), the module can perform infinite transfer. 0x0: Block count disabled for infinite transfer. 0x1: Block count enabled for multiple block transfer with known number of blocks	RW	0
0	DE	DMA Enable This bit is used to enable DMA mode for host data access. 0x0: DMA mode disable 0x1: DMA mode enable	RW	0

Table 19-67. Register Call Summary for Register MMCHS_CMD

SDIO Environment • Data Format: [0][1][2][3]
SDIO Functional Description • Interrupt Requests: [4][5] • Slave DMA Operations: [6] • Data Buffer: [7][8][9] • Transfer Stop: [10]
SDIO Programming Guide • Operational Modes Configuration: [11][12][13][14][15][16]
SDIO Register Manual • SDIO Register Summary: [17] • SDIO Register Description: [18][19][20][21][22][23][24][25][26][27][28][29][30][31][32][33]

Table 19-68. MMCHS_RSP10

Address Offset	0x0000 0210		
Physical Address	0x480D 1210	**Instance**	MMC
Description	Command Response[31:0] Register (bits [31:0] of the internal RSP register). This 32-bit register holds bits positions [31:0] of command response type R1/R1b/R2/R3/R4/R5/R5b/R6/R7		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSP1																RSP0															

Bits	Field Name	Description	Type	Reset
31:16	RSP1	Command Response [31:16]	R	0x0000
15:0	RSP0	Command Response [15:0]	R	0x0000

Table 19-69. Register Call Summary for Register MMCHS_RSP10

SDIO Functional Description • Different Types of Responses: [0][1][2]
SDIO Programming Guide • Operational Modes Configuration: [3][4][5]
SDIO Register Manual • SDIO Register Summary: [6] • SDIO Register Description: [7]

Table 19-70. MMCHS_RSP32

Address Offset	0x0000 0214		
Physical Address	0x480D 1214	**Instance**	MMC
Description	Command Response[63:32] Register (bits [63:32] of the internal RSP register) This 32-bit register holds bits positions [63:32] of command response type R2		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSP3																RSP2															

Bits	Field Name	Description	Type	Reset
31:16	RSP3	Command Response [63:48]	R	0x0000
15:0	RSP2	Command Response [47:32]	R	0x0000

Table 19-71. Register Call Summary for Register MMCHS_RSP32

SDIO Functional Description • Different Types of Responses: [0]
SDIO Programming Guide • Operational Modes Configuration: [1][2]
SDIO Register Manual • SDIO Register Summary: [3]

Table 19-72. MMCHS_RSP54

Address Offset	0x0000 0218		
Physical Address	0x480D 1218	**Instance**	MMC
Description	Command Response[95:64] Register (bits [95:64] of the internal RSP register) This 32-bit register holds bits positions [95:64] of command response type R2		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSP5																RSP4															

Bits	Field Name	Description	Type	Reset
31:16	RSP5	Command Response [95:80]	R	0x0000
15:0	RSP4	Command Response [79:64]	R	0x0000

Table 19-73. Register Call Summary for Register MMCHS_RSP54

SDIO Functional Description • Different Types of Responses: [0]
SDIO Programming Guide • Operational Modes Configuration: [1][2]
SDIO Register Manual • SDIO Register Summary: [3]

Table 19-74. MMCHS_RSP76

Address Offset	0x0000 021C		
Physical Address	0x480D 121C	**Instance**	MMC
Description	Command Response[127:96] Register (bits [127:96] of the internal RSP register) This 32-bit register holds bits positions [127:96] of command response type R1/R1b(Auto CMD12)/R2		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RSP7																RSP6															

Bits	Field Name	Description	Type	Reset
31:16	RSP7	Command Response [127:112]	R	0x0000
15:0	RSP6	Command Response [111:96]	R	0x0000

Table 19-75. Register Call Summary for Register MMCHS_RSP76

SDIO Functional Description • Different Types of Responses: [0][1][2]
SDIO Programming Guide • Operational Modes Configuration: [3][4]
SDIO Register Manual • SDIO Register Summary: [5] • SDIO Register Description: [6][7]

Table 19-76. MMCHS_DATA

Address Offset	0x0000 0220		
Physical Address	0x480D 1220	**Instance**	MMC
Description	Data Register This register is the 32-bit entry point of the buffer for read or write data transfers. The buffer size is 32bits x256(1024 bytes). Bytes within a word are stored and read in little endian format. This buffer can be used as two 512 byte buffers to transfer data efficiently without reducing the throughput. Sequential and contiguous access is necessary to increment the pointer correctly. Random or skipped access is not allowed. In little endian, if the local host accesses this register byte-wise or 16bit-wise, the least significant byte (bits [7:0]) must always be written/read first. The update of the buffer address is done on the most significant byte write for full 32-bit DATA register or on the most significant byte of the last word of block transfer. Example 1: Byte or 16-bit access Mbyteen[3:0]=0001 (1-byte) => Mbyteen[3:0]=0010 (1-byte) => Mbyteen[3:0]=1100 (2-bytes) OK Mbyteen[3:0]=0001 (1-byte) => Mbyteen[3:0]=0010 (1-byte) => Mbyteen[3:0]=0100 (1-byte) OK Mbyteen[3:0]=0001 (1-byte) => Mbyteen[3:0]=0010 (1-byte) => Mbyteen[3:0]=1000 (1-byte) Bad		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATA																															

Bits	Field Name	Description	Type	Reset
31:0	DATA	Data Register [31:0] In functional mode (MMCHS_CON[MODE] set to the default value 0) , A read access to this register is allowed only when the buffer read enable status is set to 1 (MMCHS_PSTATE[BRE]), otherwise a bad access (MMCHS_STAT[BADA]) is signaled. A write access to this register is allowed only when the buffer write enable status is set to 1(MMCHS_STATE[BWE]), otherwise a bad access (MMCHS_STAT[BADA]) is signaled and the data is not written.	RW	0x0000 0000

Table 19-77. Register Call Summary for Register MMCHS_DATA

SDIO Functional Description • Interrupt Requests: [0][1] • Data Buffer: [2][3][4][5][6][7][8][9] • Transfer or Command Status and Errors Reporting: [10]
SDIO Programming Guide • Operational Modes Configuration: [11]
SDIO Register Manual • SDIO Register Summary: [12] • SDIO Register Description: [13][14][15][16][17][18]

Table 19-78. MMCHS_PSTATE

Address Offset	0x0000 0224		
Physical Address	0x480D 1224	**Instance**	MMC
Description	Present State Register The Host can get status of the Host Controller from this 32-bit read only register.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED							CLEV	DLEV				RESERVED								BRE	BWE	RTA	WTA	RESERVED					DLA	DATI	CMDI

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x00
24	CLEV	CMD line signal level This status is used to check the CMD line level to recover from errors, and for debugging. The value of this register after reset depends on the CMD line level at that time. Read 0x1: The CMD line level is 1. Read 0x0: The CMD line level is 0.	R	-
23:20	DLEV	DAT[3:0] line signal level DAT[3] => bit 23 DAT[2] => bit 22 DAT[1] => bit 21 DAT[0] => bit 20 This status is used to check DAT line level to recover from errors, and for debugging. This is especially useful in detecting the busy signal level from DAT[0]. The value of these registers after reset depends on the DAT lines level at that time.	R	0x-
19:12	RESERVED		R	0x-
11	BRE	Buffer read enable This bit is used for non-DMA read transfers. It indicates that a complete block specified by MMCHS_BLK[BLEN] has been written in the buffer and is ready to be read. It is set to 0 when the entire block is read from the buffer. It is set to 1 when a block data is ready in the buffer and generates the Buffer read ready status of interrupt (MMCHS_STAT[BRR]). Read 0x1: Read BLEN bytes enable. Readable data exists in the buffer. Read 0x0: Read BLEN bytes disable	R	0
10	BWE	Buffer Write enable This status is used for non-DMA write transfers. It indicates if space is available for write data. Read 0x1: There is enough space in the buffer to write BLEN bytes of data. Read 0x0: There is no room left in the buffer to write BLEN bytes of data.	R	0
9	RTA	Read transfer active This status is used for detecting completion of a read transfer. It is set to 1 after the end bit of read command or by activating a continue request (MMCHS_HCTL[CR]) following a stop at block gap request. This bit is set to 0 when all data have been read by the local host after last block or after a stop at block gap request. Read 0x1: read data transfer on going. Read 0x0: No valid data on the DAT lines.	R	0

Bits	Field Name	Description	Type	Reset
8	WTA	Write transfer active This status indicates a write transfer active. It is set to 1 after the end bit of write command or by activating a continue request (MMCHS_HCTL[CR]) following a stop at block gap request. This bit is set to 0 when CRC status has been received after last block or after a stop at block gap request. Read 0x1: Write data transfer on going. Read 0x0: No valid data on the DAT lines.	R	0
7:3	RESERVED		R	0x0
2	DLA	DAT line active This status bit indicates whether one of the DAT line is in use. In the case of read transactions (card to host): This bit is set to 1 after the end bit of read command or by activating continue request MMCHS_HCTL[CR]. This bit is set to 0 when the host controller received the end bit of the last data block or at the beginning of the read wait mode. In the case of write transactions (host to card): This bit is set to 1 after the end bit of write command or by activating continue request MMCHS_HCTL[CR]. This bit is set to 0 on the end of busy event for the last block; host controller must wait 8 clock cycles with line not busy to really consider not "busy state" or after the busy block as a result of a stop at gap request. Read 0x1: DAT Line active Read 0x0: DAT Line inactive	R	0
1	DATI	Command inhibit(DAT) This status bit is generated if either DAT line is active (MMCHS_PSTATE[DLA]) or Read transfer is active (MMCHS_PSTATE[RTA]) or when a command with busy is issued. This bit prevents the local host to issue a command. A change of this bit from 1 to 0 generates a transfer complete interrupt (MMCHS_STAT[TC]). Read 0x1: Issuing of command using DAT lines is not allowed Read 0x0: Issuing of command using the DAT lines is allowed	R	0
0	CMDI	Command inhibit(CMD) This status bit indicates that the CMD line is in use. This bit is set to 0 when the most significant byte is written into the command register. This bit is not set when Auto CMD12 is transmitted. This bit is set to 0 in either the following cases: - After the end bit of the command response, excepted if there is a command conflict error (MMCHS_STAT[CCRC] or MMCHS_STAT[CEB] set to 1) or a Auto CMD12 is not executed (MMCHS_AC12[ACNE]). - After the end bit of the command without response (MMCHS_CMD[RSP_TYPE] set to "00") In case of a command data error is detected (MMCHS_STAT[CTO] set to 1), this register is not automatically cleared. Read 0x1: Issuing of command using CMD line is not allowed Read 0x0: Issuing of command using CMD line is allowed	R	0

Table 19-79. Register Call Summary for Register MMCHS_PSTATE

SDIO Functional Description • Software Reset: [0] • Interrupt Requests: [1][2][3][4][5] • Data Buffer: [6][7][8][9] • SDIO Hardware Status Features: [10][11][12][13][14][15][16][17][18]
SDIO Programming Guide • Operational Modes Configuration: [19][20][21][22][23]
SDIO Register Manual • SDIO Register Summary: [24] • SDIO Register Description: [25][26][27][28][29][30][31][32][33][34][35][36]

Table 19-80. MMCHS_HCTL

Address Offset	0x0000 0228		
Physical Address	0x480D 1228	**Instance**	MMC
Description	Host Control Register This register defines the host controls to set power, wakeup and transfer parameters. MMCHS_HCTL[24] = Wakeup control MMCHS_HCTL[23:16] = Block gap control MMCHS_HCTL[15:8] = Power control MMCHS_HCTL[7:0] = Host control		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED							IWE	RESERVED				IBG	RWC	CR	SBGR	RESERVED				SDVS			SDBP	RESERVED					HSPE	DTW	RESERVED

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24	IWE	Wakeup event enable on SD card interrupt This bit enables wakeup events for card interrupt assertion. Wakeup is generated if the wakeup feature is enabled (MMCHS_SYSCONFIG[ENAWAKEUP]). 0x0: Disable wakeup on card interrupt 0x1: Enable wakeup on card interrupt	RW	0
23:20	RESERVED		R	0x0
19	IBG	Interrupt block at gap This bit is valid only in 4-bit mode of SDIO card to enable interrupt detection in the interrupt cycle at block gap for a multiple block transfer. For SD cards this bit should be set to 0. 0x0: Disable interrupt detection at the block gap in 4-bit mode 0x1: Enable interrupt detection at the block gap in 4-bit mode	RW	0

Bits	Field Name	Description	Type	Reset
18	RWC	Read wait control The read wait function is optional only for SDIO cards. If the card supports read wait, this bit must be enabled, then requesting a stop at block gap (MMCHS_HCTL[SBGR]) generates a read wait period after the current end of block. Be careful, if read wait is not supported it may cause a conflict on DAT line. 0x0: Disable Read Wait Control. Suspend/Resume cannot be supported. 0x1: Enable Read Wait Control	RW	0
17	CR	Continue request This bit is used to restart a transaction that was stopped by requesting a stop at block gap (MMCHS_HCTL[SBGR]). Set this bit to 1 restarts the transfer. The bit is automatically set to 0 by the host controller when transfer has restarted i.e DAT line is active (MMCHS_PSTATE[DLA]) or transferring data (MMCHS_PSTATE[WTA]). The Stop at block gap request must be disabled (MMCHS_HCTL[SBGR]=0) before setting this bit. 0x0: No affect 0x1: transfer restart	RW	0
16	SBGR	Stop at block gap request This bit is used to stop executing a transaction at the next block gap. The transfer can restart with a continue request (MMCHS_HCTL[CR]) or during a suspend/resume sequence. In case of read transfer, the card must support read wait control. In case of write transfer, the host driver shall set this bit after all block data written. Until the transfer completion (MMCHS_STAT[TC] set to 1), the host driver shall leave this bit set to 1. If this bit is set, the local host shall not write to the data register (MMCHS_DATA). 0x0: Transfer mode 0x1: Stop at block gap	RW	0
15:12	RESERVED		R	0x0
11:9	SDVS	SD bus voltage select All cards. The host driver should set to these bits to select the voltage level for the card according to the voltage supported by the system (MMCHS_CAPA[VS18,VS30,VS33]) before starting a transfer. 0x6: 3.0V (Typical) 0x7: 3.3V (Typical) 0x5: 1.8V (Typical)	RW	0x0
8	SDBP	SD bus power Before setting this bit, the host driver shall select the SD bus voltage (MMCHS_HCTL[SDVS]). If the host controller detects the No card state, this bit is automatically set to 0. If the module is power off, a write in the command register (MMCHS_CMD) will not start the transfer. A write to this bit has no effect if the selected SD bus voltage MMCHS_HCTL[SDVS] is not supported according to capability register (MMCHS_CAPA[VS*]). 0x0: Power off 0x1: Power on	RW	0
7:3	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
2	HSPE	Before setting this bit, the Host Driver shall check the MMCHS_CAPA[21] HSS. This bit shall not be set when dual data rate mode is activated in MMCHS_CON[DDR]. 0x0: The Host Controller outputs CMD line and DAT lines at the falling edge of the SD Clock 0x1: The Host Controller outputs CMD line and DAT lines at the rising edge of the SD Clock **NOTE: Do not set this bit to 0x1 because device was timing closed with HSPE bit set to 0x0 for all supported modes of operation.**	RW	0
1	DTW	Data transfer width For SD/SDIO cards, this bit must be set following a valid SET_BUS_WIDTH command (ACMD6) with the value written in bit 1 of the argument. Prior to this command, the SD card configuration register (SCR) must be verified for the supported bus width by the SD card. 0x0: 1-bit Data width (DAT[0] used) 0x1: 4-bit Data width (DAT[3:0] used)	RW	0
0	RESERVED		R	0

Table 19-81. Register Call Summary for Register MMCHS_HCTL

SDIO Functional Description
- Power Management: [0][1][2]
- Interrupt Requests: [3][4]
- Transfer Stop: [5][6]
- Output Signals Generation: [7]
- Generation on Falling Edge of SDIO clock: [8]
- Generation on Rising Edge of SDIO clock: [9]

SDIO Programming Guide
- Global Initialization: [10][11]
- Operational Modes Configuration: [12]

SDIO Register Manual
- SDIO Register Summary: [13]
- SDIO Register Description: [14][15][16][17][18][19][20][21][22][23][24][25][26][27][28][29][30][31][32]

Table 19-82. MMCHS_SYSCTL

Address Offset	0x0000 022C		
Physical Address	0x480D 122C	**Instance**	MMC
Description	SD System Control Register This register defines the system controls to set software resets, clock frequency management and data timeout. MMCHS_SYSCTL[31:24] = Software resets MMCHS_SYSCTL[23:16] = Timeout control MMCHS_SYSCTL[15:0] = Clock control		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED					SRD	SRC	SRA	RESERVED				DTO				CLKD								CGS	RESERVED				CEN	ICS	ICE

Bits	Field Name	Description	Type	Reset
31:27	RESERVED		R	0x00
26	SRD	Software reset for DAT line This bit is set to 1 for reset and released to 0 when completed. For more information about SRD bit manipulation, see Section 19.5.1.2.1.2.1 *DATA Lines Reset Procedure*. DAT finite state machine in both clock domain are also reset. Here below are the registers cleared by MMCHS_SYSCTL[SRD]: - MMCHS_DATA - MMCHS_PSTATE: BRE, BWE, RTA, WTA, DLA and DATI - MMCHS_HCTL: SBGR and CR - MMCHS_STAT: BRR, BWR, BGE and TC Interconnect and SDIO buffer data management is reinitialized. 0x0: Reset completed 0x1: Software reset for DAT line	RW	0
25	SRC	Software reset for CMD line For more information about SRC bit manipulation, see Section 19.5.1.2.1.1.1 *CMD Line Reset Procedure*. This bit is set to 1 for reset and released to 0 when completed. CMD finite state machine in both clock domain are also reset. Here below the registers cleared by MMCHS_SYSCTL[SRC]: - MMCHS_PSTATE: CMDI - MMCHS_STAT: CC Interconnect and SDIO command status management is reinitialized. 0x0: Reset completed 0x1: Software reset for CMD line	RW	0
24	SRA	Software reset for all This bit is set to 1 for reset , and released to 0 when completed. This reset affects the entire host controller except for the capabilities registers (MMCHS_CAPA and MMCHS_CUR_CAPA). 0x0: Reset completed 0x1: Software reset for all the design	RW	0
23:20	RESERVED		R	0x0
19:16	DTO	Data timeout counter value and busy timeout. This value determines the interval by which DAT lines timeouts are detected. The host driver needs to set this bitfield based on - the maximum read access time (NAC) (Refer to the SD Specification Part1 Physical Layer), - the data read access time values (TAAC and NSAC) in the card specific data register (CSD) of the card, - the timeout clock base frequency (MMCHS_CAPA[TCF]). If the card does not respond within the specified number of cycles, a data timeout error occurs (MMCHS_STA[DTO]). The MMCHS_SYSCTL[DTO] register is also used to check busy duration, to generate busy timeout for commands with busy response or for busy programming during a write command. Timeout on CRC status is generated if no CRC token is present after a block write. 0xF: Reserved 0x0: TCF x 2^13 0x1: TCF x 2^14 0xE: TCF x 2^27	RW	0x0

Bits	Field Name	Description	Type	Reset
15:6	CLKD	Clock frequency select These bits define the ratio between MMC_FCLK and the output clock frequency on the mmc_clk pin. 0x0: MMC_FCLK bypass 0x1: MMC_FCLK bypass 0x2: MMC_FCLK / 2 0x3: MMC_FCLK / 3 0x3FF: MMC_FCLK / 1023	RW	0x000
5	CGS	Clock Generator Select - For SD cards Host Controller Version 3.00 supports this bit. This bit is used to select the clock generator mode in MMCHS_SYSCTL[15:6] CLKD. If the Programmable Clock Mode is supported (non-zero value is set to MMCHS_CAPA2[23:16] CM), this bit attribute is RW, and if not supported, this bit attribute is RO and zero is read. This bit depends on the setting of MMCHS_AC12[31] PV_ENABLE. If PV_ENABLE = 0, this bit is set by Host Driver. If PV_ENABLE = 1, this bit is automatically set to a value specified in one of Preset Value registers, see, Table 19-15.	R	0
4:3	RESERVED		R	0x0
2	CEN	Clock enable This bit controls if the clock is provided to the card or not. 0x0: The clock is not provided to the card . Clock frequency can be changed . 0x1: The clock is provided to the card and can be automatically gated when MMCHS_SYSCONFIG[AUTOIDLE] is set to 1 (default value) . The host driver shall wait to set this bit to 1 until the Internal clock is stable (MMCHS_SYSCTL[ICS]).	RW	0
1	ICS	Internal clock stable (status) This bit indicates either the internal clock is stable or not. Read 0x1: The internal clock is stable after enabling the clock (MMCHS_SYSCTL[ICE]) or after changing the clock ratio (MMCHS_SYSCTL[CLKD]). Read 0x0: The internal clock is not stable.	R	0
0	ICE	Internal clock enable This register controls the internal clock activity. In very low power state, the internal clock is stopped. Note: The activity of the debounce clock (used for wakeup events) and the interface clock (used for reads and writes to the module register map) are not affected by this register. 0x0: The internal clock is stopped (very low power state). 0x1: The internal clock oscillates and can be automatically gated when MMCHS_SYSCONFIG[AUTOIDLE] is set to 1 (default value) .	RW	0

Table 19-83. Register Call Summary for Register MMCHS_SYSCTL

SDIO Environment • Protocol: [0]
SDIO Functional Description • Software Reset: [1][2][3][4][5]
SDIO Programming Guide • Global Initialization: [6] • Operational Modes Configuration: [7][8][9][10][11][12][13][14][15][16][17][18][19][20]

Table 19-83. Register Call Summary for Register MMCHS_SYSCTL (continued)

SDIO Register Manual

- SDIO Register Summary: [21]
- SDIO Register Description: [22][23][24][25][26][27][28][29][30][31][32][33][34][35][36][37][38][39][40][41][42][43][44][45][46][47][48][49][50]

Table 19-84. MMCHS_STAT

Address Offset	0x0000 0230		
Physical Address	0x480D 1230	**Instance**	MMC
Description	Interrupt Status Register The interrupt status regroups all the status of the module internal events that can generate an interrupt. MMCHS_STAT[31:16] = Error Interrupt Status MMCHS_STAT[15:0] = Normal Interrupt Status		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED		BADA	CERR	RESERVED			ACE	RESERVED	DEB	DCRC	DTO	CIE	CEB	CCRC	CTO	ERRI	RESERVED						CIRQ	RESERVED		BRR	BWR	RESERVED		BGE	TC	CC

Bits	Field Name	Description	Type	Reset
31:30	RESERVED		R	0x0
29	BADA	Bad access to data space This bit is set automatically to indicate a bad access to buffer when not allowed: -This bit is set during a read access to the data register (MMCHS_DATA) while buffer reads are not allowed (MMCHS_PSTATE[BRE] =0) -This bit is set during a write access to the data register (MMCHS_DATA) while buffer writes are not allowed (MMCHS_STATE[BWE] =0) Write 0x0: Status bit unchanged Write 0x1: Status is cleared Read 0x1: Bad Access Read 0x0: No Interrupt.	RW	0
28	CERR	Card error This bit is set automatically when there is at least one error in a response of type R1, R1b, R6, R5 or R5b. Only bits referenced as type E(error) in status field in the response can set a card status error. An error bit in the response is flagged only if corresponding bit in card status response error MMCHS_CSRE in set. There is no card error detection for autoCMD12 command. The host driver shall read MMCHS_RSP76 register to detect error bits in the command response. Write 0x0: Status bit unchanged Write 0x1: Status is cleared Read 0x1: Card error Read 0x0: No Error	RW	0
27:25	RESERVED		R	0

Bits	Field Name	Description	Type	Reset
24	ACE	Auto CMD error Auto CMD12 uses this error status. This bit is set when detecting that one of the bits D00-D04 in Auto CMD Error Status register (MMCHS_AC12) has changed from 0 to 1. In case of Auto CMD12, this bit is set to 1, not only when the errors in Auto CMD12 occur but also when Auto CMD12 is not executed due to the previous command error. Write 0x0: Status bit unchanged Write 0x1: Status is cleared Read 0x1: Auto CMD error Read 0x0: No Error.	RW	0
23	RESERVED		R	0
22	DEB	Data End Bit error This bit is set automatically when detecting a 0 at the end bit position of read data on DAT line or at the end position of the CRC status in write mode. Write 0x0: Status bit unchanged Write 0x1: Status is cleared Read 0x1: Data end bit error Read 0x0: No Error	RW	0
21	DCRC	Data CRC Error This bit is set automatically when there is a CRC16 error in the data phase response following a block read command or if there is a 3-bit CRC status different of a position "010" token during a block write command. Write 0x0: Status bit unchanged Write 0x1: Status is cleared Read 0x1: Data CRC error Read 0x0: No Error.	RW	0
20	DTO	Data timeout error This bit is set automatically according to the following conditions: - busy timeout for R1b, R5b response type - busy timeout after write CRC status - write CRC status timeout - read data timeout Write 0x0: Status bit unchanged Write 0x1: Status is cleared Read 0x1: Time out Read 0x0: No error.	RW	0
19	CIE	Command index error This bit is set automatically when response index differs from corresponding command index previously emitted. It depends on the enable in MMCHS_CMD[CICE] register. Write 0x0: Status bit unchanged Write 0x1: Status is cleared Read 0x1: Command index error Read 0x0: No error.	RW	0
18	CEB	Command end bit error This bit is set automatically when detecting a 0 at the end bit position of a command response. Write 0x0: Status bit unchanged Write 0x1: Status is cleared Read 0x1: Command end bit error Read 0x0: No error.	RW	0

Bits	Field Name	Description	Type	Reset
17	CCRC	Command CRC Error This bit is set automatically when there is a CRC7 error in the command response depending on the enable in MMCHS_CMD[CCCE] register. Write 0x0: Status bit unchanged Write 0x1: Status is cleared Read 0x1: Command CRC error Read 0x0: No Error.	RW	0
16	CTO	Command Timeout Error This bit is set automatically when no response is received within 64 clock cycles from the end bit of the command. For commands that reply within 5 clock cycles - the timeout is still detected at 64 clock cycles. Write 0x0: Status bit unchanged Write 0x1: Status is cleared Read 0x1: Time Out Read 0x0: No error	RW	0
15	ERRI	Error Interrupt If any of the bits in the Error Interrupt Status register (MMCHS_STAT[31:16]) are set, then this bit is set to 1. Therefore the host driver can efficiently test for an error by checking this bit first. Writes to this bit are ignored. Read 0x1: Error interrupt event(s) occurred Read 0x0: No Interrupt.	R	0
14:9	RESERVED		R	0x0
8	CIRQ	Card interrupt This bit is only used for SD and SDIO cards. In 1-bit mode, interrupt source is asynchronous (can be a source of asynchronous wakeup). In 4-bit mode, interrupt source is sampled during the interrupt cycle. The controller interrupt must be clear by setting MMCHS_IE[CIRQ] to 0, then the host driver must start the interrupt service with card (clearing card interrupt status) to remove card interrupt source. Otherwise the Controller interrupt will be reasserted as soon as MMCHS_IE[CIRQ] is set to 1. Writes to this bit are ignored. Read 0x1: Generate card interrupt Read 0x0: No card interrupt	R	0
7:6	RESERVED		R	0x0
5	BRR	Buffer read ready This bit is set automatically during a read operation to the card (see class 2 - block oriented read commands) when one block specified by MMCHS_BLK[BLEN] is completely written in the buffer. It indicates that the memory card has filled out the buffer and that the local host needs to empty the buffer by reading it. Note: If the DMA receive-mode is enabled, this bit is never set; instead a DMA receive request to the main DMA controller of the system is generated. Write 0x0: Status bit unchanged Write 0x1: Status is cleared Read 0x1: Ready to read buffer Read 0x0: Not Ready to read buffer	RW	0

Bits	Field Name	Description	Type	Reset
4	BWR	Buffer write ready This bit is set automatically during a write operation to the card (see class 4 - block oriented write command) when the host can write a complete block as specified by MMCHS_BLK[BLEN]. It indicates that the memory card has emptied one block from the buffer and that the local host is able to write one block of data into the buffer. Note: If the DMA transmit mode is enabled, this bit is never set; instead, a DMA transmit request to the main DMA controller of the system is generated. Write 0x0: Status bit unchanged Write 0x1: Status is cleared Read 0x1: Ready to write buffer Read 0x0: Not Ready to write buffer	RW	0
3	RESERVED		R	0
2	BGE	Block gap event When a stop at block gap is requested (MMCHS_HCTL[SBGR]), this bit is automatically set when transaction is stopped at the block gap during a read or write operation. This event does not occur when the stop at block gap is requested on the last block. In read mode, a 1-to-0 transition of the DAT Line active status (MMCHS_PSTATE[DLA]) between data blocks generates a Block gap event interrupt. Write 0x0: Status bit unchanged Write 0x1: Status is cleared Read 0x1: Transaction stopped at block gap Read 0x0: No block gap event	RW	0
1	TC	Transfer completed This bit is always set when a read/write transfer is completed or between two blocks when the transfer is stopped due to a stop at block gap request (MMCHS_HCTL[SBGR]). In Read mode: This bit is automatically set on completion of a read transfer (MMCHS_PSTATE[RTA]). In write mode: This bit is set automatically on completion of the DAT line use (MMCHS_PSTATE[DLA]). Write 0x0: Status bit unchanged Write 0x1: Status is cleared Read 0x1: Data transfer complete Read 0x0: No transfer complete	RW	0
0	CC	Command complete This bit is set when a 1-to-0 transition occurs in the register command inhibit (MMCHS_PSTATE[CMDI]) If the command is a type for which no response is expected, then the command complete interrupt is generated at the end of the command. A command timeout error (MMCHS_STAT[CTO]) has higher priority than command complete (MMCHS_STAT[CC]). If a response is expected but none is received, then a command timeout error is detected and signaled instead of the command complete interrupt. Write 0x0: Status bit unchanged Write 0x1: Status is cleared Read 0x1: Command complete Read 0x0: No Command complete	RW	0

Table 19-85. Register Call Summary for Register MMCHS_STAT

SDIO Functional Description

- Power Management: [0][1][2][3][4]
- Interrupt Requests: [5][6][7][8][9][10][11][12][13][14][15][16][17][18][19][20][21][22][23][24][25][26][27][28][29][30][31]
- Interrupt-Driven Operation: [32][33][34][35][36][37]
- Polling: [38][39]
- Data Buffer: [40][41][42][43][44][45]
- Transfer or Command Status and Errors Reporting: [46][47]
- Busy Time-Out for R1b, R5b Response Type: [48]
- Busy Time-Out After Write CRC Status: [49]
- Write CRC Status Time-Out: [50]
- Read Data Time-Out: [51][52]

SDIO Programming Guide

- Operational Modes Configuration: [53][54][55][56][57][58][59]

SDIO Register Manual

- SDIO Register Summary: [60]
- SDIO Register Description: [61][62][63][64][65][66][67][68][69][70][71][72][73][74][75][76][77][78][79][80][81][82][83][84][85][86][87]

Table 19-86. MMCHS_IE

Address Offset	0x0000 0234		
Physical Address	0x480D 1234	**Instance**	MMC
Description	Interrupt Status Enable Register This register allows to enable/disable the module to set status bits, on an event-by-event basis. MMCHS_IE[31:16] = Error Interrupt Status Enable MMCHS_IE[15:0] = Normal Interrupt Status Enable		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED		BADA_ENABLE	CERR_ENABLE	RESERVED			ACE_ENABLE	RESERVED	DEB_ENABLE	DCRC_ENABLE	DTO_ENABLE	CIE_ENABLE	CEB_ENABLE	CCRC_ENABLE	CTO_ENABLE	NULL	RESERVED							CIRQ_ENABLE	RESERVED		BRR_ENABLE	BWR_ENABLE	RESERVED	BGE_ENABLE	TC_ENABLE	CC_ENABLE

Bits	Field Name	Description	Type	Reset
31:30	RESERVED		R	0x0
29	BADA_ENABLE	Bad access to data space Status Enable 0x0: Masked 0x1: Enabled	RW	0
28	CERR_ENABLE	Card Error Status Enable 0x0: Masked 0x1: Enabled	RW	0
27:25	RESERVED		R	0x0
24	ACE_ENABLE	Auto CMD Error Status Enable 0x0: Masked 0x1: Enabled	RW	0
23	RESERVED		R	0
22	DEB_ENABLE	Data End Bit Error Status Enable 0x0: Masked 0x1: Enabled	RW	0

Bits	Field Name	Description	Type	Reset
21	DCRC_ENABLE	Data CRC Error Status Enable 0x0: Masked 0x1: Enabled	RW	0
20	DTO_ENABLE	Data Timeout Error Status Enable 0x0: The data timeout detection is deactivated. The host controller provides the clock to the card until the card sends the data or the transfer is aborted. 0x1: The data timeout detection is enabled.	RW	0
19	CIE_ENABLE	Command Index Error Status Enable 0x0: Masked 0x1: Enabled	RW	0
18	CEB_ENABLE	Command End Bit Error Status Enable 0x0: Masked 0x1: Enabled	RW	0
17	CCRC_ENABLE	Command CRC Error Status Enable 0x0: Masked 0x1: Enabled	RW	0
16	CTO_ENABLE	Command Timeout Error Status Enable 0x0: Masked 0x1: Enabled	RW	0
15	NULL	Fixed to 0 The host driver shall control error interrupts using the Error Interrupt Signal Enable register. Writes to this bit are ignored	R	0
14:9	RESERVED		R	0x0
8	CIRQ_ENABLE	Card Status Enable A clear of this bit also clears the corresponding status bit. During 1-bit mode, if the interrupt routine doesn't remove the source of a card interrupt in the SDIO card, the status bit is reasserted when this bit is set to 1. 0x0: Masked 0x1: Enabled	RW	0
7:6	RESERVED		R	0x0
5	BRR_ENABLE	Buffer Read Ready Status Enable 0x0: Masked 0x1: Enabled	RW	0
4	BWR_ENABLE	Buffer Write Ready Status Enable 0x0: Masked 0x1: Enabled	RW	0
3	RESERVED		R	0
2	BGE_ENABLE	Block Gap Event Status Enable 0x0: Masked 0x1: Enabled	RW	0
1	TC_ENABLE	Transfer Complete Status Enable 0x0: Masked 0x1: Enabled	RW	0
0	CC_ENABLE	Command Complete Status Enable 0x0: Masked 0x1: Enabled	RW	0

Table 19-87. Register Call Summary for Register MMCHS_IE

SDIO Functional Description • Power Management: [0][1] • Interrupt Requests: [2][3][4][5][6][7][8][9][10][11][12][13][14][15][16][17][18][19][20][21] • Interrupt-Driven Operation: [22][23]
SDIO Programming Guide • Global Initialization: [24] • Operational Modes Configuration: [25][26]
SDIO Register Manual • SDIO Register Summary: [27] • SDIO Register Description: [28][29][30][31]

Table 19-88. MMCHS_ISE

Address Offset	0x0000 0238		
Physical Address	0x480D 1238	**Instance**	MMC
Description	Interrupt Signal Enable Register This register allows to enable/disable the module internal sources of status, on an event-by-event basis. MMCHS_ISE[31:16] = Error Interrupt Signal Enable MMCHS_ISE[15:0] = Normal Interrupt Signal Enable		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED		BADA_SIGEN	CERR_SIGEN	RESERVED			ACE_SIGEN	RESERVED	DEB_SIGEN	DCRC_SIGEN	DTO_SIGEN	CIE_SIGEN	CEB_SIGEN	CCRC_SIGEN	CTO_SIGEN	NULL	RESERVED							CIRQ_SIGEN	RESERVED		BRR_SIGEN	BWR_SIGEN	RESERVED	BGE_SIGEN	TC_SIGEN	CC_SIGEN

Bits	Field Name	Description	Type	Reset
31:30	RESERVED		R	0x0
29	BADA_SIGEN	Bad access to data space Signal Enable 0x0: Masked 0x1: Enabled	RW	0
28	CERR_SIGEN	Card Error Interrupt Signal Enable 0x0: Masked 0x1: Enabled	RW	0
27:25	RESERVED		R	0x0
24	ACE_SIGEN	Auto CMD Error Signal Enable 0x0: Masked 0x1: Enabled	RW	0
23	RESERVED		R	0
22	DEB_SIGEN	Data End Bit Error Signal Enable 0x0: Masked 0x1: Enabled	RW	0
21	DCRC_SIGEN	Data CRC Error Signal Enable 0x0: Masked 0x1: Enabled	RW	0
20	DTO_SIGEN	Data Timeout Error Signal Enable 0x0: Masked 0x1: Enabled	RW	0

Bits	Field Name	Description	Type	Reset
19	CIE_SIGEN	Command Index Error Signal Enable 0x0: Masked 0x1: Enabled	RW	0
18	CEB_SIGEN	Command End Bit Error Signal Enable 0x0: Masked 0x1: Enabled	RW	0
17	CCRC_SIGEN	Command CRC Error Signal Enable 0x0: Masked 0x1: Enabled	RW	0
16	CTO_SIGEN	Command timeout Error Signal Enable 0x0: Masked 0x1: Enabled	RW	0
15	NULL	Fixed to 0 The host driver shall control error interrupts using the Error Interrupt Signal Enable register. Writes to this bit are ignored	R	0
14:9	RESERVED		R	0x0
8	CIRQ_SIGEN	Card Interrupt Signal Enable 0x0: Masked 0x1: Enabled	RW	0
7:6	RESERVED		R	0x0
5	BRR_SIGEN	Buffer Read Ready Signal Enable 0x0: Masked 0x1: Enabled	RW	0
4	BWR_SIGEN	Buffer Write Ready Signal Enable 0x0: Masked 0x1: Enabled	RW	0
3	RESERVED		R	0
2	BGE_SIGEN	Black Gap Event Signal Enable 0x0: Masked 0x1: Enabled	RW	0
1	TC_SIGEN	Transfer Completed Status Enable 0x0: Masked 0x1: Enabled	RW	0
0	CC_SIGEN	Command Complete Status Enable 0x0: Masked 0x1: Enabled	RW	0

Table 19-89. Register Call Summary for Register MMCHS_ISE

SDIO Functional Description • Power Management: [0][1] • Interrupt Requests: [2][3][4][5][6] • Polling: [7]
SDIO Programming Guide • Operational Modes Configuration: [8]
SDIO Register Manual • SDIO Register Summary: [9] • SDIO Register Description: [10][11][12]

Table 19-90. MMCHS_AC12

Address Offset	0x0000 023C		
Physical Address	0x480D 123C	**Instance**	MMC
Description	Host Control 2 Register and Auto CMD Error Status Register This register is used to indicate CMD12 response error of Auto CMD12. The Host driver can determine what kind of Auto CMD12 errors occur by this register. Bits[7:0] are valid only when the MMCHS_CMD[3:2] ACEN bitfield is configured to enable Auto CMD and the Auto CMD Error bit (MMCHS_STAT[24]ACE) is set.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PV_ENABLE	AI_ENABLE	RESERVED								DS_SEL		V1V8_SIGEN	UHSMS			RESERVED								CNI	RESERVED		ACIE	ACEB	ACCE	ACTO	ACNE

Bits	Field Name	Description	Type	Reset
31	PV_ENABLE	Preset Value Enable Host Controller Version 3.00 supports this bit. As the operating SDCLK frequency and I/O driver strength depend on the Host System implementation, it is difficult to determine these parameters in the Standard Host Driver. When Preset Value Enable is set, automatic SDCLK frequency generation and driver strength selection is performed without considering system specific conditions. This bit enables the functions defined in the Preset Value registers, see, Table 19-15. If this bit is set to 0, MMCHS_SYSCTL[15:6] CLKD, MMCHS_SYSCTL[5] CGS and MMCHS_AC12[21:20] DS_SEL are set by Host Driver. If this bit is set to 1, MMCHS_SYSCTL[15:6] CLKD, MMCHS_SYSCTL[5] CGS and MMCHS_AC12[21:20] DS_SEL are set by Host Controller as specified in the Preset Value registers, see, Table 19-15. 0x0: SDCLK and Driver Strength (DS_SEL) are controlled by Host Driver. 0x1: Automatic Selection by Preset Value are Enabled.	RW	0
30	AI_ENABLE	Asynchronous Interrupt Enable This bit can be set to 1 if a card supports asynchronous interrupts and MMCHS_CAPA[29] AIS is set to 1. Asynchronous interrupt is effective when DAT[1] interrupt is used in 4-bit SD mode (and zero is set to Interrupt Pin Select in the Shared Bus Control register). If this bit is set to 1, the Host Driver can stop the SDCLK during asynchronous interrupt period to save power. During this period, the Host Controller continues to deliver the Card Interrupt to the host when it is asserted by the Card. 0x0: Disabled 0x1: Enabled	RW	0
29:22	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
21:20	DS_SEL	Driver Strength Select Host Controller output driver in 1.8V signaling is selected by this bit. In 3.3V signaling, this field is not effective. This field can be set depending on Driver Type A, C and D support bits (DTA, DTC and DTD respectively) in the MMCHS_CAPA2 register. This bit depends on setting of Preset Value Enable. If Preset Value Enable = 0, this field is set by Host Driver. If Preset Value Enable = 1, this field is automatically set by a value specified in the one of Preset Value registers, see, Table 19-15. 0x0: Driver Type B is selected (Default) 0x1: Driver Type A is selected 0x3: Driver Type D is selected 0x2: Driver Type C is selected	RW	0x0
19	V1V8_SIGEN	1.8V Signaling Enable This bit controls voltage regulator for I/O cell. 3.3V is supplied to the card regardless of signaling voltage. Setting this bit from 0 to 1 starts changing signal voltage from 3.3V to 1.8V. 1.8V regulator output shall be stable within 5ms. Host Controller clears this bit if switching to 1.8V signaling fails. Clearing this bit from 1 to 0 starts changing signal voltage from 1.8V to 3.3V. 3.3V regulator output shall be stable within 5ms. Host Driver can set this bit to 1 when Host Controller supports 1.8V signaling (One of support bits is set to 1: SDR50, SDR104 or DDR50 in MMCHS_CAPA2 register) and the card or device supports UHS-I (S18A=1. Refer to Bus Signal Voltage Switch Sequence in the Physical Layer Specification Version 3.0x). 0x0: 3.3V Signaling 0x1: 1.8V Signaling	RW	0
18:16	UHSMS	UHS Mode Select This field is used to select one of UHS-I modes and effective when 1.8V Signaling Enable is set to 1. If MMCHS_AC12[31] PV_ENABLE is set to 1, Host Controller sets MMCHS_SYSCTL[15:6] CLKD, MMCHS_SYSCTL[5] CGS and MMCHS_AC12[21:20] DS_SEL according to Preset Value registers, see, Table 19-15. In this case, one of preset value registers is selected by this field. Host Driver needs to reset MMCHS_SYSCTL[2] CEN before changing this field to avoid generating clock glitch. After setting this field, Host Driver sets MMCHS_SYSCTL[2] CEN again. 0x0: SDR12 0x1: SDR25 0x2 to 0x7: Reserved	RW	0x0
15:8	RESERVED		R	0x00
7	CNI	Command Not Issued By Auto CMD12 Error Setting this bit to 1 means CMD_wo_DAT is not executed due to an Auto CMD12 Error (D04-D01) in this register. Read 0x1: Command not issued Read 0x0: No error	R	0
6:5	RESERVED		R	0x0
4	ACIE	Auto CMD Index Error for Auto CMD12 This bit is set if the Command Index error occurs in response to a command. Read 0x1: Error Read 0x0: No error	R	0

Bits	Field Name	Description	Type	Reset
3	ACEB	Auto CMD End Bit Error for Auto CMD12 This bit is set when detecting that the end bit of command response is 0. Read 0x1: End bit Error Generated Read 0x0: No error	R	0
2	ACCE	Auto CMD CRC Error for Auto CMD12 This bit is set when detecting a CRC error in the command response. Read 0x1: CRC Error Generated Read 0x0: No error	R	0
1	ACTO	Auto CMD Timeout Error for Auto CMD12 This bit is set if no response is returned within 64 SDCLK cycles from the end bit of command. If this bit is set to1, the other error status bits (D04-D02) are meaningless. Read 0x1: Auto CMD Time Out Read 0x0: No error	R	0
0	ACNE	Auto CMD12 Not Executed If memory multiple block data transfer is not started due to command error, this bit is not set because it is not necessary to issue Auto CMD12. Setting this bit to 1 means the Host Controller cannot issue Auto CMD12 to stop memory multiple block data transfer due to some error. If this bit is set to 1, other error status bits (D04-D01) are meaningless. Read 0x1: Auto CMD12 Not Executed Read 0x0: Auto CMD12 Executed	R	0

Table 19-91. Register Call Summary for Register MMCHS_AC12

SDIO Functional Description • Interrupt Requests: [0] • Asynchronous Interrupt: [1][2]
SDIO Programming Guide • Operational Modes Configuration: [3][4]
SDIO Register Manual • SDIO Register Summary: [5] • SDIO Register Description: [6][7][8][9][10][11][12][13][14]

Table 19-92. MMCHS_CAPA

Address Offset	0x0000 0240		
Physical Address	0x480D 1240	**Instance**	MMC
Description	Capabilities Register This register lists the capabilities of the SDIO host controller.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED		AIS	BIT64	RESERVED	VS18	VS30	VS33	SRS	DS	HSS	RESERVED	AD2S	RESERVED	MBL		BCF								TCU	RESERVED			TCF			

Bits	Field Name	Description	Type	Reset
31:30	RESERVED		R	0x0
29	AIS	Asynchronous Interrupt Support Refer to SDIO Specification Version 3.00 about asynchronous interrupt. Read 0x1: Asynchronous Interrupt Supported Read 0x0: Asynchronous Interrupt Not Supported	R	1
28	BIT64	64 Bit System Bus Support Setting 1 to this bit indicates that the Host Controller supports 64-bit address descriptor mode and is connected to 64-bit address system bus. Read 0x1: 64 bit System bus address Read 0x0: 32 bit System bus address	R	0
27	RESERVED		R	0
26	VS18	Voltage support 1.8V Initialization of this register (via a write access to this register) depends on the system capabilities. The host driver shall not modify this register after the initilaization. This register is only reinitialized by a hard reset (via RESETN signal) Write 0x0: 1.8V Not supported Write 0x1: 1.8V Supported Read 0x1: 1.8V Supported Read 0x0: 1.8V Not Supported	RW	0
25	VS30	Voltage support 3.0V Initialization of this register (via a write access to this register) depends on the system capabilities. The host driver shall not modify this register after the initilaization. This register is only reinitialized by a hard reset (via RESETN signal) Write 0x0: 3.0V Not supported Write 0x1: 3.0V Supported Read 0x1: 3.0V Supported Read 0x0: 3.0V Not Supported	RW	0
24	VS33	Voltage support 3.3V Initialization of this register (via a write access to this register) depends on the system capabilities. The host driver shall not modify this register after the initilaization. This register is only reinitialized by a hard reset (via RESETN signal) Write 0x0: 3.3V Not supported Write 0x1: 3.3V Supported Read 0x1: 3.3V Supported Read 0x0: 3.3V Not Supported	RW	0
23	SRS	Suspend/Resume support (SDIO cards only) This bit indicates whether the host controller supports Suspend/Resume functionality. Read 0x1: The Host controller supports Suspend/Resume functionality. Read 0x0: The Host controller does not Suspend/Resume functionality.	R	1
22	DS	DMA support This bit indicates that the Host Controller is able to use DMA to transfer data between system memory and the Host Controller directly. Read 0x1: DMA Supported Read 0x0: DMA Not Supported	R	1

Bits	Field Name	Description	Type	Reset
21	HSS	High speed support This bit indicates that the host controller supports high speed operations and can supply an up-to maximum card frequency. Read 0x1: High Speed Supported Read 0x0: High Speed Not Supported **NOTE: High Speed modes are supported, but MMCHS_HCTL[HSPE] bit must always be set to 0x0 because device was timing closed with HSPE bit set to 0x0 for all supported modes of operation.**	R	1
20	RESERVED		R	0
19	AD2S	ADMA2 Support This bit indicates whether the Host Controller is capable of using ADMA2. It depends on setting of generic parameter MADMA_EN Read 0x1: ADMA2 Supported Read 0x0: ADMA2 not Supported	R	0
18	RESERVED		R	0
17:16	MBL	Maximum block length This value indicates the maximum block size that the host driver can read and write to the buffer in the host controller. This value depends on definition of generic parameter with a max value of 2048 bytes. The host controller supports 512 bytes and 1024 bytes block transfers. Read 0x2: 2048 bytes Read 0x1: 1024 bytes Read 0x0: 512 bytes	R	0x1
15:8	BCF	Base Clock Frequency For SD Clock This value indicates the base (maximum) clock frequency for the SD Clock. 8-bit Base Clock Frequency This mode is supported by the Host Controller Version 3.00. Unit values are 1MHz. The supported clock range is 10MHz to 255MHz. FFh : 255MHz .... : ..... 02h : 2MHz 01h : 1MHz 00h : Get information via another method If the real frequency is 16.5MHz, the lager value shall be set 0001 0001b (17MHz) because the Host Driver use this value to calculate the clock divider value (Refer to MMCHS_SYSCTL[15:6] CLKD) and it shall not exceed upper limit of the SD Clock frequency. If these bits are all 0, the Host System has to get information via another method. Read 0x0: The value indicating the base (maximum) frequency for the output clock provided to the card is system dependent and is not available in this register. Get the information via another method.	R	0x00
7	TCU	Timeout clock unit This bit shows the unit of base clock frequency used to detect Data Timeout Error (MMCHS_STAT[DTO]). Read 0x1: MHz Read 0x0: KHz	R	1
6	RESERVED		R	0

Bits	Field Name	Description	Type	Reset
5:0	TCF	Timeout clock frequency The timeout clock frequency is used to detect Data Timeout Error (MMCHS_STAT[DTO]). Read 0x0: The timeout clock frequency depends on the frequency of the clock provided to the card. The value of the timeout clock frequency is not available in this register.	R	0x00

Table 19-93. Register Call Summary for Register MMCHS_CAPA

SDIO Functional Description • Software Reset: [0] • Asynchronous Interrupt: [1] • Data Buffer: [2]
SDIO Programming Guide • Global Initialization: [3]
SDIO Register Manual • SDIO Register Summary: [4] • SDIO Register Description: [5][6][7][8][9][10][11]

Table 19-94. MMCHS_CAPA2

Address Offset	0x0000 0244		
Physical Address	0x480D 1244	**Instance**	MMC
Description	Capabilities 2 Register This register provides the Host Driver with information specific to the Host Controller implementation. The Host Controller may implement these values as fixed or loaded from flash memory during power on initialization. Refer to Software Reset For All in the Software Reset register for loading from flash memory and completion timing control.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								CM								RTM		TSDR50	RESERVED	TCRT				RESERVED	DTD	DTC	DTA	RESERVED	DDR50	SDR104	SDR50

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x00
23:16	CM	Clock Multiplier This field indicates clock multiplier value of programmable clock generator. Refer to MMCHS_SYSCTL [15:0]. Setting 00h means that Host Controller does not support programmable clock generator. 00h : Clock Multiplier is Not Supported 01h : Clock Multiplier M = 2 02h : Clock Multiplier M = 3 .... : ..... FFh : Clock Multiplier M = 256	R	0x00

Bits	Field Name	Description	Type	Reset
15:14	RTM	Re-Tuning Modes This field selects re-tuning method and limits the maximum data length. Bit47-46 Re-Tuning Mode Re-Tuning Method Data Length There are two re-tuning timings: Re-Tuning Request controlled by the Host Controller and expiration of a Re-Tuning Timer controlled by the Host Driver. By receiving either timing, the Host Driver executes the re-tuning procedure just before a next command issue. The maximum data length per read/write command is restricted so that re-tuning procedures can be inserted during data transfers. (1) Re-Tuning Mode 1 The host controller does not have any internal logic to detect when the re-tuning needs to be performed. In this case, the Host Driver should maintain all re-tuning timings by using a Re-Tuning Timer. To enable inserting the re-tuning procedure during data transfers, the data length per read/write command shall be limited up to 4 MiB. (2) Re-Tuning Mode 2 The host controller has the capability to indicate the re-tuning timing by Re-Tuning Request during data transfers. Then the data length per read/write command shall be limited up to 4 MiB. During non data transfer, re-tuning timing is determined by either Re-Tuning Request or Re-Tuning Timer. If Re-Tuning Request is used, Re-Tuning Timer should be disabled. (3) Re-Tuning Mode 3 The host controller has the capability to take care of the re-tuning during data transfer (Auto Re-Tuning). Re-Tuning Request shall not be generated during data transfers and there is no limitation to data length per read/write command. During non data transfer, re-tuning timing is determined by either Re-Tuning Request or Re-Tuning Timer. If Re-Tuning Request is used, Re-Tuning Timer should be disabled. Re-Tuning Timer Control Example for Re-Tuning Mode 1 The initial value of re-tuning timer is provided by Timer Count for Re-Tuning field in this register. The timer starts counting by loading the initial value. When the timer expires, the Host Driver marks an expiration flag. On receiving a command request, the Host driver checks the expiration flag. If the expiration flag is set, then the Host Driver should perform the re-tuning procedure before issuing a command. If the expiration flag is not set, then the Host Driver issues a command without performing the re-tuning procedure. Every time the re-tuning procedure is performed, the timer loads the new initial value and the expiration flag is cleared. Re-Tuning Timer Control Example for Re-Tuning Mode 2 and Mode 3 The timer control is almost the same as Re-Tuning Mode 1 except the timer loads the new initial value after data transfer (when receiving Transfer Complete). In case of Mode 3, Timer Count for Re-Tuning is set either smaller value: Tuning effective time after re-tuning procedure or after data transfer. If a Host System goes into power down mode, the Host Driver should stop the re-tuning timer and set the expiration flag to 1 when the Host System resumes from power down mode. Read 0x3: Reserved Read 0x2: Auto Re-Tuning (for transfer) - Timer and Re-Tuning Request Read 0x1: Timer and Re-Tuning Request - Max data length 4 MiB Read 0x0: Timer - Max data length 4 MiB	R	0x0

Bits	Field Name	Description	Type	Reset
13	TSDR50	Use Tuning for SDR50 If this bit is set to 1, this Host Controller requires tuning to operate SDR50. Read 0x1: SDR50 requires tuning. Read 0x0: SDR50 does not require tuning.	R	0
12	RESERVED		R	0
11:8	TCRT	Timer Count for Re-Tuning This field indicates an initial value of the Re-Tuning Timer for Re-Tuning Mode 1 to 3. Setting to 0 disables Re-Tuning Timer. Read 0x3: 4 seconds Read 0xE: Reserved Read 0xC: Reserved Read 0x4: 8 seconds Read 0xB: 1024 seconds Read 0xF: Get information from other source Read 0x2: 2 seconds Read 0x0: Re-Tuning Timer disabled Read 0xA: 512 seconds Read 0x6: 32 seconds Read 0x1: 1 second Read 0x8: 128 seconds Read 0x7: 64 seconds Read 0x9: 256 seconds Read 0xD: Reserved Read 0x5: 16 seconds	R	0xF
7	RESERVED		R	0
6	DTD	Driver Type D Support This bit indicates support of Driver Type D for 1.8 Signaling. Read 0x1: Driver Type D is Supported Read 0x0: Driver Type D is Not Supported.	R	1
5	DTC	Driver Type C Support This bit indicates support of Driver Type C for 1.8 Signaling. Read 0x1: Driver Type C is Supported. Read 0x0: Driver Type C is Not Supported.	R	1
4	DTA	Driver Type A Support This bit indicates support of Driver Type A for 1.8 Signaling. Read 0x1: Driver Type A is Supported. Read 0x0: Driver Type A is Not Supported.	R	1
3	RESERVED		R	0
2	DDR50	DDR50 Support Read 0x1: DDR50 is Supported. Read 0x0: DDR50 is Not Supported.	R	0x0
1	SDR104	SDR104 Support Read 0x1: SDR104 is Supported. Read 0x0: SDR104 is Not Supported.	R	0x0

Bits	Field Name	Description	Type	Reset
0	SDR50	SDR50 Support If SDR104 is supported, this bit shall be set to 1. Bit 13 indicates whether SDR50 requires tuning or not. Read 0x1: SDR50 is Supported. Read 0x0: SDR50 is Not Supported.	R	0x0

Table 19-95. Register Call Summary for Register MMCHS_CAPA2

SDIO Register Manual
- SDIO Register Summary: [0]
- SDIO Register Description: [1][2][3]

Table 19-96. MMCHS_CUR_CAPA

Address Offset	0x0000 0248		
Physical Address	0x480D 1248	**Instance**	MMC
Description	Maximum Current Capabilities Register This register indicates the maximum current capability for each voltage. The value is meaningful if the voltage support is set in the capabilities register (MMCHS_CAPA). Initialization of this register (via a write access to this register) depends on the system capabilities. The host driver shall not modify this register after the initilaization. This register is only reinitialized by a hard reset (via RESETN signal)		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								CUR_1V8								CUR_3V0								CUR_3V3							

Bits	Field Name	Description	Type	Reset
31:24	RESERVED		R	0x00
23:16	CUR_1V8	Maximum current for 1.8V Read 0x0: The maximum current capability for this voltage is not available. Feature not implemented.	RW	0x00
15:8	CUR_3V0	Maximum current for 3.0V Read 0x0: The maximum current capability for this voltage is not available. Feature not implemented.	RW	0x00
7:0	CUR_3V3	Maximum current for 3.3V Read 0x0: The maximum current capability for this voltage is not available. Feature not implemented.	RW	0x00

Table 19-97. Register Call Summary for Register MMCHS_CUR_CAPA

SDIO Functional Description
- Software Reset: [0]

SDIO Programming Guide
- Global Initialization: [1]

SDIO Register Manual
- SDIO Register Summary: [2]
- SDIO Register Description: [3]

Table 19-98. MMCHS_FE

Address Offset	0x0000 0250		
Physical Address	0x480D 1250	**Instance**	MMC
Description	Force Event Register for Auto CMD Error Status and Error Interrupt status The Force Event Register is not a physically implemented register. Rather, it is an address at which the Auto CMD Error Status Register (MMCHS_AC12) can be written. Writing 1 : set each bit of the Auto CMD Error Status Register Writing 0 : no effect Rather, it is an address at which the Error Interrupt Status register can be written. The effect of a write to this address will be reflected in the Error Interrupt Status Register if the corresponding bit of the Error Interrupt Status Enable Register is set. Writing 1 : set each bit of the Error Interrupt Status Register Writing 0 : no effect Note: By setting this register, the Error Interrupt can be set in the Error Interrupt Status register. In order to generate interrupt signal, both the Error Interrupt Status Enable and Error Interrupt Signal Enable shall be set.		
Type	W		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED		FE_BADA	FE_CERR	RESERVED			FE_ACE	RESERVED	FE_DEB	FE_DCRC	FE_DTO	FE_CIE	FE_CEB	FE_CCRC	FE_CTO	RESERVED								FE_CNI	RESERVED		FE_ACIE	FE_ACEB	FE_ACCE	FE_ACTO	FE_ACNE

Bits	Field Name	Description	Type	Reset
31:30	RESERVED		NA	0x0
29	FE_BADA	Force Event Bad access to data space. Write 0x0: No effect, No Interrupt. Write 0x1: Interrupt Forced	W	0
28	FE_CERR	Force Event Card error. Write 0x0: No effect, No Interrupt. Write 0x1: Interrupt Forced	W	0
27:25	RESERVED		NA	0x0
24	FE_ACE	Force Event for Auto CMD Error - For Auto CMD12 Write 0x0: No effect, No Interrupt. Write 0x1: Interrupt Forced	W	0
23	RESERVED		NA	0
22	FE_DEB	Force Event Data End Bit error. Write 0x0: No effect, No Interrupt. Write 0x1: Interrupt Forced	W	0
21	FE_DCRC	Force Event Data CRC Error. Write 0x0: No effect, No Interrupt. Write 0x1: Interrupt Forced	W	0
20	FE_DTO	Force Event Data Timeout Error. Write 0x0: No effect, No Interrupt. Write 0x1: Interrupt Forced	W	0
19	FE_CIE	Force Event Command Index Error. Write 0x0: No effect, No Interrupt. Write 0x1: Interrupt Forced	W	0
18	FE_CEB	Force Event Command End Bit Error. Write 0x0: No effect, No Interrupt. Write 0x1: Interrupt Forced	W	0

Bits	Field Name	Description	Type	Reset
17	FE_CCRC	Force Event Command CRC Error. Write 0x0: No effect, No Interrupt. Write 0x1: Interrupt Forced	W	0
16	FE_CTO	Command Timeout Error This bit is set automatically when no response is received within 64 clock cycles from the end bit of the command. For commands that reply within 5 clock cycles - the timeout is still detected at 64 clock cycles. Write 0x0: Status bit unchanged Write 0x1: Status is cleared	W	0
15:8	RESERVED		NA	0x00
7	FE_CNI	Force Event Command not issued by Auto CMD12 error Write 0x0: No effect, No Interrupt. Write 0x1: Interrupt Forced	W	0
6:5	RESERVED		NA	0x0
4	FE_ACIE	Force Event for Auto CMD Index Error - For Auto CMD12 Write 0x0: No effect, No Interrupt. Write 0x1: Interrupt Forced	W	0
3	FE_ACEB	Force Event Auto CMD End Bit Error Write 0x0: No effect, No Interrupt. Write 0x1: Interrupt Forced	W	0
2	FE_ACCE	Force Event Auto CMD CRC Error Write 0x0: No effect, No Interrupt. Write 0x1: Interrupt Forced	W	0
1	FE_ACTO	Force Event Auto CMD Timeout Error Write 0x0: No effect, No Interrupt. Write 0x1: Interrupt Forced	W	0
0	FE_ACNE	Force Event Auto CMD12 Not Executed Write 0x0: No effect, No Interrupt. Write 0x1: Interrupt Forced	W	0

Table 19-99. Register Call Summary for Register MMCHS_FE

SDIO Functional Description • Test Registers: [0]
SDIO Register Manual • SDIO Register Summary: [1]

Table 19-100. MMCHS_PVINITSD

Address Offset	0x0000 0260		
Physical Address	0x480D 1260	**Instance**	MMC
Description	Preset Value for Initialization and Default Speed modes		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DSDS_SEL		RESERVED			DSCLKGEN_SEL			DSSDCLK_SEL								INITDS_SEL		RESERVED			INITCLKGEN_SEL			INITSDCLK_SEL							

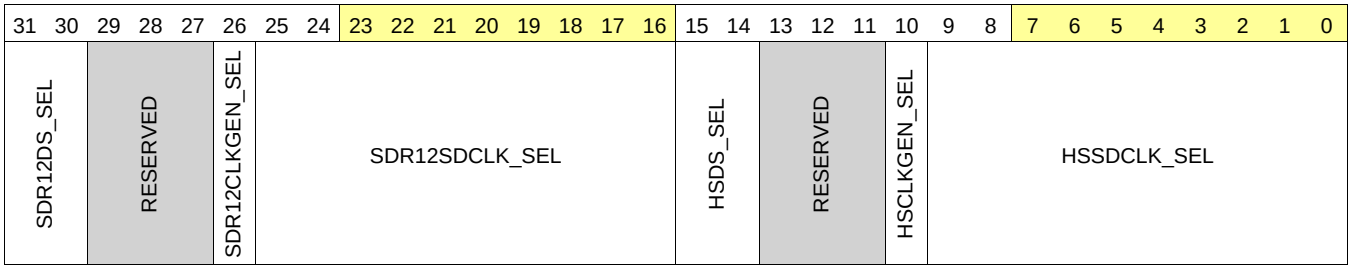
Bits	Field Name	Description	Type	Reset
31:30	DSDS_SEL	Driver Strength Select Value - Default Speed mode Driver Strength is supported by 1.8V signaling bus speed modes. This field is meaningless for 3.3V signaling. Read 0x3: Driver Type D is Selected. Read 0x2: Driver Type C is Selected. Read 0x1: Driver Type A is Selected. Read 0x0: Driver Type B is Selected.	R	0x0
29:27	RESERVED		R	0x0
26	DSCLKGEN_SEL	Clock Generator Select Value - Default Speed mode This bit is effective when Host Controller supports programmable clock generator. Read 0x1: Programmable Clock Generator. Read 0x0: Host Controller Ver2.00 Compatible Clock Generator.	R	0
25:16	DSSDCLK_SEL	SDCLK Frequency Select Value - Default Speed mode 10-bit preset value to set MMCHS_SYSCTL[15:6] CLKD is described by a host system.	R	0x004
15:14	INITDS_SEL	Driver Strength Select Value - Initialization mode Driver Strength is supported by 1.8V signaling bus speed modes. This field is meaningless for 3.3V signaling. Read 0x3: Driver Type D is Selected Read 0x2: Driver Type C is Selected Read 0x1: Driver Type A is Selected Read 0x0: Driver Type B is Selected	R	0x0
13:11	RESERVED		R	0x0
10	INITCLKGEN_SEL	Clock Generator Select Value - Initialization mode This bit is effective when Host Controller supports programmable clock generator. Read 0x1: Programmable Clock Generator. Read 0x0: Host Controller Ver2.00 Compatible Clock Generator.	R	0
9:0	INITSDCLK_SEL	SDCLK Frequency Select Value - Initialization mode 10-bit preset value to set MMCHS_SYSCTL[15:6] CLKD is described by a host system.	R	0x1E0

Table 19-101. Register Call Summary for Register MMCHS_PVINITSD

SDIO Functional Description • SDIO Hardware Status Features: [0]
SDIO Register Manual • SDIO Register Summary: [1]

Table 19-102. MMCHS_PVHSSDR12

Address Offset	0x0000 0264		
Physical Address	0x480D 1264	**Instance**	MMC
Description	Preset Value for High Speed and SDR12 speed modes		
Type	R		



Bits	Field Name	Description	Type	Reset
31:30	SDR12DS_SEL	Driver Strength Select Value - SDR12 mode Driver Strength is supported by 1.8V signaling bus speed modes. This field is meaningless for 3.3V signaling. Read 0x3: Driver Type D is Selected. Read 0x2: Driver Type C is Selected. Read 0x1: Driver Type A is Selected. Read 0x0: Driver Type B is Selected.	R	0x0
29:27	RESERVED		R	0x0
26	SDR12CLKGEN_SEL	Clock Generator Select Value - SDR12 mode This bit is effective when Host Controller supports programmable clock generator. Read 0x1: Programmable Clock Generator. Read 0x0: Host Controller Ver2.00 Compatible Clock Generator.	R	0
25:16	SDR12SDCLK_SEL	SDCLK Frequency Select Value - SDR12 mode 10-bit preset value to set MMCHS_SYSCTL[15:6] CLKD is described by a host system.	R	0x004
15:14	HSDS_SEL	Driver Strength Select Value - High Speed mode Driver Strength is supported by 1.8V signaling bus speed modes. This field is meaningless for 3.3V signaling. Read 0x3: Driver Type D is Selected. Read 0x2: Driver Type C is Selected. Read 0x1: Driver Type A is Selected. Read 0x0: Driver Type B is Selected.	R	0x0
13:11	RESERVED		R	0x0
10	HSCLKGEN_SEL	Clock Generator Select Value - High Speed mode This bit is effective when Host Controller supports programmable clock generator. Read 0x1: Programmable Clock Generator. Read 0x0: Host Controller Ver2.00 Compatible Clock Generator.	R	0
9:0	HSSDCLK_SEL	SDCLK Frequency Select Value - High Speed mode 10-bit preset value to set MMCHS_SYSCTL[15:6] CLKD is described by a host system.	R	0x002

Table 19-103. Register Call Summary for Register MMCHS_PVHSSDR12

SDIO Functional Description • SDIO Hardware Status Features: [0]
SDIO Register Manual • SDIO Register Summary: [1]

Table 19-104. MMCHS_PVSDR25SDR50

Address Offset	0x0000 0268		
Physical Address	0x480D 1268	**Instance**	MMC
Description	Preset Value for SDR25 speed mode		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																SDR25DS_SEL		RESERVED			SDR25CLKGEN_SEL	SDR25SDCLK_SEL									

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x1
15:14	SDR25DS_SEL	Driver Strength Select Value - SDR25 mode Driver Strength is supported by 1.8V signaling bus speed modes. This field is meaningless for 3.3V signaling. Read 0x3: Driver Type D is Selected. Read 0x2: Driver Type C is Selected. Read 0x1: Driver Type A is Selected. Read 0x0: Driver Type B is Selected.	R	0x0
13:11	RESERVED		R	0x0
10	SDR25CLKGEN_SEL	Clock Generator Select Value - SDR25 mode This bit is effective when Host Controller supports programmable clock generator. Read 0x1: Programmable Clock Generato. Read 0x0: Host Controller Ver2.00 Compatible Clock Generator.	R	0
9:0	SDR25SDCLK_SEL	SDCLK Frequency Select Value - SDR25 mode 10-bit preset value to set MMCHS_SYSCTL[15:6] CLKD is described by a host system.	R	0x002

Table 19-105. Register Call Summary for Register MMCHS_PVSDR25SDR50

SDIO Functional Description • SDIO Hardware Status Features: [0]
SDIO Register Manual • SDIO Register Summary: [1]

Table 19-106. MMCHS_REV

Address Offset	0x0000 02FC		
Physical Address	0x480D 12FC	**Instance**	MMC
Description	Versions Register This register contains the hard coded RTL vendor revision number, the version number of SD specification compliancy and a slot status bit. MMCHS_REV[31:16] = Host controller version MMCHS_REV[15:0] = Slot Interrupt Status		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
VREV								SREV								RESERVED															SIS

Bits	Field Name	Description	Type	Reset
31:24	VREV	Vendor Version Number: IP revision [7:4] Major revision [3:0] Minor revision Examples: 0x10 for 1.0 0x21 for 2.1	R	0x--
23:16	SREV	Specification Version Number This status indicates the Host Controller Spec. Version. The upper and lower 4-bits indicate the version. Read 0x3: Reserved Read 0x2: SD Host Specification Version 3.00. Read 0x1: SD Host Specification Version 2.00 - Including the feature of the ADMA and Test Register. Read 0x0: SD Host Specification Version 1.00.	R	0x02
15:1	RESERVED		R	0x0000
0	SIS	Slot Interrupt Status This status bit indicates the inverted state of interrupt signal for the module. By a power on reset or by setting a software reset for all (MMCHS_HCTL[SRA]), the interrupt signal shall be de-asserted and this status shall read 0.	R	0

Table 19-107. Register Call Summary for Register MMCHS_REV

SDIO Register Manual
- SDIO Register Summary: [0]
- SDIO Register Description: [1][2]

General-Purpose Interface

This chapter describes the general-purpose interface for the device.

Topic **Page**

20.1 General-Purpose Interface Overview

The general-purpose interface combines four general-purpose input/output (GPIO) banks.

Each GPIO module provides 32 dedicated general-purpose pins with input and output capabilities; thus, the general-purpose interface supports up to 128 (4 x 32) pins. Some of the pins may be reserved in this Device. For more information of the supported number of GPIO pins, see the device Data Manual.

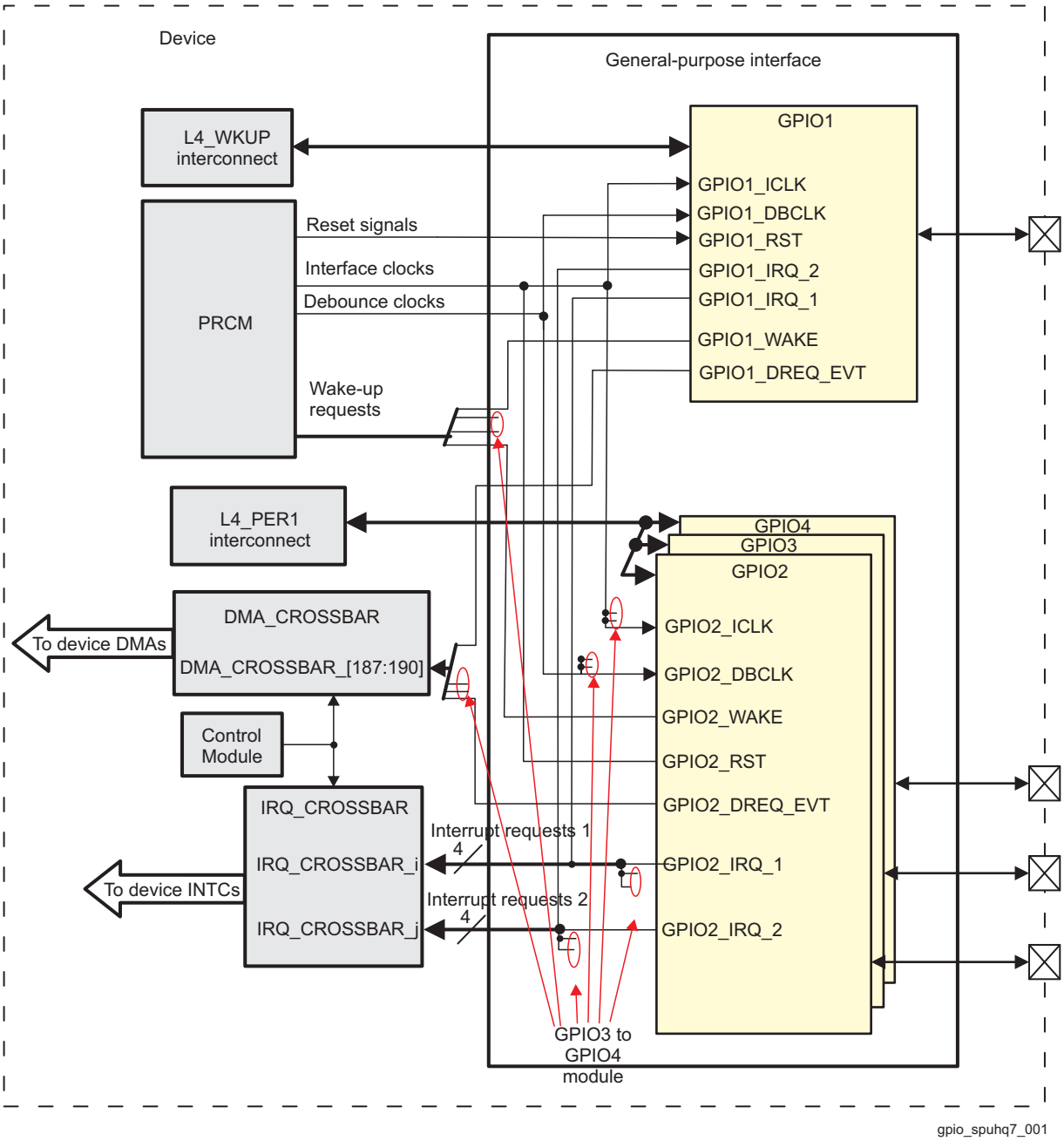
These pins can be configured for the following applications:

- Data input (capture)/output (drive)
- Keyboard interface with a debounce cell
- Interrupt generation in active mode upon the detection of external events. Detected events are processed by two parallel independent interrupt-generation submodules to support biprocessor operations.
- Wake-up request generation in idle mode upon the detection of external events

These modules do not include pad control (pullup/down control, open-drain feature). For more information, see Section 13.4.6.1, *Pad Configuration Registers,* in Chapter 13, *Control Module*.

Figure 20-1 is an overview of the general-purpose interface.

Figure 20-1. General-Purpose Interface Overview



The GPIO modules include the following global features:

- Two identical submodules can process synchronous interrupt requests from each channel to be used independently in a biprocessor environment. Each submodule controls its own synchronous interrupt request line. Each submodule also has its own interrupt-enable and interrupt status registers. The interrupt-enable register (GPIO_IRQSTATUS_SET_x [where x = 0 or 1]) selects the channel considered for the interrupt request generation. The interrupt status register (GPIO_IRQSTATUS_RAW_x) determines which channel has activated the interrupt request. Event detection on GPIO channels is reflected into GPIO_IRQSTATUS_RAW independently from the content of the interrupt-enable registers.
- Wake-up requests in idle mode from input channels are merged together to issue one wake-up signal per GPIO module.
- Data input (capture)/output (drive)

- Power-management support

The general-purpose interface has 8 interrupt lines (two interrupt lines on GPIO1 through GPIO4 modules).

Each GPIO module produces a wake-up request signal to the power, reset, and clock management (PRCM) module.

Each channel in the GPIO modules has the following features:

- The GPIOi.GPIO_OE register controls the output capability for each pin.
- The output line level reflects the value written in the GPIOi.GPIO_DATAOUT register through the level 4 (L4_WKUP and L4_PER1) interconnect.
- The input line can be fed to the GPIO module through an optional and configurable debounce cell. (Because the debouncing time value is global for all ports of one GPIO module, up to five different debouncing time values are possible.)
- The value of the input line is sampled into the GPIOi.GPIO_DATAIN register and can be read through the L4 (L4_WKUP and L4_PER1) interconnect.
- In active mode, the input line can be used through level and edge detectors to trigger synchronous interrupts. The edge (rising, falling, or both) or the level used (logical 0, logical 1, or both) can be configured.
- In idle mode, the input line can be used to activate the asynchronous wake-up request (on edge detection: rising edge, falling edge, or both).

The module provides an alternative to the atomic test and set operations for the data-output register (GPIO_DATAOUT). For this register, the module implements the set-and-clear protocol register update (see Section 20.4.9, *General-Purpose Interface Set-and-Clear Protocol*).

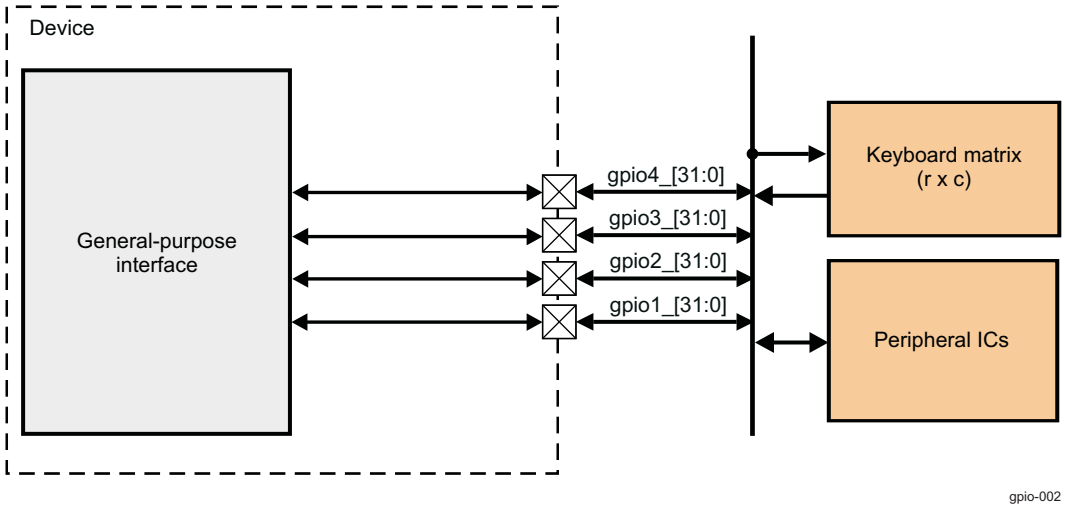
All module registers are 8-, 16-, or 32-bit accessible through the OCP-compatible interface (little-endian encoding)

20.2 General-Purpose Interface Environment

The general-purpose interface combines fourGPIO modules for a flexible, user-programmable, general-purpose input/output (I/O) controller. The general-purpose interface implements functions that are not implemented with the dedicated controllers in the device and require simple input and/or output software-controlled signals. The GPIO allows a variety of custom connections and expands the I/O capabilities of the system to the real world.

Figure 20-2 shows a typical application using the general-purpose interface.

Figure 20-2. General-Purpose Interface Typical Application

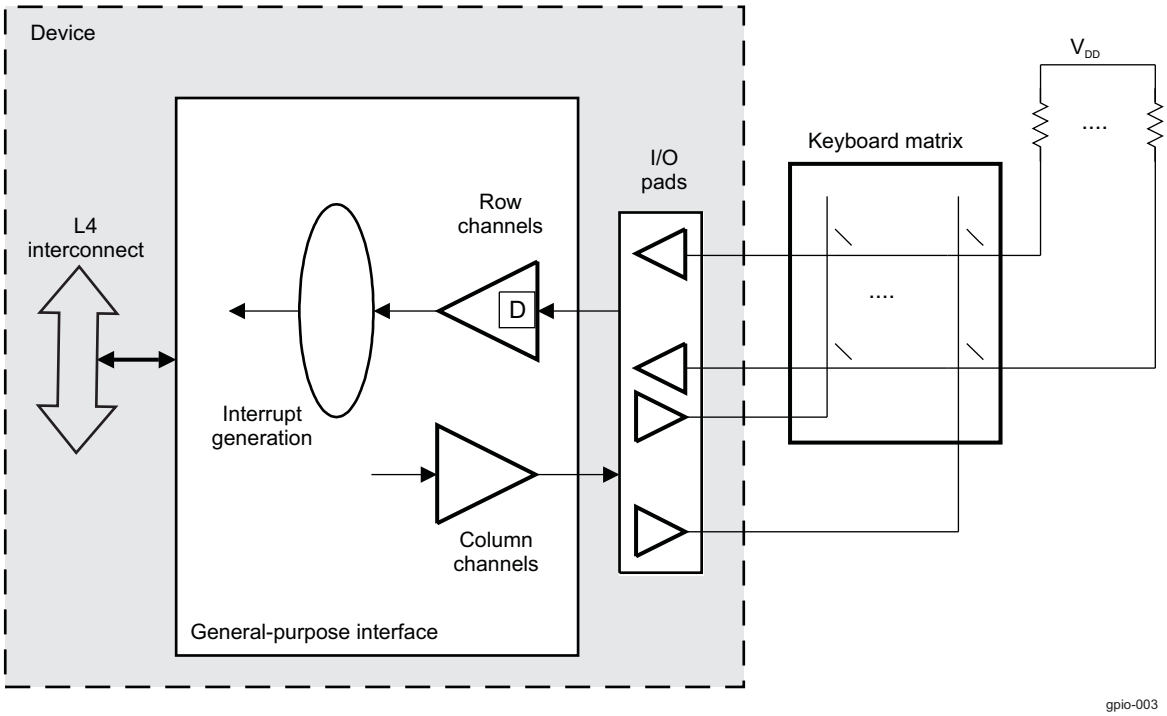


The general-purpose interface can physically connect the device to a keyboard matrix and peripheral integrated circuits (ICs).

20.2.1 General-Purpose Interface as a Keyboard Interface

The general-purpose interface can be used as a keyboard interface. Channels can be dedicated based on the keyboard matrix (r × c). Figure 20-3 shows row channels configured as inputs with the input debounce feature enabled. The row channels are driven high with an external pullup. Column channels are configured as outputs and drive a low level.

Figure 20-3. General-Purpose Interface Used as a Keyboard Interface



When a keyboard matrix key is pressed, the corresponding row and column lines are shorted together and a low level is driven on the corresponding row channel. This generates an interrupt based on the proper configuration (see Section 20.4.6, *Interrupt and Wake-up Requests*).

When the keyboard interrupt is received, the processor (digital signal processor [DSP] subsystem) can disable the keyboard interrupt and scan the column channels for the key coordinates.

- The scanning sequence has as many states as column channels: For each step in the sequence, the processor drives one column channel low and the others high.
- The processor reads the values of the row channels and thus detects which keys in the column are pressed.

At the end of the scanning sequence, the processor establishes which keys are pressed. The keyboard interface can then be reconfigured in the interrupt waiting state.

20.2.2 General-Purpose Interface Signals

Table 20-1 describes the module signals.

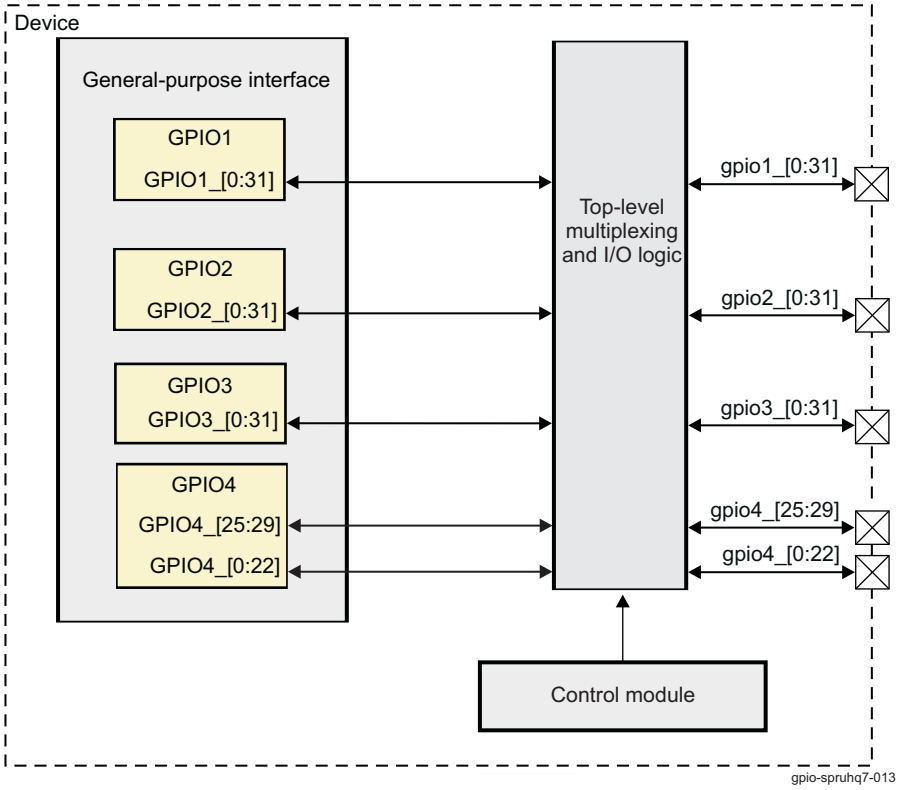
Table 20-1. I/O Description

Signal	I/O[1]	Description	Reset Value[2]
gpio1_[0:31]	I/O	GPIO	Hi-Z
gpio2_[0:31]	I/O	GPIO	Hi-Z
gpio3_[0:31]	I/O	GPIO	Hi-Z
gpio4_[0:22]	I/O	GPIO	Hi-Z
gpio4_[25:29]	I/O	GPIO	Hi-Z

(1) I = Input; O = Output; I/O = Bidirectional
(2) Hi-Z = High Impedance

Figure 20-4 shows the signal connections of GPIO1 through GPIO4.

Figure 20-4. GPIO1 Through GPIO4 Signal Connections



NOTE: For more information about the GPIO1 through GPIO4 signals and channel description, see Section 20.4.7, *General-Purpose Interface Channels Description.*

NOTE: For more information about GPIO signal multiplexing, see Section 13.4.6.1, *Pad Configuration Registers, and* in Chapter 13, *Control Module*.

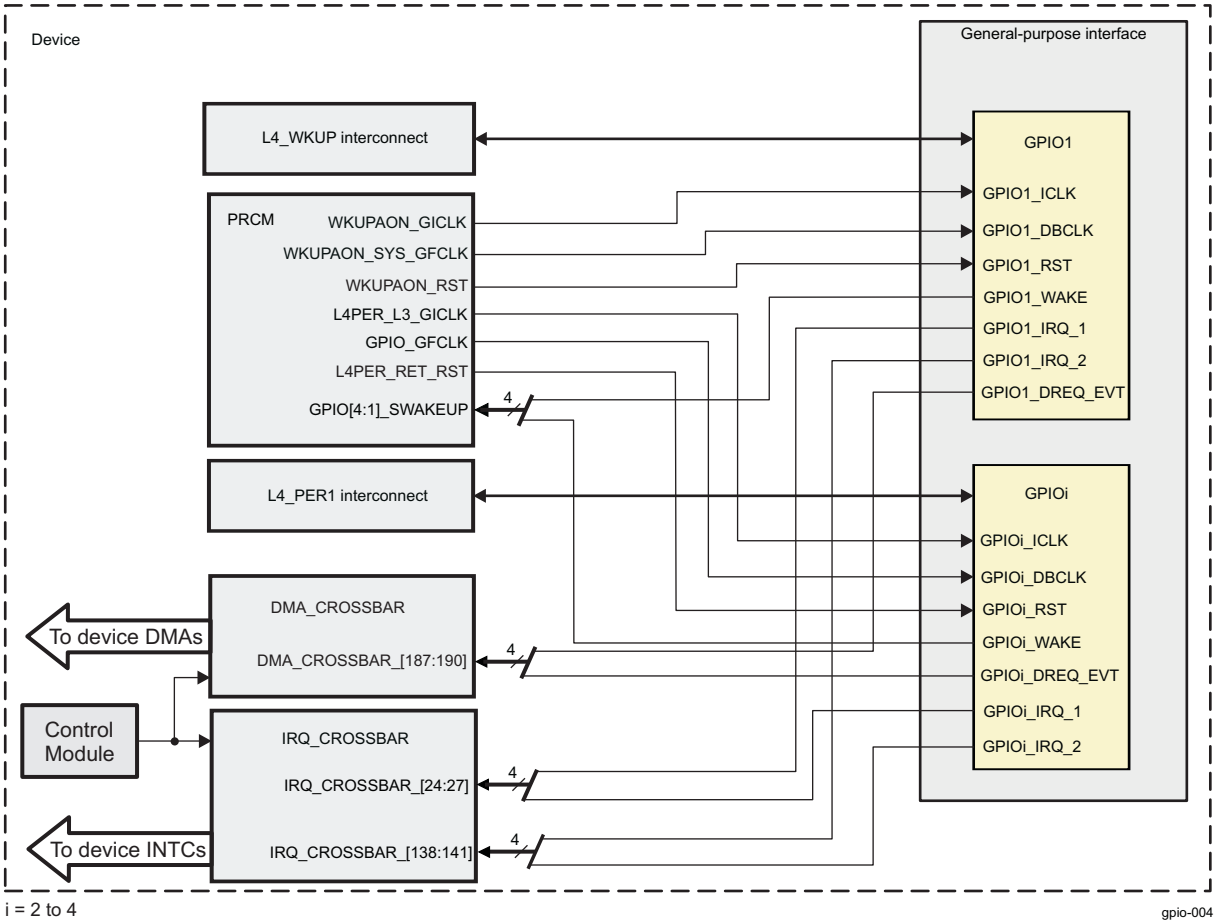
NOTE: GPIO pins can directly snoop device pads even if those are configured in the other modes. For GPIO output user needs to set pad mux.

20.3 General-Purpose Interface Integration

This section describes module integration in the device, including information about clocks, resets, and hardware requests.

Figure 20-5 shows this module integration.

Figure 20-5. GPIO Integration



i = 2 to 4
IRQ_CROSSBAR_[24:27] - GPIO1 to GPIO4 interrupt line 1
IRQ_CROSSBAR_[138:141] - GPIO1 to GPIO4 interrupt line 2

NOTE: For more information about the slave idle protocol and the wake-up request, see Section 3.1.1.1, *Clock Management*, in Chapter 3, *Power, Reset, and Clock Management*.

Table 20-2 through Table 20-4 summarize the integration of the module in the device.

Table 20-2. GPIO Integration Attributes

Module Instance	Attributes	
	Power Domain	Interconnect
GPIO1	PD_WKUPAON	L4_WKUP
GPIOi (where i = 2 to 4)	PD_COREAON	L4_PER1

Table 20-3. GPIO Clocks and Resets

Clocks				
Module Instance	**Destination Signal Name**	**Source Signal Name**	**Source**	**Description**
GPIO1	GPIO1_ICLK	WKUPAON_GICLK	PRCM	GPIO interface clock
	GPIO1_DBCLK	WKUPAON_SYS_GFCLK	PRCM	GPIO functional clock
GPIO2	GPIO2_ICLK	L4PER_L3_GICLK	PRCM	GPIO interface clock
	GPIO2_DBCLK	GPIO_GFCLK	PRCM	GPIO functional clock
GPIO3	GPIO3_ICLK	L4PER_L3_GICLK	PRCM	GPIO interface clock
	GPIO3_DBCLK	GPIO_GFCLK	PRCM	GPIO functional clock
GPIO4	GPIO4_ICLK	L4PER_L3_GICLK	PRCM	GPIO interface clock
	GPIO4_DBCLK	GPIO_GFCLK	PRCM	GPIO functional clock
Resets				
GPIO1	GPIO1_RST	WKUPAON_RST	PRCM	GPIO reset signal
GPIO2	GPIO2_RST	L4PER_RET_RST	PRCM	GPIO reset signal
GPIO3	GPIO3_RST	L4PER_RET_RST	PRCM	GPIO reset signal
GPIO4	GPIO4_RST	L4PER_RET_RST	PRCM	GPIO reset signal

Table 20-4. GPIO Hardware Requests

Interrupt Requests				
Module Instance	**Source Signal Name**	**Destination IRQ_CROSSBAR Input**	**Default Mapping**	**Description**
GPIO1	GPIO1_IRQ_2	IRQ_CROSSBAR_138	N/A	GPIO1 interrupt request (secont interrupt line). This IRQ source signal is not mapped by default to any device INTC.
	GPIO1_IRQ_1	IRQ_CROSSBAR_24		GPIO1 interrupt request (first interrupt line)
			DSP1_IRQ_55	GPIO1 interrupt request to DSP1 (first interrupt line)
			DSP2_IRQ_55	GPIO1 interrupt request to DSP2 (first interrupt line)
			IPU_IRQ_51	GPIO1 interrupt request to IPU1 (first interrupt line)
GPIO2	GPIO2_IRQ_2	IRQ_CROSSBAR_139	N/A	GPIO2 interrupt request (second interrupt line). This IRQ source signal is not mapped by default to any device INTC
	GPIO2_IRQ_1	IRQ_CROSSBAR_25		GPIO2 interrupt request (first interrupt line)
			DSP1_IRQ_56	GPIO2 interrupt request to DSP1 (first interrupt line)
			DSP2_IRQ_56	GPIO2 interrupt request to DSP2 (first interrupt line)
			IPU_IRQ_52	GPIO2 interrupt request to IPU1 (first interrupt line)
GPIO3	GPIO3_IRQ_2	IRQ_CROSSBAR_140	N/A	GPIO3 interrupt request (second interrupt line). This IRQ source signal is not mapped by default to any device INTC
	GPIO3_IRQ_1	IRQ_CROSSBAR_26		GPIO3 interrupt request (first interrupt line)
			DSP1_IRQ_57	GPIO3 interrupt request to DSP1 (first interrupt line)
			DSP2_IRQ_57	GPIO3 interrupt request to DSP2 (first interrupt line)
GPIO4	GPIO4_IRQ_2	IRQ_CROSSBAR_141	N/A	GPIO4 interrupt request (second interrupt line). This IRQ source signal is not mapped by default to any device INTC
	GPIO4_IRQ_1	IRQ_CROSSBAR_27		GPIO4 interrupt request (first interrupt line)
			DSP1_IRQ_58	GPIO4 interrupt request to DSP1 (first interrupt line)
			DSP2_IRQ_58	GPIO4 interrupt request to DSP2 (first interrupt line)

DMA Requests				
Module Instance	**Source Signal Name**	**Destination DMA_CROSSBAR Input**	**Default Mapping**	**Description**

Table 20-4. GPIO Hardware Requests (continued)

GPIO1	GPIO1_DREQ_EVT	DMA_CROSSBAR_187	N/A	GPIO1 module - event/interrupt1.This DREQ source signal is not mapped by default to any device DMA controller.
GPIO2	GPIO2_DREQ_EVT	DMA_CROSSBAR_188	N/A	GPIO2 module - event/interrupt1.This DREQ source signal is not mapped by default to any device DMA controller.
GPIO3	GPIO3_DREQ_EVT	DMA_CROSSBAR_189	N/A	GPIO3 module - event/interrupt1.This DREQ source signal is not mapped by default to any device DMA controller.
GPIO4	GPIO4_DREQ_EVT	DMA_CROSSBAR_190	N/A	GPIO4 module - event/interrupt1.This DREQ source signal is not mapped by default to any device DMA controller.

NOTE: The "**Default Mapping**" column in Table 20-4 *GPIO Hardware Requests* shows the default mapping of module IRQ and DREQ source signals. These module IRQ and DREQ source signals can also be mapped to other lines of each device Interrupt or DMA controller through the IRQ_CROSSBAR and DMA_CROSSBAR modules, respectively.
For more information about the IRQ_CROSSBAR module, see Section 13.4.6.4, *IRQ_CROSSBAR Module Functional Description*, in Chapter 13, *Control Module*.
For more information about the DMA_CROSSBAR module, see Section 13.4.6.5, *DMA_CROSSBAR Module Functional Description*, in Chapter 13, *Control Module*.
For more information about the device interrupt controllers, see Chapter 12, *Interrupt Controllers*.

NOTE: For the description of the interrupt source, see Section 20.4.6, *Interrupt and Wake-Up Requests.*

20.4 General-Purpose Interface Functional Description

20.4.1 General-Purpose Interface Block Diagram

Figure 20-6 shows the general-purpose interface block diagram.

Figure 20-6. General-Purpose Interface Block Diagram

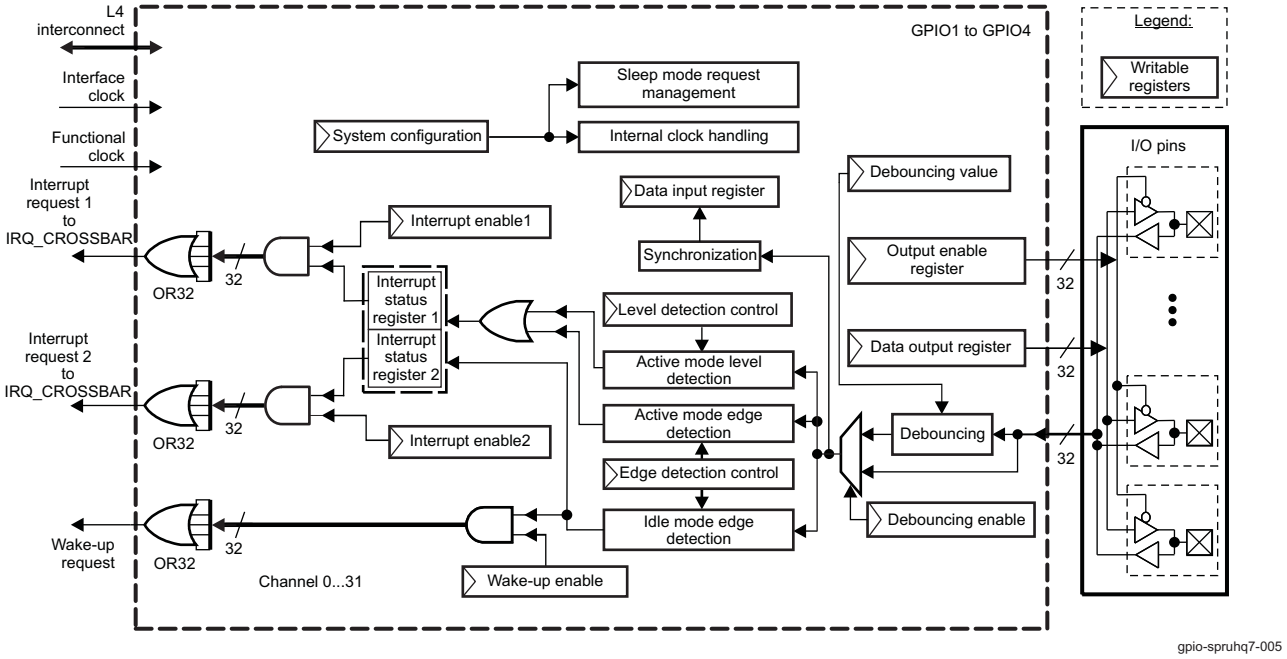
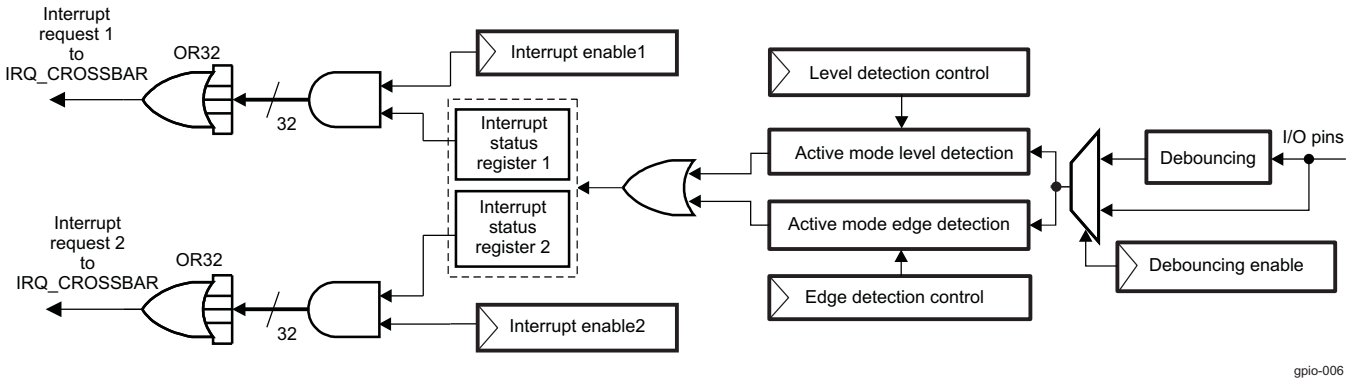


Figure 20-6 shows the details of the GPIO modules in the general-purpose interface block diagram, including their configuration registers and main functional paths:

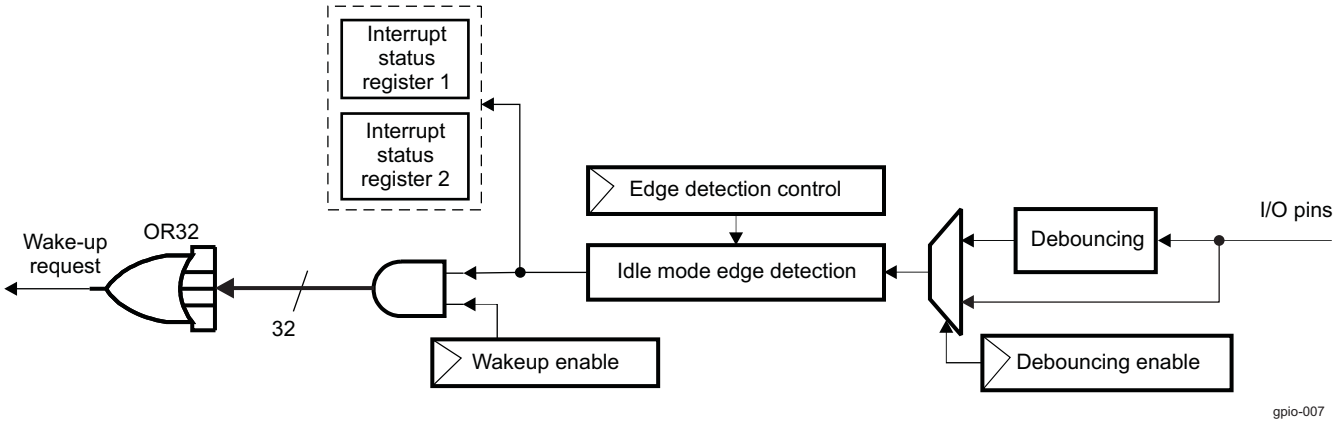
- The synchronous path (for active mode operation) used to generate a synchronous interrupt request on expected event detection on any input GPIO. Synchronous interrupt request lines 1 and 2 are active based on their respective interrupt-enable1 and 2 registers (GPIOi.GPIO_IRQSTATUS_SET_0, GPIOi.GPIO_IRQSTATUS_SET_1, GPIOi.GPIO_IRQSTATUS_CLR_0, and GPIOi.GPIO_IRQSTATUS_CLR_1). See Figure 20-7.

Figure 20-7. Synchronous Path



- The asynchronous path (for idle mode operation) used to generate an asynchronous wake-up request on the expected edge detection on any input GPIO. The asynchronous wake-up request line is active based on the wake-up-enable register. See Figure 20-8.

Figure 20-8. Asynchronous Path



- The blocks handling the internal clock (clock gating) and managing the sleep mode request/acknowledge protocol (enabling the synchronous path in active mode and the asynchronous path in idle mode)

20.4.2 General-Purpose Interface Interrupt and Wake-Up Features

20.4.2.1 Synchronous Path: Interrupt Request Generation

The general-purpose interface has 8 interrupt lines (two interrupt lines per GPIO module instance for biprocessor operation for GPIO1 through GPIO4). The 8 interrupt signals are GPIOi_IRQ_1 (used by the DSP and IPU subsystems) and GPIOi_IRQ_2 (used by the CROSSBAR), where i = 1 to 4.

Synchronous interrupt requests from each channel are processed by two identical interrupt generation submodules used independently by the CROSSBAR subsystem on one side and by the IPU, and DSP subsystems on the other side. Each submodule controls its own synchronous interrupt request line and has its own interrupt-enable registers (GPIOi.GPIO_IRQSTATUS_SET_0, GPIOi.GPIO_IRQSTATUS_SET_1, GPIOi.GPIO_IRQSTATUS_CLR_0, and GPIOi.GPIO_IRQSTATUS_CLR_1) and interrupt status registers (GPIOi.GPIO_IRQSTATUS_RAW_0 and GPIOi.GPIO_IRQSTATUS_RAW_1). The interrupt-enable register selects the channel(s) considered for the interrupt request generation, and the interrupt status register determines which channel(s) activate the interrupt request. Event detection on GPIO channels is reflected in the interrupt status registers independent of the content of the interrupt-enable registers.

In active mode, when the GPIO configuration registers are set to enable the interrupt generation (see Section 20.4.6, *General-Purpose Interface Interrupt and Wake-Up Requests*), a synchronous path samples the transitions and levels on the input GPIO with the internally gated interface clock (see Section 20.4.5.2.4, *Module Power Saving*). When an event matches the programmed settings (see Section 20.4.6, *General-Purpose Interface Interrupt and Wake-Up Requests*), the corresponding bit in the interrupt status register (GPIO_IRQSTATUS_RAW_x [where x = 0 or 1]) is set to 1, and on the following interface clock cycle, interrupt lines 1 and/or 2 are activated (depending on the interrupt-enable registers GPIO_IRQSTATUS_SET_x [where x = 0 or 1]).

Because of the sampling operation, the minimum pulse width on the input GPIO to trigger a synchronous interrupt request is two times the internally gated interface clock period (that is, *N* times the interface clock period; see Section 20.4.5.2.4, *Module Power Saving*). This minimum pulse width must be met before and after any expected level transition detection. Level detection requires the selected level to be stable for at least two times the internally gated interface clock period to trigger a synchronous interrupt.

Because the module is synchronous, latency is minimal between the expected event occurrence and the activation of the interrupt line(s). This latency must not exceed three internally gated interface clock cycles plus two interface clock cycles when the debounce feature is not used.

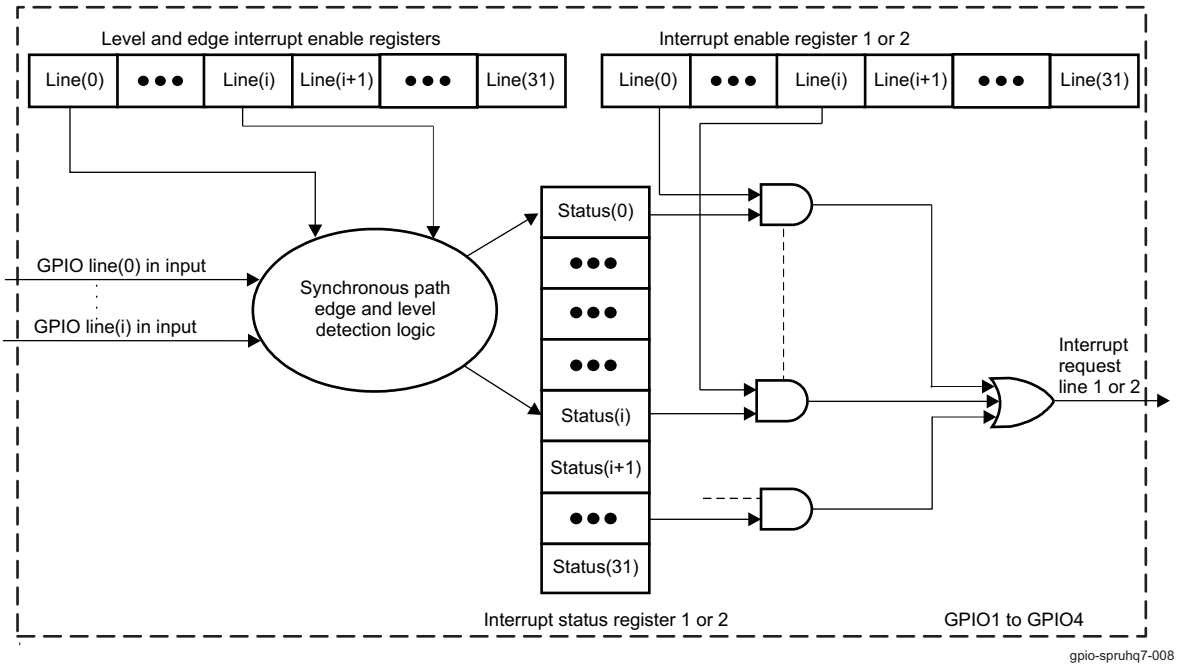
When the debounce feature is active, the latency depends on the value of the debouncing time register (GPIOi.GPIO_DEBOUNCINGTIME) (see Section 20.4.3, *General-Purpose Interface Clock Configuration*) and is less than three internally gated interface clock cycles plus two interface clock cycles plus GPIOi.GPIO_DEBOUNCINGTIME register value debounce clock cycles plus three debounce clock cycles.

Synchronous interrupt request line 1 is default mapped on the IPU, DSP subsystems.

Synchronous interrupt request line 2 is mapped on the CROSSBAR.

Figure 20-9 is an overview of the interrupt request generation.

Figure 20-9. Interrupt Request Generation



20.4.2.2 Asynchronous Path: Wake-Up Request Generation

The general-purpose interface has four wake-up lines (one wake-up line per GPIO module instance) connected to the PRCM module.

Asynchronous wake-up requests from input channels are merged to issue one wake-up signal to the system per GPIO module. The wake-up-enable registers (GPIOi.GPIO_IRQWAKEN_0 and GPIOi.GPIO_IRQWAKEN_1) select the channel(s) considered for the wake-up request generation. The asynchronous wake-up request is reflected into the synchronous interrupt status registers (GPIOi.GPIO_IRQSTATUS_RAW_0 and GPIOi.GPIO_IRQSTATUS_RAW_1).

In idle mode (the interface clock is shut down and the GPIO configuration registers are programmed; see Section 20.4.6, *General-Purpose Interface Interrupt and Wake-Up Requests*), an asynchronous path detects the expected transition(s) on a GPIO input (based on register programming) and activates an asynchronous wake-up request by the sideband signal (GPIOi_SWAKEUP [where i = 1 to 4), if the wake-up-enable register is set.

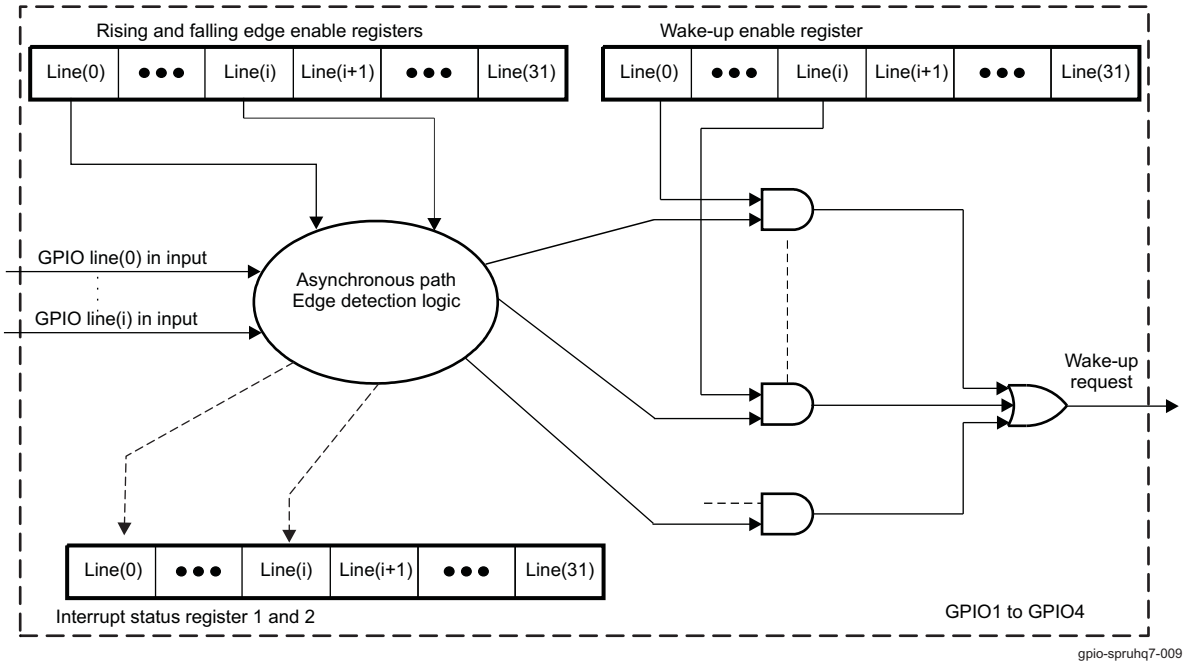
When the system is awakened, the interface clock is restarted and synchronously set to 1 based on the input GPIO pin triggering the wake-up request and the corresponding bit in the interrupt status registers (GPIOi.GPIO_IRQSTATUS_RAW_0 and GPIOi.GPIO_IRQSTATUS_RAW). On the following internal clock cycle, interrupt lines 1 and/or 2 are active (active high) when the corresponding bits are set in the interrupt-enable registers (GPIOi.GPIO_IRQSTATUS_SET_0, GPIOi.GPIO_IRQSTATUS_SET_0, GPIOi.GPIO_IRQSTATUS_CLR_0, and GPIOi.GPIO_IRQSTATUS_CLR_1).

NOTE:

- If the debouncing is not enabled, there is no minimum input pulse width to trigger the wake-up request, because there is no sampling operation.
- If the debouncing is used, the minimum pulse width is set by the debouncing specified time.
- The ENAWAKEUP bit of the GPIO_SYSCONFIG register allows the enabling or disabling of the GPIO wake-up feature globally: if this bit is set to 0, GPIO_IRQWAKEN_x has no effect

Figure 20-10 is an overview of the wake-up request generation.

Figure 20-10. Wake-Up Request Generation



20.4.2.3 Wake-Up Event Conditions During Transition To/From IDLE State

In phase A, only the synchronous path is enabled. A synchronous interrupt request (see Section 20.4.2.1, *Synchronous Path: Interrupt Request Generation*) activates the interrupt line(s) and prevents the GPIO from transitioning into IDLE state until the interrupt is cleared.

In phase B, the asynchronous path and synchronous path are enabled during the first five functional clock cycles of SLEEPTRANS state. During this period a synchronous interrupt request (see Section 20.4.2.1, *Synchronous Path: Interrupt Request Generation*) prevents the GPIO from transitioning into IDLE state. A shorter pulse puts the module into IDLE state but triggers a wakeup once in IDLE.

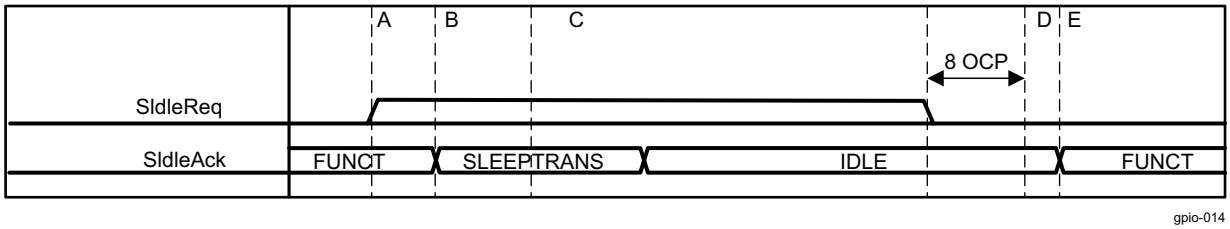
In phase C, only the asynchronous path is enabled. A wake-up request (see Section 20.4.2.2, *Asynchronous Path: Wake-Up Request Generation*) triggers a wake-up request from the GPIO and when the module is awakened an interrupt is generated. If debouncing is not enabled, there is no minimum input pulse width to trigger the wake-up request.

In phase D, eight open-core protocol (OCP) clock cycles occur until the module is in FUNCT state, the synchronous path is enabled, and an event that fulfills the pulse width requirements (see Section 20.4.2.1, *Synchronous Path: Interrupt Request Generation*) activates the interrupt line(s).

In phase E, only the synchronous path is enabled. A synchronous interrupt request (see Section 20.4.2.1, *Synchronous Path: Interrupt Request Generation*) activates the interrupt line(s).

Figure 20-11 shows the wake-up event conditions.

Figure 20-11. Wake-Up Event Conditions



20.4.2.4 Interrupt (or Wake-Up) Line Release

When the host processor (the DSP subsystem in the device) receives an interrupt request issued by the GPIO module, it reads the corresponding interrupt status register (GPIOi.GPIO_IRQSTATUS_0 or GPIOi.GPIO_IRQSTATUS_1) to determine which GPIO input triggered the interrupt (or the wake-up request).

After servicing the interrupt (or acknowledging the wake-up request), the software resets the status bit and releases the interrupt line by setting the corresponding bit of the interrupt status register to 1. If there is still a pending interrupt request to serve (all bits in the interrupt status register that are not masked by the interrupt-enable register are not cleared), the interrupt line is reasserted.

NOTE: The status bit must be reset to re-enter idle mode.

20.4.3 General-Purpose Interface Clock Configuration

20.4.3.1 Clocking

Each GPIO module uses two clocks:

- Debounce clock: The 32-kHz debounce clock, GPIOi_DBCLK (where i = 1 to 4 with one debounce clock per module), comes from the PRCM module and is used to debounce the submodule logic (without the corresponding configuration registers). This module can sample the input line and filter the input level using a programmed delay.

  The debouncing value register (GPIOi.GPIO_DEBOUNCINGTIME) is used to set the debouncing time for all input lines in the GPIO module. Because the value is global for all the ports of one GPIO module, up to eight different debouncing values are possible. The debounce cell runs with the debounce clock (32 kHz). This register represents the number of clock cycle(s) to be used.

  The following formula describes the required input stable time to be propagated to the debounced output:

  Required input line stable = (the value of the GPIOi.GPIO_DEBOUNCINGTIME[7:0] DEBOUNCETIME bit field + 1) × 31,

  where the value of the GPIOi.GPIO_DEBOUNCINGTIME[7:0] DEBOUNCETIME bit field is from 0 to 255.

  For more information, see Section 3.6.4.5, *CD_L4_PER1 Clock Domain*, and Section 3.6.4.1, *CD_WKUPAON Clock Domain*, in Chapter 3, *Power, Reset, and Clock Management*.
- Interface clock: The interface clock, GPIOi_ICLK (where i = 1 to 4), comes from the PRCM module and is used throughout the GPIO module (except within the debounce cell logic). GPIOi_ICLK clocks the data exchanges between the L4 interconnect and the internal logic. The clock-gating features allow module power consumption to be adapted to the activity.

  For more information, see Section 3.6.4.5, *CD_L4_PER1_Clock Domain*, and Section 3.6.4.1, *CD_WKUPAON Clock Domain*, in Chapter 3, *Power, Reset, and Clock Management*.

Table 20-3 describes the clocks in the GPIO modules.

Table 20-5 summarizes the functional clock configuration.

Table 20-5. Functional Clock Configuration

Interface Clock	GPIO_CTRL[2:1]GATINGRATIO	Functional Clock
GPIOi_ICLK (where i = 1 to 4)	00	GPIOi_ICLK /1
GPIOi_ICLK (where i = 1 to 4)	01	GPIOi_ICLK /2
GPIOi_ICLK (where i = 1 to 4)	10	GPIOi_ICLK /4
GPIOi_ICLK (where i = 1 to 4)	11	GPIOi_ICLK /8

20.4.4 General-Purpose Interface Hardware and Software Reset

The GPIO can be reset by using the domain reset (hardware reset) or by setting a dedicated configuration bit (software reset) in each GPIO module.

- Hardware reset: The GPIO2 to GPIO4 modules are attached to the L4PER_RET_RST reset domain. GPIO1 is attached to the WKUPAON_RST reset domain.

  The hardware reset has a global reset action on the GPIO modules of the general-purpose interface. All configuration registers and internal logic are reset when it is active (low level). In each GPIO module, the GPIOi.GPIO_SYSSTATUS[0] RESETDONE bit monitors the internal reset status; it is set when the reset completes. For more information, see Section 3.5.5, *Reset Domains*, in Chapter 3, *Power, Reset, and Clock Management*.
- Software reset: Each GPIO module has its own software reset using the GPIOi.GPIO_SYSCONFIG[1] SOFTRESET bit. The software reset has the same effect as the hardware reset signal, but this reset can be applied on one or more modules.

  Setting the GPIOi.GPIO_SYSCONFIG[1] SOFTRESET bit to 1 resets the module. A bit value of 1 remains until the reset completes. When the software reset completes, the

GPIOi.GPIO_SYSCONFIG[1] SOFTRESET bit is automatically reset to 0 and has the same effect as a hardware reset. The GPIOi.GPIO_SYSSTATUS[0] RESETDONE bit is cleared during a software reset. RESETDONE is set to 1 when the software reset completes.

20.4.5 General-Purpose Interface Power Management

20.4.5.1 Power Domain

GPIO1 is attached to the PD_WKUPAON power domain (see Section 3.1.1.2, *Power Management*, in Chapter 3, *Power, Reset, and Clock Management*). This domain is composed of the logic permanently supplied to manage domain power state transitions and detect wake-up events. The WKUPAON power domain is continuously active. The GPIO2 to GPIO4 modules are attached to the PD_L4PER power domain (see Section 3.1.1.2, *Power Management*, in Chapter 3, *Power, Reset, and Clock Management*). The PD_L4PER power domain is not active continuously.

20.4.5.2 Power Management

20.4.5.2.1 Idle Scheme

To reduce dynamic consumption, an efficient idle scheme is based on the following:

- Efficient local autoclock gating for each module
- The implementation of control sideband signals between the PRCM module and each module

This enhanced idle control allows clocks to be activated and deactivated safely without requiring complex software management.

The idle mode request, idle acknowledge, and wake-up request (GPIOi_WAKEUP) are sideband signals between the PRCM module and the general-purpose interface (see Section 20.4.6.2, *Wake-Up Requests Generation*).

20.4.5.2.2 Operating Modes

Three operating modes are defined for the module:

- Active mode: The module runs synchronously on the interface clock; interrupts can be generated based on the configuration and external signals.
- Idle mode: Power-saving mode with the module in a waiting state. The interface clock can be stopped, an interrupt cannot be generated, and a wake-up signal can be generated based on the configuration and external signals.

  If the debounce clock provided by the PRCM module is active, the debounce cell can sample and filter the input to generate a wake-up event. If the debounce clock is inactive, the debounce cell gates all input signals and thus cannot be used.
- Disabled mode: The module is not used. The internal clock paths are gated, and an interrupt or wake-up request cannot be generated.

Idle mode is configured within the module and activated on request by the host processor through sideband signals (see Section 20.4.5.2.3, *System Power Management and Wakeup*).

The disabled mode is set by software through a dedicated configuration bit, GPIOi.GPIO_CTRL[0] DISABLEMODULE (0: The module is enabled and clocks are not gated; 1: The module is disabled and clocks are gated). It unconditionally gates the internal clock paths that are not used for the system interface. When setting the GPIO_CTRL[0] DISABLEMODULE bit (enabling or disabling the GPIO module), it is important to switch the debouncing clock on or off in the following order:

- The GPIO optional debouncing clock must be enabled before the GPIO module is set (GPIO_CTRL[0]DISABLEMODULE = 0x0).
- The GPIO optional debouncing clock must be disabled after the GPIO module is disabled (GPIO_CTRL[0]DISABLEMODULE = 0x1).

20.4.5.2.3 System Power Management and Wakeup

The PRCM module can require the GPIO modules to be idled for power-saving purposes.

The general-purpose interface has eight identical idle mode request/acknowledge (handshake) mechanisms with the PRCM module (see Section 20.4.6.2, *Wake-Up Requests Generation*): one per GPIO module. The general-purpose interface allows the GPIO modules to enter idle mode based on the GPIOi.GPIO_SYSCONFIG[4:3] IDLEMODE bit field.

Idle acknowledge depends on the configuration and activity of each GPIO module:

- Smart-idle mode

  When the GPIO module is configured in smart-idle mode, it checks for more activity (capture of the input GPIO pins in the GPIOi.GPIO_DATAIN register is complete with no pending interrupt; all interrupt status bits are cleared), and there is no write access to the GPIO.GPIO_DEBOUNCINGTIME register, which is waiting to be synchronized.

  Idle acknowledge is then asserted and the module enters into idle mode. It waits for active system clock gating by the PRCM module (when all peripherals supplied by the same L4 interface clock domain are also ready for idle).

  In idle mode (that is, when the PRCM module gates the interface clock), no interrupt occurs.

- Smart-idle wake-up mode

  If the GPIOi.GPIO_SYSCONFIG[4:3] IDLEMODE bit field selects smart-idle or smart-idle wake-up mode, the GPIO module evaluates its internal capability to have the interface clock switched off. Once all internal activity ceases (the DATA INPUT REGISTER completed to capture the input GPIO pins, there is no pending interrupt, all interrupt status bits are cleared, and there is no write access to the GPIO_DEBOUNCINGTIME register pending to be synchronized), the idle acknowledge is asserted and the GPIO enters into Idle mode, ready to issue a wake-up request when the expected transition occurs on an enabled GPIO input pin. This wake-up request is effectively sent only if the GPIOi.GPIO_SYSCONFIG[2] ENAWAKEUP bit of the system configuration register enables the GPIO wake-up capability (see). When the system is awake, the IDLE request goes inactive, the idle acknowledge and wake-up request (if the GPIO triggered the wake-up in the system) signals are immediately deasserted, and the asynchronous wake-up request (if it exists) is reflected into the synchronous interrupt status registers.

- Force-idle mode

  When the GPIO module is configured in force-idle mode (the GPIOi.GPIO_SYSCONFIG[4:3] IDLEMODE bit field = 0x0) and receives an IDLE request from the PRCM module, the GPIO module waits unconditionally for active system clock gating by the PRCM module. (This occurs only when all peripherals supplied by the same L4 interface clock domain are also ready for idle.)

  When in idle mode (that is, when the PRCM module gates the interface clock), the module (in inactive mode) has no activity, the interface clock paths are gated, an interrupt cannot be generated, and the wake-up feature is totally inhibited.

- No-idle mode

  When the GPIO module is configured in no-idle mode (the GPIOi.GPIO_SYSCONFIG[4:3] IDLEMODE bit field = 0x1) and receives an IDLE request from the PRCM module, the GPIO module does not go into idle mode and the idle acknowledge is never sent.

NOTE: For more information about the idle modes, see Section 3.1.1.2, *Power Management*, in Chapter 3, *Power, Reset, and Clock Management*.

20.4.5.2.4 Module Power Saving

The GPIO module has local power management by internal clock-gating features:

- Internal interface clock gating: The clock for the system interface logic can be gated when the module is not accessed, if the GPIOi.GPIO_SYSCONFIG[0] AUTOIDLE bit is set. Otherwise, this logic is free-running on the interface clock.
- Clock gating for the input data sample logic: Clock for the input data sample logic can be gated when the data in the GPIOi.GPIO_DATAIN register is not accessed.

- Clock gating for the event detection logic: Each GPIO module implements four clock groups used for the logic in the synchronous event detection. Each group of eight input GPIO pins has a separate enable signal depending on the edge/level detection register setting (because the input is 32 bits, four groups of eight inputs are defined for each GPIO module). If a group requires no detection, the corresponding clock is gated off.

  All channels are also gated using a one-out-of-N scheme. N is the GPIOi.GPIO_CTRL[2:1] GATINGRATIO bit field and can take the values 1, 2, 4, and 8. The interface clock is enabled for this logic one cycle every N cycles. When N is 1, there is no gating and this logic is free-running on the interface clock. When N is 2, 4, or 8, this logic runs at the equivalent frequency of interface clock frequency divided by N.
- Inactive mode: In inactive mode, all internal clock paths are gated.
- Disabled mode: All internal clock paths not used for the L4 interconnect are gated. The GPIOi.GPIO_CTRL[0] DISABLEMODULE bit controls a clock-gating feature at the module level. Setting this bit to 1 forces clock gating for all internal clock paths. Module internal activity is suspended. The L4 interconnect is not affected by this bit.

The interface clock gating is controlled with the GPIOi.GPIO_SYSCONFIG[0] AUTOIDLE bit, which is used to save power when the module is not used because of the multiplexing configuration selected at the chip level. This bit has precedence over all other internal configuration bits.

Table 20-6 describes the power-management features available for the general-purpose interface module.

NOTE: For information about source clock gating and sleep/wake-up transitions, see Section 3.1.1.1.2, *Module-Level Clock Management*, in Chapter 3, *Power, Reset, and Clock Management*.

For descriptions of the EnaWakeUp, IdleMode, ClockActivity, and StandbyMode features, see Section 3.1.1.1.2, *Module-Level Clock Management*, in Chapter 3, *Power, Reset, and Clock Management*.

Table 20-6. Local Power-Management Features

Feature	Register Bits/Bit Fields	Description
Clock autogating	GPIOi.GPIO_SYSCONFIG[0] AUTOIDLE	It sets the clock-gating strategy for the OCP interface block.
Slave-idle modes	GPIOi.GPIO_SYSCONFIG[4:3] IDLEMODE	Force-idle, no-idle, and smart-idle wake-up-capable modes are available.
Clock activity	GPIOi.GPIO_CTRL[0] DISABLEMODULE	Enable and disable the module.
Debouncing enable	GPIOi.GPIO_DEBOUNCENABLE[31:0] DEBOUNCEENABLE	Debouncing mode is available.
Global wake-up enable	GPIOi.GPIO_SYSCONFIG[2] ENAWAKEUP	This bit enables the wake-up feature at the module level.
Wake-up sources enable	GPIOi.GPIO_IRQWAKEN_0[31:0] INTLINE GPIOi.GPIO_IRQWAKEN_1[31:0] INTLINE	This register enables or disables a specific IRQ request source to generate a wake-up signal.

Table 20-7 describes the clock activity settings.

Table 20-7. Clock Activity Settings

GPIOi.GPIO_SYSCONFIG[4:3] IDLEMODE	Selected Mode	Description	Wake-Up Events
00	Force-idle	The GPIO module goes into inactive mode independently of the internal module state, and the IDLE acknowledge is never sent.	No
01	No-idle	The GPIO module does not go into Idle mode and the IDLE acknowledge is never sent.	No

Table 20-7. Clock Activity Settings (continued)

GPIOi.GPIO_SYSCONFIG[4:3] IDLEMODE	Selected Mode	Description	Wake-Up Events
10	Smart-idle	The GPIO module evaluates its internal capability to have the interface clock switched off. If there is no more internal activity, the IDLE acknowledge is asserted and the GPIO enters idle mode.	No
11	Smart-idle wake-up	The GPIO module evaluates its internal capability to have the interface clock switched off. If there is no more internal activity, the IDLE acknowledge is asserted and the GPIO enters idle mode.	Yes

20.4.6 General-Purpose Interface Interrupt and Wake-Up Requests

20.4.6.1 Interrupt Requests Generation

All interrupt sources (the 32 GPIO input channels) are merged to issue two synchronous interrupt requests in each GPIO module. Thus, the general-purpose interface has 8 interrupt lines (two interrupt lines per GPIO module instance).

- Synchronous interrupt request line 1 is default mapped on the DSP and IPU INTC subsystem.
- Synchronous interrupt request line 2 is mapped on the CROSSBAR.

Table 20-8 lists the event flags, and their mask, that can cause module interrupts.

Table 20-8. Events

Event Flag	Event Mask	Synchronous	Sensitivity	Description
GPIOi.GPIO_IRQSTATUS_0 [31:0] INTLINE	GPIOi.GPIO_IRQSTATUS_SET_0[31:0] INTLINE	Yes	Edge/ level	Corresponding to the first line of interrupt
GPIOi.GPIO_IRQSTATUS_1 [31:0] INTLINE	GPIOi.GPIO_IRQSTATUS_SET_1[31:0] INTLINE	Yes	Edge/ level	Corresponding to the second line of interrupt
GPIOi.GPIO_IRQSTATUS_0 [31:0] INTLINE	GPIOi.GPIO_IRQSTATUS_SET_0[31:0] INTLINE	Yes	Edge/ level	Corresponding to the first line of interrupt
GPIOi.GPIO_IRQSTATUS_1 [31:0] INTLINE	GPIOi.GPIO_IRQSTATUS_SET_1[31:0] INTLINE	Yes	Edge/ level	Corresponding to the second line of interrupt
GPIOi.GPIO_IRQSTATUS_0 [31:0] INTLINE	GPIOi.GPIO_IRQSTATUS_CLR_0[31:0] INTLINE	Yes	Edge/ level	Corresponding to the first line of interrupt
GPIOi.GPIO_IRQSTATUS_1 [31:0] INTLINE	GPIOi.GPIO_IRQSTATUS_CLR_1[31:0] INTLINE	Yes	Edge/ level	Corresponding to the second line of interrupt
GPIOi.GPIO_IRQSTATUS_0 [31:0] INTLINE	GPIOi.GPIO_IRQWAKEN_0 [31:0] INTLINE	No	Edge/ level	Corresponding to the first line of interrupt
GPIOi.GPIO_IRQSTATUS_1 [31:0] INTLINE	GPIOi.GPIO_IRQWAKEN_1 [31:0] INTLINE	No	Edge/ level	Corresponding to the second line of interrupt

NOTE: For more information about interrupt mapping, see Table 20-4, *GPIO Hardware Request*.

Synchronous interrupt request line 1 and line 2 are active depending on their respective interrupt-enable1 and interrupt-enable 2 registers (GPIOi.GPIO_IRQSTATUS_SET_0, GPIOi.GPIO_IRQSTATUS_SET_1, GPIOi.GPIO_IRQSTATUS_CLR_0, and GPIOi.GPIO_IRQSTATUS_CLR_1).

- interrupt-enable registers (GPIOi.GPIO_IRQSTATUS_SET_0 and GPIOi.GPIO_IRQSTATUS_SET_1)

  The interrupt enable1 (or interrupt enable2) register allows masking of the expected transition on input GPIO to prevent the generation of an interrupt request on line 1 (or line 2). The interrupt-enable registers are programmed synchronously with the interface clock.

  These registers can be accessed with direct read/write operations or by using the alternate set-and-

clear protocol feature for register update. This feature allows setting or clearing explicit bits of these registers with a single write access (see Section 20.4.9, *General-Purpose Interface Set-and-Clear Protocol*).

- Interrupt status registers (GPIOi.GPIO_IRQSTATUS_0 and GPIOi.GPIO_IRQSTATUS_1)

  The interrupt status 1 (or interrupt status 2) register determines which of the input GPIO pins triggered the interrupt line1 (or interrupt line 2) request (or the wake-up line).

  When a bit in this register is set to 1, it indicates that the corresponding GPIO pin is requesting the interrupt (or the wakeup). To reset a bit in this register, set the appropriate bit to 1. However, an interrupt cannot be generated by writing 1 to the interrupt status 1 (or interrupt status 2) register.

  If 0 is written to a bit in this register, the value in the corresponding bit in the interrupt status 1 and remains unchanged. The interrupt status 1 (or interrupt status 2) register is synchronous with the interface clock. In idle mode, the event is detected through an asynchronous path, and interrupt status 2 registers are set when the GPIO module is awaken.

CAUTION

After servicing the interrupt, the status bit in the interrupt status register (GPIOi.GPIO_IRQSTATUS_0 or GPIOi.GPIO_IRQSTATUS_1) must be reset and the interrupt line released (by setting the corresponding bit of the interrupt status register to 1).

Before enabling an interrupt for the GPIO channel in the interrupt-enable register (GPIOi.GPIO_IRQSTATUS_SET_0 or GPIOi.GPIO_IRQSTATUS_SET_1) the corresponding status bit in the interrupt status register (GPIOi.GPIO_IRQSTATUS_0 or GPIOi.GPIO_IRQSTATUS_1) must be reset to prevent the occurrence of unexpected interrupts when enabling an interrupt for the GPIO channel.

20.4.6.2 Wake-Up Requests Generation

The GPIO1 module of the general-purpose interface is attached to the WKUPAON power domain (see Section 3.1.1.2, *Power Management*, in Chapter 3, *Power, Reset, and Clock Management*, and can wake up the system.

NOTE: The GPIO2 to GPIO4 modules belong to the PD_L4PER power domain and thus their wake-up capabilities are operational only when the PD_L4PER power domain is active.

All wake-up sources (the 32 input GPIO channels) are merged together to issue a single asynchronous wake-up request in each GPIO module following the expected transition(s) (based on register programming). Each GPIO module generates a wake-up signal to the PRCM module.

NOTE: Only gpio1_[3:0] can be used to generate a direct wake-up event.

The asynchronous wake-up request line is active based on the GPIOi.GPIO_IRQWAKEN_0 and GPIO_IRQWAKEN_1 wake-up-enable registers (where i = 1 to 4).

The wake-up-enable register allows masking of the expected transition on input GPIO to prevent the generation of a wake-up request. The wake-up-enable register is programmed synchronously with the interface clock before any idle mode request coming from the host processor.

This register can be accessed with direct read/write operations.

NOTE: There must be a correlation between the wake-up enable and interrupt-enable registers. If a GPIO pin has a wake-up configured on it, it must also have the corresponding interrupt enabled (on one of the two interrupt lines). Otherwise, it is possible to have a wake-up event, but after exiting the IDLE state, no interrupt is generated; thus, the corresponding bit from the interrupt status register is not cleared, and the module does not acknowledge a future IDLE request.

Table 20-9 lists the mapping of the wake-up signals.

Table 20-9. Wake-Up Signals

Name	Mapping	Comments
GPIOi_WAKE	GPIOi_SWAKEUP	Where i = 1 to 4. The destination is the PRCM module.

20.4.7 General-Purpose Interface Channels Description

Table 20-10 describes the GPIO channels.

Table 20-10. GPIO Channels Description

Channel Number	Type[1]	Mapping	Wake-Up Feature	Comments
GPIO1				
[31:0]	I/O	gpio1_[31:0]	Yes	GPIO. Wake-up path.
GPIO2				
[31:0]	I/O	gpio2_[31:0]	Yes	GPIO
GPIO3				
[31:0]	I/O	gpio3_[31:0]	Yes	GPIO
GPIO4				
[22:0]	I/O	gpio4_[22:0]	Yes	GPIO
[24:23]	-	Reserved	Reserved	Not pinned-out
[29:25]	I/O	gpio4_[29:25]	Yes	GPIO
[31:30]	-	Reserved	Reserved	Not pinned-out

(1) I = Input; O = Output; I/O = bidirectional

NOTE: For more information about pin configuration, see Section 13.4.6.1, *Pad Configuration Registers*, in Chapter 13, *Control Module*.

NOTE: GPIO pins can directly snoop device pads even if those are configured in the other modes. For GPIO output user needs to set pad mux.

20.4.8 General-Purpose Interface Data Input/Output Capabilities

The output-enable register (GPIOi.GPIO_OE) controls the I/O capability of each pin. At reset, all the GPIO-related pins are configured as inputs, and their output capabilities are disabled. This register is not used within the module. Its only function is to carry the pad configuration.

When configured as an output (the desired bit reset in the GPIOi.GPIO_OE register), the value of the corresponding bit in the GPIOi.GPIO_DATAOUT register is driven on the corresponding GPIO pin. Data is written to the data-output register synchronously with the interface clock. This register can be accessed with read/write operations or by using the alternate set-and-clear protocol register update feature. This feature gives the possibility to set or clear specific bits of this register with a single write access to the set output data register (GPIOi.GPIO_SETDATAOUT) or to the clear output data register

(GPIOi.GPIO_CLEARDATAOUT) address (see Section 20.4.9, *General-Purpose Interface Set-and-Clear Protocol*). If the application uses a pin as an output and does not want interrupt/wake-up generation from this pin, the application must properly configure the wake-up enable (GPIOi.GPIO_ WAKEUPENABLE) and the interrupt-enable (GPIOi.GPIO_IRQSTATUS_SET_0 and GPIOi.GPIO_IRQSTATUS_SET_1) registers.

When configured as an input (the desired bit is set to 1 in the GPIOi.GPIO_OE register), the state of the input can be read from the corresponding bit in the GPIOi.GPIO_DATAIN register. The input data is sampled synchronously with the interface clock and then captured in the data input register synchronously with the interface clock. When the GPIO pin levels change, they are captured into this register after two interface clock cycles (the required cycles to synchronize and write data). If the application uses a pin as an input, the application must properly configure the wake-up enable (GPIOi.GPIO_IRQWAKEN_0 and GPIOi.GPIO_IRQWAKEN_1) and the interrupt-enable (GPIOi.GPIO_IRQSTATUS_SET_0 and GPIOi.GPIO_IRQSTATUS_SET_0) registers to the interrupt and wake-up feature as needed. For using the alternate set-and-clear protocol, see Section 20.4.9, *General-Purpose Interface Set-and-Clear Protocol.*

20.4.9 General-Purpose Interface Set-and-Clear Protocol

20.4.9.1 Description

The GPIO module implements the set-and-clear protocol register update for the following registers:

- GPIOi.GPIO_DATAOUT
- GPIOi.GPIO_IRQSTATUS_CLR_0
- GPIOi.GPIO_IRQSTATUS_CLR_1
- GPIOi.GPIO_IRQSTATUS_SET_0
- GPIOi.GPIO_IRQSTATUS_SET_1

This protocol is an alternative to the atomic test and set operations and consists of writing operations at dedicated addresses (one address for setting bit[s] and one address for clearing bit[s]). The data to write is 1 at bit position(s) to clear (or to set) and 0 at unaffected bit(s). Registers can be accessed in two ways:

- Standard: Full register read and write operations at the primary register address
- Set and clear: Separate addresses are provided to set (and clear) bits in registers. Writing 1 at these addresses sets (or clears) the corresponding bit into the equivalent register; writing 0 has no effect.

Therefore, for these registers, three addresses are defined for one unique physical register. Reading these addresses has the same effect and returns the register value.

20.4.9.2 Clear Instruction

20.4.9.2.1 Clear Register Addresses

- Clear interrupt-enable registers (GPIOi.GPIO_IRQSTATUS_CLR_0 and GPIOi.GPIO_IRQSTATUS_CLR_1).

  A write operation in the GPIO_IRQSTATUS_CLR_0 (or GPIO_IRQSTATUS_CLR_1) register clears the corresponding bit in the same register when the written bit is 1; a written bit at 0 has no effect.

  A read of the clear interrupt enable0 (or enable1) register returns the value of the GPIO_IRQSTATUS_CLR_0 (or GPIO_IRQSTATUS_CLR_1) register.
- Clear data-output register (GPIOi.GPIO_CLEARDATAOUT).

  A write operation in the clear data-output register clears the corresponding bit in the data-output register when the written bit is 1; a written bit at 0 has no effect.

  A read of the clear data-output register returns the value of the data-output register.

20.4.9.2.2 Clear Instruction Example

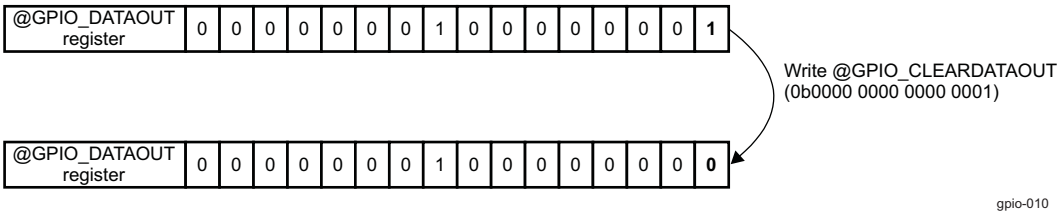
Assume the data-output register (or one of the interrupt or wake-up-enable registers) contains the binary value 0b0000 0001 0000 0001 and bit 0 is to be cleared.

With the clear instruction feature, write 0b0000 0000 0000 0001 at the address of the clear data-output register (or at the address of the clear interrupt or wake-up-enable register). After this write operation, a reading of the data-output register (or the interrupt or wake-up-enable register) returns 0b0000 0001 0000 0000; bit 0 is cleared.

NOTE: Although the general-purpose interface registers are 32 bits wide, only the 16 least-significant bits (LSBs) are represented in this example.

Figure 20-12 is an example of a clear instruction.

Figure 20-12. GPIO_CLEARDATAOUT Register Example



20.4.9.3 Set Instruction

20.4.9.3.1 Set Register Addresses

- Set interrupt-enable registers (GPIOi.GPIO_IRQSTATUS_SET_0 and GPIOi.GPIO_IRQSTATUS_SET_1).

  A write operation in the GPIOi.GPIO_IRQSTATUS_SET_0 (or GPIOi.GPIO_IRQSTATUS_SET_1) register sets the corresponding bit in the same register when the written bit is 1; a written bit at 0 has no effect.

  A read of the set interrupt-enable 0 (or enable1) register returns the value of the interrupt GPIOi.GPIO_IRQSTATUS_SET_0 (or GPIOi.GPIO_IRQSTATUS_SET_1) register.
- Set data-output register (GPIOi.GPIO_SETDATAOUT).

  A write operation in the set data-output register sets the corresponding bit in the data-output register when the written bit is 1; a written bit at 0 has no effect.

  A read of the set data-output register returns the value of the data-output register.

20.4.9.3.2 Set Instruction Example

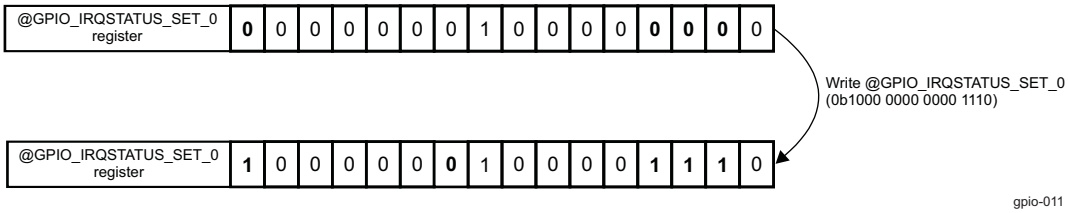
Assume the interrupt enable1 (or enable2) register (or the data-output register) contains the binary value 0b0000 0001 0000 0000 and bits 15, 3, 2, and 1 are to be set.

With the set instruction feature, the user has only to write 0b1000 0000 0000 1110 at the address of the GPIO_SETDATAOUT register. After this write operation, a reading of the GPIO_SETDATAOUT register returns 0b1000 0001 0000 1110: bits 15, 3, 2, and 1 have been set.

NOTE: Although the general-purpose interface registers are 32 bits wide, only the 16 LSBs are represented in this example.

Figure 20-13 is an example of a set instruction.

Figure 20-13. Write in GPIO_IRQSTATUS_SET_0 Register Example



The set wake-up-enable register offers the same feature with the wake-up-enable register.

20.5 General-Purpose Interface Programming Guide

20.5.1 General-Purpose Interface Low-Level Programming Models

20.5.1.1 Global Initialization

20.5.1.1.1 Surrounding Modules Global Initialization

This section identifies the requirements for initializing the surrounding modules when the general-purpose interface module is to be used for the first time after a device reset. This initialization of surrounding modules is based on the environment and integration of the general-purpose interface. For more information, see Section 20.2, *General-Purpose Interface Environment*, and Section 20.3, *General-Purpose Interface Integration*.

Table 20-11. Global Initialization of Surrounding Modules

Surrounding Modules	Comments
PRCM	Module interface and functional clocks must be enabled. For more information about the module configuration, see Section 3.1.1.1, *Clock Management*, in Chapter 3, *Power, Reset, and Clock Management*.
Control module	Module specific pad muxing must be set in the control module. For more information about the module configuration, see Section 13.4.6.1, *Pad Configuration Registers*, in Chapter 13, *Control Module*.
Device INTCs	Device INTCs must be configured to enable the interrupt request generation. For more information see Chapter 12, *Interrupt Controllers*.
IRQ_CROSSBAR	IRQ_CROSSBAR configuration must be done to allow module IRQs to be mapped to certain device INTC line. For more information see Section 13.4.6.4, *IRQ_CROSSBAR Module Functional Description*, in Chapter 13, *Control Module*.

20.5.1.1.2 General-Purpose Interface Module Global Initialization

This procedure initializes the general-purpose Interface module after a power-on reset (POR) or software reset.

Table 20-12. General-Purpose Interface Global Initialization

Step	Register/Bit Field/Programming Model	Value
Execute software reset.	GPIO_SYSCONFIG[1] SOFTRESET	0x1
Wait until reset completed?	GPIO_SYSSTATUS[0] RESETDONE	= 0x1
Configure idle mode.	GPIO_SYSCONFIG[4:3] IDLEMODE	0x-
Configure interface clock gating.	GPIO_SYSCONFIG[0] AUTOIDLE	0x-
Set clock-gating ratio.	GPIO_CTRL [2:1] GATINGRATIO	0x-
Configure GPIO channels as input or output.	GPIO_OE[31:0] OUTPUTEN	0x-
Set debounce time value.	GPIO_DEBOUNCINGTIME[7:0] DEBOUNCINGTIME	0x-
Enable/disable debouncing for desired input line. (For example, when used with a push-button)	GPIO_DEBOUNCENABLE[31:0] DEBOUNCENABLE	0x-
Interrupt and wake-up requests configuration		
(Optional) Enable/disable wake-up for desired input lines.	GPIO_IRQWAKEN_0[31:0] INTLINE and/or GPIO_IRQWAKEN_1[31:0] INTLINE	0x-
(Optional) Enable wake-up generation.	GPIO_SYSCONFIG[2] ENAWAKEUP	0x1

Table 20-12. General-Purpose Interface Global Initialization (continued)

Step	Register/Bit Field/Programming Model	Value
Configure detection events. NOTE: Simultaneous enabling of high-level and low-level detection for one given pin creates a constant-interrupt generator.	GPIO_LEVELDETECT0[31:0] LEVELDETECT0 and/or GPIO_LEVELDETECT1[31:0] LEVELDETECT1 and/or GPIO_RISINGDETECT[31:0] RISINGDETECT and/or GPIO_FALLINGDETECT[31:0] FALLINGDETECT	0x-
Clear interrupt status	GPIO_IRQSTATUS_0[31:0] INTLINE and/or GPIO_IRQSTATUS_1[31:0] INTLINE	0xFFFF FFFF
Enable interrupts for desired input lines. If wakeup is enabled, it is mandatory to enable the corresponding interrupt.	GPIO_IRQSTATUS_SET_0[31:0] INTLINE and/or GPIO_IRQSTATUS_SET_1[31:0] INTLINE	0x-

NOTE: Detection of events requires a functional clock running for every group of 8 bits. If detection of events is enabled only within one octet (for example, 0x0012 0000), power saving can be achieved. Else (for example, 0x0102 0000), using two octets requires one more clock to be run by the module.

20.5.1.2 General-Purpose Interface Operational Modes Configuration

20.5.1.2.1 General-Purpose Interface Read Input Register

Table 20-13. General-Purpose Interface Read Input Register

Step	Register/Bit Field/Programming Model	Value
Read interrupt status	GPIO_IRQSTATUS_0[31:0] INTLINE and/or GPIO_IRQSTATUS_1[31:0] INTLINE	0x-
Read input register value.	GPIO_DATAIN[31:0] DATAIN	0x-
Clear interrupt status	GPIO_IRQSTATUS_0[31:0] INTLINE and/or GPIO_IRQSTATUS_1[31:0] INTLINE	0xFFFF FFFF

20.5.1.2.2 General-Purpose Interface Set Bit Function

Table 20-14. General-Purpose Interface Set Bit Function

Step	Register/Bit Field/Programming Model	Value
Write 0x1 to set desired bit(s) in DATAOUT register.	GPIO_SETDATAOUT [31:0] INTLINE	0x-

20.5.1.2.3 General-Purpose Interface Clear Bit Function

Table 20-15. General-Purpose Interface Clear Bit Function

Step	Register/Bit Field/Programming Model	Value
Write 0x1 to clear desired bit(s) in DATAOUT register.	GPIO_CLEARDATAOUT [31:0] INTLINE	0x-

20.6 General-Purpose Interface Register Manual

20.6.1 General-Purpose Interface Instance Summary

Table 20-16 summarizes the general-purpose interface instance.

Table 20-16. Instance Summary

Module Name	Module Base Address	Size
GPIO2	0x4805 5000	408 Bytes
GPIO3	0x4805 7000	408 Bytes
GPIO4	0x4805 9000	408 Bytes
GPIO1	0x4AE1 0000	408 Bytes

20.6.2 General-Purpose Interface Registers

20.6.2.1 General-Purpose Interface Register Summary

Table 20-17 summarizes the general-purpose interface GPIO2, GPIO3 and GPIO4 registers.

Table 20-17. General-Purpose Interface GPIO2, GPIO3 and GPIO4 Registers Summary

Register Name	Type	Register Width (Bits)	Address Offset	GPIO2 L4_PER1 Physical Address	GPIO3 L4_PER1 Physical Address	GPIO4 L4_PER1 Physical Address
GPIO_REVISION	R	32	0x0000 0000	0x4805 5000	0x4805 7000	0x4805 9000
GPIO_SYSCONFIG	RW	32	0x0000 0010	0x4805 5010	0x4805 7010	0x4805 9010
GPIO_EOI	W	32	0x0000 0020	0x4805 5020	0x4805 7020	0x4805 9020
GPIO_IRQSTATUS_RAW_0	RW	32	0x0000 0024	0x4805 5024	0x4805 7024	0x4805 9024
GPIO_IRQSTATUS_RAW_1	RW	32	0x0000 0028	0x4805 5028	0x4805 7028	0x4805 9028
GPIO_IRQSTATUS_0	RW	32	0x0000 002C	0x4805 502C	0x4805 702C	0x4805 902C
GPIO_IRQSTATUS_1	RW	32	0x0000 0030	0x4805 5030	0x4805 7030	0x4805 9030
GPIO_IRQSTATUS_SET_0	RW	32	0x0000 0034	0x4805 5034	0x4805 7034	0x4805 9034
GPIO_IRQSTATUS_SET_1	RW	32	0x0000 0038	0x4805 5038	0x4805 7038	0x4805 9038
GPIO_IRQSTATUS_CLR_0	RW	32	0x0000 003C	0x4805 503C	0x4805 703C	0x4805 903C
GPIO_IRQSTATUS_CLR_1	RW	32	0x0000 0040	0x4805 5040	0x4805 7040	0x4805 9040
GPIO_IRQWAKEN_0	RW	32	0x0000 0044	0x4805 5044	0x4805 7044	0x4805 9044
GPIO_IRQWAKEN_1	RW	32	0x0000 0048	0x4805 5048	0x4805 7048	0x4805 9048
GPIO_SYSSTATUS	R	32	0x0000 0114	0x4805 5114	0x4805 7114	0x4805 9114
RESERVED	RW	32	0x0000 0118	0x4805 5118	0x4805 7118	0x4805 9118
RESERVED	RW	32	0x0000 011C	0x4805 511C	0x4805 711C	0x4805 911C
RESERVED	RW	32	0x0000 0120	0x4805 5120	0x4805 7120	0x4805 9120
RESERVED	RW	32	0x0000 0128	0x4805 5128	0x4805 7128	0x4805 9128
RESERVED	RW	32	0x0000 012C	0x4805 512C	0x4805 712C	0x4805 912C
GPIO_CTRL	RW	32	0x0000 0130	0x4805 5130	0x4805 7130	0x4805 9130

Table 20-17. General-Purpose Interface GPIO2, GPIO3 and GPIO4 Registers Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	GPIO2 L4_PER1 Physical Address	GPIO3 L4_PER1 Physical Address	GPIO4 L4_PER1 Physical Address
GPIO_OE	RW	32	0x0000 0134	0x4805 5134	0x4805 7134	0x4805 9134
GPIO_DATAIN	R	32	0x0000 0138	0x4805 5138	0x4805 7138	0x4805 9138
GPIO_DATAOUT	RW	32	0x0000 013C	0x4805 513C	0x4805 713C	0x4805 913C
GPIO_LEVELDETECT0	RW	32	0x0000 0140	0x4805 5140	0x4805 7140	0x4805 9140
GPIO_LEVELDETECT1	RW	32	0x0000 0144	0x4805 5144	0x4805 7144	0x4805 9144
GPIO_RISINGDETECT	RW	32	0x0000 0148	0x4805 5148	0x4805 7148	0x4805 9148
GPIO_FALLINGDETECT	RW	32	0x0000 014C	0x4805 514C	0x4805 714C	0x4805 914C
GPIO_DEBOUNCENABLE	RW	32	0x0000 0150	0x4805 5150	0x4805 7150	0x4805 9150
GPIO_DEBOUNCINGTIME	RW	32	0x0000 0154	0x4805 5154	0x4805 7154	0x4805 9154
RESERVED	RW	32	0x0000 0160	0x4805 5160	0x4805 7160	0x4805 9160
RESERVED	RW	32	0x0000 0164	0x4805 5164	0x4805 7164	0x4805 9164
RESERVED	RW	32	0x0000 0170	0x4805 5170	0x4805 7170	0x4805 9170
RESERVED	RW	32	0x0000 0174	0x4805 5174	0x4805 7174	0x4805 9174
RESERVED	RW	32	0x0000 0180	0x4805 5180	0x4805 7180	0x4805 9180
RESERVED	RW	32	0x0000 0184	0x4805 5184	0x4805 7184	0x4805 9184
GPIO_CLEARDATAOUT	RW	32	0x0000 0190	0x4805 5190	0x4805 7190	0x4805 9190
GPIO_SETDATAOUT	RW	32	0x0000 0194	0x4805 5194	0x4805 7194	0x4805 9194

Table 20-18 summarizes the general-purpose interface GPIO1 registers.

Table 20-18. General-Purpose Interface GPIO1 Registers Summary

Register Name	Type	Register Width (Bits)	Address Offset	GPIO1 L4_WKUP Physical Address
GPIO_REVISION	R	32	0x0000 0000	0x4AE1 0000
GPIO_SYSCONFIG	RW	32	0x0000 0010	0x4AE1 0010
GPIO_EOI	W	32	0x0000 0020	0x4AE1 0020
GPIO_IRQSTATUS_RAW_0	RW	32	0x0000 0024	0x4AE1 0024
GPIO_IRQSTATUS_RAW_1	RW	32	0x0000 0028	0x4AE1 0028
GPIO_IRQSTATUS_0	RW	32	0x0000 002C	0x4AE1 002C
GPIO_IRQSTATUS_1	RW	32	0x0000 0030	0x4AE1 0030
GPIO_IRQSTATUS_SET_0	RW	32	0x0000 0034	0x4AE1 0034
GPIO_IRQSTATUS_SET_1	RW	32	0x0000 0038	0x4AE1 0038
GPIO_IRQSTATUS_CLR_0	RW	32	0x0000 003C	0x4AE1 003C
GPIO_IRQSTATUS_CLR_1	RW	32	0x0000 0040	0x4AE1 0040
GPIO_IRQWAKEN_0	RW	32	0x0000 0044	0x4AE1 0044
GPIO_IRQWAKEN_1	RW	32	0x0000 0048	0x4AE1 0048
GPIO_SYSSTATUS	R	32	0x0000 0114	0x4AE1 0114

Table 20-18. General-Purpose Interface GPIO1 Registers Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	GPIO1 L4_WKUP Physical Address
RESERVED	RW	32	0x0000 0118	0x4AE1 0118
RESERVED	RW	32	0x0000 011C	0x4AE1 011C
RESERVED	RW	32	0x0000 0120	0x4AE1 0120
RESERVED	RW	32	0x0000 0128	0x4AE1 0128
RESERVED	RW	32	0x0000 012C	0x4AE1 012C
GPIO_CTRL	RW	32	0x0000 0130	0x4AE1 0130
GPIO_OE	RW	32	0x0000 0134	0x4AE1 0134
GPIO_DATAIN	R	32	0x0000 0138	0x4AE1 0138
GPIO_DATAOUT	RW	32	0x0000 013C	0x4AE1 013C
GPIO_LEVELDETECT0	RW	32	0x0000 0140	0x4AE1 0140
GPIO_LEVELDETECT1	RW	32	0x0000 0144	0x4AE1 0144
GPIO_RISINGDETECT	RW	32	0x0000 0148	0x4AE1 0148
GPIO_FALLINGDETECT	RW	32	0x0000 014C	0x4AE1 014C
GPIO_DEBOUNCENABLE	RW	32	0x0000 0150	0x4AE1 0150
GPIO_DEBOUNCINGTIME	RW	32	0x0000 0154	0x4AE1 0154
RESERVED	RW	32	0x0000 0160	0x4AE1 0160
RESERVED	RW	32	0x0000 0164	0x4AE1 0164
RESERVED	RW	32	0x0000 0170	0x4AE1 0170
RESERVED	RW	32	0x0000 0174	0x4AE1 0174
RESERVED	RW	32	0x0000 0180	0x4AE1 0180
RESERVED	RW	32	0x0000 0184	0x4AE1 0184
GPIO_CLEARDATAOUT	RW	32	0x0000 0190	0x4AE1 0190
GPIO_SETDATAOUT	RW	32	0x0000 0194	0x4AE1 0194

20.6.2.2 General-Purpose Interface Register Description

through describe the individual general-purpose interface registers.

Table 20-19. GPIO_REVISION

Address Offset	0x0000 0000		
Physical Address	0x4805 5000 0x4805 7000 0x4805 9000 0x4AE1 0000	**Instance**	GPIO2 GPIO3 GPIO4 GPIO1
Description	IP revision identifier (X.Y.R)		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REVISION																															

Bits	Field Name	Description	Type	Reset
31:0	REVISION	IP revision	R	See(1)

(1) TI Internal Data

Table 20-20. Register Call Summary for Register GPIO_REVISION

General-Purpose Interface Register Manual • General-Purpose Interface Register Summary: [1][4]

Table 20-21. GPIO_SYSCONFIG

Address Offset	0x0000 0010		
Physical Address	0x4805 5010 0x4805 7010 0x4805 9010 0x4AE1 0010	**Instance**	GPIO2 GPIO3 GPIO4 GPIO1
Description	System configuration register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																											IDLEMODE				ENAWAKEUP	SOFTRESET	AUTOIDLE

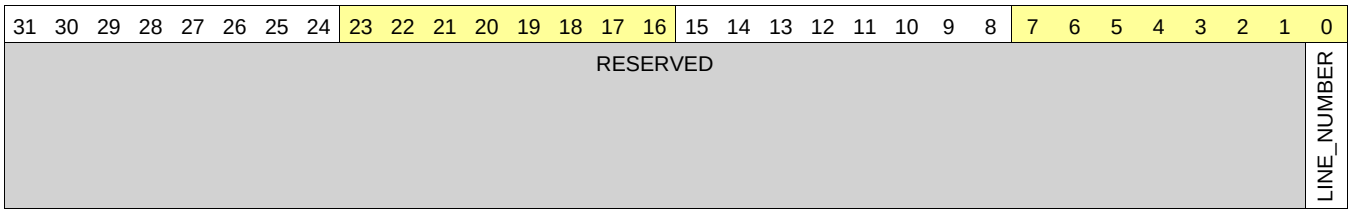
Bits	Field Name	Description	Type	Reset
31:5	RESERVED	Reserved	R	0x0000000
4:3	IDLEMODE	0x0: Force-idle: An IDLE request is acknowledged unconditionally. 0x1: No-idle: An IDLE request is never acknowledged. 0x2: Smart-idle: The acknowledgment to an IDLE request is given based on the internal activity (see Section 20.4.5.2.3, *System Power Management and Wakeup*). 0x3: Smart-idle wakeup	RW	0x0
2	ENAWAKEUP	Wake-up control. 0x0: Wake-up generation is disabled. 0x1: Wake-up capability is enabled upon expected transition on input GPIO pin	RW	0
1	SOFTRESET	Software reset. Set this bit to 1 to trigger a module reset. The bit is automatically reset by the hardware. During reads, it always returns 0. 0x0: Normal mode 0x1: The module is reset.	RW	0
0	AUTOIDLE	OCP clock gating control. 0x0: Internal interface OCP clock is free-running. 0x1: Automatic internal OCP clock gating, based on the OCP interface activity	RW	0

Table 20-22. Register Call Summary for Register GPIO_SYSCONFIG

General-Purpose Interface Functional Description • Asynchronous Path: Wake-Up Request Generation: [0] • General-Purpose Interface Hardware and Software Reset: [1][2][3] • Power Management: [4][5][6][7][8][9][10][11][12][13][14]
General-Purpose Interface Programming Guide • Global Initialization: [15][16][17][18]
General-Purpose Interface Register Manual • General-Purpose Interface Register Summary: [20][23]

Table 20-23. GPIO_EOI

Address Offset	0x0000 0020		
Physical Address	0x4805 5020 0x4805 7020 0x4805 9020 0x4AE1 0020	**Instance**	GPIO2 GPIO3 GPIO4 GPIO1
Description	Software end of interrupt.		
Type	W		



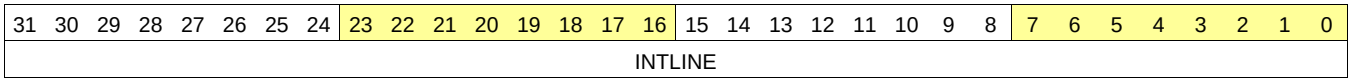
Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved	R	0x0
0	LINE_NUMBER	Software End Of Interrupt (EOI) control. 0x0: EOI for interrupt line number 0. Read returns 0. 0x1: EOI for interrupt line number 1. Read returns 0.	W	0x0

Table 20-24. Register Call Summary for Register GPIO_EOI

General-Purpose Interface Register Manual
- General-Purpose Interface Register Summary: [1][4]

Table 20-25. GPIO_IRQSTATUS_RAW_0

Address Offset	0x0000 0024		
Physical Address	0x4805 5024 0x4805 7024 0x4805 9024 0x4AE1 0024	**Instance**	GPIO2 GPIO3 GPIO4 GPIO1
Description	Per-event raw interrupt status vector, showing all active events (enabled and not enabled), (corresponding to first line of interrupt)		
Type	RW		



Bits	Field Name	Description	Type	Reset
31:0	INTLINE	Status raw for interrupt line. Writing '1' to a bit will set it to '1.' Writing '0' has no effect	RW	0x0000 0000

Table 20-26. Register Call Summary for Register GPIO_IRQSTATUS_RAW_0

General-Purpose Interface Functional Description
- Synchronous Path: Interrupt Request Generation: [0]
- Asynchronous Path: Wake-Up Request Generation: [1][2]

General-Purpose Interface Register Manual
- General-Purpose Interface Register Summary: [4][7]

Table 20-27. GPIO_IRQSTATUS_RAW_1

Address Offset	0x0000 0028		
Physical Address	0x4805 5028 0x4805 7028 0x4805 9028 0x4AE1 0028	**Instance**	GPIO2 GPIO3 GPIO4 GPIO1
Description	Per-event raw interrupt status vector, showing all active events (enabled and not enabled), (corresponding to second line of interrupt)		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTLINE																															

Bits	Field Name	Description	Type	Reset
31:0	INTLINE	Status raw for interrupt line Writing '1' to a bit will set it to '1.' Writing '0' has no effect	RW	0x0000 0000

Table 20-28. Register Call Summary for Register GPIO_IRQSTATUS_RAW_1

General-Purpose Interface Functional Description
- Synchronous Path: Interrupt Request Generation: [0]
- Asynchronous Path: Wake-Up Request Generation: [1]

General-Purpose Interface Register Manual
- General-Purpose Interface Register Summary: [3][6]

Table 20-29. GPIO_IRQSTATUS_0

Address Offset	0x0000 002C		
Physical Address	0x4805 502C 0x4805 702C 0x4805 902C 0x4AE1 002C	**Instance**	GPIO2 GPIO3 GPIO4 GPIO1
Description	Per-event interrupt status vector, showing all active and enabled events (corresponding to first line of interrupt)		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTLINE																															

Bits	Field Name	Description	Type	Reset
31:0	INTLINE	Status for interrupt line Writing 1 to a bit will clear it to 0. Writing 0 has no effect.	RW	0x0000 0000

Table 20-30. Register Call Summary for Register GPIO_IRQSTATUS_0

General-Purpose Interface Functional Description
- Interrupt (or Wake-Up) Line Release: [0]
- Interrupt Requests Generation: [1][2][3][4][5][6][7]

General-Purpose Interface Programming Guide
- Global Initialization: [8]
- General-Purpose Interface Operational Modes Configuration: [9][10]

General-Purpose Interface Register Manual
- General-Purpose Interface Register Summary: [12][15]

Table 20-31. GPIO_IRQSTATUS_1

Address Offset	0x0000 0030		
Physical Address	0x4805 5030 0x4805 7030 0x4805 9030 0x4AE1 0030	**Instance**	GPIO2 GPIO3 GPIO4 GPIO1
Description	Per-event enabled interrupt status vector, showing all active and enabled events (corresponding to second line of interrupt)		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTLINE																															

Bits	Field Name	Description	Type	Reset
31:0	INTLINE	Status for interrupt line Writing 1 to a bit will clear it to 0. Writing 0 has no effect.	RW	0x0000 0000

Table 20-32. Register Call Summary for Register GPIO_IRQSTATUS_1

General-Purpose Interface Functional Description
- Interrupt (or Wake-Up) Line Release: [0]
- Interrupt Requests Generation: [1][2][3][4][5][6][7]

General-Purpose Interface Programming Guide
- Global Initialization: [8]
- General-Purpose Interface Operational Modes Configuration: [9][10]

General-Purpose Interface Register Manual
- General-Purpose Interface Register Summary: [12][15]

Table 20-33. GPIO_IRQSTATUS_SET_0

Address Offset	0x0000 0034		
Physical Address	0x4805 5034 0x4805 7034 0x4805 9034 0x4AE1 0034	**Instance**	GPIO2 GPIO3 GPIO4 GPIO1
Description	Per-event interrupt-enable set vector (corresponding to first line of interrupt)		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTLINE																															

Bits	Field Name	Description	Type	Reset
31:0	INTLINE	Status set for interrupt line Writing 1 to a bit enables the corresponding interrupt event. Writing 0 has no effect.	RW	0x0000 0000

Table 20-34. Register Call Summary for Register GPIO_IRQSTATUS_SET_0

General-Purpose Interface Functional Description
- General-Purpose Interface Block Diagram: [0]
- Synchronous Path: Interrupt Request Generation: [1]
- Asynchronous Path: Wake-Up Request Generation: [2][3]
- Interrupt Requests Generation: [4][5][6][7][8]
- General-Purpose Interface Data Input/Output Capabilities: [9][10][11]
- Description: [12]
- Set Instruction: [13][14][15]

Table 20-34. Register Call Summary for Register GPIO_IRQSTATUS_SET_0 (continued)

General-Purpose Interface Programming Guide • Global Initialization: [16]
General-Purpose Interface Register Manual • General-Purpose Interface Register Summary: [18][21]

Table 20-35. GPIO_IRQSTATUS_SET_1

Address Offset	0x0000 0038		
Physical Address	0x4805 5038 0x4805 7038 0x4805 9038 0x4AE1 0038	**Instance**	GPIO2 GPIO3 GPIO4 GPIO1
Description	Per-event enable set interrupt vector (corresponding to second line of interrupt)		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTLINE																															

Bits	Field Name	Description	Type	Reset
31:0	INTLINE	Status set for interrupt line Writing 1 to a bit enables the corresponding interrupt event. Writing 0 has no effect.	RW	0x0000 0000

Table 20-36. Register Call Summary for Register GPIO_IRQSTATUS_SET_1

General-Purpose Interface Functional Description • General-Purpose Interface Block Diagram: [0] • Synchronous Path: Interrupt Request Generation: [1] • Interrupt Requests Generation: [2][3][4][5][6] • General-Purpose Interface Data Input/Output Capabilities: [7] • Description: [8] • Set Instruction: [9][10][11]
General-Purpose Interface Programming Guide • Global Initialization: [12]
General-Purpose Interface Register Manual • General-Purpose Interface Register Summary: [14][17]

Table 20-37. GPIO_IRQSTATUS_CLR_0

Address Offset	0x0000 003C		
Physical Address	0x4805 503C 0x4805 703C 0x4805 903C 0x4AE1 003C	**Instance**	GPIO2 GPIO3 GPIO4 GPIO1
Description	Per-event interrupt-enable clear vector (corresponding to first line of interrupt)		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTLINE																															

Bits	Field Name	Description	Type	Reset
31:0	INTLINE	Status clear for interrupt line Writing 1 to a bit disables the corresponding interrupt event. Writing 0 has no effect.	RW	0x0000 0000

Table 20-38. Register Call Summary for Register GPIO_IRQSTATUS_CLR_0

General-Purpose Interface Functional Description

- General-Purpose Interface Block Diagram: [0]
- Synchronous Path: Interrupt Request Generation: [1]
- Asynchronous Path: Wake-Up Request Generation: [2]
- Interrupt Requests Generation: [3][4]
- Description: [5]
- Clear Instruction: [6][7][8]

General-Purpose Interface Register Manual

- General-Purpose Interface Register Summary: [10][13]

Table 20-39. GPIO_IRQSTATUS_CLR_1

Address Offset	0x0000 0040		
Physical Address	0x4805 5040 0x4805 7040 0x4805 9040 0x4AE1 0040	**Instance**	GPIO2 GPIO3 GPIO4 GPIO1
Description	Per-event enable clear interrupt vector (corresponding to second line of interrupt)		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTLINE																															

Bits	Field Name	Description	Type	Reset
31:0	INTLINE	Status clear for interrupt line Writing 1 to a bit disables the corresponding interrupt event. Writing 0 has no effect.	RW	0x0000 0000

Table 20-40. Register Call Summary for Register GPIO_IRQSTATUS_CLR_1

General-Purpose Interface Functional Description

- General-Purpose Interface Block Diagram: [0]
- Synchronous Path: Interrupt Request Generation: [1]
- Asynchronous Path: Wake-Up Request Generation: [2]
- Interrupt Requests Generation: [3][4]
- Description: [5]
- Clear Instruction: [6][7][8]

General-Purpose Interface Register Manual

- General-Purpose Interface Register Summary: [10][13]

Table 20-41. GPIO_IRQWAKEN_0

Address Offset	0x0000 0044		
Physical Address	0x4805 5044 0x4805 7044 0x4805 9044 0x4AE1 0044	**Instance**	GPIO2 GPIO3 GPIO4 GPIO1
Description	Per-event wake-up enable set vector (corresponding to first line of interrupt)		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTLINE																															

Bits	Field Name	Description	Type	Reset
31:0	INTLINE	Wakeup set for interrupt line Setting a bit to 1 will enable wakeup for the corresponding event. Setting a bit to 0 will disable wakeup for the corresponding event.	RW	0x0000 0000

Table 20-42. Register Call Summary for Register GPIO_IRQWAKEN_0

General-Purpose Interface Functional Description • Asynchronous Path: Wake-Up Request Generation: [0] • Power Management: [1] • Interrupt Requests Generation: [2] • Wake-Up Requests Generation: [3] • General-Purpose Interface Data Input/Output Capabilities: [4]
General-Purpose Interface Programming Guide • Global Initialization: [5]
General-Purpose Interface Register Manual • General-Purpose Interface Register Summary: [7][10]

Table 20-43. GPIO_IRQWAKEN_1

Address Offset	0x0000 0048		
Physical Address	0x4805 5048 0x4805 7048 0x4805 9048 0x4AE1 0048	**Instance**	GPIO2 GPIO3 GPIO4 GPIO1
Description	Per-event wake-up enable set vector (corresponding to second line of interrupt)		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTLINE																															

Bits	Field Name	Description	Type	Reset
31:0	INTLINE	Wakeup set for interrupt line Setting a bit to 1 will enable wakeup for the corresponding event. Setting a bit to 0 will disable wakeup for the corresponding event.	RW	0x0000 0000

Table 20-44. Register Call Summary for Register GPIO_IRQWAKEN_1

General-Purpose Interface Functional Description • Asynchronous Path: Wake-Up Request Generation: [0] • Power Management: [1] • Interrupt Requests Generation: [2] • Wake-Up Requests Generation: [3] • General-Purpose Interface Data Input/Output Capabilities: [4]
General-Purpose Interface Programming Guide • Global Initialization: [5]
General-Purpose Interface Register Manual • General-Purpose Interface Register Summary: [7][10]

Table 20-45. GPIO_SYSSTATUS

Address Offset	0x0000 0114		
Physical Address	0x4805 5114 0x4805 7114 0x4805 9114 0x4AE1 0114	**Instance**	GPIO2 GPIO3 GPIO4 GPIO1
Description	System status register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															RESETDONE

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved	R	0x0000 0000
0	RESETDONE	Read 0x0: Internal reset is ongoing. Read 0x1: Reset completed	R	0

Table 20-46. Register Call Summary for Register GPIO_SYSSTATUS

General-Purpose Interface Functional Description
- General-Purpose Interface Hardware and Software Reset: [0][1]

General-Purpose Interface Programming Guide
- Global Initialization: [2]

General-Purpose Interface Register Manual
- General-Purpose Interface Register Summary: [4][7]

Table 20-47. GPIO_CTRL

Address Offset	0x0000 0130		
Physical Address	0x4805 5130 0x4805 7130 0x4805 9130 0x4AE1 0130	**Instance**	GPIO2 GPIO3 GPIO4 GPIO1
Description	GPIO control register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																													GATINGRATIO		DISABLEMODULE

Bits	Field Name	Description	Type	Reset
31:3	RESERVED	Reserved	R	0x0000 0000
2:1	GATINGRATIO	Clock gating ratio for event detection 0x0: N = 1 0x1: N = 2 0x2: N = 4 0x3: N = 8	RW	0x1
0	DISABLEMODULE	0x0: Module is enabled, clocks are not gated. 0x1: Module is disabled, internal clocks are gated	RW	0

Table 20-48. Register Call Summary for Register GPIO_CTRL

General-Purpose Interface Functional Description • Clocking: [0] • Power Management: [1][2][3][4][5][6][7]
General-Purpose Interface Programming Guide • Global Initialization: [8]
General-Purpose Interface Register Manual • General-Purpose Interface Register Summary: [10][13]

Table 20-49. GPIO_OE

Address Offset	0x0000 0134		
Physical Address	0x4805 5134 0x4805 7134 0x4805 9134 0x4AE1 0134	**Instance**	GPIO2 GPIO3 GPIO4 GPIO1
Description	Output enable register. 0 = Output enabled ; 1 = Output disabled		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OUTPUTEN																															

Bits	Field Name	Description	Type	Reset
31:0	OUTPUTEN	Output enable 0x0: Output enabled 0x1: Output disabled	RW	0xFFFF FFFF

Table 20-50. Register Call Summary for Register GPIO_OE

General-Purpose Interface Overview • General-Purpose Interface Overview: [0]
General-Purpose Interface Functional Description • General-Purpose Interface Data Input/Output Capabilities: [1][2][3]
General-Purpose Interface Programming Guide • Global Initialization: [4]
General-Purpose Interface Register Manual • General-Purpose Interface Register Summary: [6][9]

Table 20-51. GPIO_DATAIN

Address Offset	0x0000 0138		
Physical Address	0x4805 5138 0x4805 7138 0x4805 9138 0x4AE1 0138	**Instance**	GPIO2 GPIO3 GPIO4 GPIO1
Description	Data input register (with sampled input data)		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATAIN																															

Bits	Field Name	Description	Type	Reset
31:0	DATAIN	Sampled input data	R	0x0000 0000

Table 20-52. Register Call Summary for Register GPIO_DATAIN

General-Purpose Interface Overview
- General-Purpose Interface Overview: [0]

General-Purpose Interface Functional Description
- Power Management: [1][2]
- General-Purpose Interface Data Input/Output Capabilities: [3]

General-Purpose Interface Programming Guide
- General-Purpose Interface Operational Modes Configuration: [4]

General-Purpose Interface Register Manual
- General-Purpose Interface Register Summary: [6][9]

Table 20-53. GPIO_DATAOUT

Address Offset	0x0000 013C		
Physical Address	0x4805 513C 0x4805 713C 0x4805 913C 0x4AE1 013C	**Instance**	GPIO2 GPIO3 GPIO4 GPIO1
Description	Data-output register (data to set on output pins)		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DATAOUT																															

Bits	Field Name	Description	Type	Reset
31:0	DATAOUT	Data to set on output pins	RW	0x0000 0000

Table 20-54. Register Call Summary for Register GPIO_DATAOUT

General-Purpose Interface Overview
- General-Purpose Interface Overview: [0][1]

General-Purpose Interface Functional Description
- General-Purpose Interface Data Input/Output Capabilities: [2]
- Description: [3]

General-Purpose Interface Register Manual
- General-Purpose Interface Register Summary: [5][8]

Table 20-55. GPIO_LEVELDETECT0

Address Offset	0x0000 0140		
Physical Address	0x4805 5140 0x4805 7140 0x4805 9140 0x4AE1 0140	**Instance**	GPIO2 GPIO3 GPIO4 GPIO1
Description	Detect low-level register. 0 = Low-level detection disabled; 1 = Low-level detection enabled		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LEVELDETECT0																															

Bits	Field Name	Description	Type	Reset
31:0	LEVELDETECT0	Low-level detection 0x0: Low-level detection disabled 0x1: Low-level detection enabled	RW	0x0000 0000

Table 20-56. Register Call Summary for Register GPIO_LEVELDETECT0

General-Purpose Interface Programming Guide
- Global Initialization: [0]

General-Purpose Interface Register Manual
- General-Purpose Interface Register Summary: [2][5]

Table 20-57. GPIO_LEVELDETECT1

Address Offset	0x0000 0144		
Physical Address	0x4805 5144 0x4805 7144 0x4805 9144 0x4AE1 0144	**Instance**	GPIO2 GPIO3 GPIO4 GPIO1
Description	Detect high-level register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
LEVELDETECT1																															

Bits	Field Name	Description	Type	Reset
31:0	LEVELDETECT1	0x0: High-evel detection disabled 0x1: High-level detection enabled	RW	0x0000 0000

Table 20-58. Register Call Summary for Register GPIO_LEVELDETECT1

General-Purpose Interface Programming Guide
- Global Initialization: [0]

General-Purpose Interface Register Manual
- General-Purpose Interface Register Summary: [2][5]

Table 20-59. GPIO_RISINGDETECT

Address Offset	0x0000 0148		
Physical Address	0x4805 5148 0x4805 7148 0x4805 9148 0x4AE1 0148	**Instance**	GPIO2 GPIO3 GPIO4 GPIO1
Description	Detect rising edge register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RISINGDETECT																															

Bits	Field Name	Description	Type	Reset
31:0	RISINGDETECT	0x0: Rising edge detection disabled 0x1: Rising edge detection enabled	RW	0x0000 0000

Table 20-60. Register Call Summary for Register GPIO_RISINGDETECT

General-Purpose Interface Programming Guide
- Global Initialization: [0]

General-Purpose Interface Register Manual
- General-Purpose Interface Register Summary: [2][5]

Table 20-61. GPIO_FALLINGDETECT

Address Offset	0x0000 014C		
Physical Address	0x4805 514C 0x4805 714C 0x4805 914C 0x4AE1 014C	**Instance**	GPIO2 GPIO3 GPIO4 GPIO1
Description	Detect falling edge register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FALLINGDETECT																															

Bits	Field Name	Description	Type	Reset
31:0	FALLINGDETECT	0x0: Falling edge detection disabled 0x1: Falling edge detection enabled	RW	0x0000 0000

Table 20-62. Register Call Summary for Register GPIO_FALLINGDETECT

General-Purpose Interface Programming Guide
- Global Initialization: [0]

General-Purpose Interface Register Manual
- General-Purpose Interface Register Summary: [2][5]

Table 20-63. GPIO_DEBOUNCENABLE

Address Offset	0x0000 0150		
Physical Address	0x4805 5150 0x4805 7150 0x4805 9150 0x4AE1 0150	**Instance**	GPIO2 GPIO3 GPIO4 GPIO1
Description	Debouncing enable register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DEBOUNCEENABLE																															

Bits	Field Name	Description	Type	Reset
31:0	DEBOUNCEENABLE	0x0: No debouncing 0x1: Debouncing activated	RW	0x0000 0000

Table 20-64. Register Call Summary for Register GPIO_DEBOUNCENABLE

General-Purpose Interface Functional Description
- Power Management: [0]

General-Purpose Interface Programming Guide
- Global Initialization: [1]

General-Purpose Interface Register Manual
- General-Purpose Interface Register Summary: [3][6]

Table 20-65. GPIO_DEBOUNCINGTIME

Address Offset	0x0000 0154		
Physical Address	0x4805 5154 0x4805 7154 0x4805 9154 0x4AE1 0154	**Instance**	GPIO2 GPIO3 GPIO4 GPIO1
Description	Debouncing value register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								DEBOUNCETIME							

Bits	Field Name	Description	Type	Reset
31:8	RESERVED	Reserved	R	0x000000
7:0	DEBOUNCETIME	8-bit values specifying the debouncing time. It is n-periods of the muxed clock, which can come from either a true 32k oscillator/pad of from the system clock. It depends on which boot mode is selected. For more information see Chapter 25, *Initialization*.	RW	0x00

Table 20-66. Register Call Summary for Register GPIO_DEBOUNCINGTIME

General-Purpose Interface Functional Description
- Synchronous Path: Interrupt Request Generation: [0][1]
- Clocking: [2][3][4]
- Power Management: [5][6]

General-Purpose Interface Programming Guide
- Global Initialization: [7]

Table 20-66. Register Call Summary for Register GPIO_DEBOUNCINGTIME (continued)

General-Purpose Interface Register Manual • General-Purpose Interface Register Summary: [9][12]

Table 20-67. GPIO_CLEARDATAOUT

Address Offset	0x0000 0190		
Physical Address	0x4805 5190 0x4805 7190 0x4805 9190 0x4AE1 0190	**Instance**	GPIO2 GPIO3 GPIO4 GPIO1
Description	Clear data-output register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTLINE																															

Bits	Field Name	Description	Type	Reset
31:0	INTLINE	0x0: No effect 0x1: Clear the corresponding bit in the data-output register	RW	0x0000 0000

Table 20-68. Register Call Summary for Register GPIO_CLEARDATAOUT

General-Purpose Interface Functional Description • General-Purpose Interface Data Input/Output Capabilities: [0] • Clear Instruction: [1]
General-Purpose Interface Programming Guide • General-Purpose Interface Operational Modes Configuration: [2]
General-Purpose Interface Register Manual • General-Purpose Interface Register Summary: [4][7]

Table 20-69. GPIO_SETDATAOUT

Address Offset	0x0000 0194		
Physical Address	0x4805 5194 0x4805 7194 0x4805 9194 0x4AE1 0194	**Instance**	GPIO2 GPIO3 GPIO4 GPIO1
Description	Set data-output register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
INTLINE																															

Bits	Field Name	Description	Type	Reset
31:0	INTLINE	0x0: No effect 0x1: Set the corresponding bit in the data-output register	RW	0x0000 0000

Table 20-70. Register Call Summary for Register GPIO_SETDATAOUT

General-Purpose Interface Functional Description • General-Purpose Interface Data Input/Output Capabilities: [0] • Set Instruction: [1][2][3]
General-Purpose Interface Programming Guide • General-Purpose Interface Operational Modes Configuration: [4]
General-Purpose Interface Register Manual • General-Purpose Interface Register Summary: [6][9]

Pulse-Width Modulation Subsystem

This chapter describes the Pulse-Width Modulation (PWM) subsystem in the device.

Topic **Page**

21.1 PWM Subsystem Resources

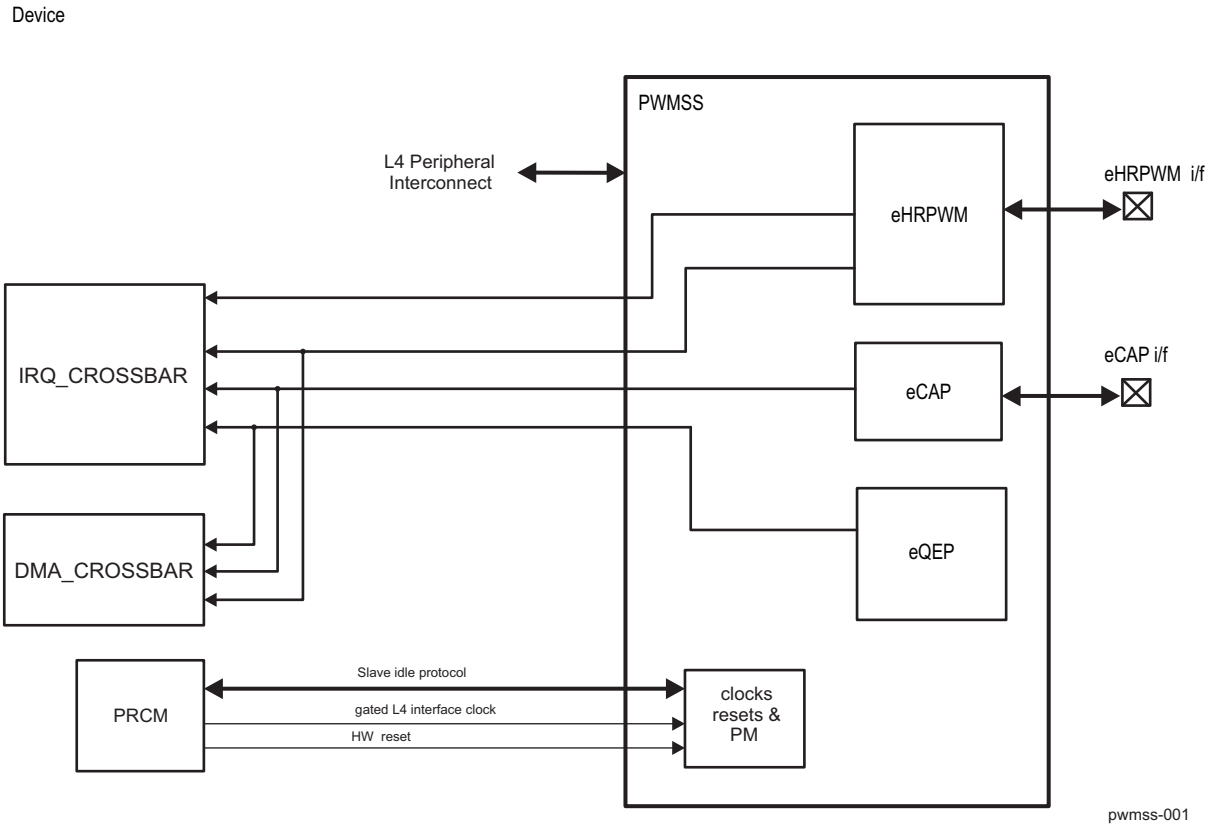
21.1.1 PWMSS Overview

The device has an embedded Pulse Width Modulation Subsystem (PWMSS), which includes one instance of:

- Enhanced High Resolution Pulse Width Modulator (eHRPWM)
- Enhanced Capture (eCAP)
- Enhanced Quadrature Encoded Pulse (eQEP)

Figure 21-1 shows an overview of the device PWMSS.

Figure 21-1. PWMSS Block Diagram



21.1.1.1 PWMSS Key Features

The supported features by the device PWMSS are:

eHRPWM

- Dedicated 16 bit time-base with Period / Frequency control
- Can support 2 independent PWM outputs with Single edge operation
- Can support 2 independent PWM outputs with Dual edge symmetric operation
- Can support 1 independent PWM output with Dual edge asymmetric operation
- Supports Dead-band generation with independent Rising and Falling edge delay control
- Provides asynchronous over-ride control of PWM signals during fault conditions
- Supports “trip zone” allocation of both latched and un-latched fault conditions
- Allows events to trigger both CPU interrupts and start of ADC conversions
- Support PWM chopping by high frequency carrier signal, used for pulse transformer gate drives.

- High-resolution module with programmable delay line:
  - Programmable on a per PWM period basis
  - Can be inserted either on the rising edge or falling edge of the PWM pulse or both or not at all

eCAP

- Dedicated input Capture pin
- 32 bit Time Base (counter)
- 4 x 32 bit Time-stamp Capture registers (PWMSS_ECAP_CAP1 - PWMSS_ECAP_CAP4)
- 4 stage sequencer (Mod4 counter) which is synchronized to external events (ECAP pin edges)
- Independent Edge polarity (Rising / Falling edge) selection for all 4 events
- Input Capture signal pre-scaling (from 1 to 16)
- One-shot compare register (2 bits) to freeze captures after 1 to 4 Time-stamp events
- Control for continuous Time-stamp captures using a 4 deep circular buffer (PWMSS_ECAP_CAP1 - PWMSS_ECAP_CAP4) scheme
- Interrupt capabilities on any of the 4 capture events

eQEP

- Input Synchronization
- Three Stage/Six Stage Digital Noise Filter
- Quadrature Decoder Unit
- Position Counter and Control unit for position measurement
- Quadrature Edge Capture unit for low speed measurement
- Unit Time base for speed/frequency measurement
- Watchdog Timer for detecting stalls

At a PWMSS system level :

- the PWMSS generates 2 x eHRPWM, 1x eCAP and 1 x eQEP event mapped to the device IRQ_CROSSBAR.
- Three of the interrupt events (excluding ePWM tripzone event) are also mapped as DMA requests to the device DMA_CROSSBAR.

21.1.1.2 PWMSS Unsupported Fetaures

The PWMSS limitations in the device are :

- No ePWM inputs are pinned-out (available at the off-chip boundary)
- Only one ePWM tripzone input is pinned-out
- No ePWM digital comparator inputs are pinned-out
- No signals of QEP are pinned-out.

Table 21-1. PWMSS Unsupported Features

Feature	Reason
ePWM inputs	Not pinned out
ePWM tripzone 1-5 inputs	Only Tripzone0 is pinned out
ePWM digital comparators	Inputs not connected
eQEP signals	No signals are connected

21.1.2 PWMSS Environment

21.1.2.1 PWMSS I/O Interface

Table 21-2 shows the device integrated PWMSS interface signals to external devices.

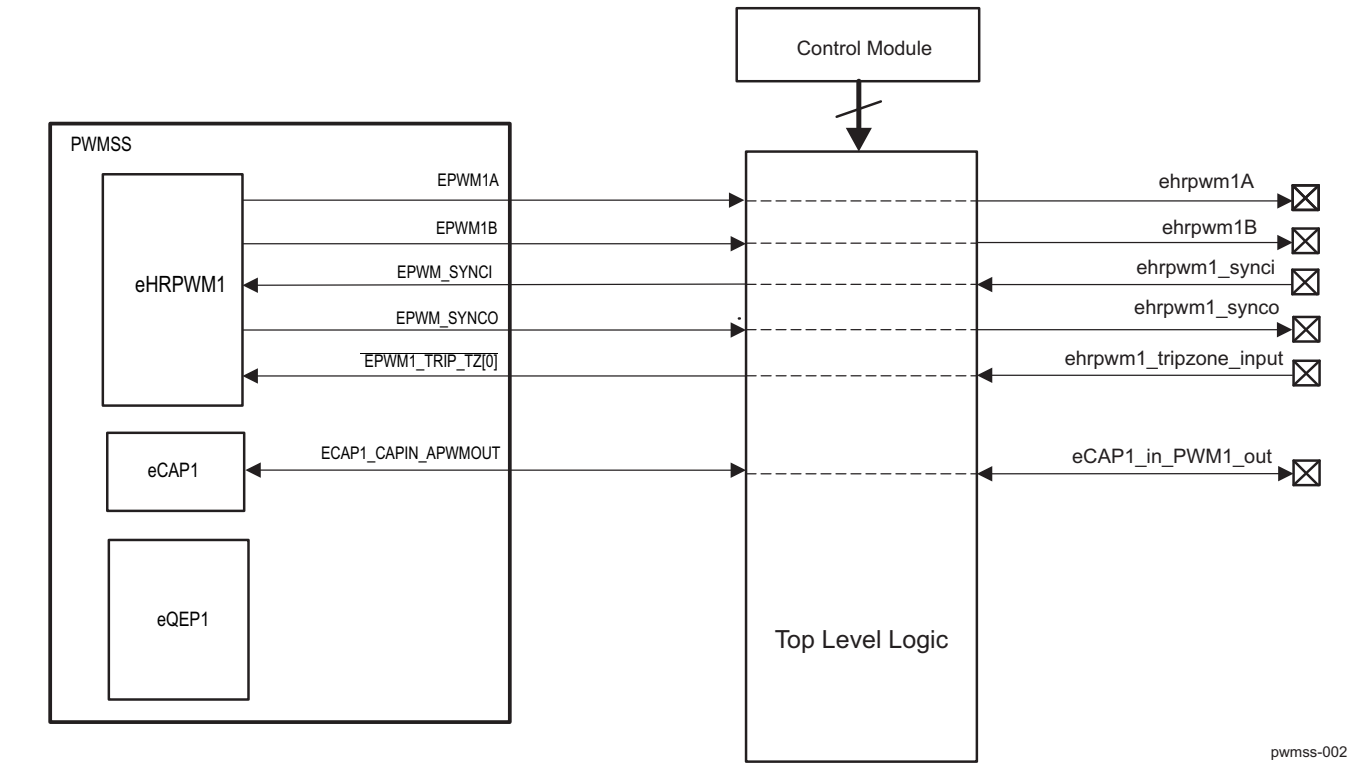
Table 21-2. PWM Subsystem I/O Signals

PWMSS Module Signal Names	Device Level Signal Names	I/O Type[1]	Description	Module Pin Reset Value
PWMSS				
EPWM1A	ehrpwm1A	O	PWM1 output A	0
EPWM1B	ehrpwm1B	O	PWM1 output B	0
EPWM_SYNCI	ehrpwm1_synci	I	PWM1 Sync input	HiZ
EPWM_SYNCO	ehrpwm1_synco	O	PWM1 Sync output	0
$\overline{\text{EPWM1\_TRIP\_TZ[0]}}$	ehrpwm1_tripzone_input	I	PWM1 TripZone input	HiZ
ECAP1_CAPIN_APWMOUT	eCAP1_in_PWM1_out	I/O	eCAP1 Capture input / PWM1 output	HiZ

(1) I = Input; O = Output

Figure 21-2 shows the external interface I/Os for the integrated modules in the PWMSS.

Figure 21-2. PWMSS External Interface I/Os

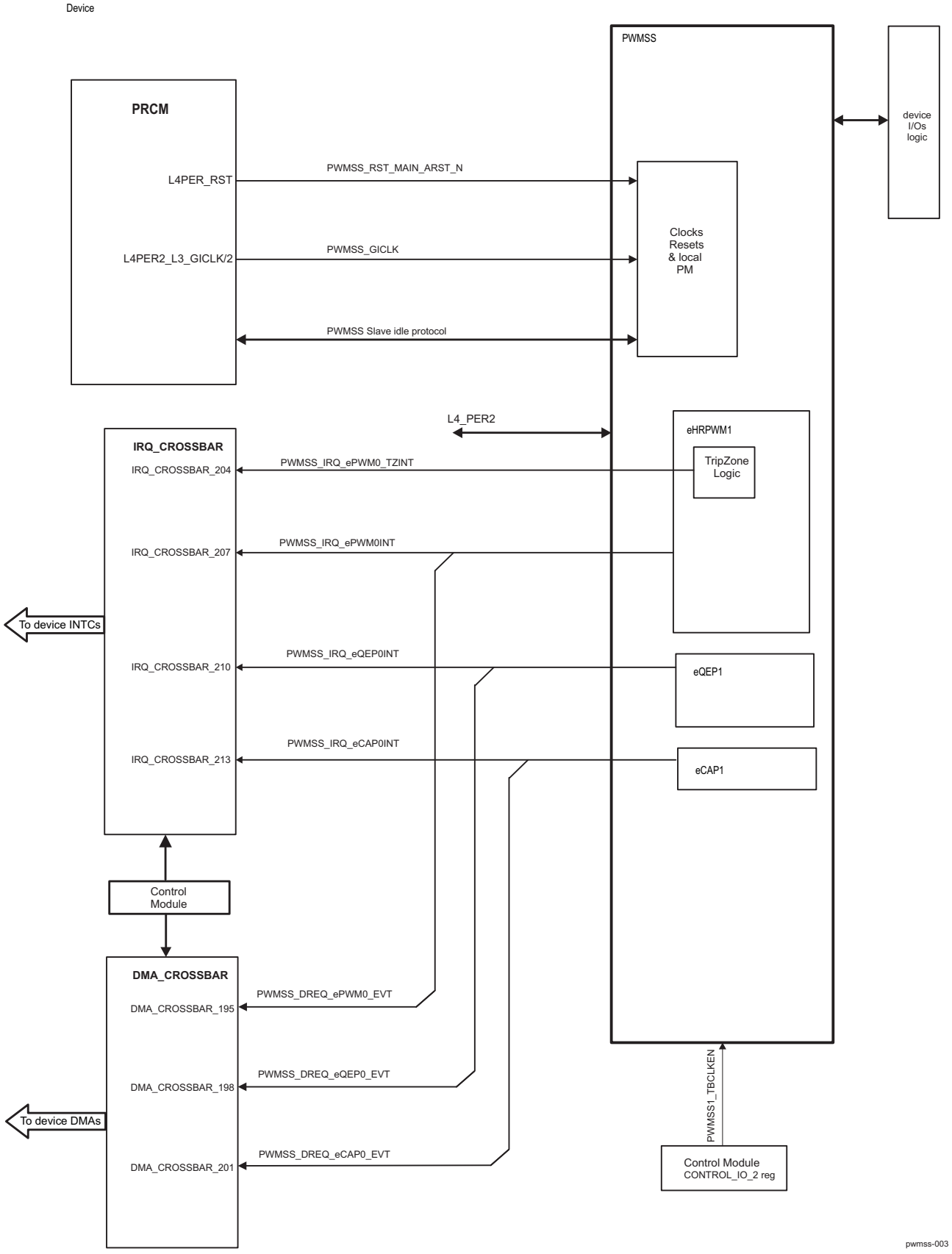


NOTE: The path from module pin to device pad(s) is defined at the device I/O logic level. The I/O logic maps the module signals to the different pads of the device and is programmable in the Control Module registers. For more information, refer to Section 13.4.6.1, *Pad Configuration Registers*, in chapter, *Control Module*.

21.1.3 PWMSS Integration

The Pulse Width Modulation Subsystem (PWMSS) includes a single instance of one pulse width modulator (ePWM) including an Enhanced High Resolution Modulator (eHRPWM), one Enhanced Capture (eCAP), and one Enhanced Quadrature Encoded Pulse (eQEP) modules.

Figure 21-3. PWMSS Integration



At system level the PWMSS integration features:

- A 32-bit slave configuration port on the L4_PER2 interconnect.
- A single gateable interface and functional clock from PRCM to the PWMSS.
- A slave idle protocol with the device PRCM
- A "non-wakeup capable Smart Idle" mode supported

NOTE: For more information about the slave idle protocol, see Section 3.1.1.1.2, *Module Level Clock Management* in Chapter 3, *Power, Reset, and Clock Management*.

- A non-retention hardware reset from PRCM to the PWMSS
- A PWMSS global level software reset which imapcts all registers of the PWMSS at the same time
- 4 hardware events. From these events the PWMSS :
  - generates 4 interrupts to the device IRQ_CROSSBAR
  - generates 3 DMA requests, mapped to the device DMA_CROSSBAR

Table 21-3 through Table 21-5 summarize the integration of the module in the device.

Table 21-3. PWMSS Integration Attributes

Module Instance	Attributes	
	Power Domain	Interconnect
PWMSS	PD_COREAON	L4_PER2

Table 21-4. PWMSS Clocks and Resets

Clocks				
Module Instance	**Destination Signal Name**	**Source Signal Name**	**Source**	**Description**
PWMSS	PWMSS_GICLK	L4PER2_L3_GICLK/2	PRCM	PWMSS gated interface and functional clock
Resets				
Module Instance	**Destination Signal Name**	**Source Signal Name**	**Source**	**Description**
PWMSS	PWMSS_RST_MAIN_ARST_N	L4PER_RST	PRCM	A nonretention hardware main reset to the PWMSS

Table 21-5. PWMSS Hardware Requests

Interrupt Requests				
Module Instance	**Source Signal Name**	**Destination IRQ_CROSSBAR Input**	**Default Mapping**	**Description**
PWMSS	PWMSS_IRQ_ePWM0_TZINT	IRQ_CROSSBAR_204	N/A	eHRPWM1 tripzone event / interrupt. This IRQ source signal is not mapped by default to any device INTC.
	PWMSS_IRQ_ePWM0INT	IRQ_CROSSBAR_207	N/A	eHRPWM1 event / interrupt. This IRQ source signal is not mapped by default to any device INTC.
	PWMSS_IRQ_eQEP0INT	IRQ_CROSSBAR_210	N/A	eQEP1 event / interrupt. This IRQ source signal is not mapped by default to any device INTC.
	PWMSS_IRQ_eCAP0INT	IRQ_CROSSBAR_213	N/A	eCAP1 event / interrupt. This IRQ source signal is not mapped by default to any device INTC.

Table 21-5. PWMSS Hardware Requests (continued)

DMA Requests				
Module Instance	**Source Signal Name**	**Destination DMA_CROSSBAR Input**	**Default Mapping**	**Description**
PWMSS	PWMSS_DREQ_ePWM0_EVT	DMA_CROSSBAR_195	N/A	eHRPWM1 event DMA request. This DMA req source signal is not mapped by default to any device DMA controller.
	PWMSS_DREQ_eQEP0_EVT	DMA_CROSSBAR_198	N/A	eQEP1 event DMA request. This DMA req source signal is not mapped by default to any device DMA controller.
	PWMSS_DREQ_eCAP0_EVT	DMA_CROSSBAR_201	N/A	eCAP1 event DMA request. This DMA req source signal is not mapped by default to any device DMA controller.

NOTE: The **"Default Mapping"** column inTable 21-5,*PWMSS Hardware Requests* shows the default mapping of module IRQ and DREQ source signals. These module IRQ and DREQ source signals can also be mapped to other lines of each device Interrupt or DMA controller through the IRQ_CROSSBAR and DMA_CROSSBAR modules, respectively.

For more information about the IRQ_CROSSBAR module, see Section 13.4.6.4, *IRQ_CROSSBAR Module Functional Description*, in Chapter 13, *Control Module*.

For more information about the DMA_CROSSBAR module, see Section 13.4.6.5, *DMA_CROSSBAR Module Functional Description*, in Chapter 13, *Control Module*.

For more information about the device interrupt controllers, see Chapter 12, *Interrupt Controllers*.

For more information about the device EDMA module, see Chapter 11, *Enhanced DMA*.

21.1.3.1 PWMSS Module Interfaces Implementation

This section describes how the PWMSS submodules - ePWM / eHRPWM, eQEP and eCAP can be used in terms of their functional interfaces considering PWMSS device specific integration.

21.1.3.1.1 Device Specific PWMSS Features

A High-Resolution PWM (HRPWM) modulator is added to the ePWM module, hence the ePWM is signified as eHRPWM.

While the eCAP functionalities are fully implemented in the device, the eHRPWM and eQEP module interface have some restrictions in functionality. The eHRPWM and eQEP restrictions are as follows:

- **For ePWM comparators - only the outputs (ehrpwm1A and ehrpwm1B) are available at chip level**
- **Only one tripzone input pin - EPWM_TRIP_TZ[0] of ePWM / eHRPWM is available to the user at chip level**
- **No eQEP input signals are available to the user at chip level**

The eHRPWM and eQEP functional interface signals which are NOT available to user for the PWMSS are summarized in Table 21-6.

Table 21-6. Device Limitations for the eHRPWM and eQEP Functional Interfaces of the PWMSS

Module Interface	Signal	Description	Comment
PWMSS eHRPWM	EPWM_COMP_EPWMDCMAH	ePWM Comparator A input (HIGH)	Not available at chip boundary (can not be used)
	EPWM_COMP_EPWMDCMAL	ePWM Comparator A input (LOW)	
	EPWM_COMP_EPWMDCMBH	ePWM Comparator B input (HIGH)	
	EPWM_COMP_EPWMDCMBL	ePWM Comparator B input (LOW)	
	EPWM_EPWMA_i	ePWM A input	
	EPWM_EPWMB_i	ePWM B input	
	EPWM_TRIP_TZ[5:1]	ePWM Tripzone inputs [5:1]	
	EPWM_ADC_SOCA	ePWM Start of ADC conversion A output	
	EPWM_ADC_SOCB	ePWM Start of ADC conversion B output	
	EPWM_TRIP_TZ_O[5:0]	ePWM Tripzone outputs	
PWMSS eQEP	EQEP_ERR_PHASE_ERR	eQEP phase error output	Not available at chip boundary (can not be used)
	EQEP_EQEPA_O	eQEP quadrature A output	
	EQEP_EQEPB_O	eQEP quadrature B output	
	EQEP1_A	eQEP1 Quadrature input	
	EQEP1_B	eQEP1 Quadrature input	
	EQEP1_INDEX	eQEP1 Index input/output	
	EQEP1_STROBE	eQEP1 Strobe input/output	

21.1.3.1.2 eHRPWM Module Time Base Clock Gating

PWMSS ePWM / eHRPWM module has an EPWMTBCLKEN module input used to individually **enable / disable its ePWM time-base clock**. This is done through the CTRL_CORE_CONTROL_IO_2[20] PWMSS1_TBCLKEN bit comming from the device control module.

To hold the TBCLK generation counter in its reset state the PWMSS1_TBCLKEN bit should be set to 0b0. When PWMSS1_TBCLKEN is set to 0b1, then the TBCLK generation counter is allowed to count.

For more details on the CTRL_CORE_CONTROL_IO_2, refer to the Section 13.5, *Control Module Register Manual*, in the chapter, *Control Module*.

21.1.4 PWMSS Subsystem Power, Reset and Clock Configuration

21.1.4.1 PWMSS Local Clock Management

The system configuration register - PWMSS_SYSCONFIG is used to configure the local clock management of the PWMSS slave configuration port on the device L4_PER2 interconnect. Note that this register impacts the behaviour of the root interface and functional clock PWMSS_GICLK inside PWMSS shared between the ePWM/eHRPWM, eQEP and eCAP modules. An idle handashake protocol is supported between PWMSS and the device PRCM.

NOTE: For more information about the slave idle protocol, see Section 3.1.1.1.2, *Module Level Clock Management* in Chapter 3, *Power, Reset, and Clock Management*.

Table 21-7. Local IDLE Clock Management Features

Feature	Register bitfield		Description
Slave idle mode	PWMSS_SYSCONFIG IDLEMODE	[3:2]	The available modes are: Force-idle, no-idle, and smart-idle (non-wakeup capable) modes.

NOTE: The device PWM subsystem is part of the CD_L4_PER2 clock domain. For more details on the PWM subsystem top level clock-management modes and control, refer to the Section 3.6.4.1.4, *Clock Domain Module Attributes*, in the Chapter 3, *Power, Reset, and Clock Management*.

21.1.4.2 PWMSS Module Local Clock Gating

NOTE: PWMSS Module Local Clock Gating feature is not supported in this family of devices.

In addition to PWMSS level IDLE clock management, described above, a clock configuration register - PWMSS_CLKCONFIG is used to individually gate (stop) or enable interface and functional clock to the ePWM / eHRPWM, eCAP and eQEP modules. By default, the interface/functional clock is enabled.

The clock status register - PWMSS_CLKSTATUS is used in the PWMSS submodule to indicated acknowledgement of a clock stop or clock enable request.

The PWMSS_CLKCONFIG and PWMSS_CLKSTATUS role is described in Table 21-8.

Table 21-8. Local Module Clock Control and Status Features

Clock Control / Status Feature	PWMSS Submodule	Register bit
Request "stop interface and functional clock" to module	ePWM / eHRPWM	PWMSS_CLKCONFIG[9] EPWM_CLKSTOP_REQ
	eQEP	PWMSS_CLKCONFIG[5] EQEP_CLKSTOP_REQ
	eCAP	PWMSS_CLKCONFIG[1] ECAP_CLKSTOP_REQ
"Stop module interface and functional clock" acknowledged status	ePWM / eHRPWM	PWMSS_CLKSTATUS[9] EPWM_CLKSTOP_ACK
	eQEP	PWMSS_CLKSTATUS[5] EQEP_CLKSTOP_ACK
	eCAP	PWMSS_CLKSTATUS[1] ECAP_CLKSTOP_ACK
Request "enable interface and functional clock" to module	ePWM / eHRPWM	PWMSS_CLKCONFIG[8] EPWM_CLK_EN
	eQEP	PWMSS_CLKCONFIG[4] EQEP_CLK_EN
	eCAP	PWMSS_CLKCONFIG[0] ECAP_CLK_EN
"Enable module Interface and functional clock" acknowledged status	ePWM / eHRPWM	PWMSS_CLKSTATUS[8] EPWM_CLK_EN_ACK
	eQEP	PWMSS_CLKSTATUS[4] EQEP_CLK_EN_ACK
	eCAP	PWMSS_CLKSTATUS[0] ECAP_CLK_EN_ACK

NOTE: In order for the PWMSS to enter "Idle state", all PWMSS submodules must have acknowledged a stop clock request, i.e. the PWMSS_CLKSTATUS bits EPWM_CLKSTOP_ACK, ECAP_CLKSTOP_ACK and EQEP_CLKSTOP_ACK must be raised 'HIGH'.

21.1.4.3 PWMSS Software Reset

The PWMSS_SYSCONFIG[0] SOFTRESET bit can be used to exert a "PWMSS" software reset impacting ePWM / eHRPWM, eCAP and eQEP modules at the same time. The software has to poll the same bit until it is deasserted by HW, to insure software reset of all modules is completed.

21.1.5 PWMSS_CFG Register Manual

This section provides description of the PWM subsystem (top) level functional registers.

21.1.5.1 PWMSS_CFG Instance Summary

Table 21-9. PWMSS_CFG Instance Summary

Module Name	Module Base Address L4_PER2 Interconnect	Size (Bytes)
PWMSS_CFG	0x4843 E000	48 Bytes

21.1.5.2 PWMSS_CFG Registers

21.1.5.2.1 PWMSS_CFG Register Summary

Table 21-10. PWMSS_CFG Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	PWMSS_CFG Physical Address L4_PER2 Interconnect
PWMSS_IDVER	RW	32	0x0000 0000	0x4843 E000
PWMSS_SYSCONFIG	RW	32	0x0000 0004	0x4843 E004
PWMSS_CLKCONFIG	RW	32	0x0000 0008	0x4843 E008
PWMSS_CLKSTATUS	RW	32	0x0000 000C	0x4843 E00C

21.1.5.2.2 PWMSS_CFG Register Description

Table 21-11. PWMSS_IDVER

Address Offset	0x0000 0000		
Physical Address	0x4843 E000	**Instance**	PWMSS_CFG
Description	IP Revision Register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REVISION																															

Bits	Field Name	Description	Type	Reset
31:0	REVISION	IP Revision value	R	0x-(1)

(1) TI Internal data

Table 21-12. Register Call Summary for Register PWMSS_IDVER

PWM Subsystem Resources
- PWMSS_CFG Registers: [0]

Table 21-13. PWMSS_SYSCONFIG

Address Offset	0x0000 0004		
Physical Address	0x4843 E004	**Instance**	PWMSS_CFG
Description	This register controls the PWMSS local Idle mode clock management and software reset		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												IDLEMODE		RESERVED	SOFTRESET

Bits	Field Name	Description	Type	Reset
31:4	RESERVED		R	0x0000
3:2	IDLEMODE	Configuration of the local target state management mode. By definition, the target can handle read/write transaction as long as it is out of IDLE state. 0x0: Force-idle mode: The local target IDLE state follows (acknowledges) the system idle requests unconditionally, that is, regardless of the internal requirements of th e IP module. Backup mode, for debug only. 0x1: No-idle mode: The local target never enters IDLE state. Backup mode, for debug only. 0x2: Smart-idle mode: The local target IDLE state eventually follows (acknowledges) the system idle requests, depending on the internal requirements of the IP module. IP module does not generate (IRQ- or DMA-request-related) wakeup events. 0x3: Reserved	RW	0x2
1	RESERVED		R	0
0	SOFTRESET	Software reset : 0x0 : Software reset is completed 0x1: Software reset assertion	RW	0x0

Table 21-14. Register Call Summary for Register PWMSS_SYSCONFIG

PWM Subsystem Resources

- PWMSS Local Clock Management: [0][1]
- PWMSS Software Reset: [2]
- PWMSS_CFG Registers: [3]

Table 21-15. PWMSS_CLKCONFIG

Address Offset	0x0000 0008		
Physical Address	0x4843 E008	**Instance**	PWMSS_CFG
Description	The clock configuration register is used in the PWMSS for clkstop req and clk_en control to the ePWM/ eHRPWM, eCAP and eQEP submodules. **Note:** PWMSS Module Local Clock Gating feature is not supported. This register should not be modified. Clock gating functionality is controlled by PRCM.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						EPWM_CLKSTOP_REQ	EPWM_CLK_EN	RESERVED		EQEP_CLKSTOP_REQ	EQEP_CLK_EN	RESERVED		ECAP_CLKSTOP_REQ	ECAP_CLK_EN

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x000000
9	EPWM_CLKSTOP_REQ	This bit controls the clock stop input to the ePWM / eHRPWM module : 0: No effect 1: A request to stop interface clock to the module is asserted	RW	0
8	EPWM_CLK_EN	This bit controls the interface clock enable (clk_en) input to the ePWM / eHRPWM module: 0: No effect 1: Enables the interface clock to the module	RW	1
7:6	RESERVED		R	0x0
5	EQEP_CLKSTOP_REQ	This bit controls the clock stop input to the eQEP module : 0: No effect 1: A request to stop interface clock to the module is asserted	RW	0
4	EQEP_CLK_EN	This bit controls the interface clock enable (clk_en) input to the eQEP module : 0: No effect 1: Enables the interface clock to the module	RW	1
3:2	RESERVED		R	0x0
1	ECAP_CLKSTOP_REQ	This bit controls the clock stop input to the eCAP module : 0: No effect 1: A request to stop interface clock to the module is asserted	RW	0
0	ECAP_CLK_EN	This bit controls the interface clock enable (clk_en) input to the eCAP module : 0: No effect 1: Enables the interface clock to the module	RW	1

Table 21-16. Register Call Summary for Register PWMSS_CLKCONFIG

PWM Subsystem Resources

- PWMSS Module Local Clock Gating: [0][1][2][3][4][5][6][7]
- PWMSS_CFG Registers: [8]

Table 21-17. PWMSS_CLKSTATUS

Address Offset	0x0000 000C		
Physical Address	0x4843 E00C	**Instance**	PWMSS_CFG
Description	The clock status register is used in the PWMSS to indicate clock stop acknowledge (clkstop_ack) and clock enable (clk_en) acknowledge status for the ePWM/ eHRPWM, eCAP and eQEP submodules. **Note:** PWMSS Module Local Clock Gating feature is not supported. Clock gating functionality is controlled by PRCM.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						EPWM_CLKSTOP_ACK	EPWM_CLK_EN_ACK	RESERVED		EQEP_CLKSTOP_ACK	EQEP_CLK_EN_ACK	RESERVED		ECAP_CLKSTOP_ACK	ECAP_CLK_EN_ACK

Bits	Field Name	Description	Type	Reset
31:10	RESERVED		R	0x000000
9	EPWM_CLKSTOP_ACK	This bit is the clkstop_req_ack status output of the ePWM / eHRPWM module : 0: No interface clock stop acknowledged 1: Interface clock stop request is acknowledged for the module	R	0
8	EPWM_CLK_EN_ACK	This bit is the clk_en status output of the ePWM / eHRPWM module : 0: No clock enable request acknowledged 1: Interface clock enable request is acknowledged for the module	R	0
7:6	RESERVED		R	0x0
5	EQEP_CLKSTOP_ACK	This bit is the clkstop_req_ack status output of the eQEP module : 0: No interface clock stop acknowledged 1: Interface clock stop request is acknowledged for the module	R	0
4	EQEP_CLK_EN_ACK	This bit is the clk_en status output of the eQEP module : 0: No clock enable request acknowledged 1: Interface clock enable request is acknowledged for the module	R	0
3:2	RESERVED		R	0x0
1	ECAP_CLKSTOP_ACK	TThis bit is the clkstop_req_ack status output of the eCAP module : 0: No interface clock stop acknowledged 1: Interface clock stop request is acknowledged for the module	R	0
0	ECAP_CLK_EN_ACK	TThis bit is the clk_en status output of the eCAP module : 0: No clock enable request acknowledged 1: Interface clock enable request is acknowledged for the module	R	0

Table 21-18. Register Call Summary for Register PWMSS_CLKSTATUS

PWM Subsystem Resources

- PWMSS Module Local Clock Gating: [0][1][2][3][4][5][6][7][8]
- PWMSS_CFG Registers: [9]

21.2 Enhanced PWM (ePWM) Module

21.2.1 ePWM Overview

An effective PWM peripheral must be able to generate complex pulse width waveforms with minimal CPU overhead or intervention. It needs to be highly programmable and very flexible while being easy to understand and use. The ePWM unit described here addresses these requirements by allocating all needed timing and control resources on a per PWM channel basis. Cross coupling or sharing of resources has been avoided; instead, the ePWM is built up from smaller single channel modules with separate resources and that can operate together as required to form a system. This modular approach results in an orthogonal architecture and provides a more transparent view of the peripheral structure, helping users to understand its operation quickly.

The ePWM module represents one complete PWM channel composed of two PWM outputs: EPWM1A and EPWM1B. The ePWM functionality can be extended with the so called **High-Resolution Pulse Width modulator**. The HRPWM functionality is described in Section 21.2.2.10.

The ePWM module supports the following features:

- Dedicated 16-bit time-base counter with period and frequency control
- Two PWM outputs (EPWM1A and EPWM1B) that can be used in the following configurations:
  - Two independent PWM outputs with single-edge operation
  - Two independent PWM outputs with dual-edge symmetric operation
  - One independent PWM output with dual-edge asymmetric operation
- Asynchronous override control of PWM signals through software.
- Hardware-locked (synchronized) phase relationship on a cycle-by-cycle basis.
- Dead-band generation with independent rising and falling edge delay control.
- Programmable trip zone allocation of both cycle-by-cycle trip and one-shot trip on fault conditions.
- A trip condition can force either high, low, or high-impedance state logic levels at PWM outputs.
- Programmable event prescaling minimizes CPU overhead on interrupts.
- PWM chopping by high-frequency carrier signal, useful for pulse transformer gate drives.

The ePWM module consists of seven submodules and is connected within a system via the signals as shown in Figure 21-4.

Figure 21-4. Submodules and Signal Connections for the ePWM Module

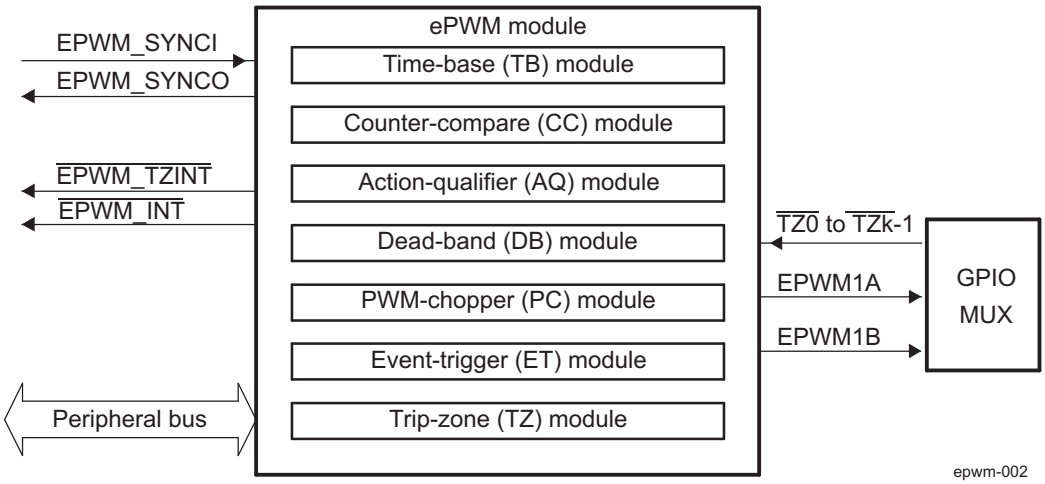
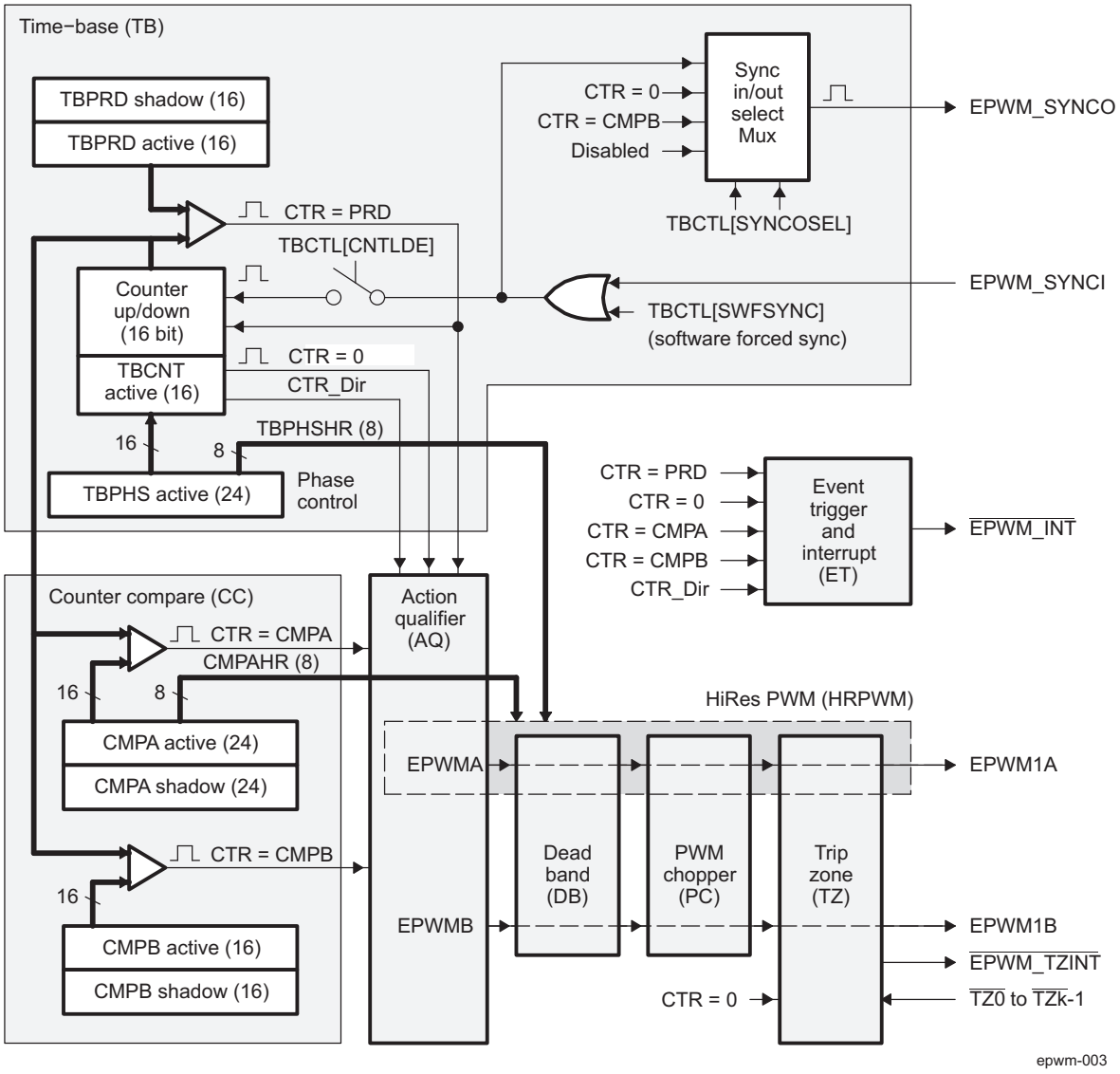


Figure 21-5 shows more internal details of a single ePWM module. The main signals used by the ePWM module are:

- **PWM output signals (EPWM1A and EPWM1B).** The PWM output signals are made available external to the device through the GPIO peripheral described in the system control and interrupts guide for your device.
- **Trip-zone signals ($\overline{\text{TZ0}}$ to $\overline{\text{TZk-1}}$).** These **k** input signals alert the ePWM module of an external fault condition. The ePWM module can be configured to either use or ignore any of the trip-zone signals. The trip-zone signal can be configured as an asynchronous input through the GPIO peripheral.
- Time-base synchronization input (EPWM_SYNCI) and output (EPWM_SYNCO) signals.
- **Peripheral Bus.** The peripheral bus is 32-bits wide and allows both 16-bit and 32-bit writes to the ePWM register file.

Figure 21-5 also shows the key internal submodule interconnect signals. Each submodule is described in Section 21.2.2.

Figure 21-5. ePWM Submodules and Critical Internal Signal Interconnects



21.2.2 ePWM Functional Description

Seven submodules are included in the ePWM peripheral. It also includes a high-resolution submodule that allows more precise control of the PWM outputs.

21.2.2.1 ePWM Submodule Features

Table 21-19 lists the eight key submodules together with a list of their main configuration parameters. For example, if you need to adjust or control the duty cycle of a PWM waveform, then you should see the counter-compare submodule in Section 21.2.2.4 for relevant details.

Table 21-19. Submodule Configuration Parameters

Submodule	Configuration Parameter or Option
Time-base (TB)	• Scale the time-base clock (TBCLK) relative to the system clock (SYSCLKOUT). • Configure the PWM time-base counter (TBCNT) frequency or period. • Set the mode for the time-base counter: – count-up mode: used for asymmetric PWM – count-down mode: used for asymmetric PWM – count-up-and-down mode: used for symmetric PWM • Configure the time-base phase relative to another ePWM module. • Synchronize the time-base counter between modules through hardware or software. • Configure the direction (up or down) of the time-base counter after a synchronization event. • Configure how the time-base counter will behave when the device is halted by an emulator. • Specify the source for the synchronization output of the ePWM module: – Synchronization input signal – Time-base counter equal to zero – Time-base counter equal to counter-compare B (CMPB) – No output synchronization signal generated.
Counter-compare (CC)	• Specify the PWM duty cycle for output EPWM1A and/or output EPWM1B • Specify the time at which switching events occur on the EPWM1A or EPWM1B output
Action-qualifier (AQ)	• Specify the type of action taken when a time-base or counter-compare submodule event occurs: – No action taken – Output EPWM1A and/or EPWM1B switched high – Output EPWM1A and/or EPWM1B switched low – Output EPWM1A and/or EPWM1B toggled • Force the PWM output state through software control • Configure and control the PWM dead-band through software
Dead-band (DB)	• Control of traditional complementary dead-band relationship between upper and lower switches • Specify the output rising-edge-delay value • Specify the output falling-edge delay value • Bypass the dead-band module entirely. In this case the PWM waveform is passed through without modification.
PWM-chopper (PC)	• Create a chopping (carrier) frequency. • Pulse width of the first pulse in the chopped pulse train. • Duty cycle of the second and subsequent pulses. • Bypass the PWM-chopper module entirely. In this case the PWM waveform is passed through without modification.

Table 21-19. Submodule Configuration Parameters (continued)

Submodule	Configuration Parameter or Option
Trip-zone (TZ)	• Configure the ePWM module to react to one, all, or none of the trip-zone pins. • Specify the tripping action taken when a fault occurs: – Force EPWM1A and/or EPWM1B high – Force EPWM1A and/or EPWM1B low – Force EPWM1A and/or EPWM1B to a high-impedance state – Configure EPWM1A and/or EPWM1B to ignore any trip condition. • Configure how often the ePWM will react to the trip-zone pin: – One-shot – Cycle-by-cycle • Enable the trip-zone to initiate an interrupt. • Bypass the trip-zone module entirely.
Event-trigger (ET)	• Enable the ePWM events that will trigger an interrupt. • Specify the rate at which events cause triggers (every occurrence or every second or third occurrence) • Poll, set, or clear event flags
High-Resolution PWM (HRPWM)	• Enable extended time resolution capabilities • Configure finer time granularity control or edge positioning

NOTE: The system clock - SYSCLKOUT is the ePWM functional clock derived from the PWMSS gateable interface and functional clock PWMSS_GICLK, described in Section 21.1.3.

Code examples are provided in the remainder of this chapter that show how to implement various ePWM module configurations. These examples use the constant definitions shown in Example 21-1.

Example 21-1. Constant Definitions Used in the ePWM Code Examples

```
// TBCTL (Time-Base Control)
// = = = = = = = = = = = = = = = = = = = = = = = = = = = = =
// TBCNT MODE bits
#define         TB_COUNT_UP         0x0
#define         TB_COUNT_DOWN       0x1
#define         TB_COUNT_UPDOWN     0x2
#define         TB_FREEZE           0x3
// PHSEN bit
#define         TB_DISABLE          0x0
#define         TB_ENABLE           0x1
// PRDLD bit
#define         TB_SHADOW           0x0
#define         TB_IMMEDIATE        0x1
// SYNCOSEL bits
#define         TB_SYNC_IN          0x0
#define         TB_CTR_ZERO         0x1
#define         TB_CTR_CMPB         0x2
#define         TB_SYNC_DISABLE     0x3
// HSPCLKDIV and CLKDIV bits
#define         TB_DIV1             0x0
#define         TB_DIV2             0x1
#define         TB_DIV4             0x2
// PHSDIR bit
#define         TB_DOWN             0x0
#define         TB_UP               0x1

// CMPCTL (Compare Control)
// = = = = = = = = = = = = = = = = = = = = = = = = = = = = =
// LOADAMODE and LOADBMODE bits
```

Example 21-1. Constant Definitions Used in the ePWM Code Examples (continued)

```
#define         CC_CTR_ZERO        0x0
#define         CC_CTR_PRD         0x1
#define         CC_CTR_ZERO_PRD    0x2
#define         CC_LD_DISABLE      0x3
// SHDWAMODE and SHDWBMODE bits
#define         CC_SHADOW          0x0
#define         CC_IMMEDIATE       0x1
// AQCTLA and AQCTLB (Action-qualifier Control)
// = = = = = = = = = = = = = = = = = = = = = = = = = = = = =
// ZRO, PRD, CAU, CAD, CBU, CBD bits
#define         AQ_NO_ACTION       0x0
#define         AQ_CLEAR           0x1
#define         AQ_SET             0x2
#define         AQ_TOGGLE          0x3
// DBCTL (Dead-Band Control)
// = = = = = = = = = = = = = = = = = = = = = = = = = = = = =
// MODE bits
#define         DB_DISABLE         0x0
#define         DBA_ENABLE         0x1
#define         DBB_ENABLE         0x2
#define DB_FULL_ENABLE 0x3
// POLSEL bits
#define         DB_ACTV_HI         0x0
#define         DB_ACTV_LOC        0x1
#define         DB_ACTV_HIC        0x2
#define         DB_ACTV_LO         0x3
// PCCTL (chopper control)
// = = = = = = = = = = = = = = = = = = = = = = = = = = = = =
// CHPEN bit
#define         CHP_ENABLE         0x0
#define CHP_DISABLE 0x1
// CHPFREQ bits
#define         CHP_DIV1           0x0
#define         CHP_DIV2           0x1
#define         CHP_DIV3           0x2
#define         CHP_DIV4           0x3
#define         CHP_DIV5           0x4
#define         CHP_DIV6           0x5
#define         CHP_DIV7           0x6
#define         CHP_DIV8           0x7
// CHPDUTY bits
#define         CHP1_8TH           0x0
#define         CHP2_8TH           0x1
#define         CHP3_8TH           0x2
#define         CHP4_8TH           0x3
#define         CHP5_8TH           0x4
#define         CHP6_8TH           0x5
#define         CHP7_8TH           0x6
// TZSEL (Trip-zone Select)
// = = = = = = = = = = = = = = = = = = = = = = = = = = = = =
// CBCn and OSHTn bits
#define         TZ_ENABLE          0x0
#define         TZ_DISABLE         0x1
// TZCTL (Trip-zone Control)
// = = = = = = = = = = = = = = = = = = = = = = = = = = = = =
// TZA and TZB bits
#define         TZ_HIZ             0x0
#define         TZ_FORCE_HI        0x1
#define         TZ_FORCE_LO        0x2
#define         TZ_DISABLE         0x3
// ETSEL (Event-trigger Select)
// = = = = = = = = = = = = = = = = = = = = = = = = = = = = =
// INTSEL, SOCASEL, SOCBSEL bits
#define         ET_CTR_ZERO        0x1
```

Example 21-1. Constant Definitions Used in the ePWM Code Examples (continued)

```
#define         ET_CTR_PRD          0x2
#define         ET_CTRU_CMPA        0x4
#define         ET_CTRD_CMPA        0x5
#define         ET_CTRU_CMPB        0x6
#define         ET_CTRD_CMPB        0x7
// ETPS (Event-trigger Prescale)
// = = = = = = = = = = = = = = = = = = = = = = = = = = = = =
// INTPRD, SOCAPRD, SOCBPRD bits
#define         ET_DISABLE          0x0
#define         ET_1ST              0x1
#define         ET_2ND              0x2
#define         ET_3RD              0x3
```

21.2.2.2 Proper ePWM Interrupt Initialization Procedure

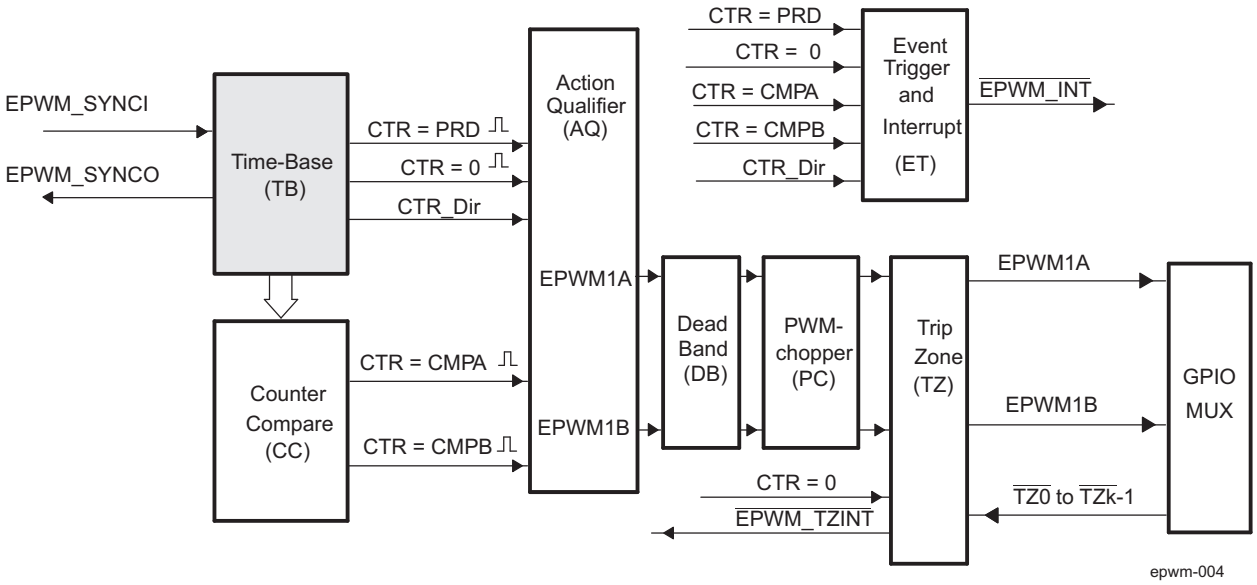
When the ePWM peripheral clock is enabled it may be possible that interrupt flags may be set due to spurious events due to the ePWM registers not being properly initialized. The proper procedure for initializing the ePWM peripheral is:

1. Disable global interrupts (CPU INTM flag)
2. Disable ePWM interrupts
3. Initialize peripheral registers
4. Clear any spurious ePWM flags
5. Enable ePWM interrupts
6. Enable global interrupts

21.2.2.3 ePWM Time-Base (TB) Submodule

The ePWM module has its own time-base submodule that determines all of the event timing for the ePWM module. Figure 21-6 illustrates the time-base module's place within the ePWM.

Figure 21-6. ePWM Time-Base Submodule Block Diagram



21.2.2.3.1 Purpose of the ePWM Time-Base Submodule

You can configure the time-base submodule for the following:

- Specify the ePWM1 time-base counter (TBCNT) frequency or period in the EPWM_TBCNT register to control how often events occur.
- Set the time-base counter to count-up, count-down, or count-up-and-down mode.
- Generate the following events:
  - TBCNT = PRD: Time-base counter (EPWM_TBCNT register) equal to the specified period in EPWM_TBPRD register (i.e. TBCNT = TBPRD) .
  - TBCNT = 0: Time-base counter equal to zero (TBCNT = 0000h).
- Configure the rate of the time-base clock; a prescaled version of the CPU system clock (SYSCLKOUT). This allows the time-base counter to increment/decrement at a slower rate.

21.2.2.3.2 Controlling and Monitoring the ePWM Time-Base Submodule

Table 21-20 lists the registers used to control and monitor the time-base submodule.

Table 21-20. ePWM Time-Base Submodule Registers

Acronym	Register Description	Address Offset	Shadowed
EPWM_TBCTL	Time-Base Control Register	0h	No
EPWM_TBSTS	Time-Base Status Register	2h	No
HRPWM_TBPHSHR	HRPWM extension Phase Register	4h	No
EPWM_TBPHS	Time-Base Phase Register	6h	No
EPWM_TBCNT	Time-Base Counter Register	8h	No
EPWM_TBPRD	Time-Base Period Register	Ah	Yes

Figure 21-7 shows the critical signals and registers of the time-base submodule. Table 21-21 provides descriptions of the key signals associated with the time-base submodule.

Figure 21-7. ePWM Time-Base Submodule Signals and Registers

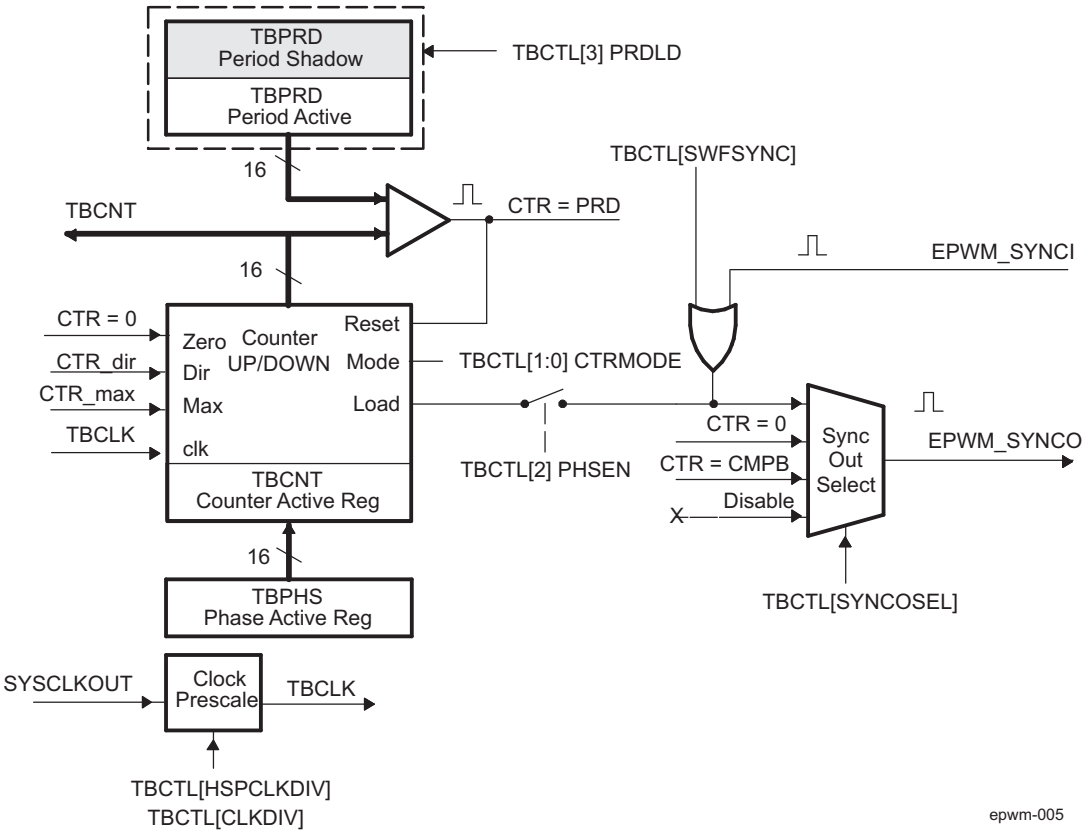


Table 21-21. ePWM Key Time-Base Signals

Signal	Description
EPWM_SYNCI	Time-base synchronization input. Input pulse used to synchronize the time-base counter with the counter of ePWM module. The ePWM peripheral can be configured to use or ignore this signal. This signal comes from a device pad. See Section 21.2.2.3.3.2 for information on the synchronization.

Table 21-21. ePWM Key Time-Base Signals (continued)

Signal	Description
EPWM_SYNCO	Time-base synchronization output. The ePWM module generates this signal from one of three event sources: 1. EPWM_SYNCI (Synchronization input pulse) 2. TBCNT = 0: The time-base counter (register EPWM_TBCNT) equal to zero (TBCNT = 0000h). 3. TBCNT = CMPB: The time-base counter (register EPWM_TBCNT) equal to the counter-compare B register - EPWM_CMPB (i.e. bitfield TBCNT = bitfield CMPB) .
TBCNT = PRD	Time-base counter equal to the specified period. This signal is generated whenever the counter value is equal to the active period register value. That is when TBCNT = TBPRD.
TBCNT = 0	Time-base counter equal to zero. This signal is generated whenever the counter value is zero. That is when TBCNT equals 0000h.
TBCNT = CMPB	Time-base counter equal to active counter-compare B register (TBCNT = CMPB). This event is generated by the counter-compare submodule and used by the synchronization out logic.
CTR_dir	Time-base counter direction. Indicates the current direction of the ePWM's time-base counter. This signal is high when the counter is increasing and low when it is decreasing.
CTR_max	Time-base counter equal max value. (TBCNT = FFFFh) Generated event when the EPWM_TBCNT value reaches its maximum value. This signal is only used only as a status bit.
TBCLK	Time-base clock. This is a prescaled version of the system clock - SYSCLKOUT(1) and is used by all submodules within the ePWM. This clock determines the rate at which time-base counter increments or decrements.

(1) The system clock - SYSCLKOUT is the ePWM functional clock derived from the PWMSS gateable interface and functional clock PWMSS_GICLK, described in Section 21.1.3.

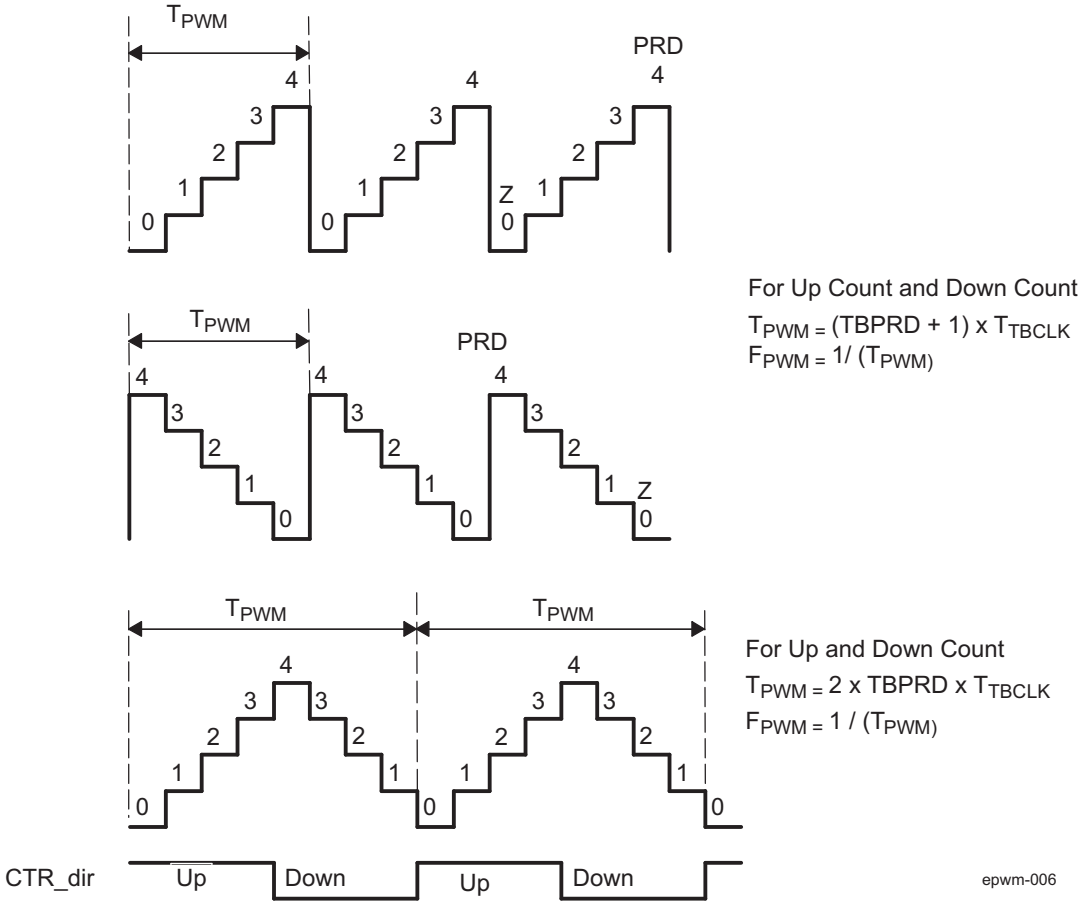
21.2.2.3.3 Calculating PWM Period and Frequency

The frequency of PWM events is controlled by the time-base period (EPWM_TBPRD) register and the mode of the time-base counter. Figure 21-8 shows the period (T_{pwm}) and frequency (F_{pwm}) relationships for the up-count, down-count, and up-down-count time-base counter modes when when the period is set to 4 (EPWM_TBPRD register bitfield TBPRD = 0x4). The time increment for each step is defined by the time-base clock (TBCLK) which is a prescaled version of the system clock (SYSCLKOUT).

The time-base counter has three modes of operation selected by the time-base control register (EPWM_TBCTL):

- **Up-Down-Count Mode:** In up-down-count mode, the time-base counter starts from zero and increments until the period (TBPRD) value is reached. When the period value is reached, the time-base counter then decrements until it reaches zero. At this point the counter repeats the pattern and begins to increment.
- **Up-Count Mode:** In this mode, the time-base counter starts from zero and increments until it reaches the value in the period register (TBPRD). When the period value is reached, the time-base counter resets to zero and begins to increment once again.
- **Down-Count Mode:** In down-count mode, the time-base counter starts from the period (TBPRD) value and decrements until it reaches zero. When it reaches zero, the time-base counter is reset to the period value and it begins to decrement once again.

Figure 21-8. ePWM Time-Base Frequency and Period



21.2.2.3.3.1 ePWM Time-Base Period Shadow Register

The time-base period register (EPWM_TBPRD) has a shadow register. Shadowing allows the register update to be synchronized with the hardware. The following definitions are used to describe all shadow registers in the ePWM module:

- **Active Register:** The active register controls the hardware and is responsible for actions that the hardware causes or invokes.
- **Shadow Register:** The shadow register buffers or provides a temporary holding location for the active register. It has no direct effect on any control hardware. At a strategic point in time the shadow register's content is transferred to the active register. This prevents corruption or spurious operation due to the register being asynchronously modified by software.

The memory address of the shadow period register is the same as the active register. Which register is written to or read from is determined by the EPWM_TBCTL[3] PRDLD bit. This bit enables and disables the EPWM_TBPRD shadow register as follows:

- **Time-Base Period Shadow Mode:** The EPWM_TBPRD shadow register is enabled when EPWM_TBCTL[3] PRDLD = 0. Reads from and writes to the EPWM_TBPRD memory address go to the shadow register. The shadow register contents are transferred to the active register (EPWM_TBPRD (Active) ← EPWM_TBPRD (shadow)) when the time-base counter (register (EPWM_TBCNT) equals zero (TBCNT = 0000h). By default the EPWM_TBPRD shadow register is enabled.
- **Time-Base Period Immediate Load Mode:** If immediate load mode is selected (EPWM_TBCTL[3] PRDLD = 1), then a read from or a write to the TBPRD memory address goes directly to the active

register.

21.2.2.3.3.2 ePWM Time-Base Counter Synchronization

The ePWM module can be configured to use or ignore the synchronization input. If the EPWM_TBCTL[2] PHSEN bit is set, then the time-base counter (TBCNT) of the ePWM module (register EPWM_TBCNT) will be automatically loaded with the phase register (EPWM_TBPHS) contents when one of the following conditions occur:

- **EPWM_SYNCI: Synchronization Input Pulse:** The value of the phase register is loaded into the counter register when an input synchronization pulse is detected (EPWM_TBPHS → EPWM_TBCNT). This operation occurs on the next valid time-base clock (TBCLK) edge.
- **Software Forced Synchronization Pulse:** Writing a 1 to the EPWM_TBCTL[6] SWFSYNC control bit invokes a software forced synchronization. This pulse is ORed with the synchronization input signal, and therefore has the same effect as a pulse on EPWM_SYNCI.

Clearing the EPWM_TBCTL[2] PHSEN bit configures the ePWM to ignore the synchronization input pulse.

21.2.2.3.4 ePWM Time-Base Counter Modes and Timing Waveforms

The time-base counter operates in one of four modes:

- Up-count mode which is asymmetrical.
- Down-count mode which is asymmetrical.
- Up-down-count which is symmetrical.
- Frozen where the time-base counter is held constant at the current value.

To illustrate the operation of the first three modes, Figure 21-9 to Figure 21-12 show when events are generated and how the time-base responds to an EPWM_SYNCI signal.

Figure 21-9. ePWM Time-Base Up-Count Mode Waveforms

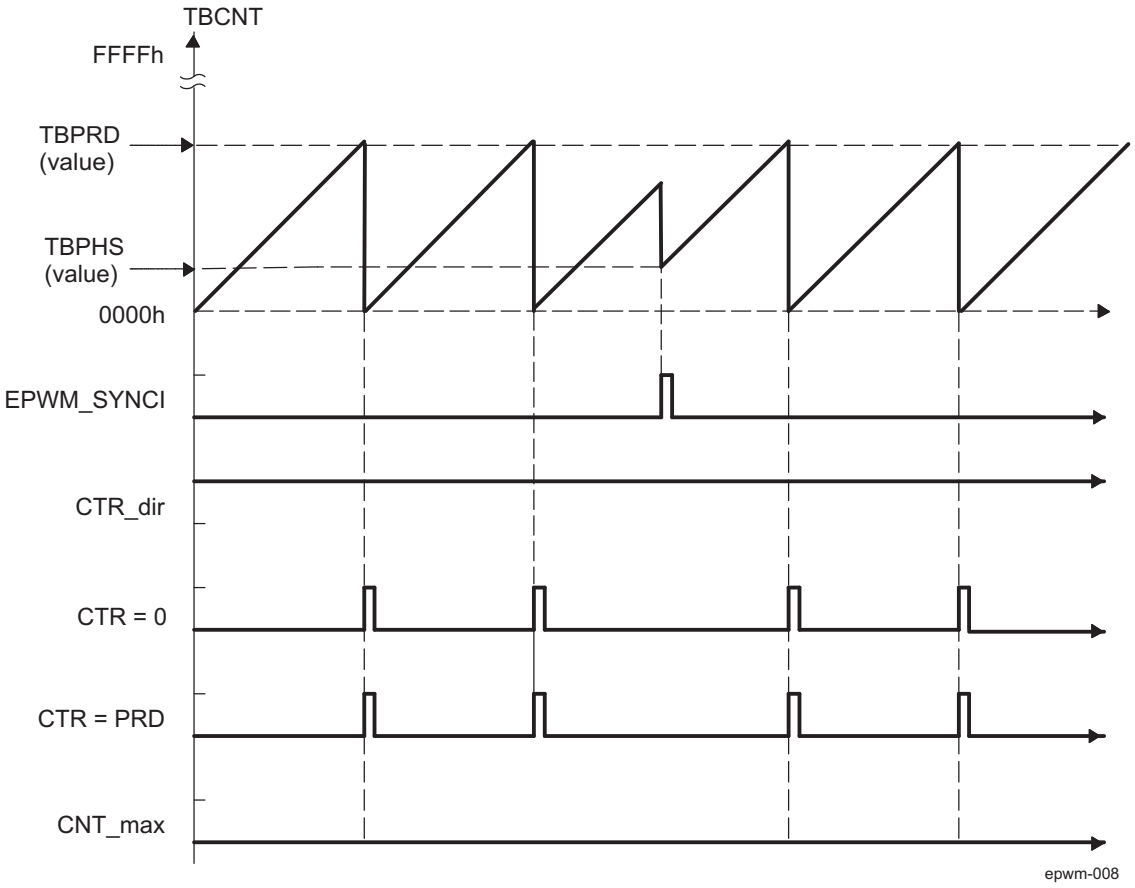


Figure 21-10. ePWM Time-Base Down-Count Mode Waveforms

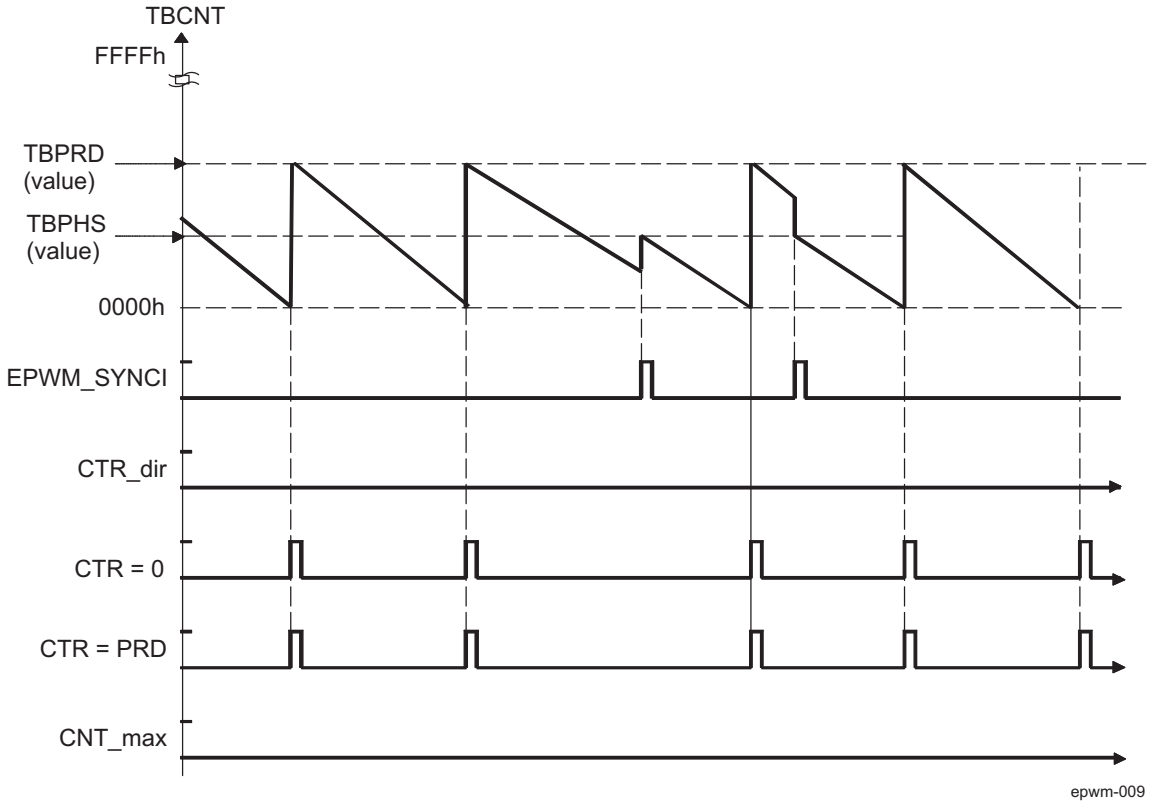


Figure 21-11. ePWM Time-Base Up-Down-Count Waveforms, EPWM_TBCTL[13] PHSDIR = 0 Count Down on Synchronization Event

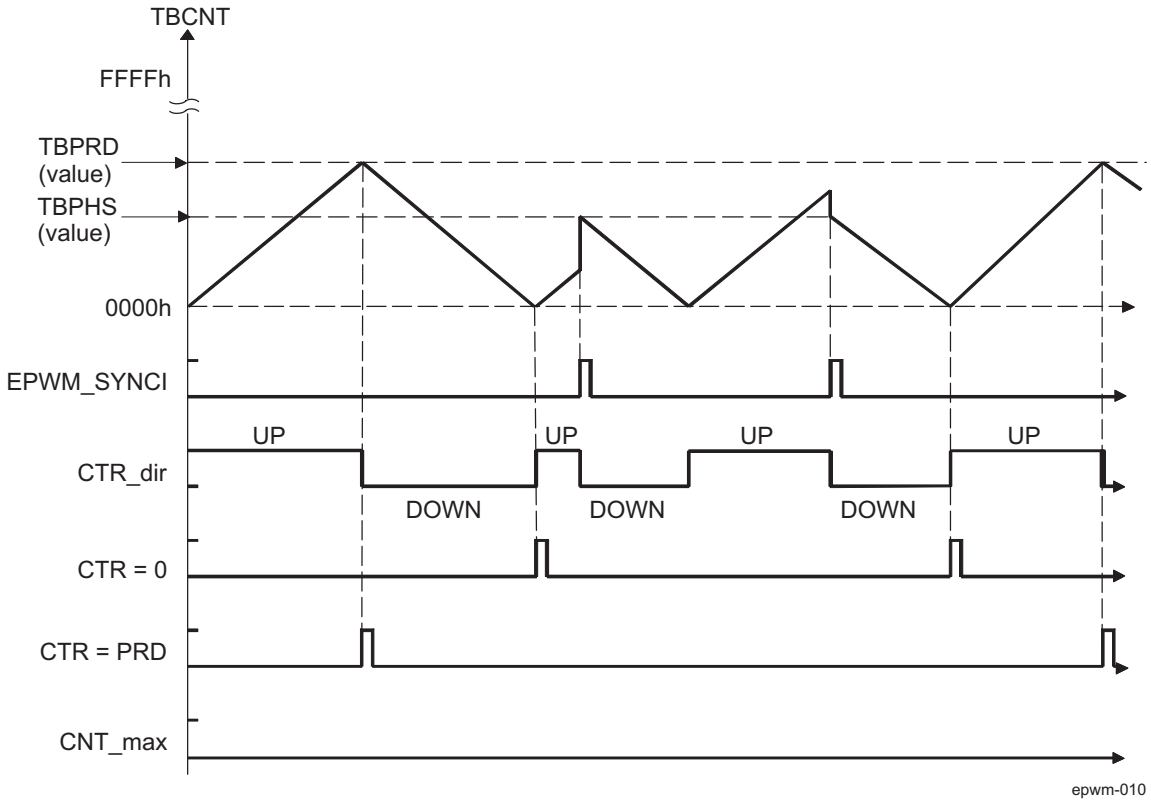
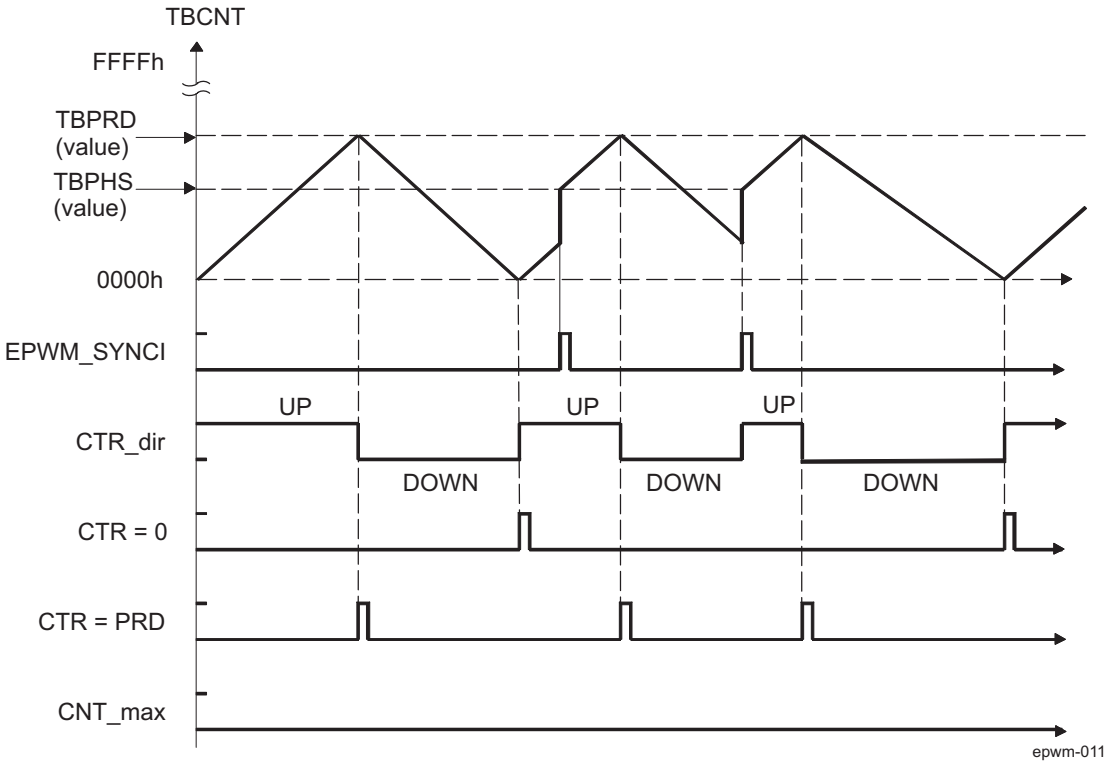


Figure 21-12. ePWM Time-Base Up-Down Count Waveforms, EPWM_TBCTL[13] PHSDIR = 1 Count Up on Synchronization Event



21.2.2.4 ePWM Counter-Compare (CC) Submodule

Figure 21-13 illustrates the counter-compare submodule within the ePWM. Figure 21-14 shows the basic structure of the counter-compare submodule.

Figure 21-13. ePWM Counter-Compare Submodule

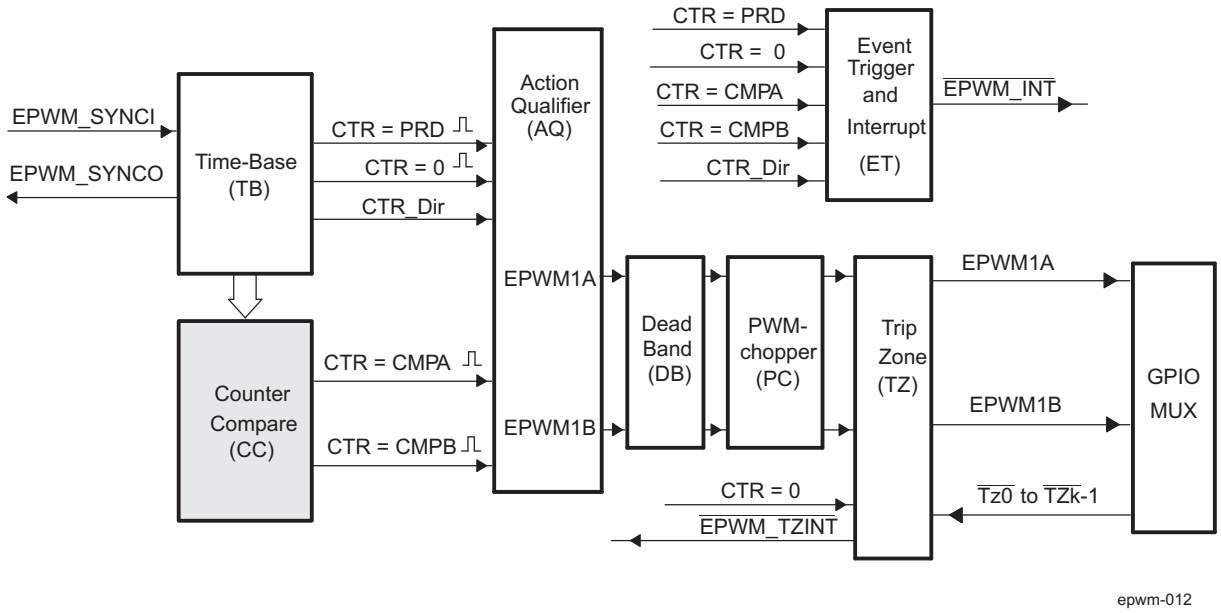
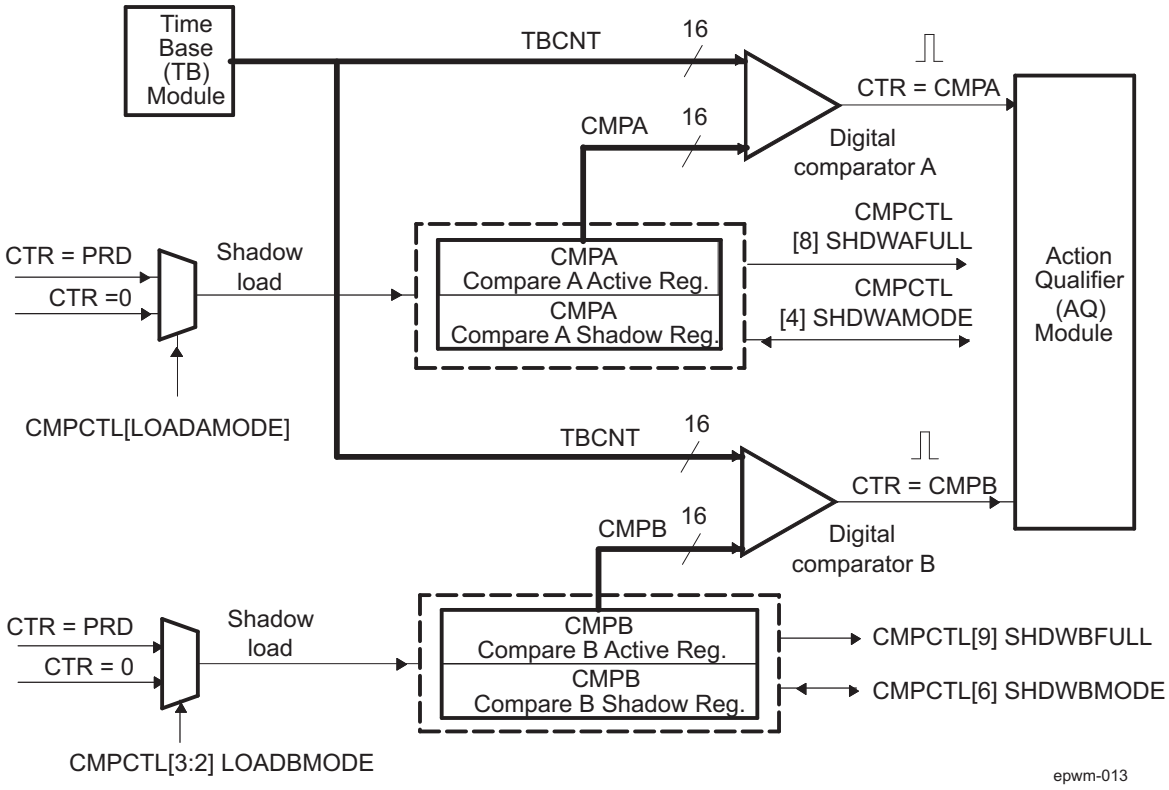


Figure 21-14. ePWM Counter-Compare Submodule Signals and Registers



21.2.2.4.1 Purpose of the ePWM Counter-Compare Submodule

The counter-compare submodule takes as input the time-base counter value. This value is continuously compared to the counter-compare A (EPWM_CMPA) and counter-compare B (EPWM_CMPB) registers. When the time-base counter is equal to one of the compare registers, the counter-compare unit generates an appropriate event.

The counter-compare submodule:

- Generates events based on programmable time stamps using the EPWM_CMPA and EPWM_CMPB registers
  - TBCNT = CMPA: Time-base counter equals counter-compare A register (TBCNT = CMPA).
  - TBCNT = CMPB: Time-base counter equals counter-compare B register (TBCNT = CMPB)
- Controls the PWM duty cycle if the action-qualifier submodule is configured appropriately
- Shadows new compare values to prevent corruption or glitches during the active PWM cycle

21.2.2.4.2 Controlling and Monitoring the ePWM Counter-Compare Submodule

Table 21-22 lists the registers used to control and monitor the counter-compare submodule. Table 21-23 lists the key signals associated with the counter-compare submodule.

Table 21-22. ePWM Counter-Compare Submodule Registers

Acronym	Register Description	Address Offset	Shadowed
EPWM_CMPCTL	Counter-Compare Control Register.	Eh	No
HRPWM_CMPAHR	HRPWM Counter-Compare A Extension Register	10h	Yes
EPWM_CMPA	Counter-Compare A Register	12h	Yes
EPWM_CMPB	Counter-Compare B Register	14h	Yes

Table 21-23. ePWM Counter-Compare Submodule Key Signals

Signal	Description of Event	Register Bitfields Compared
TBCNT = CMPA	Time-base counter equal to the active counter-compare A value	TBCNT = CMPA
TBCNT = CMPB	Time-base counter equal to the active counter-compare B value	TBCNT = CMPB
TBCNT = PRD	Time-base counter equal to the active period. Used to load active counter-compare A and B registers from the shadow register	TBCNT = TBPRD
TBCNT = 0	Time-base counter equal to zero. Used to load active counter-compare A and B registers from the shadow register	TBCNT = 0000h

21.2.2.4.3 Operational Highlights for the ePWM Counter-Compare Submodule

The counter-compare submodule is responsible for generating two independent compare events based on two compare registers:

1. TBCNT = CMPA: Time-base counter equal to counter-compare A register (EPWM_TBCNT = EPWM_CMPA).
2. TBCNT = CMPB: Time-base counter equal to counter-compare B register (TBCNT = CMPB).

For up-count or down-count mode, each event occurs only once per cycle. For up-down-count mode each event occurs twice per cycle, if the compare value is between 0000h and TBPRD; and occurs once per cycle, if the compare value is equal to 0000h or equal to TBPRD. These events are fed into the action-qualifier submodule where they are qualified by the counter direction and converted into actions if enabled. Refer to Section 21.2.2.5.1 for more details.

The counter-compare registers EPWM_CMPA and EPWM_CMPB each have an associated shadow register. Shadowing provides a way to keep updates to the registers synchronized with the hardware. When shadowing is used, updates to the active registers only occurs at strategic points. This prevents corruption or spurious operation due to the register being asynchronously modified by software. The memory address of the active register and the shadow register is identical. Which register is written to or read from is determined by the EPWM_CMPCTL[4] SHDWAMODE and EPWM_CMPCTL[6] SHDWBMODE bits. These bits enable and disable the EPWM_CMPA shadow register and EPWM_CMPB shadow register respectively. The behavior of the two load modes is described below:

- **Shadow Mode:** The shadow mode for the EPWM_CMPA is enabled by clearing the EPWM_CMPCTL[4] SHDWAMODE bit and the shadow register for CMPB is enabled by clearing the EPWM_CMPCTL[6] SHDWBMODE bit. Shadow mode is enabled by default for both EPWM_CMPA and EPWM_CMPB.

  If the shadow register is enabled then the content of the shadow register is transferred to the active register on one of the following events:

  – TBCNT = PRD: Time-base counter equal to the period (TBCNT = TBPRD).
  – TBCNT = 0: Time-base counter equal to zero (TBCNT = 0000h)
  – Both TBCNT = PRD and TBCNT = 0

  Which of these three events is specified by the EPWM_CMPCTL[1:0] LOADAMODE and EPWM_CMPCTL[3:2] LOADBMODE register bits. Only the active register contents are used by the counter-compare submodule to generate events to be sent to the action-qualifier.
- **Immediate Load Mode:** If immediate load mode is selected (EPWM_TBCTL[4] SHDWAMODE = 1 or EPWM_TBCTL[6] SHDWBMODE = 1), then a read from or a write to the register will go directly to the active register.

21.2.2.4.4 ePWM Count Mode Timing Waveforms

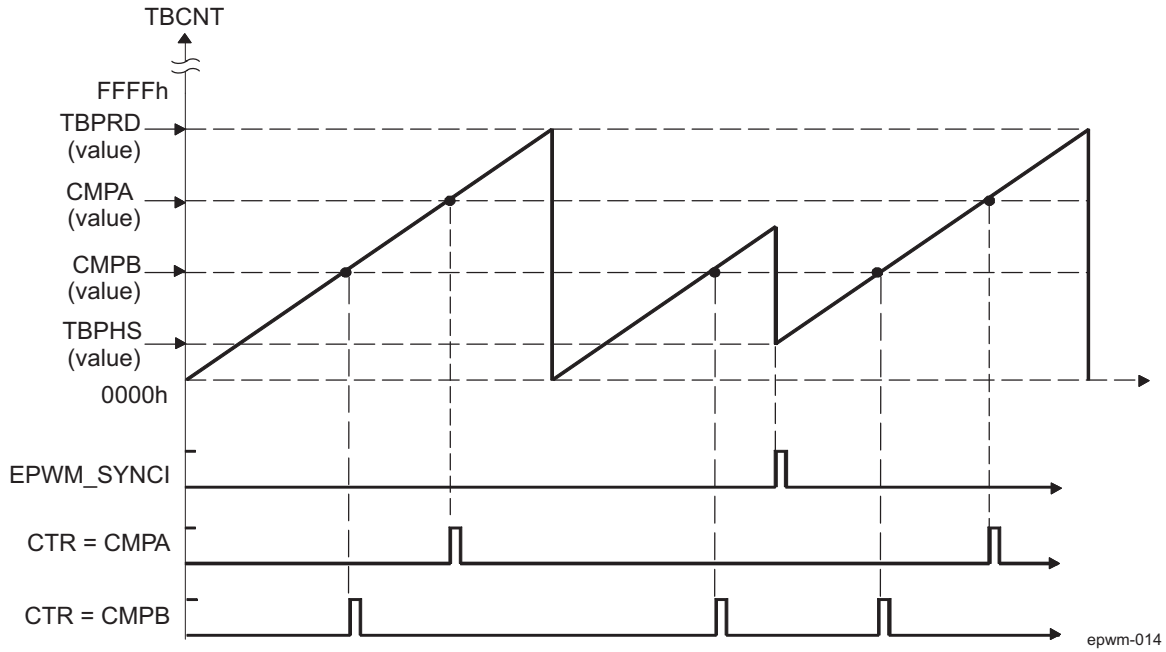
The counter-compare module can generate compare events in all three count modes:

- Up-count mode: used to generate an asymmetrical PWM waveform.
- Down-count mode: used to generate an asymmetrical PWM waveform.

- Up-down-count mode: used to generate a symmetrical PWM waveform.

To best illustrate the operation of the first three modes, the timing diagrams in Figure 21-15 to Figure 21-18 show when events are generated and how the EPWM_SYNCI signal interacts.

Figure 21-15. ePWM Counter-Compare Event Waveforms in Up-Count Mode



NOTE: An EPWM_SYNCI external synchronization event can cause a discontinuity in the TBCNT count sequence. This can lead to a compare event being skipped. This skipping is considered normal operation and must be taken into account.

Figure 21-16. ePWM Counter-Compare Events in Down-Count Mode

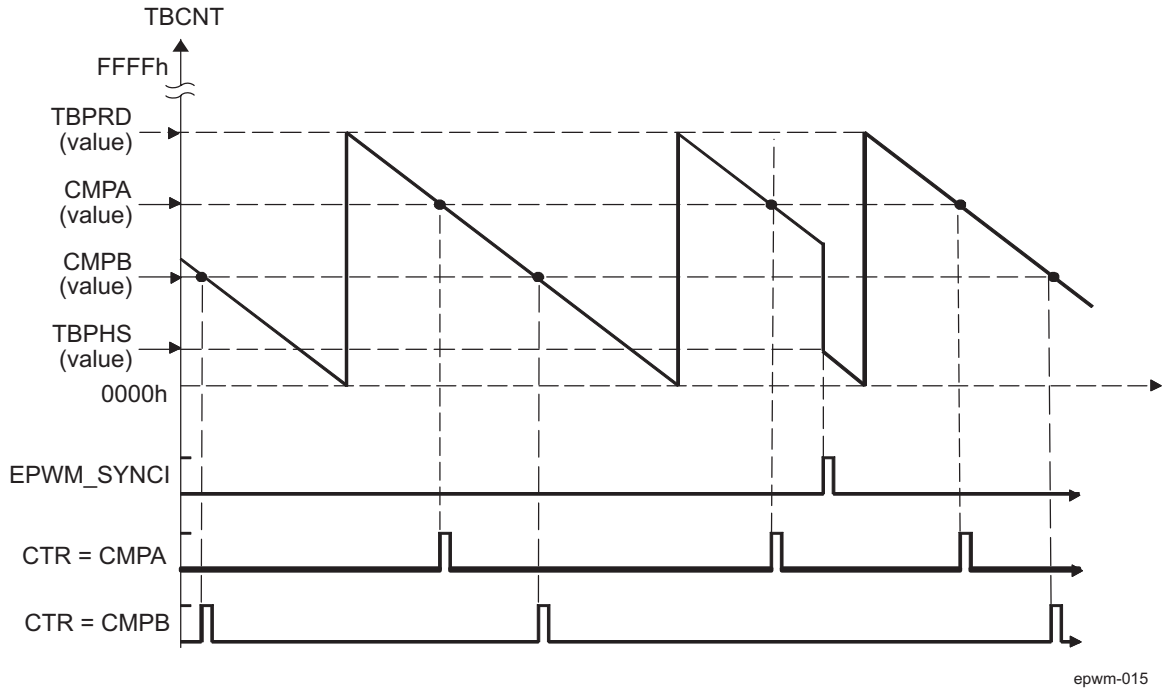


Figure 21-17. ePWM Counter-Compare Events in Up-Down-Count Mode, EPWM_TBCTL[13] PHSDIR = 0 Count Down on Synchronization Event

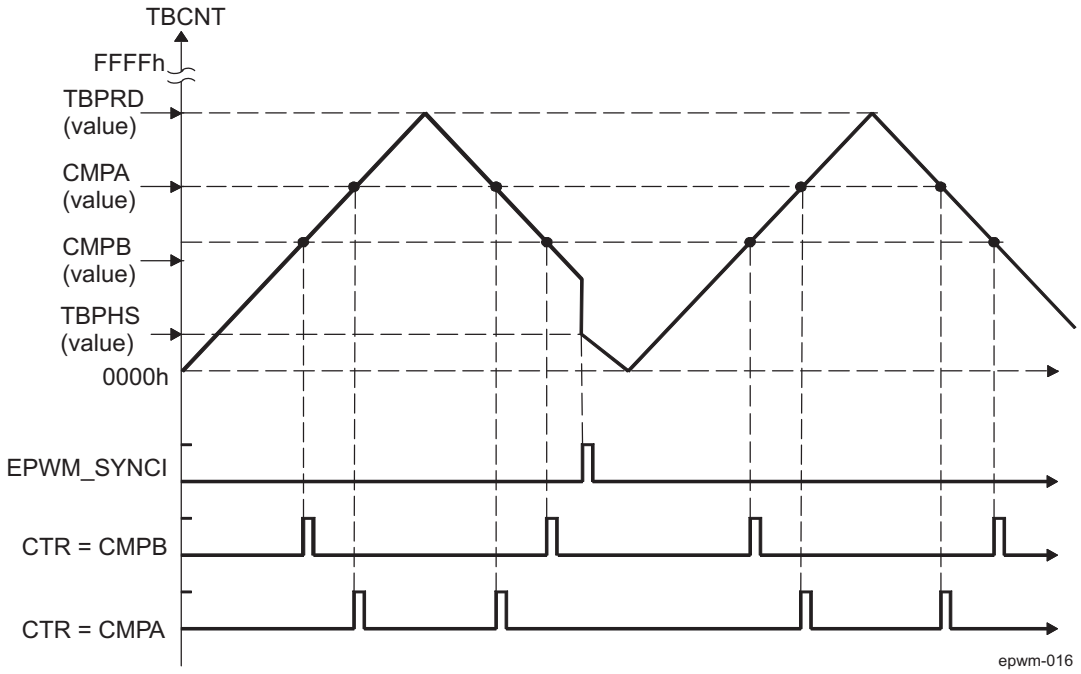
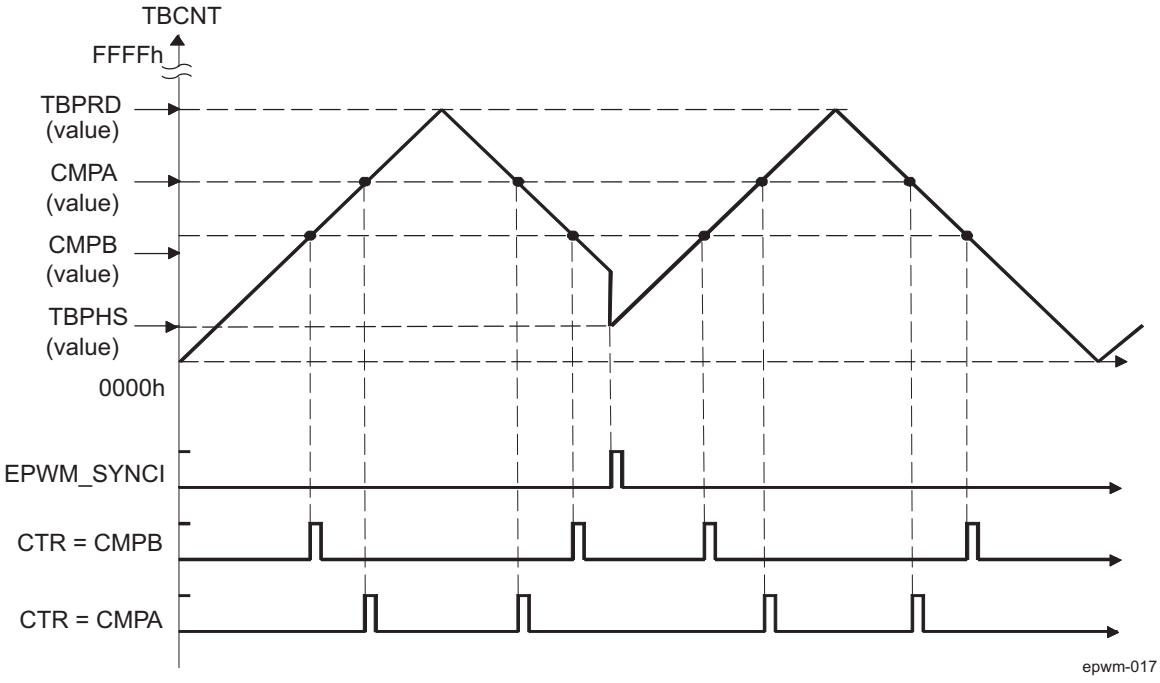


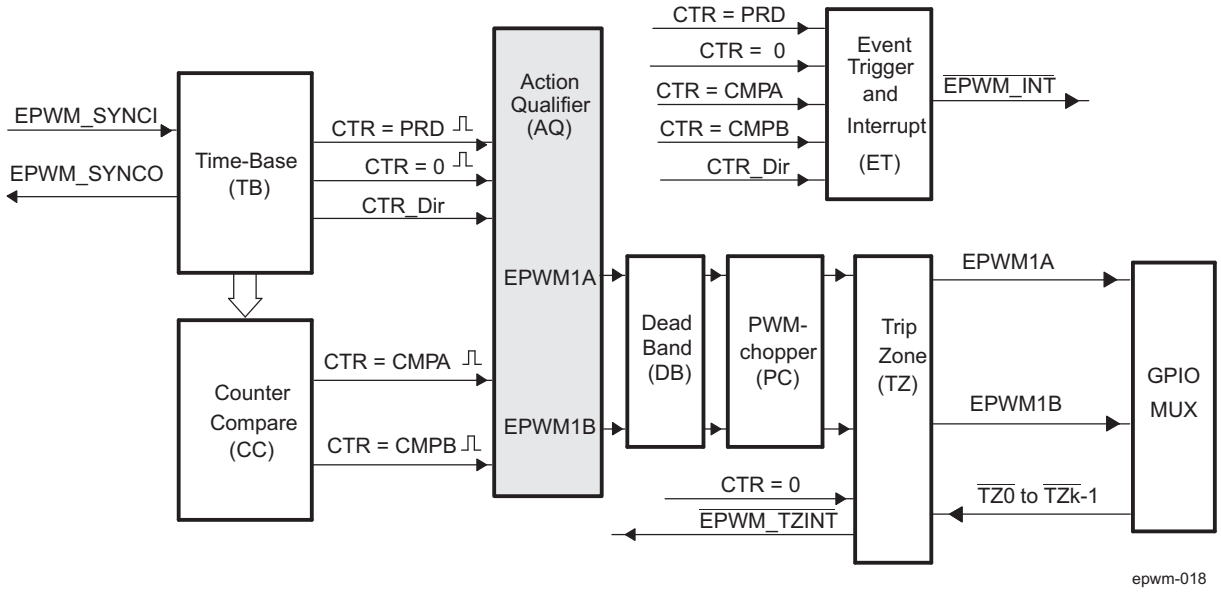
Figure 21-18. ePWM Counter-Compare Events in Up-Down-Count Mode, EPWM_TBCTL[13] PHSDIR = 1 Count Up on Synchronization Event



21.2.2.5 ePWM Action-Qualifier (AQ) Submodule

Figure 21-19 shows the action-qualifier (AQ) submodule (see shaded block) in the ePWM system. The action-qualifier submodule has the most important role in waveform construction and PWM generation. It decides which events are converted into various action types, thereby producing the required switched waveforms at the EPWM1A and EPWM1B outputs.

Figure 21-19. ePWM Action-Qualifier Submodule



21.2.2.5.1 Purpose of the ePWM Action-Qualifier Submodule

The action-qualifier submodule is responsible for the following:

- Qualifying and generating actions (set, clear, toggle) based on the following events:
  - TBCNT = PRD: Time-base counter equal to the period (TBCNT = TBPRD)
  - TBCNT = 0: Time-base counter equal to zero (TBCNT = 0000h)
  - TBCNT = CMPA: Time-base counter equal to the counter-compare A register (TBCNT = CMPA)
  - TBCNT = CMPB: Time-base counter equal to the counter-compare B register (TBCNT = CMPB)
- Managing priority when these events occur concurrently
- Providing independent control of events when the time-base counter is increasing and when it is decreasing.

21.2.2.5.2 Controlling and Monitoring the ePWM Action-Qualifier Submodule

Table 21-24 lists the registers used to control and monitor the action-qualifier submodule.

Table 21-24. Action-Qualifier Submodule Registers

Acronym	Register Description	Address Offset	Shadowed
EPWM_AQCTLA	Action-Qualifier Control Register For Output A (EPWM1A)	16h	No
EPWM_AQCTLB	Action-Qualifier Control Register For Output B (EPWM1B)	18h	No
EPWM_AQSFRC	Action-Qualifier Software Force Register	1Ah	No
EPWM_AQCSFRC	Action-Qualifier Continuous Software Force	1Ch	Yes

The action-qualifier submodule is based on event-driven logic. It can be thought of as a programmable cross switch with events at the input and actions at the output, all of which are software controlled via the set of registers shown in Figure 21-20. The possible input events are summarized again in Table 21-25.

Figure 21-20. ePWM Action-Qualifier Submodule Inputs and Outputs

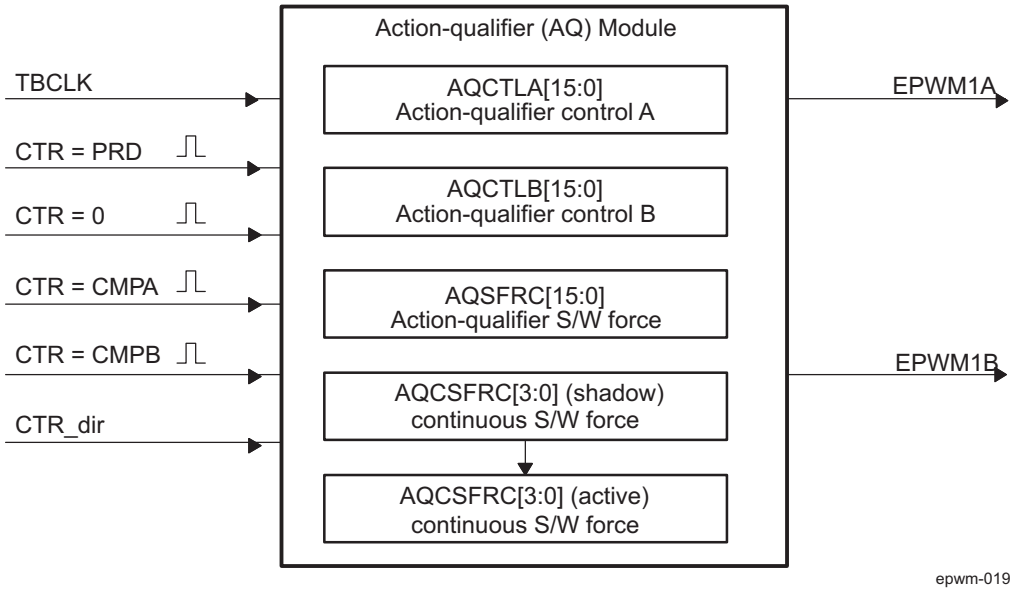


Table 21-25. ePWM Action-Qualifier Submodule Possible Input Events

Signal	Description	Register Bitfield Compared
TBCNT = PRD	Time-base counter equal to the period value	TBCNT = TBPRD
TBCNT = 0	Time-base counter equal to zero	TBCNT = 0000h
TBCNT = CMPA	Time-base counter equal to the counter-compare A	TBCNT = CMPA
TBCNT = CMPB	Time-base counter equal to the counter-compare B	TBCNT = CMPB
Software forced event	Asynchronous event initiated by software	

The software forced action is a useful asynchronous event. This control is handled by registers EPWM_AQSFRC and EPWM_AQCSFRC.

The action-qualifier submodule controls how the two outputs EPWM1A and EPWM1B behave when a particular event occurs. The event inputs to the action-qualifier submodule are further qualified by the counter direction (up or down). This allows for independent action on outputs on both the count-up and count-down phases.

The possible actions imposed on outputs EPWM1A and EPWM1B are:

- **Set High:** Set output EPWM1A or EPWM1B to a high level.
- **Clear Low:** Set output EPWM1A or EPWM1B to a low level.
- **Toggle:** If EPWM1A or EPWM1B is currently pulled high, then pull the output low. If EPWM1A or EPWM1B is currently pulled low, then pull the output high.
- **Do Nothing:** Keep outputs EPWM1A and EPWM1B at same level as currently set. Although the "Do Nothing" option prevents an event from causing an action on the EPWM1A and EPWM1B outputs, this event can still trigger interrupts. See the event-trigger submodule description in Section 21.2.2.9 for details.

Actions are specified independently for either output (EPWM1A or EPWM1B). Any or all events can be configured to generate actions on a given output. For example, both TBCNT = CMPA and TBCNT = CMPB can operate on output EPWM1A. All qualifier actions are configured via the control registers found at the end of this section.

For clarity, the drawings in this chapter use a set of symbolic actions. These symbols are summarized in Figure 21-21. Each symbol represents an action as a marker in time. Some actions are fixed in time (zero and period) while the CMPA and CMPB actions are moveable and their time positions are programmed via the counter-compare A and B registers, respectively. To turn off or disable an action, use the "Do Nothing option"; it is the default at reset.

Figure 21-21. Possible Action-Qualifier Actions for EPWM1A and EPWM1B Outputs

S/W force	TB Counter equals:				Actions
	Zero	Comp A	Comp B	Period	
SW ×	Z ×	CA ×	CB ×	P ×	Do Nothing
SW ↓	Z ↓	CA ↓	CB ↓	P ↓	Clear Low
SW ↑	Z ↑	CA ↑	CB ↑	P ↑	Set High
SW T	Z T	CA T	CB T	P T	Toggle

epwm-020

21.2.2.5.3 ePWM Action-Qualifier Event Priority

It is possible for the ePWM action qualifier to receive more than one event at the same time. In this case events are assigned a priority by the hardware. The general rule is events occurring later in time have a higher priority and software forced events always have the highest priority. The event priority levels for up-down-count mode are shown in Table 21-26. A priority level of 1 is the highest priority and level 7 is the lowest. The priority changes slightly depending on the direction of TBCNT.

Table 21-26. ePWM Action-Qualifier Event Priority for Up-Down-Count Mode

Priority Level	Event if TBCNT is Incrementing TBCNT = 0 up to TBCNT = TBPRD	Event if TBCNT is Decrementing TBCNT = TBPRD down to TBCNT = 1
1 (Highest)	Software forced event	Software forced event
2	Counter equals CMPB on up-count (CBU)	Counter equals CMPB on down-count (CBD)
3	Counter equals CMPA on up-count (CAU)	Counter equals CMPA on down-count (CAD)
4	Counter equals zero	Counter equals period (TBPRD in EPWM_TBPRD active register)
5	Counter equals CMPB on down-count (CBD) (1)	Counter equals CMPB on up-count (CBU) (1)
6 (Lowest)	Counter equals CMPA on down-count (CAD) (1)	Counter equals CMPA on up-count (CBU) (1)

(1) To maintain symmetry for up-down-count mode, both up-events (CAU/CBU) and down-events (CAD/CBD) can be generated for TBPRD. Otherwise, up-events can occur only when the counter is incrementing and down-events can occur only when the counter is decrementing.

Table 21-27 shows the action-qualifier priority for up-count mode. In this case, the counter direction is always defined as up and thus down-count events will never be taken.

Table 21-27. ePWM Action-Qualifier Event Priority for Up-Count Mode

Priority Level	Event
1 (Highest)	Software forced event
2	Counter equal to period (TBPRD)
3	Counter equal to CMPB on up-count (CBU)
4	Counter equal to CMPA on up-count (CAU)
5 (Lowest)	Counter equal to Zero

Table 21-28 shows the action-qualifier priority for down-count mode. In this case, the counter direction is always defined as down and thus up-count events will never be taken.

Table 21-28. ePWM Action-Qualifier Event Priority for Down-Count Mode

Priority Level	Event
1 (Highest)	Software forced event
2	Counter equal to Zero
3	Counter equal to CMPB on down-count (CBD)
4	Counter equal to CMPA on down-count (CAD)
5 (Lowest)	Counter equal to period (TBPRD)

It is possible to set the compare value greater than the period. In this case the action will take place as shown in Table 21-29.

Table 21-29. Behavior if CMPA/CMPB is Greater than the Period

Counter Mode	Compare on Up-Count Event CAU/CBU	Compare on Down-Count Event CAU/CBU
Up-Count Mode	If CMPA/CMPB ≤ TBPRD period, then the event occurs on a compare match (TBCNT = CMPA or CMPB). If CMPA/CMPB > TBPRD, then the event will not occur.	Never occurs.
Down-Count Mode	Never occurs.	If CMPA/CMPB < TBPRD, the event will occur on a compare match (TBCNT = CMPA or CMPB). If CMPA/CMPB ≥ TBPRD, the event will occur on a period match (TBCNT = TBPRD).
Up-Down-Count Mode	If CMPA/CMPB < TBPRD and the counter is incrementing, the event occurs on a compare match (TBCNT = CMPA or CMPB). If CMPA/CMPB is ≥ TBPRD, the event will occur on a period match (TBCNT = TBPRD).	If CMPA/CMPB < TBPRD and the counter is decrementing, the event occurs on a compare match (TBCNT = CMPA or CMPB). If CMPA/CMPB ≥ TBPRD, the event occurs on a period match (TBCNT = TBPRD).

21.2.2.5.4 Waveforms for Common ePWM Configurations

NOTE: The waveforms in this chapter show the ePWMs behavior for a static compare register value. In a running system, the active compare registers (EPWM_CMPA and EPWM_CMPB) are typically updated from their respective shadow registers once every period. The user specifies when the update will take place; either when the time-base counter reaches zero or when the time-base counter reaches period. There are some cases when the action based on the new value can be delayed by one period or the action based on the old value can take effect for an extra period. Some PWM configurations avoid this situation. These include, but are not limited to, the following:

Use up-down-count mode to generate a symmetric PWM:

- If you load EPWM_CMPA / EPWM_CMPB on zero, then use EPWM_CMPA / EPWM_CMPB values greater than or equal to 1.
- If you load EPWM_CMPA / EPWM_CMPB on period, then use EPWM_CMPA / EPWM_CMPB values less than or equal to TBPRD - 1.

  This means there will always be a pulse of at least one TBCLK cycle in a PWM period which, when very short, tend to be ignored by the system.

Use up-down-count mode to generate an asymmetric PWM:

- To achieve 50%-0% asymmetric PWM use the following configuration: Load EPWM_CMPA / EPWM_CMPB on period and use the period action to clear the PWM and a compare-up action to set the PWM. Modulate the compare value from 0 to TBPRD to achieve 50%-0% PWM duty.

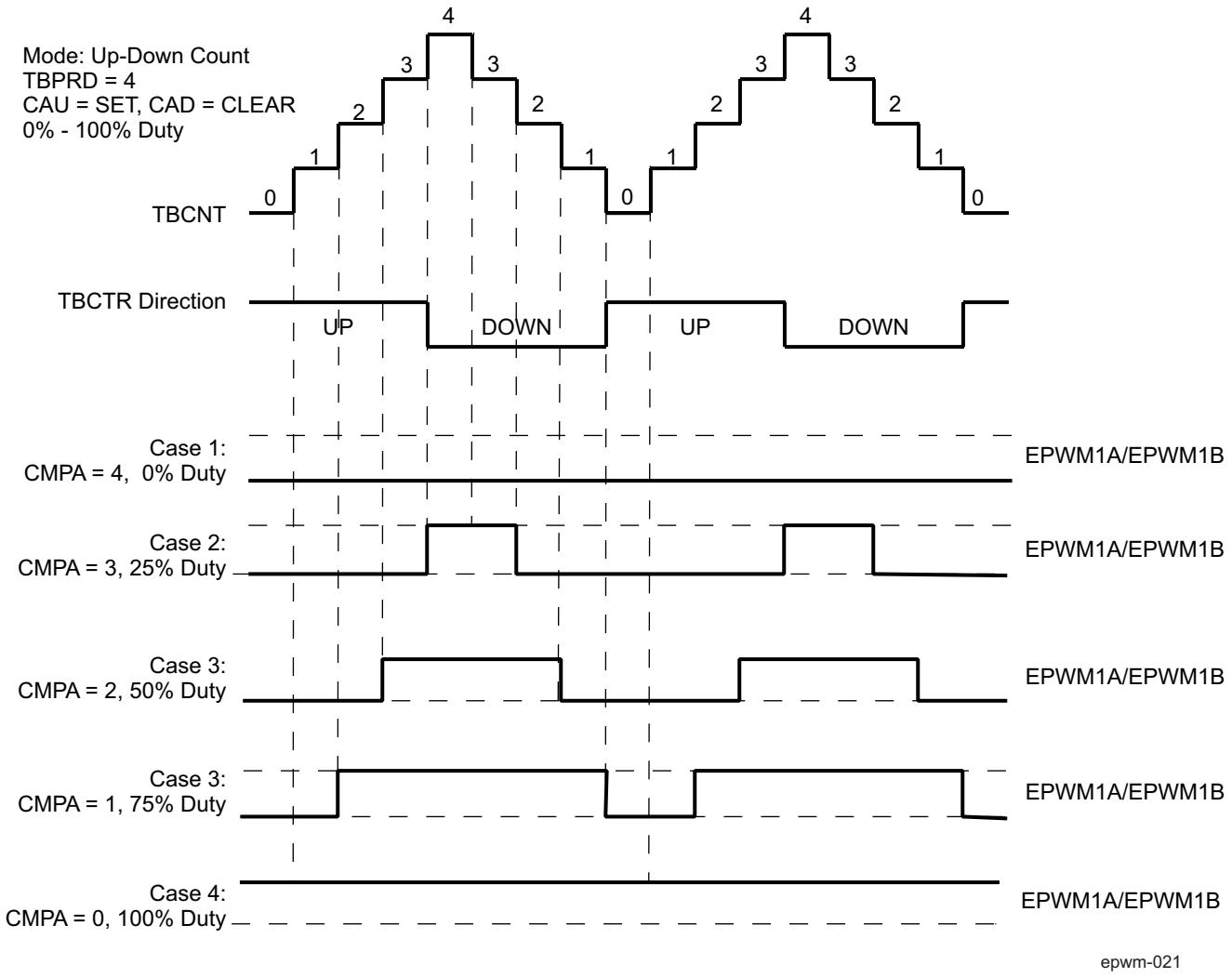
When using up-count mode to generate an asymmetric PWM:

- To achieve 0-100% asymmetric PWM use the following configuration: Load EPWM_CMPA / EPWM_CMPB on TBPRD. Use the Zero action to set the PWM and a compare-up action to clear the PWM. Modulate the compare value from 0 to TBPRD+1 to achieve 0-100% PWM duty.

Figure 21-22 shows how a symmetric PWM waveform can be generated using the up-down-count mode of the TBCNT. In this mode 0%-100% DC modulation is achieved by using equal compare matches on the up count and down count portions of the waveform. In the example shown, CMPA is used to make the comparison. When the counter is incrementing the CMPA match will pull the PWM output high. Likewise, when the counter is decrementing the compare match will pull the PWM signal low. When CMPA = 0, the PWM signal is low for the entire period giving the 0% duty waveform. When EPWM_CMPA = EPWM_TBPRD, the PWM signal is high achieving 100% duty.

When using this configuration in practice, if you load CMPA/CMPB on zero, then use CMPA/CMPB values greater than or equal to 1. If you load CMPA/CMPB on period, then use CMPA/CMPB values less than or equal to TBPRD-1. This means there will always be a pulse of at least one TBCLK cycle in a PWM period which, when very short, tend to be ignored by the system.

Figure 21-22. ePWM Up-Down-Count Mode Symmetrical Waveform

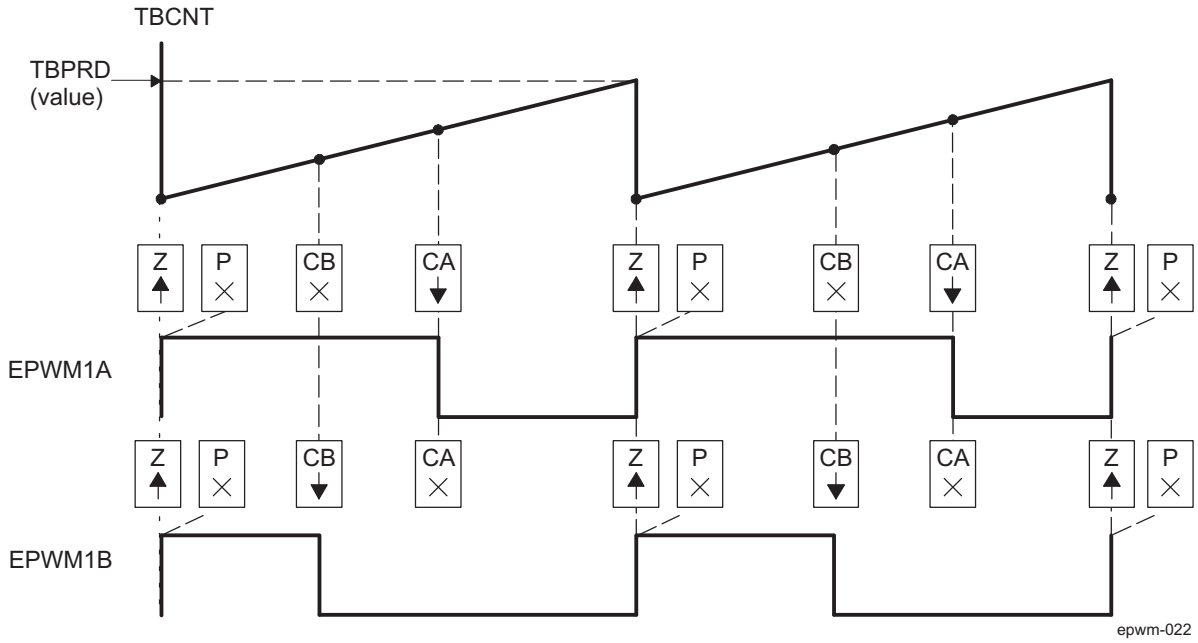


The PWM waveforms in Figure 21-23 through Figure 21-28 show some common action-qualifier configurations. Some conventions used in the figures are as follows:

- TBPRD, CMPA, and CMPB refer to the value written in their respective registers (EPWM_TBPRD, EPWM_CMPA, and EPWM_CMPB). The active register, not the shadow register, is used by the hardware.
- CMPx, refers to either CMPA or CMPB.
- EPWM1A and EPWM1B refer to the output signals from ePWM1
- Up-Down means Count-up-and-down mode, Up means up-count mode and Dwn means down-count mode
- Sym = Symmetric, Asym = Asymmetric

Table 21-30 and Table 21-31 contains initialization and runtime register configurations for the waveforms in Figure 21-23.

Figure 21-23. Up, Single Edge Asymmetric Waveform, With Independent Modulation on EPWM1A and EPWM1B—Active High



(1) PWM period = (TBPRD + 1) × T_{TBCLK}

(2) Duty modulation for EPWM1A is set by CMPA, and is active high (that is, high time duty proportional to CMPA).

(3) Duty modulation for EPWM1B is set by CMPB and is active high (that is, high time duty proportional to CMPB).

(4) The "Do Nothing" actions (X) are shown for completeness, but will not be shown on subsequent diagrams.

(5) Actions at zero and period, although appearing to occur concurrently, are actually separated by one TBCLK period. TBCNT wraps from period to 0000h.

Table 21-30. EPWM Initialization for Figure 21-23

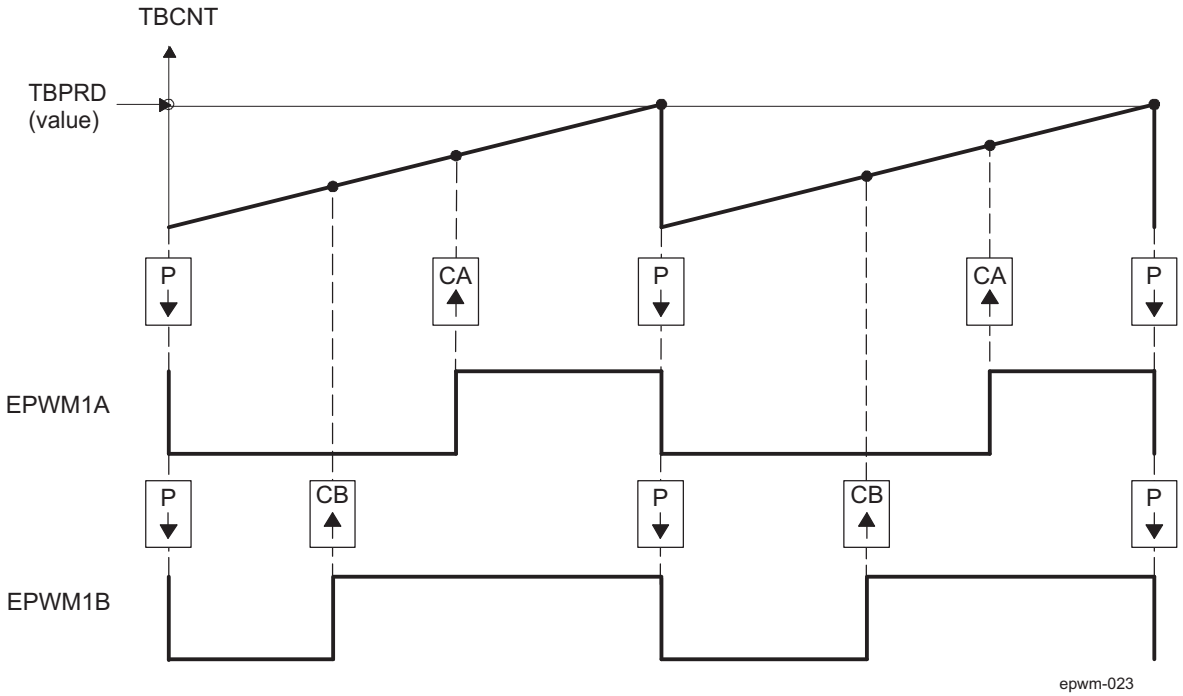
Register	Bitfield	Value	Comments
EPWM_TBPRD	TBPRD	600 (258h)	Period = 601 TBCLK counts
EPWM_TBPHS	TBPHS	0	Clear Phase Register to 0
EPWM_TBCNT	TBCNT	0	Clear TB counter
EPWM_TBCTL	CTRMODE	TB_UP	
	PHSEN	TB_DISABLE	Phase loading disabled
	PRDLD	TB_SHADOW	
	SYNCOSEL	TB_SYNC_DISABLE	
	HSPCLKDIV	TB_DIV1	TBCLK = SYSCLKOUT
	CLKDIV	TB_DIV1	
EPWM_CMPA	CMPA	350 (15Eh)	Compare A = 350 TBCLK counts
EPWM_CMPB	CMPB	200 (C8h)	Compare B = 200 TBCLK counts
EPWM_CMPCTL	SHDWAMODE	CC_SHADOW	
	SHDWBMODE	CC_SHADOW	
	LOADAMODE	CC_CTR_ZERO	Load on TBCNT = 0
	LOADBMODE	CC_CTR_ZERO	Load on TBCNT = 0
EPWM_AQCTLA	ZRO	AQ_SET	
	CAU	AQ_CLEAR	
EPWM_AQCTLB	ZRO	AQ_SET	
	CBU	AQ_CLEAR	

Table 21-31. EPWM Run Time Changes for Figure 21-23

Register	Bitfield	Value	Comments
EPWM_CMPA	CMPA	Duty1A	Adjust duty for output EPWM1A
EPWM_CMPB	CMPB	Duty1B	Adjust duty for output EPWM1B

Table 21-32 and Table 21-33 contains initialization and runtime register configurations for the waveforms in Figure 21-24.

Figure 21-24. Up, Single Edge Asymmetric Waveform With Independent Modulation on EPWM1A and EPWM1B—Active Low



(1) PWM period = (TBPRD + 1) × T_{TBCLK}

(2) Duty modulation for EPWM1A is set by CMPA, and is active low (that is, the low time duty is proportional to CMPA).

(3) Duty modulation for EPWM1B is set by CMPB and is active low (that is, the low time duty is proportional to CMPB).

(4) The Do Nothing actions (X) are shown for completeness here, but will not be shown on subsequent diagrams.

(5) Actions at zero and period, although appearing to occur concurrently, are actually separated by one TBCLK period. TBCNT wraps from period to 0000h.

Table 21-32. EPWM Initialization for Figure 21-24

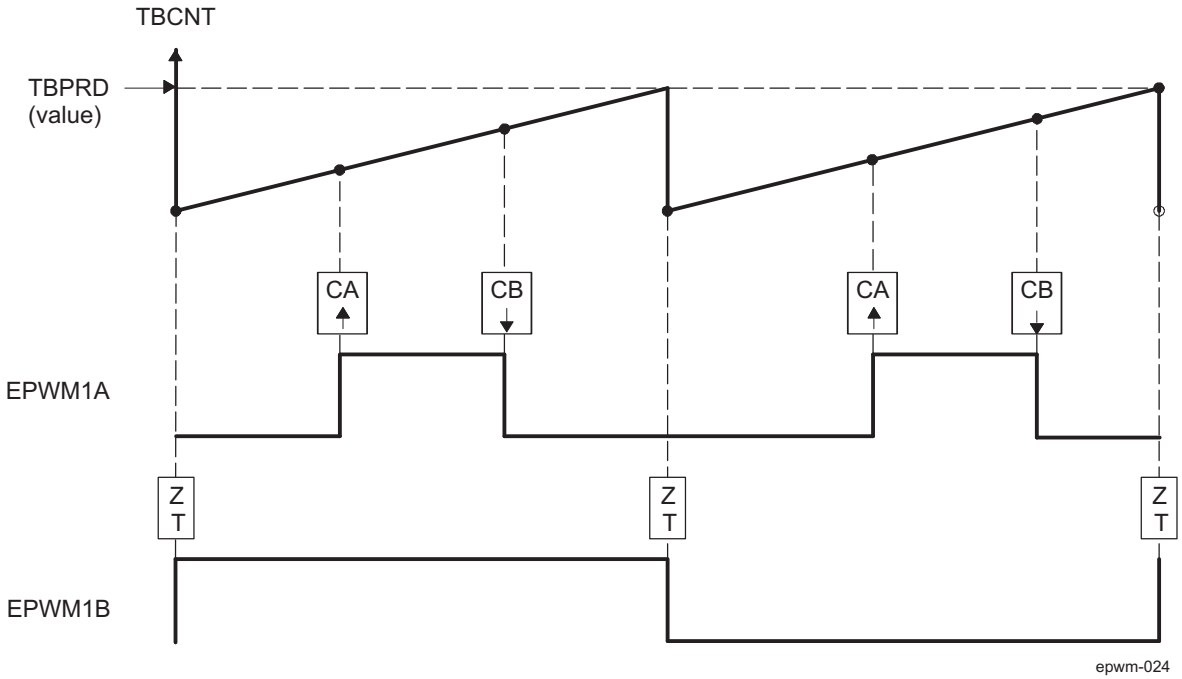
Register	Bitfiled	Value	Comments
EPWM_TBPRD	TBPRD	600 (258h)	Period = 601 TBCLK counts
EPWM_TBPHS	TBPHS	0	Clear Phase Register to 0
EPWM_TBCNT	TBCNT	0	Clear TB counter
EPWM_TBCTL	CTRMODE	TB_UP	
	PHSEN	TB_DISABLE	Phase loading disabled
	PRDLD	TB_SHADOW	
	SYNCOSEL	TB_SYNC_DISABLE	
	HSPCLKDIV	TB_DIV1	TBCLK = SYSCLKOUT
	CLKDIV	TB_DIV1	
EPWM_CMPA	CMPA	350 (15Eh)	Compare A = 350 TBCLK counts
EPWM_CMPB	CMPB	200 (C8h)	Compare B = 200 TBCLK counts
EPWM_CMPCTL	SHDWAMODE	CC_SHADOW	
	SHDWBMODE	CC_SHADOW	
	LOADAMODE	CC_CTR_ZERO	Load on TBCNT = 0
	LOADBMODE	CC_CTR_ZERO	Load on TBCNT = 0
EPWM_AQCTLA	PRD	AQ_CLEAR	
	CAU	AQ_SET	
EPWM_AQCTLB	PRD	AQ_CLEAR	
	CBU	AQ_SET	

Table 21-33. EPWM Run Time Changes for Figure 21-24

Register	Bit	Value	Comments
EPWM_CMPA	CMPA	Duty1A	Adjust duty for output EPWM1A
EPWM_CMPB	CMPB	Duty1B	Adjust duty for output EPWM1B

Table 21-34 and Table 21-35 contains initialization and runtime register configurations for the waveforms Figure 21-25. Use the code in Example 21-1 to define the headers.

Figure 21-25. Up-Count, Pulse Placement Asymmetric Waveform With Independent Modulation on EPWM1A



(1) PWM frequency = 1/((TBPRD + 1) × T_{TBCLK})

(2) Pulse can be placed anywhere within the PWM cycle (0000h - TBPRD)

(3) High time duty proportional to (CMPB - CMPA)

(4) EPWM1B can be used to generate a 50% duty square wave with frequency = 1/2 × ((TBPRD + 1) × TBCLK)

Table 21-34. EPWM Initialization for Figure 21-25

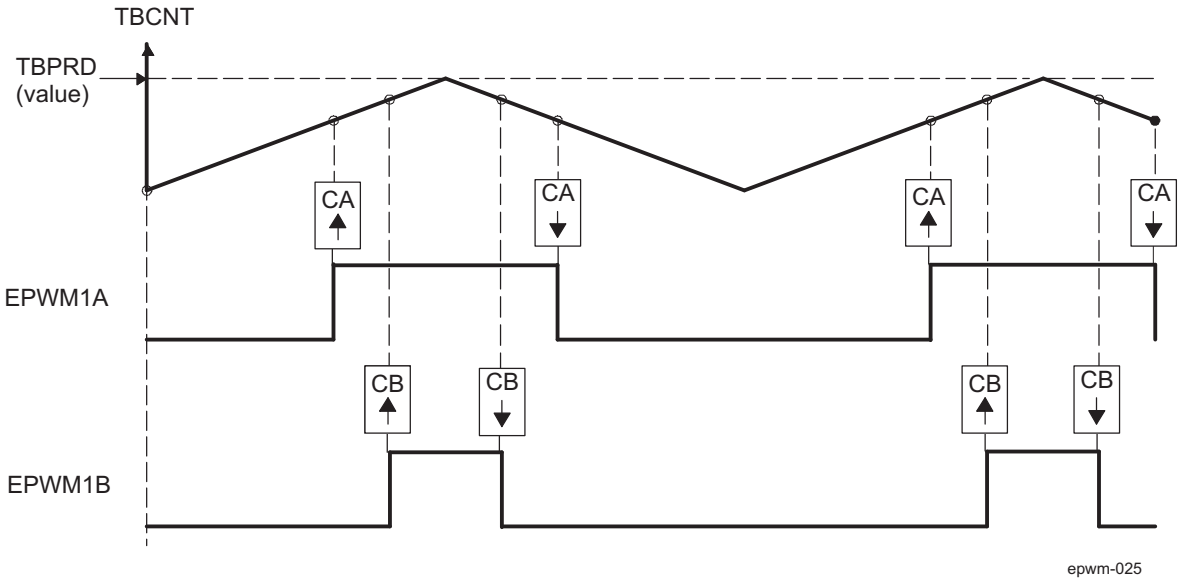
Register	Bitfield	Value	Comments
EPWM_TBPRD	TBPRD	600 (258h)	Period = 601 TBCLK counts
EPWM_TBPHS	TBPHS	0	Clear Phase Register to 0
EPWM_TBCNT	TBCNT	0	Clear TB counter
EPWM_TBCTL	CTRMODE	TB_UP	
	PHSEN	TB_DISABLE	Phase loading disabled
	PRDLD	TB_SHADOW	
	SYNCOSEL	TB_SYNC_DISABLE	
	HSPCLKDIV	TB_DIV1	TBCLK = SYSCLKOUT
	CLKDIV	TB_DIV1	
EPWM_CMPA	CMPA	200 (C8h)	Compare A = 200 TBCLK counts
EPWM_CMPB	CMPB	400 (190h)	Compare B = 400 TBCLK counts
EPWM_CMPCTL	SHDWAMODE	CC_SHADOW	
	SHDWBMODE	CC_SHADOW	
	LOADAMODE	CC_CTR_ZERO	Load on TBCNT = 0
	LOADBMODE	CC_CTR_ZERO	Load on TBCNT = 0
EPWM_AQCTLA	CAU	AQ_SET	
	CBU	AQ_CLEAR	
EPWM_AQCTLB	ZRO	AQ_TOGGLE	

Table 21-35. EPWM Run Time Changes for Figure 21-25

Register	Bitfield	Value	Comments
EPWM_CMPA	CMPA	EdgePosA	Adjust duty for output EPWM1A
EPWM_CMPB	CMPB	EdgePosB	

Table 21-36 and Table 21-37 contains initialization and runtime register configurations for the waveforms in Figure 21-26. Use the code in Example 21-1 to define the headers.

Figure 21-26. Up-Down-Count, Dual Edge Symmetric Waveform, With Independent Modulation on EPWM1A and EPWM1B — Active Low



(1) PWM period = 2 x TBPRD × T_{TBCLK}

(2) Duty modulation for EPWM1A is set by CMPA, and is active low (that is, the low time duty is proportional to CMPA).

(3) Duty modulation for EPWM1B is set by CMPB and is active low (that is, the low time duty is proportional to CMPB).

(4) Outputs EPWM1A and EPWM1B can drive independent power switches

Table 21-36. EPWM Initialization for Figure 21-26

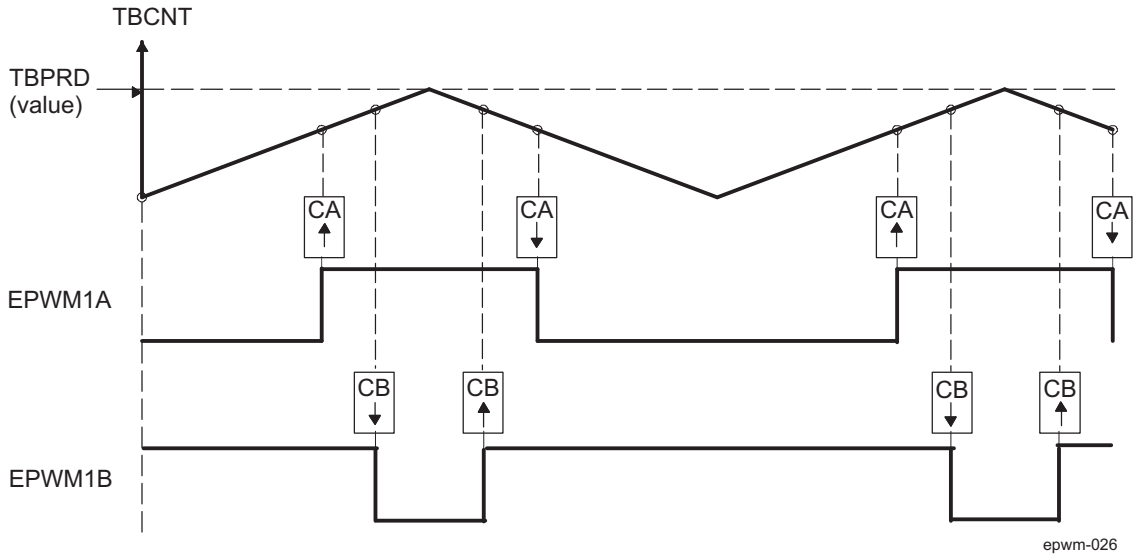
Register	Bitfield	Value	Comments
EPWM_TBPRD	TBPRD	600 (258h)	Period = 601 TBCLK counts
EPWM_TBPHS	TBPHS	0	Clear Phase Register to 0
EPWM_TBCNT	TBCNT	0	Clear TB counter
EPWM_TBCTL	CTRMODE	TB_UPDOWN	
	PHSEN	TB_DISABLE	Phase loading disabled
	PRDLD	TB_SHADOW	
	SYNCOSEL	TB_SYNC_DISABLE	
	HSPCLKDIV	TB_DIV1	TBCLK = SYSCLKOUT
	CLKDIV	TB_DIV1	
EPWM_CMPA	CMPA	400 (190h)	Compare A = 400 TBCLK counts
EPWM_CMPB	CMPB	500 (1F4h)	Compare B = 500 TBCLK counts
EPWM_CMPCTL	SHDWAMODE	CC_SHADOW	
	SHDWBMODE	CC_SHADOW	
	LOADAMODE	CC_CTR_ZERO	Load on TBCNT = 0
	LOADBMODE	CC_CTR_ZERO	Load on TBCNT = 0
EPWM_AQCTLA	CAU	AQ_SET	
	CAD	AQ_CLEAR	
EPWM_AQCTLB	CBU	AQ_SET	
	CBD	AQ_CLEAR	

Table 21-37. EPWM Run Time Changes for Figure 21-26

Register	Bitfield	Value	Comments
EPWM_CMPA	CMPA	Duty1A	Adjust duty for output EPWM1A
EPWM_CMPB	CMPB	Duty1B	Adjust duty for output EPWM1B

Table 21-38 and Table 21-39 contains initialization and runtime register configurations for the waveforms in Figure 21-27. Use the code in Example 21-1 to define the headers.

Figure 21-27. Up-Down-Count, Dual Edge Symmetric Waveform, With Independent Modulation on EPWM1A and EPWM1B — Complementary



(1) PWM period = 2 × TBPRD × T_{TBCLK}

(2) Duty modulation for EPWM1A is set by CMPA, and is active low, i.e., low time duty proportional to CMPA

(3) Duty modulation for EPWM1B is set by CMPB and is active high, i.e., high time duty proportional to CMPB

(4) Outputs EPWM can drive upper/lower (complementary) power switches

(5) Dead-band = CMPB - CMPA (fully programmable edge placement by software). Note the dead-band module is also available if the more classical edge delay method is required.

Table 21-38. EPWM Initialization for Figure 21-27

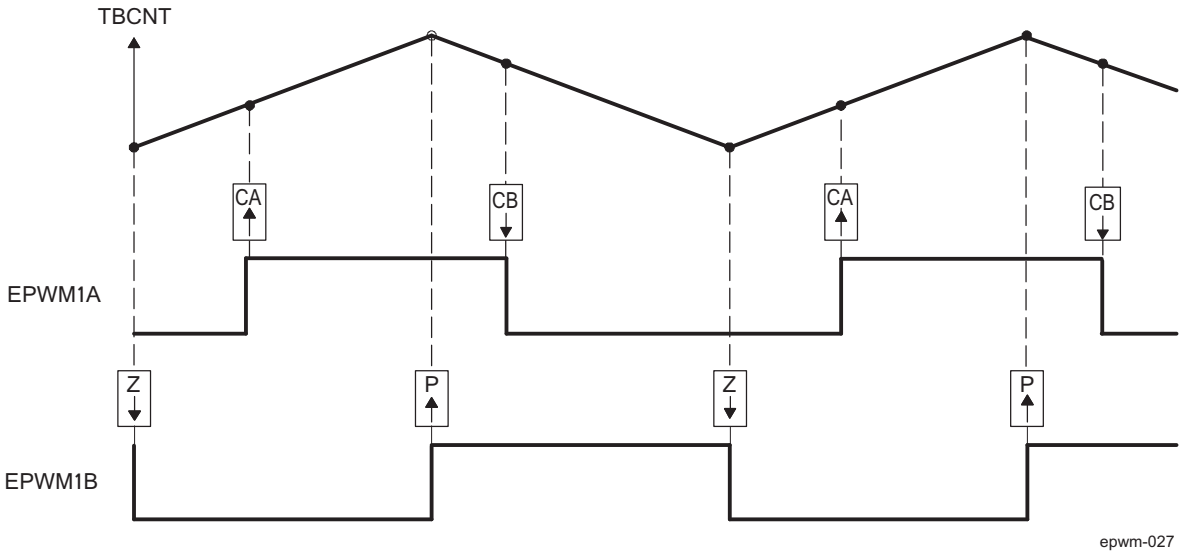
Register	Bitfield	Value	Comments
EPWM_TBPRD	TBPRD	600 (258h)	Period = 601 TBCLK counts
EPWM_TBPHS	TBPHS	0	Clear Phase Register to 0
EPWM_TBCNT	TBCNT	0	Clear TB counter
EPWM_TBCTL	CTRMODE	TB_UPDOWN	
	PHSEN	TB_DISABLE	Phase loading disabled
	PRDLD	TB_SHADOW	
	SYNCOSEL	TB_SYNC_DISABLE	
	HSPCLKDIV	TB_DIV1	TBCLK = SYSCLKOUT
	CLKDIV	TB_DIV1	
EPWM_CMPA	CMPA	350 (15Eh)	Compare A = 350 TBCLK counts
EPWM_CMPB	CMPB	400 (190h)	Compare B = 400 TBCLK counts
EPWM_CMPCTL	SHDWAMODE	CC_SHADOW	
	SHDWBMODE	CC_SHADOW	
	LOADAMODE	CC_CTR_ZERO	Load on TBCNT = 0
	LOADBMODE	CC_CTR_ZERO	Load on TBCNT = 0
EPWM_AQCTLA	CAU	AQ_SET	
	CAD	AQ_CLEAR	
EPWM_AQCTLB	CBU	AQ_CLEAR	
	CBD	AQ_SET	

Table 21-39. EPWM Run Time Changes for Figure 21-27

Register	Bitfield	Value	Comments
EPWM_CMPA	CMPA	Duty1A	Adjust duty for output EPWM1A
EPWM_CMPB	CMPB	Duty1B	Adjust duty for output EPWM1B

Table 21-40 and Table 21-41 contains initialization and runtime register configurations for the waveforms in Figure 21-28. Use the code in Example 21-1 to define the headers.

Figure 21-28. Up-Down-Count, Dual Edge Asymmetric Waveform, With Independent Modulation on EPWM1A—Active Low



(1) PWM period = 2 × TBPRD × TBCLK

(2) Rising edge and falling edge can be asymmetrically positioned within a PWM cycle. This allows for pulse placement techniques.

(3) Duty modulation for EPWM1A is set by CMPA and CMPB.

(4) Low time duty for EPWM1A is proportional to (CMPA + CMPB).

(5) To change this example to active high, CMPA and CMPB actions need to be inverted (i.e., Set ! Clear and Clear Set).

(6) Duty modulation for EPWM1B is fixed at 50% (utilizes spare action resources for EPWM1B)

Table 21-40. EPWM Initialization for Figure 21-28

Register	Bitfield	Value	Comments
EPWM_TBPRD	TBPRD	600 (258h)	Period = 601 TBCLK counts
EPWM_TBPHS	TBPHS	0	Clear Phase Register to 0
EPWM_TBCNT	TBCNT	0	Clear TB counter
EPWM_TBCTL	CTRMODE	TB_UPDOWN	
	PHSEN	TB_DISABLE	Phase loading disabled
	PRDLD	TB_SHADOW	
	SYNCOSEL	TB_SYNC_DISABLE	
	HSPCLKDIV	TB_DIV1	TBCLK = SYSCLKOUT
	CLKDIV	TB_DIV1	
EPWM_CMPA	CMPA	250 (FAh)	Compare A = 250 TBCLK counts
EPWM_CMPB	CMPB	450 (1C2h)	Compare B = 450 TBCLK counts
EPWM_CMPCTL	SHDWAMODE	CC_SHADOW	
	SHDWBMODE	CC_SHADOW	
	LOADAMODE	CC_CTR_ZERO	Load on TBCNT = 0
	LOADBMODE	CC_CTR_ZERO	Load on TBCNT = 0
EPWM_AQCTLA	CAU	AQ_SET	
	CBD	AQ_CLEAR	
EPWM_AQCTLB	ZRO	AQ_CLEAR	
	PRD	AQ_SET	

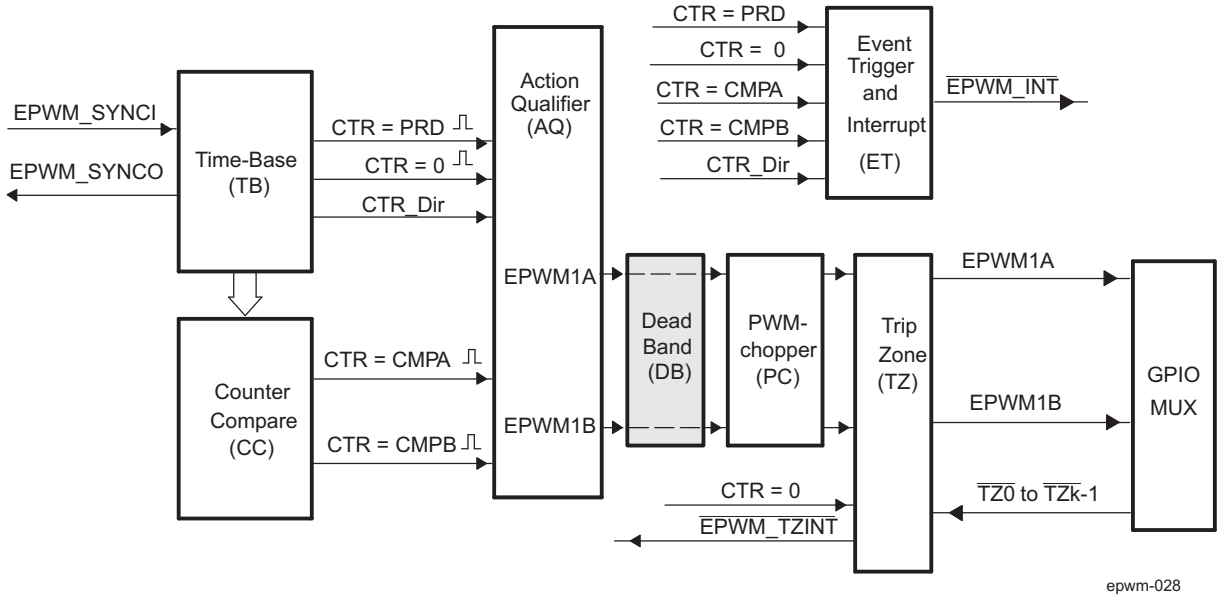
Table 21-41. EPWM Run Time Changes for Figure 21-28

Register	Bitfield	Value	Comments
EPWM_CMPA	CMPA	EdgePosA	Adjust duty for output EPWM1A
EPWM_CMPB	CMPB	EdgePosB	

21.2.2.6 ePWM Dead-Band Generator (DB) Submodule

Figure 21-29 illustrates the dead-band generator submodule within the ePWM module.

Figure 21-29. Dead-Band Generator Submodule



21.2.2.6.1 Purpose of the ePWM Dead-Band Submodule

The "Action-qualifier (AQ) Module" section discussed how it is possible to generate the required dead-band by having full control over edge placement using both the CMPA and CMPB resources of the ePWM module. However, if the more classical edge delay-based dead-band with polarity control is required, then the dead-band generator submodule should be used.

The key functions of the dead-band generator submodule are:

- Generating appropriate signal pairs (EPWM1A and EPWM1B) with dead-band relationship from a single EPWM1A input
- Programming signal pairs for:
  - Active high (AH)
  - Active low (AL)
  - Active high complementary (AHC)
  - Active low complementary (ALC)
- Adding programmable delay to rising edges (RED)
- Adding programmable delay to falling edges (FED)
- Can be totally bypassed from the signal path (note dotted lines in diagram)

21.2.2.6.2 Controlling and Monitoring the ePWM Dead-Band Submodule

The dead-band generator submodule operation is controlled and monitored via the following registers:

Table 21-42. Dead-Band Generator Submodule Registers

Acronym	Register Description	Address Offset	Shadowed
EPWM_DBCTL	Dead-Band Control Register	1Eh	No
EPWM_DBRED	Dead-Band Rising Edge Delay Count Register	20h	No
EPWM_DBFED	Dead-Band Falling Edge Delay Count Register	22h	No

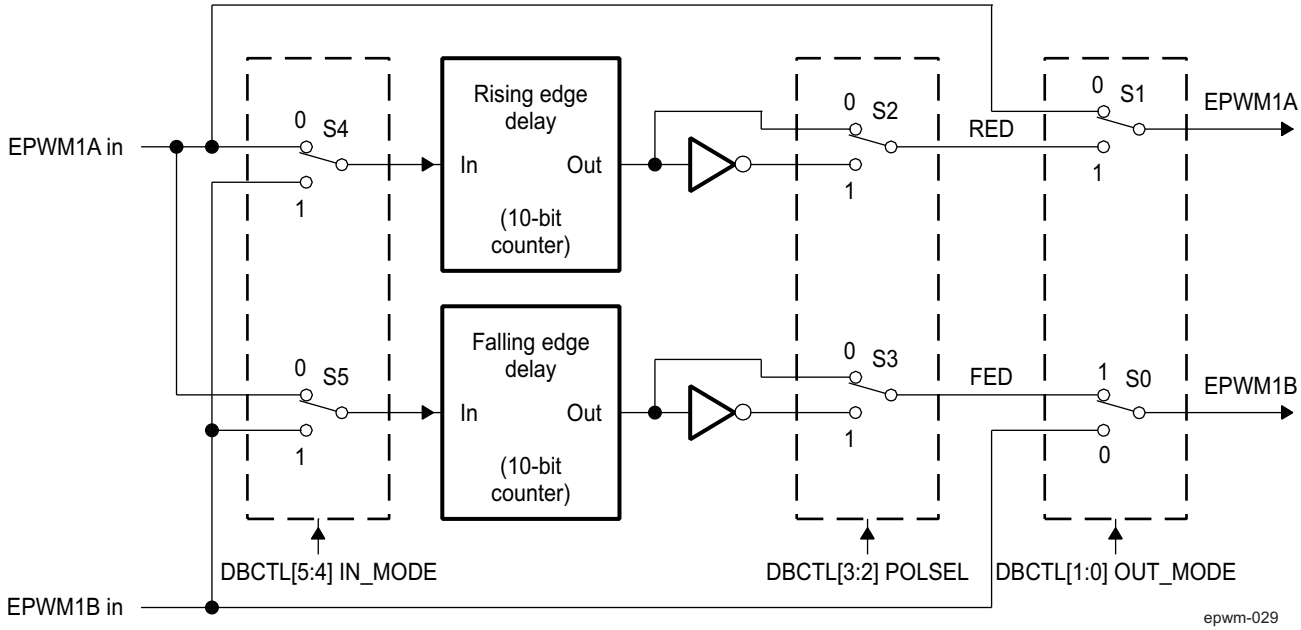
21.2.2.6.3 Operational Highlights for the ePWM Dead-Band Generator Submodule

The following sections provide the operational highlights.

The dead-band submodule has two groups of independent selection options as shown in Figure 21-30.

- **Input Source Selection:** The input signals to the dead-band module are the EPWM1A and EPWM1B output signals from the action-qualifier. In this section they will be referred to as EPWM1A In and EPWM1B In. Using the EPWM_DBCTL[5:4] IN_MODE control bits, the signal source for each delay, falling-edge or rising-edge, can be selected:
  - EPWM1A In is the source for both falling-edge and rising-edge delay. This is the default mode.
  - EPWM1A In is the source for falling-edge delay, EPWM1B In is the source for rising-edge delay.
  - EPWM1A In is the source for rising edge delay, EPWM1B In is the source for falling-edge delay.
  - EPWM1B In is the source for both falling-edge and rising-edge delay.
- **Output Mode Control:** The output mode is configured by way of the EPWM_DBCTL[1:0] OUT_MODE bits. These bits determine if the falling-edge delay, rising-edge delay, neither, or both are applied to the input signals.
- **Polarity Control:** The polarity control (EPWM_DBCTL[3:2] POLSEL) allows you to specify whether the rising-edge delayed signal and/or the falling-edge delayed signal is to be inverted before being sent out of the dead-band submodule.

Figure 21-30. Configuration Options for the ePWM Dead-Band Generator Submodule



Although all combinations are supported, not all are typical usage modes. Table 21-43 lists some classical dead-band configurations. These modes assume that the EPWM_DBCTL[5:4] IN_MODE is configured such that EPWM1A In is the source for both falling-edge and rising-edge delay. Enhanced, or non-traditional modes can be achieved by changing the input signal source. The modes shown in Table 21-43 fall into the following categories:

- **Mode 1: Bypass both falling-edge delay (FED) and rising-edge delay (RED)** Allows you to fully disable the dead-band submodule from the PWM signal path.
- **Mode 2-5: Classical Dead-Band Polarity Settings** These represent typical polarity configurations that should address all the active high/low modes required by available industry power switch gate drivers. The waveforms for these typical cases are shown in Figure 21-31. Note that to generate equivalent waveforms to Figure 21-31, configure the action-qualifier submodule to generate the signal as shown for EPWM1A.
- **Mode 6: Bypass rising-edge-delay and Mode 7: Bypass falling-edge-delay** Finally the last two entries in Table 21-43 show combinations where either the falling-edge-delay (FED) or rising-edge-delay (RED) blocks are bypassed.

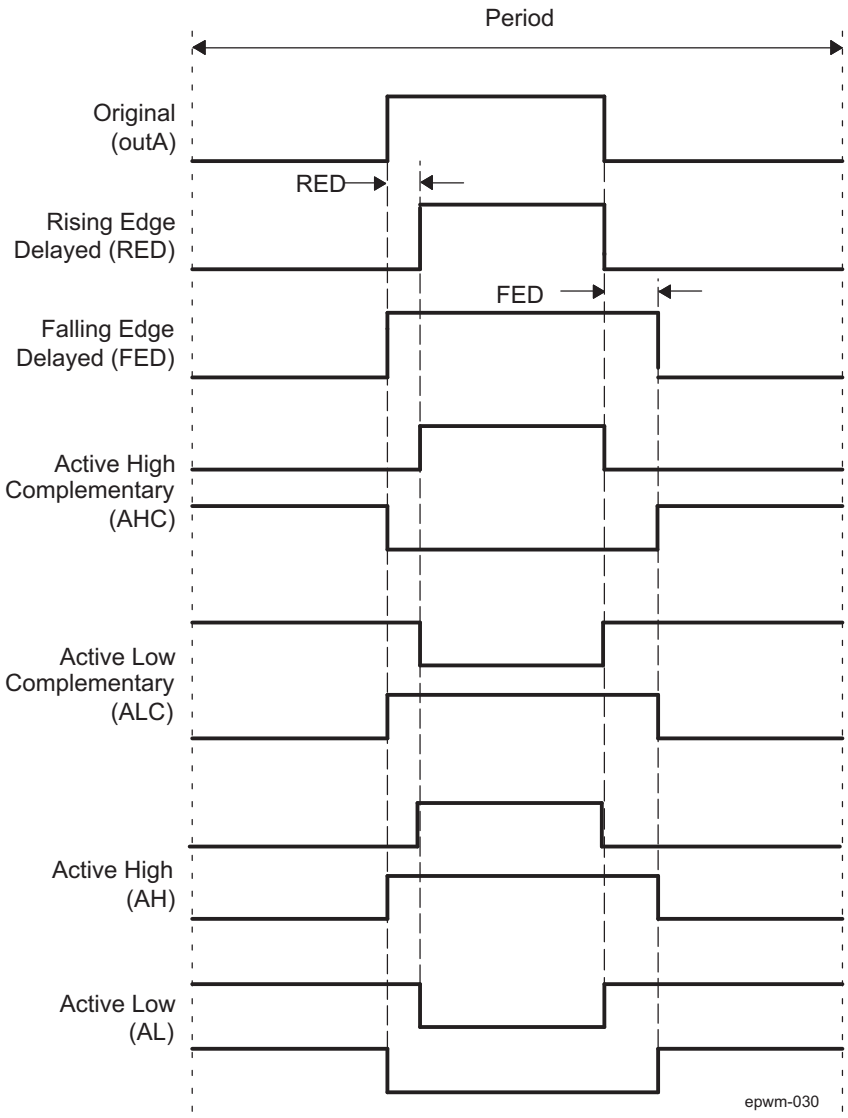
Table 21-43. Classical Dead-Band Operating Modes

		EPWM_DBCTL[3:2] POLSEL		EPWM_DBCTL[1:0] OUT_MODE	
Mode	Mode Description [1]	S3	S2	S1	S0
1	EPWM1A and EPWM1B Passed Through (No Delay)	x	x	0	0
2	Active High Complementary (AHC)	1	0	1	1
3	Active Low Complementary (ALC)	0	1	1	1
4	Active High (AH)	0	0	1	1
5	Active Low (AL)	1	1	1	1
6	EPWM1A Out = EPWM1A In (No Delay) EPWM1B Out = EPWM1A In with Falling Edge Delay	0 or 1	0 or 1	0	1
7	EPWM1A Out = EPWM1A In with Rising Edge Delay EPWM1B Out = EPWM1B In with No Delay	0 or 1	0 or 1	1	0

(1) These are classical dead-band modes and assume that EPWM_DBCTL[5:4] IN_MODE = 0b00. That is, EPWM1A in is the source for both the falling-edge and rising-edge delays. Enhanced, non-traditional modes can be achieved by changing the IN_MODE configuration.

Figure 21-31 shows waveforms for typical cases where 0% < duty < 100%.

Figure 21-31. Dead-Band Waveforms for Typical Cases (0% < Duty < 100%)



The dead-band submodule supports independent values for rising-edge (RED) and falling-edge (FED) delays. The amount of delay is programmed using the EPWM_DBRED and EPWM_DBFED registers. These are 10-bit registers and their value represents the number of time-base clock, TBCLK, periods a signal edge is delayed by. For example, the formula to calculate falling-edge-delay and rising-edge-delay are:

FED = EPWM_DBFED × T_{TBCLK}

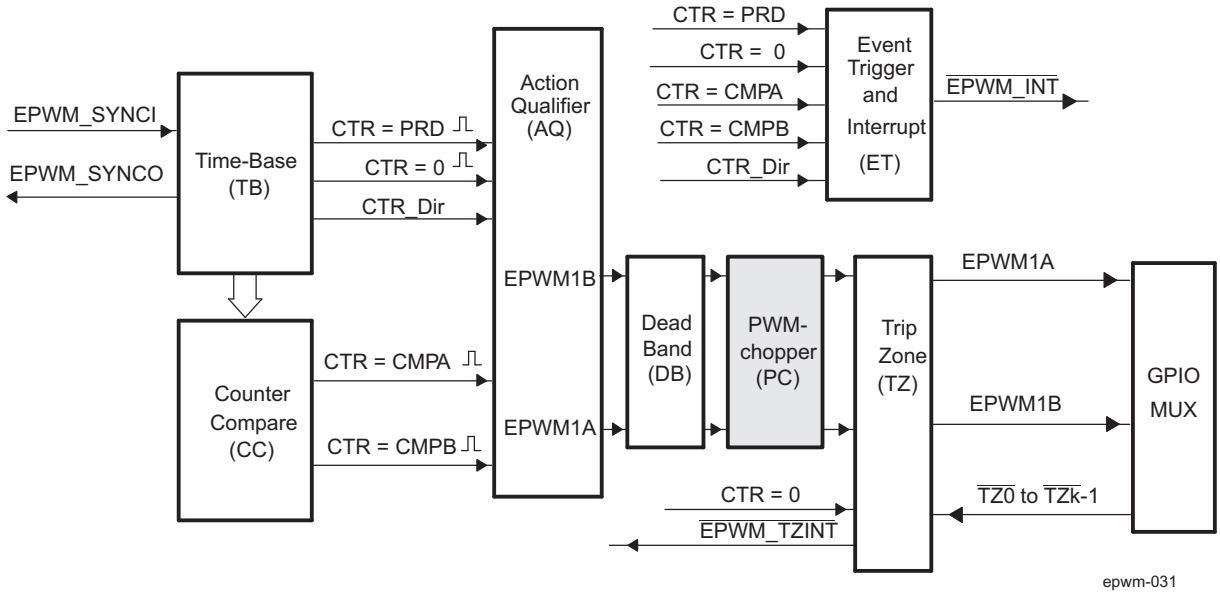
RED = EPWM_DBRED × T_{TBCLK}

Where T_{TBCLK} is the period of TBCLK, the prescaled version of SYSCLKOUT.

21.2.2.7 PWM-Chopper (PC) Submodule

Figure 21-32 illustrates the PWM-chopper (PC) submodule within the ePWM module. The PWM-chopper submodule allows a high-frequency carrier signal to modulate the PWM waveform generated by the action-qualifier and dead-band submodules. This capability is important if you need pulse transformer-based gate drivers to control the power switching elements.

Figure 21-32. PWM-Chopper Submodule



21.2.2.7.1 Purpose of the PWM-Chopper Submodule

The key functions of the PWM-chopper submodule are:

- Programmable chopping (carrier) frequency
- Programmable pulse width of first pulse
- Programmable duty cycle of second and subsequent pulses
- Can be fully bypassed if not required

21.2.2.7.2 Controlling the PWM-Chopper Submodule

The PWM-chopper submodule operation is controlled via the register in Table 21-44.

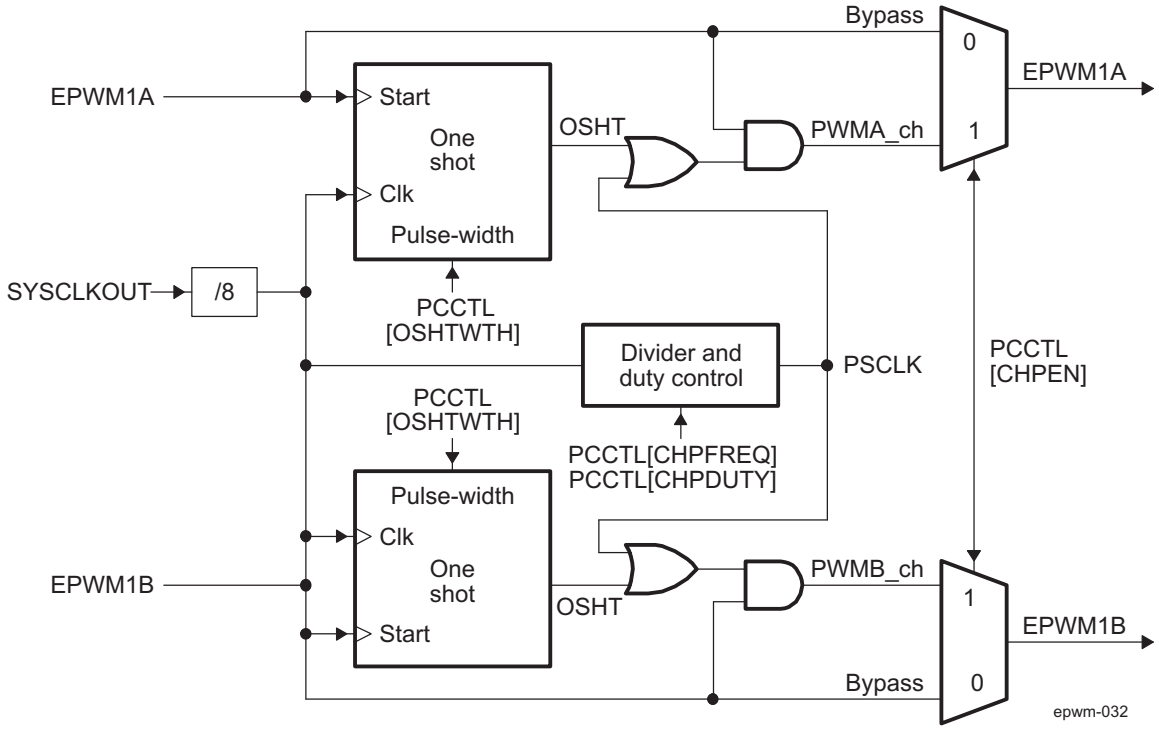
Table 21-44. PWM-Chopper Submodule Registers

Acronym	Register Description	Address Offset	Shadowed
EPWM_PCCTL	PWM-chopper Control Register	3Ch	No

21.2.2.7.3 Operational Highlights for the PWM-Chopper Submodule

Figure 21-33 shows the operational details of the PWM-chopper submodule. The carrier clock is derived from SYSCLKOUT. Its frequency and duty cycle are controlled via the CHPFREQ and CHPDUTY bits in the EPWM_PCCTL register. The one-shot block is a feature that provides a high energy first pulse to ensure hard and fast power switch turn on, while the subsequent pulses sustain pulses, ensuring the power switch remains on. The one-shot width is programmed via the OSHTWTH bits. The PWM-chopper submodule can be fully disabled (bypassed) via the CHPEN bit.

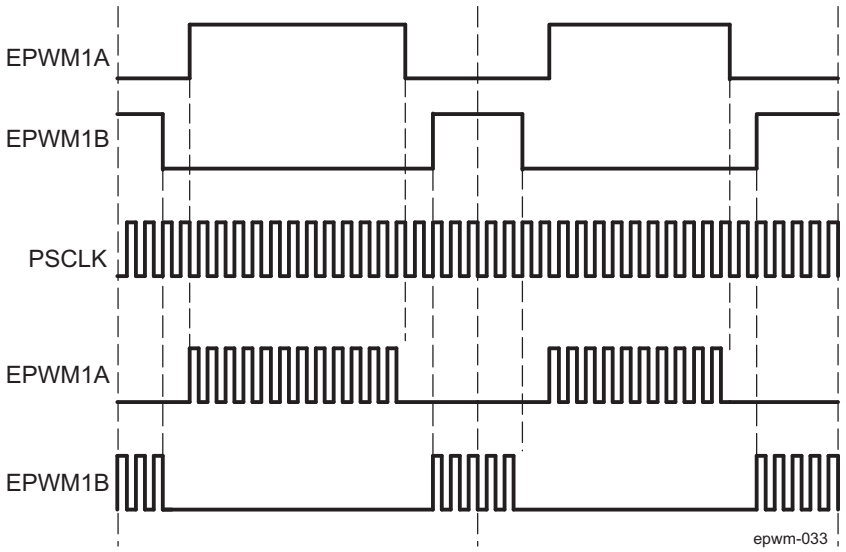
Figure 21-33. PWM-Chopper Submodule Signals and Registers



21.2.2.7.4 PWM Chopper Waveforms

Figure 21-34 shows simplified waveforms of the chopping action only; one-shot and duty-cycle control are not shown. Details of the one-shot and duty-cycle control are discussed in the following sections.

Figure 21-34. Simple PWM-Chopper Submodule Waveforms Showing Chopping Action Only



21.2.2.7.4.1 PWM-Chopper One-Shot Pulse

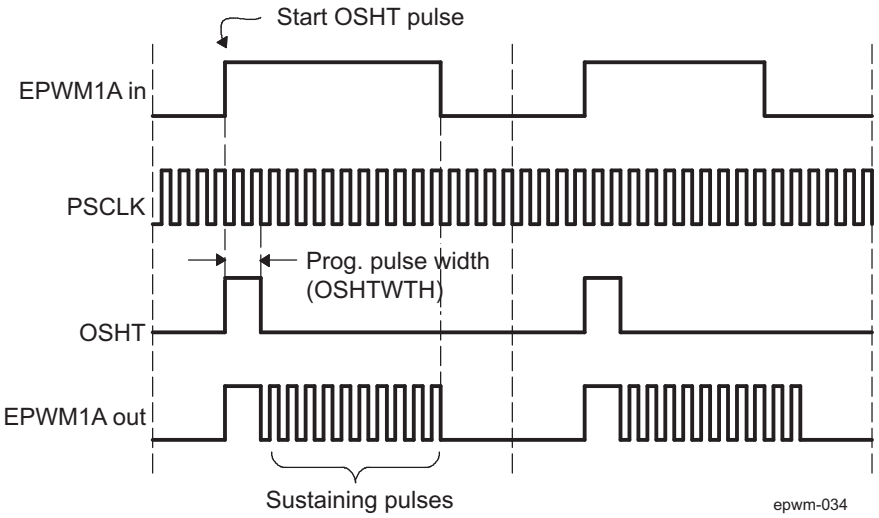
The width of the first pulse can be programmed to any of 16 possible pulse width values. The width or period of the first pulse is given by:

$T_{1stpulse} = T_{SYSCLKOUT} \times 8 \times OSHTWTH$

Where $T_{SYSCLKOUT}$ is the period of the system clock (SYSCLKOUT) and OSHTWTH is the four control bits (value from 1 to 16)

Figure 21-35 shows the first and subsequent sustaining pulses.

Figure 21-35. PWM-Chopper Submodule Waveforms Showing the First Pulse and Subsequent Sustaining Pulses

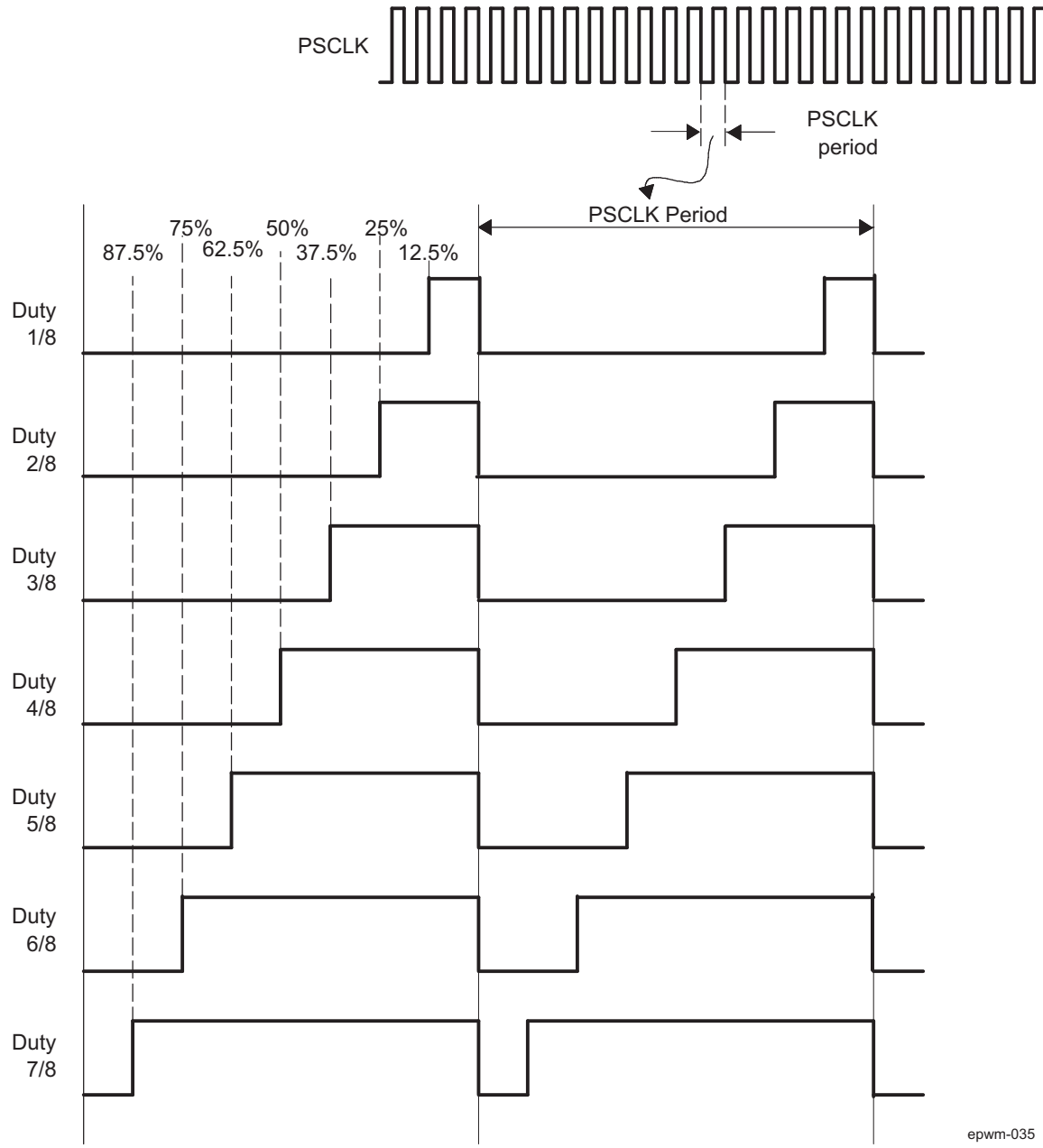


21.2.2.7.4.2 PWM-Chopper Duty Cycle Control

Pulse transformer-based gate drive designs need to comprehend the magnetic properties or characteristics of the transformer and associated circuitry. Saturation is one such consideration. To assist the gate drive designer, the duty cycles of the second and subsequent pulses have been made programmable. These sustaining pulses ensure the correct drive strength and polarity is maintained on the power switch gate during the on period, and hence a programmable duty cycle allows a design to be tuned or optimized via software control.

Figure 21-36 shows the duty cycle control that is possible by programming the CHPDUTY bits. One of seven possible duty ratios can be selected ranging from 12.5% to 87.5%.

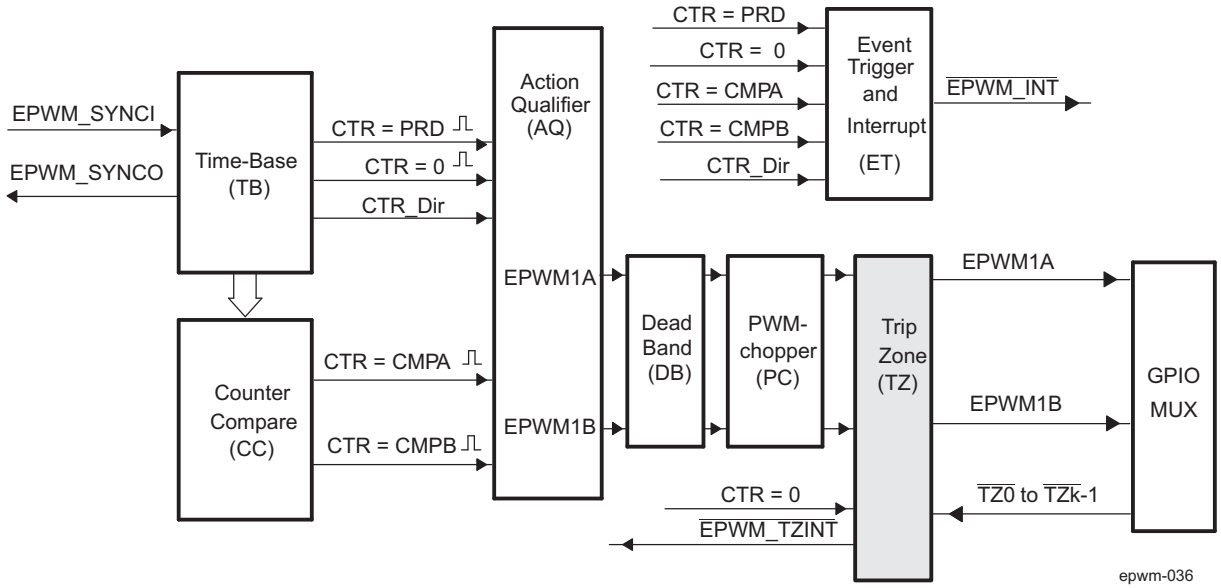
Figure 21-36. PWM-Chopper Submodule Waveforms Showing the Pulse Width (Duty Cycle) Control of Sustaining Pulses



21.2.2.8 ePWM Trip-Zone (TZ) Submodule

Figure 21-37 shows how the trip-zone (TZ) submodule fits within the ePWM module. The ePWM module is connected to every $\overline{TZ}$ signal that is sourced from the GPIO MUX. This signal indicates external fault or trip conditions, and the ePWM outputs can be programmed to respond accordingly when a fault occurs. See Section 21.1.3 to determine the number of trip-zone pins available for the device.

Figure 21-37. ePWM Trip-Zone Submodule



21.2.2.8.1 Purpose of the ePWM Trip-Zone Submodule

The key functions of the trip-zone submodule are:

- Trip inputs $\overline{TZ0}$ to $\overline{TZk\text{-}1}$ can be flexibly mapped to any ePWM module.
- Upon a fault condition, outputs EPWM1A and EPWM1B can be forced to one of the following:
  - High
  - Low
  - High-impedance
  - No action taken
- Support for one-shot trip (OSHT) for major short circuits or over-current conditions.
- Support for cycle-by-cycle tripping (CBC) for current limiting operation.
- Each trip-zone input pin can be allocated to either one-shot or cycle-by-cycle operation.
- Interrupt generation is possible on any trip-zone pin.
- Software-forced tripping is also supported.
- The trip-zone submodule can be fully bypassed if it is not required.

NOTE: For ePWM1, from the tripzone inputs $\overline{EPWM\_TRIP\_TZ[5:0]}$, ONLY the $\overline{EPWM\_TRIP\_TZ[0]}$ input of ePWM tripzone is chip accessible and usable, i.e. k=1.

21.2.2.8.2 Controlling and Monitoring the ePWM Trip-Zone Submodule

The trip-zone submodule operation is controlled and monitored through the following registers:

Table 21-45. ePWM Trip-Zone Submodule Registers

Acronym	Register Description	Address Offset	Shadowed
EPWM_TZSEL	Trip-Zone Select Register	24h	No
EPWM_TZCTL	Trip-Zone Control Register	28h	No
EPWM_TZEINT	Trip-Zone Enable Interrupt Register	2Ah	No
EPWM_TZFLG	Trip-Zone Flag Register	2Ch	No
EPWM_TZCLR	Trip-Zone Clear Register	2Eh	No
EPWM_TZFRC	Trip-Zone Force Register	30h	No

21.2.2.8.3 Operational Highlights for the ePWM Trip-Zone Submodule

The following sections describe the operational highlights and configuration options for the trip-zone submodule.

The trip-zone signals at pin $\overline{TZ0}$ to $\overline{TZk\text{-}1}$ is an active-low input signal. When the pin goes low, it indicates that a trip event has occurred. The ePWM module can be configured to ignore or use each of the trip-zone pins. Which trip-zone pins are used is determined by the EPWM_TZSEL register. The trip-zone signal may or may not be synchronized to the system clock (SYSCLKOUT). A minimum of 1 SYSCLKOUT low pulse on the $\overline{TZ}$ $\overline{n}$ inputs is sufficient to trigger a fault condition in the ePWM module. The asynchronous trip makes sure that if clocks are missing for any reason, the outputs can still be tripped by a valid event present on the $\overline{TZk}$ inputs.

NOTE: Only the TZ[0] input is accessible at chip level (i.e. k=1).

The $\overline{TZk}$ input can be individually configured to provide either a cycle-by-cycle or one-shot trip event for a ePWM module. The configuration is determined by the EPWM_TZSEL[7:0] CBCk and EPWM_TZSEL[15:8] OSHTk bits (where k corresponds to the trip pin) respectively.

- **Cycle-by-Cycle (CBC):** When a cycle-by-cycle trip event occurs, the action specified in the EPWM_TZCTL register is carried out immediately on the EPWM1A and/or EPWM1B output. Table 21-46 lists the possible actions. In addition, the cycle-by-cycle trip event flag (EPWM_TZFLG[1] CBC) is set and a ePWM0_TZINT interrupt is generated if it is enabled in the EPWM_TZEINT register.

  The specified condition on the pins is automatically cleared when the ePWM time-base counter reaches zero (EPWM_TBCNT bitfield TBCNT = 0000h) if the trip event is no longer present. Therefore, in this mode, the trip event is cleared or reset every PWM cycle. The EPWM_TZFLG[1] CBC flag bit will remain set until it is manually cleared by writing to the EPWM_TZCLR[1] CBC bit. If the cycle-by-cycle trip event is still present when the EPWM_TZFLG[1] CBC bit is cleared, then it will again be immediately set.
- **One-Shot (OSHT):** When a one-shot trip event occurs, the action specified in the EPWM_TZCTL register is carried out immediately on the EPWM1A and/or EPWM1B output. Table 21-46 lists the possible actions. In addition, the one-shot trip event flag (EPWM_TZFLG[2] OST) is set and a ePWM0_TZINT interrupt is generated if it is enabled in the EPWM_TZEINT register. The one-shot trip condition must be cleared manually by writing to the EPWM_TZCLR[2] OST bit.

The action taken when a trip event occurs can be configured individually for each of the ePWM output pins by way of the EPWM_TZCTL[1:0] TZA and EPWM_TZCTL[3:2] TZB register bits. One of four possible actions, shown in Table 21-46, can be taken on a trip event.

Table 21-46. Possible Actions On an ePWM Trip Event

EPWM_TZCTL[1:0] TZA and/or EPWM_TZCTL[3:2] TZB	EPWM1A and/or EPWM1B	Comment
0	High-Impedance	Tripped
1h	Force to High State	Tripped
2h	Force to Low State	Tripped
3h	No Change	Do Nothing. No change is made to the output.

Example 21-2. ePWM Trip-Zone Configurations

Scenario A:
A one-shot trip event on $\overline{TZ0}$ pulls both EPWM1A, EPWM1B low and also forces EPWM2A and EPWM2B high.

- Configure the ePWM1 registers as follows:
  - EPWM_TZSEL[8] OSHT1 = 1: enables $\overline{TZ}$ as a one-shot event source for ePWM1
  - EPWM_TZCTL[1:0] TZA = 2: EPWM1A will be forced low on a trip event.
  - EPWM_TZCTL[3:2] TZB = 2: EPWM1B will be forced low on a trip event.

21.2.2.8.4 Generating ePWM Trip Event Interrupts

Figure 21-38 and Figure 21-39 illustrate the trip-zone submodule control and interrupt logic, respectively.

Figure 21-38. ePWM Trip-Zone Submodule Mode Control Logic

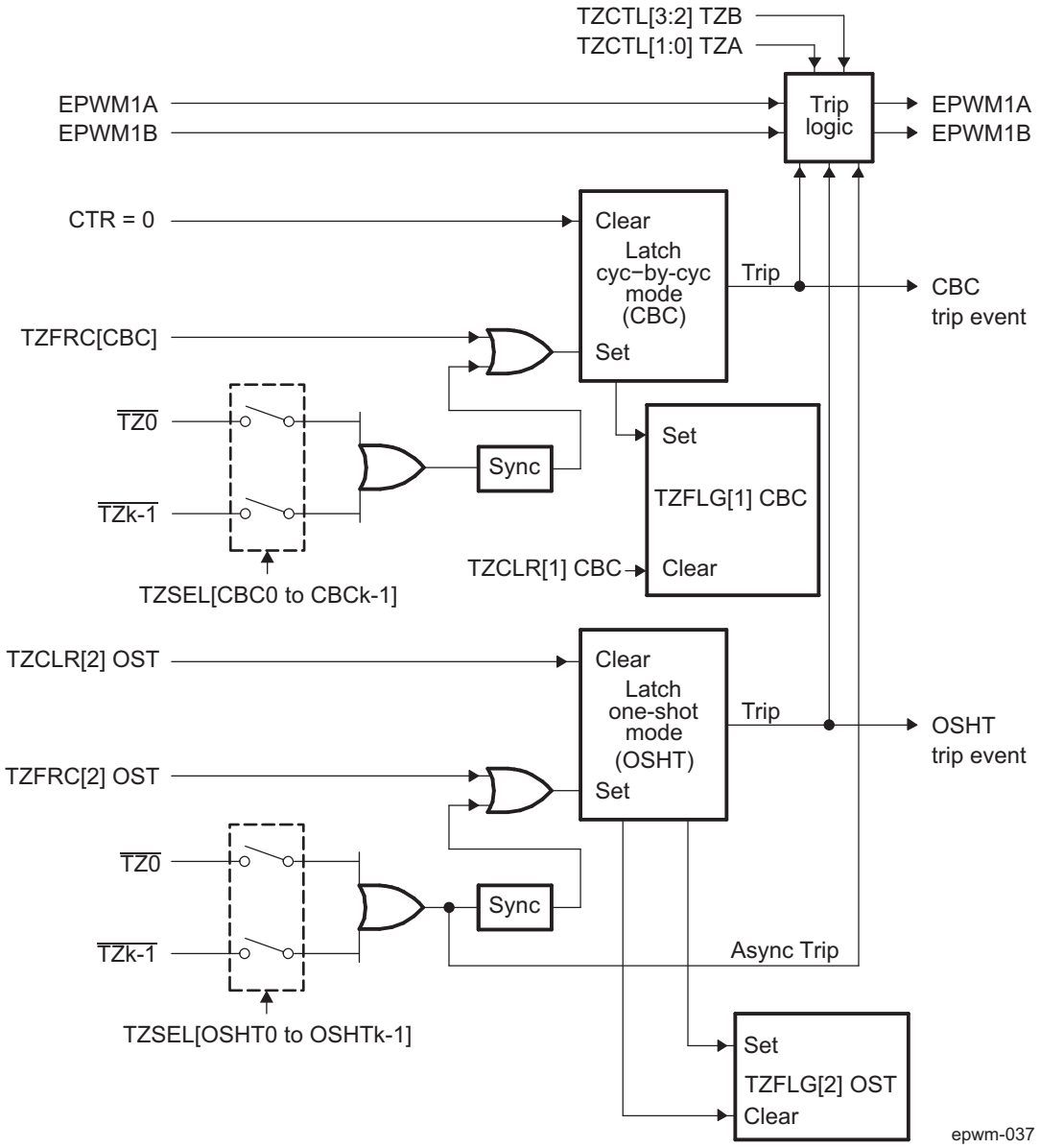
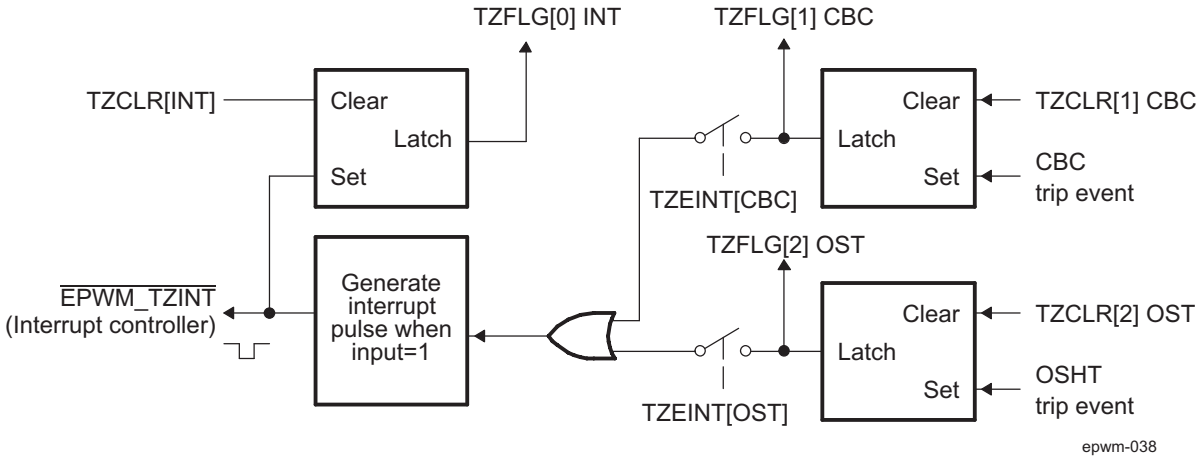


Figure 21-39. ePWM Trip-Zone Submodule Interrupt Logic

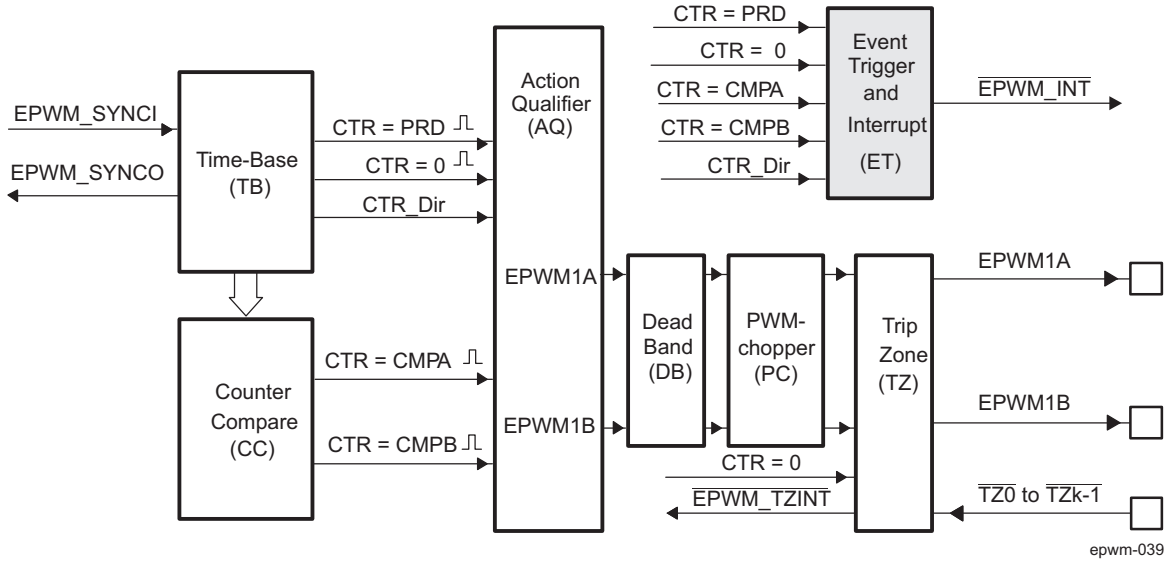


21.2.2.9 ePWM Event-Trigger (ET) Submodule

Figure 21-40 shows the event-trigger (ET) submodule in the ePWM system. The event-trigger submodule manages the events generated by the time-base submodule and the counter-compare submodule to generate an aggregated interrupt request.

NOTE: The ePWM1 ET interrupt request output is further routed via the device IRQ_CROSSBAR, to different device host interrupt controllers, located outside PWMSS. For more details on interrupt event routing outside the ePWM, refer to the Section 21.1.3.

Figure 21-40. ePWM Event-Trigger Submodule



21.2.2.9.1 Purpose of the ePWM Event-Trigger Submodule

The key functions of the event-trigger submodule are:

- Receives event inputs generated by the time-base and counter-compare submodules
- Uses the time-base direction information for up/down event qualification
- Uses prescaling logic to issue interrupt requests at:
  - Every event
  - Every second event
  - Every third event
- Provides full visibility of event generation via event counters and flags

21.2.2.9.2 Controlling and Monitoring the ePWM Event-Trigger Submodule

The key registers used to configure the event-trigger submodule are shown in Table 21-47:

Table 21-47. Event-Trigger Submodule Registers

Acronym	Register Description	Address Offset	Shadowed
EPWM_ETSEL	Event-Trigger Selection Register	32h	No
EPWM_ETPS	Event-Trigger Prescale Register	34h	No
EPWM_ETFLG	Event-Trigger Flag Register	36h	No

Table 21-47. Event-Trigger Submodule Registers (continued)

Acronym	Register Description	Address Offset	Shadowed
EPWM_ETCLR	Event-Trigger Clear Register	38h	No
EPWM_ETFRC	Event-Trigger Force Register	3Ah	No

21.2.2.9.3 Operational Overview of the ePWM Event-Trigger Submodule

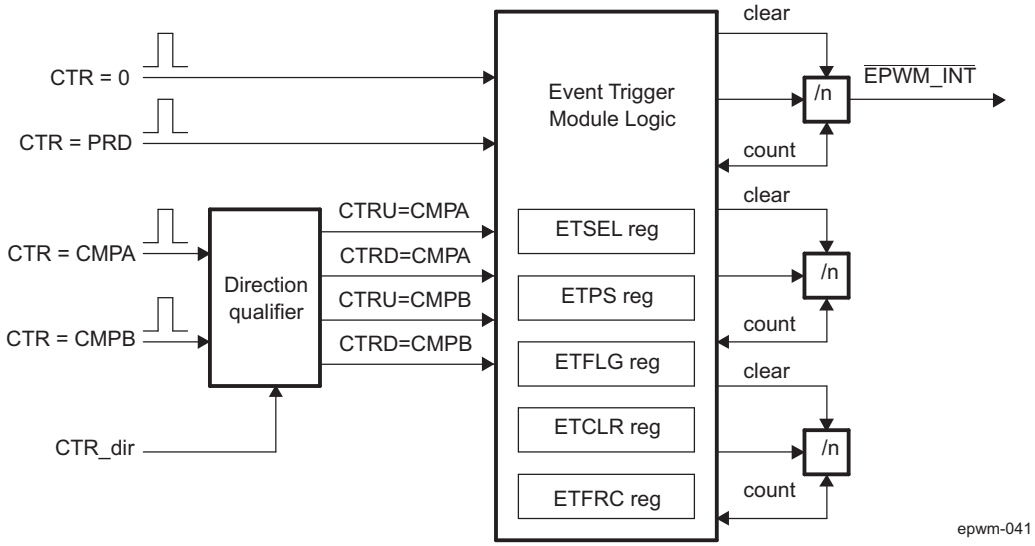
The following sections describe the event-trigger submodule's operational highlights.

The ePWM module has one interrupt request line.

The event-trigger submodule monitors various event conditions (the left side inputs to event-trigger submodule shown in Figure 21-41) and can be configured to prescale these events before issuing an Interrupt request. The event-trigger prescaling logic can issue Interrupt requests at:

- Every event
- Every second event
- Every third event

Figure 21-41. ePWM Event-Trigger Submodule Showing Event Inputs and Prescaled Outputs



- ETSEL—This selects which of the possible events will trigger an interrupt.
- ETPS—This programs the event prescaling options previously mentioned.
- ETFLG—These are flag bits indicating status of the selected and prescaled events.
- ETCLR—These bits allow you to clear the flag bits in the EPWM_ETFLG register via software.
- ETFRC—These bits allow software forcing of an event. Useful for debugging or software intervention.

A more detailed look at how the various register bits interact with the Interrupt is shown in Figure 21-42.

Figure 21-42 shows the event-trigger's interrupt generation logic. The interrupt-period (EPWM_ETPS[1:0] INTPRD) bits specify the number of events required to cause an interrupt pulse to be generated. The choices available are:

- Do not generate an interrupt
- Generate an interrupt on every event
- Generate an interrupt on every second event
- Generate an interrupt on every third event

An interrupt cannot be generated on every fourth or more events.

Which event can cause an interrupt is configured by the interrupt selection (EPWM_ETSEL[2:0] INTSEL) bits. The event can be one of the following:

- Time-base counter equal to zero (EPWM_TBCNT bitfield TBCNT = 0000h).
- Time-base counter equal to period (EPWM_TBCNT bitfield TBCNT = TBPRD value in EPWM_TBPRD active register).
- Time-base counter equal to the compare A register (EPWM_CMPA) when the timer is incrementing.
- Time-base counter equal to the compare A register (EPWM_CMPA) when the timer is decrementing.
- Time-base counter equal to the compare B register (EPWM_CMPB) when the timer is incrementing.
- Time-base counter equal to the compare B register (EPWM_CMPB) when the timer is decrementing.

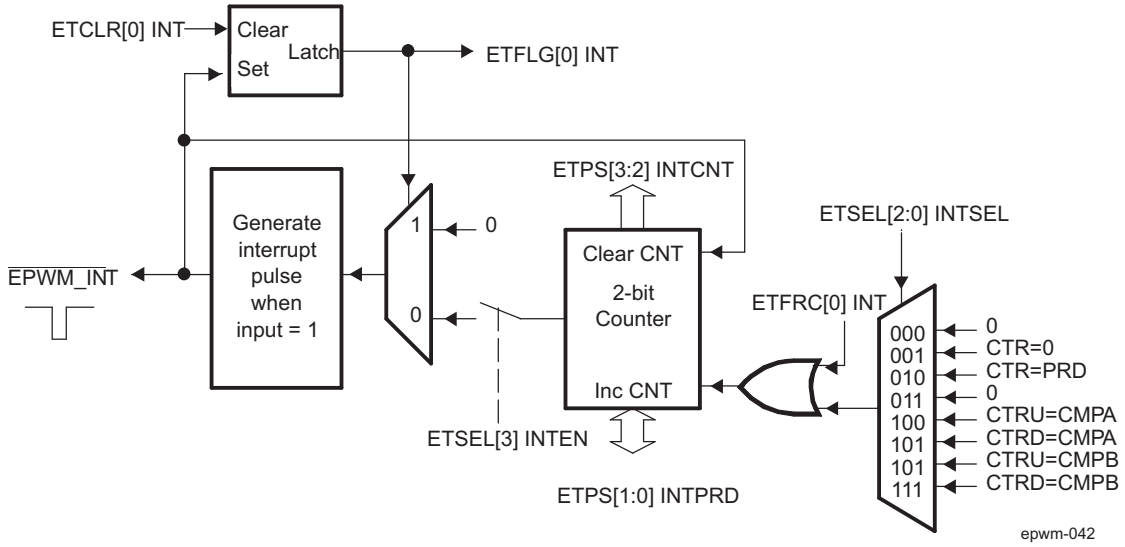
The number of events that have occurred can be read from the interrupt event counter (EPWM_ETPS[3:2] INTCNT) register bits. That is, when the specified event occurs the EPWM_ETPS[3:2] INTCNT bits are incremented until they reach the value specified by EPWM_ETPS[1:0] INTPRD. When EPWM_ETPS[3:2] INTCNT = EPWM_ETPS[1:0] INTPRD the counter stops counting and its output is set. The counter is only cleared when an interrupt is sent to the interrupt controller.

When EPWM_ETPS[3:2] INTCNT reaches EPWM_ETPS[1:0] INTPRD, one of the following behaviors will occur:

- If interrupts are enabled, EPWM_ETSEL[3] INTEN = 1 and the interrupt flag is clear, EPWM_ETFLG[0] INT = 0, then an interrupt pulse is generated and the interrupt flag is set, EPWM_ETFLG[0] INT = 1, and the event counter is cleared EPWM_ETPS[3:2] INTCNT = 0. The counter will begin counting events again.
- If interrupts are disabled, EPWM_ETSEL[3] INTEN = 0, or the interrupt flag is set, EPWM_ETFLG[0] INT = 1, the counter stops counting events when it reaches the period value EPWM_ETPS[3:2] INTCNT = EPWM_ETPS[1:0] INTPRD.
- If interrupts are enabled, but the interrupt flag is already set, then the counter will hold its output high until the ENTFLG[0] INT flag is cleared. This allows for one interrupt to be pending while one is serviced.

Writing to the INTPRD bits will automatically clear the counter INTCNT = 0 and the counter output will be reset (so no interrupts are generated). Writing a 1 to the EPWM_ETFRC[0] INT bit will increment the event counter INTCNT. The counter will behave as described above when INTCNT = INTPRD. When INTPRD = 0, the counter is disabled and hence no events will be detected and the EPWM_ETFRC[0] INT bit is also ignored.

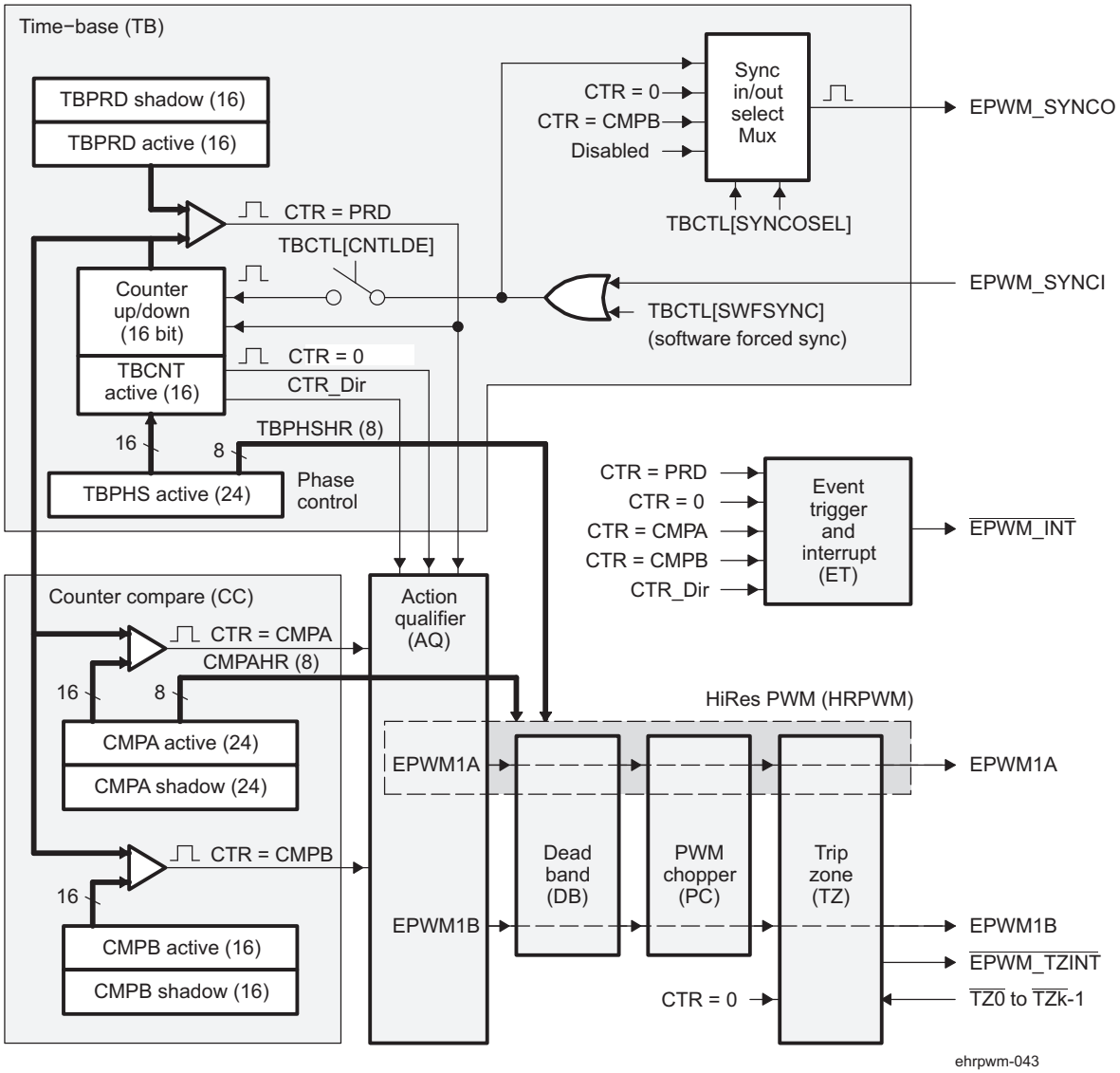
Figure 21-42. ePWM Event-Trigger Interrupt Generator



21.2.2.10 High-Resolution PWM (HRPWM) Submodule

Figure 21-43 shows the high-resolution PWM (HRPWM) submodule in the ePWM system.

Figure 21-43. HRPWM System Interface



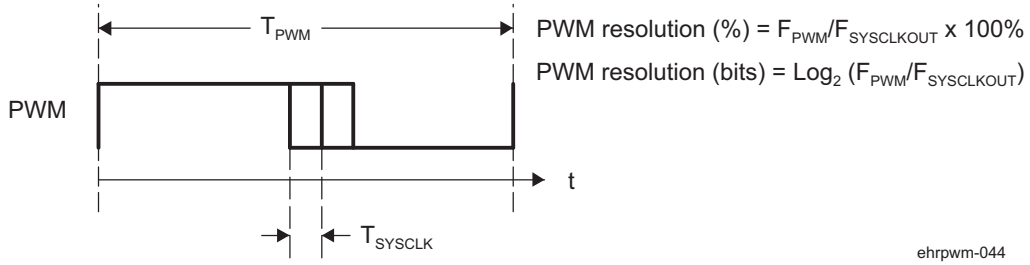
21.2.2.10.1 Purpose of the High-Resolution PWM Submodule

The enhanced high-resolution pulse-width modulator (eHRPWM) extends the time resolution capabilities of the conventionally derived digital pulse-width modulator (PWM). HRPWM is typically used when PWM resolution falls below ~9-10 bits. The key features of HRPWM are:

- Extended time resolution capability
- Used in both duty cycle and phase-shift control methods
- Finer time granularity control or edge positioning using extensions to the Compare A and Phase registers
- Implemented using the A signal path of PWM, that is, on the EPWM1A output. **EPWM1B output has conventional PWM capabilities**

The ePWM peripheral is used to perform a function that is mathematically equivalent to a digital-to-analog converter (DAC). As shown in Figure 21-44, the effective resolution for conventionally generated PWM is a function of PWM frequency (or period) and system clock frequency.

Figure 21-44. Resolution Calculations for Conventionally Generated PWM



If the required PWM operating frequency does not offer sufficient resolution in PWM mode, you may want to consider HRPWM. As an example of improved performance offered by HRPWM, Table 21-48 shows resolution in bits for various PWM frequencies. Table 21-48 values assume a MEP step size of 180 ps. See your device-specific data manual for typical and maximum performance specifications for the MEP.

Table 21-48. Resolution for PWM and HRPWM

	Regular Resolution (PWM)		High Resolution (HRPWM)	
PWM Frequency (kHz)	Bits	%	Bits	%
20	12.3	0.0	18.1	0.000
50	11.0	0.0	16.8	0.001
100	10.0	0.1	15.8	0.002
150	9.4	0.2	15.2	0.003
200	9.0	0.2	14.8	0.004
250	8.6	0.3	14.4	0.005
500	7.6	0.5	13.8	0.007
1000	6.6	1.0	12.4	0.018
1500	6.1	1.5	11.9	0.027
2000	5.6	2.0	11.4	0.036

Although each application may differ, typical low-frequency PWM operation (below 250 kHz) may not require HRPWM. HRPWM capability is most useful for high-frequency PWM requirements of power conversion topologies such as:

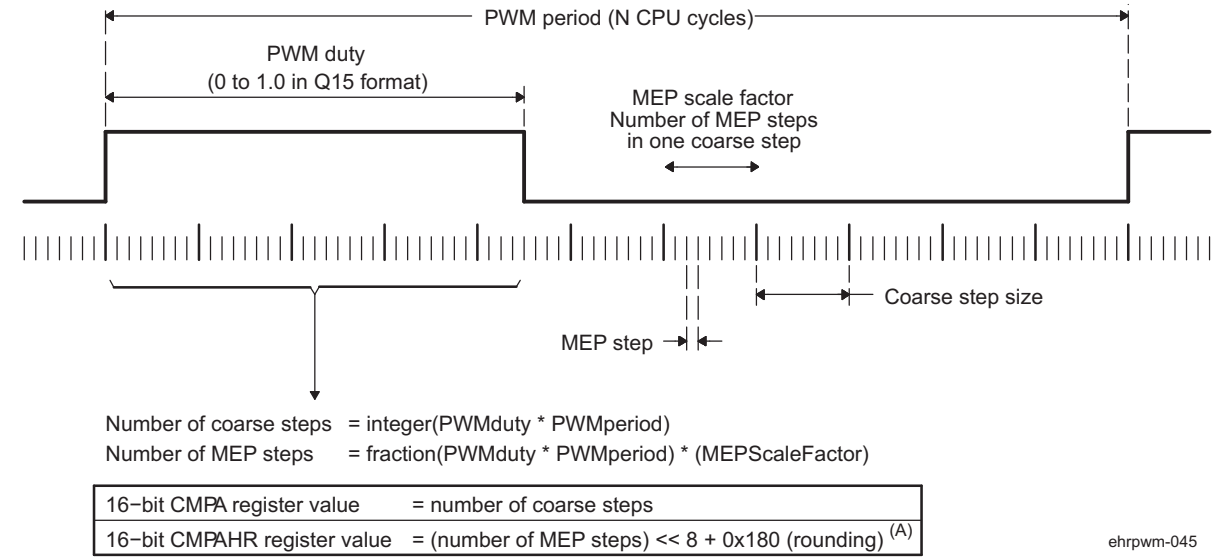
- Single-phase buck, boost, and flyback
- Multi-phase buck, boost, and flyback
- Phase-shifted full bridge
- Direct modulation of D-Class power amplifiers

21.2.2.10.2 Architecture of the High-Resolution PWM Submodule

The HRPWM is based on micro edge positioner (MEP) technology. MEP logic is capable of positioning an edge very finely by sub-dividing one coarse system clock of a conventional PWM generator. The time step accuracy is on the order of 150 ps. The HRPWM also has a self-check software diagnostics mode to check if the MEP logic is running optimally, under all operating conditions.

Figure 21-45 shows the relationship between one coarse system clock and edge position in terms of MEP steps, which are controlled via an 8-bit field in the Compare A extension register (HRPWM_CMPAHR).

Figure 21-45. Operating Logic Using MEP



A For MEP range and rounding adjustment.

To generate an HRPWM waveform, configure the TBM, CCM, and AQM registers as you would to generate a conventional PWM of a given frequency and polarity. The HRPWM works together with the TBM, CCM, and AQM registers to extend edge resolution, and should be configured accordingly. Although many programming combinations are possible, only a few are needed and practical.

21.2.2.10.3 Controlling and Monitoring the High-Resolution PWM Submodule

The MEP of the HRPWM is controlled by two extension registers, each 8-bits wide. These two HRPWM registers are concatenated with the 16-bit EPWM_TBPHS and EPWM_CMPA registers used to control PWM operation.

- HRPWM_TBPHSHR - Time-Base Phase High-Resolution Register
- HRPWM_CMPAHR - Counter-Compare A High-Resolution Register

Table 21-49 lists the registers used to control and monitor the high-resolution PWM submodule.

Table 21-49. HRPWM Submodule Registers

Acronym	Register Description	Address Offset	Shadowed
HRPWM_TBPHSHR	Extension Register for HRPWM Phase	4h	No
HRPWM_CMPAHR	Extension Register for HRPWM Duty	10h	Yes
HRPWM_HRCTL	HRPWM Configuration Register	1040h	No

21.2.2.10.4 Configuring the High-Resolution PWM Submodule

Once the ePWM has been configured to provide conventional PWM of a given frequency and polarity, the HRPWM is configured by programming the HRPWM_HRCTL register located at offset address 1040h. This register provides configuration options for the following key operating modes:

- **Edge Mode:** The MEP can be programmed to provide precise position control on the rising edge (RE), falling edge (FE), or both edges (BE) at the same time. FE and RE are used for power topologies requiring duty cycle control, while BE is used for topologies requiring phase shifting, for example, phase shifted full bridge.
- **Control Mode:** The MEP is programmed to be controlled either from the HRPWM_CMPAHR register (duty cycle control) or the HRPWM_TBPHSHR register (phase control). RE or FE control mode should be used with HRPWM_CMPAHR register. BE control mode should be used with HRPWM_TBPHSHR register.
- **Shadow Mode:** This mode provides the same shadowing (double buffering) option as in regular PWM mode. This option is valid only when operating from the HRPWM_CMPAHR register and should be chosen to be the same as the regular load option for the CMPA register. If HRPWM_TBPHSHR is used, then this option has no effect.

21.2.2.10.5 Operational Highlights for the High-Resolution PWM Submodule

The MEP logic is capable of placing an edge in one of 255 (8 bits) discrete time steps, each of which has a time resolution on the order of 150 ps. The MEP works with the TBM and CCM registers to be certain that time steps are optimally applied and that edge placement accuracy is maintained over a wide range of PWM frequencies, system clock frequencies and other operating conditions. Table 21-50 shows the typical range of operating frequencies supported by the HRPWM.

Table 21-50. Relationship Between MEP Steps, PWM Frequency and Resolution

System (MHz)	MEP Steps Per SYSCLKOUT(1) (2) (3)	PWM Minimum (Hz)(4)	PWM Maximum (MHz)	Resolution at Maximum (Bits)(5)
50.0	111	763	2.50	11.1
60.0	93	916	3.00	10.9
70.0	79	1068	3.50	10.6
80.0	69	1221	4.00	10.4
90.0	62	1373	4.50	10.3
100.0	56	1526	5.00	10.1

(1) System frequency = SYSCLKOUT, that is, CPU clock. TBCLK = SYSCLKOUT
(2) Table data based on a MEP time resolution of 180 ps (this is an example value)
(3) MEP steps applied = $T_{SYSCLKOUT}$/180 ps in this example.
(4) PWM minimum frequency is based on a maximum period value, TBPRD = 65 535. PWM mode is asymmetrical up-count.
(5) Resolution in bits is given for the maximum PWM frequency stated.

21.2.2.10.5.1 HRPWM Edge Positioning

In a typical power control loop (switch modes, digital motor control (DMC), uninterruptible power supply (UPS)), a digital controller (PID, 2pole/2zero, lag/lead, etc.) issues a duty command, usually expressed in a per unit or percentage terms.

In the following example, assume that for a particular operating point, the demanded duty cycle is 0.405 or 40.5% on-time and the required converter PWM frequency is 1.25 MHz. In conventional PWM generation with a system clock of 100 MHz, the duty cycle choices are in the vicinity of 40.5%. In Figure 21-46, a compare value of 32 counts (duty = 40%) is the closest to 40.5% that you can attain. This is equivalent to an edge position of 320 ns instead of the desired 324 ns. This data is shown in Table 21-51.

By utilizing the MEP, you can achieve an edge position much closer to the desired point of 324 ns. Table 21-51 shows that in addition to the CMPA value, 22 steps of the MEP (HRPWM_CMPAHR register) will position the edge at 323.96 ns, resulting in almost zero error. In this example, it is assumed that the MEP has a step resolution of 180 ns.

Figure 21-46. Required PWM Waveform for a Requested Duty = 40.5%

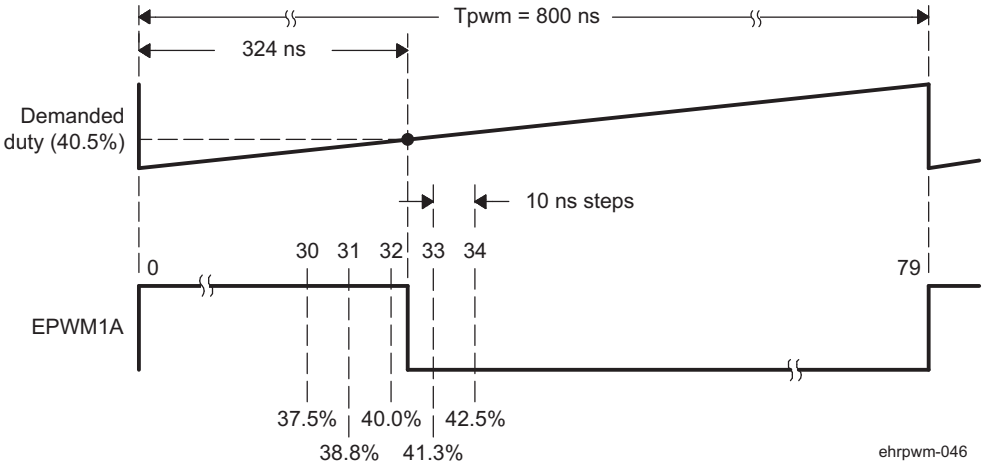


Table 21-51. CMPA vs Duty (left), and [CMPA:CMPAHR] vs Duty (right)

CMPA (count)[1] [2] [3]	DUTY (%)	High Time (ns)	CMPA (count)	CMPAHR (count)	Duty (%)	High Time (ns)
28	35.0	280	32	18	40.405	323.24
29	36.3	290	32	19	40.428	323.42
30	37.5	300	32	20	40.450	323.60
31	38.8	310	32	21	40.473	323.78
32	40.0	320	32	22	40.495	323.96
33	41.3	330	32	23	40.518	324.14
34	42.5	340	32	24	40.540	324.32
			32	25	40.563	324.50
Required			32	26	40.585	324.68
32.40	40.5	324	32	27	40.608	324.86

(1) System clock, SYSCLKOUT and TBCLK = 100 MHz, 10 ns
(2) For a PWM Period register value of 80 counts, PWM Period = 80 × 10 ns = 800 ns, PWM frequency = 1/800 ns = 1.25 MHz
(3) Assumed MEP step size for the above example = 180 ps

21.2.2.10.5.2 HRPWM Scaling Considerations

The mechanics of how to position an edge precisely in time has been demonstrated using the resources of the standard (EPWM_CMPA) and MEP (HRPWM_CMPAHR) registers. In a practical application, however, it is necessary to seamlessly provide the CPU a mapping function from a per-unit (fractional) duty cycle to a final integer (non-fractional) representation that is written to the [CMPA:CMPAHR] register combination.

To do this, first examine the scaling or mapping steps involved. It is common in control software to express duty cycle in a per-unit or percentage basis. This has the advantage of performing all needed math calculations without concern for the final absolute duty cycle, expressed in clock counts or high time in ns. Furthermore, it makes the code more transportable across multiple converter types running different PWM frequencies.

To implement the mapping scheme, a two-step scaling procedure is required.

Assumptions for this example:

System clock, SYSCLKOUT	=	10 ns (100 MHz)
PWM frequency	=	1.25 MHz (1/800 ns)
Required PWM duty cycle, **PWMDuty**	=	0.405 (40.5%)
PWM period in terms of coarse steps, **PWMperiod** (800 ns/10 ns)	=	80
Number of MEP steps per coarse step at 180 ps (10 ns/180 ps), **MEP_SF**	=	55
Value to keep CMPAHR within the range of 1-255 and fractional rounding constant (default value)	=	180h

Step 1: Percentage Integer Duty value conversion for EPWM_CMPA register

EPWM_CMPA register value	=	int(**PWMDuty** × **PWMperiod**); int means integer part
	=	int(0.405 × 80)
	=	int(32.4)
EPWM_CMPA register value	=	32 (20h)

Step 2: Fractional value conversion for HRPWM_CMPAHR register

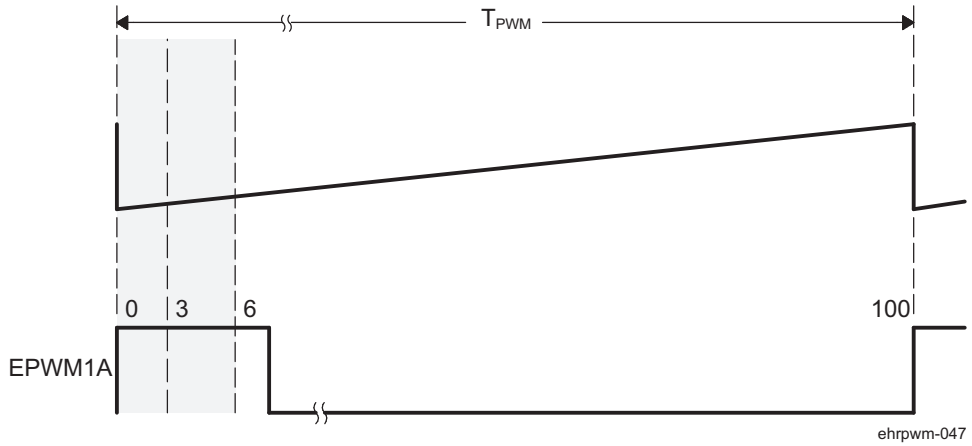
HRPWM_CMPAHR register value	=	(frac(**PWMDuty** × **PWMperiod**) × **MEP_SF**) << 8) + 180h; frac means fractional part
	=	(frac(32.4) × 55 <<8) + 180h; Shift is to move the value as CMPAHR high byte
	=	((0.4 × 55) <<8) + 180h
	=	(22 <<8) + 180h
	=	22 × 256 + 180h ; Shifting left by 8 is the same multiplying by 256.
	=	5632 + 180h
	=	1600h + 180h
HRPWM_CMPAHR value	=	1780h; HRPWM_CMPAHR value = 1700h, lower 8 bits will be ignored by hardware.

21.2.2.10.5.3 HRPWM Duty Cycle Range Limitation

In high resolution mode, the MEP is not active for 100% of the PWM period. It becomes operational 3 SYSCLKOUT cycles after the period starts.

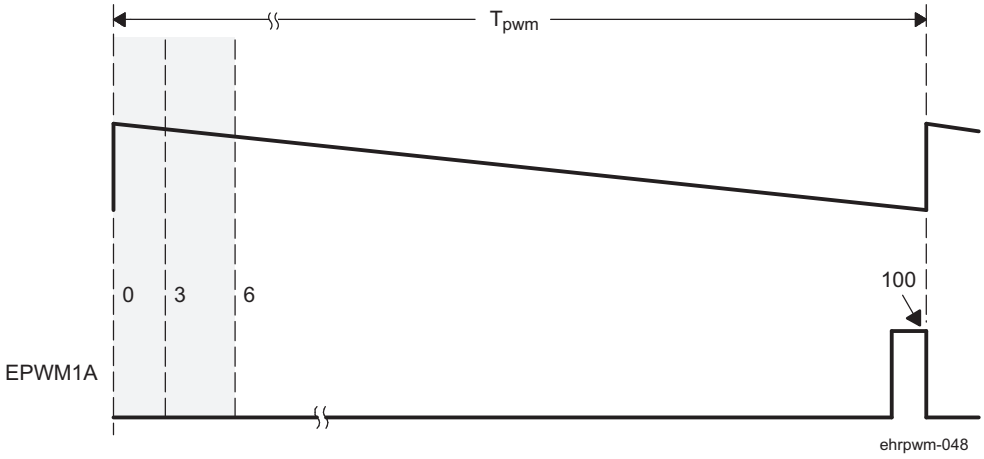
Duty cycle range limitations are illustrated in Figure 21-47. This limitation imposes a lower duty cycle limit on the MEP. For example, precision edge control is not available all the way down to 0% duty cycle. Although for the first 3 or 6 cycles, the HRPWM capabilities are not available, regular PWM duty control is still fully operational down to 0% duty. In most applications this should not be an issue as the controller regulation point is usually not designed to be close to 0% duty cycle.

Figure 21-47. Low % Duty Cycle Range Limitation Example When PWM Frequency = 1 MHz



If the application demands HRPWM operation in the low percent duty cycle region, then the HRPWM can be configured to operate in count-down mode with the rising edge position (REP) controlled by the MEP. This is illustrated in Figure 21-48. In this case low percent duty limitation is no longer an issue.

Figure 21-48. High % Duty Cycle Range Limitation Example when PWM Frequency = 1 MHz



21.2.2.11 eHRPWM Functional Register Groups

The Table 21-52 lists the groups of ePWM and the high-resolution PWM module registers according to their functionalities.

Table 21-52. ePWM / HRPWM Module Control and Status Registers Grouped by Submodule

Register Name	Offset	Size (×16)	Shadow	Register Description
Time-Base Submodule Registers				

Table 21-52. ePWM / HRPWM Module Control and Status Registers Grouped by Submodule (continued)

Register Name	Offset	Size (×16)	Shadow	Register Description
EPWM_TBCTL	0h	1	No	Time-Base Control Register
EPWM_TBSTS	2h	1	No	Time-Base Status Register
EPWM_TBPHS	6h	1	No	Time-Base Phase Register
EPWM_TBCNT	8h	1	No	Time-Base Counter Register
EPWM_TBPRD	Ah	1	Yes	Time-Base Period Register
Counter-Compare Submodule Registers				
EPWM_CMPCTL	Eh	1	No	Counter-Compare Control Register
EPWM_CMPA	12h	1	Yes	Counter-Compare A Register
EPWM_CMPB	14h	1	Yes	Counter-Compare B Register
Action-Qualifier Submodule Registers				
EPWM_AQCTLA	16h	1	No	Action-Qualifier Control Register for Output A (EPWM1A)
EPWM_AQCTLB	18h	1	No	Action-Qualifier Control Register for Output B (EPWM1B)
EPWM_AQSFRC	1Ah	1	No	Action-Qualifier Software Force Register
EPWM_AQCSFRC	1Ch	1	Yes	Action-Qualifier Continuous S/W Force Register Set
Dead-Band Generator Submodule Registers				
EPWM_DBCTL	1Eh	1	No	Dead-Band Generator Control Register
EPWM_DBRED	20h	1	No	Dead-Band Generator Rising Edge Delay Count Register
EPWM_DBFED	22h	1	No	Dead-Band Generator Falling Edge Delay Count Register
Trip-Zone Submodule Registers				
EPWM_TZSEL	24h	1	No	Trip-Zone Select Register
EPWM_TZCTL	28h	1	No	Trip-Zone Control Register
EPWM_TZEINT	2Ah	1	No	Trip-Zone Enable Interrupt Register
EPWM_TZFLG	2Ch	1	No	Trip-Zone Flag Register
EPWM_TZCLR	2Eh	1	No	Trip-Zone Clear Register
EPWM_TZFRC	30h	1	No	Trip-Zone Force Register
Event-Trigger Submodule Registers				
EPWM_ETSEL	32h	1	No	Event-Trigger Selection Register
EPWM_ETPS	34h	1	No	Event-Trigger Pre-Scale Register
EPWM_ETFLG	36h	1	No	Event-Trigger Flag Register
EPWM_ETCLR	38h	1	No	Event-Trigger Clear Register
EPWM_ETFRC	3Ah	1	No	Event-Trigger Force Register
PWM-Chopper Submodule Registers				
EPWM_PCCTL	3Ch	1	No	PWM-Chopper Control Register
High-Resolution PWM (HRPWM) Submodule Registers				
HRPWM_TBPHSHR	4h	1	No	Extension for HRPWM Phase Register
HRPWM_CMPAHR	10h	1	No	Extension for HRPWM Counter-Compare A Register
HRPWM_HRCTL	40h	1	No	HRPWM Control Register

21.2.3 PWMSS_EPWM Register Manual

This section provides description of the device PWMSS ePWM and High Resolution-PWM relevant functional registers.

21.2.3.1 PWMSS_EPWM Instance Summary

Table 21-53. PWMSS_EPWM Instance Summary

Module Name	Module Base Address L4_PER2 Interconnect	Size (Bytes)
PWMSS_EPWM	0x4843 E200	136 Bytes

21.2.3.2 PWMSS_EPWM Registers

21.2.3.2.1 PWMSS_EPWM Register Summary

Table 21-54. PWMSS_EPWM Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	PWMSS_EPWM Physical Address L4_PER2 Interconnect
EPWM_TBCTL	RW	16	0x0000 0000	0x4843 E200
EPWM_TBSTS	RW	16	0x0000 0002	0x4843 E202
HRPWM_TBPHSHR	RW	16	0x0000 0004	0x4843 E204
EPWM_TBPHS	RW	16	0x0000 0006	0x4843 E206
EPWM_TBCNT	RW	16	0x0000 0008	0x4843 E208
EPWM_TBPRD	RW	16	0x0000 000A	0x4843 E20A
EPWM_CMPCTL	RW	16	0x0000 000E	0x4843 E20E
HRPWM_CMPAHR	RW	16	0x0000 0010	0x4843 E210
EPWM_CMPA	RW	16	0x0000 0012	0x4843 E212
EPWM_CMPB	RW	16	0x0000 0014	0x4843 E214
EPWM_AQCTLA	RW	16	0x0000 0016	0x4843 E216
EPWM_AQCTLB	RW	16	0x0000 0018	0x4843 E218
EPWM_AQSFRC	RW	16	0x0000 001A	0x4843 E21A
EPWM_AQCSFRC	RW	16	0x0000 001C	0x4843 E21C
EPWM_DBCTL	RW	16	0x0000 001E	0x4843 E21E
EPWM_DBRED	RW	16	0x0000 0020	0x4843 E220
EPWM_DBFED	RW	16	0x0000 0022	0x4843 E222
EPWM_TZSEL	RW	16	0x0000 0024	0x4843 E224
EPWM_TZCTL	RW	16	0x0000 0028	0x4843 E228
EPWM_TZEINT	RW	16	0x0000 002A	0x4843 E22A
EPWM_TZFLG	R	16	0x0000 002C	0x4843 E22C
EPWM_TZCLR	RW	16	0x0000 002E	0x4843 E22E
EPWM_TZFRC	RW	16	0x0000 0030	0x4843 E230
EPWM_ETSEL	RW	16	0x0000 0032	0x4843 E232
EPWM_ETPS	RW	16	0x0000 0034	0x4843 E234
EPWM_ETFLG	R	16	0x0000 0036	0x4843 E236
EPWM_ETCLR	RW	16	0x0000 0038	0x4843 E238
EPWM_ETFRC	RW	16	0x0000 003A	0x4843 E23A
EPWM_PCCTL	RW	16	0x0000 003C	0x4843 E23C
HRPWM_HRCTL	RW	16	0x0000 00C0	0x4843 E2C0

21.2.3.2.2 PWMSS_EPWM Register Description

Table 21-55. EPWM_TBCTL

Address Offset	0x0000 0000		
Physical Address	0x4843 E200	**Instance**	PWMSS_EPWM
Description			
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FREE_SOFT		PHSDIR	CLKDIV			HSPCLKDIV		SWFSYNC	SYNCOSEL			PRDLD	PHSEN	CTRMODE	

Bits	Field Name	Description	Type	Reset
15:14	FREE_SOFT	Emulation Mode Bits. These bits select the behavior of the ePWM time-base counter during emulation events: 0x0 = Stop after the next time-base counter increment or decrement 0x1 = Stop when counter completes a whole cycle. (a) Up-count mode: stop when the time-base counter = period (EPWM_TBCNT bitfield TBCNT = TBPRD in EPWM_TBPRD active register). (b) Down-count mode: stop when the time-base counter = 0000 (EPWM_TBCNT bitfield TBCNT = 0000h). (c) Up-down-count mode: stop when the time-base counter = 0000 (EPWM_TBCNT bitfield TBCNT = 0000h). 0x2 = Free run 0x3 = Free run	RW	0x0
13	PHSDIR	Phase Direction Bit. This bit is only used when the time-base counter is configured in the up-down-count mode. The PHSDIR bit indicates the direction the time-base counter (EPWM_TBCNT) will count after a synchronization event occurs and a new phase value is loaded from the phase (EPWM_TBPHS) register. This is irrespective of the direction of the counter before the synchronization event.. In the up-count and down-count modes this bit is ignored. 0x0 = Count down after the synchronization event. 0x1 = Count up after the synchronization event.	RW	0x0
12:10	CLKDIV	Time-base Clock Prescale Bits. These bits determine part of the time-base clock prescale value. TBCLK = SYSCLKOUT / (HSPCLKDIV x CLKDIV) 0x0 = /1 (default on reset) 0x1 = /2 0x2 = /4 0x3 = /8 0x4 = /16 0x5 = /32 0x6 = /64 0x7 = /128	RW	0x0

Bits	Field Name	Description	Type	Reset
9:7	HSPCLKDIV	High-Speed Time-base Clock Prescale Bits. These bits determine part of the time-base clock prescale value. TBCLK = SYSCLKOUT / (HSPCLKDIV x CLKDIV). This divisor emulates the HSPCLK in the TMS320x281x system as used on the Event Manager (EV) peripheral. 0x0 = /1 0x1 = /2 (default on reset) 0x2 = /4 0x3 = /6 0x4 = /8 0x5 = /10 0x6 = /12 0x7 = /14	RW	0x0
6	SWFSYNC	Software Forced Synchronization Pulse. 0x0 = Writing a 0 has no effect and reads always return a 0. 0x1 = Writing a 1 forces a one-time synchronization pulse to be generated. This event is ORed with the EPWM_SYNCI input of the ePWM module. SWFSYNC is valid (operates) only when EPWM_SYNCI is selected by SYNCOSEL = 00.	RW	0x0
5:4	SYNCOSEL	Synchronization Output Select. These bits select the source of the EPWM_SYNCO signal. 0x0 = EPWM_SYNC: 0x1 = TBCNT = 0: Time-base counter equal to zero (EPWM_TBCNT bitfield TBCNT= 0000h) 0x2 = TBCNT = CMPB : Time-base counter equal to counter-compare B (EPWM_TBCNT bitfield TBCNT = CMPB bitfield in EPWM_CMPB) 0x3 = Disable EPWM_SYNCO signal	RW	0x0
3	PRDLD	Active Period Register Load From Shadow Register Select 0x0 = The period register (EPWM_TBPRD) is loaded from its shadow register when the time-base counter, TBCNT, is equal to zero. A write or read to the EPWM_TBPRD register accesses the shadow register. 0x1 = Load the EPWM_TBPRD register immediately without using a shadow register. A write or read to the EPWM_TBPRD register directly accesses the active register.	RW	0x0
2	PHSEN	Counter Register Load From Phase Register Enable 0x0 = Do not load the time-base counter (EPWM_TBCNT) from the time-base phase register (EPWM_TBPHS) 0x1 = Load the time-base counter with the phase register when an EPWM_SYNCI input signal occurs or when a software synchronization is forced by the SWFSYNC bit.	RW	0x0
1:0	CTRMODE	Counter Mode. The time-base counter mode is normally configured once and not changed during normal operation. If you change the mode of the counter, the change will take effect at the next TBCLK edge and the current counter value shall increment or decrement from the value before the mode change. These bits set the time-base counter mode of operation as follows: 0x0 = Up-count mode 0x1 = Down-count mode 0x2 = Up-down-count mode 0x3 = Stop-freeze counter operation (default on reset)	RW	0x0

Table 21-56. Register Call Summary for Register EPWM_TBCTL

Enhanced PWM (ePWM) Module

- ePWM Time-Base (TB) Submodule: [0][1][2][3][4][5][6][7]
- ePWM Counter-Compare (CC) Submodule: [8][9]
- ePWM Action-Qualifier (AQ) Submodule: [10][11][12][13][14][15]
- eHRPWM Functional Register Groups: [16]
- PWMSS_EPWM Registers: [17][18][19][20][21][22][23]

Table 21-57. EPWM_TBSTS

Address offset	0x2		
Physical Address	0x4843 E202	**Instance**	PWMSS_EPWM
Description			
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED													CTRMAX	SYNCI	CTRDIR

Bits	Field Name	Description	Type	Reset
15:3	RESERVED		R	0x0000
2	CTRMAX	Time-Base Counter Max Latched Status Bit. 0x0 = Reading a 0 indicates the time-base counter never reached its maximum value. Writing a 0 will have no effect. 0x1 = Reading a 1 on this bit indicates that the time-base counter reached the max value 0xFFFF. Writing a 1 to this bit will clear the latched event.	RW1C	0x0
1	SYNCI	Input Synchronization Latched Status Bit. 0x0 = Writing a 0 will have no effect. Reading a 0 indicates no external synchronization event has occurred. 0x1 = Reading a 1 on this bit indicates that an external synchronization event has occurred (EPWM_SYNCI). Writing a 1 to this bit will clear the latched event.	RW1C	0x0
0	CTRDIR	Time-Base Counter Direction Status Bit. At reset, the counter is frozen, therefore, this bit has no meaning. To make this bit meaningful, you must first set the appropriate mode via EPWM_TBCTL[1:0] CTRMODE. 0x0 = Time-Base Counter is currently counting down. 0x1 = Time-Base Counter is currently counting up.	R	0x0

Table 21-58. Register Call Summary for Register EPWM_TBSTS

Enhanced PWM (ePWM) Module

- ePWM Time-Base (TB) Submodule: [0]
- eHRPWM Functional Register Groups: [1]
- PWMSS_EPWM Registers: [2]

Table 21-59. HRPWM_TBPHSHR

Address offset	0x4		
Physical Address	0x4843 E204	**Instance**	PWMSS_EPWM
Description			
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TBPHSH								RESERVED							

Bits	Field Name	Description	Type	Reset
15:8	TBPHSH	Time-base phase high-resolution bits	RW	0x0
7:0	RESERVED		R	0x0

Table 21-60. Register Call Summary for Register HRPWM_TBPHSHR

Enhanced PWM (ePWM) Module
- ePWM Time-Base (TB) Submodule: [0]
- High-Resolution PWM (HRPWM) Submodule: [1][2][3][4][5]
- eHRPWM Functional Register Groups: [6]
- PWMSS_EPWM Registers: [7]

Table 21-61. EPWM_TBPHS

Address offset	0x6		
Physical Address	0x4843 E206	**Instance**	PWMSS_EPWM
Description			
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TBPHS															

Bits	Field Name	Description	Type	Reset
15:0	TBPHS	These bits set time-base counter phase of the ePWM relative to the time-base that is supplying the synchronization input signal. (a) If EPWM_TBCTL[2] PHSEN = 0, then the synchronization event is ignored and the time-base counter is not loaded with the phase. (b) If EPWM_TBCTL[2] PHSEN = 1, then the time-base counter (TBCNT) will be loaded with the phase (TBPHS) when a synchronization event occurs. The synchronization event can be initiated by the input synchronization signal (EPWM_SYNCI) or by a software forced synchronization.	RW	0x0

Table 21-62. Register Call Summary for Register EPWM_TBPHS

Enhanced PWM (ePWM) Module
- ePWM Time-Base (TB) Submodule: [0][1][2]
- ePWM Action-Qualifier (AQ) Submodule: [3][4][5][6][7][8]
- High-Resolution PWM (HRPWM) Submodule: [9]
- eHRPWM Functional Register Groups: [10]
- PWMSS_EPWM Registers: [11][12][13]

Table 21-63. EPWM_TBCNT

Address offset	0x8		
Physical Address	0x4843 E208	**Instance**	PWMSS_EPWM
Description			
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TBCNT															

Bits	Field Name	Description	Type	Reset
15:0	TBCNT	Reading these bits gives the current time-base counter value. Writing to these bits sets the current time-base counter value. The update happens as soon as the write occurs. The write is NOT synchronized to the time-base clock (TBCLK) and the register is not shadowed.	RW	0x0

Table 21-64. Register Call Summary for Register EPWM_TBCNT

Enhanced PWM (ePWM) Module

- ePWM Time-Base (TB) Submodule: [0][1][2][3][4][5][6][7][8]
- ePWM Counter-Compare (CC) Submodule: [9]
- ePWM Action-Qualifier (AQ) Submodule: [10][11][12][13][14][15]
- ePWM Trip-Zone (TZ) Submodule: [16]
- ePWM Event-Trigger (ET) Submodule: [17][18]
- eHRPWM Functional Register Groups: [19]
- PWMSS_EPWM Registers: [20][21][22][23][24][25][26][27][28][29][30][31][32][33][34]

Table 21-65. EPWM_TBPRD

Address offset	0xA		
Physical Address	0x4843 E20A	**Instance**	PWMSS_EPWM
Description			
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TBPRD															

Bits	Field Name	Description	Type	Reset
15:0	TBPRD	These bits determine the period of the time-base counter. This sets the PWM frequency. Shadowing of this register is enabled and disabled by the EPWM_TBCTL[3] PRDLD bit. By default this register is shadowed. (a) If EPWM_TBCTL[3] PRDLD = 0, then the shadow is enabled and any write or read will automatically go to the shadow register. In this case, the active register will be loaded from the shadow register when the time-base counter equals zero. (b) If EPWM_TBCTL[3] PRDLD = 1, then the shadow is disabled and any write or read will go directly to the active register, that is the register actively controlling the hardware. (c) The active and shadow registers share the same memory map address.	RW	0x0

Table 21-66. Register Call Summary for Register EPWM_TBPRD

Enhanced PWM (ePWM) Module

- ePWM Time-Base (TB) Submodule: [0][1][2][3][4][5][6][7][8][9][10]
- ePWM Action-Qualifier (AQ) Submodule: [11][12][13][14][15][16][17][18][19]
- ePWM Event-Trigger (ET) Submodule: [20]
- eHRPWM Functional Register Groups: [21]
- PWMSS_EPWM Registers: [22][23][24][25][26][27][28][29]

Table 21-67. EPWM_CMPCTL

Address offset	0xE		
Physical Address	0x4843 E20E	**Instance**	PWMSS_EPWM
Description			
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED						SHDWBFULL	SHDWAFULL	RESERVED	SHDWBMODE	RESERVED	SHDWAMODE	LOADBMODE		LOADAMODE	

Bits	Field Name	Description	Type	Reset
15:10	RESERVED		R	0x0
9	SHDWBFULL	Counter-compare B (EPWM_CMPB) Shadow Register Full Status Flag. This bit self clears once a load-strobe occurs. 0x0 = CMPB shadow FIFO not full yet 0x1 = Indicates the CMPB shadow FIFO is full. A CPU write will overwrite current shadow value.	R	0x0
8	SHDWAFULL	Counter-compare A (EPWM_CMPA) Shadow Register Full Status Flag. The flag bit is set when a 32 bit write to CMPA:CMPAHR register or a 16 bit write to EPWM_CMPA register is made. A 16 bit write to HRPWM_CMPAHR register will not affect the flag. This bit self clears once a load-strobe occurs. 0x0 = CMPA shadow FIFO not full yet 0x1 = Indicates the CMPA shadow FIFO is full, a CPU write will overwrite the current shadow value.	R	0x0
7	RESERVED		R	0x0
6	SHDWBMODE	Counter-compare B (EPWM_CMPB) Register Operating Mode. 0x0 = Shadow mode. Operates as a double buffer. All writes via the CPU access the shadow register. 0x1 = Immediate mode. Only the active compare B register is used. All writes and reads directly access the active register for immediate compare action.	RW	0x0
5	RESERVED		R	0x0
4	SHDWAMODE	Counter-compare A (EPWM_CMPA) Register Operating Mode. 0x0 = Shadow mode. Operates as a double buffer. All writes via the CPU access the shadow register. 0x1 = Immediate mode. Only the active compare register is used. All writes and reads directly access the active register for immediate compare action	RW	0x0

Bits	Field Name	Description	Type	Reset
3:2	LOADBMODE	Active Counter-Compare B (CMPB) Load From Shadow Select Mode. This bit has no effect in immediate mode (EPWM_CMPCTL[6] SHDWBMODE = 1). 0x0 = Load on TBCNT = 0: Time-base counter equal to zero (EPWM_TBCNT bitfield TBCNT= 0000h) 0x1 = Load on TBCNT = PRD: Time-base counter equal to period (EPWM_TBCNT bitfield TBCNT = TBPRD bitfield of the active EPWM_TBPRD) 0x2 = Load on either TBCNT = 0 or TBCNT = PRD 0x3 = Freeze (no loads possible)	RW	0x0
1:0	LOADAMODE	Active Counter-Compare A (EPWM_CMPA) Load From Shadow Select Mode. This bit has no effect in immediate mode (EPWM_CMPCTL[4] SHDWAMODE = 1). 0x0 = Load on TBCNT = 0: Time-base counter equal to zero (EPWM_TBCNT bitfield TBCNT = 0000h) 0x1 = Load on TBCNT = PRD: Time-base counter equal to period (EPWM_TBCNT bitfield TBCNT= TBPRD bitfield of the active EPWM_TBPRD) 0x2 = Load on either TBCNT = 0 or TBCNT = PRD 0x3 = Freeze (no loads possible)	RW	0x0

Table 21-68. Register Call Summary for Register EPWM_CMPCTL

Enhanced PWM (ePWM) Module
- ePWM Counter-Compare (CC) Submodule: [0][1][2][3][4][5][6]
- ePWM Action-Qualifier (AQ) Submodule: [7][8][9][10][11][12]
- eHRPWM Functional Register Groups: [13]
- PWMSS_EPWM Registers: [14][15][16][17][18][19][20][21][22][23][24][25][26][27]

Table 21-69. HRPWM_CMPAHR

Address offset	0x10		
Physical Address	0x4843 E210	**Instance**	PWMSS_EPWM
Description			
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMPAHR								RESERVED							

Bits	Field Name	Description	Type	Reset
15:8	CMPAHR	Compare A High-Resolution register bits for MEP step control. A minimum value of 1h is needed to enable HRPWM capabilities. Valid MEP range of operation 1-255h.	RW	0x1
7:0	RESERVED		R	0x0

Table 21-70. Register Call Summary for Register HRPWM_CMPAHR

Enhanced PWM (ePWM) Module
- ePWM Counter-Compare (CC) Submodule: [0]
- High-Resolution PWM (HRPWM) Submodule: [1][2][3][4][5][6][7][8][9][10][11][12]
- eHRPWM Functional Register Groups: [13]
- PWMSS_EPWM Registers: [14][15]

Table 21-71. EPWM_CMPA

Address offset	0x12		
Physical Address	0x4843 E212	**Instance**	PWMSS_EPWM
Description			
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMPA															

Bits	Field Name	Description	Type	Reset
15:0	CMPA	The value in the active EPWM_CMPA register is continuously compared to the time-base counter (register EPWM_TBCNT). When the values are equal, the counter-compare module generates a "time-base counter equal to counter compare A" event. This event is sent to the action-qualifier where it is qualified and converted it into one or more actions. These actions can be applied to either the EPWM1A or the EPWM1B output depending on the configuration of the EPWM_AQCTLA and EPWM_AQCTLB registers. The actions that can be defined in the EPWM_AQCTLA and EPWM_AQCTLB registers include the following. (a) Do nothing the event is ignored. (b) Clear: Pull the EPWM1A and/or EPWM1B signal low. (c) Set: Pull the EPWM1A and/or EPWM1B signal high. (d) Toggle the EPWM1A and/or EPWM1B signal. Shadowing of this register is enabled and disabled by the EPWM_CMPCTL[4] SHDWAMODE bit. By default this register is shadowed. (a) If EPWM_CMPCTL[4] SHDWAMODE = 0, then the shadow is enabled and any write or read will automatically go to the shadow register. In this case, the EPWM_CMPCTL[1:0] LOADAMODE bit field determines which event will load the active register from the shadow register. (b) Before a write, the EPWM_CMPCTL[8] SHDWAFULL bit can be read to determine if the shadow register is currently full. (c) If EPWM_CMPCTL[4] SHDWAMODE = 1, then the shadow register is disabled and any write or read will go directly to the active register, that is the register actively controlling the hardware. (d) In either mode, the active and shadow registers share the same memory map address.	RW	0x0

Table 21-72. Register Call Summary for Register EPWM_CMPA

Enhanced PWM (ePWM) Module

- ePWM Counter-Compare (CC) Submodule: [0][1][2][3][4][5][6][7]
- ePWM Action-Qualifier (AQ) Submodule: [8][9][10][11][12][13][14][15][16][17][18][19][20][21][22][23][24][25][26][27][28]
- ePWM Event-Trigger (ET) Submodule: [29][30]
- High-Resolution PWM (HRPWM) Submodule: [31][32][33][34][35]
- eHRPWM Functional Register Groups: [36]
- PWMSS_EPWM Registers: [37][38][39][40][41][42][43][44][45][46]

Table 21-73. EPWM_CMPB

Address offset	0x14		
Physical Address	0x4843 E214	**Instance**	PWMSS_EPWM
Description			
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CMPB															

Bits	Field Name	Description	Type	Reset
15:0	CMPB	The value in the active EPWM_CMPB register is continuously compared to the time-base counter (register EPWM_TBCNT). When the values are equal, the counter-compare module generates a "time-base counter equal to counter compare B" event. This event is sent to the action-qualifier where it is qualified and converted it into one or more actions. These actions can be applied to either the EPWM1A or the EPWM1B output depending on the configuration of the EPWM_AQCTLA and EPWM_AQCTLB registers. The actions that can be defined in the EPWM_AQCTLA and EPWM_AQCTLB registers include the following. (a) Do nothing, the event is ignored. (b) Clear: Pull the EPWM1A and/or EPWM1B signal low. (c) Set: Pull the EPWM1A and/or EPWM1B signal high. (d) Toggle the EPWM1A and/or EPWM1B signal. Shadowing of this register is enabled and disabled by the EPWM_CMPCTL[6] SHDWBMODE bit. By default this register is shadowed. (a) If EPWM_CMPCTL[6] SHDWBMODE = 0, then the shadow is enabled and any write or read will automatically go to the shadow register. In this case, the EPWM_CMPCTL[3:2] LOADBMODE bit field determines which event will load the active register from the shadow register: (b) Before a write, the EPWM_CMPCTL[9] SHDWBFULL bit can be read to determine if the shadow register is currently full. (c) If EPWM_CMPCTL[6] SHDWBMODE = 1, then the shadow register is disabled and any write or read will go directly to the active register, that is the register actively controlling the hardware. (d) In either mode, the active and shadow registers share the same memory map address.	RW	0x0

Table 21-74. Register Call Summary for Register EPWM_CMPB

Enhanced PWM (ePWM) Module

- ePWM Time-Base (TB) Submodule: [0]
- ePWM Counter-Compare (CC) Submodule: [1][2][3][4][5][6]
- ePWM Action-Qualifier (AQ) Submodule: [7][8][9][10][11][12][13][14][15][16][17][18][19][20][21][22][23][24][25][26]
- ePWM Event-Trigger (ET) Submodule: [27][28]
- eHRPWM Functional Register Groups: [29]
- PWMSS_EPWM Registers: [30][31][32][33][34][35][36][37][38]

Table 21-75. EPWM_AQCTLA

Address offset	0x16		
Physical Address	0x4843 E216	**Instance**	PWMSS_EPWM
Description			
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				CBD		CBU		CAD		CAU		PRD		ZRO	

Bits	Field Name	Description	Type	Reset
15:12	RESERVED		R	0x0
11:10	CBD	Action when the time-base counter equals the active EPWM_CMPB register and the counter is decrementing. 0x0 = Do nothing (action disabled) 0x1 = Clear: force EPWM1A output low. 0x2 = Set: force EPWM1A output high. 0x3 = Toggle EPWM1A output: low output signal will be forced high, and a high signal will be forced low.	RW	0x0

Bits	Field Name	Description	Type	Reset
9:8	CBU	Action when the counter equals the active EPWM_CMPB register and the counter is incrementing. 0x0 = Do nothing (action disabled) 0x1 = Clear: force EPWM1A output low. 0x2 = Set: force EPWM1A output high. 0x3 = Toggle EPWM1A output: low output signal will be forced high, and a high signal will be forced low.	RW	0x0
7:6	CAD	Action when the counter equals the active EPWM_CMPA register and the counter is decrementing. 0x0 = Do nothing (action disabled) 0x1 = Clear: force EPWM1A output low. 0x2 = Set: force EPWM1A output high. 0x3 = Toggle EPWM1A output: low output signal will be forced high, and a high signal will be forced low.	RW	0x0
5:4	CAU	Action when the counter equals the active EPWM_CMPA register and the counter is incrementing. 0x0 = Do nothing (action disabled) 0x1 = Clear: force EPWM1A output low. 0x2 = Set: force EPWM1A output high. 0x3 = Toggle EPWM1A output: low output signal will be forced high, and a high signal will be forced low.	RW	0x0
3:2	PRD	Action when the counter equals the period. Note: By definition, in count up-down mode when the counter equals period the direction is defined as 0 or counting down. 0x0 = Do nothing (action disabled) 0x1 = Clear: force EPWM1A output low. 0x2 = Set: force EPWM1A output high. 0x3 = Toggle EPWM1A output: low output signal will be forced high, and a high signal will be forced low.	RW	0x0
1:0	ZRO	Action when counter equals zero. Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 0x0 = Do nothing (action disabled) 0x1 = Clear: force EPWM1A output low. 0x2 = Set: force EPWM1A output high. 0x3 = Toggle EPWM1A output: low output signal will be forced high, and a high signal will be forced low.	RW	0x0

Table 21-76. Register Call Summary for Register EPWM_AQCTLA

Enhanced PWM (ePWM) Module
- ePWM Action-Qualifier (AQ) Submodule: [0][1][2][3][4][5][6]
- eHRPWM Functional Register Groups: [7]
- PWMSS_EPWM Registers: [8][9][10][11][12]

Table 21-77. EPWM_AQCTLB

Address offset	0x18		
Physical Address	0x4843 E218	**Instance**	PWMSS_EPWM
Description			
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				CBD		CBU		CAD		CAU		PRD		ZRO	

Bits	Field Name	Description	Type	Reset
15:12	RESERVED		R	0x0
11:10	CBD	Action when the counter equals the active EPWM_CMPB register and the counter is decrementing. 0x0 = Do nothing (action disabled) 0x1 = Clear: force EPWM1B output low. 0x2 = Set: force EPWM1B output high. 0x3 = Toggle EPWM1B output: low output signal will be forced high, and a high signal will be forced low.	RW	0x0
9:8	CBU	Action when the counter equals the active EPWM_CMPB register and the counter is incrementing. 0x0 = Do nothing (action disabled) 0x1 = Clear: force EPWM1B output low. 0x2 = Set: force EPWM1B output high. 0x3 = Toggle EPWM1B output: low output signal will be forced high, and a high signal will be forced low.	RW	0x0
7:6	CAD	Action when the counter equals the active EPWM_CMPA register and the counter is decrementing. 0x0 = Do nothing (action disabled) 0x1 = Clear: force EPWM1B output low. 0x2 = Set: force EPWM1B output high. 0x3 = Toggle EPWM1B output: low output signal will be forced high, and a high signal will be forced low.	RW	0x0
5:4	CAU	Action when the counter equals the active EPWM_CMPA register and the counter is incrementing. 0x0 = Do nothing (action disabled) 0x1 = Clear: force EPWM1B output low. 0x2 = Set: force EPWM1B output high. 0x3 = Toggle EPWM1B output: low output signal will be forced high, and a high signal will be forced low.	RW	0x0
3:2	PRD	Action when the counter equals the period. Note: By definition, in count up-down mode when the counter equals period the direction is defined as 0 or counting down. 0x0 = Do nothing (action disabled) 0x1 = Clear: force EPWM1B output low. 0x2 = Set: force EPWM1B output high. 0x3 = Toggle EPWM1B output: low output signal will be forced high, and a high signal will be forced low.	RW	0x0
1:0	ZRO	Action when counter equals zero. Note: By definition, in count up-down mode when the counter equals 0 the direction is defined as 1 or counting up. 0x0 = Do nothing (action disabled) 0x1 = Clear: force EPWM1B output low. 0x2 = Set: force EPWM1B output high. 0x3 = Toggle EPWM1B output: low output signal will be forced high, and a high signal will be forced low.	RW	0x0

Table 21-78. Register Call Summary for Register EPWM_AQCTLB

Enhanced PWM (ePWM) Module

- ePWM Action-Qualifier (AQ) Submodule: [0][1][2][3][4][5][6]
- eHRPWM Functional Register Groups: [7]
- PWMSS_EPWM Registers: [8][9][10][11][12]

Table 21-79. EPWM_AQSFRC

Address offset	0x1A		
Physical Address	0x4843 E21A	**Instance**	PWMSS_EPWM
Description			
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								RLDCSF		OTSFB	ACTSFB		OTSFA	ACTSFA	

Bits	Field Name	Description	Type	Reset
15:8	RESERVED		R	0x0
7:6	RLDCSF	EPWM_AQCSFRC Active Register Reload From Shadow Options. 0x0 = Load on event counter equals zero 0x1 = Load on event counter equals period 0x2 = Load on event counter equals zero or counter equals period 0x3 = Load immediately (the active register is directly accessed by the CPU and is not loaded from the shadow register).	RW	0x0
5	OTSFB	One-Time Software Forced Event on Output B. 0x0 = Writing a 0 (zero) has no effect. Always reads back a 0. This bit is auto cleared once a write to this register is complete, that is, a forced event is initiated. This is a one-shot forced event. It can be overridden by another subsequent event on output B. 0x1 = Initiates a single s/w forced event	RW	0x0
4:3	ACTSFB	Action when One-Time Software Force B Is invoked 0x0 = Does nothing (action disabled) 0x1 = Clear (low) 0x2 = Set (high) 0x3 = Toggle (Low -> High, High -> Low). Note: This action is not qualified by counter direction (CNT_dir)	RW	0x0
2	OTSFA	One-Time Software Forced Event on Output A. 0x0 = Writing a 0 (zero) has no effect. Always reads back a 0. This bit is auto cleared once a write to this register is complete (that is, a forced event is initiated). 0x1 = Initiates a single software forced event.	RW	0x0
1:0	ACTSFA	Action When One-Time Software Force A Is Invoked. 0x0 = Does nothing (action disabled). 0x1 = Clear (low). 0x2 = Set (high). 0x3 = Toggle (Low -> High, High -> Low). Note: This action is not qualified by counter direction (CNT_dir)	RW	0x0

Table 21-80. Register Call Summary for Register EPWM_AQSFRC

Enhanced PWM (ePWM) Module

- ePWM Action-Qualifier (AQ) Submodule: [0][1]
- eHRPWM Functional Register Groups: [2]
- PWMSS_EPWM Registers: [3][4]

Table 21-81. EPWM_AQCSFRC

Address offset	0x1C		
Physical Address	0x4843 E21C	**Instance**	PWMSS_EPWM
Description			
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												CSFB		CSFA	

Bits	Field Name	Description	Type	Reset
15:4	RESERVED		R	0x0
3:2	CSFB	Continuous Software Force on Output B. In immediate mode, a continuous force takes effect on the next TBCLK edge. In shadow mode, a continuous force takes effect on the next TBCLK edge after a shadow load into the active register. To configure shadow mode, use EPWM_AQSFRC[7:6] RLDCSF. 0x0 = Forcing disabled, that is, has no effect 0x1 = Forces a continuous low on output B 0x2 = Forces a continuous high on output B 0x3 = Software forcing is disabled and has no effect	RW	0x0
1:0	CSFA	Continuous Software Force on Output A In immediate mode, a continuous force takes effect on the next TBCLK edge. In shadow mode, a continuous force takes effect on the next TBCLK edge after a shadow load into the active register. 0x0 = Forcing disabled, that is, has no effect 0x1 = Forces a continuous low on output A 0x2 = Forces a continuous high on output A 0x3 = Software forcing is disabled and has no effect	RW	0x0

Table 21-82. Register Call Summary for Register EPWM_AQCSFRC

Enhanced PWM (ePWM) Module

- ePWM Action-Qualifier (AQ) Submodule: [0][1]
- eHRPWM Functional Register Groups: [2]
- PWMSS_EPWM Registers: [3][4]

Table 21-83. EPWM_DBCTL

Address offset	0x1E		
Physical Address	0x4843 E21E	**Instance**	PWMSS_EPWM
Description			
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED										IN_MODE		POLSEL		OUT_MODE	

Bits	Field Name	Description	Type	Reset
15:6	RESERVED		R	0x0
5:4	IN_MODE	Dead Band Input Mode Control. Bit 5 controls the S5 switch and bit 4 controls the S4 switch. This allows you to select the input source to the falling-edge and rising-edge delay. To produce classical dead-band waveforms, the default is EPWM1A In is the source for both falling and rising-edge delays. 0x0 = EPWM1A In (from the action-qualifier) is the source for both falling-edge and rising-edge delay. 0x1 = EPWM1B In (from the action-qualifier) is the source for rising-edge delayed signal. EPWM1A In (from the action-qualifier) is the source for falling-edge delayed signal. 0x2 = EPWM1A In (from the action-qualifier) is the source for rising-edge delayed signal. EPWM1B In (from the action-qualifier) is the source for falling-edge delayed signal. 0x3 = EPWM1B In (from the action-qualifier) is the source for both rising-edge delay and falling-edge delayed signal.	RW	0x0
3:2	POLSEL	Polarity Select Control. Bit 3 controls the S3 switch and bit 2 controls the S2 switch. This allows you to selectively invert one of the delayed signals before it is sent out of the dead-band submodule. The following descriptions correspond to classical upper/lower switch control as found in one leg of a digital motor control inverter. These assume that EPWM_DBCTL[1:0] OUT_MODE = 0b11 and EPWM_DBCTL[5:4] IN_MODE = 0b00. Other enhanced modes are also possible, but not regarded as typical usage modes. 0x0 = Active high (AH) mode. Neither EPWM1A nor EPWM1B is inverted (default). 0x1 = Active low complementary (ALC) mode. EPWM1A is inverted. 0x2 = Active high complementary (AHC). EPWM1B is inverted. 0x3 = Active low (AL) mode. Both EPWM1A and EPWM1B are inverted.	RW	0x0
1:0	OUT_MODE	Dead-band Output Mode Control. Bit 1 controls the S1 switch and bit 0 controls the S0 switch. This allows you to selectively enable or bypass the dead-band generation for the falling-edge and rising-edge delay. 0x0 = Dead-band generation is bypassed for both output signals. In this mode, both the EPWM1A and EPWM1B output signals from the action-qualifier are passed directly to the PWM-chopper submodule. In this mode, the POLSEL and IN_MODE bits have no effect. 0x1 = Disable rising-edge delay. The EPWM1A signal from the action-qualifier is passed straight through to the EPWM1A input of the PWM-chopper submodule. The falling-edge delayed signal is seen on output EPWM1B. The input signal for the delay is determined by EPWM_DBCTL[5:4] IN_MODE. 0x2 = Disable falling-edge delay. The EPWM1B signal from the action-qualifier is passed straight through to the EPWM1B input of the PWM-chopper submodule. The rising-edge delayed signal is seen on output EPWM1A. The input signal for the delay is determined by EPWM_DBCTL[5:4] IN_MODE. 0x3 = Dead-band is fully enabled for both rising-edge delay on output EPWM1A and falling-edge delay on output EPWM1B. The input signal for the delay is determined by EPWM_DBCTL[5:4] IN_MODE.	RW	0x0

Table 21-84. Register Call Summary for Register EPWM_DBCTL

Enhanced PWM (ePWM) Module

- ePWM Dead-Band Generator (DB) Submodule: [0][1][2][3][4][5][6][7]
- eHRPWM Functional Register Groups: [8]
- PWMSS_EPWM Registers: [9][10][11][12][13][14]

Table 21-85. EPWM_DBRED

Address offset	0x20		
Physical Address	0x4843 E220	**Instance**	PWMSS_EPWM
Description			
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DEL							

Bits	Field Name	Description	Type	Reset
15:10	RESERVED		R	0x0
9:0	DEL	Rising Edge Delay Count. 10 bit counter.	RW	0x0

Table 21-86. Register Call Summary for Register EPWM_DBRED

Enhanced PWM (ePWM) Module
- ePWM Dead-Band Generator (DB) Submodule: [0][1][2]
- eHRPWM Functional Register Groups: [3]
- PWMSS_EPWM Registers: [4]

Table 21-87. EPWM_DBFED

Address offset	0x22		
Physical Address	0x4843 E222	**Instance**	PWMSS_EPWM
Description			
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DEL							

Bits	Field Name	Description	Type	Reset
15:10	RESERVED		R	0x0
9:0	DEL	Falling Edge Delay Count. 10 bit counter	RW	0x0

Table 21-88. Register Call Summary for Register EPWM_DBFED

Enhanced PWM (ePWM) Module
- ePWM Dead-Band Generator (DB) Submodule: [0][1][2]
- eHRPWM Functional Register Groups: [3]
- PWMSS_EPWM Registers: [4]

Table 21-89. EPWM_TZSEL

Address offset	0x24		
Physical Address	0x4843 E224	**Instance**	PWMSS_EPWM
Description			
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
OSHTN								RESERVED							CBC0

Bits	Field Name	Description	Type	Reset
15:8	OSHTN	Trip-zone n (TZn) select. One-Shot (OSHT) trip-zone enable/disable. When any of the enabled pins go low, a one-shot trip event occurs for this ePWM module. When the event occurs, the action defined in the EPWM_TZCTL register is taken on the EPWM1A and EPWM1B outputs. The one-shot trip condition remains latched until you clear the condition via the EPWM_TZCLR register. 0x0 = Disable TZn as a one-shot trip source for this ePWM module. 0x1 = Enable TZn as a one-shot trip source for this ePWM module.	RW	0x0
7:1	RESERVED		R	0x00
0	CBC0	Trip-zone 0 (TZ0) select. Cycle-by-Cycle (CBC) trip-zone enable/disable. When any of the enabled pins go low, a cycle-by-cycle trip event occurs for this ePWM module. When the event occurs, the action defined in the EPWM_TZCTL register is taken on the EPWM1A and EPWM1B outputs. A cycle-by-cycle trip condition is automatically cleared when the time-base counter reaches zero. 0x0 = Disable TZ0 as a CBC trip source for this ePWM module. 0x1 = Enable TZ0 as a CBC trip source for this ePWM module.	RW	0x0

Table 21-90. Register Call Summary for Register EPWM_TZSEL

Enhanced PWM (ePWM) Module

- ePWM Trip-Zone (TZ) Submodule: [0][1][2][3][4]
- eHRPWM Functional Register Groups: [5]
- PWMSS_EPWM Registers: [6][7][8]

Table 21-91. EPWM_TZCTL

Address offset	0x28		
Physical Address	0x4843 E228	**Instance**	PWMSS_EPWM
Description			
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												TZB		TZA	

Bits	Field Name	Description	Type	Reset
15:4	RESERVED		R	0x0
3:2	TZB	When a trip event occurs the following action is taken on output EPWM1B. Which trip-zone pins can cause an event is defined in the EPWM_TZSEL register. 0x0 = High impedance (EPWM1B = High-impedance state) 0x1 = Force EPWM1B to a high state 0x2 = Force EPWM1B to a low state 0x3 = Do nothing, no action is taken on EPWM1B.	RW	0x0
1:0	TZA	When a trip event occurs the following action is taken on output EPWM1A. Which trip-zone pins can cause an event is defined in the EPWM_TZSEL register. 0x0 = High impedance (EPWM1A = High-impedance state) 0x1 = Force EPWM1A to a high state 0x2 = Force EPWM1A to a low state 0x3 = Do nothing, no action is taken on EPWM1A.	RW	0x0

Table 21-92. Register Call Summary for Register EPWM_TZCTL

Enhanced PWM (ePWM) Module

- ePWM Trip-Zone (TZ) Submodule: [0][1][2][3][4][5][6][7][8]
- eHRPWM Functional Register Groups: [9]
- PWMSS_EPWM Registers: [10][11][12]

Table 21-93. EPWM_TZEINT

Address offset	0x2A		
Physical Address	0x4843 E22A	**Instance**	PWMSS_EPWM
Description			
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED													OST	CBC	RESERVED

Bits	Field Name	Description	Type	Reset
15:3	RESERVED		R	0x0
2	OST	Trip-zone One-Shot Interrupt Enable 0x0 = Disable one-shot interrupt generation 0x1 = Enable Interrupt generation; a one-shot trip event will cause a ePWM0_TZINT interrupt.	RW	0x0
1	CBC	Trip-zone Cycle-by-Cycle Interrupt Enable 0x0 = Disable cycle-by-cycle interrupt generation. 0x1 = Enable interrupt generation; a cycle-by-cycle trip event will cause an ePWM0_TZINT interrupt.	RW	0x0
0	RESERVED		R	0x0

Table 21-94. Register Call Summary for Register EPWM_TZEINT

Enhanced PWM (ePWM) Module

- ePWM Trip-Zone (TZ) Submodule: [0][1][2]
- eHRPWM Functional Register Groups: [3]
- PWMSS_EPWM Registers: [4]

Table 21-95. EPWM_TZFLG

Address offset	0x2C		
Physical Address	0x4843 E22C	**Instance**	PWMSS_EPWM
Description			
Type	R		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED													OST	CBC	INT

Bits	Field Name	Description	Type	Reset
15:3	RESERVED		R	0x0
2	OST	Latched Status Flag for A One-Shot Trip Event. 0x0 = No one-shot trip event has occurred. 0x1 = Indicates a trip event has occurred on a pin selected as a one-shot trip source. This bit is cleared by writing the appropriate value to the EPWM_TZCLR register.	R	0x0
1	CBC	Latched Status Flag for Cycle-By-Cycle Trip Event 0x0 = No cycle-by-cycle trip event has occurred. 0x1 = Indicates a trip event has occurred on a pin selected as a cycle-by-cycle trip source. The EPWM_TZFLG[1] CBC bit will remain set until it is manually cleared by the user. If the cycle-by-cycle trip event is still present when the CBC bit is cleared, then CBC will be immediately set again. The specified condition on the pins is automatically cleared when the ePWM time-base counter reaches zero (EPWM_TBCNT bitfield TBCNT = 0000h) if the trip condition is no longer present. The condition on the pins is only cleared when the TBCNT = 0000h no matter where in the cycle the CBC flag is cleared. This bit is cleared by writing the appropriate value to the EPWM_TZCLR register.	R	0x0
0	INT	Latched Trip Interrupt Status Flag 0x0 = Indicates no interrupt has been generated. 0x1 = Indicates an ePWM0_TZINT interrupt was generated because of a trip condition. No further ePWM0_TZINT interrupts will be generated until this flag is cleared. If the interrupt flag is cleared when either CBC or OST is set, then another interrupt pulse will be generated. Clearing all flag bits will prevent further interrupts. This bit is cleared by writing the appropriate value to the EPWM_TZCLR register.	R	0x0

Table 21-96. Register Call Summary for Register EPWM_TZFLG

Enhanced PWM (ePWM) Module

- ePWM Trip-Zone (TZ) Submodule: [0][1][2][3][4]
- eHRPWM Functional Register Groups: [5]
- PWMSS_EPWM Registers: [6][7][8][9][10][11]

Table 21-97. EPWM_TZCLR

Address offset	0x2E		
Physical Address	0x4843 E22E	**Instance**	PWMSS_EPWM
Description			
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED													OST	CBC	INT

Bits	Field Name	Description	Type	Reset
15:3	RESERVED		R	0x0
2	OST	Clear Flag for One-Shot Trip (OST) Latch 0x0 = Has no effect. Always reads back a 0. 0x1 = Clears this Trip (set) condition.	RW	0x0
1	CBC	Clear Flag for Cycle-By-Cycle (CBC) Trip Latch 0x0 = Has no effect. Always reads back a 0. 0x1 = Clears this Trip (set) condition.	RW	0x0

Bits	Field Name	Description	Type	Reset
0	INT	Global Interrupt Clear Flag 0x0 = Has no effect. Always reads back a 0. 0x1 = Clears the trip-interrupt flag for this ePWM module (EPWM_TZFLG[0] INT). Note: No further ePWM0_TZINT interrupts will be generated until the flag is cleared. If the EPWM_TZFLG[0] INT bit is cleared and any of the other flag bits are set, then another interrupt pulse will be generated. Clearing all flag bits will prevent further interrupts.	RW	0x0

Table 21-98. Register Call Summary for Register EPWM_TZCLR

Enhanced PWM (ePWM) Module
- ePWM Trip-Zone (TZ) Submodule: [0][1][2]
- eHRPWM Functional Register Groups: [3]
- PWMSS_EPWM Registers: [4][5][6][7][8]

Table 21-99. EPWM_TZFRC

Address offset	0x30		
Physical Address	0x4843 E230	**Instance**	PWMSS_EPWM
Description			
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED													OST	CBC	RESERVED

Bits	Field Name	Description	Type	Reset
15:3	RESERVED		R	0x0
2	OST	Force a One-Shot Trip Event via Software 0x0 = Writing of 0 is ignored. Always reads back a 0. 0x1 = Forces a one-shot trip event and sets the EPWM_TZFLG[2] OST bit.	RW	0x0
1	CBC	Force a Cycle-by-Cycle Trip Event via Software 0x0 = Writing of 0 is ignored. Always reads back a 0. 0x1 = Forces a cycle-by-cycle trip event and sets the EPWM_TZFLG[1] CBC bit.	RW	0x0
0	RESERVED		R	0x0

Table 21-100. Register Call Summary for Register EPWM_TZFRC

Enhanced PWM (ePWM) Module
- ePWM Trip-Zone (TZ) Submodule: [0]
- eHRPWM Functional Register Groups: [1]
- PWMSS_EPWM Registers: [2]

Table 21-101. EPWM_ETSEL

Address offset	0x32		
Physical Address	0x4843 E232	**Instance**	PWMSS_EPWM
Description			
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												INTEN	INTSEL		

Bits	Field Name	Description	Type	Reset
15:4	RESERVED		R	0x0
3	INTEN	Enable ePWM Interrupt (ePWM0INT) Generation 0x0 = Disable ePWM0INT generation 0x1 = Enable ePWM0INT generation	RW	0x0
2:0	INTSEL	ePWM Interrupt (ePWM0INT) Selection Options 0x0 = Reserved 0x1 = Enable event time-base counter equal to zero. (TBCNT = 0000h) 0x2 = Enable event time-base counter equal to period (TBCNT = TBPRD) 0x3 = Reserved 0x4 = Enable event time-base counter equal to CMPA when the timer is incrementing. 0x5 = Enable event time-base counter equal to CMPA when the timer is decrementing. 0x6 = Enable event: time-base counter equal to CMPB when the timer is incrementing. 0x7 = Enable event: time-base counter equal to CMPB when the timer is decrementing.	RW	0x0

Table 21-102. Register Call Summary for Register EPWM_ETSEL

Enhanced PWM (ePWM) Module

- ePWM Event-Trigger (ET) Submodule: [0][1][2][3]
- eHRPWM Functional Register Groups: [4]
- PWMSS_EPWM Registers: [5][6][7][8][9][10]

Table 21-103. EPWM_ETPS

Address offset	0x34		
Physical Address	0x4843 E234	**Instance**	PWMSS_EPWM
Description			
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												INTCNT		INTPRD	

Bits	Field Name	Description	Type	Reset
15:4	RESERVED		R	0x0
3:2	INTCNT	ePWM Interrupt Event (ePWM0INT) Counter Register. These bits indicate how many selected EPWM_ETSEL[2:0] INTSEL events have occurred. These bits are automatically cleared when an interrupt pulse is generated. If interrupts are disabled, EPWM_ETSEL[0] INT = 0 or the interrupt flag is set, EPWM_ETFLG[0] INT = 1, the counter will stop counting events when it reaches the period value EPWM_ETPS[3:2] INTCNT = EPWM_ETPS[1:0] INTPRD. 0x0 = No events have occurred. 0x1 = 1 event has occurred. 0x2 = 2 events have occurred. 0x3 = 3 events have occurred.	R	0x0
1:0	INTPRD	ePWM Interrupt (ePWM0INT) Period Select. These bits determine how many selected EPWM_ETSEL[2:0] INTSEL events need to occur before an interrupt is generated. To be generated, the interrupt must be enabled (EPWM_ETSEL[0] INT = 1). If the interrupt status flag is set from a previous interrupt (EPWM_ETFLG[0] INT = 1) then no interrupt will be generated until the flag is cleared via the EPWM_ETCLR[0] INT bit. This allows for one interrupt to be pending while another is still being serviced. Once the interrupt is generated, the EPWM_ETPS[3:2] INTCNT bits will automatically be cleared. Writing a INTPRD value that is the same as the current counter value will trigger an interrupt if it is enabled and the status flag is clear. Writing a INTPRD value that is less than the current counter value will result in an undefined state. If a counter event occurs at the same instant as a new zero or non-zero INTPRD value is written, the counter is incremented. 0x0 = Disable the interrupt event counter. No interrupt will be generated and EPWM_ETFRC[0] INT is ignored. 0x1 = Generate an interrupt on the first event INTCNT = 01 (first event) 0x2 = Generate interrupt on EPWM_ETPS[3:2] INTCNT = 0b10 (second event) 0x3 = Generate interrupt on EPWM_ETPS[3:2] INTCNT = 0b11 (third event)	RW	0x0

Table 21-104. Register Call Summary for Register EPWM_ETPS

Enhanced PWM (ePWM) Module

- ePWM Event-Trigger (ET) Submodule: [0][1][2][3][4][5][6][7][8][9][10][11]
- eHRPWM Functional Register Groups: [12]
- PWMSS_EPWM Registers: [13][14][15][16][17][18]

Table 21-105. EPWM_ETFLG

Address offset	0x36		
Physical Address	0x4843 E236	**Instance**	PWMSS_EPWM
Description			
Type	R		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED															INT

Bits	Field Name	Description	Type	Reset
15:1	RESERVED		R	0x0
0	INT	Latched ePWM Interrupt (EPWM_INT) Status Flag 0x0 = Indicates no event occurred 0x1 = Indicates that an ePWM interrupt (EWPM_INT) was generated. No further interrupts will be generated until the flag bit is cleared. Up to one interrupt can be pending while the EPWM_ETFLG[0] INT bit is still set. If an interrupt is pending, it will not be generated until after the EPWM_ETFLG[0] INT bit is cleared.	R	0x0

Table 21-106. Register Call Summary for Register EPWM_ETFLG

Enhanced PWM (ePWM) Module
- ePWM Event-Trigger (ET) Submodule: [0][1][2][3][4]
- eHRPWM Functional Register Groups: [5]
- PWMSS_EPWM Registers: [6][7][8][9][10][11]

Table 21-107. EPWM_ETCLR

Address offset	0x38		
Physical Address	0x4843 E238	**Instance**	PWMSS_EPWM
Description			
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED															INT

Bits	Field Name	Description	Type	Reset
15:1	RESERVED		R	0x0
0	INT	ePWM Interrupt (ePWM0INT) Flag Clear Bit 0x0 = Writing a 0 has no effect. Always reads back a 0. 0x1 = Writing 1 clears the EPWM_ETFLG[0] INT flag bit and enable further interrupts pulses to be generated.	RW	0x0

Table 21-108. Register Call Summary for Register EPWM_ETCLR

Enhanced PWM (ePWM) Module
- ePWM Event-Trigger (ET) Submodule: [0]
- eHRPWM Functional Register Groups: [1]
- PWMSS_EPWM Registers: [2][3]

Table 21-109. EPWM_ETFRC

Address offset	0x3A		
Physical Address	0x4843 E23A	**Instance**	PWMSS_EPWM
Description			
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED															INT

Bits	Field Name	Description	Type	Reset
15:1	RESERVED		R	0x0
0	INT	INT Force Bit. The interrupt will only be generated if the event is enabled in the EPWM_ETSEL register. The INT flag bit will be set regardless. 0x0 = Writing 0 to this bit will be ignored. Always reads back a 0. 0x1 = Writing 1 generates an interrupt on ePWM0INT and set the INT flag bit. This bit is used for test purposes.	RW	0x0

Table 21-110. Register Call Summary for Register EPWM_ETFRC

Enhanced PWM (ePWM) Module

- ePWM Event-Trigger (ET) Submodule: [0][1][2]
- eHRPWM Functional Register Groups: [3]
- PWMSS_EPWM Registers: [4][5]

Table 21-111. EPWM_PCCTL

Address offset	0x3C		
Physical Address	0x4843 E23C	**Instance**	PWMSS_EPWM
Description			
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED					CHPDUTY			CHPFREQ			OSHTWTH				CHPEN

Bits	Field Name	Description	Type	Reset
15:11	RESERVED		R	0x0
10:8	CHPDUTY	Chopping Clock Duty Cycle 0x0 = Duty = 1/8 (12.5%) 0x1 = Duty = 2/8 (25.0%) 0x2 = Duty = 3/8 (37.5%) 0x3 = Duty = 4/8 (50.0%) 0x4 = Duty = 5/8 (62.5%) 0x5 = Duty = 6/8 (75.0%) 0x6 = Duty = 7/8 (87.5%) 0x7 = Reserved.	RW	0x0
7:5	CHPFREQ	Chopping Clock Frequency 0x0 = Divide by 1 (no prescale). 0x1 = Divide by 2. 0x2 = Divide by 3. 0x3 = Divide by 4. 0x4 = Divide by 5. 0x5 = Divide by 6. 0x6 = Divide by 7. 0x7 = Divide by 8.	RW	0x0
4:1	OSHTWTH	One-Shot Pulse Width 0x0 = 1 - SYSCLKOUT/8 wide 0x1 = 2 - SYSCLKOUT/8 wide 0x2 = 3 - SYSCLKOUT/8 wide 0x3 = 4 - SYSCLKOUT/8 wide 0xF = 16 - SYSCLKOUT/8 wide	RW	0x0

Bits	Field Name	Description	Type	Reset
0	CHPEN	PWM-chopping Enable 0x0 = Disable (bypass) PWM chopping function 0x1 = Enable chopping function	RW	0x0

Table 21-112. Register Call Summary for Register EPWM_PCCTL

Enhanced PWM (ePWM) Module

- PWM-Chopper (PC) Submodule: [0][1]
- eHRPWM Functional Register Groups: [2]
- PWMSS_EPWM Registers: [3]

Table 21-113. HRPWM_HRCTL

Address offset	0xC0		
Physical Address	0x4843 E2C0	**Instance**	PWMSS_EPWM
Description			
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												PULSESEL	DELBUSSEL	DELMODE	

Bits	Field Name	Description	Type	Reset
15:4	RESERVED		R	0x0
3	PULSESEL	Pulse select bits. Selects which pulse to use for timing events in the HRPWM module. Note: The user needs to select the pulse to match the selection in the EPWM module. If TBPHSHR bus is selected, then CNT_zero pulse should be used. If COMPAHR bus is selected, then it should match the bit setting of the EPWM_CMPCTL[LOADMODE] bits in the EPWM module as follows. 0: CNT_zero pulse. 1h: PRD_eq pulse. 2h: CNT_zero or PRD_eq (should not use with HRPWM). 3h: No loads (should not use with HRPWM). 0x0 = Select CNT_zero pulse 0x1 = Select PRD_eq pulse	RW	0x0
2	DELBUSSEL	Delay Bus Select Bit: Selects which bus is used to select the delay for the PWM pulse. 0x0 = Select CMPAHR(8) bus from compare module of EPWM (default on reset). 0x1 = Select TBPHSHR(8) bus from time base module.	RW	0x0
1:0	DELMODE	Delay Mode Bits: Selects which edge of the PWM pulse the delay is inserted. Note: When DELMODE = 0b00, the HRCALM[CALMODE] bits are ignored and the delay line is in by-pass mode. Additionally, DLYIN is connected to CALIN and a continuous low value is fed to the delay line to minimize activity in the module. 0x0 = No delay inserted (default on reset) 0x1 = Delay inserted rising edge 0x2 = Delay inserted falling edge 0x3 = Delay inserted on both edges	RW	0x0

Table 21-114. Register Call Summary for Register HRPWM_HRCTL

Enhanced PWM (ePWM) Module

- High-Resolution PWM (HRPWM) Submodule: [0][1]
- eHRPWM Functional Register Groups: [2]
- PWMSS_EPWM Registers: [3]

21.3 Enhanced Capture (eCAP) Module

21.3.1 eCAP Overview

21.3.1.1 Purpose of the eCAP Peripheral

Uses for eCAP include:

- Sample rate measurements of audio inputs
- Speed measurements of rotating machinery (for example, toothed sprockets sensed via Hall sensors)
- Elapsed time measurements between position sensor pulses
- Period and duty cycle measurements of pulse train signals
- Decoding current or voltage amplitude derived from duty cycle encoded current/voltage sensors

21.3.1.2 eCAP Features

The eCAP module includes the following features:

- 32-bit time base counter
- 4-event time-stamp registers (each 32 bits)
- Edge polarity selection for up to four sequenced time-stamp capture events
- Interrupt on either of the four events
- Single shot capture of up to four event time-stamps
- Continuous mode capture of time-stamps in a four-deep circular buffer
- Absolute time-stamp capture
- Difference (Delta) mode time-stamp capture
- All above resources dedicated to a single input pin
- When not used in capture mode, the ECAP module can be configured as a single channel PWM output

21.3.2 eCAP Functional Description

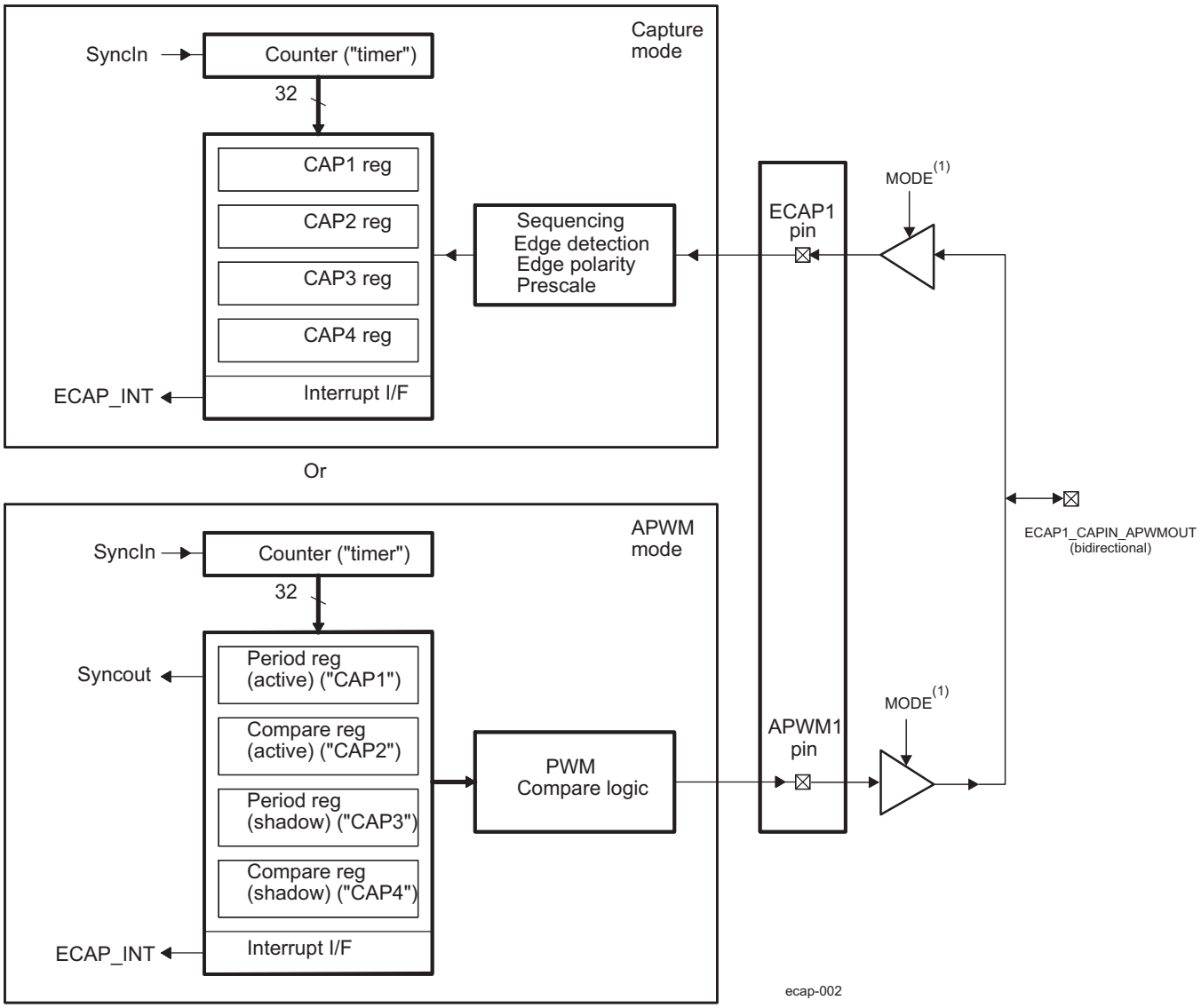
The eCAP module represents one complete capture channel that can be instantiated multiple times depending on the target device. In the context of this guide, one eCAP channel has the following independent key resources:

- Dedicated input capture pin
- 32-bit time base counter
- 4 × 32-bit time-stamp capture registers (PWMSS_ECAP_CAP1-PWMSS_ECAP_CAP4)
- 4-stage sequencer (Modulo4 counter) that is synchronized to external events, ECAP pin rising/falling edges.
- Independent edge polarity (rising/falling edge) selection for all 4 events
- Input capture signal prescaling (from 2-62)
- One-shot compare register (2 bits) to freeze captures after 1 to 4 time-stamp events
- Control for continuous time-stamp captures using a 4-deep circular buffer (PWMSS_ECAP_CAP1-PWMSS_ECAP_CAP4) scheme
- Interrupt capabilities on any of the 4 capture events

21.3.2.1 Capture and APWM Operating Mode

The eCAP module resources can be used to implement a single-channel PWM generator (with 32 bit capabilities) when it is not being used for input captures. The counter operates in count-up mode, providing a time-base for asymmetrical pulse width modulation (PWM) waveforms. The PWMSS_ECAP_CAP1 and PWMSS_ECAP_CAP2 registers become the active period and compare registers, respectively, while PWMSS_ECAP_CAP3 and PWMSS_ECAP_CAP4 registers become the period and capture shadow registers, respectively. Figure 21-49 is a high-level view of both the capture and auxiliary pulse-width modulator (APWM) modes of operation.

Figure 21-49. Capture and APWM Modes of Operation



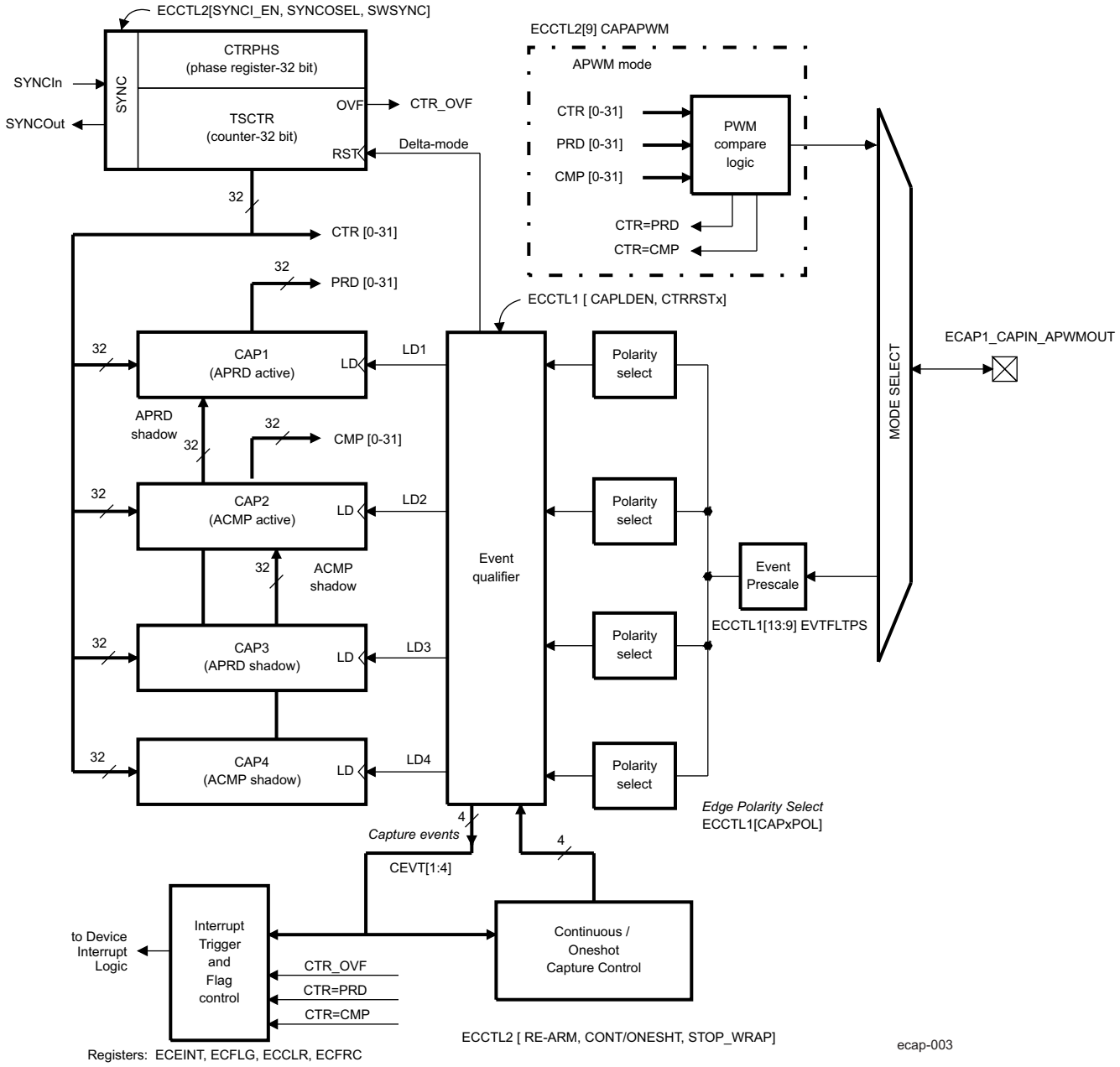
(1) A single pin is shared between CAP and APWM functions. In capture mode, it is an input; in APWM mode, it is an output.

(2) In APWM mode, writing any value to PWMSS_ECAP_CAP1/PWMSS_ECAP_CAP2 active registers also writes the same value to the corresponding shadow registers PWMSS_ECAP_CAP3/PWMSS_ECAP_CAP4. This emulates immediate mode. Writing to the shadow registers PWMSS_ECAP_CAP3/PWMSS_ECAP_CAP4 invokes the shadow mode.

21.3.2.2 eCAP Capture Mode Description

Figure 21-50 shows the various components that implement the capture function.

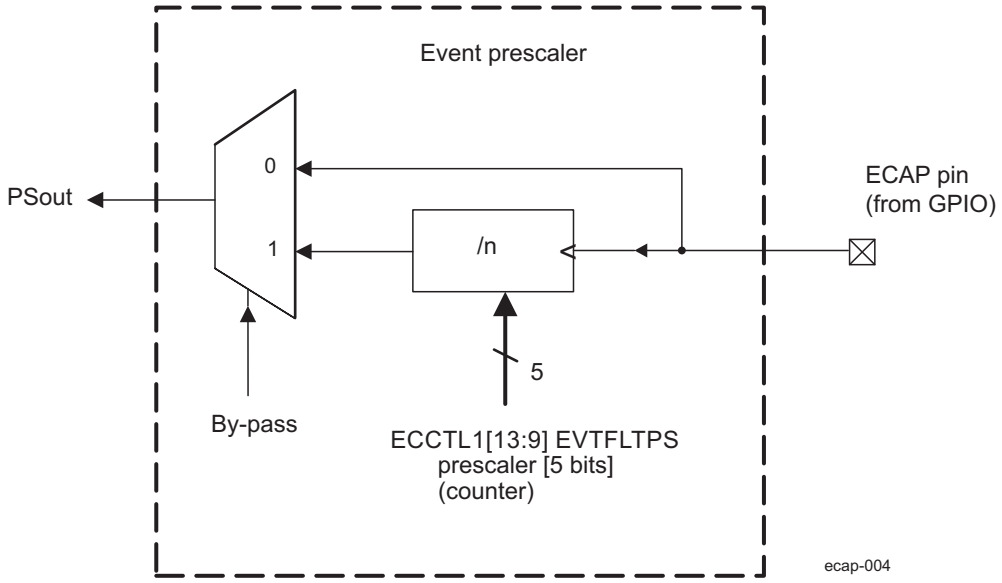
Figure 21-50. Capture Function Diagram



21.3.2.2.1 eCAP Event Prescaler

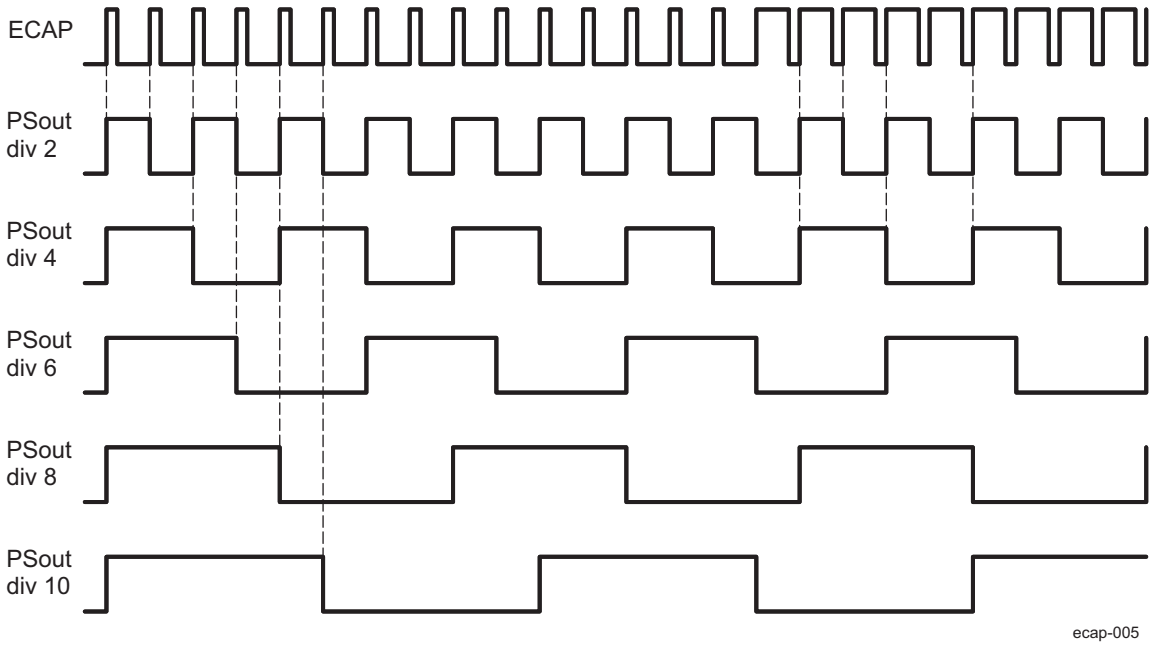
An input capture signal (pulse train) can be prescaled by N = 2-62 (in multiples of 2) or can bypass the prescaler. This is useful when very high frequency signals are used as inputs. Figure 21-51 shows a functional diagram and Figure 21-52 shows the operation of the prescale function.

Figure 21-51. Event Prescale Control



(1) When a prescale value of 1 is chosen (PWMSS_ECAP_ECCTL1[13:9] EVTFLTPS = 0b0000) the input capture signal by-passes the prescale logic completely.

Figure 21-52. Prescale Function Waveforms



21.3.2.2.2 eCAP Edge Polarity Select and Qualifier

- Four independent edge polarity (rising edge/falling edge) selection multiplexers are used, one for each capture event.
- Each edge (up to 4) is event qualified by the Modulo4 sequencer.
- The edge event is gated to its respective CAP*n* register by the Mod4 counter. The CAP*n* register is loaded on the falling edge.

21.3.2.2.3 eCAP Continuous/One-Shot Control

- The Mod4 (2 bit) counter is incremented via edge qualified events (CEVT1-CEVT4).
- The Mod4 counter continues counting (0->1->2->3->0) and wraps around unless stopped.
- A 2-bit stop register is used to compare the Mod4 counter output, and when equal stops the Mod4 counter and inhibits further loads of the PWMSS_ECAP_CAP1-PWMSS_ECAP_CAP4 registers. This occurs during one-shot operation.

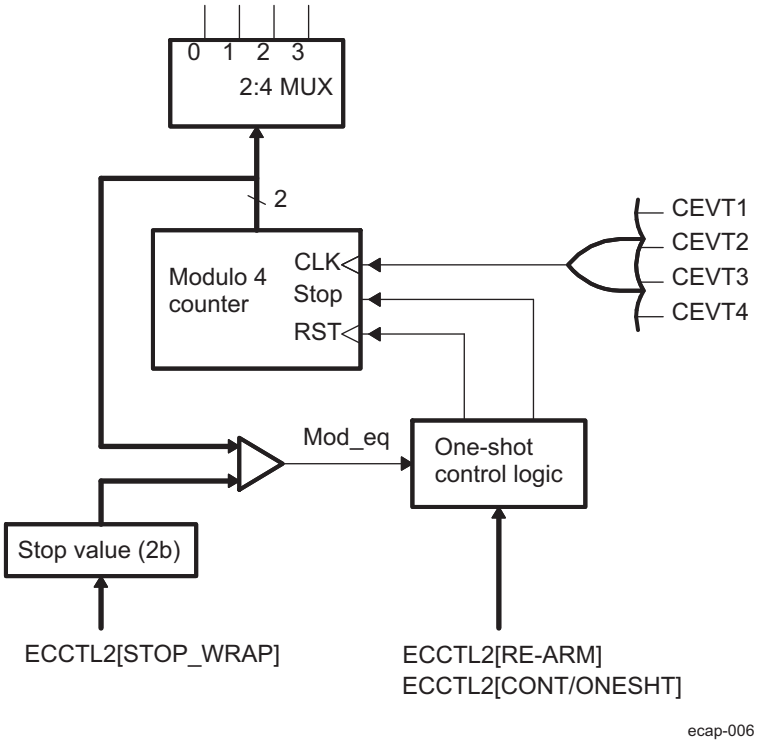
The continuous/one-shot block (Figure 21-53) controls the start/stop and reset (zero) functions of the Mod4 counter via a mono-shot type of action that can be triggered by the stop-value comparator and re-armed via software control.

Once armed, the eCAP module waits for 1-4 (defined by stop-value) capture events before freezing both the Mod4 counter and contents of PWMSS_ECAP_CAP1-4 registers (time-stamps).

Re-arming prepares the eCAP module for another capture sequence. Also re-arming clears (to zero) the Mod4 counter and permits loading of PWMSS_ECAP_CAP1-4 registers again, providing the CAPLDEN bit is set.

In continuous mode, the Mod4 counter continues to run (0->1->2->3->0, the one-shot action is ignored, and capture values continue to be written to PWMSS_ECAP_CAP1-4 in a circular buffer sequence.

Figure 21-53. eCAP Continuous/One-shot Block Diagram



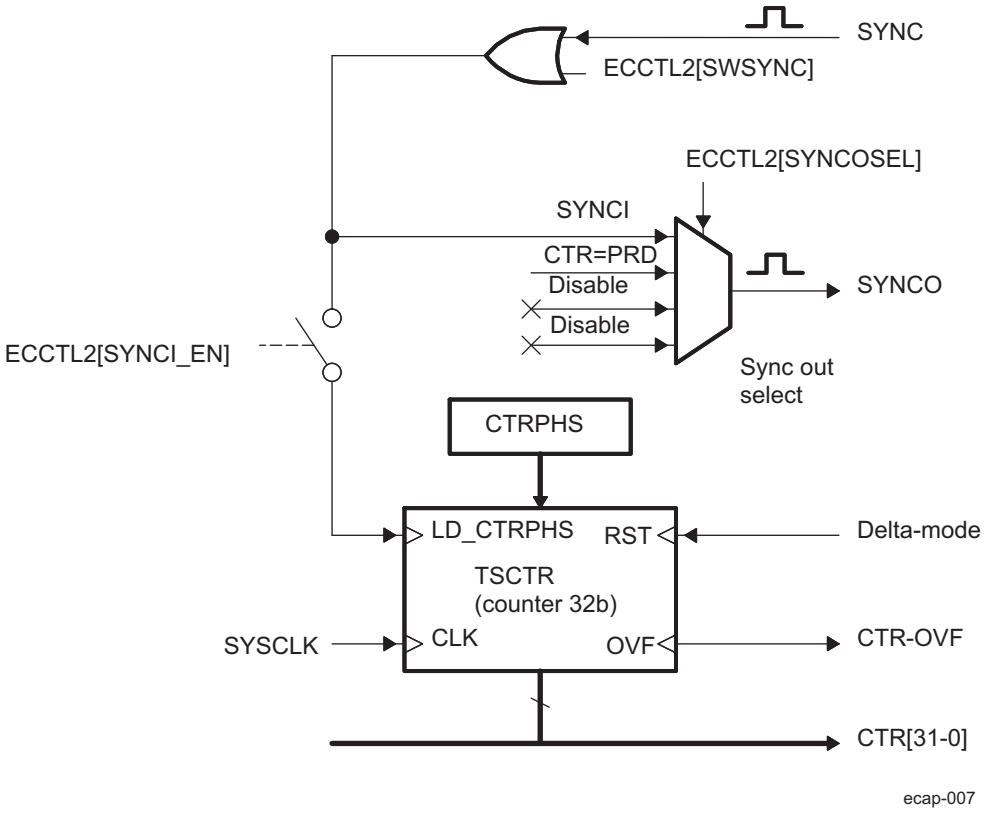
21.3.2.2.4 eCAP 32-Bit Counter and Phase Control

This counter (Figure 21-54) provides the time-base for event captures, and is clocked via the system clock.

A phase register is provided to achieve synchronization with other counters, via a hardware and software forced sync.

On any of the four event loads, an option to reset the 32-bit counter is given. This is useful for time difference capture. The 32-bit counter value is captured first, then it is reset to 0 by any of the LD1-LD4 signals.

Figure 21-54. eCAP Counter and Synchronization Block Diagram



21.3.2.2.5 CAP1-CAP4 Registers

These 32-bit registers are fed by the 32-bit counter timer bus, CTR[0-31] and are loaded (capture a time-stamp) when their respective LD inputs are strobed.

Loading of the capture registers can be inhibited via control bit CAPLDEN. During one-shot operation, this bit is cleared (loading is inhibited) automatically when a stop condition occurs, StopValue = Mod4.

PWMSS_ECAP_CAP1 and PWMSS_ECAP_CAP2 registers become the active period and compare registers, respectively, in APWM mode.

PWMSS_ECAP_CAP3 and PWMSS_ECAP_CAP4 registers become the respective shadow registers (APRD and ACMP) for PWMSS_ECAP_CAP1 and PWMSS_ECAP_CAP2 during APWM operation.

21.3.2.2.6 eCAP Interrupt Control

An Interrupt can be generated on capture events (CEVT1-CEVT4, CNTOVF) or APWM events (TSCNT = PRD, TSCNT = CMP). See Figure 21-55.

A counter overflow event (FFFF FFFFh->0000 0000h) is also provided as an interrupt source (CNTOVF).

The capture events are edge and sequencer qualified (that is, ordered in time) by the polarity select and Mod4 gating, respectively.

One of these events can be selected as the interrupt source (from the eCAP*n* module) going to the interrupt controller.

Seven interrupt events (CEVT1, CEVT2, CEVT3, CEVT4, CNTOVF, TSCNT = PRD, TSCNT = CMP) can be generated. The interrupt enable register (PWMSS_ECAP_ECEINT) is used to enable/disable individual interrupt event sources. The interrupt flag register (PWMSS_ECAP_ECFLG) indicates if any interrupt event has been latched and contains the global interrupt flag bit (INT). An interrupt pulse is generated to the interrupt controller only if any of the interrupt events are enabled, the flag bit is 1, and the INT flag bit is 0. The interrupt service routine must clear the global interrupt flag bit and the serviced event via the interrupt clear register (PWMSS_ECAP_ECCLR) before any other interrupt pulses are generated. You can force an interrupt event via the interrupt force register (PWMSS_ECAP_ECFRC). This is useful for test purposes.

21.3.2.2.7 eCAP Shadow Load and Lockout Control

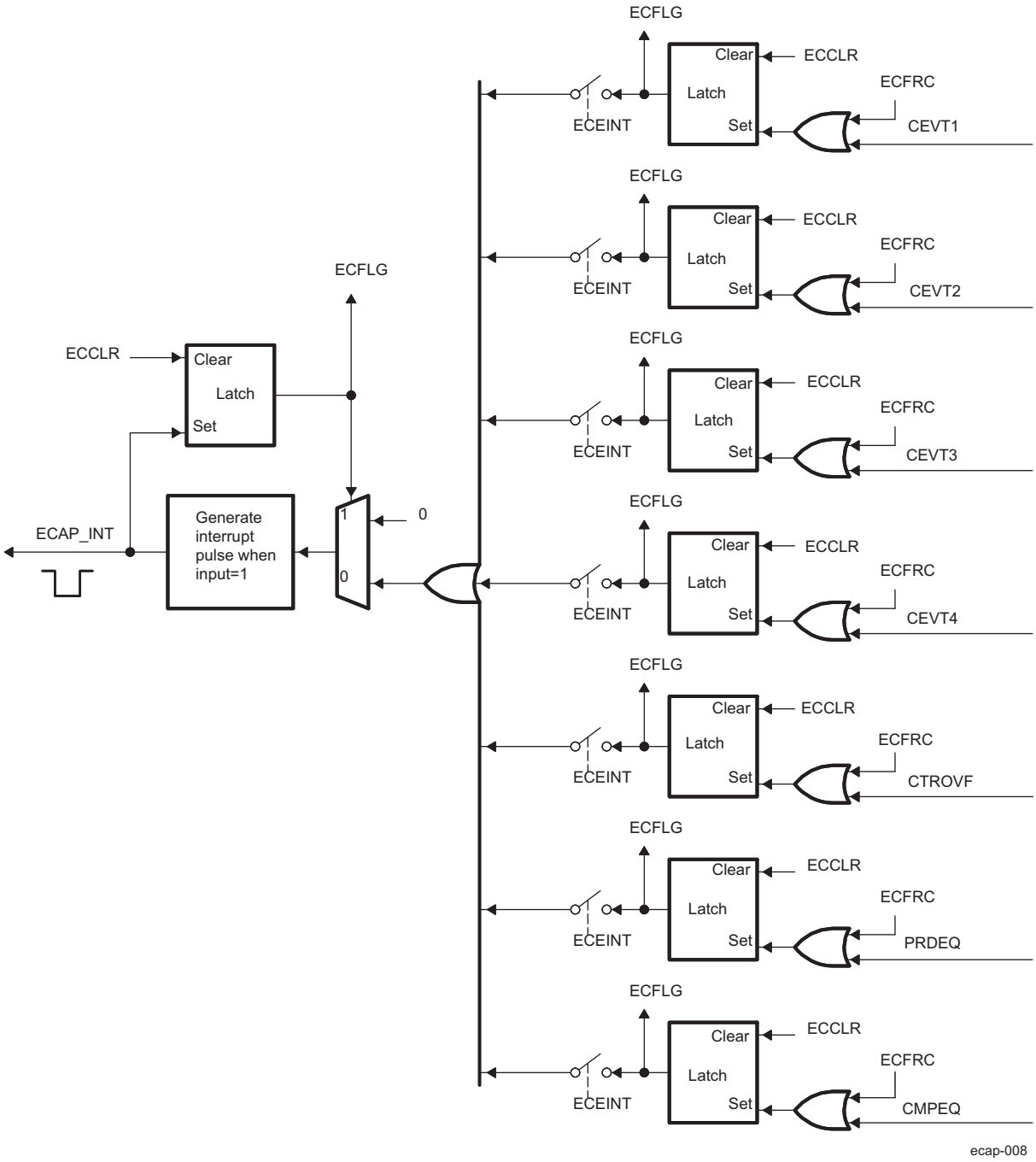
In capture mode, this logic inhibits (locks out) any shadow loading of PWMSS_ECAP_CAP1 or PWMSS_ECAP_CAP2 from APRD and ACMP registers, respectively.

In APWM mode, shadow loading is active and two choices are permitted:

- Immediate - APRD or ACMP are transferred to PWMSS_ECAP_CAP1 or PWMSS_ECAP_CAP2 immediately upon writing a new value.
- On period equal, CTR[31:0] = PRD[31:0]

NOTE: The CEVT1, CEVT2, CEVT3, CEVT4 flags are only active in capture mode (PWMSS_ECAP_ECCTL2[9] CAPAPWM == 0). The TSCNT = PRD, TSCNT = CMP flags are only valid in APWM mode (PWMSS_ECAP_ECCTL2[9] CAPAPWM == 1). CNTOVF flag is valid in both modes.

Figure 21-55. Interrupts in eCAP Module

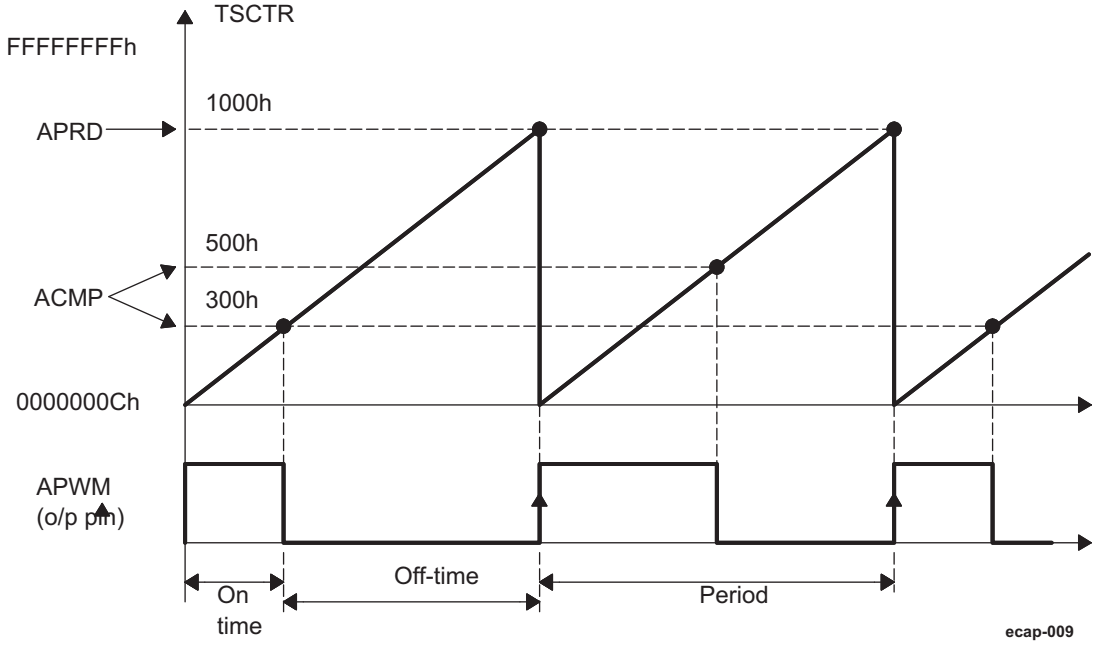


21.3.2.2.8 eCAP Module APWM Mode Operation

Main operating highlights of the APWM section:

- The time-stamp counter bus is made available for comparison via 2 digital (32-bit) comparators.
- When PWMSS_ECAP_CAP1/2 registers are not used in capture mode, their contents can be used as Period and Compare values in APWM mode.
- Double buffering is achieved via shadow registers APRD and ACMP (PWMSS_ECAP_CAP3/4). The shadow register contents are transferred over to PWMSS_ECAP_CAP1/2 registers either immediately upon a write, or on a TSCNT = PRD trigger.
- In APWM mode, writing to PWMSS_ECAP_CAP1/PWMSS_ECAP_CAP2 active registers will also write the same value to the corresponding shadow registers PWMSS_ECAP_CAP3/PWMSS_ECAP_CAP4. This emulates immediate mode. Writing to the shadow registers PWMSS_ECAP_CAP3/PWMSS_ECAP_CAP4 will invoke the shadow mode.
- During initialization, you must write to the active registers for both period and compare. This automatically copies the initial values into the shadow values. For subsequent compare updates, during run-time, you only need to use the shadow registers.

Figure 21-56. PWM Waveform Details Of eCAP APWM Mode Operation



The behavior of APWM active-high mode (APWMPOL == 0) is:

```
CMP = 0x00000000, output low for duration of period (0% duty)
CMP = 0x00000001, output high 1 cycle
CMP = 0x00000002, output high 2 cycles
CMP = PERIOD, output high except for 1 cycle (<100% duty)
CMP = PERIOD+1, output high for complete period (100% duty)
CMP > PERIOD+1, output high for complete period
```

The behavior of APWM active-low mode (APWMPOL == 1) is:

```
CMP = 0x00000000, output high for duration of period (0% duty)
CMP = 0x00000001, output low 1 cycle
CMP = 0x00000002, output low 2 cycles
CMP = PERIOD, output low except for 1 cycle (<100% duty)
```

```
CMP = PERIOD+1, output low for complete period (100% duty)
CMP > PERIOD+1, output low for complete period
```

21.3.2.3 Summary of eCAP Functional Registers

Table 21-115 shows the eCAP module control and status register set. All 32-bit registers are aligned on even address boundaries and are organized in little-endian mode. The 16 least-significant bits of a 32-bit register are located on lowest address (even address).

NOTE: In APWM mode, writing to PWMSS_ECAP_CAP1/PWMSS_ECAP_CAP2 active registers also writes the same value to the corresponding shadow registers PWMSS_ECAP_CAP3/PWMSS_ECAP_CAP4. This emulates immediate mode. Writing to the shadow registers PWMSS_ECAP_CAP3/PWMSS_ECAP_CAP4 invokes the shadow mode.

Table 21-115. eCAP Control and Status Functional Registers

Offset	Register Name	Description	Size (×16)
0h	PWMSS_ECAP_TSCNT	Time-Stamp Counter Register	2
4h	PWMSS_ECAP_CNTPHS	Counter Phase Offset Value Register	2
8h	PWMSS_ECAP_CAP1	Capture 1 Register	2
Ch	PWMSS_ECAP_CAP2	Capture 2 Register	2
10h	PWMSS_ECAP_CAP3	Capture 3 Register	2
14h	PWMSS_ECAP_CAP4	Capture 4 Register	2
28h	PWMSS_ECAP_ECCTL1	Capture Control Register 1	1
2Ah	PWMSS_ECAP_ECCTL2	Capture Control Register 2	1
2Ch	PWMSS_ECAP_ECEINT	Capture Interrupt Enable Register	1
2Eh	PWMSS_ECAP_ECFLG	Capture Interrupt Flag Register	1
30h	PWMSS_ECAP_ECCLR	Capture Interrupt Clear Register	1
32h	PWMSS_ECAP_ECFRC	Capture Interrupt Force Register	1
5Ch	PWMSS_ECAP_PID	Revision ID Register	2

21.3.3 PWMSS_ECAP Register Manual

This section provides description of the PWMSS eCAP relevant functional registers.

21.3.3.1 PWMSS_ECAP Instance Summary

Table 21-116. PWMSS_ECAP Instance Summary

Module Name	Module Base Address L4_PER2 Interconnect	Size (Bytes)
PWMSS_ECAP	0x4843 E100	400 Bytes

21.3.3.2 PWMSS_ECAP Registers

21.3.3.2.1 PWMSS_ECAP Register Summary

Table 21-117. PWMSS_ECAP Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	PWMSS_ECAP Physical Address L4_PER2 Interconnect
PWMSS_ECAP_TSCNT	RW	32	0x0	0x4843 E100

Table 21-117. PWMSS_ECAP Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	PWMSS_ECAP Physical Address L4_PER2 Interconnect
PWMSS_ECAP_CNTPHS	RW	32	0x4	0x4843 E104
PWMSS_ECAP_CAP1	RW	32	0x8	0x4843 E108
PWMSS_ECAP_CAP2	RW	32	0xC	0x4843 E10C
PWMSS_ECAP_CAP3	RW	32	0x10	0x4843 E110
PWMSS_ECAP_CAP4	RW	32	0x14	0x4843 E114
PWMSS_ECAP_ECCTL1	RW	16	0x28	0x4843 E128
PWMSS_ECAP_ECCTL2	RW	16	0x2A	0x4843 E12A
PWMSS_ECAP_ECEINT	RW	16	0x2C	0x4843 E12C
PWMSS_ECAP_ECFLG	R	16	0x2E	0x4843 E12E
PWMSS_ECAP_ECCLR	RW	16	0x30	0x4843 E130
PWMSS_ECAP_ECFRC	RW	16	0x32	0x4843 E132
PWMSS_ECAP_PID	R	32	0x5C	0x4843 E15C

21.3.3.2.2 PWMSS_ECAP Register Description

Table 21-118. PWMSS_ECAP_TSCNT

Address Offset	0x0000 0000		
Physical Address	0x4843 E100	**Instance**	PWMSS_ECAP
Description	Time Stamp Counter Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
TSCNT																															

Bits	Field Name	Description	Type	Reset
31:0	TSCNT	Active 32 bit-counter register that is used as the capture time-base	RW	0x0

Table 21-119. Register Call Summary for Register PWMSS_ECAP_TSCNT

Enhanced Capture (eCAP) Module

- Summary of eCAP Functional Registers: [0]
- PWMSS_ECAP Registers: [1][2]

Table 21-120. PWMSS_ECAP_CNTPHS

Address Offset	0x0000 0004		
Physical Address	0x4843 E104	**Instance**	PWMSS_ECAP
Description	Counter Phase Control Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CNTPHS																															

Bits	Field Name	Description	Type	Reset
31:0	CNTPHS	Counter phase value register that can be programmed for phase lag/lead. This register shadows TSCNT and is loaded into PWMSS_ECAP_TSCNT upon either a SYNCI event or S/W force via a control bit. Used to achieve phase control synchronization with respect to other eCAP and EPWM time-bases.	RW	0x0

Table 21-121. Register Call Summary for Register PWMSS_ECAP_CNTPHS

Enhanced Capture (eCAP) Module
- Summary of eCAP Functional Registers: [0]
- PWMSS_ECAP Registers: [1][2]

Table 21-122. PWMSS_ECAP_CAP1

Address Offset	0x0000 0008		
Physical Address	0x4843 E108	**Instance**	PWMSS_ECAP
Description	Capture-1 Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CAP1																															

Bits	Field Name	Description	Type	Reset
31:0	CAP1	This register can be loaded (written) by the following. (a) Time-Stamp (that is, counter value) during a capture event. (b) Software may be useful for test purposes. (c) APRD active register when used in APWM mode.	RW	0x0

Table 21-123. Register Call Summary for Register PWMSS_ECAP_CAP1

PWM Subsystem Resources
- PWMSS Key Features: [0][1]

Enhanced Capture (eCAP) Module
- eCAP Functional Description: [2][3]
- Capture and APWM Operating Mode: [4][5]
- eCAP Capture Mode Description: [6][7][8][9][10][11][12][13][14][15][16]
- Summary of eCAP Functional Registers: [17][18]
- PWMSS_ECAP Registers: [19][20][21][22][23][24][25][26]

Table 21-124. PWMSS_ECAP_CAP2

Address Offset	0x0000 000C		
Physical Address	0x4843 E10C	**Instance**	PWMSS_ECAP
Description	Capture-2 Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CAP2																															

Bits	Field Name	Description	Type	Reset
31:0	CAP2	This register can be loaded (written) by the following. (a) Time- Stamp (that is, counter value) during a capture event. (b) Software may be useful for test purposes. (c) APRD active register when used in APWM mode.	RW	0x0

Table 21-125. Register Call Summary for Register PWMSS_ECAP_CAP2

Enhanced Capture (eCAP) Module

- Capture and APWM Operating Mode: [0][1]
- eCAP Capture Mode Description: [2][3][4][5][6]
- Summary of eCAP Functional Registers: [7][8]
- PWMSS_ECAP Registers: [9][10][11]

Table 21-126. PWMSS_ECAP_CAP3

Address Offset	0x0000 0010		
Physical Address	0x4843 E110	**Instance**	PWMSS_ECAP
Description	Capture-3 Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CAP3																															

Bits	Field Name	Description	Type	Reset
31:0	CAP3	In CMP mode, this is a time-stamp capture register. In APWM mode, this is the period shadow (APRD) register. User SW updates the PWM period value through this register. In this mode, CAP3 shadows CAP1.	RW	0x0

Table 21-127. Register Call Summary for Register PWMSS_ECAP_CAP3

Enhanced Capture (eCAP) Module

- Capture and APWM Operating Mode: [0][1][2]
- eCAP Capture Mode Description: [3][4][5][6]
- Summary of eCAP Functional Registers: [7][8][9]
- PWMSS_ECAP Registers: [10]

Table 21-128. PWMSS_ECAP_CAP4

Address Offset	0x0000 0014		
Physical Address	0x4843 E114	**Instance**	PWMSS_ECAP
Description	Capture-4 Register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CAP4																															

Bits	Field Name	Description	Type	Reset
31:0	CAP4	In CMP mode, this is a time-stamp capture register. In APWM mode, this is the compare shadow (ACMP) register. User SW updates the PWM compare value through this register. In this mode, CAP4 shadows CAP2.	RW	0x0

Table 21-129. Register Call Summary for Register PWMSS_ECAP_CAP4

PWM Subsystem Resources
- PWMSS Key Features: [0][1]

Enhanced Capture (eCAP) Module
- eCAP Functional Description: [2][3]
- Capture and APWM Operating Mode: [4][5][6]
- eCAP Capture Mode Description: [7][8][9][10]
- Summary of eCAP Functional Registers: [11][12][13]
- PWMSS_ECAP Registers: [14][15][16][17][18][19]

Table 21-130. PWMSS_ECAP_ECCTL1

Address Offset	0x0000 0028		
Physical Address	0x4843 E128	**Instance**	PWMSS_ECAP
Description	ECAP Control Register1		
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FREE_SOFT		EVTFLTPS					CAPLDEN	CTRRST4	CAP4POL	CTRRST3	CAP3POL	CTRRST2	CAP2POL	CTRRST1	CAP1POL

Bits	Field Name	Description	Type	Reset
15:14	FREE_SOFT	Emulation Control 0x0 = TSCNT counter stops immediately on emulation suspend. 0x1 = TSCNT counter runs until = 0. 0x2 = TSCNT counter is unaffected by emulation suspend (Run Free). 0x3 = TSCNT counter is unaffected by emulation suspend (Run Free).	RW	0x0
13:9	EVTFLTPS	Event Filter prescale select: 0x0 = Divide by 1 (i.e,. no prescale, by-pass the prescaler) 0x1 = Divide by 2 0x2 = Divide by 4 0x3 = Divide by 6 0x4 = Divide by 8 0x5 = Divide by 10 ... 0x1E = Divide by 60 0x1F = Divide by 62	RW	0x0
8	CAPLDEN	Enable Loading of PWMSS_ECAP_CAP1 to PWMSS_ECAP_CAP4 registers on a capture event 0x0 = Disable PWMSS_ECAP_CAP1-PWMSS_ECAP_CAP4 register loads at capture event time. 0x1 = Enable PWMSS_ECAP_CAP1-PWMSS_ECAP_CAP4 register loads at capture event time.	RW	0x0
7	CTRRST4	Counter Reset on Capture Event 4 0x0 = Do not reset counter on Capture Event 4 (absolute time stamp operation) 0x1 = Reset counter after Capture Event 4 time-stamp has been captured (used in difference mode operation)	RW	0x0
6	CAP4POL	Capture Event 4 Polarity select 0x0 = Capture Event 4 triggered on a rising edge (RE) 0x1 = Capture Event 4 triggered on a falling edge (FE)	RW	0x0

Bits	Field Name	Description	Type	Reset
5	CTRRST3	Counter Reset on Capture Event 3 0x0 = Do not reset counter on Capture Event 3 (absolute time stamp) 0x1 = Reset counter after Event 3 time-stamp has been captured (used in difference mode operation)	RW	0x0
4	CAP3POL	Capture Event 3 Polarity select 0x0 = Capture Event 3 triggered on a rising edge (RE) 0x1 = Capture Event 3 triggered on a falling edge (FE)	RW	0x0
3	CTRRST2	Counter Reset on Capture Event 2 0x0 = Do not reset counter on Capture Event 2 (absolute time stamp) 0x1 = Reset counter after Event 2 time-stamp has been captured (used in difference mode operation)	RW	0x0
2	CAP2POL	Capture Event 2 Polarity select 0x0 = Capture Event 2 triggered on a rising edge (RE) 0x1 = Capture Event 2 triggered on a falling edge (FE)	RW	0x0
1	CTRRST1	Counter Reset on Capture Event 1 0x0 = Do not reset counter on Capture Event 1 (absolute time stamp) 0x1 = Reset counter after Event 1 time-stamp has been captured (used in difference mode operation)	RW	0x0
0	CAP1POL	Capture Event 1 Polarity select 0x0 = Capture Event 1 triggered on a rising edge (RE) 0x1 = Capture Event 1 triggered on a falling edge (FE)	RW	0x0

Table 21-131. Register Call Summary for Register PWMSS_ECAP_ECCTL1

Enhanced Capture (eCAP) Module

- eCAP Capture Mode Description: [0]
- Summary of eCAP Functional Registers: [1]
- PWMSS_ECAP Registers: [2]

Table 21-132. PWMSS_ECAP_ECCTL2

Address Offset	0x0000 002A		
Physical Address	0x4843 E12A	**Instance**	PWMSS_ECAP
Description	ECAP Control Register 2		
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED					APWMPOL	CAPAPWM	SWSYNC	SYNCO_SEL		SYNCI_EN	TSCNTSTP	REARMRESET	STOPVALUE		CONTONESHT

Bits	Field Name	Description	Type	Reset
15:11	RESERVED		R	0x0
10	APWMPOL	APWM output polarity select. This is applicable only in APWM operating mode 0x0 = Output is active high (Compare value defines high time) 0x1 = Output is active low (Compare value defines low time)	RW	0x0

Bits	Field Name	Description	Type	Reset
9	CAPAPWM	CAP/APWM operating mode select **0x0 = ECAP module operates in capture mode. This mode forces the following configuration.** (a) Inhibits TSCNT resets via TSCNT = PRD event. (b) Inhibits shadow loads on PWMSS_ECAP_CAP1 and PWMSS_ECAP_CAP2 registers. (c) Permits user to enable PWMSS_ECAP_CAP1-PWMSS_ECAP_CAP4 register load. (d) ECAP input / APWM output pin operates as a capture input. **0x1 = ECAP module operates in APWM mode. This mode forces the following configuration.** (a) Resets TSCNT on TSCNT = PRD event (period boundary). (b) Permits shadow loading on PWMSS_ECAP_CAP1 and PWMSS_ECAP_CAP2 registers. (c) Disables loading of time-stamps into PWMSS_ECAP_CAP1 - PWMSS_ECAP_CAP4 registers. (d) ECAP input / APWM ouput pin operates as a APWM output.	RW	0x0
8	SWSYNC	Software-forced Counter (TSCNT) Synchronizing. This provides a convenient software method to synchronize some or all ECAP time bases. In APWM mode, the synchronizing can also be done via the TSCNT = PRD event. Note: Selection TSCNT = PRD is meaningful only in APWM mode. However, you can choose it in CAP mode if you find doing so useful. 0x0 = Writing a zero has no effect. Reading always returns a zero 0x1 = Writing a one forces a TSCNT shadow load of ECAP module providing the SYNCO_SEL bits are 0b00. After writing a 1, this bit returns to a zero.	RW	0x0
7:6	SYNCO_SEL	Sync-Out Select 0x0 = Select sync-in event to be the sync-out signal (pass through) 0x1 = Select TSCNT = PRD event to be the sync-out signal 0x2 = Disable sync out signal 0x3 = Disable sync out signal	RW	0x0
5	SYNCI_EN	Counter (TSCNT) Sync-In select mode 0x0 = Disable sync-in option 0x1 = Enable counter (TSCNT) to be loaded from PWMSS_ECAP_CNTPHS register upon either a SYNCI signal or a S/W force event.	RW	0x0
4	TSCNTSTP	Time Stamp (TSCNT) Counter Stop (freeze) Control 0x0 = TSCNT stopped 0x1 = TSCNT free-running	RW	0x0
3	REARMRESET	One-Shot Re-Arming Control, that is, wait for stop trigger. Note: The re-arm function is valid in one shot or continuous mode. 0x0 = Has no effect (reading always returns a 0) 0x1 = Arms the one-shot sequence as follows: 1) Resets the Mod4 counter to zero. 2) Unfreezes the Mod4 counter. 3) Enables capture register loads.	RW	0x0

Bits	Field Name	Description	Type	Reset
2:1	STOPVALUE	Stop value for one-shot mode. This is the number (between 1 and 4) of captures allowed to occur before the CAP (1 through 4) registers are frozen, that is, capture sequence is stopped. Wrap value for continuous mode. This is the number (between 1 and 4) of the capture register in which the circular buffer wraps around and starts again. Notes: STOPVALUE is compared to Mod4 counter and, when equal, the following two actions occur. (1) Mod4 counter is stopped (frozen). (2) Capture register loads are inhibited. In one-shot mode, further interrupt events are blocked until re-armed. 0x0 = Stop after Capture Event 1 in one-shot mode. Wrap after Capture Event 1 in continuous mode. 0x1 = Stop after Capture Event 2 in one-shot mode. Wrap after Capture Event 2 in continuous mode. 0x2 = Stop after Capture Event 3 in one-shot mode. Wrap after Capture Event 3 in continuous mode. 0x3 = Stop after Capture Event 4 in one-shot mode. Wrap after Capture Event 4 in continuous mode.	RW	0x3
0	CONTONESHT	Continuous or one-shot mode control (applicable only in capture mode) 0x0 = Operate in continuous mode 0x1 = Operate in one-shot mode	RW	0x0

Table 21-133. Register Call Summary for Register PWMSS_ECAP_ECCTL2

Enhanced Capture (eCAP) Module
- eCAP Capture Mode Description: [0][1]
- Summary of eCAP Functional Registers: [2]
- PWMSS_ECAP Registers: [3]

Table 21-134. PWMSS_ECAP_ECEINT

Address Offset	0x0000 002C		
Physical Address	0x4843 E12C	**Instance**	PWMSS_ECAP
Description	ECAP Interrupt Enable Register		
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								CMPEQ	PRDEQ	CNTOVF	CEVT4	CEVT3	CEVT2	CEVT1	RESERVED

Bits	Field Name	Description	Type	Reset
15:8	RESERVED		R	0x0
7	CMPEQ	Counter Equal **Compare** Interrupt Enable. 0x0 = Disable Compare Equal as an Interrupt source. 0x1 = Enable Compare Equal as an Interrupt source.	RW	0x0
6	PRDEQ	Counter Equal **Period** Interrupt Enable. 0x0 = Disable Period Equal as an Interrupt source. 0x1 = Enable Period Equal as an Interrupt source.	RW	0x0
5	CNTOVF	Counter Overflow Interrupt Enable. 0x0 = Disable counter Overflow as an Interrupt source. 0x1 = Enable counter Overflow as an Interrupt source.	RW	0x0

Bits	Field Name	Description	Type	Reset
4	CEVT4	Capture Event 4 Interrupt Enable. 0x0 = Disable Capture Event 4 as an Interrupt source. 0x1 = Enable Capture Event 4 as an Interrupt source.	RW	0x0
3	CEVT3	Capture Event 3 Interrupt Enable. 0x0 = Disable Capture Event 3 as an Interrupt source. 0x1 = Enable Capture Event 3 as an Interrupt source.	RW	0x0
2	CEVT2	Capture Event 2 Interrupt Enable. 0x0 = Disable Capture Event 2 as an Interrupt source. 0x1 = Enable Capture Event 2 as an Interrupt source.	RW	0x0
1	CEVT1	Capture Event 1 Interrupt Enable . 0x0 = Disable Capture Event 1 as an Interrupt source. 0x1 = Enable Capture Event 1 as an Interrupt source.	RW	0x0
0	RESERVED		R	0x0

Table 21-135. Register Call Summary for Register PWMSS_ECAP_ECEINT

Enhanced Capture (eCAP) Module

- eCAP Capture Mode Description: [0]
- Summary of eCAP Functional Registers: [1]
- PWMSS_ECAP Registers: [2]

Table 21-136. PWMSS_ECAP_ECFLG

Address Offset	0x0000 002E		
Physical Address	0x4843 E12E	**Instance**	PWMSS_ECAP
Description	ECAP Interrupt Flag Register		
Type	R		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								CMPEQ	PRDEQ	CNTOVF	CEVT4	CEVT3	CEVT2	CEVT1	INT

Bits	Field Name	Description	Type	Reset
15:8	RESERVED		R	0x0
7	CMPEQ	Compare Equal Compare Status Flag. This flag is only active in APWM mode. 0x0 = Indicates no event occurred 0x1 = Indicates the counter (TSCNT) reached the compare register value (ACMP)	R	0x0
6	PRDEQ	Counter Equal Period Status Flag. This flag is only active in APWM mode. 0x0 = Indicates no event occurred 0x1 = Indicates the counter (TSCNT) reached the period register value (APRD) and was reset.	R	0x0
5	CNTOVF	Counter Overflow Status Flag. This flag is active in CAP and APWM mode. 0x0 = Indicates no event occurred. 0x1 = Indicates the counter (TSCNT) has made the transition from 0xFFFFFFFF to 0x00000000	R	0x0

Bits	Field Name	Description	Type	Reset
4	CEVT4	Capture Event 4 Status Flag This flag is only active in CAP mode. 0x0 = Indicates no event occurred 0x1 = Indicates the fourth event occurred at ECAP pin	R	0x0
3	CEVT3	Capture Event 3 Status Flag. This flag is active only in CAP mode. 0x0 = Indicates no event occurred. 0x1 = Indicates the third event occurred at ECAP pin.	R	0x0
2	CEVT2	Capture Event 2 Status Flag. This flag is only active in CAP mode. 0x0 = Indicates no event occurred. 0x1 = Indicates the second event occurred at ECAP pin.	R	0x0
1	CEVT1	Capture Event 1 Status Flag. This flag is only active in CAP mode. 0x0 = Indicates no event occurred. 0x1 = Indicates the first event occurred at ECAP pin.	R	0x0
0	INT	Global Interrupt Status Flag 0x0 = Indicates no interrupt generated. 0x1 = Indicates that an interrupt was generated.	R	0x0

Table 21-137. Register Call Summary for Register PWMSS_ECAP_ECFLG

Enhanced Capture (eCAP) Module

- eCAP Capture Mode Description: [0]
- Summary of eCAP Functional Registers: [1]
- PWMSS_ECAP Registers: [2]

Table 21-138. PWMSS_ECAP_ECCLR

Address Offset	0x0000 0030		
Physical Address	0x4843 E130	**Instance**	PWMSS_ECAP
Description	ECAP Interrupt Clear Register		
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								CMPEQ	PRDEQ	CNTOVF	CEVT4	CEVT3	CEVT2	CEVT1	INT

Bits	Field Name	Description	Type	Reset
15:8	RESERVED		R	0x0
7	CMPEQ	Counter Equal Compare Status Flag 0x0 = Writing a 0 has no effect. Always reads back a 0 0x1 = Writing a 1 clears the TSCNT=CMP flag condition	RW	0x0
6	PRDEQ	Counter Equal Period Status Flag 0x0 = Writing a 0 has no effect. Always reads back a 0 0x1 = Writing a 1 clears the TSCNT=PRD flag condition	RW	0x0
5	CNTOVF	Counter Overflow Status Flag 0x0 = Writing a 0 has no effect. Always reads back a 0 0x1 = Writing a 1 clears the CNTOVF flag condition	RW	0x0

Bits	Field Name	Description	Type	Reset
4	CEVT4	Capture Event 4 Status Flag 0x0 = Writing a 0 has no effect. Always reads back a 0. 0x1 = Writing a 1 clears the CEVT3 flag condition.	RW	0x0
3	CEVT3	Capture Event 3 Status Flag 0x0 = Writing a 0 has no effect. Always reads back a 0. 0x1 = Writing a 1 clears the CEVT3 flag condition.	RW	0x0
2	CEVT2	Capture Event 2 Status Flag 0x0 = Writing a 0 has no effect. Always reads back a 0. 0x1 = Writing a 1 clears the CEVT2 flag condition.	RW	0x0
1	CEVT1	Capture Event 1 Status Flag 0x0 = Writing a 0 has no effect. Always reads back a 0. 0x1 = Writing a 1 clears the CEVT1 flag condition.	RW	0x0
0	INT	Global Interrupt Clear Flag 0x0 = Writing a 0 has no effect. Always reads back a 0. 0x1 = Writing a 1 clears the INT flag and enable further interrupts to be generated if any of the event flags are set to 1.	RW	0x0

Table 21-139. Register Call Summary for Register PWMSS_ECAP_ECCLR

Enhanced Capture (eCAP) Module

- eCAP Capture Mode Description: [0]
- Summary of eCAP Functional Registers: [1]
- PWMSS_ECAP Registers: [2]

Table 21-140. PWMSS_ECAP_ECFRC

Address Offset	0x0000 0034		
Physical Address	0x4843 E132	**Instance**	PWMSS_ECAP
Description	ECAP Interrupt Forcing Register		
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								CMPEQ	PRDEQ	CNTOVF	CEVT4	CEVT3	CEVT2	CEVT1	RESERVED

Bits	Field Name	Description	Type	Reset
15:8	RESERVED		R	0x0
7	CMPEQ	Force Counter Equal Compare Interrupt 0x0 = No effect. Always reads back a 0. 0x1 = Writing a 1 sets the TSCNT=CMP flag bit.	RW	0x0
6	PRDEQ	Force Counter Equal Period Interrupt 0x0 = No effect. Always reads back a 0. 0x1 = Writing a 1 sets the TSCNT=PRD flag bit.	RW	0x0
5	CNTOVF	Force Counter Overflow 0x0 = No effect. Always reads back a 0. 0x1 = Writing a 1 to this bit sets the CNTOVF flag bit.	RW	0x0
4	CEVT4	Force Capture Event 4 0x0 = No effect. Always reads back a 0. 0x1 = Writing a 1 sets the CEVT4 flag bit	RW	0x0

Bits	Field Name	Description	Type	Reset
3	CEVT3	Force Capture Event 3 0x0 = No effect. Always reads back a 0. 0x1 = Writing a 1 sets the CEVT3 flag bit	RW	0x0
2	CEVT2	Force Capture Event 2 0x0 = No effect. Always reads back a 0. 0x1 = Writing a 1 sets the CEVT2 flag bit.	RW	0x0
1	CEVT1	Always reads back a 0. Force Capture Event 1 0x0 = No effect. 0x1 = Writing a 1 sets the CEVT1 flag bit.	RW	0x0
0	RESERVED		R	0x0

Table 21-141. Register Call Summary for Register PWMSS_ECAP_ECFRC

Enhanced Capture (eCAP) Module
- eCAP Capture Mode Description: [0]
- Summary of eCAP Functional Registers: [1]
- PWMSS_ECAP Registers: [2]

Table 21-142. PWMSS_ECAP_PID

Address Offset	0x0000 005C		
Physical Address	0x4843 E15C	**Instance**	PWMSS_ECAP
Description	ECAP Revision ID		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REVISION																															

Bits	Field Name	Description	Type	Reset
31:0	REVISION	IP Revision	R	0x-(1)

(1) TI Internal information

Table 21-143. Register Call Summary for Register PWMSS_ECAP_PID

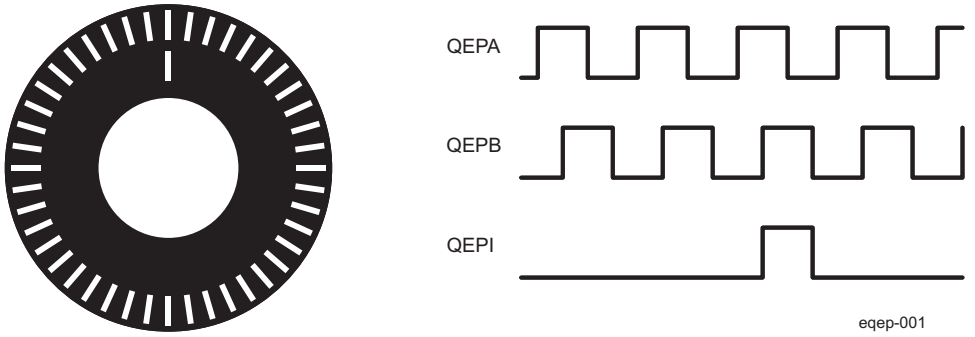
Enhanced Capture (eCAP) Module
- Summary of eCAP Functional Registers: [0]
- PWMSS_ECAP Registers: [1]

21.4 Enhanced Quadrature Encoder Pulse (eQEP) Module

21.4.1 eQEP Overview

A single track of slots patterns the periphery of an incremental encoder disk, as shown in Figure 21-57. These slots create an alternating pattern of dark and light lines. The disk count is defined as the number of dark/light line pairs that occur per revolution (lines per revolution). As a rule, a second track is added to generate a signal that occurs once per revolution (index signal: QEPI), which can be used to indicate an absolute position. Encoder manufacturers identify the index pulse using different terms such as index, marker, home position, and zero reference.

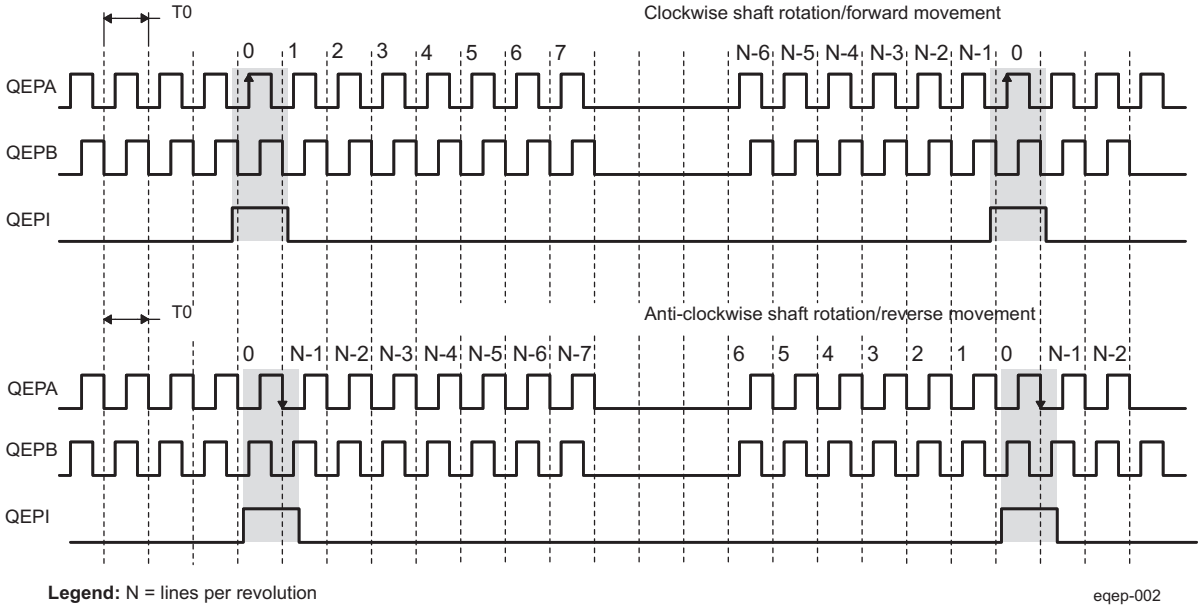
Figure 21-57. Optical Encoder Disk



To derive direction information, the lines on the disk are read out by two different photo-elements that "look" at the disk pattern with a mechanical shift of 1/4 the pitch of a line pair between them. This shift is realized with a reticle or mask that restricts the view of the photo-element to the desired part of the disk lines. As the disk rotates, the two photo-elements generate signals that are shifted 90 degrees out of phase from each other. These are commonly called the quadrature QEPA and QEPB signals. The clockwise direction for most encoders is defined as the QEPA channel going positive before the QEPB channel and vise versa as shown in Figure 21-58.

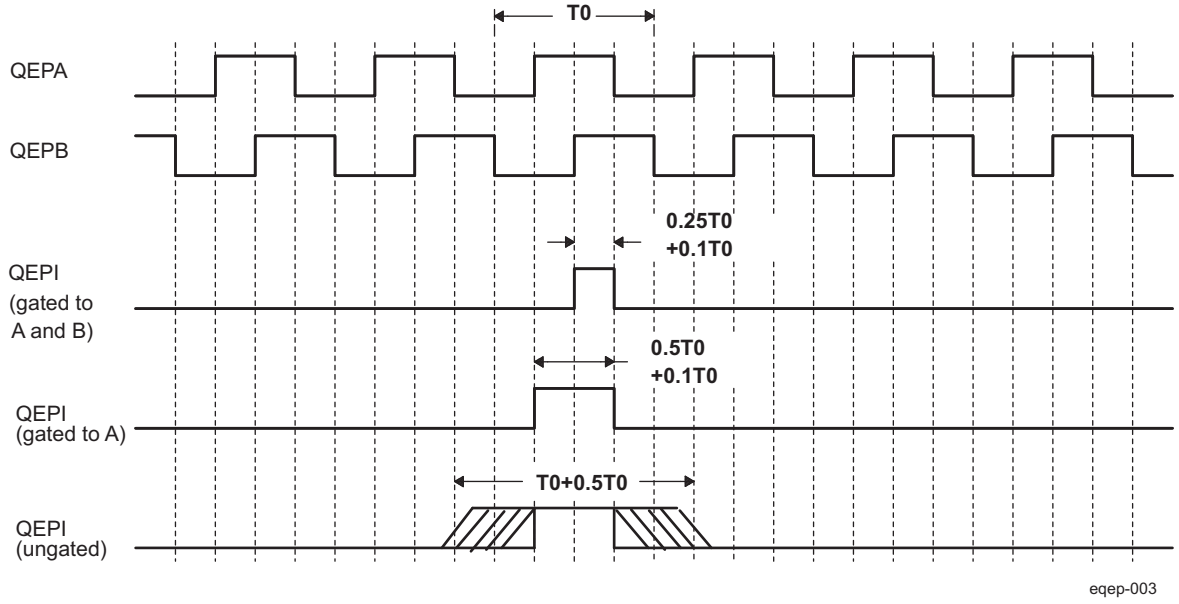
The encoder wheel typically makes one revolution for every revolution of the motor or the wheel may be at a geared rotation ratio with respect to the motor. Therefore, the frequency of the digital signal coming from the QEPA and QEPB outputs varies proportionally with the velocity of the motor. For example, a 2000-line encoder directly coupled to a motor running at 5000 revolutions per minute (rpm) results in a frequency of 166.6 KHz, so by measuring the frequency of either the QEPA or QEPB output, the processor can determine the velocity of the motor.

Figure 21-58. QEP Encoder Output Signal for Forward/Reverse Movement



Quadrature encoders from different manufacturers come with two forms of index pulse (gated index pulse or ungated index pulse) as shown in Figure 21-59. A nonstandard form of index pulse is ungated. In the ungated configuration, the index edges are not necessarily coincident with A and B signals. The gated index pulse is aligned to any of the four quadrature edges and width of the index pulse and can be equal to a quarter, half, or full period of the quadrature signal.

Figure 21-59. Index Pulse Example



Some typical applications of shaft encoders include robotics and even computer input in the form of a mouse. Inside your mouse you can see where the mouse ball spins a pair of axles (a left/right, and an up/down axle). These axles are connected to optical shaft encoders that effectively tell the computer how fast and in what direction the mouse is moving.

General Issues: Estimating velocity from a digital position sensor is a cost-effective strategy in motor control. Two different first order approximations for velocity may be written as:

$$V(k) \approx \frac{X(k) - X(k-1)}{T} = \frac{\Delta X}{T} \quad \text{eqep-004} \tag{8}$$

$$V(k) \approx \frac{X}{t(k) - t(k-1)} = \frac{X}{\Delta T} \quad \text{eqep-005} \tag{9}$$

where

v(k): Velocity at time instant k

x(k): Position at time instant k

x(k-1): Position at time instant k - 1

T: Fixed unit time or inverse of velocity calculation rate

ΔX: Incremental position movement in unit time

t(k): Time instant "k"

t(k-1): Time instant "k - 1"

X: Fixed unit position

ΔT: Incremental time elapsed for unit position movement.

Equation 8 is the conventional approach to velocity estimation and it requires a time base to provide unit time event for velocity calculation. Unit time is basically the inverse of the velocity calculation rate.

The encoder count (position) is read once during each unit time event. The quantity [x(k) - x(k-1)] is formed by subtracting the previous reading from the current reading. Then the velocity estimate is computed by multiplying by the known constant 1/T (where T is the constant time between unit time events and is known in advance).

Estimation based on Equation 8 has an inherent accuracy limit directly related to the resolution of the position sensor and the unit time period T. For example, consider a 500-line per revolution quadrature encoder with a velocity calculation rate of 400 Hz. When used for position the quadrature encoder gives a four-fold increase in resolution, in this case, 2000 counts per revolution. The minimum rotation that can be detected is therefore 0.0005 revolutions, which gives a velocity resolution of 12 rpm when sampled at 400 Hz. While this resolution may be satisfactory at moderate or high speeds, for example, 1% error at 1200 rpm, it would clearly prove inadequate at low speeds. In fact, at speeds below 12 rpm, the speed estimate would erroneously be zero much of the time.

At low speed, Equation 9 provides a more accurate approach. It requires a position sensor that outputs a fixed interval pulse train, such as the aforementioned quadrature encoder. The width of each pulse is defined by motor speed for a given sensor resolution. Equation 9 can be used to calculate motor speed by measuring the elapsed time between successive quadrature pulse edges. However, this method suffers from the opposite limitation, as does Equation 8. A combination of relatively large motor speeds and high sensor resolution makes the time interval ΔT small, and thus more greatly influenced by the timer resolution. This can introduce considerable error into high-speed estimates.

For systems with a large speed range (that is, speed estimation is needed at both low and high speeds), one approach is to use Equation 9 at low speed and have the software switch over to Equation 8 when the motor speed rises above some specified threshold.

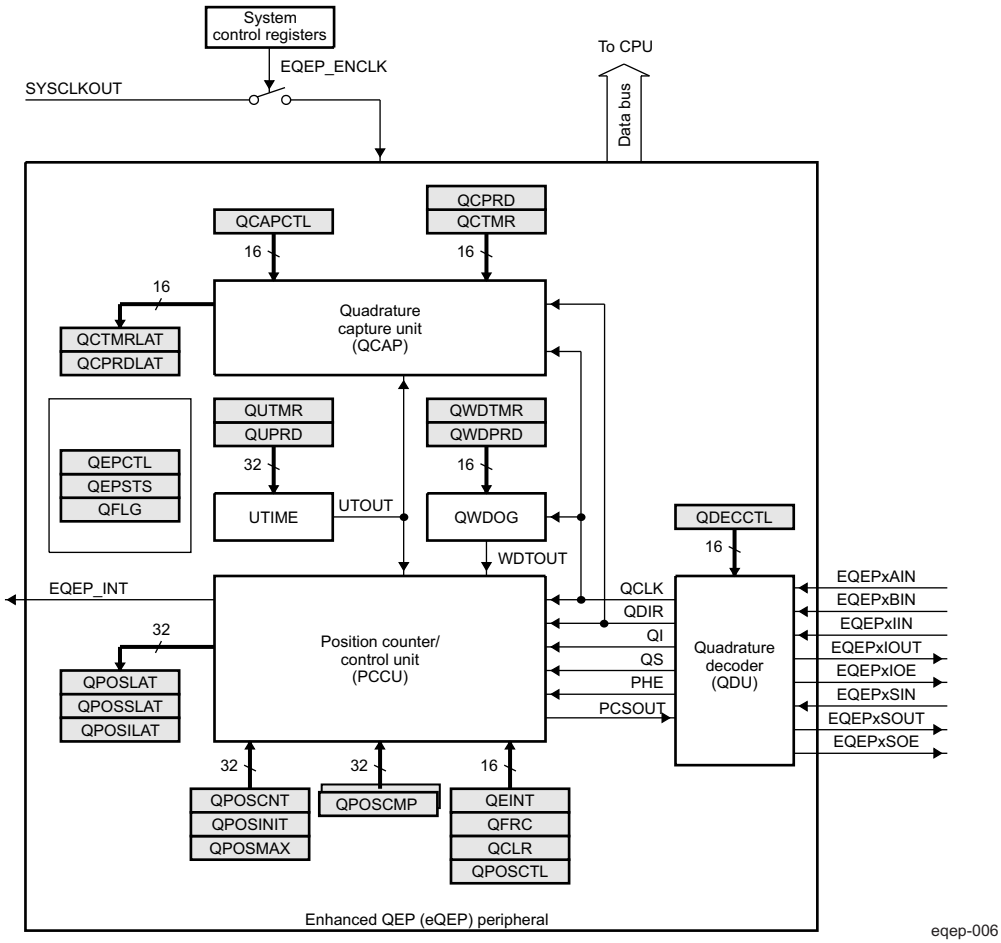
21.4.2 eQEP Module Functional Description

This section provides the eQEP functional description and corresponding functional details about EQEP inputs .

The eQEP peripheral contains the following major functional units (as shown in Figure 21-60):

- Programmable input qualification for each pin (part of the GPIO MUX)
- Quadrature decoder unit (QDU)
- Position counter and control unit for position measurement (PCCU)
- Quadrature edge-capture unit for low-speed measurement (QCAP)
- Unit time base for speed/frequency measurement (UTIME)
- Watchdog timer for detecting stalls (QWDOG)

Figure 21-60. Functional Block Diagram of the eQEP Peripheral



21.4.2.1 eQEP Inputs

The eQEP inputs include two pins for quadrature-clock mode or direction-count mode, an index (or 0 marker), and a strobe input.

- QEPA/XCLK and QEPB/XDIR: These two pins can be used in quadrature-clock mode or direction-count mode.
  - Quadrature-clock Mode: The eQEP encoders provide two square wave signals (A and B) 90 electrical degrees out of phase whose phase relationship is used to determine the direction of rotation of the input shaft and number of eQEP pulses from the index position to derive the relative position information. For forward or clockwise rotation, QEPA signal leads QEPB signal and vice versa. The quadrature decoder uses these two inputs to generate quadrature-clock and direction

signals.

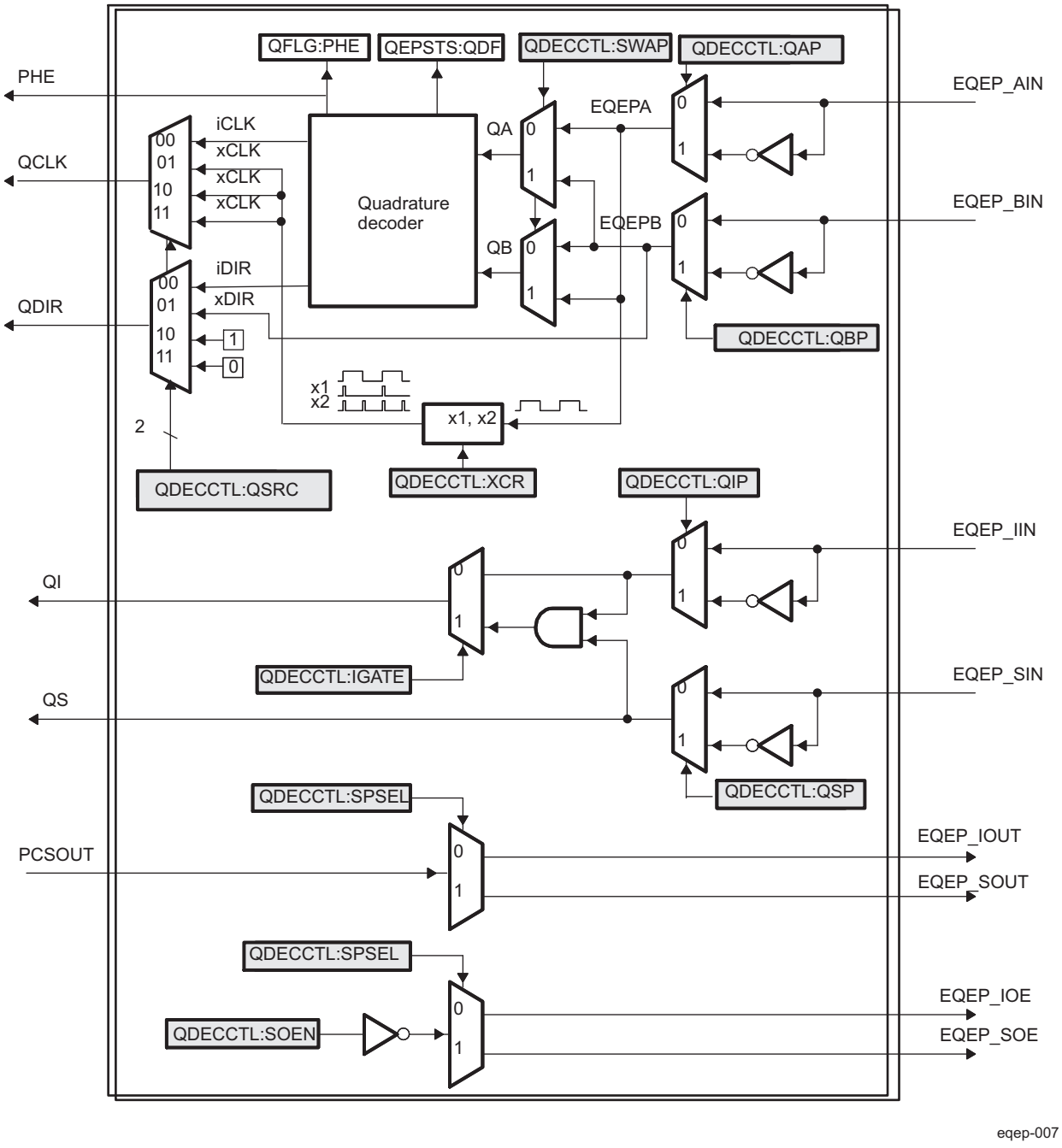
  - Direction-count Mode: In direction-count mode, direction and clock signals are provided directly from the external source. Some position encoders have this type of output instead of quadrature output. The QEPA pin provides the clock input and the QEPB pin provides the direction input.

- QEPI: Index or Zero Marker: The eQEP encoder uses an index signal to assign an absolute start position from which position information is incrementally encoded using quadrature pulses. This pin is connected to the index output of the eQEP encoder to optionally reset the position counter for each revolution. This signal can be used to initialize or latch the position counter on the occurrence of a desired event on the index pin.
- QEPS: Strobe Input: This general-purpose strobe signal can initialize or latch the position counter on the occurrence of a desired event on the strobe pin. This signal is typically connected to a sensor or limit switch to notify that the motor has reached a defined position.

21.4.2.2 eQEP Quadrature Decoder Unit (QDU)

Figure 21-61 shows a functional block diagram of the QDU.

Figure 21-61. Functional Block Diagram of Decoder Unit



21.4.2.2.1 eQEP Position Counter Input Modes

Clock and direction input to position counter is selected using the QSRC bit in the eQEP decoder control register (EQEP_QDECCTL), based on interface input requirement as follows:

- Quadrature-count mode
- Direction-count mode
- UP-count mode
- DOWN-count mode

21.4.2.2.1.1 Quadrature Count Mode

The quadrature decoder generates the direction and clock to the position counter in quadrature count mode.

Direction Decoding— The direction decoding logic of the eQEP circuit determines which one of the sequences (QEPA, QEPB) is the leading sequence and accordingly updates the direction information in the QDF bit in the eQEP status register (EQEP_QEPSTS). Table 21-144 and Figure 21-62 show the direction decoding logic in truth table and state machine form. Both edges of the QEPA and QEPB signals are sensed to generate count pulses for the position counter. Therefore, the frequency of the clock generated by the eQEP logic is four times that of each input sequence. Figure 21-63 shows the direction decoding and clock generation from the eQEP input signals.

Phase Error Flag— In normal operating conditions, quadrature inputs QEPA and QEPB will be 90 degrees out of phase. The phase error flag (PHE) is set in the EQEP_QFLG register when edge transition is detected simultaneously on the QEPA and QEPB signals to optionally generate interrupts. State transitions marked by dashed lines in Figure 21-62 are invalid transitions that generate a phase error.

Count Multiplication— The eQEP position counter provides 4x times the resolution of an input clock by generating a quadrature-clock (QCLK) on the rising/falling edges of both eQEP input clocks (QEPA and QEPB) as shown in Figure 21-63.

Reverse Count— In normal quadrature count operation, QEPA input is fed to the QA input of the quadrature decoder and the QEPB input is fed to the QB input of the quadrature decoder. Reverse counting is enabled by setting the SWAP bit in the eQEP decoder control register (EQEP_QDECCTL). This will swap the input to the quadrature decoder thereby reversing the counting direction.

Table 21-144. Quadrature Decoder Truth Table

Previous Edge	Present Edge	QDIR	QPOSCNT
QA↑	QB↑	UP	Increment
	QB↓	DOWN	Decrement
	QA↓	TOGGLE	Increment or Decrement
QA↓	QB↓	UP	Increment
	QB↑	DOWN	Decrement
	QA↑	TOGGLE	Increment or Decrement
QB↑	QA↑	DOWN	Increment
	QA↓	UP	Decrement
	QB↓	TOGGLE	Increment or Decrement
QB↓	QA↓	DOWN	Increment
	QA↑	UP	Decrement
	QB↑	TOGGLE	Increment or Decrement

Figure 21-62. Quadrature Decoder State Machine

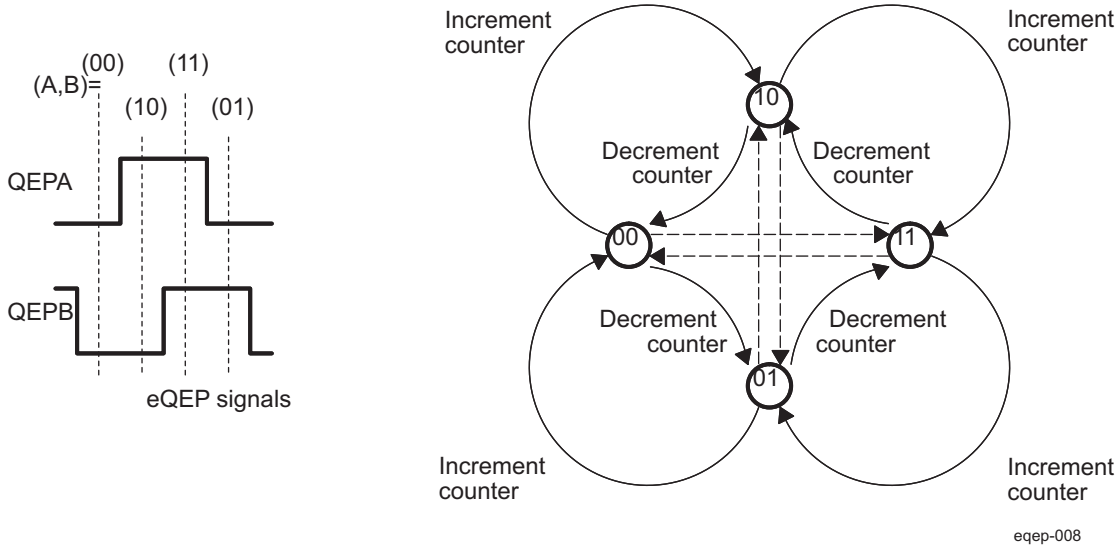
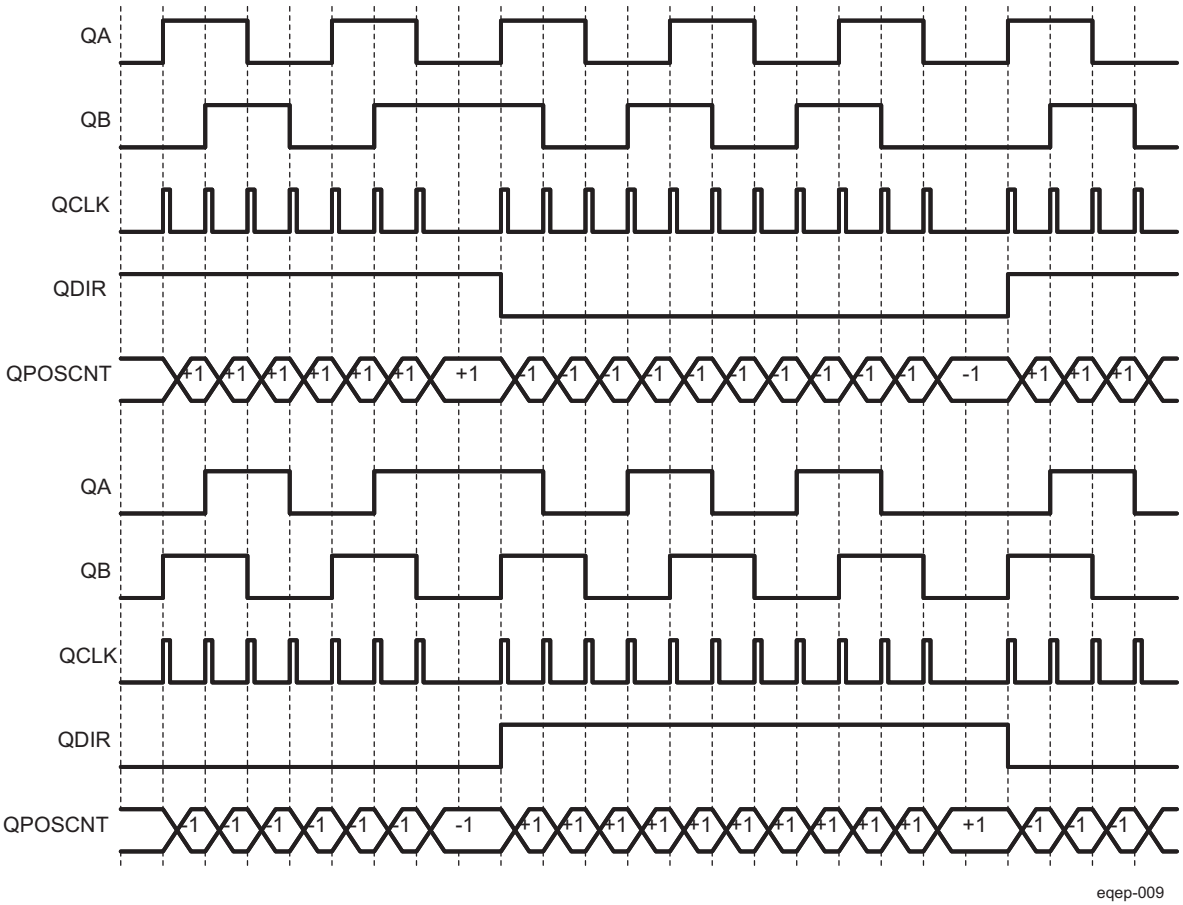


Figure 21-63. Quadrature-clock and Direction Decoding



21.4.2.2.1.2 eQEP Direction-count Mode

Some position encoders provide direction and clock outputs, instead of quadrature outputs. In such cases, direction-count mode can be used. QEPA input will provide the clock for position counter and the QEPB input will have the direction information. The position counter is incremented on every rising edge of a QEPA input when the direction input is high and decremented when the direction input is low.

21.4.2.2.1.3 eQEP Up-Count Mode

The counter direction signal is hard-wired for up count and the position counter is used to measure the frequency of the QEPA input. Setting of the XCR bit in the eQEP decoder control register (EQEP_QDECCTL) enables clock generation to the position counter on both edges of the QEPA input, thereby increasing the measurement resolution by 2× factor.

21.4.2.2.1.4 eQEP Down-Count Mode

The counter direction signal is hardwired for a down count and the position counter is used to measure the frequency of the QEPA input. Setting of the XCR bit in the eQEP decoder control register (EQEP_QDECCTL) enables clock generation to the position counter on both edges of a QEPA input, thereby increasing the measurement resolution by 2× factor.

21.4.2.2.2 eQEP Input Polarity Selection

Each eQEP input can be inverted using the in the eQEP decoder control register (EQEP_QDECCTL[8:5]) control bits. As an example, setting of the QIP bit in EQEP_QDECCTL inverts the index input.

21.4.2.2.3 eQEP Position-Compare Sync Output

The eQEP peripheral includes a position-compare unit that is used to generate the position-compare sync signal on compare match between the position counter register (EQEP_QPOSCNT) and the position-compare register (EQEP_QPOSCMP). This sync signal can be output using an index pin or strobe pin of the EQEP peripheral.

Setting the SOEN bit in the eQEP decoder control register (EQEP_QDECCTL) enables the position-compare sync output and the SPSEL bit in EQEP_QDECCTL selects either an eQEP index pin or an eQEP strobe pin.

21.4.2.3 eQEP Position Counter and Control Unit (PCCU)

The position counter and control unit provides two configuration registers (EQEP_QEPCTL and EQEP_QPOSCTL) for setting up position counter operational modes, position counter initialization/latch modes and position-compare logic for sync signal generation.

21.4.2.3.1 eQEP Position Counter Operating Modes

Position counter data may be captured in different manners. In some systems, the position counter is accumulated continuously for multiple revolutions and the position counter value provides the position information with respect to the known reference. An example of this is the quadrature encoder mounted on the motor controlling the print head in the printer. Here the position counter is reset by moving the print head to the home position and then position counter provides absolute position information with respect to home position.

In other systems, the position counter is reset on every revolution using index pulse and position counter provides rotor angle with respect to index pulse position.

Position counter can be configured to operate in following four modes

- Position Counter Reset on Index Event
- Position Counter Reset on Maximum Position
- Position Counter Reset on the first Index Event
- Position Counter Reset on Unit Time Out Event (Frequency Measurement)

In all the above operating modes, position counter is reset to 0 on overflow and to QPOSMAX bifield value in EQEP_QPOSMAX register on underflow. Overflow occurs when the position counter counts up after QPOSMAX value. Underflow occurs when position counter counts down after "0". Interrupt flag is set to indicate overflow/underflow in EQEP_QFLG register.

21.4.2.3.1.1 eQEP Position Counter Reset on Index Event (EQEP_QEPCTL[31:12] PCRM] = 0b00)

If the index event occurs during the forward movement, then position counter is reset to 0 on the next eQEP clock. If the index event occurs during the reverse movement, then the position counter is reset to the value in the EQEP_QPOSMAX register on the next eQEP clock.

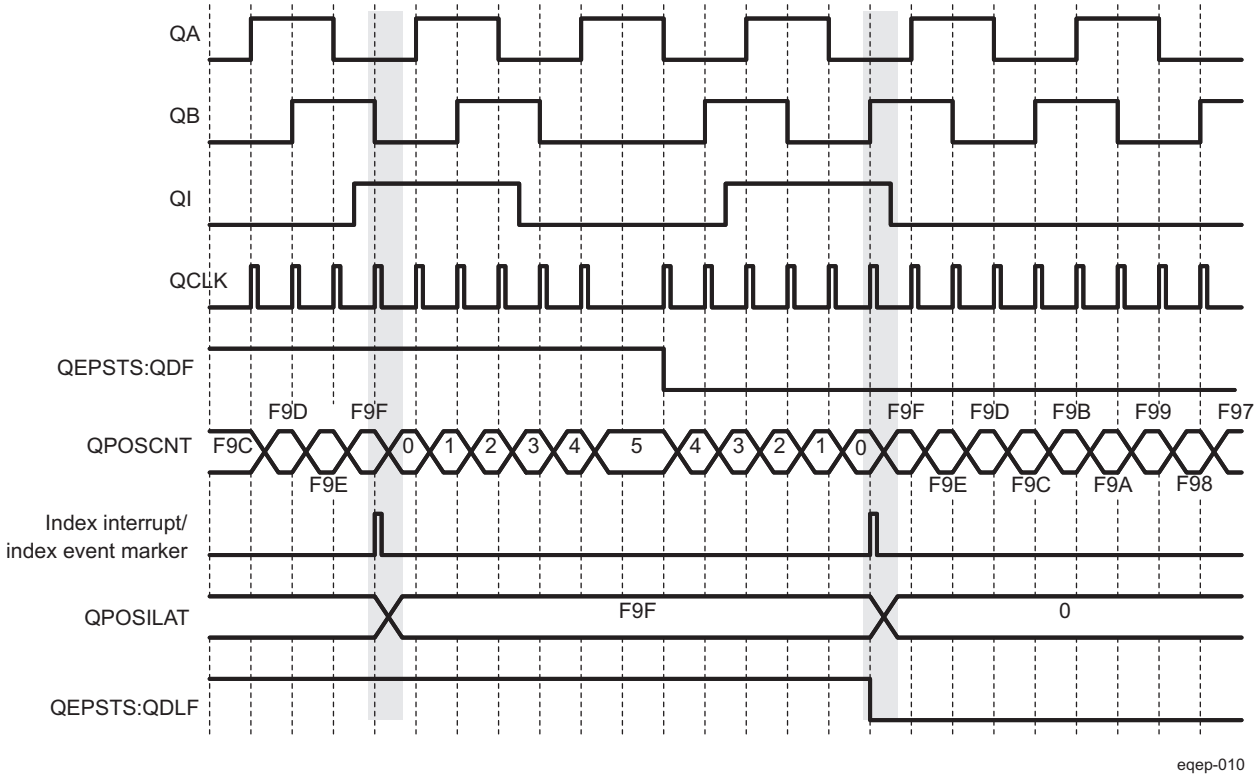
First index marker is defined as the quadrature edge following the first index edge. The eQEP peripheral records the occurrence of the first index marker (EQEP_QEPSTS[FIMF]) and direction on the first index event marker (EQEP_QEPSTS[FIDF]) in EQEP_QEPSTS registers, it also remembers the quadrature edge on the first index marker so that same relative quadrature transition is used for index event reset operation.

For example, if the first reset operation occurs on the falling edge of QEPB during the forward direction, then all the subsequent reset must be aligned with the falling edge of QEPB for the forward rotation and on the rising edge of QEPB for the reverse rotation as shown in Figure 21-64.

The position-counter value is latched to the EQEP_QPOSILAT register and direction information is recorded in the EQEP_QEPSTS[QDLF] bit on every index event marker. The position-counter error flag (EQEP_QEPSTS[PCEF]) and error interrupt flag (EQEP_QFLG[PCE]) are set if the latched value is not equal to 0 or QPOSMAX. The position-counter error flag (EQEP_QEPSTS[PCEF]) is updated on every index event marker and an interrupt flag (EQEP_QFLG[PCE]) will be set on error that can be cleared only through software.

The index event latch configuration EQEP_QEPCTL[5:4] IEL bits are ignored in this mode and position counter error flag/interrupt flag are generated only in index event reset mode.

Figure 21-64. Position Counter Reset by Index Pulse for 1000 Line Encoder (QPOSMAX = 3999 or F9Fh)

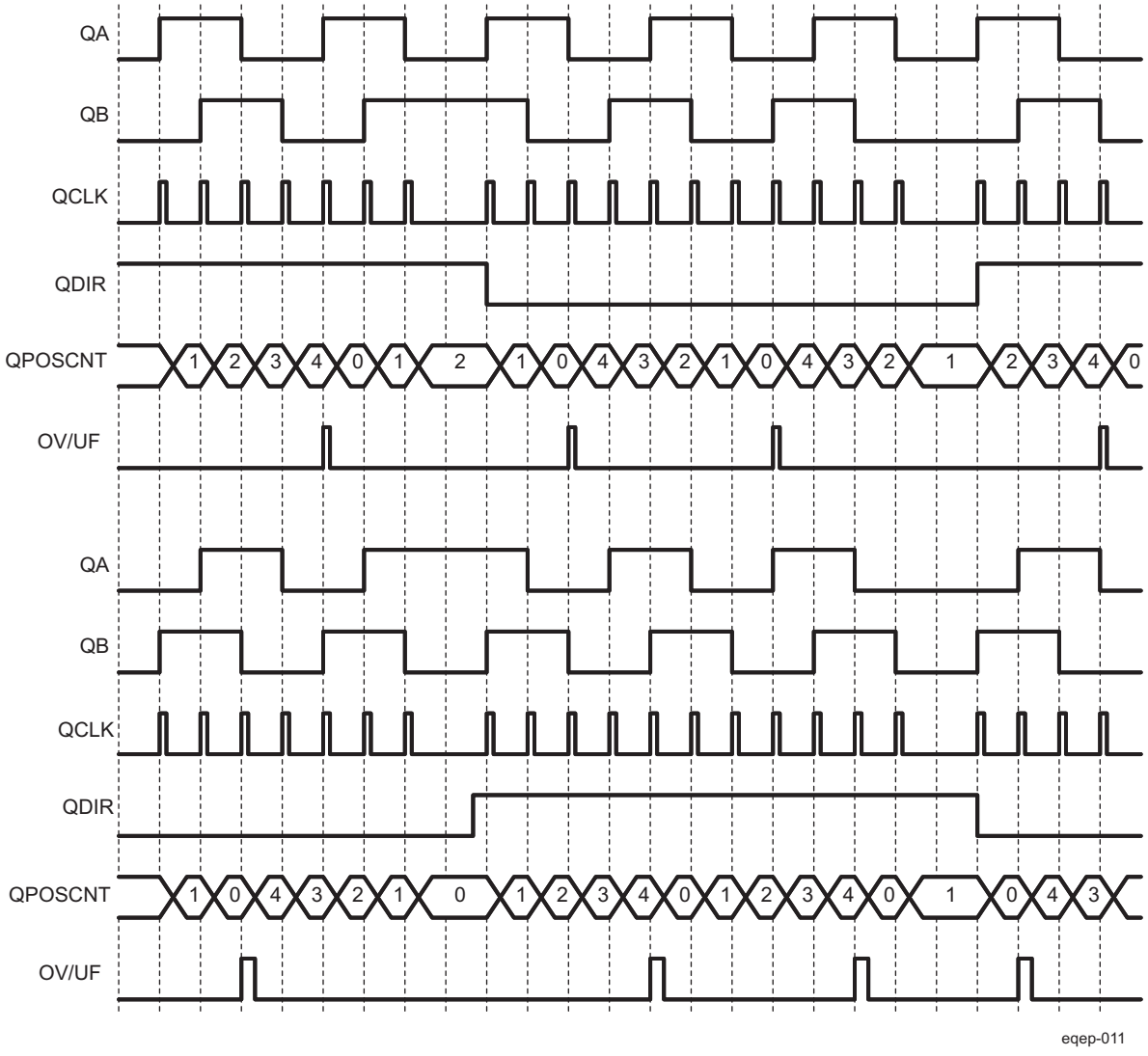


21.4.2.3.1.2 eQEP Position Counter Reset on Maximum Position (EQEP_QEPCTL[13:12] PCRM=0b01)

If the position counter is equal to QPOSMAX (in EQEP_QPOSMAX register), then the position counter is reset to 0 on the next eQEP clock for forward movement and position counter overflow flag is set. If the position counter is equal to ZERO, then the position counter is reset to QPOSMAX on the next QEP clock for reverse movement and position counter underflow flag is set. Figure 21-65 shows the position counter reset operation in this mode.

First index marker is defined as the quadrature edge following the first index edge. The eQEP peripheral records the occurrence of the first index marker (EQEP_QEPSTS[FIMF]) and direction on the first index event marker (EQEP_QEPSTS[FIDF]) in the EQEP_QEPSTS registers; it also remembers the quadrature edge on the first index marker so that the same relative quadrature transition is used for the software index marker (EQEP_QEPCTL[5:4] IEL=0b11).

Figure 21-65. Position Counter Underflow/Overflow (QPOSMAX = 4)



21.4.2.3.1.3 Position Counter Reset on the First Index Event (EQEP_QEPCTL[13:12] PCRM = 0b10)

If the index event occurs during forward movement, then the position counter is reset to 0 on the next eQEP clock. If the index event occurs during the reverse movement, then the position counter is reset to the value in the EQEP_QPOSMAX register on the next eQEP clock. Note that this is done only on the first occurrence and subsequently the position counter value is not reset on an index event; rather, it is reset based on maximum position as described in Section 21.4.2.3.1.2.

First index marker is defined as the quadrature edge following the first index edge. The eQEP peripheral records the occurrence of the first index marker (EQEP_QEPSTS[FIMF]) and direction on the first index event marker (EQEP_QEPSTS[FIDF]) in EQEP_QEPSTS registers. It also remembers the quadrature edge on the first index marker so that same relative quadrature transition is used for software index marker (EQEP_QEPCTL[5:4] IEL= 0b11).

21.4.2.3.1.4 Position Counter Reset on Unit Time out Event (EQEP_QEPCTL[13:12] PCRM = 0b11)

In this mode, the QPOSCNT value is latched to the EQEP_QPOSLAT register and then the QPOSCNT field is reset (to 0 or the QPOSMAX value in the EQEP_QPOSMAX register, depending on the direction mode selected by EQEP_QDECCTL[QSRC] bits on a unit time event). This is useful for frequency measurement.

21.4.2.3.2 eQEP Position Counter Latch

The eQEP index and strobe input can be configured to latch the position counter QPOSCNT (EQEP_QPOSCNT) into QPOSILAT (EQEP_QPOSILAT register) and QPOSSLAT (EQEP_QPOSSLAT register) bitfields, respectively, on occurrence of a definite event on these pins.

21.4.2.3.2.1 Index Event Latch

In some applications, it may not be desirable to reset the position counter on every index event and instead it may be required to operate the position counter in full 32-bit mode (EQEP_QEPCTL[13:12] PCRM = 0b01 and EQEP_QEPCTL[13:12] PCRM = 0b10 modes).

In such cases, the eQEP position counter can be configured to latch on the following events and direction information is recorded in the EQEP_QEPSTS[QDLF] bit on every index event marker.

- Latch on Rising edge (EQEP_QEPCTL[5:4] IEL = 0b01)
- Latch on Falling edge (EQEP_QEPCTL[5:4] IEL = 0b10)
- Latch on Index Event Marker (EQEP_QEPCTL[5:4] IEL = 0b11)

This is particularly useful as an error checking mechanism to check if the position counter accumulated the correct number of counts between index events. As an example, the 1000-line encoder must count 4000 times when moving in the same direction between the index events.

The index event latch interrupt flag (EQEP_QFLG[IEL]) is set when the position counter is latched to the EQEP_QPOSILAT register. The index event latch configuration bits (QEPCTZ[IEL]) are ignored when EQEP_QEPCTL[13:12] PCRM = 0b00.

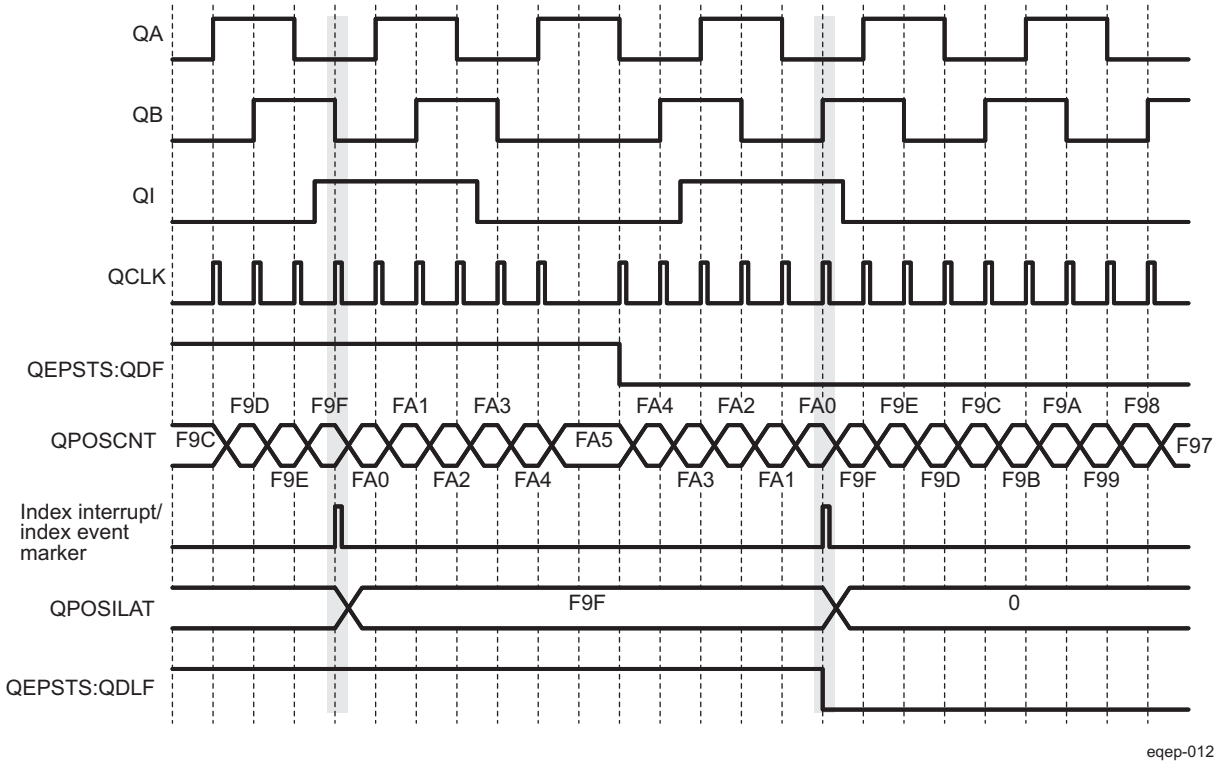
Latch on Rising Edge (EQEP_QEPCTL[5:4] IEL = 0b01)— The position counter value (QPOSCNT) is latched to the EQEP_QPOSILAT register on every rising edge of an index input.

Latch on Falling Edge (EQEP_QEPCTL[5:4] IEL = 0b10)— The position counter value (QPOSCNT) is latched to the EQEP_QPOSILAT register on every falling edge of index input.

Latch on Index Event Marker/Software Index Marker (EQEP_QEPCTL[5:4] IEL = 0b11)— The first index marker is defined as the quadrature edge following the first index edge. The eQEP peripheral records the occurrence of the first index marker (EQEP_QEPSTS[FIMF]) and direction on the first index event marker (EQEP_QEPSTS[FIDF]) in the EQEP_QEPSTS registers. It also remembers the quadrature edge on the first index marker so that same relative quadrature transition is used for latching the position counter (EQEP_QEPCTL[5:4] IEL = 0b11).

Figure 21-66 shows the position counter latch using an index event marker.

Figure 21-66. Software Index Marker for 1000-line Encoder (EQEP_QEPCTL[5:4] IEL = 0b01)



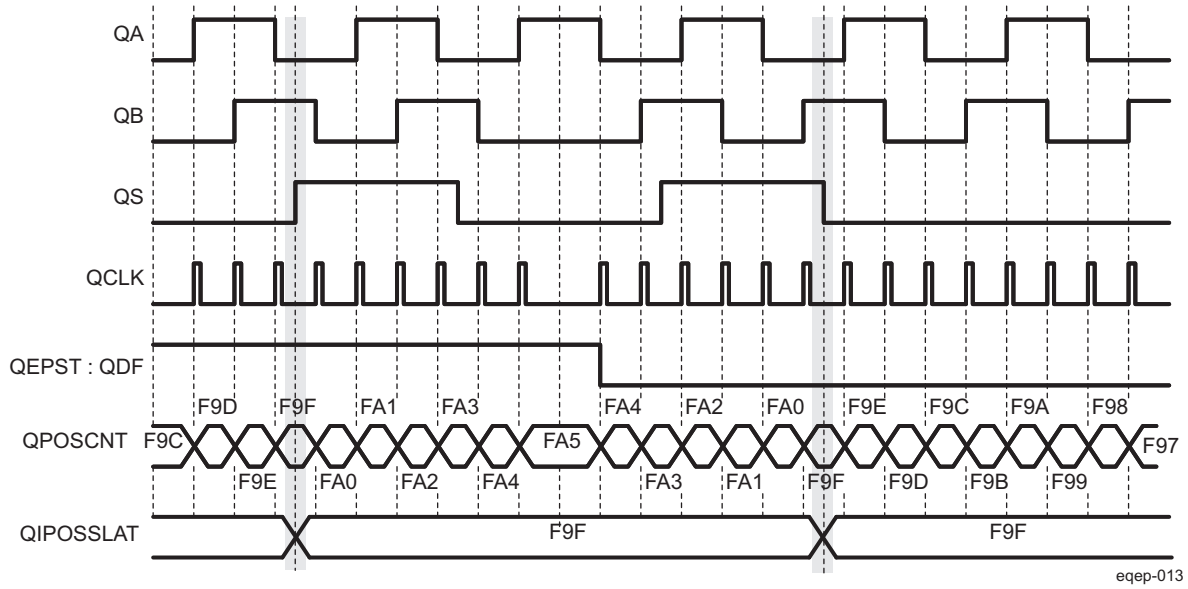
21.4.2.3.2.2 eQEP Strobe Event Latch

The position-counter value is latched to the EQEP_QPOSSLAT register on the rising edge of the strobe input by clearing the EQEP_QEPCTL[6] SEL bit.

If the EQEP_QEPCTL[6] SEL bit is set, then the position counter value is latched to the EQEP_QPOSSLAT register on the rising edge of the strobe input for forward direction and on the falling edge of the strobe input for reverse direction as shown in Figure 21-67.

The strobe event latch interrupt flag (EQEP_QFLG[SEL]) is set when the position counter is latched to the EQEP_QPOSSLAT register.

Figure 21-67. eQEP Strobe Event Latch (EQEP_QEPCTL[6] SEL = 0b1)



21.4.2.3.3 eQEP Position Counter Initialization

The position counter can be initialized using following events:

- Index event
- Strobe event
- Software initialization

Index Event Initialization (IEI)— The QEPI index input can be used to trigger the initialization of the position counter at the rising or falling edge of the index input.

If the EQEP_QEPCTL[9:8] IEI bits are 0b10, then the position counter (EQEP_QPOSCNT) is initialized with a value in the EQEP_QPOSINIT register on the rising edge of strobe input for forward direction and on the falling edge of strobe input for reverse direction.

The index event initialization interrupt flag (EQEP_QFLG[IEI]) is set when the position counter is initialized with a value in the EQEP_QPOSINIT register.

Strobe Event Initialization (SEI)— If the EQEP_QEPCTL[11:10] SEI bits are 0b10, then the position counter is initialized with a value in the EQEP_QPOSINIT register on the rising edge of strobe input.

If the EQEP_QEPCTL[11:10] SEI bits are 0b11, then the position counter (EQEP_QPOSCNT) is initialized with a value in the EQEP_QPOSINIT register on the rising edge of strobe input for forward direction and on the falling edge of strobe input for reverse direction.

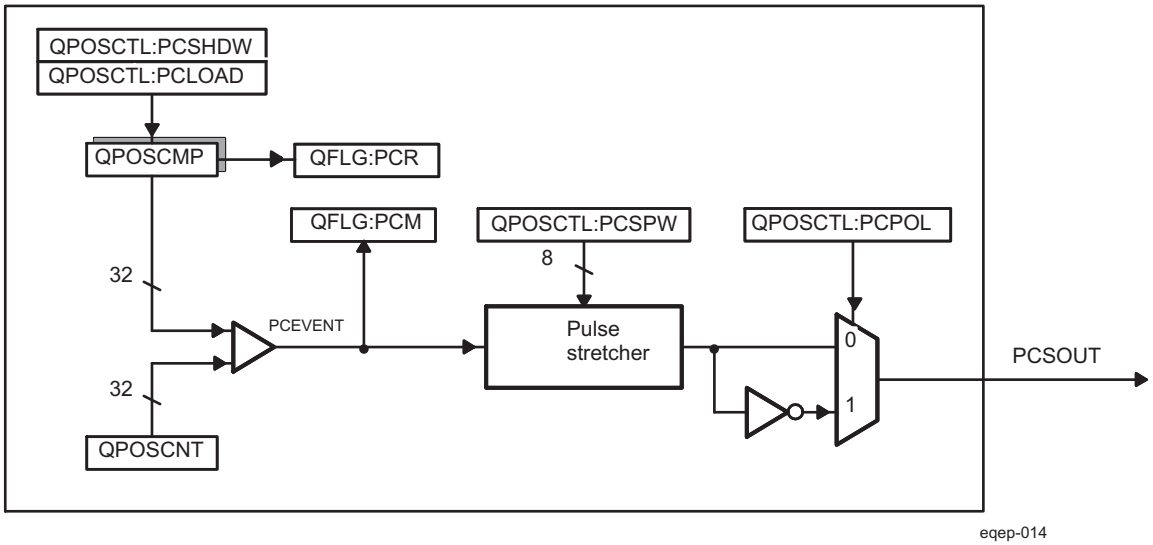
The strobe event initialization interrupt flag (EQEP_QFLG[SEI]) is set when the position counter is initialized with a value in the EQEP_QPOSINIT register.

Software Initialization (SWI)— The position counter can be initialized in software by writing a '1' to the EQEP_QEPCTL[7] SWI bit, which will automatically be cleared after initialization.

21.4.2.3.4 eQEP Position-Compare Unit

The eQEP peripheral includes a position-compare unit that is used to generate a sync output and/or interrupt on a position-compare match. Figure 21-68 shows a diagram. The position-compare (EQEP_QPOSCMP) register is shadowed and shadow mode can be enabled or disabled using the EQEP_QPOSCTL[PSSHDW] bit. If the shadow mode is not enabled, the CPU writes directly to the active position compare register.

Figure 21-68. eQEP Position-compare Unit



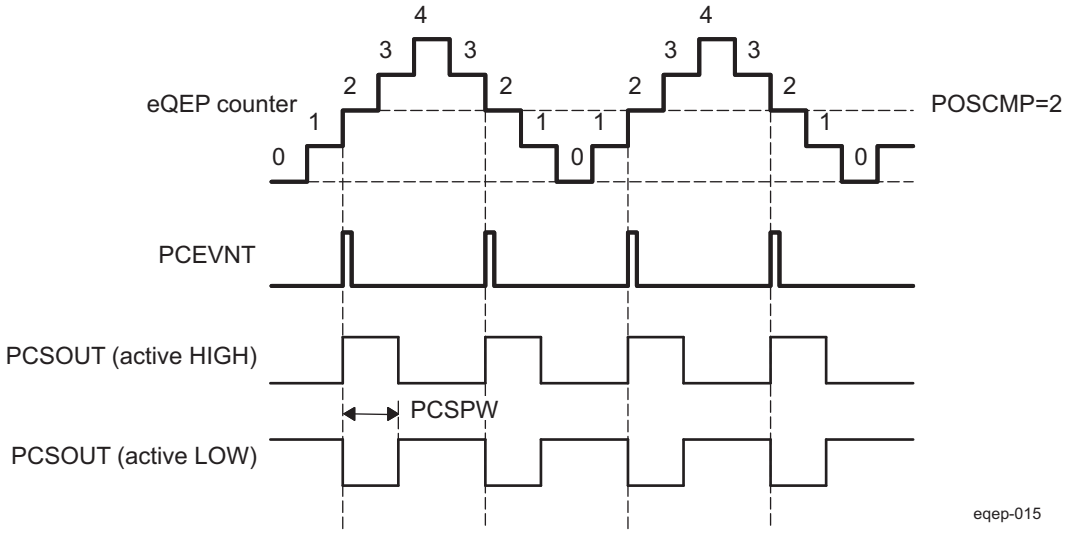
In shadow mode, you can configure the position-compare unit (EQEP_QPOSCTL[PCLOAD]) to load the shadow register value into the active register on the following events and to generate the position-compare ready (EQEP_QFLG[PCR]) interrupt after loading.

- Load on compare match
- Load on position-counter zero event

The position-compare match (EQEP_QFLG[PCM]) is set when the position-counter value (QPOSCNT) matches with the active position-compare register (EQEP_QPOSCMP) and the position-compare sync output of the programmable pulse width is generated on compare match to trigger an external device.

For example, if EQEP_QPOSCMP bitfield QPOSCMP = 0x2, the position-compare unit generates a position-compare event on 1 to 2 transitions of the eQEP position counter for forward counting direction and on 3 to 2 transitions of the eQEP position counter for reverse counting direction (see Figure 21-69).

Figure 21-69. eQEP Position-compare Event Generation Points



The pulse stretcher logic in the position-compare unit generates a programmable position-compare sync pulse output on the position-compare match. In the event of a new position-compare match while a previous position-compare pulse is still active, then the pulse stretcher generates a pulse of specified duration from the new position-compare event.

21.4.2.4 eQEP Edge Capture Unit

The eQEP peripheral includes an integrated edge capture unit to measure the elapsed time between the unit position events as shown in Figure 21-70. This feature is typically used for low speed measurement using the following equation:

$$V(k) = \frac{X}{t(k) - t(k-1)} = \frac{X}{\Delta T} \quad \text{eqep-017} \qquad (10)$$

where,

- X - Unit position is defined by integer multiple of quadrature edges (see Figure 21-71)
- ΔT - Elapsed time between unit position events
- v(k) - Velocity at time instant "k"

The eQEP capture timer (QCTMR bitfield in EQEP_QCTMR register) runs from prescaled SYSCLKOUT and the prescaler is programmed by the EQEP_QCAPCTL[CCPS] bits. The capture timer QCTMR value is latched into the capture period register (EQEP_QCPRD) on every unit position event and then the capture timer is reset, a flag is set in EQEP_QEPSTS[UPEVNT] to indicate that new value is latched into the EQEP_QCPRD register. Software can check this status flag before reading the period register for low speed measurement and clear the flag by writing 1.

NOTE: The system clock - SYSCLKOUT is the eQEP functional clock derived from the PWMSS gateable interface and functional clock PWMSS_GICLK, described in the Section 21.1.3.

Time measurement (ΔT) between unit position events will be correct if the following conditions are met:

- No more than 65,535 counts have occurred between unit position events.
- No direction change between unit position events.

The capture unit sets the eQEP overflow error flag (EQEP_QEPSTS[COEF]) in the event of capture timer overflow between unit position events. If a direction change occurs between the unit position events, then an error flag is set in the status register (EQEP_QEPSTS[CDEF]).

Capture Timer (EQEP_QCTMR register) and Capture period register (EQEP_QCPRD) can be configured to latch on following events.

- CPU read of EQEP_QPOSCNT register
- Unit time-out event

If the EQEP_QEPCTL[2] QCLM bit is cleared, then the capture timer and capture period values are latched into the EQEP_QCTMRLAT and EQEP_QCPRDLAT registers, respectively, when the CPU reads the position counter in EQEP_QPOSCNT.

If the EQEP_QEPCTL[2] QCLM bit is set, then the position counter, capture timer, and capture period values are latched into the EQEP_QPOSLAT, EQEP_QCTMRLAT and EQEP_QCPRDLAT registers, respectively, on unit time out.

Figure 21-72 shows the capture unit operation along with the position counter.

NOTE: The EQEP_QCAPCTL register should not be modified dynamically (such as switching CAPCLK prescaling mode from QCLK/4 to QCLK/8). The capture unit must be disabled before changing the prescaler.

Figure 21-70. eQEP Edge Capture Unit

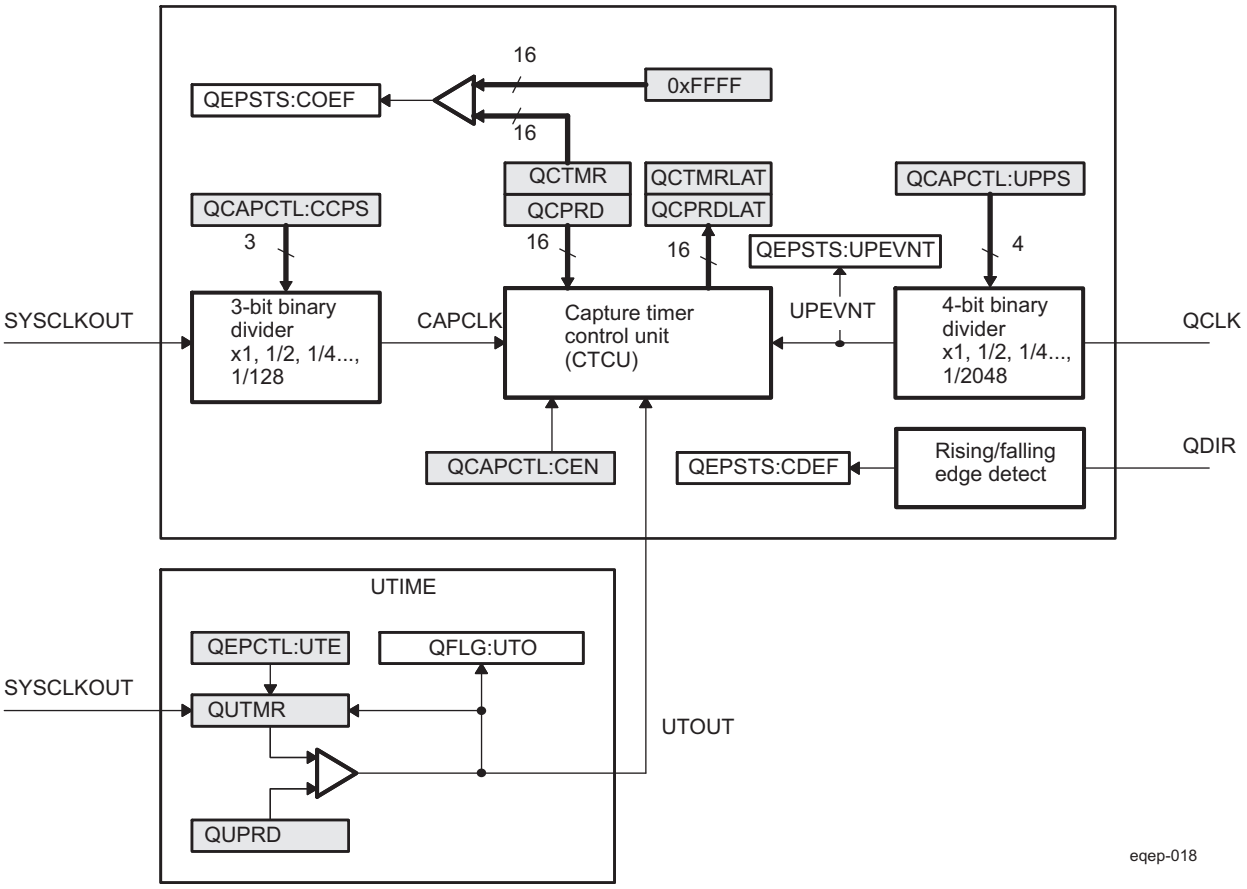
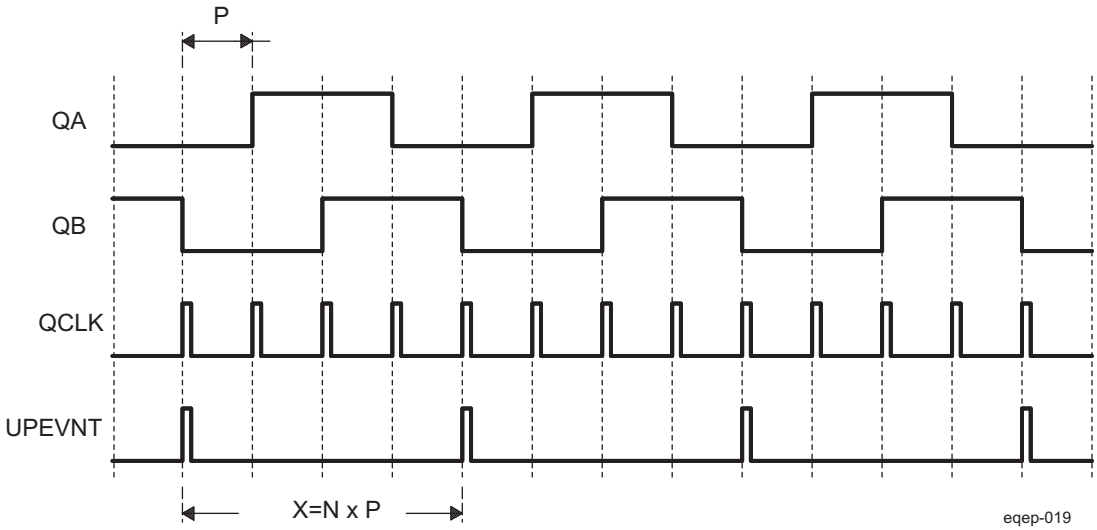
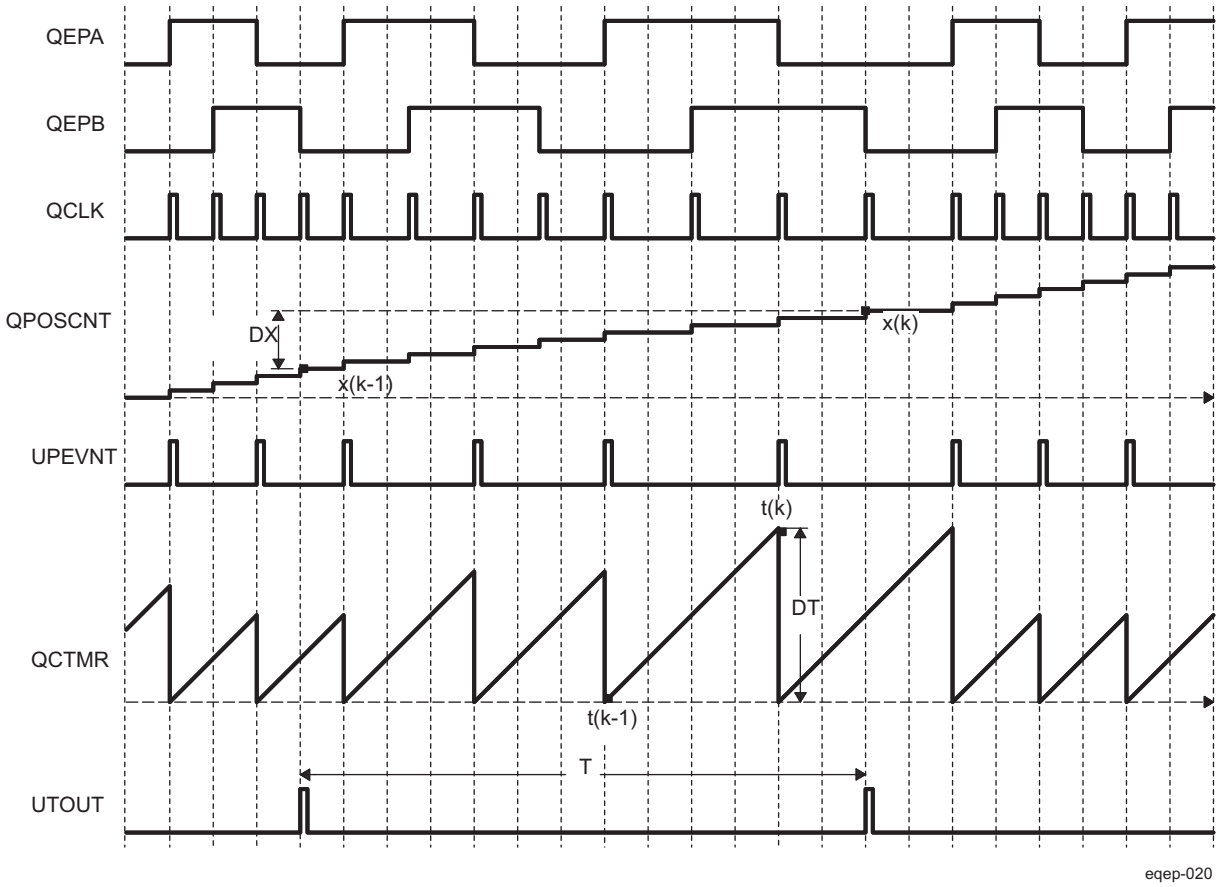


Figure 21-71. Unit Position Event for Low Speed Measurement (EQEP_QCAPCTL[UPPS] = 0010)



N - Number of quadrature periods selected using EQEP_QCAPCTL[UPPS] bits

Figure 21-72. eQEP Edge Capture Unit - Timing Details



Velocity Calculation Equations:

$$V(k) = \frac{x(k) - x(k-1)}{T} = \frac{\Delta X}{T} \quad \text{eqep\_021} \qquad (11)$$

where

v(k): Velocity at time instant k

x(k): Position at time instant k

x(k-1): Position at time instant k - 1

T: Fixed unit time or inverse of velocity calculation rate

ΔX: Incremental position movement in unit time

X: Fixed unit position

ΔT: Incremental time elapsed for unit position movement

t(k): Time instant "k"

t(k-1): Time instant "k - 1"

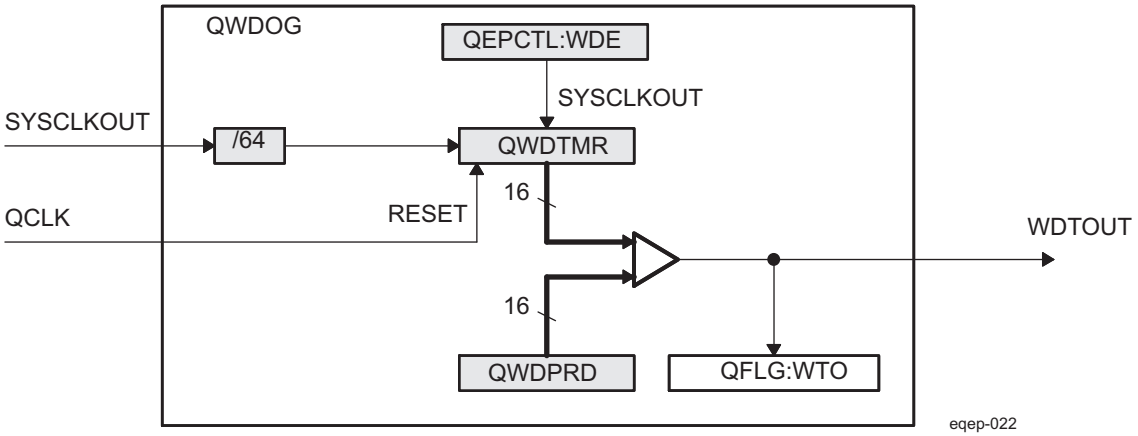
Unit time (T) and unit period (X) are configured using the EQEP_QUPRD and EQEP_QCAPCTL[UPPS] registers. Incremental position output and incremental time output is available in the EQEP_QPOSLAT and EQEP_QCPRDLAT registers.

Parameter	Relevant Register to Configure or Read the Information
T	Unit Period Register (EQEP_QUPRD)
ΔX	Incremental Position = QPOSLAT(k) - QPOSLAT(K - 1)
X	Fixed unit position defined by sensor resolution and ZCAPCTL[UPPS] bits
ΔT	Capture Period Latch (QCPRDLAT)

21.4.2.5 eQEP Watchdog

The eQEP peripheral contains a 16-bit watchdog timer that monitors the quadrature-clock to indicate proper operation of the motion-control system. The eQEP watchdog timer is clocked from SYSCLKOUT/64 and the quadrate clock event (pulse) resets the watchdog timer. If no quadrature-clock event is detected until a period match (QWDPRD = QWDTMR), then the watchdog timer will time out and the watchdog interrupt flag will be set (EQEP_QFLG[WTO]). The time-out value is programmable through the watchdog period register (EQEP_QWDPRD).

Figure 21-73. eQEP Watchdog Timer

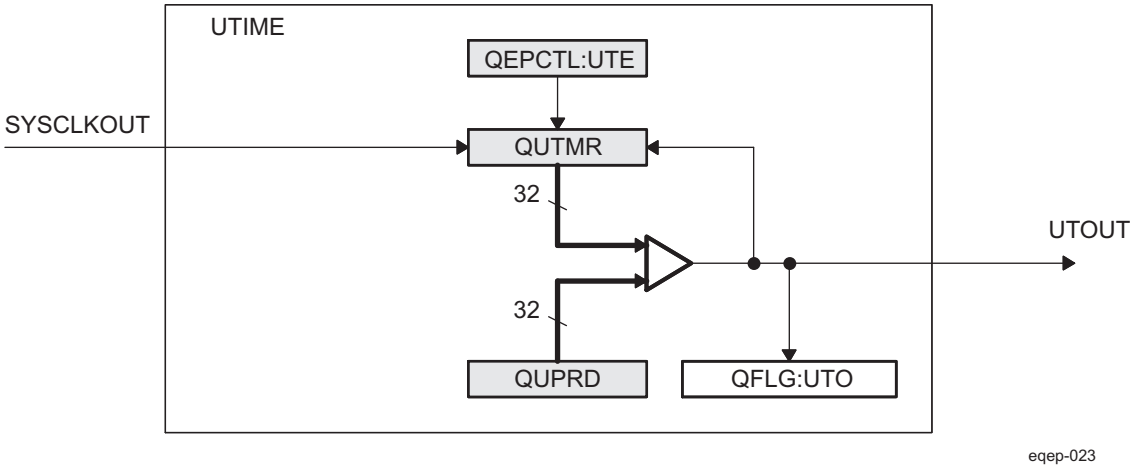


21.4.2.6 Unit Timer Base

The eQEP peripheral includes a 32-bit timer (QUTMR) that is clocked by SYSCLKOUT to generate periodic interrupts for velocity calculations. The unit time out interrupt is set (EQEP_QFLG[UTO]) when the unit timer (QUTMR) matches the unit period register (EQEP_QUPRD).

The eQEP peripheral can be configured to latch the position counter, capture timer, and capture period values on a unit time out event so that latched values are used for velocity calculation as described in Section 21.4.2.4.

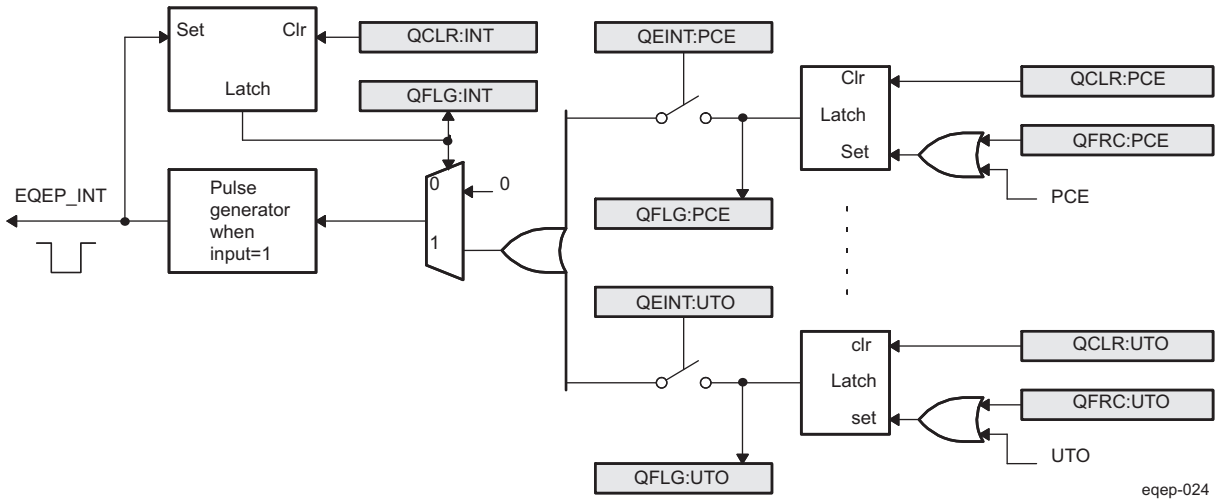
Figure 21-74. eQEP Unit Time Base



21.4.2.7 eQEP Interrupt Structure

Figure 21-75 shows how the interrupt mechanism works in the EQEP module.

Figure 21-75. EQEP Interrupt Generation



Eleven interrupt events (PCE, PHE, QDC, WTO, PCU, PCO, PCR, PCM, SEL, IEL, and UTO) can be generated. The interrupt control register (EQEP_QEINT) is used to enable/disable individual interrupt event sources. The interrupt flag register (EQEP_QFLG) indicates if any interrupt event has been latched and contains the global interrupt flag bit (INT). An interrupt pulse is generated only to the interrupt controller if any of the interrupt events is enabled, the flag bit is 1 and the INT flag bit is 0. The interrupt service routine will need to clear the global interrupt flag bit and the serviced event, via the interrupt clear register (EQEP_QCLR), before any other interrupt pulses are generated. You can force an interrupt event by way of the interrupt force register (EQEP_QFRC), which is useful for test purposes.

21.4.2.8 Summary of PWMSS eQEP Functional Registers

Table 21-145 lists the registers with their memory locations, sizes, and reset values.

Table 21-145. eQEP Control and Status Functional Registers

Offset	Acronym	Register Description	Size(×16)/ #shadow
0h	EQEP_QPOSCNT	eQEP Position Counter Register	2/0
4h	EQEP_QPOSINIT	eQEP Position Counter Initialization Register	2/0
8h	EQEP_QPOSMAX	eQEP Maximum Position Count Register	2/0
Ch	EQEP_QPOSCMP	eQEP Position-Compare Register	2/1
10h	EQEP_QPOSILAT	eQEP Index Position Latch Register	2/0
14h	EQEP_QPOSSLAT	eQEP Strobe Position Latch Register	2/0
18h	EQEP_QPOSLAT	eQEP Position Counter Latch Register	2/0
1Ch	EQEP_QUTMR	eQEP Unit Timer Register	2/0
20h	EQEP_QUPRD	eQEP Unit Period Register	2/0
24h	EQEP_QWDTMR	eQEP Watchdog Timer Register	1/0
26h	EQEP_QWDPRD	eQEP Watchdog Period Register	1/0
28h	EQEP_QDECCTL	eQEP Decoder Control Register	1/0
2Ah	EQEP_QEPCTL	eQEP Control Register	1/0
2Ch	EQEP_QCAPCTL	eQEP Capture Control Register	1/0
2Eh	EQEP_QPOSCTL	eQEP Position-Compare Control Register	1/0
30h	EQEP_QEINT	eQEP Interrupt Enable Register	1/0
32h	EQEP_QFLG	eQEP Interrupt Flag Register	1/0
34h	EQEP_QCLR	eQEP Interrupt Clear Register	1/0
36h	EQEP_QFRC	eQEP Interrupt Force Register	1/0
38h	EQEP_QEPSTS	eQEP Status Register	1/0
3Ah	EQEP_QCTMR	eQEP Capture Timer Register	1/0
3Ch	EQEP_QCPRD	eQEP Capture Period Register	1/0
3Eh	EQEP_QCTMRLAT	eQEP Capture Timer Latch Register	1/0
40h	EQEP_QCPRDLAT	eQEP Capture Period Latch Register	1/0
5Ch	EQEP_REVID	eQEP Revision ID Register	2/0

21.4.3 PWMSS_EQEP Register Manual

This section provides description of the PWMSS eQEP relevant functional registers.

21.4.3.1 PWMSS_EQEP Instance Summary

Table 21-146. PWMSS_EQEP Instance Summary

Module Name	Module Base Address L4_PER2 Interconnect	Size (Bytes)
PWMSS_EQEP	0x4843 E180	116 Bytes

21.4.3.2 PWMSS_EQEP Registers

21.4.3.2.1 PWMSS_EQEP Register Summary

Table 21-147. PWMSS_EQEP Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	PWMSS_EQEP Physical Address L4_PER2 Interconnect
EQEP_QPOSCNT	RW	32	0x0	0x4843 E180
EQEP_QPOSINIT	RW	32	0x4	0x4843 E184
EQEP_QPOSMAX	RW	32	0x8	0x4843 E188

Table 21-147. PWMSS_EQEP Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	PWMSS_EQEP Physical Address L4_PER2 Interconnect
EQEP_QPOSCMP	RW	32	0xC	0x4843 E18C
EQEP_QPOSILAT	R	32	0x10	0x4843 E190
EQEP_QPOSSLAT	R	32	0x14	0x4843 E194
EQEP_QPOSLAT	R	32	0x18	0x4843 E198
EQEP_QUTMR	RW	32	0x1C	0x4843 E19C
EQEP_QUPRD	RW	32	0x20	0x4843 E1A0
EQEP_QWDTMR	RW	16	0x24	0x4843 E1A4
EQEP_QWDPRD	RW	16	0x26	0x4843 E1A6
EQEP_QDECCTL	RW	16	0x28	0x4843 E1A8
EQEP_QEPCTL	RW	16	0x2A	0x4843 E1AA
EQEP_QCAPCTL	RW	16	0x2C	0x4843 E1AC
EQEP_QPOSCTL	RW	16	0x2E	0x4843 E1AE
EQEP_QEINT	RW	16	0x30	0x4843 E1B0
EQEP_QFLG	R	16	0x32	0x4843 E1B2
EQEP_QCLR	RW	16	0x34	0x4843 E1B4
EQEP_QFRC	RW	16	0x36	0x4843 E1B6
EQEP_QEPSTS	RW	16	0x38	0x4843 E1B8
EQEP_QCTMR	RW	16	0x3A	0x4843 E1BA
EQEP_QCPRD	RW	16	0x3C	0x4843 E1BC
EQEP_QCTMRLAT	R	16	0x3E	0x4843 E1BE
EQEP_QCPRDLAT	RW	16	0x40	0x4843 E1C0
EQEP_REVID	R	32	0x5C	0x4843 E1DC

21.4.3.2.2 PWMSS_EQEP Register Description

Table 21-148. EQEP_QPOSCNT

Address offset	0x0		
Physical Address	0x4843 E180	**Instance**	PWMSS_EQEP
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
QPOSCNT																															

Bits	Field Name	Description	Type	Reset
31:0	QPOSCNT	This 32 bit position counter register counts up/down on every eQEP pulse based on direction input. This counter acts as a position integrator whose count value is proportional to position from a give reference point.	RW	0x0

Table 21-149. Register Call Summary for Register EQEP_QPOSCNT

Enhanced Quadrature Encoder Pulse (eQEP) Module
- eQEP Quadrature Decoder Unit (QDU): [0]
- eQEP Position Counter and Control Unit (PCCU): [1][2][3]
- eQEP Edge Capture Unit: [4][5]
- Summary of PWMSS eQEP Functional Registers: [6]
- PWMSS_EQEP Registers: [7][8][9][10]

Table 21-150. EQEP_QPOSINIT

Address offset	0x4		
Physical Address	0x4843 E184	**Instance**	PWMSS_EQEP
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
QPOSINIT																															

Bits	Field Name	Description	Type	Reset
31:0	QPOSINIT	This register contains the position value that is used to initialize the position counter based on external strobe or index event. The position counter can be initialized through software.	RW	0x0

Table 21-151. Register Call Summary for Register EQEP_QPOSINIT

Enhanced Quadrature Encoder Pulse (eQEP) Module
- eQEP Position Counter and Control Unit (PCCU): [0][1][2][3][4]
- Summary of PWMSS eQEP Functional Registers: [5]
- PWMSS_EQEP Registers: [6]

Table 21-152. EQEP_QPOSMAX

Address offset	0x8		
Physical Address	0x4843 E188	**Instance**	PWMSS_EQEP
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
QPOSMAX																															

Bits	Field Name	Description	Type	Reset
31:0	QPOSMAX	This register contains the maximum position counter value.	RW	0x0

Table 21-153. Register Call Summary for Register EQEP_QPOSMAX

Enhanced Quadrature Encoder Pulse (eQEP) Module

- eQEP Position Counter and Control Unit (PCCU): [0][1][2][3][4]
- Summary of PWMSS eQEP Functional Registers: [5]
- PWMSS_EQEP Registers: [6]

Table 21-154. EQEP_QPOSCMP

Address offset	0xC		
Physical Address	0x4843 E18C	**Instance**	PWMSS_EQEP
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
QPOSCMP																															

Bits	Field Name	Description	Type	Reset
31:0	QPOSCMP	The position-compare value in this register is compared with the position counter (QPOSCNT field in EQEP_QPOSCNT) to generate sync output and/or interrupt on compare match.	RW	0x0

Table 21-155. Register Call Summary for Register EQEP_QPOSCMP

Enhanced Quadrature Encoder Pulse (eQEP) Module

- eQEP Quadrature Decoder Unit (QDU): [0]
- eQEP Position Counter and Control Unit (PCCU): [1][2][3]
- Summary of PWMSS eQEP Functional Registers: [4]
- PWMSS_EQEP Registers: [5]

Table 21-156. EQEP_QPOSILAT

Address offset	0x10		
Physical Address	0x4843 E190	**Instance**	PWMSS_EQEP
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
QPOSILAT																															

Bits	Field Name	Description	Type	Reset
31:0	QPOSILAT	The position-counter value is latched into this register on an index event as defined by the QEPCTL[IEL] bits.	R	0x0

Table 21-157. Register Call Summary for Register EQEP_QPOSILAT

Enhanced Quadrature Encoder Pulse (eQEP) Module
- eQEP Position Counter and Control Unit (PCCU): [0][1][2][3][4]
- Summary of PWMSS eQEP Functional Registers: [5]
- PWMSS_EQEP Registers: [6][7]

Table 21-158. EQEP_QPOSSLAT

Address offset	0x14		
Physical Address	0x4843 E194	**Instance**	PWMSS_EQEP
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
QPOSSLAT																															

Bits	Field Name	Description	Type	Reset
31:0	QPOSSLAT	The position-counter value is latched into this register on strobe event as defined by the QEPCTL[SEL] bits.	R	0x0

Table 21-159. Register Call Summary for Register EQEP_QPOSSLAT

Enhanced Quadrature Encoder Pulse (eQEP) Module
- eQEP Position Counter and Control Unit (PCCU): [0][1][2][3]
- Summary of PWMSS eQEP Functional Registers: [4]
- PWMSS_EQEP Registers: [5][6]

Table 21-160. EQEP_QPOSLAT

Address offset	0x18		
Physical Address	0x4843 E198	**Instance**	PWMSS_EQEP
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
QPOSLAT																															

Bits	Field Name	Description	Type	Reset
31:0	QPOSLAT	The position-counter value is latched into this register on unit time out event.	R	0x0

Table 21-161. Register Call Summary for Register EQEP_QPOSLAT

Enhanced Quadrature Encoder Pulse (eQEP) Module
- eQEP Position Counter and Control Unit (PCCU): [0]
- eQEP Edge Capture Unit: [1][2]
- Summary of PWMSS eQEP Functional Registers: [3]
- PWMSS_EQEP Registers: [4][5]

Table 21-162. EQEP_QUTMR

Address offset	0x1C		
Physical Address	0x4843 E19C	**Instance**	PWMSS_EQEP
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
QUTMR																															

Bits	Field Name	Description	Type	Reset
31:0	QUTMR	This register acts as time base for unit time event generation. When this timer value matches with unit time period value, unit time event is generated.	RW	0x0

Table 21-163. Register Call Summary for Register EQEP_QUTMR

Enhanced Quadrature Encoder Pulse (eQEP) Module
- Summary of PWMSS eQEP Functional Registers: [0]
- PWMSS_EQEP Registers: [1]

Table 21-164. EQEP_QUPRD

Address offset	0x20		
Physical Address	0x4843 E1A0	**Instance**	PWMSS_EQEP
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
QUPRD																															

Bits	Field Name	Description	Type	Reset
31:0	QUPRD	This register contains the period count for unit timer to generate periodic unit time events to latch the eQEP position information at periodic interval and optionally to generate interrupt.	RW	0x0

Table 21-165. Register Call Summary for Register EQEP_QUPRD

Enhanced Quadrature Encoder Pulse (eQEP) Module
- eQEP Edge Capture Unit: [0][1]
- Unit Timer Base: [2]
- Summary of PWMSS eQEP Functional Registers: [3]
- PWMSS_EQEP Registers: [4]

Table 21-166. EQEP_QWDTMR

Address offset	0x24		
Physical Address	0x4843 E1A4	**Instance**	PWMSS_EQEP
Description			
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
QWDTMR															

Bits	Field Name	Description	Type	Reset
15:0	QWDTMR	This register acts as time base for watch dog to detect motor stalls. When this timer value matches with watch dog period value, watch dog timeout interrupt is generated. This register is reset upon edge transition in quadrature-clock indicating the motion.	RW	0x0

Table 21-167. Register Call Summary for Register EQEP_QWDTMR

Enhanced Quadrature Encoder Pulse (eQEP) Module

- Summary of PWMSS eQEP Functional Registers: [0]
- PWMSS_EQEP Registers: [1]

Table 21-168. EQEP_QWDPRD

Address offset	0x26		
Physical Address	0x4843 E1A6	**Instance**	PWMSS_EQEP
Description			
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
QWDPRD															

Bits	Field Name	Description	Type	Reset
15:0	QWDPRD	This register contains the time-out count for the eQEP peripheral watch dog timer. When the watchdog timer value matches the watchdog period value, a watchdog timeout interrupt is generated.	RW	0x0

Table 21-169. Register Call Summary for Register EQEP_QWDPRD

Enhanced Quadrature Encoder Pulse (eQEP) Module

- eQEP Watchdog: [0]
- Summary of PWMSS eQEP Functional Registers: [1]
- PWMSS_EQEP Registers: [2]

Table 21-170. EQEP_QDECCTL

Address offset	0x28		
Physical Address	0x4843 E1A8	**Instance**	PWMSS_EQEP
Description			
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
QSRC		SOEN	SPSEL	XCR	SWAP	IGATE	QAP	QBP	QIP	QSP	RESERVED				

Bits	Field Name	Description	Type	Reset
15:14	QSRC	Position-counter source selection. 0x0 = Quadrature count mode (QCLK = iCLK, QDIR = iDIR) 0x1 = Direction-count mode (QCLK = xCLK, QDIR = xDIR) 0x2 = UP count mode for frequency measurement (QCLK = xCLK, QDIR = 1) 0x3 = DOWN count mode for frequency measurement (QCLK = xCLK, QDIR = 0)	RW	0x0
13	SOEN	Sync output-enable 0x0 = Disable position-compare sync output 0x1 = Enable position-compare sync output	RW	0x0
12	SPSEL	Sync output pin selection 0x0 = Index pin is used for sync output 0x1 = Strobe pin is used for sync output	RW	0x0
11	XCR	External clock rate 0x0 = 2x resolution: Count the rising/falling edge 0x1 = 1x resolution: Count the rising edge only	RW	0x0
10	SWAP	Swap quadrature clock inputs. This swaps the input to the quadrature decoder, reversing the counting direction. 0x0 = Quadrature-clock inputs are not swapped 0x1 = Quadrature-clock inputs are swapped	RW	0x0
9	IGATE	Index pulse gating option 0x0 = Disable gating of Index pulse 0x1 = Gate the index pin with strobe	RW	0x0
8	QAP	QEPA input polarity 0x0 = No effect 0x1 = Negates QEPA input	RW	0x0
7	QBP	QEPB input polarity 0x0 = No effect 0x1 = Negates QEPB input	RW	0x0
6	QIP	QEPI input polarity 0x0 = No effect 0x1 = Negates QEPI input	RW	0x0
5	QSP	QEPS input polarity 0x0 = No effect 0x1 = Negates QEPS input	RW	0x0
4:0	RESERVED		R	0x0

Table 21-171. Register Call Summary for Register EQEP_QDECCTL

Enhanced Quadrature Encoder Pulse (eQEP) Module

- eQEP Quadrature Decoder Unit (QDU): [0][1][2][3][4][5][6][7]
- eQEP Position Counter and Control Unit (PCCU): [8]
- Summary of PWMSS eQEP Functional Registers: [9]
- PWMSS_EQEP Registers: [10][11]

Table 21-172. EQEP_QEPCTL

Address offset	0x2A		
Physical Address	0x4843 E1AA	**Instance**	PWMSS_EQEP
Description			
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FREE_SOFT		PCRM		SEI		IEI		SWI	SEL	IEL		PHEN	QCLM	UTE	WDE

Bits	Field Name	Description	Type	Reset
15:14	FREE_SOFT	Emulation Control Bits. In the values 0 through 3 listed below, x is different for the four following behaviors. EQEP_QPOSCNT behavior, x refers to the Position counter. QWDTMR behavior, x refers to the Watchdog counter. QUTMR behavior, x refers to the Unit timer. QCTMR behavior, x refers to the Capture timer. 0x0 = x stops immediately. For QPOSCNT behavior, the stop is on emulation suspend. 0x1 = x continues to count until the rollover. 0x2 = x is unaffected by emulation suspend. 0x3 = x is unaffected by emulation suspend.	RW	0x0
13:12	PCRM	Position counter reset mode 0x0 = Position counter reset on an index event 0x1 = Position counter reset on the maximum position 0x2 = Position counter reset on the first index event 0x3 = Position counter reset on a unit time event	RW	0x0
11:10	SEI	Strobe event initialization of position counter 0x0 = Does nothing (action disabled) 0x1 = Does nothing (action disabled) 0x2 = Initializes the position counter on rising edge of the QEPS signal 0x3 = Clockwise Direction: Initializes the position counter on the rising edge of QEPS strobe. Counter Clockwise Direction: Initializes the position counter on the falling edge of QEPS strobe	RW	0x0
9:8	IEI	Index event initialization of position counter 0x0 = Do nothing (action disabled) 0x1 = Do nothing (action disabled) 0x2 = Initializes the position counter on the rising edge of the QEPI signal (QPOSCNT = QPOSINIT) 0x3 = Initializes the position counter on the falling edge of QEPI signal (QPOSCNT = QPOSINIT)	RW	0x0
7	SWI	Software initialization of position counter 0x0 = Do nothing (action disabled) 0x1 = Initialize position counter, this bit is cleared automatically	RW	0x0

Bits	Field Name	Description	Type	Reset
6	SEL	Strobe event latch of position counter 0x0 = The position counter is latched on the rising edge of QEPS strobe (QPOSSLAT = POSCCNT). Latching on the falling edge can be done by inverting the strobe input using the QSP bit in the EQEP_QDECCTL register. 0x1 = Clockwise Direction: Position counter is latched on rising edge of QEPS strobe. Counter Clockwise Direction: Position counter is latched on falling edge of QEPS strobe.	RW	0x0
5:4	IEL	Index event latch of position counter (software index marker) 0x0 = Reserved 0x1 = Latches position counter on rising edge of the index signal 0x2 = Latches position counter on falling edge of the index signal 0x3 = Software index marker. Latches the position counter and quadrature direction flag on index event marker. The position counter is latched to the EQEP_QPOSILAT register and the direction flag is latched in the EQEP_QEPSTS[QDLF] bit. This mode is useful for software index marking.	RW	0x0
3	PHEN	Quadrature position counter enable/software reset 0x0 = Reset the eQEP peripheral internal operating flags/read-only registers. Control/configuration registers are not disturbed by a software reset. 0x1 = eQEP position counter is enabled	RW	0x0
2	QCLM	eQEP capture latch mode 0x0 = Latch on position counter read by CPU. Capture timer and capture period values are latched into EQEP_QCTMRLAT and EQEP_QCPRDLAT registers when CPU reads the EQEP_QPOSCNT register. 0x1 = Latch on unit time out. Position counter, capture timer and capture period values are latched into EQEP_QPOSLAT, EQEP_QCTMRLAT and EQEP_QCPRDLAT registers on unit time out.	RW	0x0
1	UTE	eQEP unit timer enable 0x0 = Disable eQEP unit timer 0x1 = Enable unit timer	RW	0x0
0	WDE	eQEP watchdog enable 0x0 = Disable the eQEP watchdog timer 0x1 = Enable the eQEP watchdog timer	RW	0x0

Table 21-173. Register Call Summary for Register EQEP_QEPCTL

Enhanced Quadrature Encoder Pulse (eQEP) Module

- eQEP Position Counter and Control Unit (PCCU): [0][1][2][3][4][5][6][7][8][9][10][11][12][13][14][15][16]
- eQEP Edge Capture Unit: [17][18]
- Summary of PWMSS eQEP Functional Registers: [19]
- PWMSS_EQEP Registers: [20]

Table 21-174. EQEP_QCAPCTL

Address offset	0x2C		
Physical Address	0x4843 E1AC	**Instance**	PWMSS_EQEP
Description			
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CEN	RESERVED							CCPS				UPPS			

Bits	Field Name	Description	Type	Reset
15	CEN	Enable eQEP capture 0x0 = eQEP capture unit is disabled 0x1 = eQEP capture unit is enabled	RW	0x0
14:7	RESERVED		R	0x0
6:4	CCPS	eQEP capture timer clock prescaler 0x0 = CAPCLK = SYSCLKOUT/1 0x1 = CAPCLK = SYSCLKOUT/2 0x2 = CAPCLK = SYSCLKOUT/4 0x3 = CAPCLK = SYSCLKOUT/8 0x4 = CAPCLK = SYSCLKOUT/16 0x5 = CAPCLK = SYSCLKOUT/32 0x6 = CAPCLK = SYSCLKOUT/64 0x7 = CAPCLK = SYSCLKOUT/128	RW	0x0
3:0	UPPS	Unit position event prescaler 0x0 = UPEVNT = QCLK/1 0x1 = UPEVNT = QCLK/2 0x2 = UPEVNT = QCLK/4 0x3 = UPEVNT = QCLK/8 0x4 = UPEVNT = QCLK/16 0x5 = UPEVNT = QCLK/32 0x6 = UPEVNT = QCLK/64 0x7 = UPEVNT = QCLK/128 0x8 = UPEVNT = QCLK/256 0x9 = UPEVNT = QCLK/512 0xA = UPEVNT = QCLK/1024 0xB = UPEVNT = QCLK/2048 0xC = Reserved 0xD = Reserved 0xE = Reserved 0xF = Reserved	RW	0x0

Table 21-175. Register Call Summary for Register EQEP_QCAPCTL

Enhanced Quadrature Encoder Pulse (eQEP) Module

- eQEP Edge Capture Unit: [0][1][2][3]
- Summary of PWMSS eQEP Functional Registers: [4]
- PWMSS_EQEP Registers: [5]

Table 21-176. EQEP_QPOSCTL

Address offset	0x2E		
Physical Address	0x4843 E1AE	**Instance**	PWMSS_EQEP
Description			
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
PCSHDW	PCLOAD	PCPOL	PCE	PCSPW											

Bits	Field Name	Description	Type	Reset
15	PCSHDW	Position-compare shadow enable 0x0 = Shadow disabled, load Immediate 0x1 = Shadow enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
14	PCLOAD	Position-compare shadow load mode 0x0 = Load on QPOSCNT = 0 0x1 = Load when QPOSCNT = QPOSCMP	RW	0x0
13	PCPOL	Polarity of sync output 0x0 = Active HIGH pulse output 0x1 = Active LOW pulse output	RW	0x0
12	PCE	Position-compare enable/disable 0x0 = Disable position compare unit 0x1 = Enable position compare unit	RW	0x0
11:0	PCSPW	Select-position-compare sync output pulse width ... 0x0 = 1 x 4 x SYSCLKOUT cycles 0x1 = 2 x 4 x SYSCLKOUT cycles 0x2 = 3 x 4 x SYSCLKOUT cycles to 4096 x 4 x SYSCLKOUT cycles 0xFFF = 3 x 4 x SYSCLKOUT cycles to 4096 x 4 x SYSCLKOUT cycles	RW	0x0

Table 21-177. Register Call Summary for Register EQEP_QPOSCTL

Enhanced Quadrature Encoder Pulse (eQEP) Module

- eQEP Position Counter and Control Unit (PCCU): [0][1][2]
- Summary of PWMSS eQEP Functional Registers: [3]
- PWMSS_EQEP Registers: [4]

Table 21-178. EQEP_QEINT

Address offset	0x30		
Physical Address	0x4843 E1B0	**Instance**	PWMSS_EQEP
Description			
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				UTO	IEL	SEL	PCM	PCR	PCO	PCU	WTO	QDC	PHE	PCE	RESERVED

Bits	Field Name	Description	Type	Reset
15:12	RESERVED		R	0x0
11	UTO	Unit time out interrupt enable 0x0 = Interrupt is disabled 0x1 = Interrupt is enabled	RW	0x0
10	IEL	Index event latch interrupt enable 0x0 = Interrupt is disabled 0x1 = Interrupt is enabled	RW	0x0
9	SEL	Strobe event latch interrupt enable 0x0 = Interrupt is disabled 0x1 = Interrupt is enabled	RW	0x0
8	PCM	Position-compare match interrupt enable 0x0 = Interrupt is disabled 0x1 = Interrupt is enabled	RW	0x0

Bits	Field Name	Description	Type	Reset
7	PCR	Position-compare ready interrupt enable 0x0 = Interrupt is disabled 0x1 = Interrupt is enabled	RW	0x0
6	PCO	Position counter overflow interrupt enable 0x0 = Interrupt is disabled 0x1 = Interrupt is enabled	RW	0x0
5	PCU	Position counter underflow interrupt enable 0x0 = Interrupt is disabled 0x1 = Interrupt is enabled	RW	0x0
4	WTO	Watchdog time out interrupt enable 0x0 = Interrupt is disabled 0x1 = Interrupt is enabled	RW	0x0
3	QDC	Quadrature direction change interrupt enable 0x0 = Interrupt is disabled 0x1 = Interrupt is enabled	RW	0x0
2	PHE	Quadrature phase error interrupt enable 0x0 = Interrupt is disabled 0x1 = Interrupt is enabled	RW	0x0
1	PCE	Position counter error interrupt enable 0x0 = Interrupt is disabled 0x1 = Interrupt is enabled	RW	0x0
0	RESERVED		R	0x0

Table 21-179. Register Call Summary for Register EQEP_QEINT

Enhanced Quadrature Encoder Pulse (eQEP) Module

- eQEP Interrupt Structure: [0]
- Summary of PWMSS eQEP Functional Registers: [1]
- PWMSS_EQEP Registers: [2]

Table 21-180. EQEP_QFLG

Address offset	0x32		
Physical Address	0x4843 E1B2	**Instance**	PWMSS_EQEP
Description			
Type	R		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				UTO	IEL	SEL	PCM	PCR	PCO	PCU	WTO	QDC	PHE	PCE	INT

Bits	Field Name	Description	Type	Reset
15:12	RESERVED		R	0x0
11	UTO	Unit time out interrupt flag 0x0 = No interrupt generated 0x1 = Set by eQEP unit timer period match	R	0x0
10	IEL	Index event latch interrupt flag 0x0 = No interrupt generated 0x1 = This bit is set after latching the QPOSCNT to QPOSILAT	R	0x0
9	SEL	Strobe event latch interrupt flag 0x0 = No interrupt generated 0x1 = This bit is set after latching the QPOSCNT to EQEP_QPOSSLAT	R	0x0

Bits	Field Name	Description	Type	Reset
8	PCM	eQEP compare match event interrupt flag 0x0 = No interrupt generated 0x1 = This bit is set on position-compare match	R	0x0
7	PCR	Position-compare ready interrupt flag 0x0 = No interrupt generated 0x1 = This bit is set after transferring the shadow register value to the active position compare register.	R	0x0
6	PCO	Position counter overflow interrupt flag 0x0 = No interrupt generated 0x1 = This bit is set on position counter overflow.	R	0x0
5	PCU	Position counter underflow interrupt flag 0x0 = No interrupt generated 0x1 = This bit is set on position counter underflow.	R	0x0
4	WTO	Watchdog timeout interrupt flag 0x0 = No interrupt generated 0x1 = Set by watch dog timeout	R	0x0
3	QDC	Quadrature direction change interrupt flag 0x0 = No interrupt generated 0x1 = This bit is set during change of direction	R	0x0
2	PHE	Quadrature phase error interrupt flag 0x0 = No interrupt generated 0x1 = Set on simultaneous transition of QEPA and QEPB	R	0x0
1	PCE	Position counter error interrupt flag 0x0 = No interrupt generated 0x1 = Position counter error	R	0x0
0	INT	Global interrupt status flag 0x0 = No interrupt generated 0x1 = Interrupt was generated	R	0x0

Table 21-181. Register Call Summary for Register EQEP_QFLG

Enhanced Quadrature Encoder Pulse (eQEP) Module

- eQEP Quadrature Decoder Unit (QDU): [0]
- eQEP Position Counter and Control Unit (PCCU): [1][2][3][4][5][6][7][8][9]
- eQEP Watchdog: [10]
- Unit Timer Base: [11]
- eQEP Interrupt Structure: [12]
- Summary of PWMSS eQEP Functional Registers: [13]
- PWMSS_EQEP Registers: [14]

Table 21-182. EQEP_QCLR

Address offset	0x34		
Physical Address	0x4843 E1B4	**Instance**	PWMSS_EQEP
Description			
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				UTO	IEL	SEL	PCM	PCR	PCO	PCU	WTO	QDC	PHE	PCE	INT

Bits	Field Name	Description	Type	Reset
15:12	RESERVED		R	0x0
11	UTO	Clear unit time out interrupt flag 0x0 = No effect 0x1 = Clears the interrupt flag	RW	0x0
10	IEL	Clear index event latch interrupt flag 0x0 = No effect 0x1 = Clears the interrupt flag	RW	0x0
9	SEL	Clear strobe event latch interrupt flag 0x0 = No effect 0x1 = Clears the interrupt flag	RW	0x0
8	PCM	Clear eQEP compare match event interrupt flag 0x0 = No effect 0x1 = Clears the interrupt flag	RW	0x0
7	PCR	Clear position-compare ready interrupt flag 0x0 = No effect 0x1 = Clears the interrupt flag	RW	0x0
6	PCO	Clear position counter overflow interrupt flag 0x0 = No effect 0x1 = Clears the interrupt flag	RW	0x0
5	PCU	Clear position counter underflow interrupt flag 0x0 = No effect 0x1 = Clears the interrupt flag	RW	0x0
4	WTO	Clear watchdog timeout interrupt flag 0x0 = No effect 0x1 = Clears the interrupt flag	RW	0x0
3	QDC	Clear quadrature direction change interrupt flag 0x0 = No effect 0x1 = Clears the interrupt flag	RW	0x0
2	PHE	Clear quadrature phase error interrupt flag 0x0 = No effect 0x1 = Clears the interrupt flag	RW	0x0
1	PCE	Clear position counter error interrupt flag 0x0 = No effect 0x1 = Clears the interrupt flag	RW	0x0
0	INT	Global interrupt clear flag 0x0 = No effect 0x1 = Clears the interrupt flag and enables further interrupts to be generated if an event flags is set to 1.	RW	0x0

Table 21-183. Register Call Summary for Register EQEP_QCLR

Enhanced Quadrature Encoder Pulse (eQEP) Module

- eQEP Interrupt Structure: [0]
- Summary of PWMSS eQEP Functional Registers: [1]
- PWMSS_EQEP Registers: [2]

Table 21-184. EQEP_QFRC

Address offset	0x36		
Physical Address	0x4843 E1B6	**Instance**	PWMSS_EQEP
Description			
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				UTO	IEL	SEL	PCM	PCR	PCO	PCU	WTO	QDC	PHE	PCE	RESERVED

Bits	Field Name	Description	Type	Reset
15:12	RESERVED		R	0x0
11	UTO	Force unit time out interrupt 0x0 = No effect 0x1 = Force the interrupt	RW	0x0
10	IEL	Force index event latch interrupt 0x0 = No effect 0x1 = Force the interrupt	RW	0x0
9	SEL	Force strobe event latch interrupt 0x0 = No effect 0x1 = Force the interrupt	RW	0x0
8	PCM	Force position-compare match interrupt 0x0 = No effect 0x1 = Force the interrupt	RW	0x0
7	PCR	Force position-compare ready interrupt 0x0 = No effect 0x1 = Force the interrupt	RW	0x0
6	PCO	Force position counter overflow interrupt 0x0 = No effect 0x1 = Force the interrupt	RW	0x0
5	PCU	Force position counter underflow interrupt 0x0 = No effect 0x1 = Force the interrupt	RW	0x0
4	WTO	Force watchdog time out interrupt 0x0 = No effect 0x1 = Force the interrupt	RW	0x0
3	QDC	Force quadrature direction change interrupt 0x0 = No effect 0x1 = Force the interrupt	RW	0x0
2	PHE	Force quadrature phase error interrupt 0x0 = No effect 0x1 = Force the interrupt	RW	0x0
1	PCE	Force position counter error interrupt 0x0 = No effect 0x1 = Force the interrupt	RW	0x0
0	RESERVED		R	0x0

Table 21-185. Register Call Summary for Register EQEP_QFRC

Enhanced Quadrature Encoder Pulse (eQEP) Module

- eQEP Interrupt Structure: [0]
- Summary of PWMSS eQEP Functional Registers: [1]
- PWMSS_EQEP Registers: [2]

Table 21-186. EQEP_QEPSTS

Address offset	0x38		
Physical Address	0x4843 E1B8	**Instance**	PWMSS_EQEP
Description			
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								UPEVNT	FDF	QDF	QDLF	COEF	CDEF	FIMF	PCEF

Bits	Field Name	Description	Type	Reset
15:8	RESERVED		R	0x0
7	UPEVNT	Unit position event flag 0x0 = No unit position event detected 0x1 = Unit position event detected. Write 1 to clear.	R	0x0
6	FDF	Direction on the first index marker. Status of the direction is latched on the first index event marker. 0x0 = Counter-clockwise rotation (or reverse movement) on the first index event 0x1 = Clockwise rotation (or forward movement) on the first index event	R	0x0
5	QDF	Quadrature direction flag 0x0 = Counter-clockwise rotation (or reverse movement) 0x1 = Clockwise rotation (or forward movement)	R	0x0
4	QDLF	eQEP direction latch flag. Status of direction is latched on every index event marker. 0x0 = Counter-clockwise rotation (or reverse movement) on index event marker 0x1 = Clockwise rotation (or forward movement) on index event marker	R	0x0
3	COEF	Capture overflow error flag 0x0 = Sticky bit, cleared by writing 1 0x1 = Overflow occurred in eQEP Capture timer (QEPCTMR)	RW	0x0
2	CDEF	Capture direction error flag 0x0 = Sticky bit, cleared by writing 1 0x1 = Direction change occurred between the capture position event.	RW	0x0
1	FIMF	First index marker flag 0x0 = Sticky bit, cleared by writing 1 0x1 = Set by first occurrence of index pulse	RW	0x0
0	PCEF	Position counter error flag. This bit is not sticky and it is updated for every index event. 0x0 = No error occurred during the last index transition. 0x1 = Position counter error	R	0x0

Table 21-187. Register Call Summary for Register EQEP_QEPSTS

Enhanced Quadrature Encoder Pulse (eQEP) Module
- eQEP Quadrature Decoder Unit (QDU): [0]
- eQEP Position Counter and Control Unit (PCCU): [1][2][3][4][5][6][7][8][9][10][11][12][13][14][15][16]
- eQEP Edge Capture Unit: [17][18][19]
- Summary of PWMSS eQEP Functional Registers: [20]
- PWMSS_EQEP Registers: [21][22]

Table 21-188. EQEP_QCTMR

Address offset	0x3A		
Physical Address	0x4843 E1BA	**Instance**	PWMSS_EQEP
Description			
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
QCTMR															

Bits	Field Name	Description	Type	Reset
15:0	QCTMR	This register provides time base for edge capture unit.	RW	0x0

Table 21-189. Register Call Summary for Register EQEP_QCTMR

Enhanced Quadrature Encoder Pulse (eQEP) Module
- eQEP Edge Capture Unit: [0][1]
- Summary of PWMSS eQEP Functional Registers: [2]
- PWMSS_EQEP Registers: [3]

Table 21-190. EQEP_QCPRD

Address offset	0x3C		
Physical Address	0x4843 E1BC	**Instance**	PWMSS_EQEP
Description			
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
QCPRD															

Bits	Field Name	Description	Type	Reset
15:0	QCPRD	This register holds the period count value between the last successive eQEP position events	RW	0x0

Table 21-191. Register Call Summary for Register EQEP_QCPRD

Enhanced Quadrature Encoder Pulse (eQEP) Module
- eQEP Edge Capture Unit: [0][1][2]
- Summary of PWMSS eQEP Functional Registers: [3]
- PWMSS_EQEP Registers: [4]

Table 21-192. EQEP_QCTMRLAT

Address offset	0x3E		
Physical Address	0x4843 E1BE	**Instance**	PWMSS_EQEP
Description			
Type	R		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
QCTMRLAT															

Bits	Field Name	Description	Type	Reset
15:0	QCTMRLAT	The eQEP capture timer value can be latched into this register on two events, that is, unit timeout event, reading the eQEP position counter.	R	0x0

Table 21-193. Register Call Summary for Register EQEP_QCTMRLAT

Enhanced Quadrature Encoder Pulse (eQEP) Module

- eQEP Edge Capture Unit: [0][1]
- Summary of PWMSS eQEP Functional Registers: [2]
- PWMSS_EQEP Registers: [3][4][5]

Table 21-194. EQEP_QCPRDLAT

Address offset	0x40		
Physical Address	0x4843 E1C0	**Instance**	PWMSS_EQEP
Description			
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
QCPRDLAT															

Bits	Field Name	Description	Type	Reset
15:0	QCPRDLAT	eQEP capture period value can be latched into this register on two events, that is, unit timeout event, reading the eQEP position counter.	RW	0x0

Table 21-195. Register Call Summary for Register EQEP_QCPRDLAT

Enhanced Quadrature Encoder Pulse (eQEP) Module

- eQEP Edge Capture Unit: [0][1][2]
- Summary of PWMSS eQEP Functional Registers: [3]
- PWMSS_EQEP Registers: [4][5][6]

Table 21-196. EQEP_REVID

Address offset	0x5C		
Physical Address	0x4843 E1DC	**Instance**	PWMSS_EQEP
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REVISION																															

Bits	Field Name	Description	Type	Reset
31:0	REVISION	IP Revision	R	0x-[1]

(1) TI Internal data

Table 21-197. Register Call Summary for Register EQEP_REVID

Enhanced Quadrature Encoder Pulse (eQEP) Module

- Summary of PWMSS eQEP Functional Registers: [0]
- PWMSS_EQEP Registers: [1]

Audio Tracking Logic

This chapter describes the Audio Tracking Logic (ATL) module.

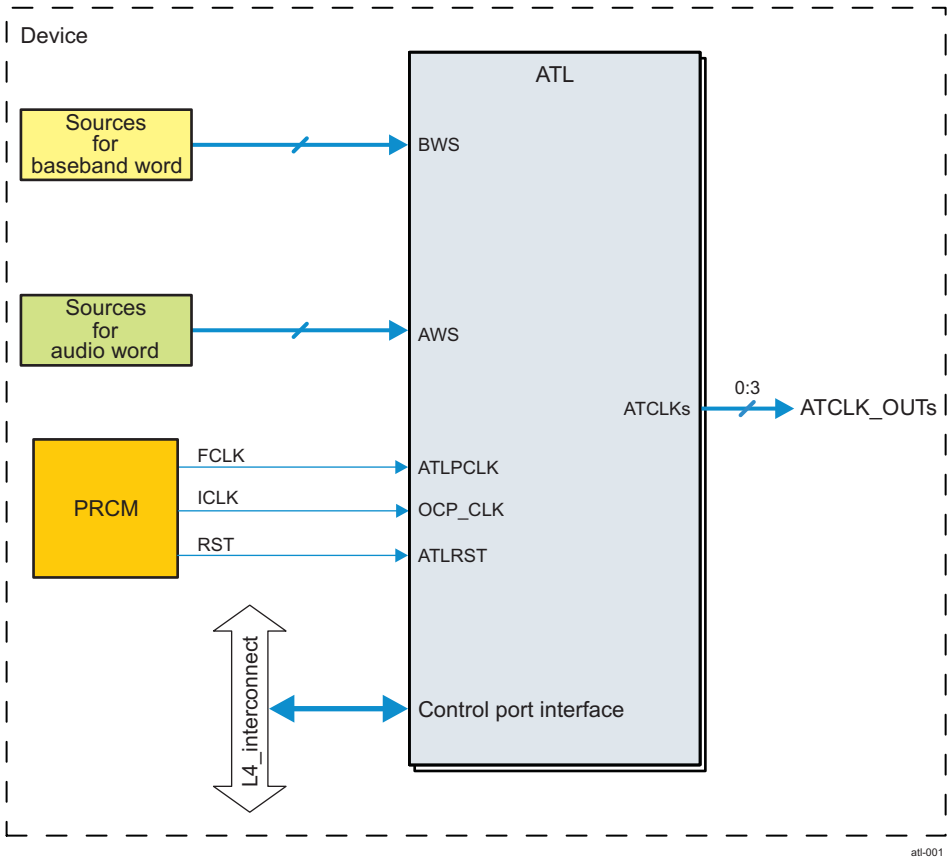
Topic **Page**

22.1 ATL Overview

The audio tracking logic (ATL) is used by HD Radio™ applications to synchronize the digital audio output to the baseband clock. This same IP can also be used generically to track errors between two reference signals (such as frame syncs) and generate a modulated clock output (using software-controlled cycle stealing) which averages to some desired frequency. This process can be used as a hardware assist for asynchronous sample rate conversion algorithms. The tracking range is limited, so direct conversion between the two standard sample rate groups frequencies from 44.1 to 48 kHz is not possible.

Figure 22-1 shows an overview of the ATL module.

Figure 22-1. ATL Module Overview



The ATL includes the following main features:

- One ATL module, containing four ATL instances, for HD Radio support and asynchronous sample rate conversion assistance
- Each instance tracks the time error between two syncs (local audio word select [AWS] and baseband word select [BWS])
- Each instance selects between 16 mux choices for each of AWS and BWS.
- Each instance generates modulated ATCLK_OUT clock signal with software-initiated pulse stealing.
- Selection between INTERCONNECT clock or functional ATLPCLK to run error couting timers and to derive modulated ATCLK_OUT clock outputs
- Clock and reset management: Receives clock and reset signals from the device PRCM module. The ATL module receives hardware reset from the CORE_RST reset domain.
- Power management: The ATL belongs to the PD_COREAON power domain.

ADC

This chapter describes the analog-to-digital converter in the device.

Topic **Page**

23.1 ADC Overview

The analog-to-digital converter (ADC) module is a successive-approximation-register (SAR) general-purpose analog-to-digital converter.

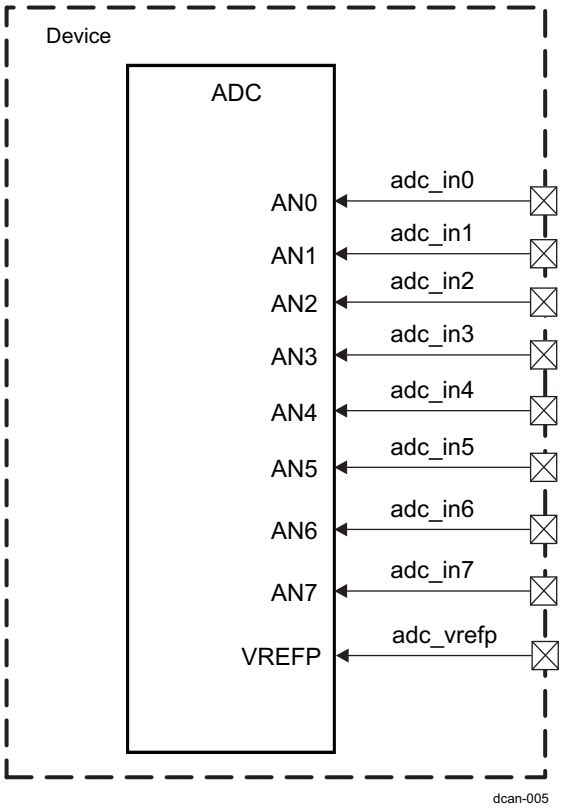
The main features of the ADC include:

- 10-bit data
- 750 KSPS at 13.5-MHz ADC_CLK
- 8 channels
- Programmable FSM sequencer that supports 16 steps:
  - Software register bit for start of conversion
  - Single conversion (one-shot)
  - Continuous conversions
  - Sequence through all input channels based on a mask
  - Programmable OpenDelay before sampling each channel
  - Programmable sampling delay for each channel
  - Programmable averaging of input samples - 16/8/4/2/1
  - Store data in either of two FIFO groups – 64-deep each
  - Option to encode channel number with data
  - Support for servicing FIFOs via DMA or CPU
  - Programmable DMA Request event (for each FIFO)
  - Dynamically enable or disable channel inputs during operation
  - Stop bit to end conversion
  - Support for error offset (internal calibration or external calibration via eFuse) inside the AFE
- Support for the following interrupts and status, with masking:
  - Interrupt after a sequence of conversions (all non-masked channels)
  - Interrupt for FIFO threshold levels
  - Interrupt if sampled data is out of a programmable range
  - Interrupt for FIFO overflow and underflow conditions
  - Status bit to indicate if ADC is busy converting

23.2 ADC Environment

Figure 23-1 shows the ADC IO pins.

Figure 23-1. ADC Environment



23.2.1 ADC Signals

Table 23-1 describes the module I/O signals.

Table 23-1. I/O Description

Device Signal	Signal	I/O[1]	Description	Reset Value[2]
adc_in0	AN0	A/I	Analog channel input 0	Hi-Z
adc_in1	AN1	A/I	Analog channel input 1	Hi-Z
adc_in2	AN2	A/I	Analog channel input 2	Hi-Z
adc_in3	AN3	A/I	Analog channel input 3	Hi-Z
adc_in4	AN4	A/I	Analog channel input 4	Hi-Z
adc_in5	AN5	A/I	Analog channel input 5	Hi-Z
adc_in6	AN6	A/I	Analog channel input 6	Hi-Z
adc_in7	AN7	A/I	Analog channel input 7	Hi-Z
adc_vrefp	VREFP	A/I	Analog reference positive voltage	Hi-Z

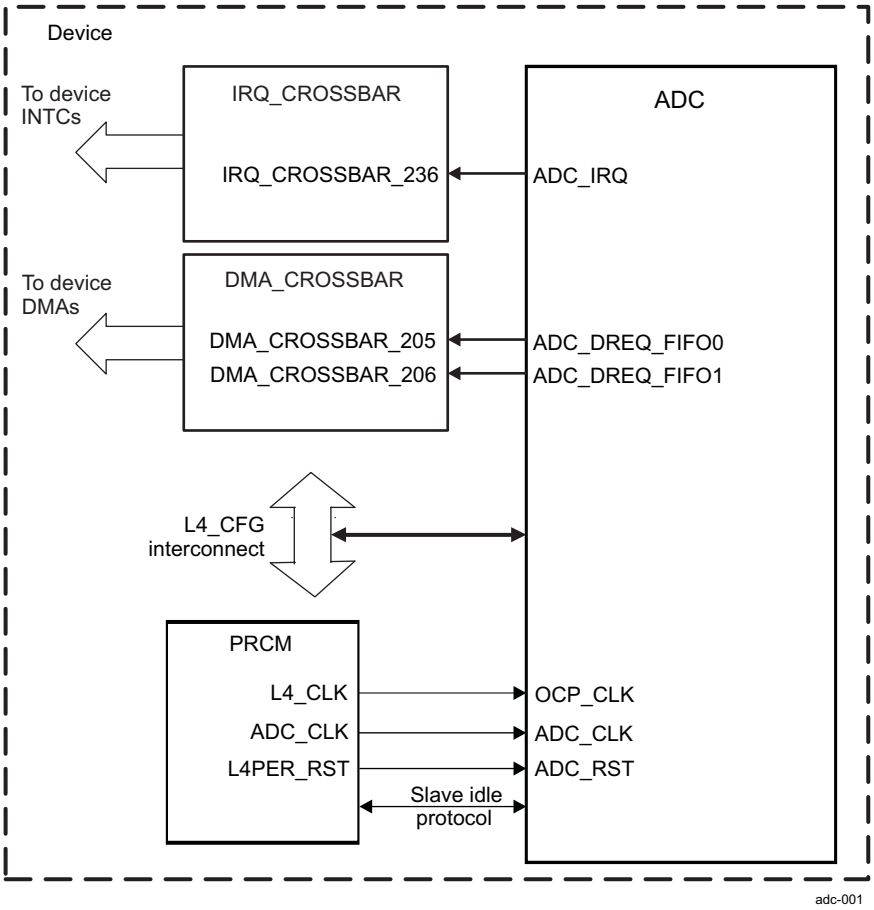
(1) I = Input; O = Output; I/O = Bidirectional; A = Analog
(2) Hi-Z = High Impedance

23.3 ADC Integration

This section describes module integration in the device, including information about clocks, resets, and hardware requests.

Figure 23-2 shows the module integration.

Figure 23-2. ADC Integration



NOTE: For more information about the slave idle protocol, see Section 3.1.1.1, *Clock Management*, in Chapter 3, *Power, Reset, and Clock Management*.

Table 23-2 through Table 23-4 summarize the integration of the module in the device.

Table 23-2. Integration Attributes

Module Instance	Attributes		
	Power Domain	Wake-Up Capability	Interconnect
ADC	PD_COREAON	No	L4_CFG

Table 23-3. Clocks and Resets

Clocks				
Module Instance	**Destination Signal Name**	**Source Signal Name**	**Source**	**Description**
ADC	OCP_CLK	L4_CLK	PRCM	ADC interface clock
	ADC_CLK	ADC_CLK	Control Module	ADC functional clock. 20-MHz max. Clock selection is done within Control Module. Clock division by 2 is selected via sysboot[5] pin.
Resets				
ADC	ADC_RST	L4PER_RST	PRCM	ADC reset signal

Table 23-4. GPIO Hardware Requests

Interrupt Requests				
Module Instance	**Source Signal Name**	**IRQ_CROSSBAR Input**	**Default Mapping**	**Description**
ADC	ADC_IRQ	IRQ_CROSSBAR_236	N/A	ADC interrupt request (level)
DMA Requests				
Module Instance	**Source Signal Name**	**DMA_CROSSBAR Input**	**Default Mapping**	**Description**
ADC	ADC_DREQ_FIFO0	DMA_CROSSBAR_205	N/A	FIFO0 reached desired data level
	ADC_DREQ_FIFO1	DMA_CROSSBAR_206	N/A	FIFO1 reached desired data level

NOTE: For more information about the IRQ_CROSSBAR module, see Section 13.4.6.4, *IRQ_CROSSBAR Module Functional Description*, in Chapter 13, *Control Module*. For more information about the DMA_CROSSBAR module, see Section 13.4.6.5, *DMA_CROSSBAR Module Functional Description*, in Chapter 13, *Control Module*.

23.4 ADC Functional Description

Before enabling the ADC module, the user must first program the ADC_STEPCONFIGi registers in order to configure a channel input to be sampled. There are 16 programmable ADC_STEPCONFIGi registers which are used by the sequencer to control which channel to sample and which mode to use (one-shot or continuous, averaging, where to save the FIFO data, and more).

23.4.1 Open Delay

The user can program the delay between driving the inputs to the AFE and the time to send the start of conversion signal. This delay is called “open delay” and can also be programmed to zero. The open delay for each step can be independently configured using the ADC_STEPDELAYi register.

23.4.2 Averaging of Samples (1, 2, 4, 8, and 16)

Each step has the capability to average the sampled data. The valid averaging options are 1 (no average), 2, 4, 8, and 16. If averaging is turned on, then the channel is immediately sampled again (up to 16 times) and final averaged sample data is stored in the FIFO. Each step can be independently configured using the ADC_STEPCONFIGi registers.

23.4.3 One-Shot (Single) or Continuous Mode

When the sequencer finishes cycling through all the enabled steps, the user can decide if the sequencer should stop (one-shot), or loop back and schedule the step again (continuous).

If one-shot mode is selected, the sequencer will take care of disabling the step enable bit after the conversion. If continuous mode is selected, it is the software’s responsibility to turn off the step enable bit.

23.4.4 Interrupts

The following interrupts are supported through enable bits and are maskable.

- Each FIFO has support for generating interrupts when the FIFO word count has reached a programmable threshold level. The user can program the desired word count at which the CPU should be interrupted. Whenever the threshold counter value is reached, it sets the FIFOx_THRESHOLD (x = 0 or 1) interrupt flag, and the CPU is interrupted if the FIFOx_THRESHOLD interrupt enable bit is set. The user can clear the interrupt flag, after emptying the FIFO, by writing a 1 to the FIFOxTHRESHOLD interrupt status bit. To determine how many samples are currently in the FIFO at a given moment, the ADC_FIFO0COUNT/ADC_FIFO1COUNT register can be read by the CPU.
- The FIFO can also generate FIFOx_OVERRUN and FIFOx_UNDERFLOW interrupts. The user can mask these events by programming the ADC_IRQENABLE_CLR register. To clear a FIFO underflow

or FIFO overrun interrupt, the user should write a 1 to the status bit in the ADC_IRQSTATUS register. The ADC does not recover from these conditions automatically. Therefore, the software will need to disable and then again enable the ADC. Before the user can turn the module back on, the user must read the ADC_ADCSTAT register to check if the status of the ADC FSM is idle and the current step is the idle step

- An END_OF_SEQUENCE interrupt is generated after the sequencer finishes servicing the last enabled step.
- An OUT_OF_RANGE interrupt is generated if sampled data is greater than programmable value, or less than a programmable value. High and low range thresholds are programmed in the ADC_ADCRANGE register.

23.4.5 DMA Requests

Each FIFO group can be serviced by either a DMA or by the CPU. To generate DMA requests, the user must set the enable bit in the ADC_DMAENABLE_SET Register. Also, the user can program the desired number of words to generate a DMA request using the ADC_DMA0REQ/ADC_DMA1REQ register. When the FIFO level reaches or exceeds that value, a DMA request is generated.

Subsequently, if the FIFO contains enough data in the FIFO, then a new DMA request is generated after the current DMA access (if this read does not cause the FIFO to be empty). The DMA request occurs on the very next cycle after the previous DMA FIFO read. The CPU can also read from the FIFO by reading from the FIFO DATA register. The internal logic will pop the next data from the FIFO and increment the FIFO read pointers internally.

The DMA slave port allows for burst reads in order to effectively move the FIFO data. Internally, the OCP DMA address (MSB) is decoded for either FIFO 0 or FIFO 1. The lower bits of the DMA addresses are ignored since the FIFO pointers are incremented internally.

23.4.6 Power Management

The ADC supports slave idle/request handshaking, and also supports a smart-idle mode. If the PRCM requests to power down when the ADC_MODULE_ENABLE bit is disabled, then acknowledge is returned. The ADC will also return acknowledge if sequencer is in the Idle state, and if all the ADC_STEPENABLE bits are off.

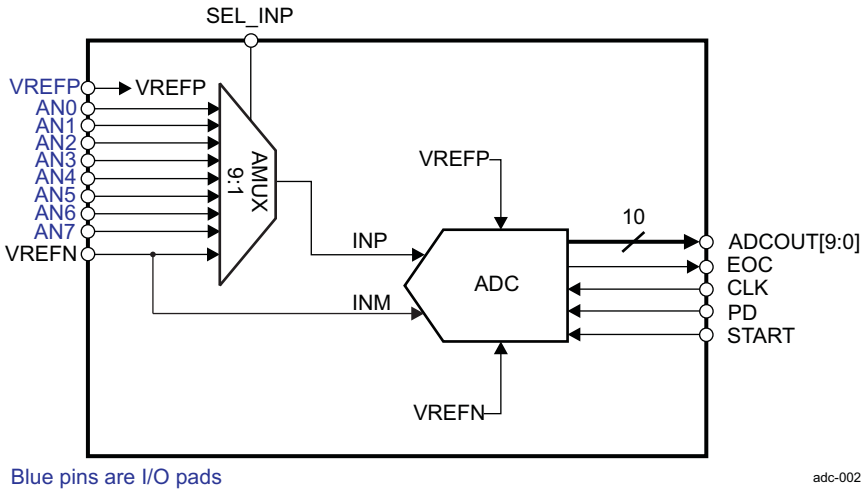
The software also has the option to turn off the AFE power by writing to the AFE ADC_CTRL[4] POWER_DOWN register bit. During initial power on, the AFE is powered down by default and software is responsible for turning the AFE on. software must wait minimum 4 µs after a AFE power up and before starting a conversion (must be followed every time the AFE is powered up). It is the software's responsibility to empty the FIFO before requesting the ADC to enter sleep state.

23.4.7 Analog Front End (AFE) Functional Block Diagram

The AFE features are listed below, and some are controlled by the ADC:

- ADC sampled data is 10-bit wide
- Sampling rate can be as fast as every 17 ADC clock cycles
- Support for internal ADC clock divider logic
- Support for configuring the delay between samples also the sampling time

Figure 23-3. Functional Block Diagram



23.4.8 Operational Modes

The sequencer is completely controlled by software and behaves accordingly to how the **Step Registers** are programmed. A **step** is the general term for sampling a channel input. It is defined by the programmer who decides which input values to send to the AFE as well as how and when to sample a channel input.

The choices for each step can all be programmed using the ADC_STEPCONFIGi registers.

A step requires using these registers:

- ADC_STEPENABLE: Enables or disables the step
- ADC_STEPCONFIGi: Controls the input values to the ADC (which input channel to sample, averaging, and which FIFO group to save the data, range check)
- ADC_STEPDELAYi: Controls the OpenDelay (the time between driving the AFE inputs until sending the start-of-conversion (SOC) signal to the AFE).

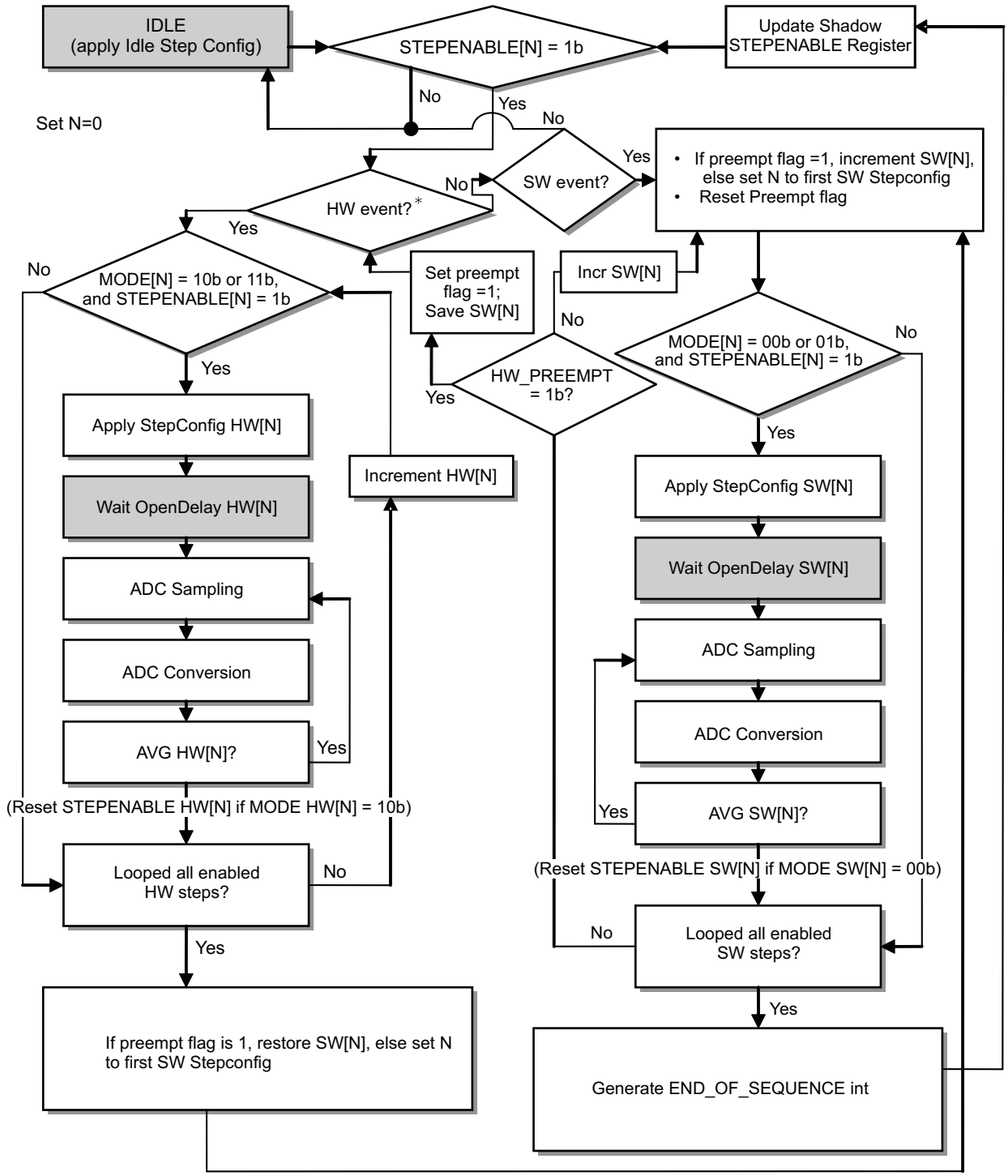
The sequencer supports a total of 16 programmable steps. Each step requires using the registers listed above.

When the ADC is first enabled, the sequencer will a then wait for a ADC_STEPENABLE bit to turn on. After a step is enabled, the sequencer will start with the lowest step (1) and continue until step (16). If a step is not enabled, then sequencer will skip to the next step. If all steps are disabled, then the sequencer will remain in the IDLE state.

An END_OF_SEQUENCE interrupt is generated after the last active step is completed before going back to the IDLE state. The END_OF_SEQUENCE interrupt does not mean data is in the FIFO (should use the FIFO interrupts and ADC_FIFO0COUNT/ADC_FIFO1COUNT register).

The flowchart for the sequencer is shown in Figure 23-4 and an example timing diagram in Figure 23-5.

Figure 23-4. Sequencer FSM

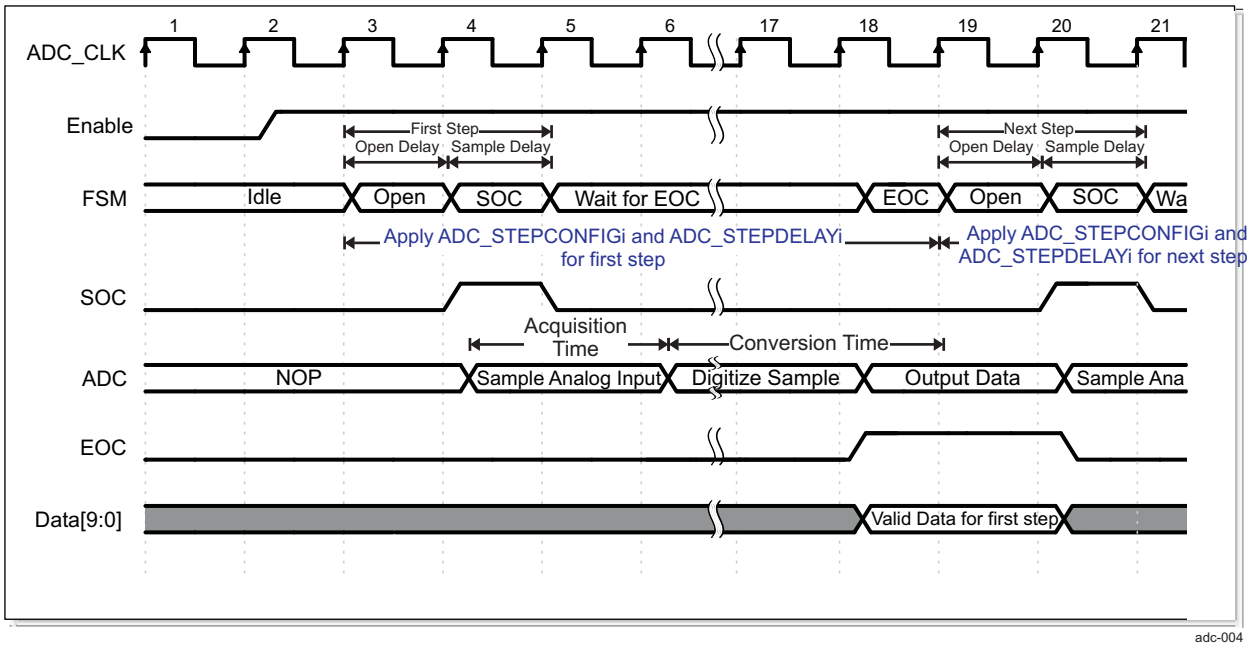


* HW event is not pinned out on this device

adc-003

Figure 23-4 does not actually represent clock cycles but instead illustrates how the scheduler will work. However, each shaded box above does represent a FSM state and will use a clock cycle. Using the minimum delay values, the ADC can sample at 17 ADC clocks per sample. Below is an example timing diagram illustrating the states of the sequencer and also the showing when the ADC_STEPCONFIGi and the ADC_STEPDELAYi registers values are applied. The below example assumes the steps are software controlled, and averaging is turned off.

Figure 23-5. Example Timing Diagram for Sequencer



Once the ADC is enabled and assuming at least one step is active, the FSM will transition from the idle state and apply the first active ADC_STEPCONFIGi and ADC_STEPDELAYi register settings. It is possible for the Open Delay value to be 0, and the FSM will immediately skip to the SOC state. The AFE will begin sampling of the analog voltage on high level of the SOC signal. Voltage sampling duration is 4 clock cycles long. After the AFE is finished converting the channel data (13 more cycles later), the EOC signal is sent and the FSM will then apply the next active step.

This process is repeated and continued (from step 1 to step 16) until the last active step is completed.

23.5 ADC Programming Guide

23.5.1 ADC Low-Level Programming Models

23.5.1.1 Global Initialization

23.5.1.1.1 Surrounding Modules Global Initialization

This section identifies the requirements for initializing the surrounding modules when the module is to be used for the first time after a device reset.

Table 23-5. Global Initialization of Surrounding Modules

Surrounding Modules	Comments
PRCM	Module interface and functional clocks must be enabled. For more information about the module configuration, see Section 3.1.1.1, *Clock Management*, in Chapter 3 *Power, Reset, and Clock Management*.
Control module	Module specific pad muxing must be set in the control module. For more information about the module configuration, see Section 13.4.6.1, *Pad Configuration Registers*, in Chapter 13 *Control Module*. For more information about the IRQ_CROSSBAR module, see Section 13.4.6.4, *IRQ_CROSSBAR Module Functional Description*. For more information about the DMA_CROSSBAR module, see Section 13.4.6.5, *DMA_CROSSBAR Module Functional Description*.

Table 23-5. Global Initialization of Surrounding Modules (continued)

Surrounding Modules	Comments
DSP INTC	DSP INTC configuration must be done to enable the interrupts from the general-purpose interface module. See Section 12.3.1, *Interrupt Requests to DSP1_INTC* , in Chapter 12, *Interrupt Controllers*.
IPU INTC	IPU INTC configuration must be done to enable the interrupts from the general-purpose interface module. See Section 12.3.3, *Interrupt Requests to IPU1_Cx_INTC*, in Chapter 12, *Interrupt Controllers*.

23.5.1.1.2 General Programming Model

Table 23-6. General Programming Model

Step	Register/Bit Field/Programming Model	Value
Power up the AFE	ADC_CTRL[4] POWER_DOWN	0
Setup the input analog ADC_CLK frequency input	ADC_ADC_CLKDIV[11:0] ADC_CLKDIV	0x-
If using DMA, then enable the DMA requests lines and program DMA threshold levels; or set the CPU threshold levels to read FIFO with CPU	ADC_DMAENABLE_SET/ADC_DMA0REQ/ADC_DMA1REQ ADC_FIFO0THRESHOLD/ADC_FIFO1THRESHOLD	0x-
Setup or mask the appropriate interrupts	ADC_IRQENABLE_SET	0x-
Program the StepConfig registers for averaging, mode, etc.	ADC_STEPCONFIGi	0x-
Program the StepDelay registers for OpenDelay	ADC_STEPDELAYi	0x-
Turn on the correct StepEnable bits	ADC_STEPENABLE	0x-
Wait minimum 4 µs before starting a conversion (reference from POWER_DOWN write to 0)		
Enable the ADC module	ADC_CTRL[0] ADC_MODULE_ENABLE	1

23.5.1.2 During Operation

Table 23-7. Turn-Off

Step	Register/Bit Field/Programming Model	Value
Turn off ADC module bit when finished. This will stop the ADC after the current step is complete.	ADC_CTRL[0] ADC_MODULE_ENABLE	0
Or, if desired to finish the complete sequence of steps, then disable all of the StepEnable bits and the ADC will stop after the last step and wait (end of sequence interrupt is generated)	ADC_STEPENABLE	0x0

Table 23-8. Turn-On After Turn-Off

Step	Register/Bit Field/Programming Model	Value
If desired to turn on the ADC again, then software must read the ADC status register to make sure the current step is idle.	ADC_ADCSTAT[4:0] STEP_ID	=0x10
Enable the ADC module enable bit	ADC_CTRL[0] ADC_MODULE_ENABLE	1

23.6 ADC Register Manual

23.6.1 ADC Instance Summary

Table 23-9 summarizes the ADC instance.

Table 23-9. Instance Summary

Module Name	Module Base Address	Size
ADC	0x4A26 4000	4KiB

23.6.2 ADC Registers

23.6.2.1 ADC Register Summary

Table 23-10 summarizes the ADC registers.

Table 23-10. ADC Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	Physical Address
ADC_REVISION	R	32	0x0000 0000	0x4A26 4000
ADC_SYSCONFIG	RW	32	0x0000 0010	0x4A26 4010
ADC_IRQ_EOI	RW	32	0x0000 0020	0x4A26 4020
ADC_IRQSTATUS_RAW	RW	32	0x0000 0024	0x4A26 4024
ADC_IRQSTATUS	RW	32	0x0000 0028	0x4A26 4028
ADC_IRQENABLE_SET	RW	32	0x0000 002C	0x4A26 402C
ADC_IRQENABLE_CLR	RW	32	0x0000 0030	0x4A26 4030
RESERVED	RW	32	0x0000 0034	0x4A26 4034
ADC_DMAENABLE_SET	RW	32	0x0000 0038	0x4A26 4038
ADC_DMAENABLE_CLR	RW	32	0x0000 003C	0x4A26 403C
ADC_CTRL	RW	32	0x0000 0040	0x4A26 4040
ADC_ADCSTAT	R	32	0x0000 0044	0x4A26 4044
ADC_ADCRANGE	RW	32	0x0000 0048	0x4A26 4048
ADC_ADC_CLKDIV	RW	32	0x0000 004C	0x4A26 404C
ADC_ADC_MISC	RW	32	0x0000 0050	0x4A26 4050
ADC_STEPENABLE	RW	32	0x0000 0054	0x4A26 4054
RESERVED	RW	32	0x0000 0058	0x4A26 4058
RESERVED	RW	32	0x0000 005C	0x4A26 405C
RESERVED	RW	32	0x0000 0060	0x4A26 4060
ADC_STEPCONFIGi[(1)]	RW	32	0x0000 0064 + (i * 8)	0x4A26 4064 + (I * 8)
ADC_STEPDELAYi[(1)]	RW	32	0x0000 0068 + (i * 8)	0x4A26 4068 + (I * 8)
ADC_FIFO0COUNT	R	32	0x0000 00E4	0x4A26 40E4
ADC_FIFO0THRESHOLD	RW	32	0x0000 00E8	0x4A26 40E8
ADC_DMA0REQ	RW	32	0x0000 00EC	0x4A26 40EC
ADC_FIFO1COUNT	R	32	0x0000 00F0	0x4A26 40F0
ADC_FIFO1THRESHOLD	RW	32	0x0000 00F4	0x4A26 40F4
ADC_DMA1REQ	RW	32	0x0000 00F8	0x4A26 40F8
ADC_FIFO0DATA	R	32	0x0000 0100	0x4A26 4100
ADC_FIFO1DATA	R	32	0x0000 0200	0x4A26 4200

(1) i = 0 to 15

23.6.2.2 ADC Register Description

Table 23-11 through Table 23-59 describe the individual ADC registers.

Table 23-11. ADC_REVISION

Address Offset	0x0000 0000		
Physical Address	0x4A26 4000	**Instance**	ADC
Description	Revision identifier		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
REVISION																															

Bits	Field Name	Description	Type	Reset
31:0	REVISION	IP revision	R	See[(1)]

(1) TI Internal Data

Table 23-12. Register Call Summary for Register ADC_REVISION

ADC Register Manual
- ADC Register Summary: [0]

Table 23-13. ADC_SYSCONFIG

Address Offset	0x0000 0010		
Physical Address	0x4A26 4010	**Instance**	ADC
Description	System configuration register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												IDLEMODE		RESERVED	

Bits	Field Name	Description	Type	Reset
31:4	RESERVED	Reserved	R	0x0000000
3:2	IDLEMODE	0x0: Force-idle: An IDLE request is acknowledged unconditionally. 0x1: No-idle: An IDLE request is never acknowledged. 0x2: Smart-idle: The acknowledgment to an IDLE request is given based on the internal activity 0x3: Reserved	RW	0x2
1:0	RESERVED	Reserved	RW	0

Table 23-14. Register Call Summary for Register ADC_SYSCONFIG

ADC Register Manual
- ADC Register Summary: [0]

Table 23-15. ADC_IRQ_EOI

Address Offset	0x0000 0020		
Physical Address	0x4A26 4020	**Instance**	ADC
Description	Software end of interrupt.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																															LINE_NUMBER

Bits	Field Name	Description	Type	Reset
31:1	RESERVED	Reserved	R	0x0
0	LINE_NUMBER	Software End Of Interrupt (EOI) control. Write 0x0: EOI for interrupt line number 0. Read returns 0.	W	0x0

Table 23-16. Register Call Summary for Register ADC_IRQ_EOI

ADC Register Manual
- ADC Register Summary: [0]

Table 23-17. ADC_IRQSTATUS_RAW

Address Offset	0x0000 0024		
Physical Address	0x4A26 4024	**Instance**	ADC
Description	Per-event raw interrupt status vector, showing all active events (enabled and not enabled)		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							OUT_OF_RANGE	FIFO1_UNDERFLOW	FIFO1_OVERRUN	FIFO1_THRESHOLD	FIFO0_UNDERFLOW	FIFO0_OVERRUN	FIFO0_THRESHOLD	END_OF_SEQUENCE	RESERVED

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reserved	RW	0
8	OUT_OF_RANGE	Status raw for interrupt. Write 0: No action Write 1: Set event (debug) Read 0: No event pending Read 1: Event pending	RW	0
7	FIFO1_UNDERFLOW	Status raw for interrupt. Write 0: No action Write 1: Set event (debug) Read 0: No event pending Read 1: Event pending	RW	0
6	FIFO1_OVERRUN	Status raw for interrupt. Write 0: No action Write 1: Set event (debug) Read 0: No event pending Read 1: Event pending	RW	0
5	FIFO1_THRESHOLD	Status raw for interrupt. Write 0: No action Write 1: Set event (debug) Read 0: No event pending Read 1: Event pending	RW	0

Bits	Field Name	Description	Type	Reset
4	FIFO0_UNDERFLOW	Status raw for interrupt. Write 0: No action Write 1: Set event (debug) Read 0: No event pending Read 1: Event pending	RW	0
3	FIFO0_OVERRUN	Status raw for interrupt. Write 0: No action Write 1: Set event (debug) Read 0: No event pending Read 1: Event pending	RW	0
2	FIFO0_THRESHOLD	Status raw for interrupt. Write 0: No action Write 1: Set event (debug) Read 0: No event pending Read 1: Event pending	RW	0
1	END_OF_SEQUENCE	Status raw for interrupt. Write 0: No action Write 1: Set event (debug) Read 0: No event pending Read 1: Event pending	RW	0
0	RESERVED	Reserved	RW	0

Table 23-18. Register Call Summary for Register ADC_IRQSTATUS_RAW

ADC Register Manual

- ADC Register Summary: [0]

Table 23-19. ADC_IRQSTATUS

Address Offset	0x0000 0028		
Physical Address	0x4A26 4028	**Instance**	ADC
Description	Per-event “enabled” status register vector. The enabled status isn’t set unless the event is enabled.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							OUT_OF_RANGE	FIFO1_UNDERFLOW	FIFO1_OVERRUN	FIFO1_THRESHOLD	FIFO0_UNDERFLOW	FIFO0_OVERRUN	FIFO0_THRESHOLD	END_OF_SEQUENCE	RESERVED

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reserved	RW	0
8	OUT_OF_RANGE	Enabled status for interrupt. Write 0: No action Write 1: Clear (raw) event Read 0: No (enabled) event pending Read 1: Event pending	RW	0
7	FIFO1_UNDERFLOW	Enabled status for interrupt. Write 0: No action Write 1: Clear (raw) event Read 0: No (enabled) event pending Read 1: Event pending	RW	0

Bits	Field Name	Description	Type	Reset
6	FIFO1_OVERRUN	Enabled status for interrupt. Write 0: No action Write 1: Clear (raw) event Read 0: No (enabled) event pending Read 1: Event pending	RW	0
5	FIFO1_THRESHOLD	Enabled status for interrupt. Write 0: No action Write 1: Clear (raw) event Read 0: No (enabled) event pending Read 1: Event pending	RW	0
4	FIFO0_UNDERFLOW	Enabled status for interrupt. Write 0: No action Write 1: Clear (raw) event Read 0: No (enabled) event pending Read 1: Event pending	RW	0
3	FIFO0_OVERRUN	Enabled status for interrupt. Write 0: No action Write 1: Clear (raw) event Read 0: No (enabled) event pending Read 1: Event pending	RW	0
2	FIFO0_THRESHOLD	Enabled status for interrupt. Write 0: No action Write 1: Clear (raw) event Read 0: No (enabled) event pending Read 1: Event pending	RW	0
1	END_OF_SEQUENCE	Enabled status for interrupt. Write 0: No action Write 1: Clear (raw) event Read 0: No (enabled) event pending Read 1: Event pending	RW	0
0	RESERVED	Reserved	RW	0

Table 23-20. Register Call Summary for Register ADC_IRQSTATUS

ADC Functional Description • Interrupts: [0]
ADC Register Manual • ADC Register Summary: [1]

Table 23-21. ADC_IRQENABLE_SET

Address Offset	0x0000 002C		
Physical Address	0x4A26 402C	**Instance**	ADC
Description	Per-event interrupt enable bit vector		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							OUT_OF_RANGE	FIFO1_UNDERFLOW	FIFO1_OVERRUN	FIFO1_THRESHOLD	FIFO0_UNDERFLOW	FIFO0_OVERRUN	FIFO0_THRESHOLD	END_OF_SEQUENCE	RESERVED

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reserved	RW	0
8	OUT_OF_RANGE	Interrupt enable. Write 0: No action Write 1: Enable interrupt Read 0: Interrupt disabled (masked) Read 1: Interrupt enabled	RW	0
7	FIFO1_UNDERFLOW	Interrupt enable. Write 0: No action Write 1: Enable interrupt Read 0: Interrupt disabled (masked) Read 1: Interrupt enabled	RW	0
6	FIFO1_OVERRUN	Interrupt enable. Write 0: No action Write 1: Enable interrupt Read 0: Interrupt disabled (masked) Read 1: Interrupt enabled	RW	0
5	FIFO1_THRESHOLD	Interrupt enable. Write 0: No action Write 1: Enable interrupt Read 0: Interrupt disabled (masked) Read 1: Interrupt enabled	RW	0
4	FIFO0_UNDERFLOW	Interrupt enable. Write 0: No action Write 1: Enable interrupt Read 0: Interrupt disabled (masked) Read 1: Interrupt enabled	RW	0
3	FIFO0_OVERRUN	Interrupt enable. Write 0: No action Write 1: Enable interrupt Read 0: Interrupt disabled (masked) Read 1: Interrupt enabled	RW	0
2	FIFO0_THRESHOLD	Interrupt enable. Write 0: No action Write 1: Enable interrupt Read 0: Interrupt disabled (masked) Read 1: Interrupt enabled	RW	0
1	END_OF_SEQUENCE	Interrupt enable. Write 0: No action Write 1: Enable interrupt Read 0: Interrupt disabled (masked) Read 1: Interrupt enabled	RW	0
0	RESERVED	Reserved	RW	0

Table 23-22. Register Call Summary for Register ADC_IRQENABLE_SET

ADC Programming Guide • Global Initialization: [0]
ADC Register Manual • ADC Register Summary: [1]

Table 23-23. ADC_IRQENABLE_CLR

Address Offset	0x0000 0030		
Physical Address	0x4A26 4030	**Instance**	ADC
Description	Per-event interrupt disable bit vector		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							OUT_OF_RANGE	FIFO1_UNDERFLOW	FIFO1_OVERRUN	FIFO1_THRESHOLD	FIFO0_UNDERFLOW	FIFO0_OVERRUN	FIFO0_THRESHOLD	END_OF_SEQUENCE	RESERVED

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reserved	RW	0
8	OUT_OF_RANGE	Interrupt disable. Write 0: No action Write 1: Disable interrupt Read 0: Interrupt disabled (masked) Read 1: Interrupt enabled	RW	0
7	FIFO1_UNDERFLOW	Interrupt disable. Write 0: No action Write 1: Disable interrupt Read 0: Interrupt disabled (masked) Read 1: Interrupt enabled	RW	0
6	FIFO1_OVERRUN	Interrupt disable. Write 0: No action Write 1: Disable interrupt Read 0: Interrupt disabled (masked) Read 1: Interrupt enabled	RW	0
5	FIFO1_THRESHOLD	Interrupt disable. Write 0: No action Write 1: Disable interrupt Read 0: Interrupt disabled (masked) Read 1: Interrupt enabled	RW	0
4	FIFO0_UNDERFLOW	Interrupt disable. Write 0: No action Write 1: Disable interrupt Read 0: Interrupt disabled (masked) Read 1: Interrupt enabled	RW	0
3	FIFO0_OVERRUN	Interrupt disable. Write 0: No action Write 1: Disable interrupt Read 0: Interrupt disabled (masked) Read 1: Interrupt enabled	RW	0
2	FIFO0_THRESHOLD	Interrupt disable. Write 0: No action Write 1: Disable interrupt Read 0: Interrupt disabled (masked) Read 1: Interrupt enabled	RW	0
1	END_OF_SEQUENCE	Interrupt disable. Write 0: No action Write 1: Disable interrupt Read 0: Interrupt disabled (masked) Read 1: Interrupt enabled	RW	0
0	RESERVED	Reserved	RW	0

Table 23-24. Register Call Summary for Register ADC_IRQENABLE_CLR

ADC Functional Description • Interrupts: [0]
ADC Register Manual • ADC Register Summary: [1]

Table 23-25. ADC_DMAENABLE_SET

Address Offset	0x0000 0038		
Physical Address	0x4A26 4038	**Instance**	ADC
Description	Per-line DMA enable bit vector.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														ENABLE_1	ENABLE_0

Bits	Field Name	Description	Type	Reset
31:2	RESERVED	Reserved	RW	0
1	ENABLE_1	Enable DMA request FIFO 1 Write 0: No action Write 1: Enable DMA line Read 0: DMA line disabled Read 1: DMA line enabled	RW	0
0	ENABLE_0	Enable DMA request FIFO 0 Write 0: No action Write 1: Enable DMA line Read 0: DMA line disabled Read 1: DMA line enabled	RW	0

Table 23-26. Register Call Summary for Register ADC_DMAENABLE_SET

ADC Functional Description • DMA Requests: [0]
ADC Programming Guide • Global Initialization: [1]
ADC Register Manual • ADC Register Summary: [2]

Table 23-27. ADC_DMAENABLE_CLR

Address Offset	0x0000 003C		
Physical Address	0x4A26 403C	**Instance**	ADC
Description	Per-line DMA disable bit vector.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														ENABLE_1	ENABLE_0

Bits	Field Name	Description	Type	Reset
31:2	RESERVED	Reserved	RW	0
1	ENABLE_1	Disable DMA request FIFO 1 Write 0: No action Write 1: Disable DMA line Read 0: DMA line disabled Read 1: DMA line enabled	RW	0
0	ENABLE_0	Disable DMA request FIFO 0 Write 0: No action Write 1: Disable DMA line Read 0: DMA line disabled Read 1: DMA line enabled	RW	0

Table 23-28. Register Call Summary for Register ADC_DMAENABLE_CLR

ADC Register Manual
- ADC Register Summary: [0]

Table 23-29. ADC_CTRL

Address Offset	0x0000 0040		
Physical Address	0x4A26 4040	**Instance**	ADC
Description	ADC control register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																						HW_PREEMPT	RESERVED				POWER_DOWN	RESERVED		STEP_ID_TAG	ADC_MODULE_ENABLE

Bits	Field Name	Description	Type	Reset
31:10	RESERVED	Reserved	R	0
9	HW_PREEMPT	0: SW steps are not preempted by HW events 1: SW steps are preempted by HW events HW EVENTS ARE NOT SUPPORTED IN THIS DEVICE	RW	0
8:5	RESERVED	Reserved	RW	0
4	POWER_DOWN	ADC Power Down control 0: AFE is powered up 1 : AFE is powered down (default) At default, AFE is powered down; Software must write 0 to turn on the power and wait 4 us before starting a conversion	RW	1
3:2	RESERVED		RW	0
1	STEP_ID_TAG	Writing 1 to this bit will store the Step ID number with the captured ADC data in the FIFO. 0: write zeros 1: store the channel id tag	RW	0
0	ADC_MODULE_ENABLE	ADC module enable bit. After programming all the steps and configuration registers, write a 1 to this bit to start conversion. Writing a 0 will disable the module (after the current conversion completes)	RW	0

Table 23-30. Register Call Summary for Register ADC_CTRL

ADC Functional Description • Power Management: [0]
ADC Programming Guide • Global Initialization: [1][2] • During Operation: [3][4]
ADC Register Manual • ADC Register Summary: [5]

Table 23-31. ADC_ADCSTAT

Address Offset	0x0000 0044		
Physical Address	0x4A26 4044	**Instance**	ADC
Description	ADC sequencer status. Software can read this register to find out the currently scheduled step id being converted on the ADC port. If desired to turn the controller off and then back on, then the STEP_ID bit should be checked and compared to IDLE state before enabling the ADC module again. Also, before enabling the ADC, the user should wait for the FSM_BUSY bit to read idle.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																							AFE_BUSY			FSM_BUSY	STEP_ID				

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reserved	R	0
8	AFE_BUSY	Monitor the AFE internal calibration (busy bit)	R	0
7:6	RESERVED	Reserved	R	0
5	FSM_BUSY	Status of OCP FSM and ADC FSM 0: idle 1: busy (conversion in progress)	R	0
4:0	STEP_ID	0x10: Idle 0x11 – 0x1F: Reserved 0x0 – 0xF: Step 1 – Step 16	R	0x10

Table 23-32. Register Call Summary for Register ADC_ADCSTAT

ADC Functional Description • Interrupts: [0]
ADC Programming Guide • During Operation: [1]
ADC Register Manual • ADC Register Summary: [2] • ADC Register Description: [3]

Table 23-33. ADC_ADCRANGE

Address Offset	0x0000 0048		
Physical Address	0x4A26 4048	**Instance**	ADC
Description	ADC range check register. This feature requires the range check interrupt bit to be enabled first. Also, the user can decide which channel input is compared by programming the RANGE_CHECK bit of the ADC_STEPCONFIGi registers. It is up to software to sort through FIFO data to determine which channel data was out of range.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								HIGH_RANGE_DATA								RESERVED				LOW_RANGE_DATA											

Bits	Field Name	Description	Type	Reset
31:28	RESERVED	Reserved	R	0
27:16	HIGH_RANGE_DATA	Sampled ADC data is compared to this value. If the sampled data is > HIGH_RANGE_DATA, then interrupt is generated.	RW	0
15:12	RESERVED	Reseved	R	0
11:0	LOW_RANGE_DATA	Sampled ADC data is compared to this value. If the sampled data is < LOW_RANGE_DATA, then interrupt is generated.	RW	0

Table 23-34. Register Call Summary for Register ADC_ADCRANGE

ADC Functional Description
- Interrupts: [0]

ADC Register Manual
- ADC Register Summary: [1]
- ADC Register Description: [2]

Table 23-35. ADC_ADC_CLKDIV

Address Offset	0x0000 004C		
Physical Address	0x4A26 404C	**Instance**	ADC
Description	The ADC_CLK input will go through this clock divider first before being sent to the AFE clock input . The ADC clock should be 6× slower than OCP clock.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																				ADC_CLKDIV											

Bits	Field Name	Description	Type	Reset
31:12	RESERVED	Reserved	R	0
11:0	ADC_CLKDIV	The input ADC clock will be divided by this value and sent to the AFE. Program to the value minus 1	RW	0

Table 23-36. Register Call Summary for Register ADC_ADC_CLKDIV

ADC Programming Guide
- Global Initialization: [0]

ADC Register Manual
- ADC Register Summary: [1]

Table 23-37. ADC_ADC_MISC

Address Offset	0x0000 0050		
Physical Address	0x4A26 4050	**Instance**	ADC
Description	Zero Offset Correction Using Efuse: Pro: – Covers the ground bounce due to package routing. Con: – one- time calibration; doesn't take into account the Si degradation over time – T-V variation not accounted Zero Offset Correction using Internal Calibration: Pro: – Dynamic Calibration – Accounts for device ageing, temperature and Voltage variation Con: – 320 clk cycles after every Power up un-usable for actual data conversion		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																AFE_SPARE_OUTPUT								AFE_SPARE_INPUT							

Bits	Field Name	Description	Type	Reset
31:16	RESERVED	Reserved	R	0
15:8	AFE_SPARE_OUTPUT	Connected to AFE Spare Output pins.	R	0
7:0	AFE_SPARE_INPUT	Connected to AFE Spare Input pins. bit 0: Error correction 0: disable 1: enable bit 1: Calibration 0: Use internal calibration 1: Use eFuse for offset bit 2: Internal calibration start (then monitor ADC_ADCSTAT[8] AFE_BUSY to see when AFE is finished with calibration. This is a one time event after power up) 0: No action 1: Start internal calibration	RW	0

Table 23-38. Register Call Summary for Register ADC_ADC_MISC

ADC Register Manual

- ADC Register Summary: [0]

Table 23-39. ADC_STEPENABLE

Address Offset	0x0000 0054		
Physical Address	0x4A26 4054	**Instance**	ADC
Description	This register contains the enable bit for each step in the sequencer. When all steps are disabled, the FSM will stay in IDLE state. These bits can be enabled or disabled dynamically during operation. When a write to this register occurs during operational mode, the HW will make sure the new settings are updated after the END_OF_SEQUENCE event (registers are shadowed and updated at that time). If a step is configured in one-shot mode, the HW will automatically reset the STEPi bit after it is scheduled.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED															STEP16	STEP15	STEP14	STEP13	STEP12	STEP11	STEP10	STEP9	STEP8	STEP7	STEP6	STEP5	STEP4	STEP3	STEP2	STEP1	RESERVED

Bits	Field Name	Description	Type	Reset
31:17	RESERVED	Reserved	R	0
16	STEP16	Enable step 16	RW	0
15	STEP15	Enable step 15	RW	0
14	STEP14	Enable step 14	RW	0
13	STEP13	Enable step 13	RW	0
12	STEP12	Enable step 12	RW	0
11	STEP11	Enable step 11	RW	0
10	STEP10	Enable step 10	RW	0
9	STEP9	Enable step 9	RW	0
8	STEP8	Enable step 8	RW	0
7	STEP7	Enable step 7	RW	0
6	STEP6	Enable step 6	RW	0
5	STEP5	Enable step 5	RW	0
4	STEP4	Enable step 4	RW	0
3	STEP3	Enable step 3	RW	0
2	STEP2	Enable step 2	RW	0
1	STEP1	Enable step 1	RW	0
0	RESERVED	Reserved	RW	0

Table 23-40. Register Call Summary for Register ADC_STEPENABLE

ADC Functional Description • Power Management: [0] • Operational Modes: [1][2]
ADC Programming Guide • Global Initialization: [3] • During Operation: [4]
ADC Register Manual • ADC Register Summary: [5]

Table 23-41. ADC_STEPCONFIGi

Address Offset	0x0000 0064 + (i * 8)	**Index**	i = 0 to 15
Physical Address	0x4A26 4064 + (I * 8)	**Instance**	ADC
Description	Values sent to the AFE during step i.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				RANGE_CHECK	FIFO_SELECT	RESERVED		SEL_INP_SWC								RESERVED								AVERAGING				MODE			

Bits	Field Name	Description	Type	Reset
31:28	RESERVED	Reserved	RW	0
27	RANGE_CHECK	0: Disable out-of-range check 1: Compare ADC data with ADC_ADCRANGE register	RW	0
26	FIFO_SELECT	Sampled data will be stored in FIFO. 0: FIFO 0 1: FIFO 1	RW	0

Bits	Field Name	Description	Type	Reset
25:23	RESERVED	Reserved	RW	0
22:19	SEL_INP_SWC	SEL_INP pins SW configuration. 0x0 = Channel 1 0x1 = Channel 2 0x2 = Channel 3 0x3 = Channel 4 0x4 = Channel 5 0x5 = Channel 6 0x6 = Channel 7 0x7 = Channel 8 0x8 = VREFN all other values = VREFN.	RW	0
18:5	RESERVED	Reserved	RW	0
4:2	AVERAGING	Number of samplings to average: 0x0: no average 0x1: 2 samples average 0x2: 4 samples average 0x3: 8 samples average 0x4: 16 samples average	RW	0
1:0	MODE	0x0: SW enabled, one-shot 0x1: SW enabled, continuous 0x2: HW synchronized, one-shot (NOT SUPPORTED IN THIS DEVICE) 0x3: HW synchronized, continuous (NOT SUPPORTED IN THIS DEVICE)	RW	0

Table 23-42. Register Call Summary for Register ADC_STEPCONFIGi

ADC Functional Description • ADC Functional Description: [0][1] • Averaging of Samples (1, 2, 4, 8, and 16): [2] • Operational Modes: [3][4][5][6]
ADC Programming Guide • Global Initialization: [7]
ADC Register Manual • ADC Register Summary: [8] • ADC Register Description: [9]

Table 23-43. ADC_STEPDELAYi

Address Offset	0x0000 0068 + (i * 8)	**Index**	i = 0 to 15
Physical Address	0x4A26 4068 + (I * 8)	**Instance**	ADC
Description	Step i delay register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
SAMPLEDELAY								RESERVED								OPENDELAY															

Bits	Field Name	Description	Type	Reset
31:24	SAMPLEDELAY	This register will control the number of ADC clock cycles to hold SOC high. Any value programmed here will be added to the minimum requirement of 1 clock cycle.	RW	0
23:18	RESERVED	Reserved	RW	0
17:0	OPENDELAY	Program the number of ADC clock cycles to wait after applying the step configuration registers and before sending the start of ADC conversion (SOC)	RW	0

Table 23-44. Register Call Summary for Register ADC_STEPDELAYi

ADC Functional Description • Open Delay: [0] • Operational Modes: [1][2][3]
ADC Programming Guide • Global Initialization: [4]
ADC Register Manual • ADC Register Summary: [5]

Table 23-45. ADC_FIFO0COUNT

Address Offset	0x0000 00E4		
Physical Address	0x4A26 40E4	**Instance**	ADC
Description	FIFO0 word count register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								WORDS_IN_FIFO							

Bits	Field Name	Description	Type	Reset
31:7	RESERVED	Reserved	R	0
6:0	WORDS_IN_FIFO	Number of words currently in the FIFO0	R	0

Table 23-46. Register Call Summary for Register ADC_FIFO0COUNT

ADC Functional Description • Interrupts: [0] • Operational Modes: [1]
ADC Register Manual • ADC Register Summary: [2]

Table 23-47. ADC_FIFO0THRESHOLD

Address Offset	0x0000 00E8		
Physical Address	0x4A26 40E8	**Instance**	ADC
Description	FIFO0 threshold level register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								FIFO_THRESHOLD_LEVEL							

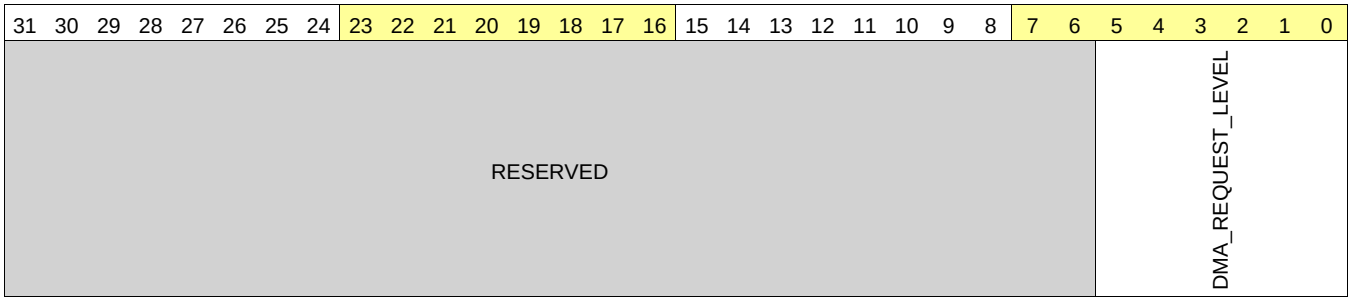
Bits	Field Name	Description	Type	Reset
31:6	RESERVED	Reserved	RW	0
5:0	FIFO_THRESHOLD_LEVEL	Program the desired FIFO0 data sample level to reach before generating interrupt to CPU (program to value minus 1)	RW	0

Table 23-48. Register Call Summary for Register ADC_FIFO0THRESHOLD

ADC Programming Guide • Global Initialization: [0]
ADC Register Manual • ADC Register Summary: [1]

Table 23-49. ADC_DMA0REQ

Address Offset	0x0000 00EC		
Physical Address	0x4A26 40EC	**Instance**	ADC
Description	FIFO0 DMA request level register		
Type	RW		



Bits	Field Name	Description	Type	Reset
31:6	RESERVED	Reserved	RW	0
5:0	DMA_REQUEST_LEVEL	Number of words in FIFO0 before generating a DMA request (program to value minus 1)	RW	0

Table 23-50. Register Call Summary for Register ADC_DMA0REQ

ADC Functional Description • DMA Requests: [0]
ADC Programming Guide • Global Initialization: [1]
ADC Register Manual • ADC Register Summary: [2]

Table 23-51. ADC_FIFO1COUNT

Address Offset	0x0000 00F0		
Physical Address	0x4A26 40F0	**Instance**	ADC
Description	FIFO1 word count register		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								WORDS_IN_FIFO							

Bits	Field Name	Description	Type	Reset
31:7	RESERVED	Reserved	R	0
6:0	WORDS_IN_FIFO	Number of words currently in the FIFO1	R	0

Table 23-52. Register Call Summary for Register ADC_FIFO1COUNT

ADC Functional Description
- Interrupts: [0]
- Operational Modes: [1]

ADC Register Manual
- ADC Register Summary: [2]

Table 23-53. ADC_FIFO1THRESHOLD

Address Offset	0x0000 00F4		
Physical Address	0x4A26 40F4	**Instance**	ADC
Description	FIFO1 threshold level register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																													FIFO_THRESHOLD_LEVEL		

Bits	Field Name	Description	Type	Reset
31:6	RESERVED	Reserved	RW	0
5:0	FIFO_THRESHOLD_LEVEL	Program the desired FIFO1 data sample level to reach before generating interrupt to CPU (program to value minus 1)	RW	0

Table 23-54. Register Call Summary for Register ADC_FIFO1THRESHOLD

ADC Programming Guide
- Global Initialization: [0]

ADC Register Manual
- ADC Register Summary: [1]

Table 23-55. ADC_DMA1REQ

Address Offset	0x0000 00F8		
Physical Address	0x4A26 40F8	**Instance**	ADC
Description	FIFO1 DMA request level register		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								DMA_REQUEST_LEVEL							

Bits	Field Name	Description	Type	Reset
31:6	RESERVED	Reserved	RW	0
5:0	DMA_REQUEST_LEVEL	Number of words in FIFO1 before generating a DMA request (program to value minus 1)	RW	0

Table 23-56. Register Call Summary for Register ADC_DMA1REQ

ADC Functional Description • DMA Requests: [0]
ADC Programming Guide • Global Initialization: [1]
ADC Register Manual • ADC Register Summary: [2]

Table 23-57. ADC_FIFO0DATA

Address Offset	0x0000 0100		
Physical Address	0x4A26 4100	**Instance**	ADC
Description	The CPU can read from this register to read data in FIFO 0. A read from this register will auto increment the FIFO read pointer. If software reads when FIFO is empty, an underflow interrupt will be triggered. All of the ADC conversions will be output as positive binary values.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												ADCCHNLID				RESERVED				ADCDATA											

Bits	Field Name	Description	Type	Reset
31:20	RESERVED	Reserved	R	0
19:16	ADCCHNLID	Optional ID tag of channel that captured the data. If tag option is disabled, these bits will be 0. 0x0: Step 1; 0x1: Step2;...; 0xF: Step 16.	R	0
15:12	RESERVED	Reserved	R	0
11:0	ADCDATA	12 bit sampled ADC converted data value stored in FIFO 0	R	0

Table 23-58. Register Call Summary for Register ADC_FIFO0DATA

ADC Register Manual

- ADC Register Summary: [0]

Table 23-59. ADC_FIFO1DATA

Address Offset	0x0000 0200		
Physical Address	0x4A26 4200	**Instance**	ADC
Description	The CPU can read from this register to read data in FIFO 1. A read from this register will auto increment the FIFO read pointer. If software reads when FIFO is empty, an underflow interrupt will be triggered. All of the ADC conversions will be output as positive binary values.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED												ADCCHNLID				RESERVED				ADCDATA											

Bits	Field Name	Description	Type	Reset
31:20	RESERVED	Reserved	R	0
19:16	ADCCHNLID	Optional ID tag of channel that captured the data. If tag option is disabled, these bits will be 0. 0x0: Step 1; 0x1: Step2;...; 0xF: Step 16.	R	0
15:12	RESERVED	Reserved	R	0
11:0	ADCDATA	12 bit sampled ADC converted data value stored in FIFO 1	R	0

Table 23-60. Register Call Summary for Register ADC_FIFO1DATA

ADC Register Manual

- ADC Register Summary: [0]

Real Time Interrupt Module

This chapter describes the Real Time Interrupt (RTI) module in the device.

Topic **Page**

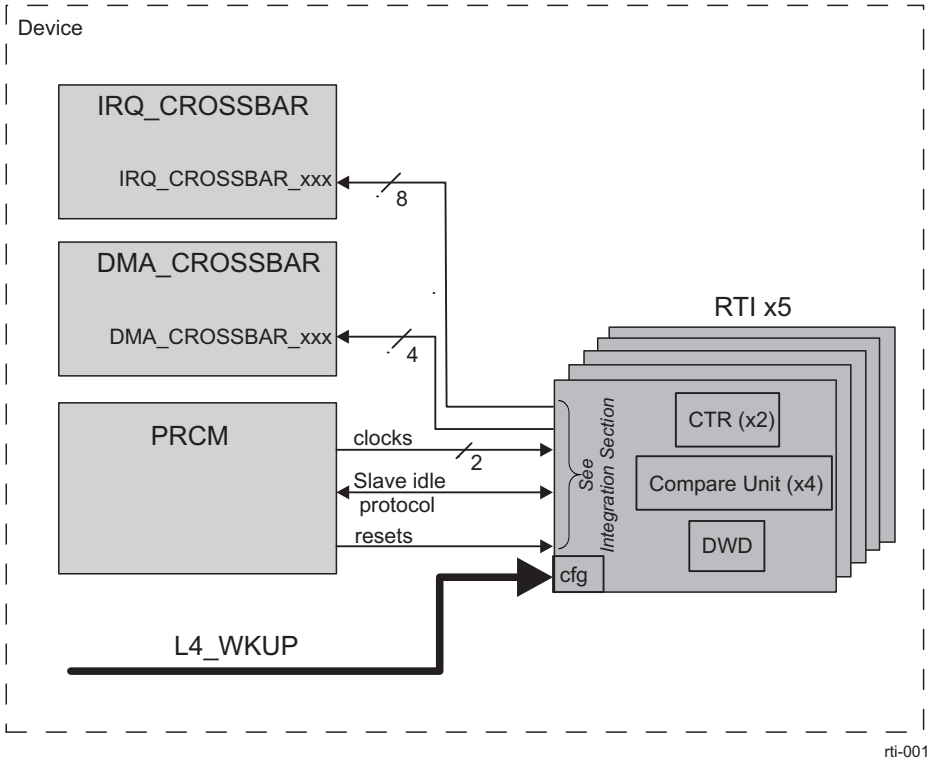
24.1 Real Time Interrupt Module

The Real Time Interrupt (RTI) module provides the following main features:

- Two independent 64 bit counter blocks (counter block 0 or counter block 1).
- Four configurable compares for generating operating system ticks or DMA requests. Each event can be driven by either counter block 0 or counter block 1.
- Fast enabling/disabling of events.
- Capture events to capture timestamps through recording of timer status.
- Two time-stamp (capture) functions for system or peripheral interrupts, one for each counter block.
- Digital Watchdog with an option to enable a windowed watchdog feature.

Figure 24-1 shows an overview block diagram of the RTI module within the device

Figure 24-1. RTI Overview



24.1.1 RTI Integration

This section describes the RTI integration in the device.

NOTE: For more information about the slave idle protocol and the wake-up request, see Section 3.7, *Power Management Functional Description*, in Chapter 3, *Power, Reset, and Clock Management*.

Table 24-1 through Table 24-3 summarize the integration of the module in the device.

Table 24-1. RTI Integration Attributes

Module Instance	Attributes	
	Power Domain	Interconnect
RTI1	PD_WKUPAON	L4_WKUP
RTI2	PD_WKUPAON	L4_WKUP

Table 24-1. RTI Integration Attributes (continued)

RTI3	PD_WKUPAON	L4_WKUP
RTI4	PD_WKUPAON	L4_WKUP
RTI5	PD_WKUPAON	L4_WKUP

Table 24-2. RTI Clocks and Resets

Clocks				
Module Instance	**Destination Signal Name**	**Source Signal Name**	**Source**	**Description**
RTI1	OCP_CLK_PI	L4_ICLK	PRCM	RTI1 Interface clock
RTI1	RTI_CLK_PI	RTI_CLK	PRCM	RTI1 counters clock
RTI2	OCP_CLK_PI	L4_ICLK	PRCM	RTI2 Interface clock
RTI2	RTI_CLK_PI	RTI_CLK	PRCM	RTI2 counters clock
RTI3	OCP_CLK_PI	L4_ICLK	PRCM	RTI3 Interface clock
RTI3	RTI_CLK_PI	RTI_CLK	PRCM	RTI3 counters clock
RTI4	OCP_CLK_PI	L4_ICLK	PRCM	RTI4 Interface clock
RTI4	RTI_CLK_PI	RTI_CLK	PRCM	RTI4 counters clock
RTI5	OCP_CLK_PI	L4_ICLK	PRCM	RTI5 Interface clock
RTI5	RTI_CLK_PI	RTI_CLK	PRCM	RTI5 counters clock
Resets				
Module Instance	**Destination Signal Name**	**Source Signal Name**	**Source**	**Description**
RTI1	rst_main_arst_n_pi	-	PRCM	RTI1 domain reset
RTI1	rst_por_arst_n_pi	-	PRCM	RTI1 POR only
RTI2	rst_main_arst_n_pi	-	PRCM	RTI2 domain reset
RTI2	rst_por_arst_n_pi	-	PRCM	RTI2 POR only
RTI3	rst_main_arst_n_pi	-	PRCM	RTI3 domain reset
RTI3	rst_por_arst_n_pi	-	PRCM	RTI3 POR only
RTI4	rst_main_arst_n_pi	-	PRCM	RTI4 domain reset
RTI4	rst_por_arst_n_pi	-	PRCM	RTI4 POR only
RTI5	rst_main_arst_n_pi	-	PRCM	RTI5 domain reset
RTI5	rst_por_arst_n_pi	-	PRCM	RTI5 POR only

Table 24-3. RTI Hardware Requests

Interrupt Requests				
Module Instance	**Source Signal Name**	**Destination IRQ_CROSSBAR Input**	**Default Mapping**	**Destination**
RTI1	RTI1_IRQ_WWD	IRQ_CROSSBAR_338	-	This IRQ source signal is not mapped by default to any device INTC.
RTI1	RTI1_IRQ_INT0	IRQ_CROSSBAR_402	-	This IRQ source signal is not mapped by default to any device INTC.
RTI1	RTI1_IRQ_INT1	IRQ_CROSSBAR_403	-	This IRQ source signal is not mapped by default to any device INTC.
RTI1	RTI1_IRQ_INT2	IRQ_CROSSBAR_408	-	This IRQ source signal is not mapped by default to any device INTC.
RTI1	RTI1_IRQ_INT3	IRQ_CROSSBAR_409	-	This IRQ source signal is not mapped by default to any device INTC.

Table 24-3. RTI Hardware Requests (continued)

RTI1	RTI1_IRQ_OVL0	IRQ_CROSSBAR_410	-	This IRQ source signal is not mapped by default to any device INTC.
RTI1	RTI1_IRQ_OVL1	IRQ_CROSSBAR_411	-	This IRQ source signal is not mapped by default to any device INTC.
RTI2	RTI2_IRQ_WWD	IRQ_CROSSBAR_75	-	This IRQ source signal is not mapped by default to any device INTC.
RTI2	RTI2_IRQ_INT0	IRQ_CROSSBAR_412	-	This IRQ source signal is not mapped by default to any device INTC.
RTI2	RTI2_IRQ_INT1	IRQ_CROSSBAR_425	-	This IRQ source signal is not mapped by default to any device INTC.
RTI2	RTI2_IRQ_INT2	IRQ_CROSSBAR_426	-	This IRQ source signal is not mapped by default to any device INTC.
RTI2	RTI2_IRQ_INT3	IRQ_CROSSBAR_427	-	This IRQ source signal is not mapped by default to any device INTC.
RTI2	RTI2_IRQ_OVL0	IRQ_CROSSBAR_428	-	This IRQ source signal is not mapped by default to any device INTC.
RTI2	RTI2_IRQ_OVL1	IRQ_CROSSBAR_429	-	This IRQ source signal is not mapped by default to any device INTC.
RTI3	RTI3_IRQ_WWD	IRQ_CROSSBAR_405	-	This IRQ source signal is not mapped by default to any device INTC.
RTI3	RTI3_IRQ_INT0	IRQ_CROSSBAR_430	-	This IRQ source signal is not mapped by default to any device INTC.
RTI3	RTI3_IRQ_INT1	IRQ_CROSSBAR_431	-	This IRQ source signal is not mapped by default to any device INTC.
RTI3	RTI3_IRQ_INT2	IRQ_CROSSBAR_432	-	This IRQ source signal is not mapped by default to any device INTC.
RTI3	RTI3_IRQ_INT3	IRQ_CROSSBAR_433	-	This IRQ source signal is not mapped by default to any device INTC.
RTI3	RTI3_IRQ_OVL0	IRQ_CROSSBAR_434	-	This IRQ source signal is not mapped by default to any device INTC.
RTI3	RTI3_IRQ_OVL1	IRQ_CROSSBAR_435	-	This IRQ source signal is not mapped by default to any device INTC.
RTI4	RTI4_IRQ_WWD	IRQ_CROSSBAR_406	-	This IRQ source signal is not mapped by default to any device INTC.
RTI4	RTI4_IRQ_INT0	IRQ_CROSSBAR_436	-	This IRQ source signal is not mapped by default to any device INTC.
RTI4	RTI4_IRQ_INT1	IRQ_CROSSBAR_437	-	This IRQ source signal is not mapped by default to any device INTC.
RTI4	RTI4_IRQ_INT2	IRQ_CROSSBAR_438	-	This IRQ source signal is not mapped by default to any device INTC.

Table 24-3. RTI Hardware Requests (continued)

RTI4	RTI4_IRQ_INT3	IRQ_CROSSBAR_439	-	This IRQ source signal is not mapped by default to any device INTC.
RTI4	RTI4_IRQ_OVL0	IRQ_CROSSBAR_440	-	This IRQ source signal is not mapped by default to any device INTC.
RTI4	RTI4_IRQ_OVL1	IRQ_CROSSBAR_441	-	This IRQ source signal is not mapped by default to any device INTC.
RTI4	RTI4_IRQ_TBINT	IRQ_CROSSBAR_451	-	This IRQ source signal is not mapped by default to any device INTC.
RTI5	RTI5_IRQ_WWD	IRQ_CROSSBAR_407	-	This IRQ source signal is not mapped by default to any device INTC.
RTI5	RTI5_IRQ_INT0	IRQ_CROSSBAR_442	-	This IRQ source signal is not mapped by default to any device INTC.
RTI5	RTI5_IRQ_INT1	IRQ_CROSSBAR_443	-	This IRQ source signal is not mapped by default to any device INTC.
RTI5	RTI5_IRQ_INT2	IRQ_CROSSBAR_444	-	This IRQ source signal is not mapped by default to any device INTC.
RTI5	RTI5_IRQ_INT3	IRQ_CROSSBAR_445	-	This IRQ source signal is not mapped by default to any device INTC.
RTI5	RTI5_IRQ_OVL0	IRQ_CROSSBAR_446	-	This IRQ source signal is not mapped by default to any device INTC.
RTI5	RTI5_IRQ_OVL1	IRQ_CROSSBAR_447	-	This IRQ source signal is not mapped by default to any device INTC.

DMA Requests

Module Instance	Source Signal Name	Destination DMA_CROSSBAR Input	Default Mapping	Destination
RTI1	RTI1_DREQ_EVT0	DMA_CROSSBAR_210	-	This DMA source signal is not mapped by default to any device DMA.
RTI1	RTI1_DREQ_EVT1	DMA_CROSSBAR_211	-	This DMA source signal is not mapped by default to any device DMA.
RTI1	RTI1_DREQ_EVT2	DMA_CROSSBAR_212	-	This DMA source signal is not mapped by default to any device DMA.
RTI1	RTI1_DREQ_EVT3	DMA_CROSSBAR_213	-	This DMA source signal is not mapped by default to any device DMA.
RTI2	RTI2_DREQ_EVT0	DMA_CROSSBAR_214	-	This DMA source signal is not mapped by default to any device DMA.
RTI2	RTI2_DREQ_EVT1	DMA_CROSSBAR_215	-	This DMA source signal is not mapped by default to any device DMA.
RTI2	RTI2_DREQ_EVT2	DMA_CROSSBAR_216	-	This DMA source signal is not mapped by default to any device DMA.
RTI2	RTI2_DREQ_EVT3	DMA_CROSSBAR_217	-	This DMA source signal is not mapped by default to any device DMA.

Table 24-3. RTI Hardware Requests (continued)

RTI3	RTI3_DREQ_EVT0	DMA_CROSSBAR_218	-	This DMA source signal is not mapped by default to any device DMA.
RTI3	RTI3_DREQ_EVT1	DMA_CROSSBAR_219	-	This DMA source signal is not mapped by default to any device DMA.
RTI3	RTI3_DREQ_EVT2	DMA_CROSSBAR_220	-	This DMA source signal is not mapped by default to any device DMA.
RTI3	RTI3_DREQ_EVT3	DMA_CROSSBAR_221	-	This DMA source signal is not mapped by default to any device DMA.
RTI4	RTI4_DREQ_EVT0	DMA_CROSSBAR_222	-	This DMA source signal is not mapped by default to any device DMA.
RTI4	RTI4_DREQ_EVT1	DMA_CROSSBAR_223	-	This DMA source signal is not mapped by default to any device DMA.
RTI4	RTI4_DREQ_EVT2	DMA_CROSSBAR_224	-	This DMA source signal is not mapped by default to any device DMA.
RTI4	RTI4_DREQ_EVT3	DMA_CROSSBAR_225	-	This DMA source signal is not mapped by default to any device DMA.
RTI5	RTI5_DREQ_EVT0	DMA_CROSSBAR_226	-	This DMA source signal is not mapped by default to any device DMA.
RTI5	RTI5_DREQ_EVT1	DMA_CROSSBAR_227	-	This DMA source signal is not mapped by default to any device DMA.
RTI5	RTI5_DREQ_EVT2	DMA_CROSSBAR_228	-	This DMA source signal is not mapped by default to any device DMA.
RTI5	RTI5_DREQ_EVT3	DMA_CROSSBAR_229	-	This DMA source signal is not mapped by default to any device DMA.

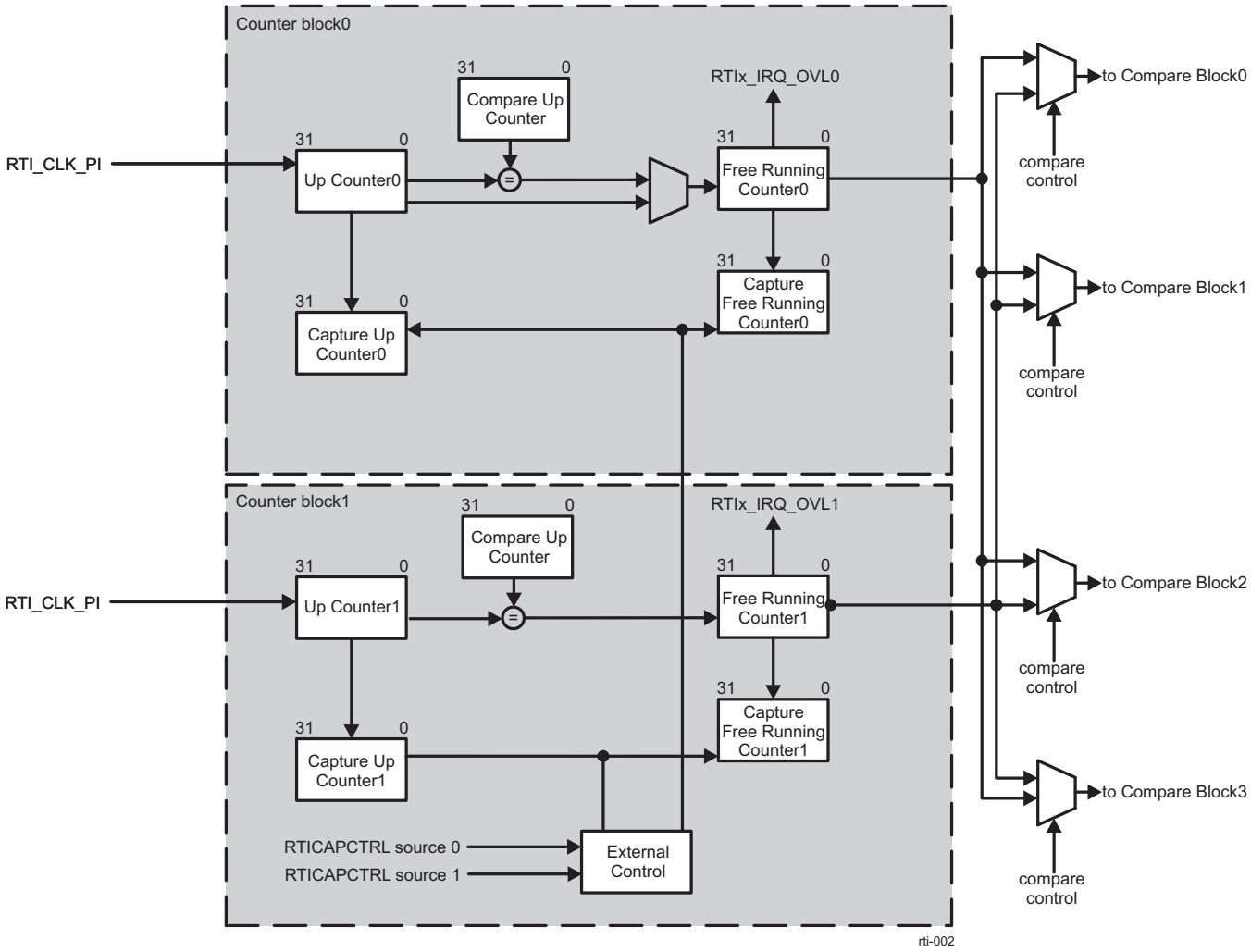
NOTE: The "**Default Mapping**" column in *RTI* shows the default mapping of module IRQ and DREQ source signals. These module IRQ and DREQ source signals can also be mapped to other lines of each device Interrupt or DMA controller through the IRQ_CROSSBAR and DMA_CROSSBAR modules, respectively.
For more information about the IRQ_CROSSBAR module, see Section 13.4.6.4, *IRQ_CROSSBAR Module Functional Description*, in Chapter 13, *Control Module*.
For more information about the DMA_CROSSBAR module, see Section 13.4.6.5, *DMA_CROSSBAR Module Functional Description*, in Chapter 13, *Control Module*.
For more information about the device interrupt controllers, see Chapter 12, *Interrupt Controllers*.
For more information about the device EDMA module, see Chapter 11, *Enhanced DMA*.

24.1.2 RTI Functional Description

24.1.2.1 RTI Counter Operation

Figure 24-2 shows the RTI module counter blocks. The RTI module supports two counter blocks.

Figure 24-2. RTI Counters Block diagram



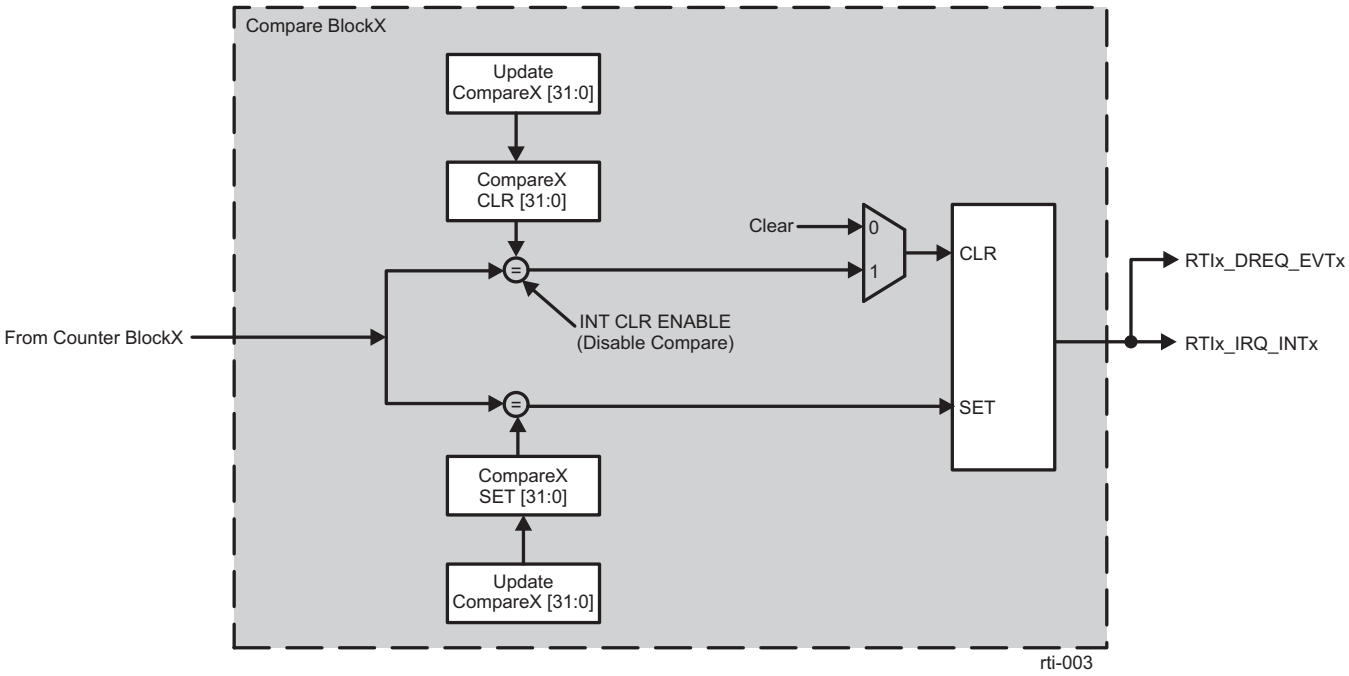
Each block consists of two 32 bit up counters - Up Counter and Free Running Counter (FRC). The Up Counter (RTIUC0 or RTIUC1 register) is driven by the RTI_CLK_PI and counts up until the compare value in the Compare Up Counter register (RTICPUC0 or RTICPUC1) is reached. When the compare matches, the second counter (RTIFRC0 or RTIFRC1 register), which is a free running counter is incremented. At the same time UCx is reset to zero. If FRCx overflows a interrupt is generated.

To ensure the consistency of the counters, when both counter value have to be determined, the Free Running Counter has to be read first. This will ensure that at the CPU read cycle, the Up Counter value is stored in the counter register. The second read is done on the Up Counter register, which holds then the value of the counter cycle of the previous read on the Free Running Counter register.

Both blocks provide also a capture feature on external events. Two capture sources can trigger the capture event. Which event triggers block 0 or block 1 is configurable from the RTICAPCTRL register . The sources are coming from the interrupt manager, in order to be able to generate a capture event when one of the peripheral modules has generated an interrupt. The peripheral, which can generate an event is configured in the interrupt manager. When the event is detected, UCx and FRCx are stored in Capture Up Counter (RTICAUC0 or RTICAUC1) and Capture Free Running Counter (RTICAFRC0 or RTICAFRC1) registers. The read order of the captured values has to be like the order of the actual counters. So the CAFRCx has to be read first and the CAUCx registers has to be read after the CAFRCx value was determined. While the CAFRCx is read the CAUCx value is loaded into a shadow register to ensure data consistency, when during the two reads of the captured data another capture event happens. If the application fails to read the two registers before a second capture event happens, the previous data will be overwritten.

Figure 24-3 shows the block diagram for one compare block. The RTI module supports four compare blocks.

Figure 24-3. RTI Compare Block Diagram



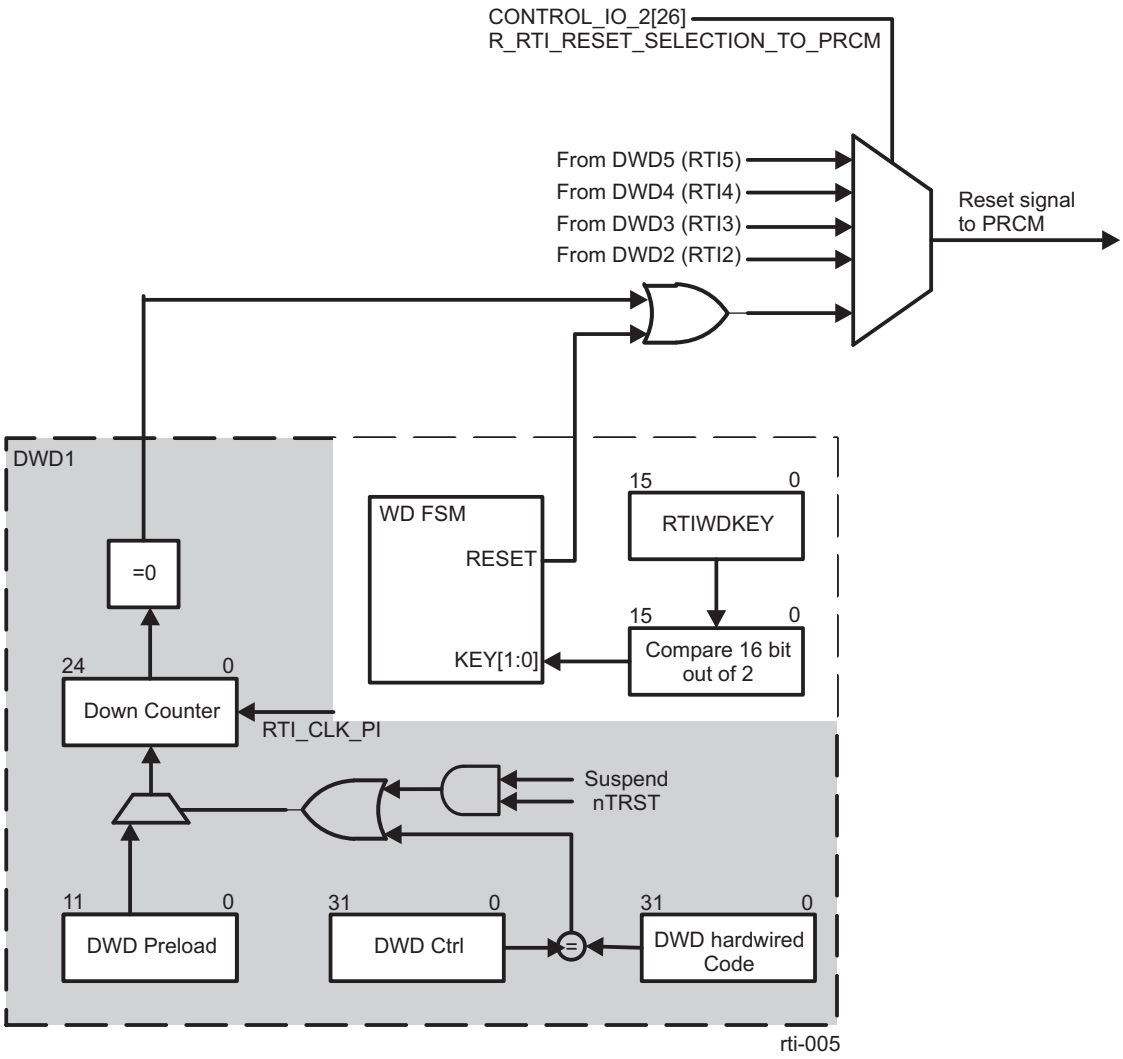
In order to generate interrupt requests to the interrupt manager, there are four compare registers (RTICOMP0, RTICOMP1, RTICOMP2, and RTICOMP3). Each of the compare registers can be configured to work either on FRC0(Counter block0) or FRC1(Counter block1). When the counter value matches the compare value, an interrupt is generated. This sets an interrupt request line to the interrupt manager. The compare value gets updated automatically with the value stored in Update Compare (RTIUDCP0, RTIUDCP1, RTIUDCP2, and RTIUDCP3) registers when the compare matches. This gives the ability to generate periodic interrupts/DMA requests without having to update the compare value by software.

An optional feature allows an application to program another compare value which is then used to clear the interrupt request line. This feature is supported by four compare clear registers (RTICOMP0CLR, RTICOMP1CLR, RTICOMP2CLR, and RTICOMP3CLR). When the counter value matches the compare clear value, the interrupt line is cleared. This clears the interrupt request line to the interrupt manager. The compare clear value gets updated automatically with the value stored in Update Compare (RTIUDCPx) registers when the compare matches.

24.1.2.2 RTI Digital Watchdog

Some applications might use a digital watchdog (DWD) integrated in the RTI module. The digital watchdog generates resets after a programmable period, if no correct key sequence is written to the RTIWDKEY register. Figure 24-4 show the Digital Watchdog functional block.

Figure 24-4. RTI Digital Watchdog Functional Block Diagram



The digital watchdog functionality is implemented such that it can be enabled by software.

The DWD starts counting down from the reset value of the RTIDWDCNTR (DWD Counter Register). The DWD preload register can be configured at any time by the application according to the desired time-out period.

When enabled by software, the digital watchdog is disabled after system reset. If it should be used, it has to be enabled by writing 0xA98559DA to the RTIDWDCTRL register. The DWD timeout period must be configured using the DWD preload register before the DWD is enabled. The DWD cannot be disabled by the application once it is enabled.

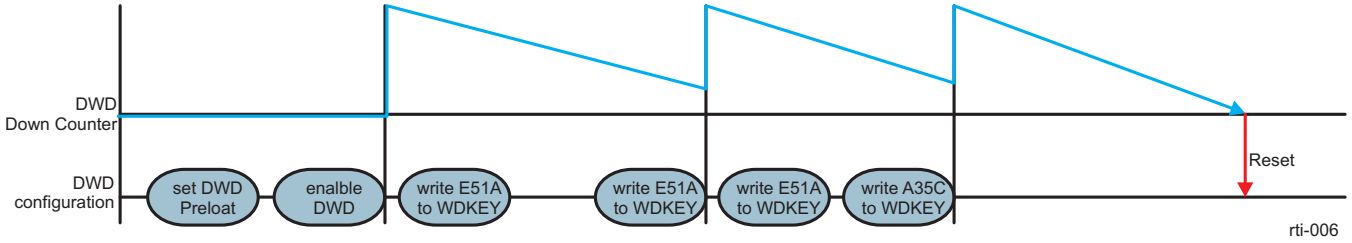
NOTE: When the DWD is enabled by software, any system reset will disable the DWD. This reset could have been generated by the watchdog itself.

If the correct key sequence is written to the RTIWDKEY register (0xE51A followed by 0xA35C), the 25-bit DWD Down Counter is reloaded with the 12-bit preload value stored in RTIDWDPRLD register. If any incorrect value is written to the RTIWDKEY register, a watchdog reset will occur immediately. A reset will also be generated, when the DWD Down Counter is decremented to 0.

The user has to take into account that the write to the DWDKEY register takes 3 OCP_CLK_PI cycles. This needs to be considered for the DWD expiration calculation.

The DWD Down Counter will be decremented with RTI_CLK_PI frequency. If the RTI_CLK_PI is switched off via the PRCM clock domain disable registers, the DWD counter stops decrementing. The DWD module cannot generate a reset under this condition.

Figure 24-5. RTI DWD Operation



The expiration time of the DWD Down Counter can be determined with following equation:

t_{exp}= (RTIDWDPRLD + 1) x 2^{13} / RTI_CLK_PI (12)

where RTIDWDPRLD = 0...4095

24.1.2.3 RTI Digital Windowed Watchdog

In addition to the time-out boundary configurable via the DWD discussed in previous section, some applications may also want to configure the start-time boundary of the watchdog. This is enabled by the digital windowed watchdog (DWWD) feature.

Functional Behavior

The DWWD opens a configurable time window in which the watchdog must be serviced. Any attempt to service the watchdog outside this time window, or a failure to service the watchdog in this time window, will cause the watchdog to generate either a reset or a non-maskable interrupt to the CPU. This is controlled by configuring the RTIWWDRXNCTRL register. As stated earlier, when the watchdog needs to be enabled by software, the watchdog counter is disabled on a system reset. When the DWWD is configured to generate a non-maskable interrupt on a window violation, the watchdog counter continues to count down. The RTIx_IRQ_WWD interrupt handler needs to clear the watchdog violation status flag(s) and then service the watchdog by writing the correct sequence in the watchdog key RTIWDKEY register. This service will cause the watchdog counter to get reloaded from the preload value and start counting down. If the RTIx_IRQ_WWD handler does not service the watchdog in time, it could count down all the way to zero and wrap around. No second exception for a time out is generated in this case.

Configuration of DWWD

The DWWD preload value (same as DWD preload) can only be configured when the DWWD counter is disabled. The window size and watchdog reaction to a violation can be configured even after the watchdog has been enabled. Any changes to the window size and watchdog reaction configurations will only take effect after the next servicing of the DWWD.

Figure 24-6. RTI Digital Windowed Watchdog Timing Example

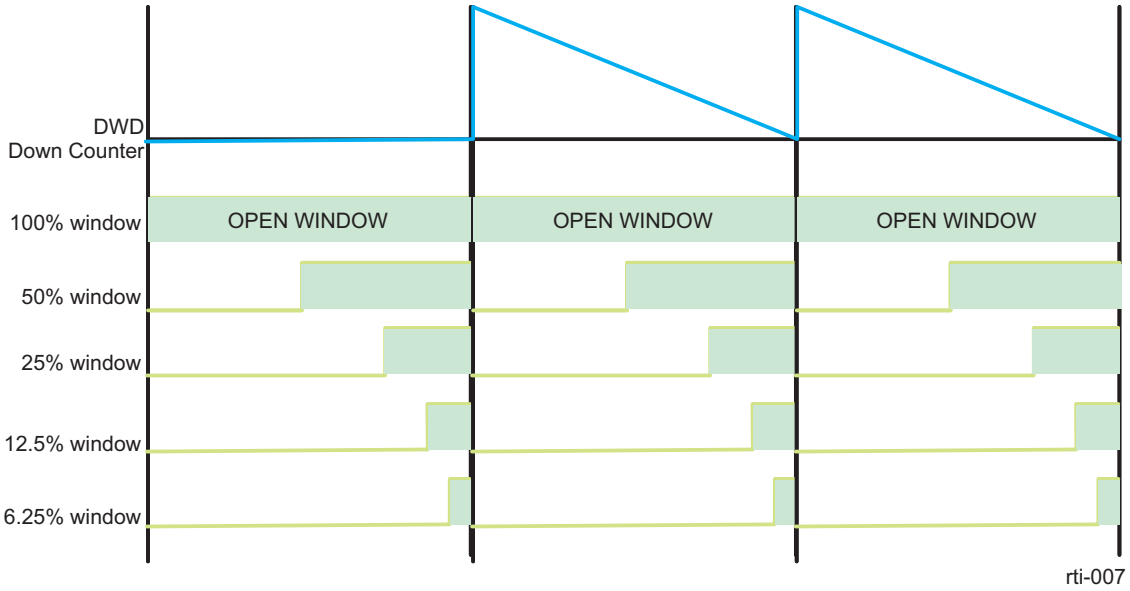
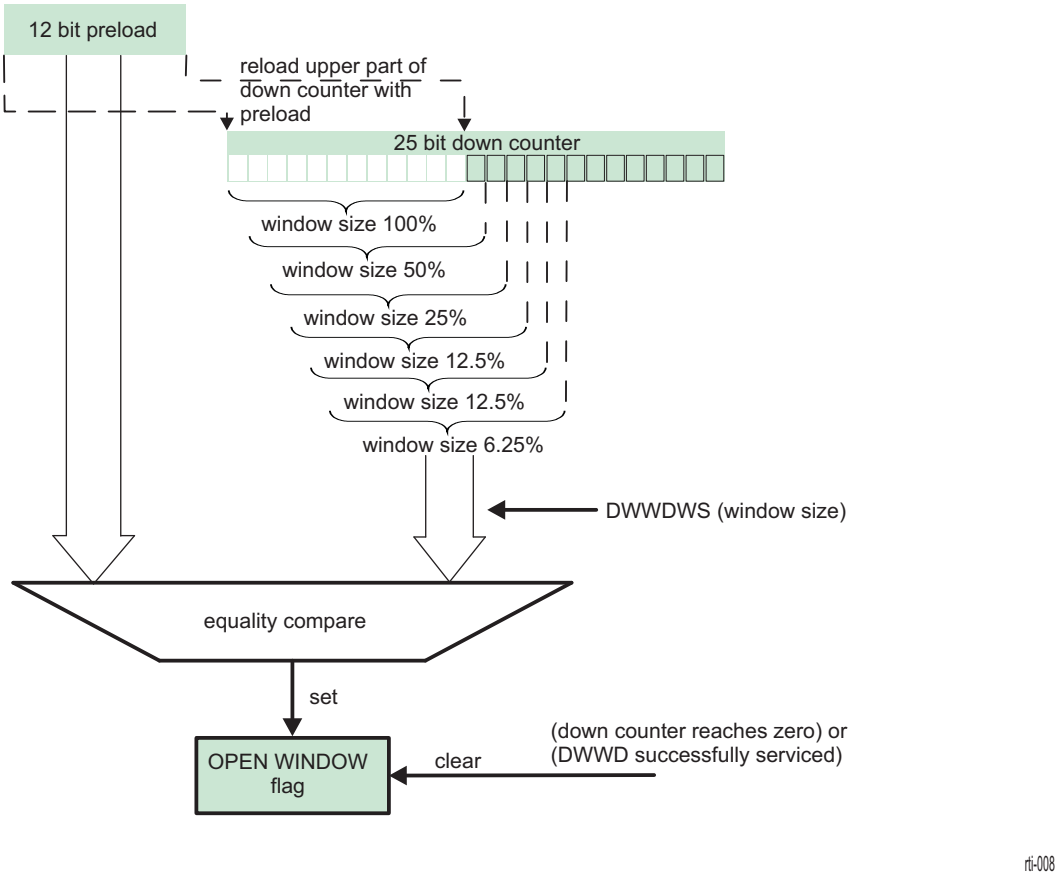


Figure 24-7. RTI Digital Windowed Watchdog Operation Block Diagram



24.1.2.4 RTI Low Power Mode Operation

The operation of the RTI Module is guaranteed in Run, Doze and Snooze mode. In Sleep or Hibernate mode all clocks will be switched off and the RTI Module will not work.

In Doze and Snooze modes all parts of the RTI are active, since it has to be able to wake up the device with Compare and Timebases interrupts. Capturing events generated by the Interrupt Module is also possible since in both modes the peripheral modules are able to generate interrupts, which can trigger capture events. The RTI Module will generate Compare, Timebases and Overflow interrupts. The Compare interrupts will periodically wake up the device. The Overflow interrupt will notify the operating system that a counter overflow occurred.

NOTE: In the special case of Doze Mode with DPLL off, RTI_CLK_PI might have a different period than with DPLL enabled, since RTI_CLK_PI will be derived from the oscillator output. It has to be ensured that the OCP_CLK_PI to RTI_CLK_PI ratio is at least 3:1.

The DWD/DWWD remains active when the device enters low power mode as long as the RTI_CLK_PI is kept active.

24.1.2.5 RTI Debug Mode Behavior

Once the system enters debug mode, the behavior of the RTI depends on the RTIGCTRL[15] COS bit. If the bit is cleared and debug mode is active, all counters will stop operation. If the bit is set to one, all counters will be clocked normally and the RTI will work like in normal mode. The DWD counter will not decrement in debug mode and will hold its current value.

NOTE: The user must not service the watchdog while in debug mode.

24.1.3 RTI Register Manual

24.1.3.1 RTI Instance Summary

Table 24-4. RTI Instance Summary

Module Name	Base Address L4_WKUP	Size
RTI1	0x4AE3 1000	288 Bytes
RTI2	0x4AE3 3000	288 Bytes
RTI3	0x4AE3 5000	288 Bytes
RTI4	0x4AE3 7000	288 Bytes
RTI5	0x4AE3 9000	288 Bytes

24.1.3.2 RTI registers

24.1.3.2.1 RTI Register Summary

Table 24-5. RTI Registers Mapping Summary 1

Register Name	Type	Register Width (Bits)	Address Offset	RTI1 Physical Address L4_WKUP	RTI2 Physical Address L4_WKUP	RTI3 Physical Address L4_WKUP
RTIGCTRL	RW	32	0x0000 0000	0x4AE3 1000	0x4AE3 3000	0x4AE3 5000
RESERVED	RW	32	0x0000 0004	0x4AE3 1004	0x4AE3 3004	0x4AE3 5004
RTICAPCTRL	RW	32	0x0000 0008	0x4AE3 1008	0x4AE3 3008	0x4AE3 5008
RTICOMPCTRL	RW	32	0x0000 000C	0x4AE3 100C	0x4AE3 300C	0x4AE3 500C
RTIFRC0	RW	32	0x0000 0010	0x4AE3 1010	0x4AE3 3010	0x4AE3 5010
RTIUC0	RW	32	0x0000 0014	0x4AE3 1014	0x4AE3 3014	0x4AE3 5014
RTICPUC0	RW	32	0x0000 0018	0x4AE3 1018	0x4AE3 3018	0x4AE3 5018

Table 24-5. RTI Registers Mapping Summary 1 (continued)

Register Name	Type	Register Width (Bits)	Address Offset	RTI1 Physical Address L4_WKUP	RTI2 Physical Address L4_WKUP	RTI3 Physical Address L4_WKUP
RTICAFRC0	R	32	0x0000 0020	0x4AE3 1020	0x4AE3 3020	0x4AE3 5020
RTICAUC0	R	32	0x0000 0024	0x4AE3 1024	0x4AE3 3024	0x4AE3 5024
RTIFRC1	RW	32	0x0000 0030	0x4AE3 1030	0x4AE3 3030	0x4AE3 5030
RTIUC1	RW	32	0x0000 0034	0x4AE3 1034	0x4AE3 3034	0x4AE3 5034
RTICPUC1	RW	32	0x0000 0038	0x4AE3 1038	0x4AE3 3038	0x4AE3 5038
RTICAFRC1	R	32	0x0000 0040	0x4AE3 1040	0x4AE3 3040	0x4AE3 5040
RTICAUC1	R	32	0x0000 0044	0x4AE3 1044	0x4AE3 3044	0x4AE3 5044
RTICOMP0	RW	32	0x0000 0050	0x4AE3 1050	0x4AE3 3050	0x4AE3 5050
RTIUDCP0	RW	32	0x0000 0054	0x4AE3 1054	0x4AE3 3054	0x4AE3 5054
RTICOMP1	RW	32	0x0000 0058	0x4AE3 1058	0x4AE3 3058	0x4AE3 5058
RTIUDCP1	RW	32	0x0000 005C	0x4AE3 105C	0x4AE3 305C	0x4AE3 505C
RTICOMP2	RW	32	0x0000 0060	0x4AE3 1060	0x4AE3 3060	0x4AE3 5060
RTIUDCP2	RW	32	0x0000 0064	0x4AE3 1064	0x4AE3 3064	0x4AE3 5064
RTICOMP3	RW	32	0x0000 0068	0x4AE3 1068	0x4AE3 3068	0x4AE3 5068
RTIUDCP3	RW	32	0x0000 006C	0x4AE3 106C	0x4AE3 306C	0x4AE3 506C
RESERVED	RW	32	0x0000 0070	0x4AE3 1070	0x4AE3 3070	0x4AE3 5070
RESERVED	RW	32	0x0000 0074	0x4AE3 1074	0x4AE3 3074	0x4AE3 5074
RTISETINT	RW	32	0x0000 0080	0x4AE3 1080	0x4AE3 3080	0x4AE3 5080
RTICLEARINT	RW	32	0x0000 0084	0x4AE3 1084	0x4AE3 3084	0x4AE3 5084
RTIINTFLAG	R	32	0x0000 0088	0x4AE3 1088	0x4AE3 3088	0x4AE3 5088
RTIDWDCTRL	RW	32	0x0000 0090	0x4AE3 1090	0x4AE3 3090	0x4AE3 5090
RTIDWDPRLD	RW	32	0x0000 0094	0x4AE3 1094	0x4AE3 3094	0x4AE3 5094
RTIWDSTATUS	R	32	0x0000 0098	0x4AE3 1098	0x4AE3 3098	0x4AE3 5098
RTIWDKEY	RW	32	0x0000 009C	0x4AE3 109C	0x4AE3 309C	0x4AE3 509C
RTIDWDCNTR	R	32	0x0000 00A0	0x4AE3 10A0	0x4AE3 30A0	0x4AE3 50A0
RTIWWDRXNCTRL	RW	32	0x0000 00A4	0x4AE3 10A4	0x4AE3 30A4	0x4AE3 50A4
RTIWWDSIZECTRL	RW	16	0x0000 00A8	0x4AE3 10A8	0x4AE3 30A8	0x4AE3 50A8
RTIINTCLRENABLE	RW	32	0x0000 00AC	0x4AE3 10AC	0x4AE3 30AC	0x4AE3 50AC
RTICOMP0CLR	RW	32	0x0000 00B0	0x4AE3 10B0	0x4AE3 30B0	0x4AE3 50B0
RTICOMP1CLR	RW	32	0x0000 00B4	0x4AE3 10B4	0x4AE3 30B4	0x4AE3 50B4
RTICOMP2CLR	RW	32	0x0000 00B8	0x4AE3 10B8	0x4AE3 30B8	0x4AE3 50B8
RTICOMP3CLR	RW	32	0x0000 00BC	0x4AE3 10BC	0x4AE3 30BC	0x4AE3 50BC

Table 24-6. RTI Registers Mapping Summary 2

Register Name	Type	Register Width (Bits)	Address Offset	RTI4 Physical Address L4_WKUP	RTI5 Physical Address L4_WKUP
RTIGCTRL	RW	32	0x0000 0000	0x4AE3 7000	0x4AE3 9000
RESERVED	RW	32	0x0000 0004	0x4AE3 7004	0x4AE3 9004
RTICAPCTRL	RW	32	0x0000 0008	0x4AE3 7008	0x4AE3 9008
RTICOMPCTRL	RW	32	0x0000 000C	0x4AE3 700C	0x4AE3 900C
RTIFRC0	RW	32	0x0000 0010	0x4AE3 7010	0x4AE3 9010
RTIUC0	RW	32	0x0000 0014	0x4AE3 7014	0x4AE3 9014

Table 24-6. RTI Registers Mapping Summary 2 (continued)

Register Name	Type	Register Width (Bits)	Address Offset	RTI4 Physical Address L4_WKUP	RTI5 Physical Address L4_WKUP
RTICPUC0	RW	32	0x0000 0018	0x4AE3 7018	0x4AE3 9018
RTICAFRC0	R	32	0x0000 0020	0x4AE3 7020	0x4AE3 9020
RTICAUC0	R	32	0x0000 0024	0x4AE3 7024	0x4AE3 9024
RTIFRC1	RW	32	0x0000 0030	0x4AE3 7030	0x4AE3 9030
RTIUC1	RW	32	0x0000 0034	0x4AE3 7034	0x4AE3 9034
RTICPUC1	RW	32	0x0000 0038	0x4AE3 7038	0x4AE3 9038
RTICAFRC1	R	32	0x0000 0040	0x4AE3 7040	0x4AE3 9040
RTICAUC1	R	32	0x0000 0044	0x4AE3 7044	0x4AE3 9044
RTICOMP0	RW	32	0x0000 0050	0x4AE3 7050	0x4AE3 9050
RTIUDCP0	RW	32	0x0000 0054	0x4AE3 7054	0x4AE3 9054
RTICOMP1	RW	32	0x0000 0058	0x4AE3 7058	0x4AE3 9058
RTIUDCP1	RW	32	0x0000 005C	0x4AE3 705C	0x4AE3 905C
RTICOMP2	RW	32	0x0000 0060	0x4AE3 7060	0x4AE3 9060
RTIUDCP2	RW	32	0x0000 0064	0x4AE3 7064	0x4AE3 9064
RTICOMP3	RW	32	0x0000 0068	0x4AE3 7068	0x4AE3 9068
RTIUDCP3	RW	32	0x0000 006C	0x4AE3 706C	0x4AE3 906C
RESERVED	RW	32	0x0000 0070	0x4AE3 7070	0x4AE3 9070
RESERVED	RW	32	0x0000 0074	0x4AE3 7074	0x4AE3 9074
RTISETINT	RW	32	0x0000 0080	0x4AE3 7080	0x4AE3 9080
RTICLEARINT	RW	32	0x0000 0084	0x4AE3 7084	0x4AE3 9084
RTIINTFLAG	R	32	0x0000 0088	0x4AE3 7088	0x4AE3 9088
RTIDWDCTRL	RW	32	0x0000 0090	0x4AE3 7090	0x4AE3 9090
RTIDWDPRLD	RW	32	0x0000 0094	0x4AE3 7094	0x4AE3 9094
RTIWDSTATUS	R	32	0x0000 0098	0x4AE3 7098	0x4AE3 9098
RTIWDKEY	RW	32	0x0000 009C	0x4AE3 709C	0x4AE3 909C
RTIDWDCNTR	R	32	0x0000 00A0	0x4AE3 70A0	0x4AE3 90A0
RTIWWDRXNCTRL	RW	32	0x0000 00A4	0x4AE3 70A4	0x4AE3 90A4
RTIWWDSIZECTRL	RW	16	0x0000 00A8	0x4AE3 70A8	0x4AE3 90A8
RTIINTCLRENABLE	RW	32	0x0000 00AC	0x4AE3 70AC	0x4AE3 90AC
RTICOMP0CLR	RW	32	0x0000 00B0	0x4AE3 70B0	0x4AE3 90B0
RTICOMP1CLR	RW	32	0x0000 00B4	0x4AE3 70B4	0x4AE3 90B4
RTICOMP2CLR	RW	32	0x0000 00B8	0x4AE3 70B8	0x4AE3 90B8
RTICOMP3CLR	RW	32	0x0000 00BC	0x4AE3 70BC	0x4AE3 90BC

24.1.3.2.2 RTI Register Description

Table 24-7. RTIGCTRL

Address Offset	0x0000 0000		
Physical Address	0x4AE3 1000	**Instance**	RTI1
	0x4AE3 3000		RTI1
	0x4AE3 3000		RTI2
	0x4AE3 5000		RTI1
	0x4AE3 5000		RTI3
	0x4AE3 7000		RTI2
	0x4AE3 7000		RTI4
	0x4AE3 9000		RTI2
	0x4AE3 9000		RTI5

Table 24-7. RTIGCTRL (continued)

Description	
Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								RESERVED								COS	RESERVED													CNT1EN	CNT0EN

Bits	Field Name	Description	Type	Reset
31:20	RESERVED	Reserved.	R	0x0
19:16	RESERVED	Reserved.	RW	0x0
15	COS	Continue On Suspend. This bit determines if both counters are stopped when the device goes into debug mode or if they continue counting. User and privilege mode (read): 0 = counters are stopped while in debug mode 1 = counters are running while in debug mode Privilege mode (write): 0 = stop counters in debug mode 1 = continue counting in debug mode	RW	0x0
14:2	RESERVED	Reserved.	R	0x0
1	CNT1EN	Counter 1 Enable. The CNT1EN bit starts and stops the operation of counter block 1 (UC1 and FRC1). User and privilege mode (read): 0 = counters are stopped 1 = counters are running Privilege mode (write): 0 = stop counters 1 = start counters	RW	0x0
0	CNT0EN	Counter 0 Enable. The CNT0EN bit starts and stops the operation of counter block 0 (UC0 and FRC0). User and privilege mode (read): 0 = counters are stopped 1 = counters are running Privilege mode (write): 0 = stop counters 1 = start counters	RW	0x0

Table 24-8. Register Call Summary for Register RTIGCTRL

Real Time Interrupt Module

- RTI Debug Mode Behavior: [0]
- RTI Register Summary: [1][2]
- RTI Register Description: [3][4][5][6]

Table 24-9. RTICAPCTRL

Address Offset	0x0000 0008		
Physical Address	0x4AE3 1008	**Instance**	RTI1
	0x4AE3 3008		RTI1
	0x4AE3 3008		RTI2
	0x4AE3 5008		RTI1
	0x4AE3 5008		RTI3
	0x4AE3 7008		RTI2
	0x4AE3 7008		RTI4
	0x4AE3 9008		RTI2
	0x4AE3 9008		RTI5
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																														CAPCNTR1	CAPCNTR0

Bits	Field Name	Description	Type	Reset
31:2	RESERVED		R	0x0
1	CAPCNTR1	Capture Counter 1.	RW	0x0
0	CAPCNTR0	Capture Counter 0. This bit determines, which external interrupt source triggers a capture event of both UC0 and FRC0. User and privilege mode (read): 0 = capture event is triggered by Capture Event Source 0 1 = capture event is triggered by Capture Event Source 1 Privilege mode (write): 0 = enable capture event triggered by Capture Event Source 0 1 = enable capture event triggered by Capture Event Source 1	RW	0x0

Table 24-10. Register Call Summary for Register RTICAPCTRL

Real Time Interrupt Module

- RTI Counter Operation: [0]
- RTI Register Summary: [1][2]

Table 24-11. RTICOMPCTRL

Address Offset	0x0000 000C		
Physical Address	0x4AE3 100C	**Instance**	RTI1
	0x4AE3 300C		RTI1
	0x4AE3 300C		RTI2
	0x4AE3 500C		RTI1
	0x4AE3 500C		RTI3
	0x4AE3 700C		RTI2
	0x4AE3 700C		RTI4
	0x4AE3 900C		RTI2
	0x4AE3 900C		RTI5
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																			COMPSEL3	RESERVED			COMPSEL2	RESERVED			COMPSEL1	RESERVED			COMPSEL0

Bits	Field Name	Description	Type	Reset
31:13	RESERVED		R	0x0
12	COMPSEL3	Compare Select 3. This bit determines the counter with which the compare value hold in compare register 3 is compared. User and privilege mode (read): 0 = value will be compared with FRC 0 1 = value will be compared with FRC 1 Privilege mode (write): 0 = enable compare with FRC 0 1 = enable compare with FRC 1	RW	0x0
11:9	RESERVED		R	0x0

Bits	Field Name	Description	Type	Reset
8	COMPSEL2	Compare Select 2. This bit determines the counter with which the compare value hold in compare register 2 is compared. User and privilege mode (read): 0 = value will be compared with FRC 0 1 = value will be compared with FRC 1 Privilege mode (write): 0 = enable compare with FRC 0 1 = enable compare with FRC 1	RW	0x0
7:5	RESERVED		R	0x0
4	COMPSEL1	Compare Select 1. This bit determines the counter with which the compare value hold in compare register 1 is compared. User and privilege mode (read): 0 = value will be compared with FRC 0 1 = value will be compared with FRC 1 Privilege mode (write): 0 = enable compare with FRC 0 1 = enable compare with FRC 1	RW	0x0
3:1	RESERVED		R	0x0
0	COMPSEL0	Compare Select 0. This bit determines the counter with which the compare value hold in compare register 0 is compared. User and privilege mode (read): 0 = value will be compared with FRC 0 1 = value will be compared with FRC 1 Privilege mode (write): 0 = enable compare with FRC 0 1 = enable compare with FRC 1	RW	0x0

Table 24-12. Register Call Summary for Register RTICOMPCTRL

Real Time Interrupt Module

- RTI Register Summary: [0][1]

Table 24-13. RTIFRC0

Address Offset	0x0000 0010		
Physical Address	0x4AE3 1010	**Instance**	RTI1
	0x4AE3 3010		RTI1
	0x4AE3 3010		RTI2
	0x4AE3 5010		RTI1
	0x4AE3 5010		RTI3
	0x4AE3 7010		RTI2
	0x4AE3 7010		RTI4
	0x4AE3 9010		RTI2
	0x4AE3 9010		RTI5
Description	This registers holds the current value of the Free Running Counter 0 and will be updated continuously.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FRC0																															

Bits	Field Name	Description	Type	Reset
31:0	FRC0	Free Running Counter 0. User and privilege mode (read): current value of the counter Privilege mode (write): The counter can be preset by writing to this register. The counter increments then from this written value upwards. Presetting counters, If counters have to be preset, they have to be stopped from counting in the RTIGCTRL register in order to ensure consistency between RTIUC0 and RTIFRC0.	RW	0x0

Table 24-14. Register Call Summary for Register RTIFRC0

Real Time Interrupt Module

- RTI Counter Operation: [0]
- RTI Register Summary: [1][2]
- RTI Register Description: [3][4]

Table 24-15. RTIUC0

Address Offset	0x0000 0014		
Physical Address	0x4AE3 1014	**Instance**	RTI1
	0x4AE3 3014		RTI1
	0x4AE3 3014		RTI2
	0x4AE3 5014		RTI1
	0x4AE3 5014		RTI3
	0x4AE3 7014		RTI2
	0x4AE3 7014		RTI4
	0x4AE3 9014		RTI2
	0x4AE3 9014		RTI5
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UC0																															

Bits	Field Name	Description	Type	Reset
31:0	UC0	Up Counter 0. User and privilege mode (read): value of the counter when the Free Running Counter 0 was read Privilege mode (write): the counter can be preset by writing to this register. The counter increments then from this written value upwards. Presetting counters: If counters have to be preset, they have to be stopped from counting in the RTIGCTRL register in order to ensure consistency between RTIUC0 and RTIFRC0. Preset value concern: If the preset value is bigger than the compare value stored in register RTICPUC0 then it can take a long time until a compare matches, since RTIUC0 has to count up until it overflows.	RW	0x0

Table 24-16. Register Call Summary for Register RTIUC0

Real Time Interrupt Module

- RTI Counter Operation: [0]
- RTI Register Summary: [1][2]
- RTI Register Description: [3][4][5]

Table 24-17. RTICPUC0

Address Offset	0x0000 0018		
Physical Address	0x4AE3 1018	**Instance**	RTI1
	0x4AE3 3018		RTI1
	0x4AE3 3018		RTI2
	0x4AE3 5018		RTI1
	0x4AE3 5018		RTI3
	0x4AE3 7018		RTI2
	0x4AE3 7018		RTI4
	0x4AE3 9018		RTI2
	0x4AE3 9018		RTI5
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CPUC0																															

Bits	Field Name	Description	Type	Reset
31:0	CPUC0	Compare Up Counter 0. ARRAY(0x1f04380) ARRAY(0x1f04410) User and privilege mode (read): current compare value Privilege mode (write when TBEXT = 0): the compare value is updated Privilege mode (write when TBEXT = 1): the compare value is not changed	RW	0x0

Table 24-18. Register Call Summary for Register RTICPUC0

Real Time Interrupt Module

- RTI Counter Operation: [0]
- RTI Register Summary: [1][2]
- RTI Register Description: [3]

Table 24-19. RTICAFRC0

Address Offset	0x0000 0020		
Physical Address	0x4AE3 1020	**Instance**	RTI1
	0x4AE3 3020		RTI1
	0x4AE3 3020		RTI2
	0x4AE3 5020		RTI1
	0x4AE3 5020		RTI3
	0x4AE3 7020		RTI2
	0x4AE3 7020		RTI4
	0x4AE3 9020		RTI2
	0x4AE3 9020		RTI5
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CAFRC0																															

Bits	Field Name	Description	Type	Reset
31:0	CAFRC0	Capture Free Running Counter 0. User and privilege mode (read): value of Free Running Counter 0 on a capture event	R	0x0

Table 24-20. Register Call Summary for Register RTICAFRC0

Real Time Interrupt Module

- RTI Counter Operation: [0]
- RTI Register Summary: [1][2]

Table 24-21. RTICAUC0

Address Offset	0x0000 0024		
Physical Address	0x4AE3 1024	**Instance**	RTI1
	0x4AE3 3024		RTI1
	0x4AE3 3024		RTI2
	0x4AE3 5024		RTI1
	0x4AE3 5024		RTI3
	0x4AE3 7024		RTI2
	0x4AE3 7024		RTI4
	0x4AE3 9024		RTI2
	0x4AE3 9024		RTI5

Table 24-21. RTICAUC0 (continued)

Description	
Type	R

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CAUC0																															

Bits	Field Name	Description	Type	Reset
31:0	CAUC0	Capture Up Counter 0. User and privilege mode (read): value of Up Counter 0 on a capture event	R	0x0

Table 24-22. Register Call Summary for Register RTICAUC0

Real Time Interrupt Module
- RTI Counter Operation: [0]
- RTI Register Summary: [1][2]

Table 24-23. RTIFRC1

Address Offset	0x0000 0030		
Physical Address	0x4AE3 1030	**Instance**	RTI1
	0x4AE3 3030		RTI1
	0x4AE3 3030		RTI2
	0x4AE3 5030		RTI1
	0x4AE3 5030		RTI3
	0x4AE3 7030		RTI2
	0x4AE3 7030		RTI4
	0x4AE3 9030		RTI2
	0x4AE3 9030		RTI5
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
FRC1																															

Bits	Field Name	Description	Type	Reset
31:0	FRC1	Free Running Counter 1. User and privilege mode (read): current value of the counter Privilege mode (write): the counter can be preset by writing to this register. The counter increments then from this written value upwards. If counters have to be preset, they have to be stopped from counting in the RTIGCTRL register in order to ensure consistency between UC1 and FRC1.	RW	0x0

Table 24-24. Register Call Summary for Register RTIFRC1

Real Time Interrupt Module
- RTI Counter Operation: [0]
- RTI Register Summary: [1][2]

Table 24-25. RTIUC1

Address Offset	0x0000 0034		
Physical Address	0x4AE3 1034	**Instance**	RTI1
	0x4AE3 3034		RTI1
	0x4AE3 3034		RTI2
	0x4AE3 5034		RTI1
	0x4AE3 5034		RTI3
	0x4AE3 7034		RTI2
	0x4AE3 7034		RTI4
	0x4AE3 9034		RTI2
	0x4AE3 9034		RTI5
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UC1																															

Bits	Field Name	Description	Type	Reset
31:0	UC1	Up Counter 1. User and privilege mode (read): value of the counter when the Free Running Counter 1 was read Privilege mode (write): the counter can be preset by writing to this register. The counter increments then from this written value upwards. Presetting counters: If counters have to be preset, they have to be disabled in the RTIGCTRL register in order to ensure consistency between UC1 and FRC1. Preset value concern: If the preset value is bigger than the compare value stored in register RTICPUC1 then it can take a long time until a compare matches, since UC1 has to count up until it overflows.	RW	0x0

Table 24-26. Register Call Summary for Register RTIUC1

Real Time Interrupt Module

- RTI Counter Operation: [0]
- RTI Register Summary: [1][2]

Table 24-27. RTICPUC1

Address Offset	0x0000 0038		
Physical Address	0x4AE3 1038	**Instance**	RTI1
	0x4AE3 3038		RTI1
	0x4AE3 3038		RTI2
	0x4AE3 5038		RTI1
	0x4AE3 5038		RTI3
	0x4AE3 7038		RTI2
	0x4AE3 7038		RTI4
	0x4AE3 9038		RTI2
	0x4AE3 9038		RTI5
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CPUC1																															

Bits	Field Name	Description	Type	Reset
31:0	CPUC1	Compare Up Counter 1. ARRAY(0x1f1c750) ARRAY(0x1f1c7e0) User and privilege mode (read): current compare value Privilege mode (write): the compare value is updated	RW	0x0

Table 24-28. Register Call Summary for Register RTICPUC1

Real Time Interrupt Module

- RTI Counter Operation: [0]
- RTI Register Summary: [1][2]
- RTI Register Description: [3]

Table 24-29. RTICAFRC1

Address Offset	0x0000 0040		
Physical Address	0x4AE3 1040	**Instance**	RTI1
	0x4AE3 3040		RTI1
	0x4AE3 3040		RTI2
	0x4AE3 5040		RTI1
	0x4AE3 5040		RTI3
	0x4AE3 7040		RTI2
	0x4AE3 7040		RTI4
	0x4AE3 9040		RTI2
	0x4AE3 9040		RTI5
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CAFRC1																															

Bits	Field Name	Description	Type	Reset
31:0	CAFRC1	Capture Free Running Counter 1. User and privilege mode (read): value of Free Running Counter 1 on a capture event	R	0x0

Table 24-30. Register Call Summary for Register RTICAFRC1

Real Time Interrupt Module

- RTI Counter Operation: [0]
- RTI Register Summary: [1][2]

Table 24-31. RTICAUC1

Address Offset	0x0000 0044		
Physical Address	0x4AE3 1044	**Instance**	RTI1
	0x4AE3 3044		RTI1
	0x4AE3 3044		RTI2
	0x4AE3 5044		RTI1
	0x4AE3 5044		RTI3
	0x4AE3 7044		RTI2
	0x4AE3 7044		RTI4
	0x4AE3 9044		RTI2
	0x4AE3 9044		RTI5
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
CAUC1																															

Bits	Field Name	Description	Type	Reset
31:0	CAUC1	Capture Up Counter 1. User and privilege mode (read): value of Up Counter 1 on a capture event	R	0x0

Table 24-32. Register Call Summary for Register RTICAUC1

Real Time Interrupt Module
- RTI Counter Operation: [0]
- RTI Register Summary: [1][2]

Table 24-33. RTICOMP0

Address Offset	0x0000 0050		
Physical Address	0x4AE3 1050	**Instance**	RTI1
	0x4AE3 3050		RTI1
	0x4AE3 3050		RTI2
	0x4AE3 5050		RTI1
	0x4AE3 5050		RTI3
	0x4AE3 7050		RTI2
	0x4AE3 7050		RTI4
	0x4AE3 9050		RTI2
	0x4AE3 9050		RTI5
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
COMP0																															

Bits	Field Name	Description	Type	Reset
31:0	COMP0	Compare 0. User and privilege mode (read): current compare value Privilege mode (write): update of the compare register with a new compare value Reset behavior: A reset does not generate a compare match, since the compare logic will only be active, when the associated counter block is enabled.	RW	0x0

Table 24-34. Register Call Summary for Register RTICOMP0

Real Time Interrupt Module
- RTI Counter Operation: [0]
- RTI Register Summary: [1][2]

Table 24-35. RTIUDCP0

Address Offset	0x0000 0054		
Physical Address	0x4AE3 1054	**Instance**	RTI1
	0x4AE3 3054		RTI1
	0x4AE3 3054		RTI2
	0x4AE3 5054		RTI1
	0x4AE3 5054		RTI3
	0x4AE3 7054		RTI2
	0x4AE3 7054		RTI4
	0x4AE3 9054		RTI2
	0x4AE3 9054		RTI5
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UDCP0																															

Bits	Field Name	Description	Type	Reset
31:0	UDCP0	Update Compare 0 Register. User and privilege mode (read): value to be added to the compare 0 register on the next compare match Privilege mode (write): new update value	RW	0x0

Table 24-36. Register Call Summary for Register RTIUDCP0

Real Time Interrupt Module

- RTI Counter Operation: [0]
- RTI Register Summary: [1][2]

Table 24-37. RTICOMP1

Address Offset	0x0000 0058		
Physical Address	0x4AE3 1058	**Instance**	RTI1
	0x4AE3 3058		RTI1
	0x4AE3 3058		RTI2
	0x4AE3 5058		RTI1
	0x4AE3 5058		RTI3
	0x4AE3 7058		RTI2
	0x4AE3 7058		RTI4
	0x4AE3 9058		RTI2
	0x4AE3 9058		RTI5
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
COMP1																															

Bits	Field Name	Description	Type	Reset
31:0	COMP1	Compare 1. User and privilege mode (read): current compare value Privilege mode (write): update of the compare register with a new compare value Reset behavior: A reset does not generate a compare match, since the compare logic will only be active, when the associated counter block is enabled.	RW	0x0

Table 24-38. Register Call Summary for Register RTICOMP1

Real Time Interrupt Module

- RTI Counter Operation: [0]
- RTI Register Summary: [1][2]

Table 24-39. RTIUDCP1

Address Offset	0x0000 005C		
Physical Address	0x4AE3 105C	**Instance**	RTI1
	0x4AE3 305C		RTI1
	0x4AE3 305C		RTI2
	0x4AE3 505C		RTI1
	0x4AE3 505C		RTI3
	0x4AE3 705C		RTI2
	0x4AE3 705C		RTI4
	0x4AE3 905C		RTI2
	0x4AE3 905C		RTI5

Table 24-39. RTIUDCP1 (continued)

Description	
Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UDCP1																															

Bits	Field Name	Description	Type	Reset
31:0	UDCP1	Update Compare 1 Register. User and privilege mode (read): value to be added to the compare 1 register on the next compare match Privilege mode (write): new update value	RW	0x0

Table 24-40. Register Call Summary for Register RTIUDCP1

Real Time Interrupt Module
- RTI Counter Operation: [0]
- RTI Register Summary: [1][2]

Table 24-41. RTICOMP2

Address Offset	0x0000 0060		
Physical Address	0x4AE3 1060	**Instance**	RTI1
	0x4AE3 3060		RTI1
	0x4AE3 3060		RTI2
	0x4AE3 5060		RTI1
	0x4AE3 5060		RTI3
	0x4AE3 7060		RTI2
	0x4AE3 7060		RTI4
	0x4AE3 9060		RTI2
	0x4AE3 9060		RTI5
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
COMP2																															

Bits	Field Name	Description	Type	Reset
31:0	COMP2	Compare 2. User and privilege mode (read): current compare value Privilege mode (write): update of the compare register with a new compare value Reset behavior: A reset does not generate a compare match, since the compare logic will only be active, when the associated counter block is enabled.	RW	0x0

Table 24-42. Register Call Summary for Register RTICOMP2

Real Time Interrupt Module
- RTI Counter Operation: [0]
- RTI Register Summary: [1][2]

Table 24-43. RTIUDCP2

Address Offset	0x0000 0064		
Physical Address	0x4AE3 1064	**Instance**	RTI1
	0x4AE3 3064		RTI1
	0x4AE3 3064		RTI2
	0x4AE3 5064		RTI1
	0x4AE3 5064		RTI3
	0x4AE3 7064		RTI2
	0x4AE3 7064		RTI4
	0x4AE3 9064		RTI2
	0x4AE3 9064		RTI5
Description	This registers holds a value, which is added to the value in the compare 2 register each time a compare matches. This gives the possibility to generate periodic interrupts without software intervention.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UDCP2																															

Bits	Field Name	Description	Type	Reset
31:0	UDCP2	Update Compare 2 Register. User and privilege mode (read): value to be added to the compare 2 register on the next compare match Privilege mode (write): new update value	RW	0x0

Table 24-44. Register Call Summary for Register RTIUDCP2

Real Time Interrupt Module

- RTI Counter Operation: [0]
- RTI Register Summary: [1][2]

Table 24-45. RTICOMP3

Address Offset	0x0000 0068		
Physical Address	0x4AE3 1068	**Instance**	RTI1
	0x4AE3 3068		RTI1
	0x4AE3 3068		RTI2
	0x4AE3 5068		RTI1
	0x4AE3 5068		RTI3
	0x4AE3 7068		RTI2
	0x4AE3 7068		RTI4
	0x4AE3 9068		RTI2
	0x4AE3 9068		RTI5
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
COMP3																															

Bits	Field Name	Description	Type	Reset
31:0	COMP3	Compare 3. User and privilege mode (read): current compare value Privilege mode (write): update of the compare register with a new compare value Reset behavior: A reset does not generate a compare match, since the compare logic will only be active, when the associated counter block is enabled.	RW	0x0

Table 24-46. Register Call Summary for Register RTICOMP3

Real Time Interrupt Module

- RTI Counter Operation: [0]
- RTI Register Summary: [1][2]

Table 24-47. RTIUDCP3

Address Offset	0x0000 006C		
Physical Address	0x4AE3 106C	**Instance**	RTI1
	0x4AE3 306C		RTI1
	0x4AE3 306C		RTI2
	0x4AE3 506C		RTI1
	0x4AE3 506C		RTI3
	0x4AE3 706C		RTI2
	0x4AE3 706C		RTI4
	0x4AE3 906C		RTI2
	0x4AE3 906C		RTI5
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
UDCP3																															

Bits	Field Name	Description	Type	Reset
31:0	UDCP3	Update Compare 3 Register. User and privilege mode (read): value to be added to the compare 3 register on the next compare match Privilege mode (write): new update value	RW	0x0

Table 24-48. Register Call Summary for Register RTIUDCP3

Real Time Interrupt Module

- RTI Counter Operation: [0]
- RTI Register Summary: [1][2]

Table 24-49. RTISETINT

Address Offset	0x0000 0080		
Physical Address	0x4AE3 1080	**Instance**	RTI1
	0x4AE3 3080		RTI1
	0x4AE3 3080		RTI2
	0x4AE3 5080		RTI1
	0x4AE3 5080		RTI3
	0x4AE3 7080		RTI2
	0x4AE3 7080		RTI4
	0x4AE3 9080		RTI2
	0x4AE3 9080		RTI5
Description	This register prevents the necessity of a read-modify-write operation if a particular interrupt should be enabled.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED													SETOVL1INT	SETOVL0INT	RESERVED	RESERVED				SETDMA3	SETDMA2	SETDMA1	SETDMA0	RESERVED				SETINT3	SETINT2	SETINT1	SETINT0

Bits	Field Name	Description	Type	Reset
31:19	RESERVED	Reserved.	R	0x0
18	SETOVL1INT	Set Free Running Counter 1 Overflow Interrupt. User and privilege mode (read): 0 = interrupt is disabled 1 = interrupt is enabled Privilege mode (write): 0 = leaves the corresponding bit unchanged 1 = enable interrupt	RW	0x0
17	SETOVL0INT	Set Free Running Counter 0 Overflow Interrupt. User and privilege mode (read): 0 = interrupt is disabled 1 = interrupt is enabled Privilege mode (write): 0 = leaves the corresponding bit unchanged 1 = enable interrupt	RW	0x0
16	RESERVED	Reserved.	RW	0x0
15:12	RESERVED	Reserved.	R	0x0
11	SETDMA3	Set Compare DMA Request 3. User and privilege mode (read): 0 = DMA request is disabled 1 = DMA request is enabled Privilege mode (write): 0 = leaves the corresponding bit unchanged 1 = enable DMA request	RW	0x0
10	SETDMA2	Set Compare DMA Request 2. User and privilege mode (read): 0 = DMA request is disabled 1 = DMA request is enabled Privilege mode (write): 0 = leaves the corresponding bit unchanged 1 = enable DMA request	RW	0x0
9	SETDMA1	Set Compare DMA Request 1. User and privilege mode (read): 0 = DMA request is disabled 1 = DMA request is enabled Privilege mode (write): 0 = leaves the corresponding bit unchanged 1 = enable DMA request	RW	0x0
8	SETDMA0	Set Compare DMA Request 0. User and privilege mode (read): 0 = DMA request is disabled 1 = DMA request is enabled Privilege mode (write): 0 = leaves the corresponding bit unchanged 1 = enable DMA request	RW	0x0
7:4	RESERVED	Reserved.	R	0x0
3	SETINT3	Set Compare Interrupt 3. User and privilege mode (read): 0 = interrupt is disabled 1 = interrupt is enabled Privilege mode (write): 0 = leaves the corresponding bit unchanged 1 = enable interrupt	RW	0x0
2	SETINT2	Set Compare Interrupt 2. User and privilege mode (read): 0 = interrupt is disabled 1 = interrupt is enabled Privilege mode (write): 0 = leaves the corresponding bit unchanged 1 = enable interrupt	RW	0x0
1	SETINT1	Set Compare Interrupt 1. User and privilege mode (read): 0 = interrupt is disabled 1 = interrupt is enabled Privilege mode (write): 0 = leaves the corresponding bit unchanged 1 = enable interrupt	RW	0x0
0	SETINT0	Set Compare Interrupt 0. User and privilege mode (read): 0 = interrupt is disabled 1 = interrupt is enabled Privilege mode (write): 0 = leaves the corresponding bit unchanged 1 = enable interrupt	RW	0x0

Table 24-50. Register Call Summary for Register RTISETINT

Real Time Interrupt Module

- RTI Register Summary: [0][1]

Table 24-51. RTICLEARINT

Address Offset	0x0000 0084		
Physical Address	0x4AE3 1084	**Instance**	RTI1
	0x4AE3 3084		RTI1
	0x4AE3 3084		RTI2
	0x4AE3 5084		RTI1
	0x4AE3 5084		RTI3
	0x4AE3 7084		RTI2
	0x4AE3 7084		RTI4
	0x4AE3 9084		RTI2
	0x4AE3 9084		RTI5

Table 24-51. RTICLEARINT (continued)

Description	This register prevents the necessity of a read-modify-write operation if a particular interrupt should be disabled.
Type	RW

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED													CLEAROVL1INT	CLEAROVL0INT	RESERVED	RESERVED				CLEARDMA3	CLEARDMA2	CLEARDMA1	CLEARDMA0	RESERVED				CLEARINT3	CLEARINT2	CLEARINT1	CLEARINT0

Bits	Field Name	Description	Type	Reset
31:19	RESERVED	Reserved.	R	0x0
18	CLEAROVL1INT	Clear Free Running Counter 1 Overflow Interrupt. User and privilege mode (read): 0 = interrupt is disabled 1 = interrupt is enabled Privilege mode (write): 0 = leaves the corresponding bit unchanged 1 = disable interrupt	RW	0x0
17	CLEAROVL0INT	Clear Free Running Counter 0 Overflow Interrupt. User and privilege mode (read): 0 = interrupt is disabled 1 = interrupt is enabled Privilege mode (write): 0 = leaves the corresponding bit unchanged 1 = disable interrupt	RW	0x0
16	RESERVED	Reserved.	RW	0x0
15:12	RESERVED	Reserved.	R	0x0
11	CLEARDMA3	Clear Compare DMA Request 3. User and privilege mode (read): 0 = DMA request is disabled 1 = DMA request is enabled Privilege mode (write): 0 = leaves the corresponding bit unchanged 1 = disable DMA request	RW	0x0
10	CLEARDMA2	Clear Compare DMA Request 2. User and privilege mode (read): 0 = DMA request is disabled 1 = DMA request is enabled Privilege mode (write): 0 = leaves the corresponding bit unchanged 1 = disable DMA request	RW	0x0
9	CLEARDMA1	Clear Compare DMA Request 1. User and privilege mode (read): 0 = DMA request is disabled 1 = DMA request is enabled Privilege mode (write): 0 = leaves the corresponding bit unchanged 1 = disable DMA request	RW	0x0
8	CLEARDMA0	Clear Compare DMA Request 0.	RW	0x0
7:4	RESERVED	Reserved.	R	0x0
3	CLEARINT3	Clear Compare Interrupt 3. User and privilege mode (read): 0 = interrupt is disabled 1 = interrupt is enabled Privilege mode (write): 0 = leaves the corresponding bit unchanged 1 = disable interrupt	RW	0x0
2	CLEARINT2	Clear Compare Interrupt 2. User and privilege mode (read): 0 = interrupt is disabled 1 = interrupt is enabled Privilege mode (write): 0 = leaves the corresponding bit unchanged 1 = disable interrupt	RW	0x0
1	CLEARINT1	Clear Compare Interrupt 1.	RW	0x0
0	CLEARINT0	Clear Compare Interrupt 0. User and privilege mode (read): 0 = interrupt is disabled 1 = interrupt is enabled Privilege mode (write): 0 = leaves the corresponding bit unchanged 1 = disable interrupt	RW	0x0

Table 24-52. Register Call Summary for Register RTICLEARINT

Real Time Interrupt Module

- RTI Register Summary: [0][1]

Table 24-53. RTIINTFLAG

Address Offset	0x0000 0088		
Physical Address	0x4AE3 1088	**Instance**	RTI1
	0x4AE3 3088		RTI1
	0x4AE3 3088		RTI2
	0x4AE3 5088		RTI1
	0x4AE3 5088		RTI3
	0x4AE3 7088		RTI2
	0x4AE3 7088		RTI4
	0x4AE3 9088		RTI2
	0x4AE3 9088		RTI5
Description	The corresponding flags are set at every compare match of Free Running Counterx and RTICOMPx value, regardless if the interrupt is enabled or not.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED													OVL1INT	OVL0INT	RESERVED	RESERVED												INT3	INT2	INT1	INT0

Bits	Field Name	Description	Type	Reset
31:19	RESERVED	Reserved.	R	0x0
18	OVL1INT	Free Running Counter 1 Overflow Interrupt Flag. User and privilege mode (read): determines if an interrupt is pending 0 = no interrupt pending 1 = interrupt pending Privilege mode (write): 0 = leaves the bit unchanged 1 = set the bit to 0	R	0x0
17	OVL0INT	Free Running Counter 0 Overflow Interrupt Flag. User and privilege mode (read): determines if an interrupt is pending 0 = no interrupt pending 1 = interrupt pending Privilege mode (write): 0 = leaves the bit unchanged 1 = set the bit to 0	R	0x0
16	RESERVED	Reserved.	R	0x0
15:4	RESERVED	Reserved.	R	0x0
3	INT3	Interrupt Flag 3. User and privilege mode (read): determines if a interrupt is pending 0 = no interrupt pending 1 = interrupt pending Privilege mode (write): 0 = leaves the bit unchanged 1 = set the bit to 0	R	0x0
2	INT2	Interrupt Flag 2. User and privilege mode (read): determines if a interrupt is pending 0 = no interrupt pending 1 = interrupt pending Privilege mode (write): 0 = leaves the bit unchanged 1 = set the bit to 0	R	0x0
1	INT1	Interrupt Flag 1. User and privilege mode (read): determines if a interrupt is pending 0 = no interrupt pending 1 = interrupt pending Privilege mode (write): 0 = leaves the bit unchanged 1 = set the bit to 0	R	0x0
0	INT0	Interrupt Flag 0. User and privilege mode (read): determines if a interrupt is pending 0 = no interrupt pending 1 = interrupt pending Privilege mode (write): 0 = leaves the bit unchanged 1 = set the bit to 0	R	0x0

Table 24-54. Register Call Summary for Register RTIINTFLAG

Real Time Interrupt Module

- RTI Register Summary: [0][1]

Table 24-55. RTIDWDCTRL

Address Offset	0x0000 0090		
Physical Address	0x4AE3 1090	**Instance**	RTI1
	0x4AE3 3090		RTI1
	0x4AE3 3090		RTI2
	0x4AE3 5090		RTI1
	0x4AE3 5090		RTI3
	0x4AE3 7090		RTI2
	0x4AE3 7090		RTI4
	0x4AE3 9090		RTI2
	0x4AE3 9090		RTI5
Description	This register s functionality is dependent on whether the DWD is implemented to be always enabled or not. If the DWD is always enabled, then the DWD is automatically enabled after system reset is released and cannot be disabled by software. In that case, this register is redundant and any writes to this register have no effect on the DWD functionality. If, however, the DWD is not enabled upon release of system reset, then the software has to write to the DWDCTRL field in order to enable the DWD, as described below. Once enabled, the watchdog can only be disabled by a system reset. The application cannot disable the watchdog. The DWDCTRL register also implements a one-time-write constraint. That is, once the application writes to this register to enable the watchdog, all further writes are ignored.		
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
DWDCTRL_31_16																DWDCTRL_15_0															

Bits	Field Name	Description	Type	Reset
31:16	DWDCTRL_31_16	Digital Watchdog Control. User and privilege mode (read): 5312ACEDh = DWD counter is disabled. This is the default value. A98559DAh = DWD counter is enabled Any other value = DWD counter state is unchanged (enabled or disabled) Privilege mode (write): A98559DAh = DWD counter is enabled Any other value = State of DWD counter is unchanged (stays enabled or disabled) One-Write Functionality of DWDCTRL Register: The RTIDWDCTRL register implements a one-write functionality, such that the application cannot write to this register more than once. Writing the default value will not enable the watchdog as described above. Writing the enable value will start the watchdog counters. A write to RTIDWDCTRL will only be enabled after a system reset again.	RW	0x14c0
15:0	DWDCTRL_15_0	Digital Watchdog Control. User and privilege mode (read): 5312ACEDh = DWD counter is disabled. This is the default value. A98559DAh = DWD counter is enabled Any other value = DWD counter state is unchanged (enabled or disabled) Privilege mode (write): A98559DAh = DWD counter is enabled Any other value = State of DWD counter is unchanged (stays enabled or disabled) One-Write Functionality of DWDCTRL Register: The RTIDWDCTRL register implements a one-write functionality, such that the application cannot write to this register more than once. Writing the default value will not enable the watchdog as described above. Writing the enable value will start the watchdog counters. A write to RTIDWDCTRL will only be enabled after a system reset again.	RW	0x0

Table 24-56. Register Call Summary for Register RTIDWDCTRL

Real Time Interrupt Module

- RTI Digital Watchdog: [0]
- RTI Register Summary: [1][2]
- RTI Register Description: [3][4][5][6]

Table 24-57. RTIDWDPRLD

Address Offset	0x0000 0094		
Physical Address	0x4AE3 1094	**Instance**	RTI1
	0x4AE3 3094		RTI1
	0x4AE3 3094		RTI2
	0x4AE3 5094		RTI1
	0x4AE3 5094		RTI3
	0x4AE3 7094		RTI2
	0x4AE3 7094		RTI4
	0x4AE3 9094		RTI2
	0x4AE3 9094		RTI5
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																				DWDPRLD											

Bits	Field Name	Description	Type	Reset
31:12	RESERVED		R	0x0
11:0	DWDPRLD	Digital Watchdog Preload Value. User and privilege mode (read): A read from this register in any CPU mode returns the current preload value. Privilege mode (write): If the DWD is always enabled after reset is released: The DWD starts counting down from the reset value of the counter, that is, 0x002DFFFF. The application can configure the DWD preload register any time before this down counter expires. When the application services the DWD, the preload register contents are copied left-justified into the DWD down counter and it starts counting down from that value. If the DWD is implemented such that the down counter is enabled by software: The DWD preload register can be configured only when the DWD is disabled. Therefore, the application can only configure the DWD preload register before it enables the DWD down counter. The expiration time of the DWD Down Counter can be determined with following equation: ARRAY(0x1f75d70) where: RTIDWDPRLD = 0...4095	RW	0x0

Table 24-58. Register Call Summary for Register RTIDWDPRLD

Real Time Interrupt Module

- RTI Digital Watchdog: [0][1]
- RTI Register Summary: [2][3]
- RTI Register Description: [4]

Table 24-59. RTIWDSTATUS

Address Offset	0x0000 0098		
Physical Address	0x4AE3 1098	**Instance**	RTI1
	0x4AE3 3098		RTI1
	0x4AE3 3098		RTI2
	0x4AE3 5098		RTI1
	0x4AE3 5098		RTI3
	0x4AE3 7098		RTI2
	0x4AE3 7098		RTI4
	0x4AE3 9098		RTI2
	0x4AE3 9098		RTI5
Description	The values of the following status bits will not be affected by a system reset. These bits are cleared by a power up reset, or by the application.		
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																										DWWD_ST	END_TIME_VIOL	START_TIME_VIOL	KEYST	_DWDST	AWDST

Bits	Field Name	Description	Type	Reset
31:6	RESERVED		R	0x0
5	DWWD_ST	Windowed Watchdog Status. This bit denotes whether the time-window defined by the windowed watchdog configuration has been violated, or if a wrong key or key sequence was written to service the watchdog. User and privilege mode (read): 0 = no time-window violation has occurred. 1 = a time-window violation has occurred. The watchdog will generate either a system reset or a non-maskable interrupt to the CPU in this case. Privilege mode (write): 0 = leaves the current value unchanged. 1 = clears the bit to 0. This will also clear all other status flags in the RTIWDSTATUS register except for the AWD ST flag. Clearing of the status flags will deassert the non-maskable interrupt generated due to violation of the DWWD.	R	0x0
4	END_TIME_VIOL	Windowed Watchdog End Time Violation Status. This bit denotes whether the end-time defined by the windowed watchdog configuration has been violated. This bit is effectively a copy of the DWD ST status flag. User and priviledge mode (read): 0 = no end-time window violation has occurred. 1 = the end-time defined by the windowed watchdog configuration has been violated. Priviledge mode (write): 0 = leaves the current value unchanged. 1 = clears the bit to 0.	R	0x0
3	START_TIME_VIOL	Windowed Watchdog End Time Violation Status. This bit denotes whether the start-time defined by the windowed watchdog configuration has been violated. This indicates that the WWD was serviced before the service window was opened. User and priviledge mode (read): 0 = no start-time window violation has occurred. 1 = the start-time defined by the windowed watchdog configuration has been violated. Priviledge mode (write): 0 = leaves the current value unchanged. 1 = clears the bit to 0.	R	0x0
2	KEYST	Watchdog KeyStatus.	R	0x0
1	_DWDST	Digital Watchdog Status.	R	0x0
0	AWDST	Analog Watchdog Status.	R	0x0

Table 24-60. Register Call Summary for Register RTIWDSTATUS

Real Time Interrupt Module
- RTI Register Summary: [0][1]
- RTI Register Description: [2]

Table 24-61. RTIWDKEY

Address Offset	0x0000 009C		
Physical Address	0x4AE3 109C	**Instance**	RTI1
	0x4AE3 309C		RTI1
	0x4AE3 309C		RTI2
	0x4AE3 509C		RTI1
	0x4AE3 509C		RTI3
	0x4AE3 709C		RTI2
	0x4AE3 709C		RTI4
	0x4AE3 909C		RTI2
	0x4AE3 909C		RTI5
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																WDKEY															

Bits	Field Name	Description	Type	Reset
31:16	RESERVED		R	0x0
15:0	WDKEY	Watchdog Key. User and privilege mode reads are indeterminate. Privilege mode (write): A write of 0xE51A followed by 0xA35C in two separate write operations defines the Key Sequence and discharges the watchdog capacitor. This also causes the upper 12 bits of the DWD down counter to be reloaded with the contents of the DWD preload register and the lower 13 bits to become all 1 s. Writing any other value causes a digital watchdog reset. Register write access time precaution: The user has to take into account that the write to the register takes 3 OCP_CLK_PI cycle. This needs to be considered for the AWD/DWD expiration calculation.	RW	0xa35c

Table 24-62. Register Call Summary for Register RTIWDKEY

Real Time Interrupt Module

- RTI Digital Watchdog: [0][1][2]
- RTI Digital Windowed Watchdog: [3]
- RTI Register Summary: [4][5]

Table 24-63. RTIDWDCNTR

Address Offset	0x0000 00A0		
Physical Address	0x4AE3 10A0	**Instance**	RTI1
	0x4AE3 30A0		RTI1
	0x4AE3 30A0		RTI2
	0x4AE3 50A0		RTI1
	0x4AE3 50A0		RTI3
	0x4AE3 70A0		RTI2
	0x4AE3 70A0		RTI4
	0x4AE3 90A0		RTI2
	0x4AE3 90A0		RTI5
Description			
Type	R		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED								DWDCNTR_24_16								DWDCNTR_15_0															

Bits	Field Name	Description	Type	Reset
31:25	RESERVED		R	0x0
24:16	DWDCNTR_24_16	Digital Watchdog Down Counter. The value of the DWDCNTR after a system reset is 0x002D_FFFF. When the DWD is enabled and the DWD counter starts counting down from this value with an RTI_CLK_PI time base of 3MHz, a watchdog reset will be generated in 1 second. User and privilege mode (read): Reads return the current counter value. Privilege mode (write): Writes don t have an effect.	R	0x2d
15:0	DWDCNTR_15_0	Digital Watchdog Down Counter. The value of the DWDCNTR after a system reset is 0x002D_FFFF. When the DWD is enabled and the DWD counter starts counting down from this value with an RTI_CLK_PI time base of 3MHz, a watchdog reset will be generated in 1 second. User and privilege mode (read): Reads return the current counter value. Privilege mode (write): Writes don t have an effect.	R	0xffff

Table 24-64. Register Call Summary for Register RTIDWDCNTR

Real Time Interrupt Module

- Real Time Interrupt Module:
- RTI Digital Watchdog: [1]
- RTI Register Summary: [2][3]

Table 24-65. RTIWWDRXNCTRL

Address Offset	0x0000 00A4		
Physical Address	0x4AE3 10A4	**Instance**	RTI1
	0x4AE3 30A4		RTI1
	0x4AE3 30A4		RTI2
	0x4AE3 50A4		RTI1
	0x4AE3 50A4		RTI3
	0x4AE3 70A4		RTI2
	0x4AE3 70A4		RTI4
	0x4AE3 90A4		RTI2
	0x4AE3 90A4		RTI5
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																												WWDRXN			

Bits	Field Name	Description	Type	Reset
31:4	RESERVED		R	0x0
3:0	WWDRXN	Digital Windowed Watchdog Reaction. User and privilege mode (read), privileged mode (write): 0x5 = This is the default value. The windowed watchdog will cause a reset if the watchdog is serviced outside the time window defined by the configuration, or if the watchdog is not serviced at all. 0xA = The windowed watchdog will generate a non-maskable interrupt to the CPU if the watchdog is serviced outside the time window defined by the configuration, or if the watchdog is not serviced at all. Writing any other value will cause a system reset if the watchdog is serviced outside the time window defined by the configuration, or if the watchdog is not serviced at all. Configuration of DWWD Reaction: The DWWD reaction can be selected by the application even when the DWWD counter is already enabled. If a change to the WWDRXN is made before the watchdog service window is opened, then the change in the configuration takes effect immediately. If a change to the WWDRXN is made when the watchdog service window is already open, then the change in configuration takes effect only after the watchdog is serviced.	RW	0x5

Table 24-66. Register Call Summary for Register RTIWWDRXNCTRL

Real Time Interrupt Module

- RTI Digital Windowed Watchdog: [0]
- RTI Register Summary: [1][2]

Table 24-67. RTIWWDSIZECTRL

Address Offset	0x0000 00A8		
Physical Address	0x4AE3 10A8	**Instance**	RTI1
	0x4AE3 30A8		RTI1
	0x4AE3 30A8		RTI2
	0x4AE3 50A8		RTI1
	0x4AE3 50A8		RTI3
	0x4AE3 70A8		RTI2
	0x4AE3 70A8		RTI4
	0x4AE3 90A8		RTI2
	0x4AE3 90A8		RTI5
Description			
Type	RW		

15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
WWDSIZE															

Bits	Field Name	Description	Type	Reset
15:0	WWDSIZE	Digital Windowed Watchdog Window Size. User and privilege mode (read), privileged mode (write): Value written to WWDSIZE: 0x00000005 = 100% Window Size (The functionality is the same as the standard time-out digital watchdog.) Value written to WWDSIZE: 0x00000050 = 50% Window Size Value written to WWDSIZE: 0x00000500 = 25% Window Size Value written to WWDSIZE: 0x00005000 = 12.5% Window Size Value written to WWDSIZE: 0x00050000 = 6.25% Window Size Value written to WWDSIZE: 0x00500000 = 3.125% Window Size Value written to WWDSIZE: Any other value = 3.125% Window Size Incorrect value being written to watchdog window size control register: If an incorerct value is written to the WWDSIZE field, or if a system disturbance causes the WWDSIZE field to have a value other than 0x5, 0x50, 0x500, 0x5000, 0x50000, or 0x500000, then the window size will be configured to be 3.125%. This increases the chances of getting a reset due to the windowed watchdog, which enables the system to handle the cause for the incorrect configuration. Configuration of DWWD Window Size: The DWWD window size can be selected by the application even when the DWWD counter is already enabled. If a change to the WWDSIZE is made before the watchdog service window is opened, then the change in the configuration takes effect immediately. If a change to the WWDSIZE is made when the watchdog service window is already open, then the change in configuration takes effect only after the watchdog is serviced.	RW	0x5

Table 24-68. Register Call Summary for Register RTIWWDSIZECTRL

Real Time Interrupt Module

- RTI Register Summary: [0][1]

Table 24-69. RTIINTCLRENABLE

Address Offset	0x0000 00AC		
Physical Address	0x4AE3 10AC	**Instance**	RTI1
	0x4AE3 30AC		RTI1
	0x4AE3 30AC		RTI2
	0x4AE3 50AC		RTI1
	0x4AE3 50AC		RTI3
	0x4AE3 70AC		RTI2
	0x4AE3 70AC		RTI4
	0x4AE3 90AC		RTI2
	0x4AE3 90AC		RTI5
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED				INTCLRENABLE3				RESERVED				INTCLRENABLE2				RESERVED				INTCLRENABLE1				RESERVED				INTCLRENABLE0			

Bits	Field Name	Description	Type	Reset
31:28	RESERVED		R	0x0
27:24	INTCLRENABLE3	Enables the auto-clear functionality on the compare 3 interrupt. User and Privileged mode (read): 0x5 = Auto-clear for compare 3 interrupt is disabled. Any other value = Auto-clear for compare 3 interrupt is enabled. Privileged mode (write): 0x5 = Disables the auto-clear functionality on the compare 3 interrupt. Any other value = Enables the auto-clear functionality on the compare 3 interrupt.	RW	0x5
23:20	RESERVED		R	0x0
19:16	INTCLRENABLE2	Enables the auto-clear functionality on the compare 2 interrupt. User and Privileged mode (read): 0x5 = Auto-clear for compare 2 interrupt is disabled. Any other value = Auto-clear for compare 2 interrupt is enabled. Privileged mode (write): 0x5 = Disables the auto-clear functionality on the compare 2 interrupt. Any other value = Enables the auto-clear functionality on the compare 2 interrupt.	RW	0x5
15:12	RESERVED		R	0x0
11:8	INTCLRENABLE1	Enables the auto-clear functionality on the compare 1 interrupt. User and Privileged mode (read): 0x5 = Auto-clear for compare 1 interrupt is disabled. Any other value = Auto-clear for compare 1 interrupt is enabled. Privileged mode (write): 0x5 = Disables the auto-clear functionality on the compare 1 interrupt. Any other value = Enables the auto-clear functionality on the compare 1 interrupt.	RW	0x5
7:4	RESERVED		R	0x0
3:0	INTCLRENABLE0	Enables the auto-clear functionality on the compare 0 interrupt. User and Privileged mode (read): 0x5 = Auto-clear for compare 0 interrupt is disabled. Any other value = Auto-clear for compare 0 interrupt is enabled. Privileged mode (write): 0x5 = Disables the auto-clear functionality on the compare 0 interrupt. Any other value = Enables the auto-clear functionality on the compare 0 interrupt. Hook-up of Compare Interrupt to a device pin: The RTI module generates up to 4 compare interrupts. The connection between one or more of these compare interrupt(s) to a device pin is completely device-dependent. Refer to the device datasheet to identify the actual pin(s) that connects to the compare interrupt(s).	RW	0x5

Table 24-70. Register Call Summary for Register RTIINTCLRENABLE

Real Time Interrupt Module

- RTI Register Summary: [0][1]

Table 24-71. RTICOMP0CLR

Address Offset	0x0000 00B0		
Physical Address	0x4AE3 10B0	**Instance**	RTI1
	0x4AE3 30B0		RTI1
	0x4AE3 30B0		RTI2
	0x4AE3 50B0		RTI1
	0x4AE3 50B0		RTI3
	0x4AE3 70B0		RTI2
	0x4AE3 70B0		RTI4
	0x4AE3 90B0		RTI2
	0x4AE3 90B0		RTI5
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
COMP0CLR																															

Bits	Field Name	Description	Type	Reset
31:0	COMP0CLR	Compare 0 Clear. User and privilege mode (read): current compare value Privilege mode (write): update of the compare register with a new compare value Reset behavior: A reset does not generate a compare match, since the compare logic will only be active, when the associated counter block is enabled.	RW	0x0

Table 24-72. Register Call Summary for Register RTICOMP0CLR

Real Time Interrupt Module

- RTI Counter Operation: [0]
- RTI Register Summary: [1][2]

Table 24-73. RTICOMP1CLR

Address Offset	0x0000 00B4		
Physical Address	0x4AE3 10B4	**Instance**	RTI1
	0x4AE3 30B4		RTI1
	0x4AE3 30B4		RTI2
	0x4AE3 50B4		RTI1
	0x4AE3 50B4		RTI3
	0x4AE3 70B4		RTI2
	0x4AE3 70B4		RTI4
	0x4AE3 90B4		RTI2
	0x4AE3 90B4		RTI5
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
COMP1CLR																															

Bits	Field Name	Description	Type	Reset
31:0	COMP1CLR	Compare 1 Clear. User and privilege mode (read): current compare value Privilege mode (write): update of the compare register with a new compare value Reset behavior: A reset does not generate a compare match, since the compare logic will only be active, when the associated counter block is enabled.	RW	0x0

Table 24-74. Register Call Summary for Register RTICOMP1CLR

Real Time Interrupt Module

- RTI Counter Operation: [0]
- RTI Register Summary: [1][2]

Table 24-75. RTICOMP2CLR

Address Offset	0x0000 00B8		
Physical Address	0x4AE3 10B8	**Instance**	RTI1
	0x4AE3 30B8		RTI1
	0x4AE3 30B8		RTI2
	0x4AE3 50B8		RTI1
	0x4AE3 50B8		RTI3
	0x4AE3 70B8		RTI2
	0x4AE3 70B8		RTI4
	0x4AE3 90B8		RTI2
	0x4AE3 90B8		RTI5
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
COMP2CLR																															

Bits	Field Name	Description	Type	Reset
31:0	COMP2CLR	Compare 2 Clear. User and privilege mode (read): current compare value Privilege mode (write): update of the compare register with a new compare value Reset behavior: A reset does not generate a compare match, since the compare logic will only be active, when the associated counter block is enabled.	RW	0x0

Table 24-76. Register Call Summary for Register RTICOMP2CLR

Real Time Interrupt Module

- RTI Counter Operation: [0]
- RTI Register Summary: [1][2]

Table 24-77. RTICOMP3CLR

Address Offset	0x0000 00BC		
Physical Address	0x4AE3 10BC	**Instance**	RTI1
	0x4AE3 30BC		RTI1
	0x4AE3 30BC		RTI2
	0x4AE3 50BC		RTI1
	0x4AE3 50BC		RTI3
	0x4AE3 70BC		RTI2
	0x4AE3 70BC		RTI4
	0x4AE3 90BC		RTI2
	0x4AE3 90BC		RTI5
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
COMP3CLR																															

Bits	Field Name	Description	Type	Reset
31:0	COMP3CLR	Compare 3 Clear. User and privilege mode (read): current compare value Privilege mode (write): update of the compare register with a new compare value Reset behavior: A reset does not generate a compare match, since the compare logic will only be active, when the associated counter block is enabled.	RW	0x0

Table 24-78. Register Call Summary for Register RTICOMP3CLR

Real Time Interrupt Module • RTI Counter Operation: [0] • RTI Register Summary: [1][2]

Initialization

This chapter introduces the steps in the general-purpose (GP) device initialization.

Topic **Page**

25.1 Initialization Overview

This chapter provides an overview of the requirements to initialize the device from power on to operating system (OS) and application execution, the overall initialization process (including hardware- and software-related steps), the general ROM code operational requirements, and behavior expectations.

25.1.1 Terminology

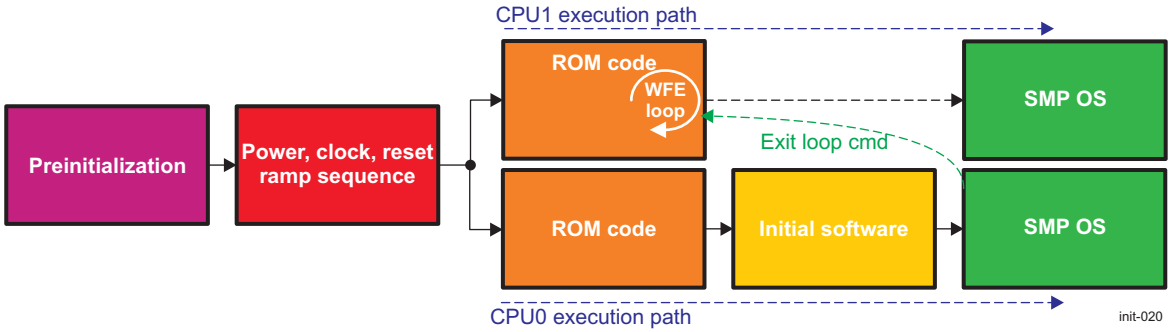
- **Bootstrap:** Initial software launched by the ROM code during the memory booting phase
- **Configuration Header (CH):** Optional structure that precedes the initial software and allows the redefinition of the ROM code default settings
- **Downloaded software:** Initial software downloaded into the internal static RAM (SRAM) by the ROM code during the peripheral booting phase
- **eFuse:** A one-time programmable memory location usually set at the factory
- **Flash loader:** Downloaded software launched by the ROM code during the preflashing stage. It also programs an image in external memories.
- **GP device:** General-purpose device.
- **Initial software:** Software executed by any of the ROM code mechanisms (memory booting or peripheral booting). Initial software is a generic term for bootstrap and downloaded software.
- **Memory booting:** ROM code mechanism that consists of executing initial software from external memory
- **Master CPU:** The Arm® Cortex®-M4 CPU for which CPU-ID is 0. It configures the multicore platform and starts the ROM code to ensure device booting from a mass storage memory (memory booting) or a peripheral interface (peripheral booting).
- **Peripheral booting:** ROM code mechanism that consists of polling selected interfaces, downloading, and executing initial software (in this case, downloaded software) in the internal RAM
- **Permanent booting device:** Memory device containing, by default, the image to be executed during the booting sequence. It is the default memory booting device. The permanent booting device is used after a warm reset.
- **Preflashing:** A specific case of peripheral booting where the ROM code mechanism is used to program the external flash memory
- **ROM Code:** The on-chip software in device ROM that implements booting
- **ROM Code-controlled Boot Phase:** This phase covers the sequence operations from the time the platform releases the reset to the time first user- or customer-owned software starts execution. This phase is fully controlled by the device ROM code.
- **Slave CPU:** The Arm Cortex-M4 MPCore CPU for which CPU-ID is 1. It is brought to the wait-for-event (WFE) state by the ROM code, waiting to be woken up by the master CPU.

25.1.2 Initialization Process

Figure 25-1 is an overview of the initialization process and its steps:

- **Preinitialization:** Power, clock, and control connections must be present, and the boot configuration pins must be held at the desired logical levels.
- **Power, clock, reset ramp sequence:** Specific sequence that is applied by the power-management chip
- **ROM code:** Responsible for finding, for downloading, and for executing the initial software by using the master CPU
- **Initial software:** Software that prepares and passes control to application software or to the high-level operating system (HLOS)
- **Symmetric multiprocessing (SMP)-capable HLOS** or application (primarily for diagnostics)

Figure 25-1. Initialization Process



The first two steps in the initialization process are hardware-oriented; however, they require an understanding of the process of configuring these system interface pins (balls on the device), which have software-configurable functionality. This configuration is an essential part of the chip configuration and is application-dependent. This chapter discusses these system-interface pins and the associated configuration registers that are vital to the correct initialization of the device.

25.2 Preinitialization

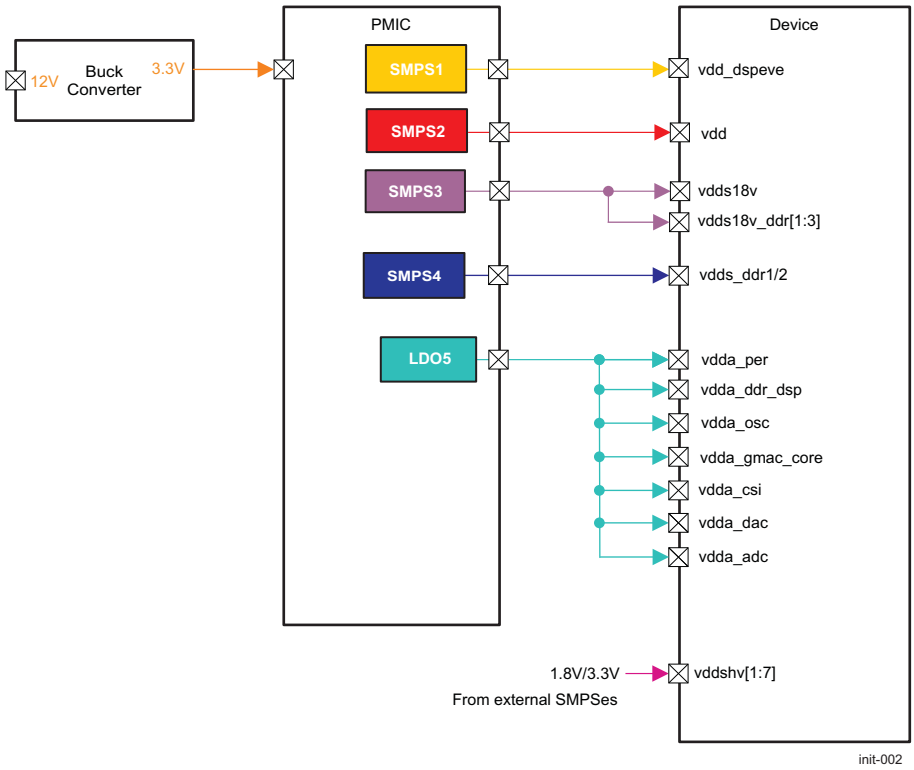
To accomplish a successful boot-up operation, certain hardware configuration settings must be in place. Clock, reset, and power connections, as well as pins involved in setting the boot memory space for the master CPU, must be connected and driven correctly to successfully initialize the device. The following sections describe the specific requirements for the preinitialization stage.

25.2.1 Power Requirements

The device can be supplied by an external power-management integrated circuit (PMIC). TI provides a global solution with the device connected to the power-management IC companion chips. Refer to Data Manual for information about the TI's power-management IC companion chips supported for this device.

Figure 25-2 shows typical power connections between the device and a PMIC companion chip.

Figure 25-2. Power Supply Connections Example



NOTE: Figure 25-2 is an example of power connections between the device and the PMIC, representing only one of the multiple PMIC/OTP options.
These connections depend on the actual application, PMIC, and OTP used. Refer to the Device Data Manual, respective PMIC Data Sheet, and all related application notes before starting a new design.

Table 25-1 describes the device power balls.

Table 25-1. Device Power Balls

Voltage Ball Name	Subsystems and Peripherals
vdd	Subsystems and modules supplied by CORE voltage domain
vdd_dspeve	DSP and EVE voltage domain
vdds18v	EMIF 1.8v vref. GENERAL, GPMC, MMC4, RGMII, VIN1, VOUT1 Power Groups. I/O supply for iFORCE/vSENSE. DSP LDO. SLDO
vdds_ddr1	EMIF I/O power supply

Table 25-1. Device Power Balls (continued)

Voltage Ball Name	Subsystems and Peripherals
vdds_ddr2	EMIF I/O power supply
vdds_ddr3	EMIF I/O power supply
vdds18v_ddr1	EMIF 1.8v bias supply
vdds18v_ddr2	EMIF 1.8v bias supply
vdds18v_ddr3	EMIF 1.8v bias supply
ddr1_ext_vref0	EMIF channel 1 vref
ddr1_ext_vref1	EMIF channel 2 vref
vdda_ddr_dsp	Analog power supply for DPLL_DDR and DDR HSDIVIDER , DPLL_EVE_VID_DSP
vdda_per	Analog power supply for DPLL_PER, PER HSDIVIDER
vdda_gmac_core	Analog power supply for DPLL_GMAC_DSP and HSDIVIDER, DPLL_CORE and HSDIVIDER
vdda_spi	I/O Supply for NAND Flash on PoP memory (QSPI)
vddshv1	Dual-voltage GENERAL group I/Os
vddshv2	Dual-voltage GPMC group I/Os
vddshv3	Dual-voltage MMC4 group I/Os
vddshv4	Dual-voltage RGMII group I/Os
vddshv5	Dual-voltage VIN1 group I/Os
vddshv6	Dual-voltage VOUT group I/Os
vddshv7	Dual-voltage WIFI group I/Os
vdda_csi	CSI analog power supply
vdda_dac	DAC analog power supply
vdda_adc	ADC analog power supply
vdda_osc	I/O supply for oscillator section

NOTE: For a complete description of the power balls on the device package, see the *Device Data Manual*.

For more information about power management, see Section 3.6 *Clock Management Functional Description*, in Chapter 3, *Power, Reset, and Clock Management*.

25.2.2 Interaction With the PMIC Companion

The ROM code does not perform any I^2C/SPI transactions with the PMIC. This makes ROM code PMIC-independent. The following system conditions must be met to perform device initialization:

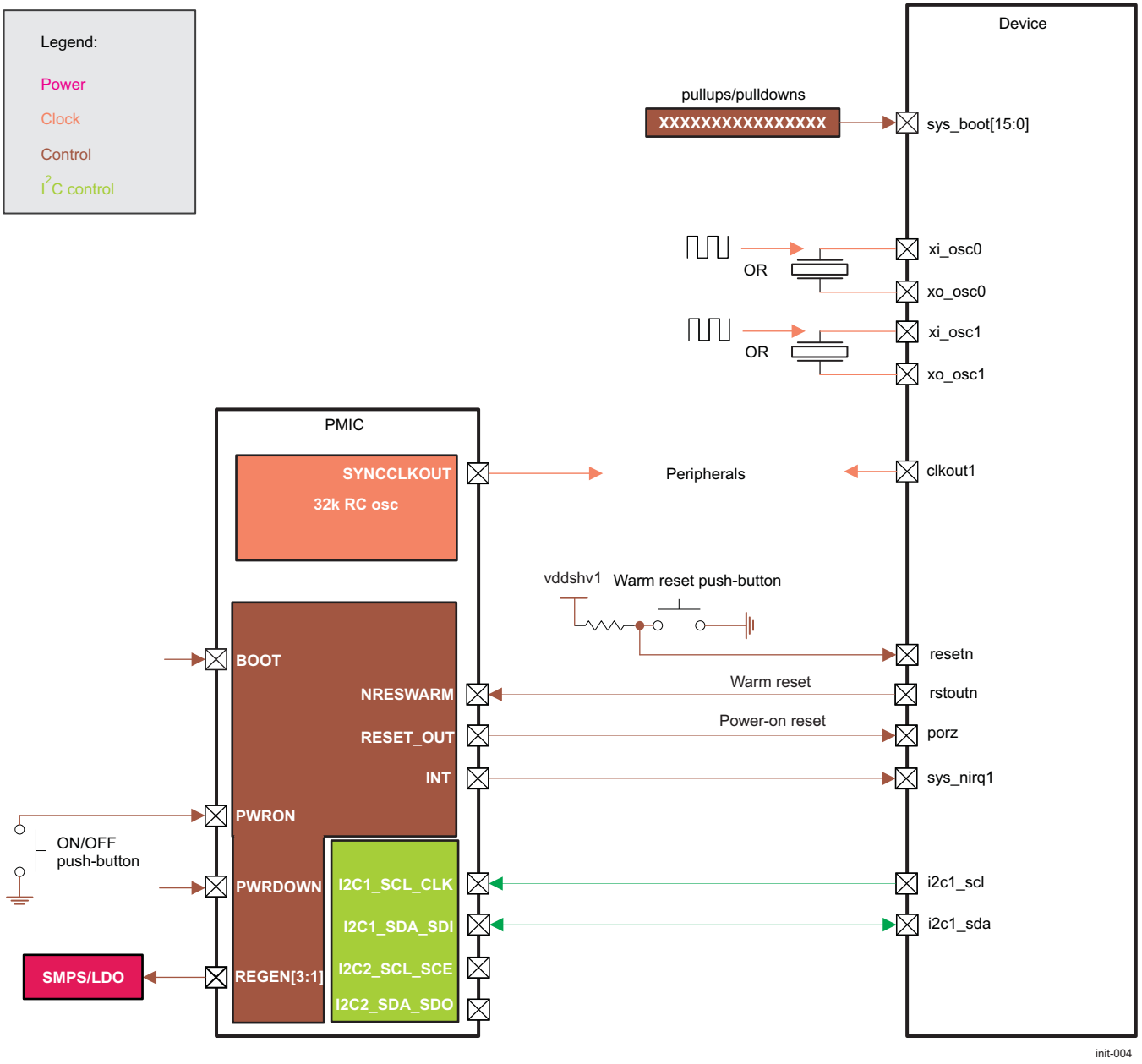
- Devices must be appropriately powered and be up and ready at platform startup:
  - QSPI
  - GPMC

25.2.3 Clock, Reset, and Control

25.2.3.1 Overview

Figure 25-3 shows the clock and reset environment where clocks and reset-related signals are gathered at the system level, the system-expansion signals, and the crystal oscillator connection.

Figure 25-3. Clock, Reset, and Control Environment Overview



NOTE: Figure 25-3 is a typical example of clock, reset and control connections between the device and a PMIC. Refer to the Device Data Manual for the supported PMIC(s) for your device and for more information on these balls.
For PMIC ball description, see the respective PMIC Data Sheet.

The main features of the system interface are:

- Accepts crystals connected between device xi_osc0 and xo_osc0 pads, and xi_osc1 and xo_osc1 pads to generate SYS_CLK1 and SYS_CLK2, resepectively
- Accepts external LVCMOS clock sources connected to xi_osc0, and xi_osc1 pads for SYS_CLK1 and SYS_CLK2, respectively
- Two reference clock inputs
- Three configurable clock outputs

- sysboot[15:0] input signals to define the boot mode, system clock speed, and GPMC in XIP mode
- Two reset sources
  - Power-on reset (porz)
  - Warm reset (resetn)
- Three external interrupt lines (sys_nirq1, sys_nirq2, and nmin)
- Four external DMA requests

25.2.3.2 Clocking Scheme

The device operation requires external input clocks, as follows:

- System clocks: The device supports two system clocks with two clock sources each:
  - SYS_CLK1 (main system clock):
    - Crystal on OSC0 pins. An internal oscillator (OSC0) embedded in the device is used.
    - External LVCMOS clock on xi_osc0
  - SYS_CLK2 (optional system clock):
    - Crystal on OSC1 pins. An internal oscillator (OSC1) embedded in the device is used.
    - External LVCMOS clock on xi_osc1
- The device can deliver digital clocks to peripherals ICs.

The device provides wide choice of clocks that can be delivered on clkout[1:3] pads to companion devices. For more information, see Section 3.3, *PRCM Subsystem Environment*, in Chapter 3, *Power, Reset, and Clock Management*. For more information on pad multiplexing, see Chapter 13, *Control Module*.

Table 25-2 lists the mapping for the device clock input sources. lists the PMIC clock requirements.

Table 25-2. Mapping for Input Sources

Clock	Clock Source	Ball Mapping	Frequency Range/List	Type
SYS_CLK1	Internal oscillator 0 (OSC0)	xi_osc0 and xo_osc0	19.2, 20, and 27 MHz	Crystal connection pins
	External	xi_osc0	19.2, 20, and 27 MHz	External LVCMOS
SYS_CLK2	Internal oscillator 1 (OSC1)	xi_osc1 and xo_osc1	19.2 ÷ 32 MHz	Crystal connection pins
	External	xi_osc1	12 ÷ 38.4 MHz	External LVCMOS

25.2.3.3 Reset Configuration

25.2.3.3.1 ON/OFF Interconnect and Power-On-Reset

The entire system is typically awakened by an ON/OFF push button connected to the PMIC chip. This signal belongs to the VSYS - system power domain and is active low (the PMIC internal pullup ties it to VSYS). The PMIC power-up event is propagated through its NRESPWRON output pin to the device porz pad (that is, the PRM SYS_PWRON_RST signal) when the PMIC power-up sequence is achieved. The device porz input pin is held low all the time during VDD core and I/O power-up.

25.2.3.3.2 Warm Reset

A warm reset can be asserted by the device, by an external button (typically for development platform), or by any other chip connected to it (normally tied to PMIC companion NRESWARM output pin).

The device warm reset pad (resetn - device signal SYS_WARM_IN_RST) is used to trigger a warm reset on the device, which resets part of the device when it has already booted (for example, to recover from a software crash). When an internal device reset occurs, SYS_NRES_WARM_OUT output and device pad rstoutn go low and reset all the peripherals.

The device releases the SYS_NRES_WARM_OUT output signal after SYS_PWRON_RST is deasserted.

25.2.3.3.3 Peripheral Reset by GPIO

Most peripherals can be reset and powered on or off by GPIO. By default, under POR, most device signals are in safe mode with a default value driven by the I/O cell. The value is driven by an internal pullup or pulldown. Depending on the peripheral reset active level, users must select one GPIO or another (according to the reset value).

Once POR is released, the value on the pad is driven by the default configuration of the device control module. Most of time, this configuration is aligned with the default value selected on the I/O cell.

The next step is application-dependent: Users must configure the device registers to validate GPIO use and the default configuration of the control module.

25.2.3.3.4 Warm Reset Impact on GPIOs

When a warm reset event occurs:

- The GPIO controller is reset. Consequently, the GPIO is automatically turned in input mode.
- The control module is not reset. Information related to signal multiplexing mode and pullup or pulldown configuration is still valid.

Therefore, when a warm reset event occurs, the output buffer is disabled. Consequently, two different behaviors can be defined with regard to what is expected by the platform:

- GPIO sensitive to warm reset:

  To prevent a floating pad, user software is designed to have the internal pad PU and PD resistors enabled immediately, before the software warm reset action, because the warm reset-sensitive GPIO controllers will change I/O direction to input after an device warm reset. This is necessary if the warm reset-sensitive GPIO controller pin has been configured for output before the warm reset occurrence. The pulls-enabled-before-warm-reset condition should be set by default in case the user has configured a GPIO as an input, because in this case the user is expected to have enabled the internal PU and PD pads during GPIO configuration (unless external pull resistors were used).

  NOTE: If the PU and PD resistors are enabled immediately by software after a POR (cold reset) for a GPIO that is planned to be used only as an output, then unnecessary consumption can occur.

  While the dynamically-enable-the-pull-just-before-warm-reset condition is possible during a software warm reset (because the user software is aware of the exact moment a warm reset event occurs), it is not possible when the warm reset is triggered by hardware (for example, a watchdog reset, SYS_NRESWARM signal assertion, and so forth), because the software is not aware of the exact moment of these warm reset assertions.
- GPIO not sensitive to warm reset:

  To avoid getting a floating signal during (and after) a warm reset event and to keep the same value that was driven before the reset, users must align the pull value with the drive value each time a dedicated GPIO register is accessed.

  NOTE: To avoid unnecessary consumption, the user software must ensure that internal pull resistor is disabled when the GPIO buffer is driving.

For the description of the reset sequences and information about the device reset management, see Section 3.5, *Reset Management Functional Description*, in Chapter 3, *Power, Reset and Clock Management*. For more information about the device reset sequences, see the device *Data Manual*.

25.2.3.4 Power-Management IC Companion Control

The device and PMIC companion implement the basic power-management interface, as follows:

- Two system resets: power-on (cold) reset and warm reset
- I^2C control
- One interrupt

- I²C:

  The device interfaces: system interface I/Os, and I2C1 are involved in system interactions between the device and external power, reset and clock management IC companions.
- INT:

  PMICs can activate their output interrupt request signal (INT) at any time when requires the device (host) device to monitor their activity. When receiving such an interruption, the device checks, through the I²C, to determine the source of the interrupt. These INT pins are active low.

25.2.3.5 PMIC Request Signals

The PMIC drives three external enable-output signals, which allows switching on some external resources at different stages of the power-up sequence:

- REGEN[1:3] belong to the VSYS power domain. REGEN[1:3] can typically be used for buck boost control and power switches.

The PMIC companion chip can receive three power resource requests: ENABLE[1:2] and NSLEEP. These pins allow an external device to request PMIC internal resources. The PMIC companion power behavior when pins are activated must be programmed after the first boot (resources and ENABLE pins allocated by group, resource behavior upon group activation, and so forth).

The use of NSLEEP is especially required when PMIC chip is in sleep mode, because it cannot handle an I²C command. Any device that requires the PMIC companion resource to wake up must first activate its associated ENABLE signal. All power-management chip power regulators are off or in sleep mode when it is in sleep mode.

25.2.4 Sysboot Configuration

The device implements 16 sampled-on-reset sysboot pads.

Table 25-3. Sysboot Pads Description[(1)]

Pad	Description
sysboot[15]	Must be pulled high to 1 for proper device operation.
sysboot[14]	Selects between SYS_CLK1 and SYS_CLK2 as reference clock to DPLL_CORE.
sysboot[13:10]	Used to configure the GPMC interface when booting from XIP memory connected to GPMC. See Section 25.2.4.1, *GPMC Configuration for XIP*.
sysboot[9:8]	Selects the SYS_CLK1 clock speed. sysboot[9:8] must be set correctly on the PCB according to the speed of the connected crystal. See Section 25.2.4.2, *System Clock Speed Selection*.
sysboot[7]	Device package selection. See Section 25.2.4.3, *Miscellaneous Sysboot Settings*.
sysboot[6]	Reserved. Pull down to 0.
sysboot[5]	ADC clock divider. See Section 25.2.4.3, *Miscellaneous Sysboot Settings*.
sysboot[4:0]	Select interfaces or devices for the booting list. See Section 25.2.4.4, *Booting Device Order Selection*.

(1) Boards should be implemented with a mechanism to easily modify the pull-up/down state to enable any future modifications.

All sysboot pads are sampled and latched onto the CTRL_CORE_BOOTSTRAP register (in control module) after POR. After booting, these pads can be used for other functions such as GPIOs, and the associated register bit field is not updated by the new functionality. For more information about pad multiplexing configuration, see Section 13.4.6.1, *Pad Configuration Registers*, in Chapter 13, *Control Module*.

NOTE: If used as GPIOs, the sysboot[15:0] pads must be used only in output mode to ensure that the input values always match a certain hardware predefined boot pattern, interpreted after each POR.

25.2.4.1 GPMC Configuration for XIP

Table 25-4 describes the GPMC interface configuration used in XIP and fast-XIP modes controlled by sysboot[13:10]. These pin values are routed to the GPMC controller by hardware.

Table 25-4. GPMC for XIP Configuration

sysboot[13]	Bus Width
0	8-bit
1	16-bit
sysboot[12:11]	**A/D-muxed/non-muxed Device on CS0**
0b00	Non-muxed device
0b01	A/D-muxed device
sysboot[10]	**Wait-pin Monitoring for Read Accesses**
0	Disabled
1	Enabled

25.2.4.2 System Clock Speed Selection

There are three crystal speeds available to be selected by means of sysboot[9:8], as described in Table 25-5. User is responsible to set the correct value depending on the actual clock supplied to the device.

Table 25-5. System Clock (SYS_CLK1) Speed Selection

sysboot[9:8]	SYS_CLK1 Speed
0b00	Reserved
0b01	20 MHz
0b10	27 MHz
0b11	19.2 MHz

25.2.4.3 Miscellaneous Sysboot Settings

Additional sysboot settings are described in Table 25-6.

Table 25-6. Miscellaneous Sysboot Settings

sysboot[14]	DPLL_CORE Reference Clock(1)
0	SYS_CLK1
1	SYS_CLK2(2)
sysboot[7]	**Device Package**
0	15×15
1	12×12
sysboot[6]	**Reserved**
sysboot[5]	**ADC Clock Divider**
0	Divide by 2 (assuming SYS_CLK1 = 27 MHz)
1	No division (assuming SYS_CLK1 up to 20 MHz)

(1) There is also a software control provided. The device can boot with DPLL_CORE REF_CLK = SYS_CLK1 and later switch to SYS_CLK2. For more information, refer to Section 3.6.3.3.1, *Programming Guide For Control Module*.

(2) If SYS_CLK2 is selected, set sysboot[9:8] to 0b10 (27 MHz) to avoid DPLL_CORE generating clocks faster than allowed during ROM boot time.

25.2.4.4 Booting Device Order Selection

The ROM code creates the device list (order) based on information from the sysboot[4:0] external configuration pins. They are sensed in the device CTRL_CORE_BOOTSTRAP register during POR. The sysboot[4:0] configuration pads have two main purposes: configure ROM code software in terms of interfaces and devices used for booting and configuring hardware after a POR or cold reset.

The SYSBOOT pins are used to index a booting device list from a table with possible booting scenarios. The order of examined booting devices is from the first to the third devices.

The following names are used in the tables:

- Memory types:
  - Execute-in-place (XIP): NOR (CFI) flash memory or other XIP device. See Section 25.8.3, *XIP Memory*
  - QSPI_1: 1-bit SPI flash memories. See Section 25.8.4, *SPI/QSPI Flash Devices*
  - QSPI_4: 4-bit (Quad) SPI flash memories. See Section 25.8.4, *SPI/QSPI Flash Devices*
- Peripheral interfaces:
  - UART: UART interface. See Section 25.7, *Peripheral Booting*
- Table 25-7 lists the permanent booting devices in bold typeface.

NOTE: After a warm reset, the ROM code builds a device list featuring only the permanent booting devices (in **bold**)

Table 25-7 lists the booting device order selected by ROM code depending on sysboot[4:0] pins.

Table 25-7. Booting Devices Order

sysboot[4]	sysboot[3:0]	First Device	Second Device	Third Device
Peripheral Preferred Booting				
0b0	0b0000	UART	**QSPI_1**	
0b0	0b0001	UART	**QSPI_4**	
0b0	0b0010	UART	**XIP**	
0b0	0b0011	UART	**QSPI_4(XIP)**	
Automotive Recovery/Upgrade or Development Booting (1)				
0b0	0b1000	**UART**		
0b0	0b1001	**UART**		
0b0	0b1010	**UART**		
0b0	0b1011	**UART**		
Memory Preferred Booting				
0b1	0b0000	**QSPI_1**	UART	
0b1	0b0001	**QSPI_4**	UART	
0b1	0b0010	**XIP**	UART	
0b1	0b0011	**QSPI_4(XIP)**	UART	
Automotive Production Booting(1)				
0b1	0b1000	**QSPI_1**		
0b1	0b1001	**QSPI_4**		
0b1	0b1010	**XIP**		
0b1	0b1011	**QSPI_4(XIP)**		

(1) After 10 failed loops through the device list, ROM code initiates immediate global warm reset.

25.2.4.5 Sysboot Pin Mapping

Table 25-8 lists the sysboot pin multiplexing for 15×15 and 12×12 packages. There is no dedicated sysboot balls in either of the packages.

Table 25-8. Sysboot Pin Mapping

Sysboot Pin	15×15 Package		12×12 Package	
	Device Ball	Note	Device Ball	Note
sysboot[15]	gpmc_ad[15]	In MuxMode=0xF(1)	N/A, Tied-off to 0b1	
sysboot[14]	gpmc_ad[14]	In MuxMode=0xF(1)	N/A, Tied-off to 0b0	DPLL_CORE REF clock is SYS_CLK1
sysboot[13]	gpmc_ad[13]	In MuxMode=0xF(1)	N/A, Tied-off to 0b0	GPMC not supported
sysboot[12]	gpmc_ad[12]	In MuxMode=0xF(1)	N/A, Tied-off to 0b0	GPMC not supported
sysboot[11]	gpmc_ad[11]	In MuxMode=0xF(1)	N/A, Tied-off to 0b0	GPMC not supported
sysboot[10]	gpmc_ad[10]	In MuxMode=0xF(1)	N/A, Tied-off to 0b0	GPMC not supported
sysboot[9]	gpmc_ad[9]	In MuxMode=0xF(1)	spi1_d[0]	In MuxMode=0xF(1)
sysboot[8]	gpmc_ad[8]	In MuxMode=0xF(1)	spi1_d[1]	In MuxMode=0xF(1)
sysboot[7]	spi2_d[0]	In MuxMode=0xF(1)	spi2_d[0]	In MuxMode=0xF(1)
sysboot[6]	gpmc_ad[6]	In MuxMode=0xF(1)	N/A, Tied-off to 0b0	
sysboot[5]	gpmc_ad[5]	In MuxMode=0xF(1)	N/A, Tied-off to 0b1	No ADC clock division. ADC supports up to 20-MHz SYS_CLK1
sysboot[4]	gpmc_ad[4]	In MuxMode=0xF(1)	spi2_d[1]	In MuxMode=0xF(1)
sysboot[3]	gpmc_ad[3]	In MuxMode=0xF(1)	N/A, Tied-off to 0b1	Using only UART, QSPI_1 and QSPI_4
sysboot[2]	gpmc_ad[2]	In MuxMode=0xF(1)	N/A, Tied-off to 0b0	Using only UART, QSPI_1 and QSPI_4
sysboot[1]	gpmc_ad[1]	In MuxMode=0xF(1)	N/A, Tied-off to 0b0	Using only UART, QSPI_1 and QSPI_4
sysboot[0]	gpmc_ad[0]	In MuxMode=0xF(1)	uart2_ctsn	In MuxMode=0xF(1)

(1) Reset-default mux mode.

25.2.4.6 Boot Interface Pin Multiplexing

Table 25-9 lists the pin multiplexing configuration supported by ROM code according to boot interface. These settings are not restored to default values at ROM Code exit.

NOTE: The ROM code examines the interfaces that are selected to be searched until a valid bootable interface or device is found. The activities on the pads of the searched interfaces must be considered if they are connected to any other peripherals for any other purposes (for example, a LED connected to a GPMC pad muxed internally to a GPIO).

Table 25-9. Pin Multiplexing According to Boot Interface

Boot Device	Boot Interface	Pads	In MuxMode	Interface Signals
XIP	GPMC	gpmc_cs[0]	0x0	gpmc_cs[0]
		gpmc_advn_ale	0x0	gpmc_advn_ale
		gpmc_oen_ren	0x0	gpmc_oen_ren
		gpmc_wen	0x0	gpmc_wen
		gpmc_wait0	0x0	gpmc_wait0
		gpmc_ad[0:7]	0x0	gpmc_ad[0:7]
		gpmc_ad[8:15]	0x0	gpmc_ad[8:15]
		vout1_d[16:23]	0x3	gpmc_a[0:7]
		vin1a_d[8:15]	0x3	gpmc_a[8:15]
		uart1_ctsn	0x3	gpmc_a16
		uart1_rtsn	0x3	gpmc_a17
		uart2_ctsn	0x3	gpmc_a18
		uart2_rtsn	0x3	gpmc_a19

Table 25-9. Pin Multiplexing According to Boot Interface (continued)

Boot Device	Boot Interface	Pads	In MuxMode	Interface Signals
QSPI_1	QSPI1	gpmc_cs[1]	0x1	qspi1_cs[0]
		gpmc_cs[6]	0x1	qspi1_sclk
		uart1_rtsn	0x8	qspi1_rtclk (1)
		gpmc_cs[5:2]	0x1	qspi1_d[0:3]
QSPI_4	QSPI1	gpmc_cs[1]	0x1	qspi1_cs[0]
		gpmc_cs[6]	0x1	qspi1_sclk
		uart1_rtsn	0x8	qspi1_rtclk (1)
		gpmc_cs[5:2]	0x1	qspi1_d[0:3]
UART	UART3	spi1_sclk	0x1	uart3_rxd
		spi1_cs[0]	0x1	uart3_txd

(1) When the QSPI interface operates in Mode 3, this signal is not required to be connected to QSPI chip.

25.3 Booting Overview

25.3.1 Booting Types

Booting is the process of starting a bootstrap from one of the booting devices.

The ROM code has two functions for booting: Peripheral booting and memory booting.

- In peripheral booting, the ROM code polls a selected communication interface such as UART, downloads the executable code over the interface, and executes it in internal RAM. Downloaded software from an external host can be used to program flash memories connected to the device. This special case of peripheral booting is called preflashing; software downloaded for preflashing is called the flash loader. The flash loader burns a new client application image in external flash memory. Initial software is a generic term for bootstrap, downloaded software, and flash loader. A software (warm) reset can be performed after the image is burned.
- In memory booting, the ROM code finds the bootstrap in permanent memories such as flash memory devices and executes it. This process is normally performed after a cold or warm device reset.

The ROM code detects whether the device should download software from a peripheral interface by using the sysboot pad configuration. This mechanism encompasses initial flashing in production (external memory is empty) and reflashing in service (external memory is already programmed).

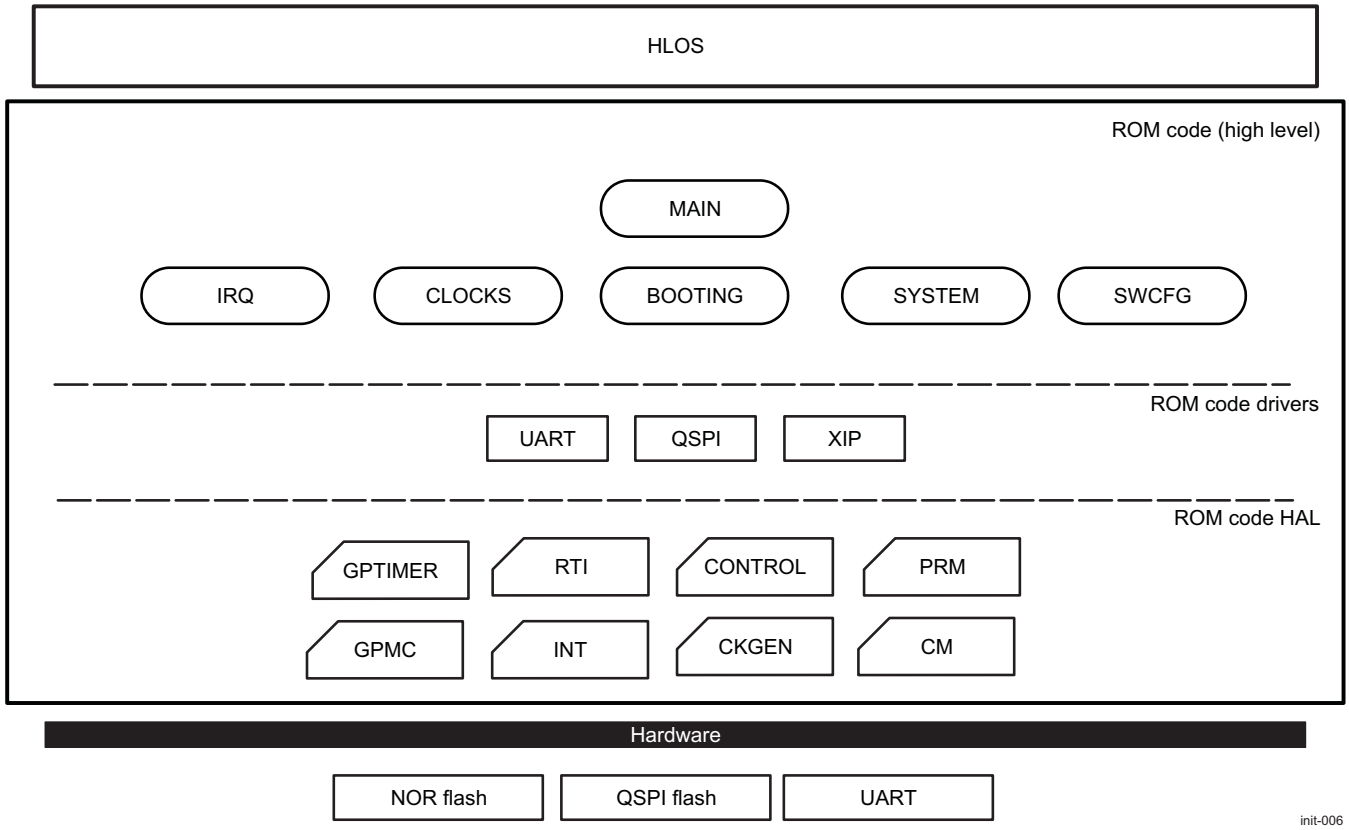
25.3.2 ROM Code Architecture

Figure 25-4 shows the ROM code architecture. It is split into three main layers with a top-down approach: high-level, drivers, and hardware abstraction layer (HAL). One layer communicates with a lower-level layer through a unified interface.

- The high-level layer performs the main tasks of the ROM code: multicore startup, RTI and clock configurations, interrupt management, and main booting routine.
- The driver layer implements the logical and communication protocols for any booting device in accordance with the interface specification.
- The HAL implements the lowest level code for interacting with the hardware infrastructure modules. End booting devices (typically external flash components) are attached to the device I/O pads.

Figure 25-4 shows the three layers with their modules.

Figure 25-4. ROM Code Architecture

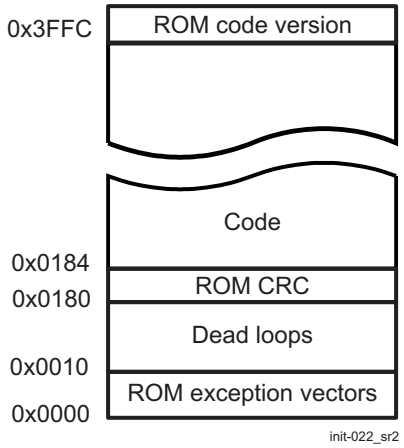


25.4 Memory Maps

25.4.1 ROM Memory Map

Figure 25-5 shows the ROM memory map.

Figure 25-5. ROM Memory Map



- ROM exception vectors

  Exceptions are redirected to ROM exception vectors (see Table 25-10). The reset exception is redirected to the ROM code startup. Once the ROM Code has started its execution, exception vector base address can be remapped in Internal RAM

Table 25-10. ROM Exception Vectors

Address	Exception	Content
0x0000	Stack Pointer	Address for r13 (stack pointer)
0x0004	Reset	Entry of the ROM Code
0x0008	NMI ISR Location	Address of the DEADLOOP_NMI
0x000C	Hard Fault ISR Location	Address of the DEADLOOP_HardFault

  By default the NVIC vector table starts at address 0x0 at boot process, and these four exception vectors are mandatory. They are required because the NMI and Hard Fault can potentially occur during the boot process. Other exceptions cannot take place until they are enabled.

- ROM code cyclic redundancy check (CRC)

  The ROM code CRC is calculated as 32-bit CRC code (CRC-32-IEEE 802.3) for the address range 0x0000–0x3FFC. The 4-byte CRC code is stored at location 0x180.

- Dead loops

  Dead loops are branch instructions coded in Arm mode. They have multiple purposes (see Table 25-11). The fixed location of these dead loops facilitates debugging and testing

Table 25-11. Dead Loops

Address	Purpose
0x0010	NMI Interrupt
0x0012	Hard Fault
0x0014	Memory Management
0x0016	Bus Fault
0x0018	Usage Fault
0x001A	SVC Call
0x001C	Debug Monitor

Table 25-11. Dead Loops (continued)

Address	Purpose
0x001E	Pendable request
0x0020	System Tick timer
0x0022	Test Pass
0x0024	Test Fail
0x0026	Bad Cortex
0x0028	BadFlash
0x002A	BootMemError
0x002C	Reserved
0x002E	Reserved

- Code

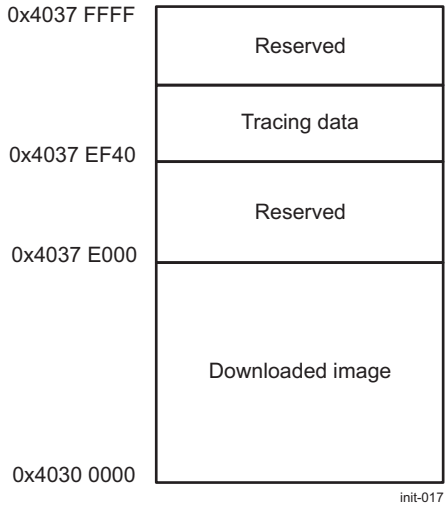
  This space is used to hold code and constant data.
- ROM code version

  The ROM code version consists of two BCD numbers: major and minor. It can be used to identify the ROM code release version burned in a given IC. The ROM code version is a 32-bit hexadecimal value at address 0x3FFC.
  ROM Code version number is:
  – 0x4002

25.4.2 RAM Memory Map

The partitioning of the on-chip SRAM (L3 OCM RAM) shown in Figure 25-6 is used during the booting process.

Figure 25-6. RAM Memory Map



- Downloaded image

  This space is used by the ROM code to store a downloaded booting image. It can be up to 504 KiB.
- Tracing data

  This area contains trace vectors reflecting the execution path of the ROM code. Table 25-12 describes the ROM code tracing data. For more information about ROM code tracing, see Section 25.10, *Tracing*.

Table 25-12. Tracing Data

Address	Size	Description
0x4037 F040	32 bits	Current tracing vector, word 1
0x4037 F044	32 bits	Current tracing vector, word 2
0x4037 F048	32 bits	Current tracing vector, word 3
0x4037 F04C	32 bits	Current tracing vector, word 4
0x4037 F050	32 bits	Cold reset run tracing vector, word 1
0x4037 F054	32 bits	Cold reset run tracing vector, word 2
0x4037 F058	32 bits	Cold reset run tracing vector, word 3
0x4037 F05C	32 bits	Cold reset run tracing vector, word 4
0x4037 F060	32 bits	Current copy of the PRM_RSTST register (reset reasons)

25.4.3 AMMU Mapping

The ROM Code configures AMMU to map certain address space to allow eXecute-In-Place (XIP) for IPU. Table 25-13 indicates the physical/virtual mapping correspondence.

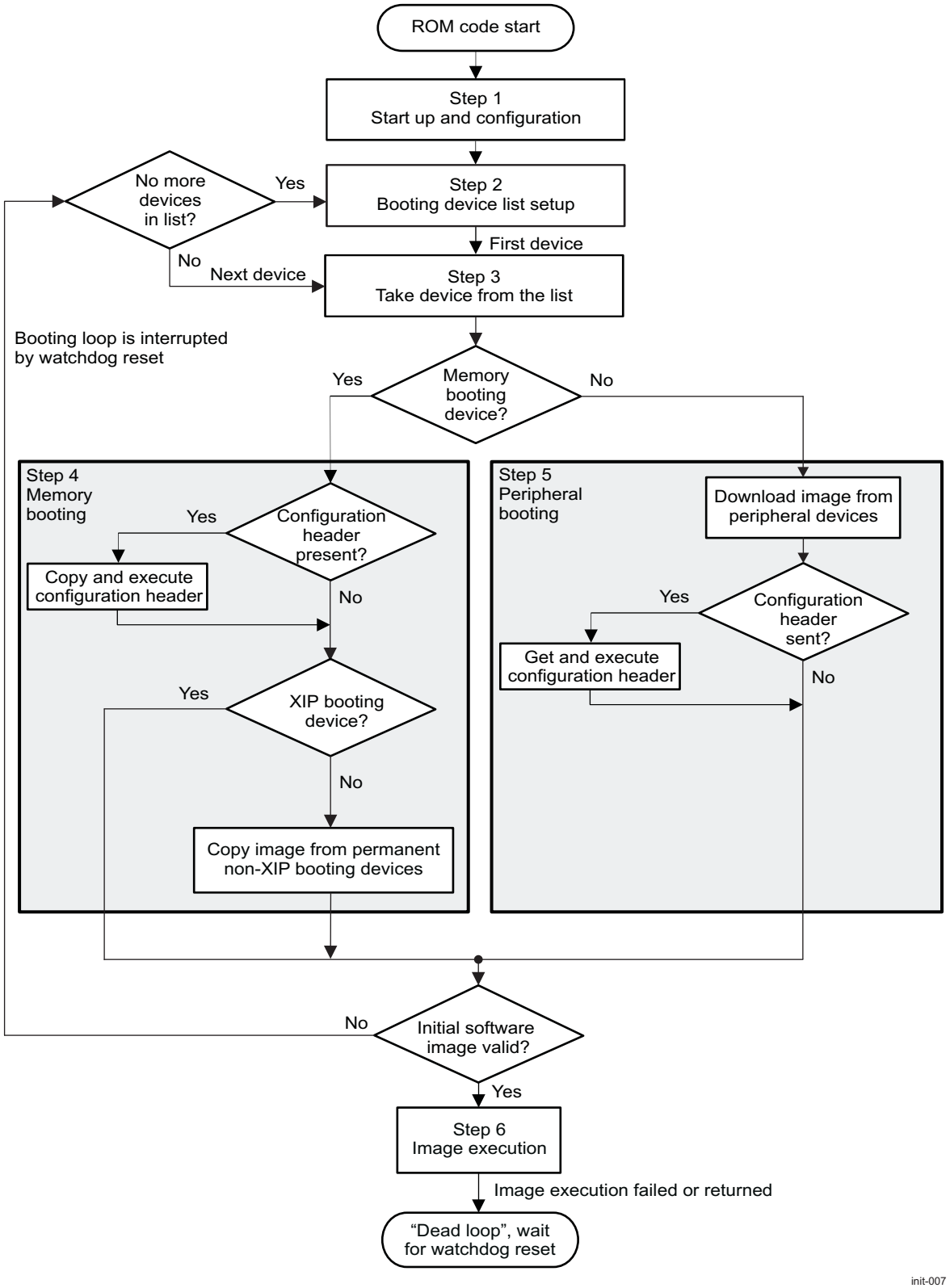
Table 25-13. AMMU Mapping

Module	Physical Address	Virtual Address	Mapped Size	Note
IPU ROM	0x5500 0000	0x0000 0000	32 KiB	
IPU RAM	0x5502 0000	0x0002 0000	64 KiB	
OCMC RAM(L3)	0x4030 0000	0x0030 0000	512 KiB	For XIP
GPMC - Memory	0x0800 0000	0x0800 0000	32 MiB	For XIP
QSPI- Memory	0x5C00 0000	0x0400 0000	32 MiB	For XIP

25.5 Overall Booting Sequence

Figure 25-7 shows the ROM code flow chart.

Figure 25-7. Overall Booting Sequence



The main loop of the booting module goes through the booting device list and tries to get an image from the currently selected booting device. The ROM code performs the following steps:

Step 1. Basic configuration and initialization. Reading of SYSBOOT pins.

Step 2. A booting device list is created (see Section 25.6.5, *Booting Device List Setup*). The list consists of all devices to be searched for a booting image. The list is created based on the SYSBOOT pins.

Step 3. The main loop of the booting procedure goes through the booting device list and tries to search for an image from the currently selected booting device. This loop is exited if a valid booting image is found and successfully executed or when the watchdog expires. If an image is found, ROM code executes memory booting or peripheral booting, depending on the type of the current booting device:

- Memory booting is executed when the booting device is XIP memory, QSPI, or SPI.
- Peripheral booting is executed when the booting device is UART.

Step 4. Memory booting reads data from memory-type devices. Memory booting is described in detail in Section 25.8, *Memory Booting*.

Step 5. Peripheral booting downloads data from communication interfaces. Peripheral booting is described in Section 25.7, *Peripheral Booting*.

Step 6. For a GP device, boot image automatically starts.

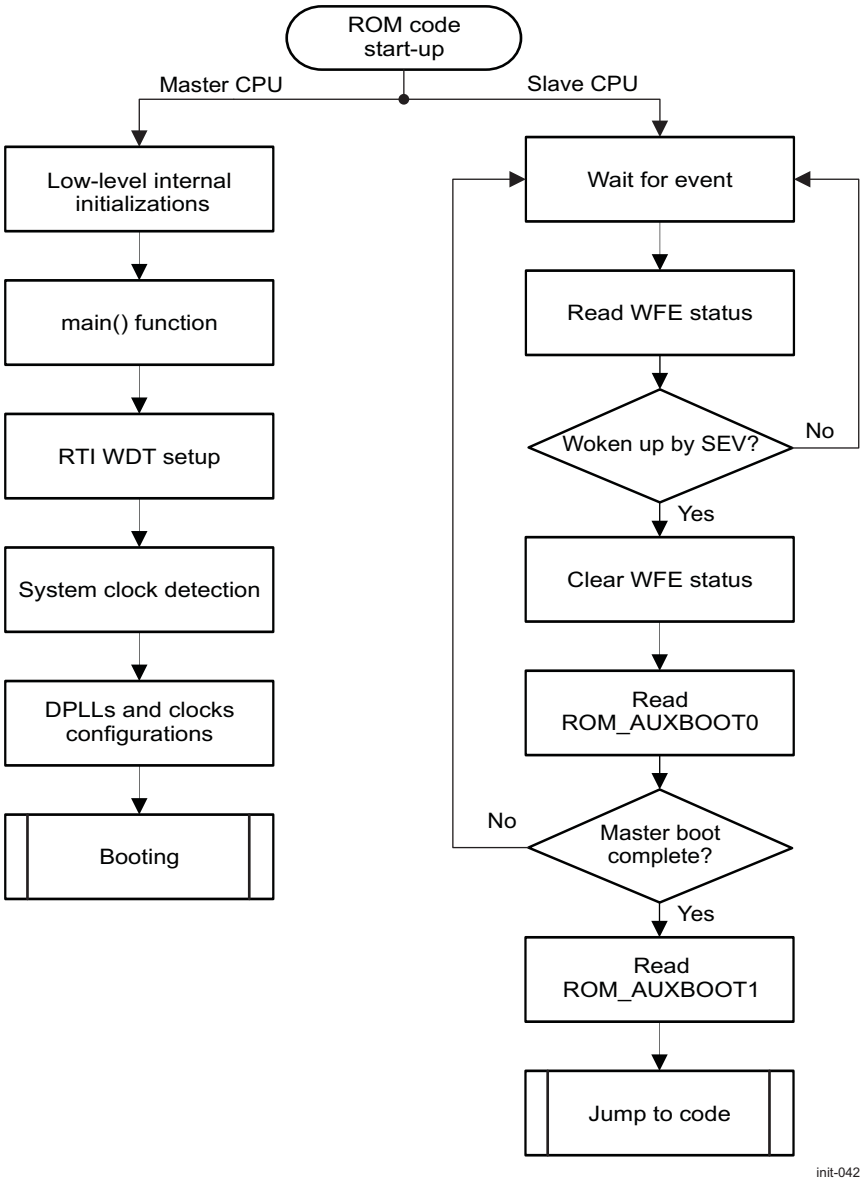
An additional feature of the booting module is the execution of the Configuration Header (CH). The CH configures the system for faster and more flexible booting from the selected permanent or peripheral booting device. The CH, which is optional, is described in *Section 25.9.2, Configuration Header*.

25.6 Startup and Configuration

25.6.1 Startup

Figure 25-8 shows the ROM code start-up sequence.

Figure 25-8. ROM Code Multiprocessor Start-Up Sequence



The base address of the public vector is configured to the reset vector of ROM code (0x0000). The master CPU performs the basic initialization. Next, CPU configures RTI WDT (set to 3 minutes), detects system clock, and configures the system clock tree. Finally, the CPU jumps to the booting routine.

No specific configuration is performed for the slave CPU, which keeps its default configuration after reset. The slave CPU is rapidly held in wait-for-event (WFE) state. It stays in this state while the master CPU completes the public boot process and until jumping to the external software (for example, HLOS). At this stage, the external software can wake up the slave CPU by executing an SEV command.

Two internal memory-mapped registers are available to the OS for communicating start-up information. The ROM_AUXBOOT0 and ROM_AUXBOOT1 registers are in the Control Module.

- ROM_AUXBOOT0 is used as a status register to signal the slave CPU that it must wake up after the

send event operation initiated by the master CPU.

- ROM_AUXBOOT1 contains the physical address location to which the slave CPU must jump after wakeup.

25.6.2 Control Module Configuration

Table 25-14 lists the Control Module registers modified at each ROM code startup. It is a Control Module requirement to be met prior to modify any of the pad control registers. These registers are not reverted back to default values (that is, to LOCK state) when ROM code completes.

Table 25-14. Control Module Registers Modified by ROM Code at Each Startup

Register	Value	Meaning
CTRL_CORE_MMR_LOCK_1	0x2FF1AC2B	Unlock Control Module registers starting at address offset 0x0000 0100 and ending at 0x0000 079F
CTRL_CORE_MMR_LOCK_2	0xF757FDC0	Unlock Control Module registers starting at address offset 0x0000 07A0 and ending at 0x0000 0D9F
CTRL_CORE_MMR_LOCK_3	0xE2BC3A6D	Unlock Control Module registers starting at address offset 0x0000 0DA0 and ending at 0x0000 0FFF
CTRL_CORE_MMR_LOCK_4	0x1EBF131D	Unlock Control Module registers starting at address offset 0x0000 1000 and ending at 0x0000 13FF
CTRL_CORE_MMR_LOCK_5	0x6F361E05	Unlock Control Module registers starting at address offset 0x0000 1400 and ending at 0x0000 1FFF

Once the booting device list is completed, the ROM code applies the pin multiplexing settings as described in Section 25.2.4.6, *Pin Multiplexing According to Boot Peripheral*.

25.6.3 PRCM Module Mode Configuration

Table 25-15 lists the PRCM module mode, clock control, and power control registers modified at each ROM code startup. These registers are not reverted back to default values when ROM code completes.

Table 25-15. PRCM Module Mode Registers Modified by ROM Code

Register	Field	Value
CM_L3INSTR_L3_INSTR_CLKCTRL	MODULEMODE	DISABLED
CM_L3INSTR_OCP_WP_NOC_CLKCTRL	MODULEMODE	DISABLED
CM_CM_CORE_PROFILING_CLKCTRL	MODULEMODE	DISABLED
CM_PRM_PROFILING_CLKCTRL	MODULEMODE	DISABLED
CM_CM_CORE_AON_PROFILING_CLKCTRL	MODULEMODE	DISABLED
CM_EMU_CLKSTCTRL	CLKTRCTRL	HW_AUTO
CM_DSP1_CLKSTCTRL	CLKTRCTRL	HW_AUTO
CM_DSP1_DSP1_CLKCTRL	MODULEMODE	DISABLED
PM_DSP1_PWRSTCTRL	POWERSTATE	OFF
CM_DSP2_CLKSTCTRL	CLKTRCTRL	HW_AUTO
CM_DSP2_DSP2_CLKCTRL	MODULEMODE	DISABLED
PM_DSP2_PWRSTCTRL	POWERSTATE	OFF

25.6.4 Clocking Configuration

The ROM code detects the system clock frequency (SYS_CLK1) from the sysboot[9:8] pin value. These pins should be set accordingly on the PCB. The supported system clock frequencies in the device are:

- 19.2 MHz
- 20 MHz

- 27 MHz

See Section 25.2.3.2, *Clocking scheme*, and Section 25.2.4.2, *System clock speed configuration*.

After detecting the input clock, the ROM code configures the clocks and DPLLs required for ROM code execution.

The configured DPLLs are:

- DPLL_PER: locked to provide clocks to peripheral blocks
- DPLL_CORE: locked to provide L3_MAIN interconnect and L4 interconnect

The DPLLs and PRCM clock dividers are configured with the default values of the ROM code (depending on the detected system input clock) after cold or warm reset in order to give the same working conditions to the ROM code sequence.

Table 25-16 summarizes the default ROM code clock settings.

Table 25-16. ROM Code Default Clock Settings

Clock	Frequency (MHz)	Source
DPLL_CORE clock with F_{DPLL} locked frequency	2128	Gated SYS_CLK1
EMIF_PHY_GCLK(1)	44.33	DPLL_DDR (M2)
EMIF_DLL_GCLK	266	DPLL_DDR .HSDIVIDER (H11)
CORE_X2_CLK	266	DPLL_CORE.HSDIVIDER (H12)
CORE_CLK	266	CORE_X2_CLK
DSP2_CLK	212.8	DPLL_CORE.HSDIVIDER (H14)
CORE_IPU_ISS_BOOST_CLK	212.8	DPLL_CORE.HSDIVIDER (H22)
CORE_ISS_MAIN_CLK	152	DPLL_CORE.HSDIVIDER (H23)
L3_ICLK	133	CORE_CLK
L4_ICLK	66.5	L3_ICLK
DSP_DPLL_HS_CLK	266	CORE_X2_CLK
DPLL_PER – clock with Fdpll locked frequency	768	Gated SYS_CLK1
FUNC_192M_CLK	192	DPLL_PER (M2)
FUNC_256M_CLK	256	DPLL_PER.HSDIVIDER (H11)
DSS_GFCLK	192	DPLL_PER.HSDIVIDER (H12)
PER_QSPI_CLK	192	DPLL_PER.HSDIVIDER (H13)

(1) This clock is intentionally set up at low frequency to ensure correct initialization of external DRAM components.

However it is possible to override the default clock settings. There are three ways to change DPLLs and all related clock divider, gating, and multiplexer configurations during the boot:

- ROM code default settings, described in Table 25-16. They are always applied at any reset.
- The CH, described in Section 25.9.2, *Configuration Header*. The CH lets users have a known configuration (about GPMC and clock registers) after memory or peripheral booting.

25.6.5 Booting Device List Setup

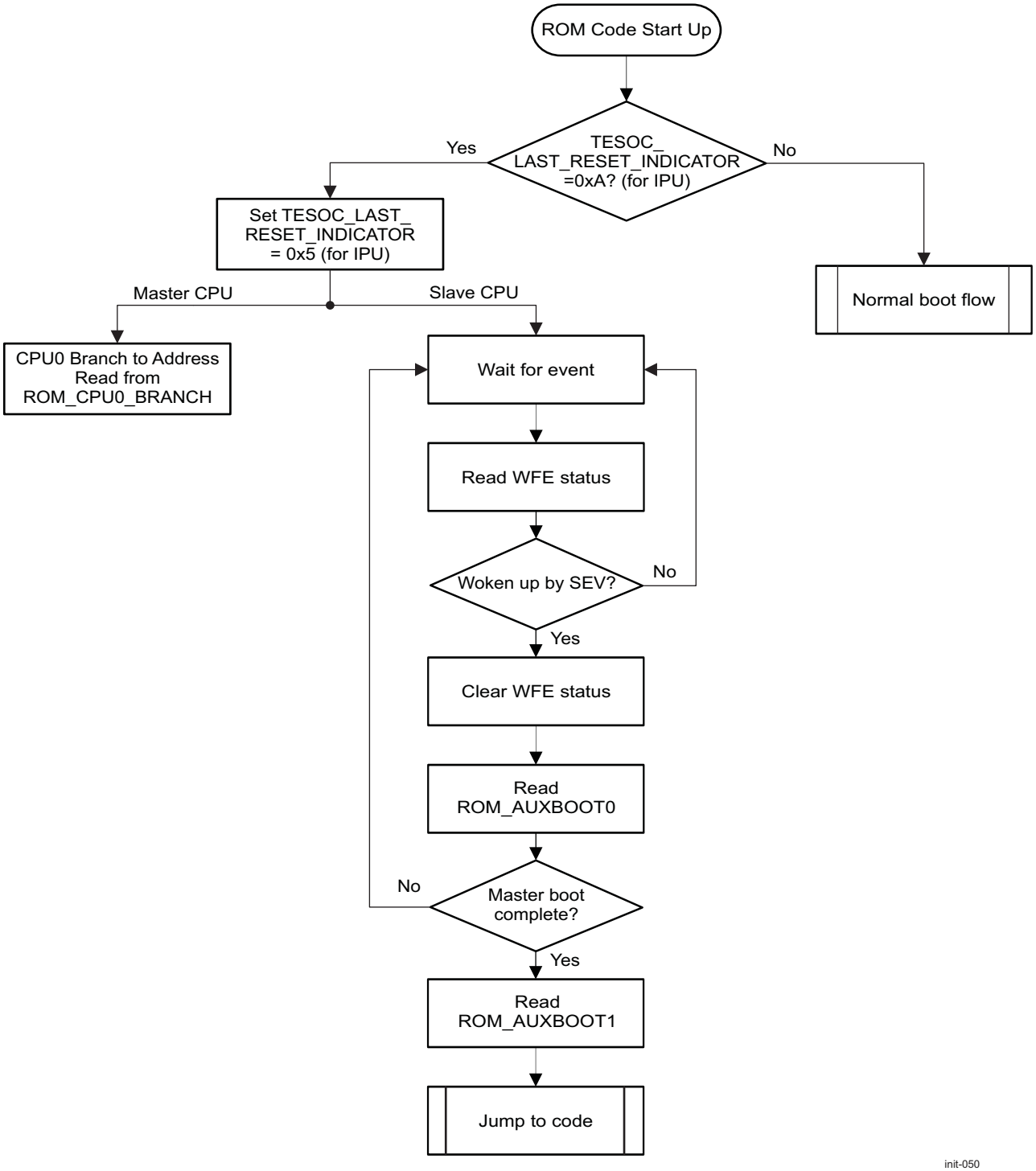
The ROM code creates a device list based on sysboot pins. The sysboot[4:0] signals latched in the control module are used to index the device table from which the list of devices is extracted. Only permanent booting devices are put to the list when the reset is not power-on and the devices are taken from the sysboot pins.

25.6.6 Warm Reset Device Selection

After a warm reset the ROM Code wakeups IPU to software wakeup registers available in Control Module (if present and valid) as shown in Figure 25-9. When the reset is a Tester-On-Chip (TESOC) reset, only partial boot is performed. Master CPU jumps to address written in the ROM_CPU0_BRANCH register. When TESOC is not the reset reason, the SYSBOOT configuration takes place and normal boot is performed.

NOTE: TESOC is not supported on the DRA78x family of devices.

Figure 25-9. Warm Reset Wakeup Flow



25.7 Peripheral Booting

25.7.1 Description

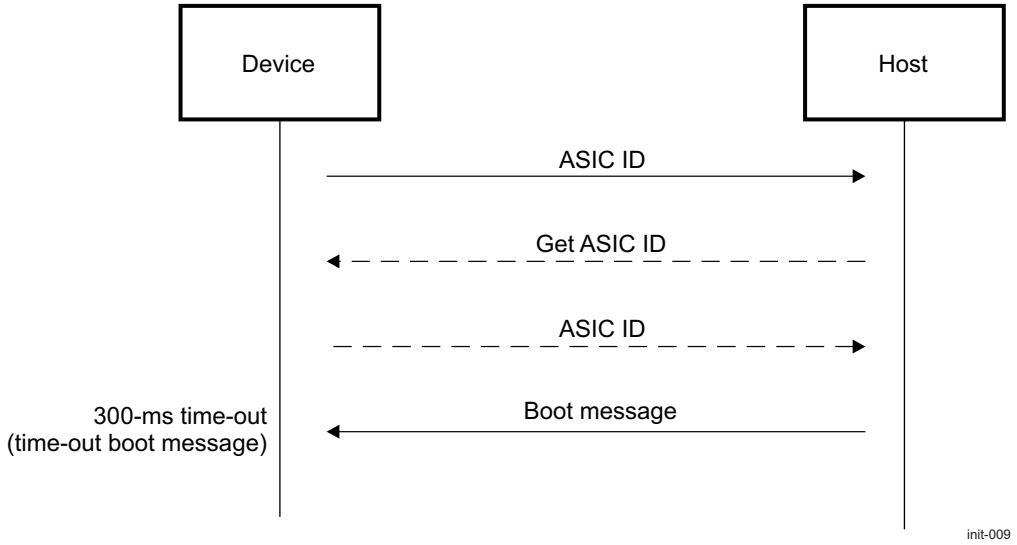
The ROM code can boot from these peripherals:

- UART3: 115.2 Kbps, 8 bits, even parity, 1 stop-bit, no flow control

The purpose of booting from a peripheral is to download a flash loader code from an external host. This booting method is used primarily for programming flash memories connected to the device (for example, in the case of initial flashing, firmware update or servicing). Figure 25-11 shows the overall peripheral booting procedure. It consists of a synchronization phase (handshake between the host and the device) and a transfer phase.

When booting from the UART, the ROM code first initializes the UART3 interface. Then the ROM code sends an ASIC ID block of data. From there, it expects to receive a boot message from the host within 300 ms, by default. Figure 25-10 shows this procedure.

Figure 25-10. Synchronization Phase for UART



During the synchronization phase (see Figure 25-10), the device can provide a small packet of data called the ASIC ID (described in Table 25-17). It is a simple structure that contains different kinds of information, such as ROM version, checksums, and ID.

The host can decide the desired operation by providing a booting message (see Table 25-20). This message can be: Get ASIC ID, peripheral boot, change device, or next device. If the device receives the Get ASIC ID boot message, it sends back the ASIC ID contents.

If the change device or next device message is received, the ROM code stops the current peripheral booting procedure and returns to the main booting, which decides about the next booting device according to the boot message received.

If the peripheral boot message is received without a time-out, the device is entering the transfer phase. From there, the flash loader image size (as a 32-bit word) and the flash loader image itself are expected to be received. The ROM code waits up to 1 minute for completion of image size reception, and up to 1 more minute to download the image. If the download procedure does not complete before this time, the peripheral booting procedure aborts. ROM code continues to examine the devices included in the booting device list. If the download procedure passes, then the image can be executed.

The flash loader image is downloaded directly into internal RAM from address 0x4030 0000 and the maximum size of the downloaded image is 504 KiB.

NOTE: Sending an image size of zero skips the peripheral booting procedure.

The UART connection is left open at the end of the transfer phase and once exiting the ROM code for the initial software to take over. It means the initial software can reuse the currently established connection.

Table 25-17. ASIC ID Structure

ASIC ID Item	Size (Bytes)	Description
Items	1	Number of subblocks
ID subblock	7	Device identification information
Reserved subblock	4	This subblock is transmitted, but does not contain useful information in case of GP device
Reserved subblock	23	This subblock is transmitted, but does not contain useful information in case of GP device
Reserved subblock	35	This subblock is transmitted, but does not contain useful information in case of GP device

Table 25-18. Items

Offset	Size (Bytes)	Description
0x00	1	0x04: Number of subblocks (UART boot)

Table 25-19. ID Subblock

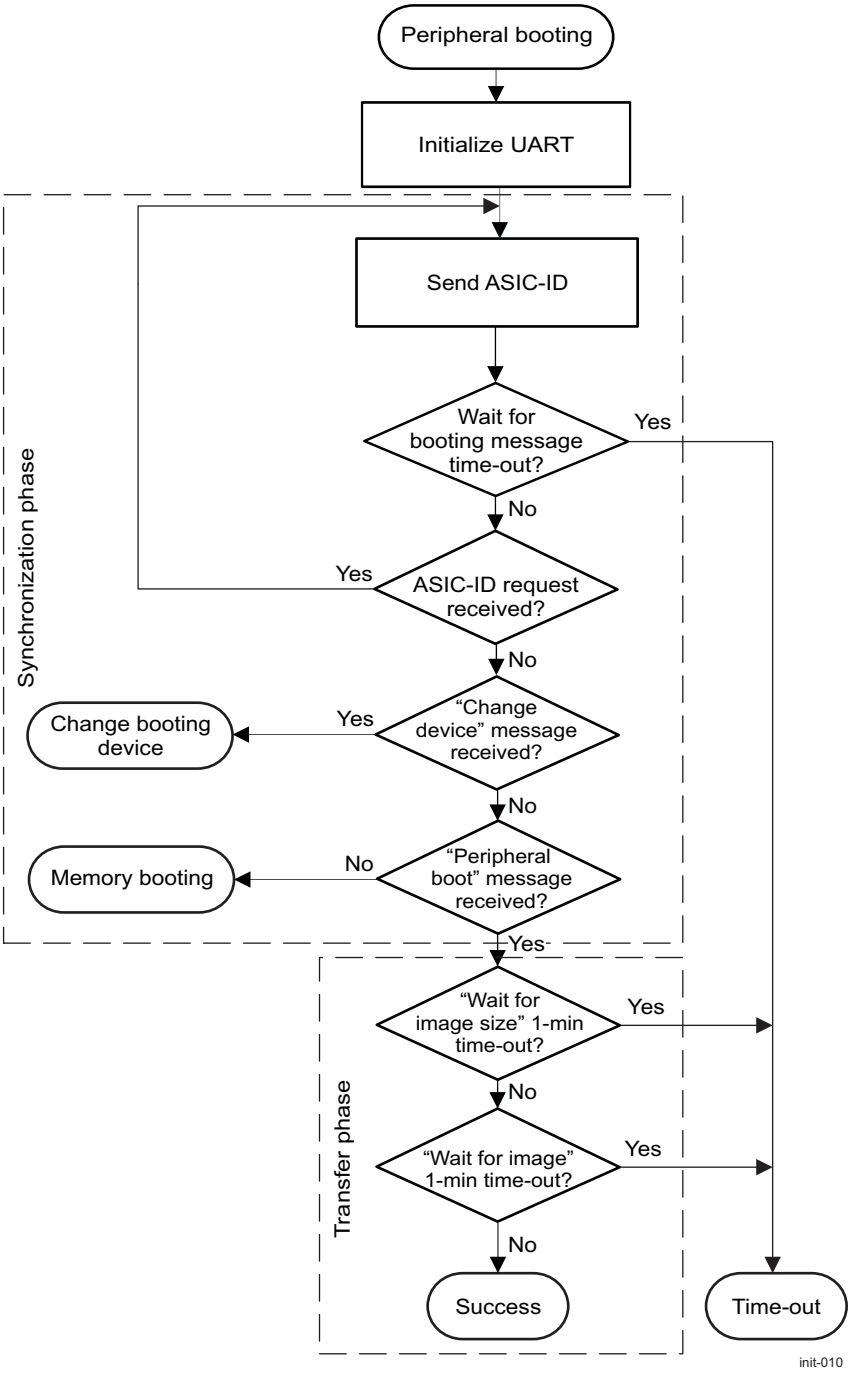
Offset	Size (Bytes)	Description
0x01	1	0x01: Subblock ID
0x02	1	0x05: Subblock size
0x03	1	0x01: Fixed value
0x04	2	Device ID number
0x06	1	0x07: CH enabled 0x17: CH disabled
0x07	1	ROM code revision 0x02: SR2.0

Table 25-20. Booting Messages

Message Name	Value	Description
Peripheral boot	0xF003 0002	Continue peripheral booting.
Get ASIC ID	0xF003 0003	ASIC ID request. The Get ASIC ID request message is optional. If received, the ROM code sends its ASIC ID data to the host in return. The host can issue the Get ASIC ID message multiple times if required. Table 25-17 describes the structure of the ASIC ID.
Change device	0xF003 xx06	Skip current peripheral booting and continue booting from device type indicated by xx: 0x01: XIP 0x02: XIP (with wait monitoring) 0x0A: QSPI_1 0x0B: QSPI_4 0x0C: QSPI_4(XIP) 0x43: UART Others: Reserved
Next device	0xFFFF FFFF	Skip current device and move to the next device on the device list.
Memory booting	Others	Skip current peripheral booting and move to the first device for memory booting.

Figure 25-11 shows the peripheral booting procedure.

Figure 25-11. Peripheral Booting Procedure



25.7.2 Initialization Phase for UART Boot

The ROM code supports booting from a UART interface with the following characteristics:

- UART3 interface
- Communication parameters set to 115.2 Kbps, 8 bits, even parity, 1 stop-bit, no flow control
- Two-pin interface: RX/TX
- The boot message default time-out is 300 ms (time-out boot message)

25.8 Memory Booting

25.8.1 Overview

The memory booting process starts an external code in memory devices. ROM code can use only the memory type of booting devices as permanent booting devices (that is, devices examined after both cold [POR] and warm resets). Temporary booting devices are examined only after cold resets. The supported permanent booting devices are:

- NOR flash devices (or other XIP devices) on GPMC
- SPI/QSPI flash memories

Two main groups of permanent booting devices are distinguished by code shadowing. Code shadowing means copying code from a nondirectly addressable device (non-XIP) to RAM, where the code can be executed. Directly addressable devices are XIP devices.

Figure 25-12 shows the general memory booting procedure common to all types of devices. First, CH is copied to internal RAM. It is copied even for XIP devices, because the device can temporarily lose a connection with XIP memory during CH execution (for example, while updating interface timings). The second step is to shadow the image, if the device is not XIP. The last step is image execution and any return from image results in a dead loop.

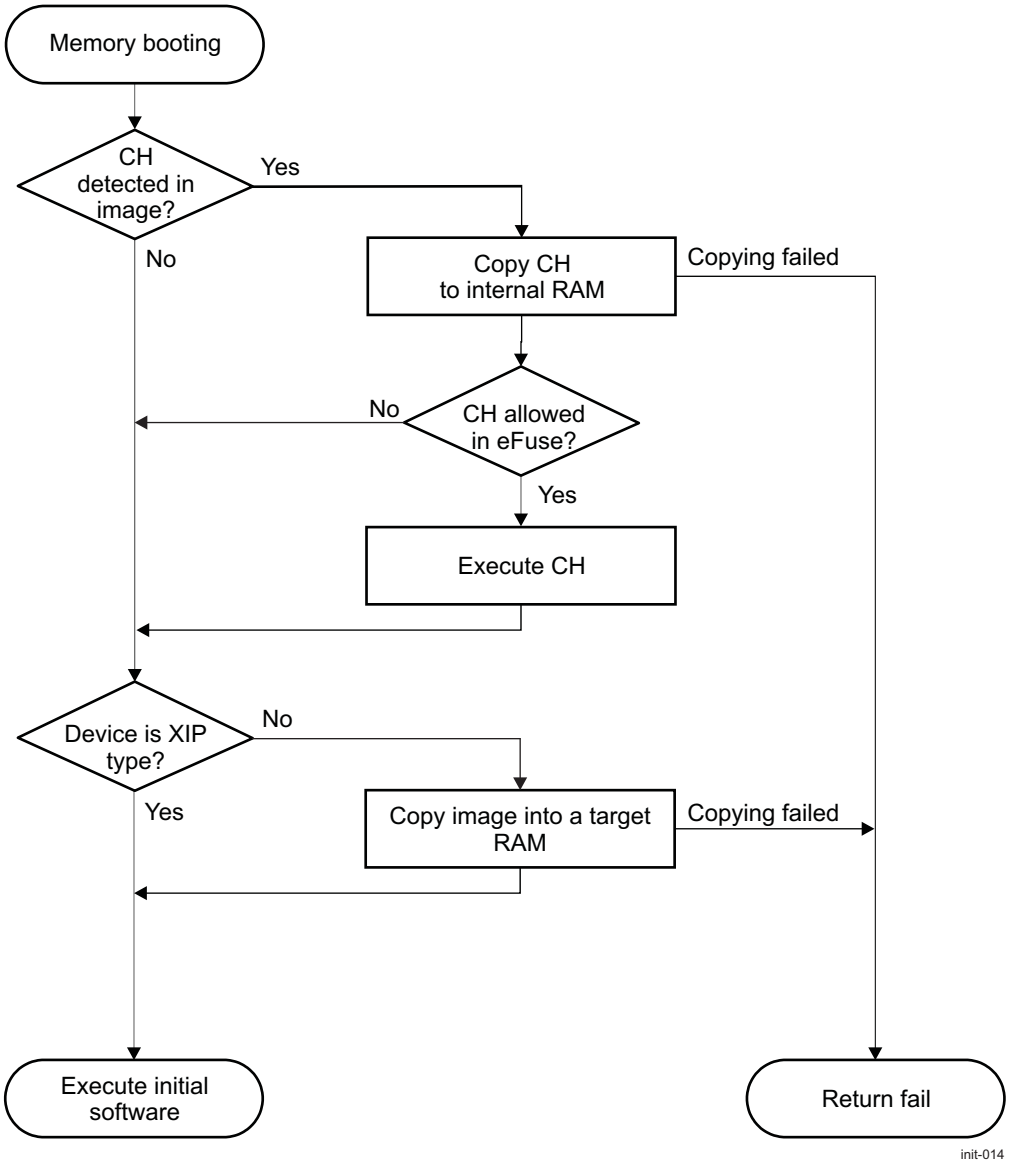
If CH copying or shadowing fails, memory booting returns to the main booting procedure, which selects the next device for booting.

NOTE: A booting image is considered to be present on a GP device, when the first 4-byte word of the sector is not equal to 0000 0000h or FFFF FFFFh.

During the first read sector (512 bytes) call, sectors are copied to a temporary device on-chip SRAM buffer. Once the image is found and the destination address is known, the content of the temporary buffer is moved to the target device on-chip SRAM location so it is required to reread the first image sector. For a GP device, the GP header is not copied into target buffer location; therefore, only executable code is in device on-chip RAM, with the first executable instruction at the destination address.

SPI/QSPI devices can hold up to two copies of the booting image. Therefore, the ROM code searches for one valid image, if present, by walking over the first blocks of mass storage space.

Figure 25-12. Memory Booting Procedure

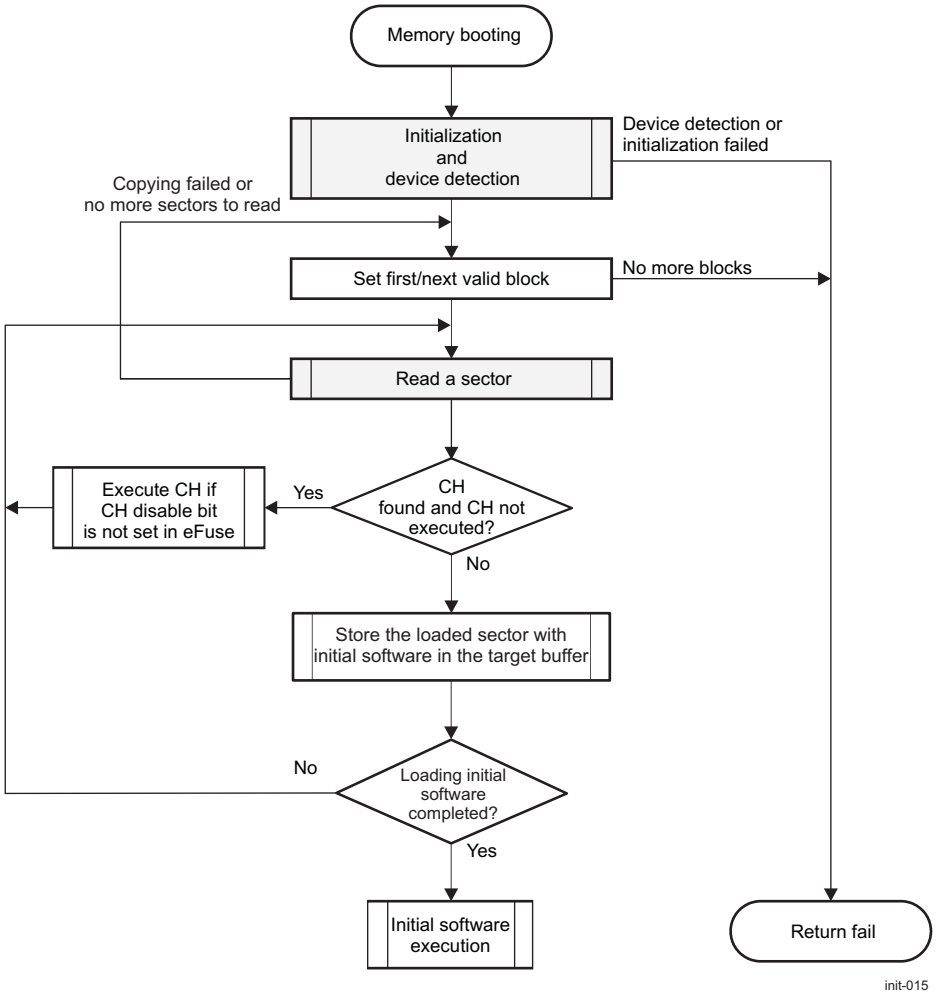


25.8.2 Non-XIP Memory

Figure 25-13 shows the procedure used when memory booting runs with non-XIP memories. The shaded procedures are specific to each memory. The ROM code searches for the image in the first two physical blocks of the memories. Other devices use only one copy of the image and the block loop runs only once.

During image shadowing on a GP device, the CH is expected to be in a separate sector before the initial software.

Figure 25-13. Image Shadowing on GP Device



For more information about the GPMC module, see Section 10.3, *General-Purpose Memory Controller*.

The following sections describe the supported device types.

25.8.3 XIP Memory

The ROM code can boot directly from XIP devices, such as NOR flash memories, that have the following characteristics:

- The GPMC is the communication interface.
- Memories up to 1Gibit (128MiB) can be connected.
- 8-bit or 16-bit data bus width, configured through sysboot[13] (=1 for 16-bit)
- Non-muxed or address/data multiplexed mode, configured through sysboot[12:11] (=0x1 for A/D mux)
- The GPMC clock is 133 MHz.
- The device is connected to CS0 mapped to address 0x0800 0000.
- The wait pin gpmc_wait0 signal is monitored according to the sysboot[10] configuration pin (=1 is enabled)
- For an XIP memory booting, no user intervention is required; the following debugging steps are described. Only the CH, which is not mandatory, lets users change clock settings and GPMC parameters. Failure in CH copying causes a return to the main booting procedure, which selects the next device for booting.

Booting from an XIP device consists of the following steps:

1. Configure the GPMC for XIP device access.
2. Verify that the CH is present at address 0x0800 0000. If the CH is present, copy the entire sector (512 bytes) to internal RAM and execute the CH.
3. Set the image location:
   - 0x0800 0000 if the CH is not found
   - 0x0800 0200 if the CH is found
4. Verify that a bootable image is at the image location.
5. If the image is found, execute it.
6. If the image is not found, return from XIP booting to the main booting loop.

25.8.3.1 GPMC Initialization

Table 25-21 lists the timing settings of the GPMC when set for XIP and other address-data accessible devices. Table 25-21 is included for debug information.

Table 25-21. XIP Timing Parameters

Parameter	Value (Clock Cycles)[1]	Register Initialized (where i = 0)
Write cycle period	17	GPMC_CONFIG5_i[12:8] WRCYCLETIME
Read cycle period	17	GPMC_CONFIG5_i[4:0] RDCYCLETIME
CS low time	1	GPMC_CONFIG2_i[3:0] CSONTIME
CS high time	16	GPMC_CONFIG2_i[12:8] CSRDOFFTIME
ADV low time	1	GPMC_CONFIG3_i[3:0] ADVONTIME
ADV high time	2	GPMC_CONFIG3_i[12:8] ADVRDOFFTIME
OE low time	3	GPMC_CONFIG4_i[3:0] OEONTIME
OE high time	16	GPMC_CONFIG4_i[12:8] OEOFFTIME
WE low time	3	GPMC_CONFIG4_i[19:16] WEONTIME
WE high time	15	GPMC_CONFIG4_i[28:24] WEOFFTIME
Data latch time	15	GPMC_CONFIG5_i[20:16] RDACCESSTIME

(1) The one clock cycle is approximately 7.5 ns, which corresponds to a 133-MHz frequency.

There is no specific identification routine executed before booting from an XIP device.

25.8.4 SPI/QSPI Flash Devices

SPI/QSPI Flash memories provide a storage solution for systems with limited space, pins and power.

The ROM code support for SPI/QSPI devices has the following characteristics:

- 24-bit addressing, up to 128 Mbit (16 MiB), no banking
- QSPI1 on CS0 is the communication interface
- Uses Mode 3:
  - Clock inactive state = high
  - Data input captured on rising edge of clock
  - Data output generated on falling edge of clock
- QSPI 4-bit data read mode at 48 MHz in configuration port mode
  - Read command is 0x6B (Fast Quad Read), 3 address bytes,1 dummy bytes and read type is quad read
  - ROM will not perform any quad-enable sequence nor bank register update
- SPI 1-bit data read mode at 12 MHz in configuration port mode
  - Read command is 0x03 (Single Read), 3 address bytes, 0 dummy bytes and read type is normal read.

- Up to two redundant images can be stored in SPI/QSPI. The offset between them must be 256 KiB.

A booting image is considered to be present on a GP device, when the first 4-bytes word is not equal to 0000 0000h or FFFF FFFFh.

NOTE: ROM code does not perform any specific action to detect, reset or power up the QSPI device. QSPI is assumed to be properly powered and reset to be completed before every attempt to boot by ROM code.

25.9 Image Format

25.9.1 Overview

An image has two major parts:

- An optional Configuration Header (CH)
- Software to execute

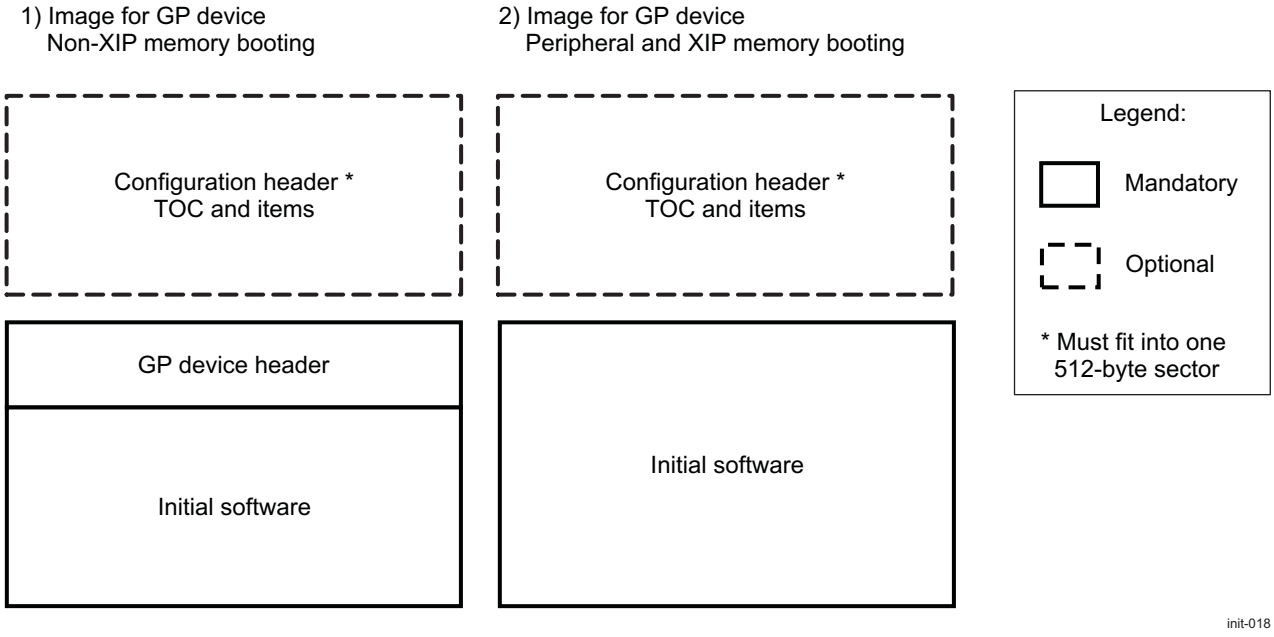
The CH can contain several parameters set by users to speed up booting. It is further described in Section 25.9.2, *Configuration Header*.

The second part contains the software that is loaded into the memory and executed.

Figure 25-14 is an overview of the boot image formats. There are two image types:

1. GP non-XIP memory booting: This image type is used for memories that require shadowing (for example, SPI). A GP image for non-XIP memory may not contain a CH and start straight from the GP header. Next, there must be a small header (referred to as a GP header) that contains information about the size and the destination address.
2. When the memory device is of XIP type (for example, NOR), the GP header is not required, and the image can contain code for direct execution. Optionally, the first sector can contain a CH. The same image format is used for peripheral booting (where the code is transferred to internal RAM).

Figure 25-14. Image Formats



25.9.2 Configuration Header

The ROM code default settings (such as clock frequencies, GPMC, or QSPI interfaces) can be tuned by the user by using the CH.

The CH can contain the following parts:

- Settings: Various clock settings (mandatory)
- FLASH: Flash interface (GPMC) settings
- QSPI: QSPI interface settings

The beginning of the CH is a table of contents (TOC), which points to each item. This is described in Figure 25-15. Each TOC item is a simple structure described in Table 25-22. The complete CH (CH TOC and items) should fit in a 512-byte sector.

The ROM code identifies the presence of a CH by reading the first TOC item if it contains a known string (CHSETTINGS, CHFLASH, etc.). Next, the TOC is identified and searched until a 0xFFFF FFFF offset is found. The CH is read and parameters are executed sequentially.

For the sake of simplicity, each field represents the content of a register to be modified. Only fields required for the configuration are used; fields for status, for instance, are not modified and therefore are not shown in the tables.

Figure 25-15. CH Format

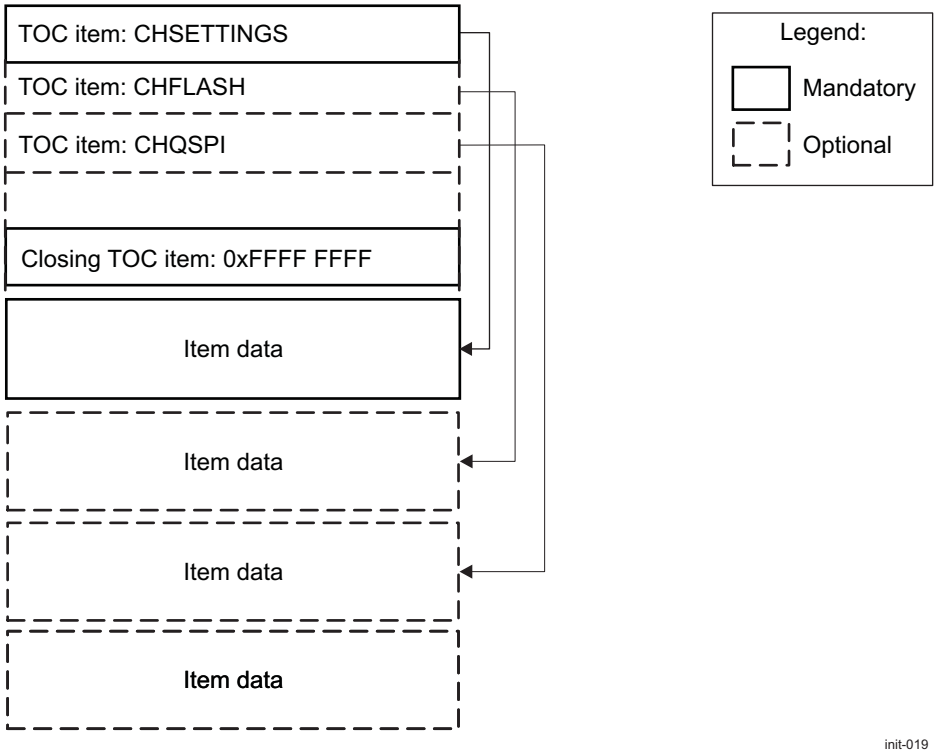


Table 25-22. CH TOC Item

Offset	Field	Size (Bytes)	Description
0x0000	Start	4	Offset from the start address of the TOC to the actual address of item contents
0x0004	Size	4	Size of item
0x0008	Reserved	4	Unused
0x000C	Reserved	4	Unused
0x0010	Reserved	4	Unused
0x0014	TOC Filename	12	12-character name of a TOC item, including the NULL termination character. That is, this is an array where first byte accomodates the first character.

The ROM Code recognizes sections pointed to by the TOC based on the filename as described in Table 25-23

- The X-LOADER, 2ND, or ULO section contains the Initial Software
- Optionally, TOC may contain CH sections. The CHSETTINGS is a mandatory section of CH and is used to recognize CH presence.

Table 25-23. TOC Filenames

Filename	Item Type	Usage	Description
ULO	Initial Software	UART Peripheral Booting	“UART LOader”
2ND	Initial Software	UART Peripheral Booting	“Secondary Loader”
X-LOADER	Initial Software	Other devices	“eXternal LOADER”
CHSETTINGS	Configuration Header	Memory and Peripheral Booting	Configuration Header General Setting Item
CHFLASH	Configuration Header	Memory and Peripheral Booting	Configuration Header GPMC Item
CHQSPI	Configuration Header	Memory and Peripheral Booting	Configuration Header QSPI Item

25.9.2.1 CHSETTINGS Item

The CHSETTINGS configuration header contains settings specific to the clock system. The ROM code configures the device clocking to some default settings as described in Section 25.6.4, *Clocking Configuration*. The CH CHSETTINGS section contains a method to override the ROM code default clock settings.

Table 25-24 describes the fields. The clocking setting structure are described in Table 25-25, *Clocking Settings*.

Table 25-24. CHSETTINGS Item

Offset	Field	Description
0000h	Section key	Key used for item verification: C0C0C0C1h
0004h	Valid	Enables/disables the section: 00h: Disable Others: Enable
0005h	Version	Configuration header version 01h Others: Reserved
0006h	Reserved	
0008h	Clocking settings	Described in Table 25-25, *Clocking Settings*.

Table 25-25. Clocking Settings

Field	Size (Bytes)	Description
Flags	4	Bit mask of various switches, active when set to 1: Bit [0]: Clock configuration defined in this structure is applied. Bit [1]: Reserved Bit [2]: Apply general clock settings. Bit [3]: Set and lock DPLL_PER. Bit [4]: Set and lock DPLL_MPU. Bit [5]: Set and lock DPLL_CORE. Bit [7]: Bypass DPLL_PER before setting clocks. Bit [8]: Bypass DPLL_MPU before setting clocks. Bit [9]: Bypass DPLL_CORE before setting clocks. Others: Reserved
General Clock Settings		
CM_CLKSEL_CORE	4	Register value
CM_BYPCLK_DPLL_MPU	4	Register value(reserved)
CM_BYPCLK_DPLL_IVA	4	Register value(reserved)
CM_MPU_MPU_CLKCTRL	4	Register value(reserved)

Table 25-25. Clocking Settings (continued)

Field	Size (Bytes)	Description
CM_CLKSEL_USB_60MHZ	4	Register value(reserved)
MPU DPLL Settings (reserved)		
CM_CLKMODE_DPLL_MPU	4	Register value
CM_AUTOIDLE_DPLL_MPU	4	Register value
CM_CLKSEL_DPLL_MPU	4	Register value
CM_DIV_M2_DPLL_MPU	4	Register value
Core DPLL Settings		
CM_CLKMODE_DPLL_CORE	4	Register value
CM_AUTOIDLE_DPLL_CORE	4	Register value
CM_CLKSEL_DPLL_CORE	4	Register value
CM_DIV_M2_DPLL_CORE	4	Register value
CM_DIV_M3_DPLL_CORE	4	Register value
CM_DIV_H11_DPLL_CORE	4	Register value
CM_DIV_H12_DPLL_CORE	4	Register value
CM_DIV_H13_DPLL_CORE	4	Register value
CM_DIV_H14_DPLL_CORE	4	Register value
CM_DIV_H21_DPLL_CORE	4	Register value
CM_DIV_H22_DPLL_CORE	4	Register value
CM_DIV_H23_DPLL_CORE	4	Register value
CM_DIV_H24_DPLL_CORE	4	Register value
PER DPLL Settings		
CM_CLKMODE_DPLL_PER	4	Register value
CM_AUTOIDLE_DPLL_PER	4	Register value
CM_CLKSEL_DPLL_PER	4	Register value
CM_DIV_M2_DPLL_PER	4	Register value
CM_DIV_M3_DPLL_PER	4	Register value
CM_DIV_H11_DPLL_PER	4	Register value
CM_DIV_H12_DPLL_PER	4	Register value
CM_DIV_H13_DPLL_PER	4	Register value
CM_DIV_H14_DPLL_PER	4	Register value
USB DPLL Settings (no function in this device)		
CM_CLKMODE_DPLL_USB	4	Register value
CM_AUTOIDLE_DPLL_USB	4	Register value
CM_CLKSEL_DPLL_USB	4	Register value
CM_DIV_M2_DPLL_USB	4	Register value

25.9.2.2 CHFLASH Item

The CHFLASH configuration header contains settings specific to the general-purpose memory controller (GPMC). For more information, see Section 10.3, *General-Purpose Memory Controller*. Table 25-26 describes the fields.

Table 25-26. CHFLASH Item

Offset	Field	Description
0000h	Section Key	Key used for section verification: C0C0C0C3h.
0004h	Valid	Enables/disables the section: 00h: Disable Others: Enable

Table 25-26. CHFLASH Item (continued)

Offset	Field	Description
0005h	Reserved	
0008h	GPMC_SYSCONFIG (LSW)	Register values
000Ah	GPMC_IRQENABLE (LSW)	
000Ch	GPMC_TIMEOUT_CONTROL (LSW)	
000Eh	GPMC_CONFIG (LSW)	
0010h	GPMC_CONFIG1_0	
0014h	GPMC_CONFIG2_0	
0018h	GPMC_CONFIG3_0	
001Ch	GPMC_CONFIG4_0	
0020h	GPMC_CONFIG5_0	
0024h	GPMC_CONFIG6_0	
0028h	GPMC_CONFIG7_0	
002Ch	GPMC_PREFETCH_CONFIG1	
0030h	GPMC_PREFETCH_CONFIG2 (LSW)	
0032h	GPMC_PREFETCH_CONTROL (LSW)	
0034h	GPMC_ECC_CONFIG (LSW)	
0036h	GPMC_ECC_CONTROL (LSW)	
0038h	GPMC_ECC_SIZE_CONFIG	
003Ch	Reserved	

25.9.2.3 CHQSPI Item

The CHQSPI configuration header contains settings specific to the QSPI interface controller. For more information, see Section 18.4, *Quad Serial Peripherial Interface*. Table 25-27 describes the fields.

Table 25-27. CHQSPI Item

Offset	Register Field Modified	Description
0000h	Section key	Key used for section verification C0C0C0C6h
0004h	Valid	Enables/disables the section: 00h: Disable Other: Enable
0005h	Reserved	
0008h	SPI Clock	SPI clock rate: 0x01: 64 MHz 0x03: 32 MHz 0x07: 16 MHz 0x13: 48 MHz 0x17: 24 MHz 0x1F: 12 MHz All others: reserved
0009h	RCMD	Read command
000Ah	READ_TYPE	Determines if the read command is a single, dual or quad read mode command
000Bh	NUM_A_BYTES	Number of address bytes to be sent
000Ch	NUM_D_BYTES	Number of dummy bytes to be used for fast read

25.9.3 GP Header

When the booting memory device is non-XIP (for example, SPI) the image must contain a small header, located before the executable code, and having the size of the software to load and the destination address of where to store it. Table 25-28 describes the image format. The GP header is not required when booting from an XIP memory device (for example, NOR) or in case of peripheral booting. In this case, the peripheral or memory booting image starts directly with executable code.

Table 25-28. GP Header Image Format

Field	Non-XIP Device Offset	XIP Device Offset	Size (Bytes)	Description
Size	0x0000	–	4	Size of the image (including GP header)
Destination	0x0004	–	4	Address where to store the code or code entry point
Image Code	0x0008	0x0000	x	Executable code

NOTE: The Destination address field stands for:

- Target address for the image copy from the non-XIP storage to the target XIP location (for example, internal RAM or SDRAM)
- Entry point for image code

Users must take care to locate the code entry point to the target address for image copy.

25.9.4 Image Execution

The image is executed when the ROM code performs the branch to the first executable instruction in the initial software. For a GP device in non-XIP, the execution address is the first word after the GP header. The branch is performed in public Arm supervisor mode. The R0 register points to the booting parameter structure that contains information about booting execution. Table 25-29 shows the booting parameter structure.

Table 25-29. Booting Parameter Structure

Offset	Field	Size (Bytes)	Description
0x00	Booting message	4	Last received booting message
0x04	Memory booting device descriptor	4	Pointer to the memory device descriptor that has been used during the memory booting process
0x08	Current booting device	1	Code of device used for booting: 0x01: XIP 0x02: XIP (with wait monitoring) 0x0A: QSPI_1 0x0B: QSPI_4 0x0C: QSPI_4(XIP) 0x43: UART Others: Reserved
0x09	Reset reason	1	Current reset reason bit mask (bit = 1, event present): direct copy from lower byte of PRM_RSTST (more bits exist in PRM_RSTST): [0]: Power-on (cold) reset [1]: Global software warm reset [2]: Reserved [3]: MPU watchdog reset [RESERVED] [4]: Reserved [5]: External warm reset

Table 25-29. Booting Parameter Structure (continued)

Offset	Field	Size (Bytes)	Description
			[6]: Reserved [7]: Reserved
0x0A	CH flags	1	Configuration header items flag. Each item is described by 1 bit. A set bit indicates that the item was executed: [0]: CHSETTINGS [2]: CHFLASH [4]: CHQSPI Other bits: Reserved

25.10 Tracing

Tracing in the public ROM code consists in 32-bit vectors for which each bit corresponds to a particular way point in the ROM code execution sequence. Table 25-30 through Table 25-32 list the organization of the tracing data in RAM. Tracing vectors are initialized at the beginning of the start-up phase and are updated all along the boot process.

There are two sets of tracing vectors (Table 25-12, *Tracing Data)*. The first set is the current trace information (after a cold or warm reset). The second set holds a copy of trace vectors collected at the first ROM code run after a cold reset. As a result, after a warm reset it is possible to have visibility on the boot scenario that occurred during a cold reset.

Table 25-30 lists the organization of tracing vector 1.

Table 25-30. Tracing Vector 1

Bit	Group	Meaning
0	Boot	Passed the reset vector
1	Boot	Entered main function
2	Boot	Running after the cold reset
3	Boot	Main booting routine entered
4	Memory boot	Memory booting started
5	Peripheral boot	Peripheral booting started
6	Boot	Booting loop reached last device
7	Boot	GP header found
8	Peripheral Boot	Booting message Skip peripheral booting received
9	Peripheral Boot	Booting message Change device received
10	Peripheral boot	Booting message Peripheral booting received
11	Peripheral boot	Booting message Get ASIC ID received
12	Peripheral boot	Device initialized
13	Peripheral boot	ASIC ID sent
14	Peripheral boot	Image received
15	Peripheral boot	Peripheral booting failed
16	Peripheral boot	Booting message not received (time-out)
17	Peripheral boot	Image size not received (time-out)
18	Peripheral boot	Image not received (time-out)
19	Reserved	
20	Boot	Configuration header found
21	Boot	CHSETTINGS item processed
22	Boot	Reserved
23	Boot	CHFLASH item processed
24	Boot	Reserved
25	Boot	Reserved
26	Boot	Reserved
27	Reserved	
28	Boot	SWCFG general detected
29	Boot	SWCFG clocks detected
30	Boot	SWCFG time-out detected
31	Reserved	

Table 25-31 lists the organization of tracing vector 2.

Table 25-31. Tracing Vector 2

Bit	Group	Meaning
0:7	Reserved	
8	Boot	CHQSPI item processed (clock speed)
9	Boot	CHQSPI item processed (Read command)
10:11	Reserved	Reserved
12	Memory boot	Memory booting trial (first block)
13	Memory boot	Memory booting trial (second block)
14	Memory boot	Memory booting trial (third block)
15	Memory boot	Memory booting trial (fourth block)
16:27	Reserved	
28:29	Reserved	
30	Boot	Jumping to Initial Software
31	Reserved	

Table 25-32 lists the organization of tracing vector 3.

Table 25-32. Tracing Vector 3

Bit	Group	Meaning
0	Reserved	
1	Memory boot	Memory booting device XIP
2	Memory boot	Memory booting device XIPWAIT
3	Memory boot	Reserved
4:7	Memory boot	Reserved
8:9	Memory boot	Reserved
10	Memory boot	Memory booting device: QSPI_1
11	Memory boot	Memory booting device: QSPI_4
12	Memory boot	Memory booting device: QSPI_4(XIP)
13:15	Reserved	
16:18	Reserved	
19	Peripheral boot	Peripheral booting device UART
20:31	Reserved	

On-Chip Debug Support

This chapter describes the on-chip debug support.

NOTE: The L3_MAIN interconnect is instantiation of the NoC interconnect from Arteris, Inc. Arteris is a registered trademark of Arteris, Inc.

This document contains materials that are ©Arteris, Inc., and that constitute proprietary information of Arteris, Inc.

All materials and trademarks are used under license from Arteris, Inc. For additional information, see the Arteris Reference manuals, or contact Arteris, Inc.

NoC is an abbreviation for Network On Chip.

Topic **Page**

26.1 Introduction

Debugging a system that contains an embedded processor involves an environment that connects high-level debugging software running on a host computer to a low-level debug interface supported by the target device. Between these levels, a debug and trace controller (DTC) facilitates communication between the host debugger and the debug support logic on the target chip.

The DTC is a combination of hardware and software that connects the host debugger to the target system. The DTC uses one or more hardware interfaces and/or protocols to convert actions dictated by the debugger user to JTAG® commands and scans that execute the core hardware.

The debug software and hardware components let the user control multiple central processing unit (CPU) cores embedded in the device in a global or local manner. This environment provides:

- Synchronized global starting and stopping of multiple processors
- Starting and stopping of an individual processor
- Each processor can generate triggers that can be used to alter the execution flow of other processors

System topics include but are not limited to:

- System clocking and power-down issues
- Interconnection of multiple devices
- Trigger channels

For easy integration into applications, a set of application-programming interfaces (APIs) for debug-IP programming and a software message library are provided. CToolsLib is a library of embedded target APIs to enable easy programmatic access to the chip tools (CTools), which are system-level debug facilities included in the debug subsystem capabilities of TI devices. More information about the APIs, download files, and other useful links for available libraries can be found on the CToolsLib Wiki site:

http://processors.wiki.ti.com/index.php/CToolsLib

The previous link connects to TI community resources. Linked contents are provided “AS IS” by the respective contributors. They do not constitute TI specifications and do not necessarily reflect TI's views; see TI's Terms of Use.

NOTE: TI also provides a Register Descriptor Tool (RDT). The RDT is a Java™- based stand-alone application. It is an interactive device register database configuration software. It allows the user to:

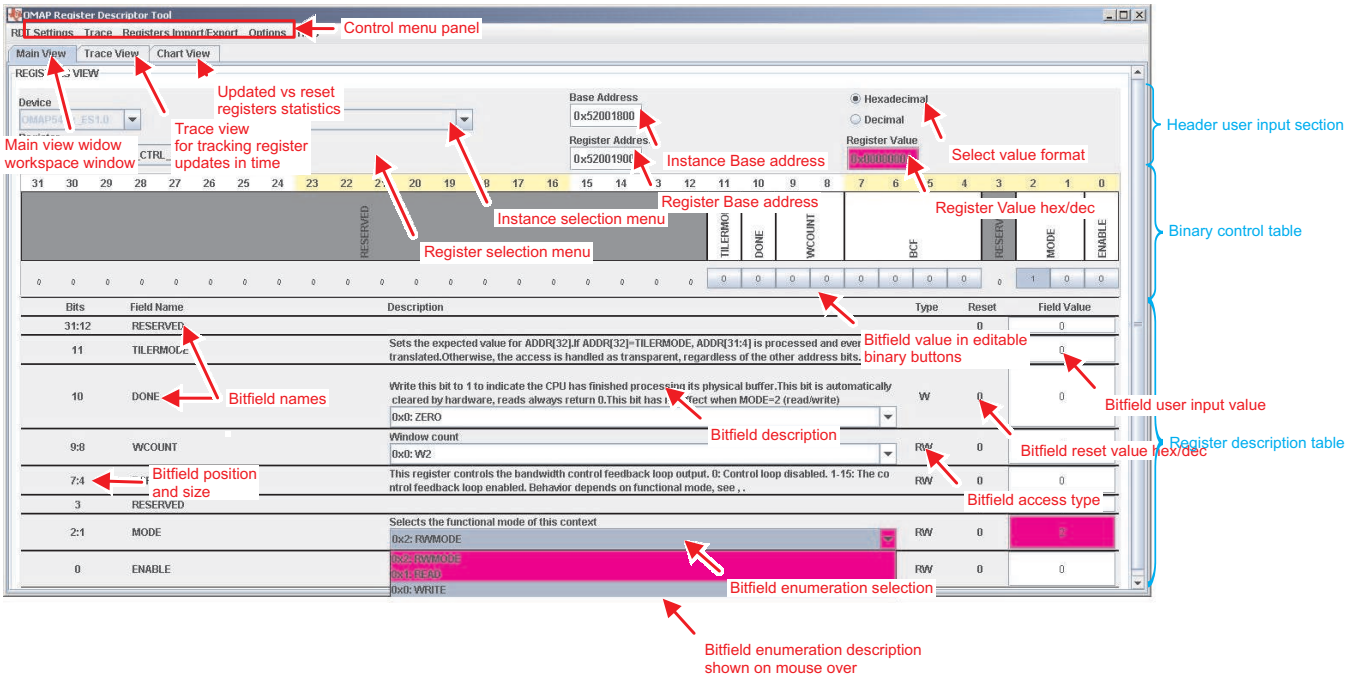
- Visualize device register settings
- Manipulate device registers easily
- Read in complete device configurations from various file formats into registers easily
- Dump out complete device configurations to various file formats
- Trace the register changes in time and during debugging
- See, understand, and work with device IPs, Instances, Registers, Bitfields, descriptions, and so on
- Extract and read in registers and register dumps for Code Composer Studio, and Lauterbach

The advantage of the tool is that the user can visualize the register state on power-on reset and then customize the configuration of the device for the specific use-case and identify the device register settings associated to that configuration.

Being an interactive visual tool, the RDT gives the user a global view of the device setting architecture and allows determining the exact register settings to obtain the specific configuration.

Figure 26-1 is a screenshot diagram of the RDT.

Figure 26-1. Register Descriptor Tool (RDT)



26.1.1 Key Features

The device deploys Texas Instrument's CTools debug technology for on-chip debug and trace support. It provides the following features:

- External debug interfaces:
  - Primary debug interface - IEEE1149.1 (JTAG)
    - Used for debugger connection
    - Controls ICEPick™ (generic test access port [TAP] for dynamic TAP insertion) to allow the debugger to access several debug resources through its secondary (output) JTAG ports (for more information, see Section 26.3.3.1, *ICEPick Secondary TAPs*).
    - For more information about IEEE1149.1, see Section 26.2.1, *IEEE1149.1*.
  - Debug (trace) port
    - Can be used to export processor or system trace off-chip (to an external trace receiver)
    - Can be used for cross-triggering with an external device
    - Configured through debug resources manager (DRM) module instantiated in the debug subsystem
    - For more information about debug (trace) port, see Section 26.2.2, *Debug (Trace) Port* , and Section 26.10, *Concurrent Debug Modes*.
- JTAG based processor debug on:
  - C66x in DSP1, DSP2
  - Cortex-M4 (x2) in IPU
  - ARP32 in EVE
- Dynamic TAP insertion
  - Controlled by ICEPick
  - For more information, see Section 26.3.3, *Dynamic TAP Insertion.*
- Power and clock management
  - Debugger can get the status of the power domain associated to each TAP.

  - Debugger may prevent the application software switching off the power domain.
  - Application power management behavior can be preserved during debug across power transitions.
  - For more information, see Section 26.6.1, *Power and Clock Management.*
- Reset management
  - Debugger can configure ICEPick to assert, block, or extend the reset of a given subsystem.
  - For more information, see Section 26.6.2, *Reset Management.*
- Cross-triggering
  - Provides a way to propagate debug (trigger) events from one processor, subsystem, or module to another:
    - Subsystem A can be programmed to generate a debug event, which can then be exported as a global trigger across the device.
    - Subsystem B can be programmed to be sensitive to the trigger line input and to generate an action on trigger detection.
  - Two global trigger lines are implemented
  - Device-level cross-triggering is handled by the XTRIG (TI cross-trigger) module implemented in the debug subsystem
  - For more information about cross-triggering, see Section 26.4.2, *Cross-Triggering.*
- Suspend
  - Provides a way to stop a closely coupled hardware process running on a peripheral module when the host processor enters debug state
  - For more information about suspend, see Section 26.4.3, *Suspend.*
- Processor trace
  - C66x (DSP) processor trace is supported
  - Two exclusive trace sinks:
    - CoreSight Trace Port Interface Unit (CS_TPIU) – trace export to an external trace receiver
    - CTools Trace Buffer Router (CT_TBR) in buffer mode – trace history store into on-chip trace buffer
  - For more information, see Section 26.8, *Processor Trace.*
- System instrumentation (trace)
  - Supported by a CTools System Trace Module (CT_STM), implementing MIPI System Trace Protocol (STP) (rev 2.0)
  - Real-time software trace
    - System-on-chip (SoC) software instrumentation through CT_STM (STP2.0)
  - OCP watchpoint (OCP_WP_NOC)
    - OCP target traffic monitoring: OCP_WP_NOC can be configured to generate a trigger upon watchpoint match (that is, when target transaction attributes match the user-defined attributes).
    - SoC events trace
    - DMA transfer profiling
  - Statistics collector (performance probes)
    - Computes traffic statistics within a user-defined window and periodically reports to the user through the CT_STM interface
    - Embedded in the L3_MAIN interconnect
    - 10 instances:
      - 1 instance dedicated to target (SDRAM) load monitoring
      - 9 instances dedicated to master latency monitoring
  - EVE instrumentation
    - Supported through a software message and system trace event (SMSET) module embedded in the EVE subsystem

  - ISS instrumentation (Note: ISS is not supported on the DRA78x family of devices)
    - Supported through system trace event (CTSET) module embedded in the ISS subsystem
  - Power-management events profiling (PM instrumentation [PMI])
    - Monitoring major power-management events. The PM state changes are handled as generic events and encapsulated in STP messages.
  - Clock-management events profiling (CM instrumentation [CMI]) for CM_CORE_AON clocks
    - Monitoring major clock management events. The CM state changes are handled as generic events and encapsulated in STP messages.
    - One instance:
      - CM1 Instrumentation (CMI1) module mapped in the PD_CORE_AON power domain
  - For more information, see Section 26.9, *System Instrumentation.*
- Performance monitoring
  - Supported by subsystem counter timer module (SCTM) for IPU
  - For more information, see Section 26.7, *Performance Monitoring*.

26.2 Debug Interfaces

26.2.1 IEEE1149.1

The target debug interface has the following signals:

- Five standard IEEE1149.1 JTAG signals: nTRST, TCK, TMS, TDI, and TDO
- Two EMU [1:0] TI extensions

Table 26-1 describes the IEEE1149.1 signals.

Table 26-1. IEEE1149.1 Signals

Device Pad Name	Internal Signal Name	Type(1)	Function	Description
trstn	nTRST	I	Test reset	When asserted (active low), resets all test and debug logic in the device along with the IEEE1149.1 interface.
tclk	TCK	I	Test clock	This is the test clock used to drive an IEEE1149.1 TAP state-machine and logic. This is either a free-running clock or a gated clock, depending on the DTC attached to the device and the RTCK monitoring.
rtck	RTCK	O	Returned (synchronized) test clock	Depending on the DTC attached to the device, the JTAG signals are either clocked from RTCK or the RTCK is monitored by the DTC to the gate TCK.
tms	TMS	I/O	Test mode select input	Directs the next state of the IEEE1149.1 TAP state-machine.
tdi	TDI	I	Test data input	Scans data input to the device.
tdo	TDO	O	Test data output	Scans data output by the device.
emu0	EMU0	I/O	Emulation 0	Channel 0 trigger or boot mode or trace port.
emu1	EMU1	I/O	Emulation 1	Channel 1 trigger or boot mode or trace port.

(1) I = Input; O = Output; I/O = bidirectional

NOTE: For more information about device pads pull type resistors, see the device Data Manual, Section Signal Descriptions in Chapter Terminal Configuration and Functions.

NOTE: The device JTAG ID code can be accessed through ICEPick. For information about the JTAG ID code value, see Chapter 1, *Introduction*.

26.2.2 Debug (Trace) Port

On-chip debug and trace events can be exported to external equipment through the debug (trace) port of the device. The following exportable debug events and trace sources are supported:

- Debug events
  - Triggers. For more information about triggers, see Section 26.4.2, *Cross-Triggering*.
- Trace sources
  - Processor trace: C66x DSP trace is supported. For more information about the processor trace, see Section 26.8, *Processor Trace*.
  - System trace: Trace coming from various system instrumentation modules, and supported by the CT_STM module. For more information about the system trace, see Section 26.9, *System Instrumentation*.

Note that not all debug and trace features can be supported concurrently because of the limited number of pins allocated to debug. Thus, multiplexing among debug and trace sources is implemented. The configuration and the debug/trace source selection occur through the DRM module embedded in the debug subsystem.

Table 26-2 describes the trace port signals.

Table 26-2. Trace Port Signals

Pin Name	Internal Signal Name	I/O[1]	Pull Type [2]	Description
emu19	EMU19	O	PD	Emulator pin 19
emu18	EMU18	O	PD	Emulator pin 18
emu17	EMU17	O	PD	Emulator pin 17
emu16	EMU16	O	PD	Emulator pin 16
emu15	EMU15	O	PD	Emulator pin 15
emu14	EMU14	O	PD	Emulator pin 14
emu13	EMU13	O	PD	Emulator pin 13
emu12	EMU12	O	PD	Emulator pin 12
emu11	EMU11	O	PD	Emulator pin 11
emu10	EMU10	O	PD	Emulator pin 10
emu9	EMU9	O	PD	Emulator pin 9
emu8	EMU8	O	PD	Emulator pin 8
emu7	EMU7	O	PD	Emulator pin 7
emu6	EMU6	O	PD	Emulator pin 6
emu5	EMU5	O	PD	Emulator pin 5
emu4	EMU4	O	PD	Emulator pin 4
emu3	EMU3	O	PD	Emulator pin 3
emu2	EMU2	O	PD	Emulator pin 2
emu1	EMU1	I/O	PU	Emulator pin 1
emu0	EMU0	I/O	PU	Emulator pin 0

(1) I = Input; O = Output; I/O = bidirectional
(2) PU = internal pullup; PD = internal pulldown

NOTE: The emu[19:0] pins are shared with other functional (application) pins on the device boundary. To use the emu[19:0] pins, the user must program the device application pin manager (Control Module) appropriately. For more information, see Chapter 13, *Control Module*.

For more information about DRM multiplexing and concurrent debug modes, see Section 26.10, *Concurrent Debug Modes*.

26.2.3 Trace Connector and Board Layout Considerations

Texas Instruments supports a variety of eXtended Development System (XDS) JTAG controllers with various debug capabilities beyond only JTAG support. Because this device supports the export of processor trace and system trace over the EMU pins, if you want your target to be compatible with XDS products capable of acquiring either trace types, see the following document for guidelines: *Emulation and Trace Headers* (literature number SPRU655). You can also find more information at the “XDS Target Connection Guide” TI wiki page:

http://processors.wiki.ti.com/index.php/XDS_Target_Connection_Guide

The previous link connects to TI community resources. Linked contents are provided “AS IS” by the respective contributors. They do not constitute TI specifications and do not necessarily reflect TI's views; see TI's Terms of Use.

26.3 Debugger Connection

26.3.1 ICEPick Module

The debugger connects to the device through its JTAG interface. The first level of debug interface seen by the debugger is the ICEPick module embedded in the debug subsystem.

NOTE: ICEPick version D (ICEPick-D) is used in the device.

SoC designs typically have multiple processors, each having a JTAG TAP embedded in the processor. The ICEPick module manages these TAPs and the power, reset, and clock controls for modules that have TAPs.

The ICEPick module is visible only from the debugger point of view and thus cannot be programmed by application software. The debugger can configure ICEPick through its own TAP controller. The ICEPick TAP has an instruction length of 6 bits and is the primary TAP. It is always visible in the scan chain and is used to control and monitor the other secondary TAPs.

ICEPick provides the following debug capabilities:

- Debug connect logic for enabling or disabling most ICEPick instructions
- Dynamic TAP insertion
  - Serially linking up to 32 TAP controllers
  - Individually selecting one or more of the TAPs for scan without disrupting the instruction register (IR) state of other TAPs
- Power, reset, and clock management (PRCM)
  - Provides the power and clock states of each domain
  - Provides debugger control of the power domain of a processor. Can force the domain power and clocks on, and prohibit the domain from being clock-gated or powered down while a debugger is connected.
  - Applies system reset
  - Provides wait-in-reset (WIR) boot mode
  - Provides global and local WIR release
  - Provides global and local reset blocking

The ICEPick module implements a connect register, which must be configured with a predefined key to enable the full set of JTAG instructions. Once the debug connect key is properly programmed, ICEPick signals and subsystems emulation logics should be turned on.

For more information about ICEPick dynamic TAP insertion, see Section 26.3.3, *Dynamic TAP Insertion*.

For more information about ICEPick PRCM features, see Section 26.6, *Power, Reset, and Clock Management Debug Support*.

26.3.2 ICEPick Boot Modes

The initial configuration of ICEPick is determined by the level of the EMU0 and EMU1 pins at POR release. At POR, EMU0 and EMU1 are automatically configured as inputs. The EMU0 and EMU1 pins are free when POR is released.

Table 26-3 summarizes the ICEPick boot modes.

Table 26-3. ICEPick Boot Modes at POR

EMU1	EMU0	TAPs in the TDI → TDO Path	Other Effects/Comments
0	0	None	Reserved (do not use)
0	1	None	Reserved (do not use)
1	0	ICEPick	TAP only + WIR mode
1	1	ICEPick	TAP only (default mode)

26.3.2.1 Default Boot Mode

In ICEPick-only configuration, none of the secondary TAPs are selected. The ICEPick TAP is the only TAP between device-level TDI and TDO pins. This mode is the recommended boot mode.

26.3.2.2 Wait-In-Reset

The device can boot to invoke WIR mode. If the device is booted in this mode, all processors within the device that support a TAP or non-JTAG debug cores through ICEPick are held in reset until released. Individual processors can be released from reset (local), or all processors held in the reset state can be released at the same time (global).

26.3.3 Dynamic TAP Insertion

26.3.3.1 ICEPick Secondary TAPs

To include more or fewer secondary TAPs in the scan chain, the debugger must use the ICEPick TAP router to program the TAPs. At its root, ICEPick is a scan-path linker that lets the DTC selectively choose which subsystem TAPs are accessible through the device-level debug interface. Each secondary TAP can be dynamically included in or excluded from the scan path. From the external JTAG interface point of view, secondary TAPs that are not selected appear not to exist.

Table 26-4 lists the secondary debug TAPs connected to the ICEPick scan chain along with the modules that can be accessed. The TAP number shows the position of the TAP in the scan chain.

Table 26-4. ICEPick Secondary Debug TAPs Mapping

Secondary JTAG Port	CoreSight	TAP Number	Modules Accessed Through That JTAG Port	
Debug Bank				
Reserved	No	0	–	
DSP1	No	1	C66x / ICEMaker	
Reserved	No	2	–	
Reserved	No	3	–	
IPU	No	4	Cortex-M4 / ICECrusher-CS	
	No	5	Cortex-M4 / ICECrusher-CS	
Reserved	No	6-9	–	
DSP2	No	10	C66x / ICEMaker	
Reserved	No	11-14	–	
CS_DAP (APB-AP)	No	15	Debug Subsystem	CT_TBR
	Yes			CS_TF_DEBUGSS
	Yes			CS_TPIU
	No			DRM
	No			CT_STM
	Yes			CS_CTI
CS_DAP (AHB-AP)	No		EVE	SCTM
	No			SMSET
	No			ARP32
	No		DSP1	ADTF1
	No		DSP2	ADTF2
	No		L3 NoC statistics collectors	All instances
	No		L3 OCP watchpoint	OCP_WP_NOC
	No		Clock management instrumentation	CMI
	No		Power management instrumentation	PMI
Test Bank				

Table 26-4. ICEPick Secondary Debug TAPs Mapping (continued)

Secondary JTAG Port	CoreSight	TAP Number	Modules Accessed Through That JTAG Port
DFT-SS	No	0	P1500 for DFT
Reserved	No	1	Reserved
System Control	No	2	P1500
Reserved	No	3	Reserved
Reserved	No	4	Reserved

For more information about ICEPick scan sequences (adding one or more TAPs to the scan chain), see:

http://processors.wiki.ti.com/images/f/f6/Router_Scan_Sequence-ICEpick-D.pdf

The preceding links connects to TI community resources. Linked contents are provided “AS IS” by the respective contributors. They do not constitute TI specifications and do not necessarily reflect the views of TI; see the TI Terms of Use.

Besides secondary debug TAPs, ICEPick also supports power, reset, and clock controls for non-JTAG debug cores. The debug cores are accessible through CS_DAP.

Table 26-5 summarizes the ICEPick debug core mapping.

Table 26-5. ICEPick Debug Core Mapping

Debug Core #	Module
0 – 8	Reserved
9	ARP32 in EVE
10 – 15	Reserved

26.4 Primary Debug Support

26.4.1 Processor Native Debug Support

26.4.1.1 Cortex-M4 Processor

The Cortex-M4 processor supports the following native debug features:

- Program halt and stepping
- Hardware breakpoints, breakpoint instruction
- Data watchpoint on access to data add, add range, and data value
- Register value accesses
- Debug monitor exception
- Memories accesses

For more information about Cortex-M4 native debug support features, see the *Cortex-M4 Technical Reference Manual*.

26.4.1.2 DSP C66x

The main components of the DSP subsystem are:

- TMS320C66x DSP core with an execution control and analysis module that uses TI's ICEMaker™ technology. The core includes the following debug capabilities:
  - Up to four hardware breakpoints and a breakpoint counter
  - Software breakpoints
  - Internally and externally generated triggers
  - Reset control
  - Emulation interrupt support
  - Modes for debugging without halting time-critical blocks of code
  - Options to protect user-selected blocks of code against debug activity
- Emulation memory access hardware. This hardware enables several methods for reading and writing memory and registers in the DSP subsystem during debugging.
- Advanced Event Triggering (AET) unit is used to generate debug actions for managing breakpoints, watchpoint, trace, timers/counters, event outputs to external logic of DSP, based on events detected by instruction and data bus comparators or by auxiliary event detectors. DSP's AET can also handle complex events with event state machine and event counters.

26.4.1.3 ARP32

The APR32 processor in the EVE subsystem offers the following debug capabilities:

- Manual halt
- Single-step execution
- Breakpoint
- Cross-triggering
- Global run

26.4.2 Cross-Triggering

The device supports a cross-triggering feature that provides a way to propagate debug (trigger) events from one processor subsystem/module to another. For example, a given subsystem A can be programmed to generate a debug event, which can then be exported as a global trigger across the device. Another subsystem B can be programmed to be sensitive to the trigger line input and to generate an action on trigger detection.

Examples of debug events are: Processor entering debug state, watchpoint match, and so forth.

Examples of debug actions are: Debug request generation, restart, interrupt request generation, start/stop trace, and so forth.

The device implements two global cross-triggering lines: Trigger0 and Trigger1, also referred to as EMU0 and EMU1, respectively.

Subsystem cross-triggering is consolidated at the device level by the XTRIG module, which is embedded in the debug subsystem.

NOTE: XTRIG is not programmatically visible from the JTAG interface or any device processor. Thus, cross-triggering is programmed at the subsystem level.

The Trigger0 and Trigger1 lines can also be configured as external triggers and contribute to cross-triggering.

26.4.2.1 SoC-Level Cross-Triggering

Device-level cross-triggering is handled by the XTRIG module. XTRIG manages two emulation triggers: Trigger0 and Trigger1. These trigger lines are shared by all the device subsystems implementing cross-triggering and are used to facilitate co-emulation.

Table 26-6 summarizes the device cross-triggering capabilities.

Table 26-6. Device Cross-Triggering

Subsystem	Module		Cortex-M4	Cortex-M4	DSP-C66x	DSP-C66x	EVE ARP32	EVE SMSET	ISS CMSET (1)	PMI	CMI	NOC_SC	OCP_WP_NOC	EMU0/EMU1
		Trigger Input	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
		Trigger Source												
IPU	Cortex-M4	✓		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	–
	Cortex-M4	✓	✓		✓	✓	✓	✓	✓	✓	✓	✓	✓	–
DSP1	C66x	✓	✓	✓		✓	✓	✓	✓	✓	✓	✓	✓	✓
DSP2	C66x	✓	✓	✓	✓		✓	✓	✓	✓	✓	✓	✓	✓
EVE	ARP32	✓	✓	✓	✓	✓		✓	✓	✓	✓	✓	✓	✓
	SMSET	✓	✓	✓	✓	✓	✓		✓	✓	✓	✓	✓	✓
ISS (1)	SMSET	–	–	–	–	–	–	–		–	–	–	–	–
PM	PMI	–	–	–	–	–	–	–	–		–	–	–	–
CM	CMI	–	–	–	–	–	–	–	–	–		–	–	–
SoC	NOC_SC	–	–	–	–	–	–	–	–	–	–		–	–
	OCP_WP_NOC	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓		✓
	EMU0, EMU1	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	

(1) ISS is not supported on the DRA78x family of devices.

26.4.2.2 Cross-Triggering With External Device

Based on DRM settings, the Trigger0 and Trigger1 lines may also be used as external triggers. Trigger0 is connected to the EMU0 device pin, and Trigger1 is connected to the EMU1 device pin. The user must make sure to program the DRM module in accordance with Table 26-27.

26.4.3 Suspend

The device supports a suspend feature, which provides a way to stop a closely coupled hardware process running on a peripheral-IP when the host processor enters debug state. The suspend mechanism is important for debug to ensure that peripheral-IPs operate in a lock-step manner with a host controller processor.

An entry is provided for each peripheral-IP that must consider the suspend signals from a number of processors (IPU or DSP). For each peripheral-IP, sensitivity to the suspend signals is defined within two possibilities (and so coded using 1 bit):

- Peripheral-IP is sensitive to the suspend line request.
- Peripheral-IP ignores the suspend line request.

For more information about how to program the sensitivity, see the corresponding peripheral-IP TRM chapter.

26.4.3.1 Debug Aware Peripherals and Host Processors

Table 26-7 lists the mapping of the device processors to the suspend control input lines.

Table 26-7. Debug Subsystem Suspend Input Lines

Suspend Input Line	Host Processor
0	DSP1
1	Reserved
2	Reserved
3	IPU1_C0
4	IPU1_C1
5	Reserved
6	Reserved
7	DSP2
8	Reserved
9	Reserved
10	EVE_ARP32
11 – 18	Reserved

Table 26-8 lists the mapping of the device peripheral-IPs to the suspend control output lines.

Table 26-8. Debug Subsystem Suspend Output Lines

Suspend Output Line	Peripheral-IP Module	DRM Suspend Control Register
0	DCC1[(1)]	DRM_SUSPEND_CTRL0
1	DCAN	DRM_SUSPEND_CTRL1
2	MCAN	DRM_SUSPEND_CTRL2
3	DCC2[(1)]	DRM_SUSPEND_CTRL3
4	DCC3[(1)]	DRM_SUSPEND_CTRL4
5	TIMER5	DRM_SUSPEND_CTRL5
6	TIMER6	DRM_SUSPEND_CTRL6
7	TIMER7	DRM_SUSPEND_CTRL7

(1) DCC and MCRC are not supported on the DRA78x family of devices.

Table 26-8. Debug Subsystem Suspend Output Lines (continued)

Suspend Output Line	Peripheral-IP Module	DRM Suspend Control Register
8	TIMER8	DRM_SUSPEND_CTRL8
9	DCC4(1)	DRM_SUSPEND_CTRL9
10	DCC5(1)	DRM_SUSPEND_CTRL10
11	DCC6(1)	DRM_SUSPEND_CTRL11
12	DCC7(1)	DRM_SUSPEND_CTRL12
13	PWMSS1	DRM_SUSPEND_CTRL13
14	Reserved	DRM_SUSPEND_CTRL14
15	MCRC(1)	DRM_SUSPEND_CTRL15
16	TIMER1	DRM_SUSPEND_CTRL16
17	TIMER2	DRM_SUSPEND_CTRL17
18	TIMER3	DRM_SUSPEND_CTRL18
19	TIMER4	DRM_SUSPEND_CTRL19
20	RTI1	DRM_SUSPEND_CTRL20
21	RTI2	DRM_SUSPEND_CTRL21
22	RTI3	DRM_SUSPEND_CTRL22
23	RTI4	DRM_SUSPEND_CTRL23
24	RTI5	DRM_SUSPEND_CTRL24
25	I2C	DRM_SUSPEND_CTRL25
26	Reserved	DRM_SUSPEND_CTRL26
27	Reserved	DRM_SUSPEND_CTRL27
28	Reserved	DRM_SUSPEND_CTRL28
29	Reserved	DRM_SUSPEND_CTRL29
30	Reserved	DRM_SUSPEND_CTRL30
31	COUNTER_32K	DRM_SUSPEND_CTRL31
32	Statistics collector	DRM_SUSPEND_CTRL32

NOTE: Table 26-8 lists only peripherals that support the suspend feature. For modules not listed in this table, the suspend feature is not supported.

26.5 Real-Time Debug

26.5.1 Real-Time Debug Events

26.5.1.1 Emulation Interrupts

A few device interrupt channels are dedicated to debug support. Table 26-9 summarizes the emulation interrupt events.

Table 26-9. Emulation Interrupts

Interrupt Request	Subsystem	Source	Description
RTOS_INT	DSP	DSP ICEMaker	RTOS interrupt from analysis event
DLOG_INT			Data logging transfer complete
EMUINT2			Monitor mode debug
ICNEMUINTR	IPU	Cortex-M4 ICECrusher-CS	Debug Control & Status register; Emulation trigger
ICNCOMMRX			Debug Communication Channel
ICNCOMMTX			
ICNEMUINTR		Cortex-M4 ICECrusher-CS	Debug Control & Status register; Emulation trigger
ICNCOMMRX			Debug Communication Channel
ICNCOMMTX			
SC_ALERT	L3_MAIN	Statistics collectors	Performance monitoring alert (metric out of range)
DCC_UART_TX	DEBUGSS	CT_UART (BDX client)	Transmit empty
DCC_UART_RX			Receiver data available

26.6 Power, Reset, and Clock Management Debug Support

The global PRCM module implements facilities to support debug across power and clock domain cycles. The debugger can control or get the status of each power and clock domain associated with an ICEPick secondary TAP.

ICEPick provides a set of directives allowing the debugger to:

- Get visibility on the associated power and clock domains state. This includes:
  - Current power setting indicating whether the power domain is on or off
  - Loss of power detected since the software last checked the status
  - Current clock setting indicating whether the clock domain is on or off
  - Sleep desired (PM and CM indicate that the debug settings in ICEPick are changing the application state. If it were not for the ICEPick controls, the power or clock would be turned off.)
  - Subsystem reset state
  - Subsystem has entered a debug state that requires the attention of the host debug software.
- Override power/clock control settings to wake up a power or clock domain or to prevent a power or clock domain from going to sleep once it is in ACTIVE state
- Assert/block/extend reset; release from extended reset (WIR)

26.6.1 Power and Clock Management

26.6.1.1 Power and Clock Control Override From Debugger

The debugger can override the application software power and clock management settings through the ICEPick module. It can configure ICEPick to force a domain active or prevent it from going to sleep once it is active. This can be achieved through the following debugger directives:

- FORCEACTIVE
- INHIBITSLEEP

26.6.1.1.1 Debugger Directives

26.6.1.1.1.1 FORCEACTIVE Debugger Directive

To ensure that the subsystem debug registers can always be accessed, regardless of the application power-management scenarios, a FORCEACTIVE directive can be issued through the debugger along the entire debug session. From an application standpoint, the system state, status, and timing are preserved, but the subsystem power domain is never shut down. Therefore, the emulation setup is preserved across power transitions, regardless of where the debug hardware is implemented.

If the debugger connects and the subsystem were previously powered down, a FORCEACTIVE directive wakes up the system and allows the debugger to take control of the system.

26.6.1.1.1.2 INHIBITSLEEP Debugger Directive

The debugger can use the INHIBITSLEEP directive to keep a subsystem powered and clocked, even if the applicative settings request that this subsystem go to sleep. Contrary to the FORCEACTIVE directive, the INHIBITSLEEP command does not wake up a subsystem that is already powered down by the application.

The typical use of the INHIBITSLEEP directive is to prevent power and clock transitions on the subsystem during a debug session. In this situation, when the applicative scenario initiates a transition, the transition does not take place, but from an applicative point of view the subsystem is not accessible.

26.6.1.1.2 Intrusive Debug Model

The use of debugger directives is intrusive from the standpoint of the power and clock controls, because they affect the power-management behavior of the application.

26.6.1.2 Debug Across Power Transition

26.6.1.2.1 Nonintrusive Debug Model

To preserve the power-management behavior of the device and allow the subsystem power and clock to be switched off by application software, the subsystem TAP must be disconnected from the ICEPick scan chain. The subsystem is then completely ignored by the debugger and the host-to-target communication is no longer affected by the state of the subsystem power and clocks. The debugger can still be informed that the disconnected subsystem entered the debug state by polling the Debug Attention status bit from ICEPick. The debugger can then insert the TAP, take control of the subsystem power and clock, and examine the system state.

This debug model is nonintrusive, because it allows the power-management behavior of the application to be preserved.

26.6.1.2.2 Debug Context Save and Restore

26.6.1.2.2.1 Debug Context Save

The device partitioning is such that not all the debug components are mapped to an always-on domain. Typically, the programmer wants the debug setup to be preserved along a debug session, including subsystem power cycling. When debug registers are memory mapped and not implemented within the emulation power domain, the application software must save the state of the debug registers before going to sleep and restore them upon wakeup.

26.6.1.2.2.2 Debug Context Restore

When the application software performs a context restore, it must be able to write to all the debug registers to restore their contents, regardless of the previous ownership. After subsystem power domain ramp up, the debug resources are in the available state, and ownership is restored. The debug context save and restore sequences are protected. All debug functionality is disabled and debugger accesses are blocked.

26.6.2 Reset Management

The debugger can take control of the system reset for each subsystem through ICEPick. The debugger can configure ICEPick to assert, block, or extend the subsystem reset.

26.6.2.1 Debugger Directives

26.6.2.1.1 Assert Reset

The debugger can program ICEPick to generate a subsystem reset request to the device reset management module. The debugger reset event is then merged with system reset events.

26.6.2.1.2 Block Reset

The debugger can program ICEPick to request the device reset management module to block an unsafe application system or subsystem reset event.

A reset originated by some safe reset sources cannot be blocked by the debugger.

26.6.2.1.3 Wait-In-Reset

WIR mode is latched from boot mode (see Section 26.3.2, *Boot Modes*) and allows the user to hold a secondary TAP module in reset state when a reset is applied (and thus, extend the reset). This mode lets the user control the following system level activities:

- Gain emulation control of any processor in a power domain at POR
- Capture and extend system-generated functional resets while running (under emulation control or not)
- Hold an entire power domain in reset until emulation control of the subsystem can be established

- Stall the entire system while reset is extended to a power domain
- Reset extension is visible external to the power domain.
- Debug execution of code from the first cycle of execution
- Prevent processor execution of random instructions in uninitialized program memory at power up
- Download code before any code execution takes place
- Coordinate debug initialization across multiple cores before code execution begins

WIR mode extends only the processor reset. During reset extension, the debugger can still access modules such as L2 memory and MMU, even if they are embedded in the device subsystem affected by WIR.

When the debugger task is complete, the DTC releases the subsystem reset by programming the corresponding ICEPick TAP control register.

26.7 Performance Monitoring

26.7.1 IPU Subsystem Performance Monitoring

26.7.1.1 Subsystem Counter Timer Module

The IPU subsystem includes a subsystem counter timer module (SCTM), which is embedded in shared cache and provides additional data to the user timing or profiling capability. SCTM integrates eight profiling counters that collect:

- Forty-four shared cache events
- Four sleep/deep-sleep events from Cortex-M4 cores (one sleep and one deep sleep event per core)

Table 26-10 describes the repartition of the IPU SCTM counters.

Table 26-10. IPU SCTM Counters Repartition

Counters	Features
0–1	Timer and event
2–3	64-bit chained + shadowing
4–5	64-bit chained + shadowing
6–7	Event

26.7.1.2 Cache Events

Table 26-11 summarizes the SCTM events for the IPU subsystem.

Table 26-11. SCTM Events for IPU Subsystem

Input Index	Event Description
1	Cache locks
2	Cache line replacements
3	Cache evictions
4	Cache maintenance operations (slave 0)
5	Cache maintenance operations (slave 1)
6	Cache maintenance operations (slave 2)
7	Cache maintenance operations (slave 3)
8	Cache OCP access (slave 0)
9	Cache OCP access (slave 1)
10	Cache OCP access (slave 2)
11	Cache OCP access (slave 3)
12	Cacheable access (slave 0)
13	Cacheable access (slave 1)
14	Cacheable access (slave 2)
15	Cacheable access (slave 3)
16	Cache bank conflicts (slave 0)
17	Cache bank conflicts (slave 1)
18	Cache bank conflicts (slave 2)
19	Cache bank conflicts (slave 3)
20	Cache allocations
21	Cache write buffer accesses (slave 0)
22	Cache write buffer accesses (slave 1)
23	Cache write buffer accesses (slave 2)
24	Cache write buffer accesses (slave 3)

Table 26-11. SCTM Events for IPU Subsystem (continued)

Input Index	Event Description
25	Cache line fills (slave 0)
26	Cache line fills (slave 1)
27	Cache line fills (slave 2)
28	Cache line fills (slave 3)
29	Cache write fills (slave 0)
30	Cache write fills (slave 1)
31	Cache write fills (slave 2)
32	Cache write fills (slave 3)
33	Cache read fills (slave 0)
34	Cache read fills (slave 1)
35	Cache read fills (slave 2)
36	Cache read fills (slave 3)
37	Cache misses (slave 0)
38	Cache misses (slave 1)
39	Cache misses (slave 2)
40	Cache misses (slave 3)
41	Cache hits (slave 0)
42	Cache hits (slave 1)
43	Cache hits (slave 2)
44	Cache hits (slave 3)
45	IPU_C1 deep sleep
46	IPU_C1 sleep
47	IPU_C0 deep sleep
48	IPU_C0 sleep

NOTE: Input index [0] is reserved for free-running subsystem clock (used for total cycle profiling).

26.7.2 DSP Subsystem Performance Monitoring

26.7.2.1 Advanced Event Triggering

The Advanced Event Trigger (AET) unit has the capability to generate the debug actions based on events detected by instruction and data bus comparators or by auxiliary event detectors to manage:

- Hardware Program Breakpoints: specify addresses or address ranges that can generate events such as halting the processor or triggering the trace capture.
- Data Watchpoints: specify data variable address, address ranges, or data values that can generate events such as halting the processor or triggering the trace captures.
- Counters: count the occurrence of an event of cycles for performance monitoring.
- State Sequencing: allows combinations of hardware program breakpoint and data Watchpoints to precisely generate events for complex sequences.

For more information on AET, see the following documents:

- *Using Advanced Event triggering to Find and Fix Intermittent Real-Time Bugs* application report (SPRA753).
- *Using Advanced Event Triggering to Debug Real-Time Problems in High Speed Embedded Microprocessor System* application report (SPRA387)

26.7.3 EVE Subsystem Performance Monitoring

26.7.3.1 EVE Subsystem Counter Timer Module

The EVE subsystem includes an SCTM, which is embedded in the EVE subsystem and provides additional data to the user timing or profiling capability. SCTM integrates eight profiling counters that collect:

- ARP32 program cache events
- EDMA events
- Interrupt events
- VCOP processor events

Table 26-12 describes the configuration of the EVE SCTM counters.

Table 26-12. EVE SCTM Counters Configuration

Counters	Features
0–1	Timer and event
2–3	64-bit chained + shadowing
4–5	64-bit chained + shadowing
6–7	Event

26.7.3.2 EVE Subsystem SCTM Events

Table 26-13 summaries the SCTM events for EVE subsystem.

Table 26-13. SCTM Events for EVE Subsystem

Input Index	Event Description
0	Cache miss
1	Cache hit
2	Cache miss stall
3	Prefetch compulsory count
4	Prefetch look ahead count
5	Prefetch hit count
6	Prefetch flush line count
7	Prefetch flush occur count
8	Prefetch discard stall
9	TPCC AET
10	ARP32 Interrupt 4
11	ARP32 Interrupt 5
12	ARP32 Interrupt 6
13	ARP32 Interrupt 7
14	VCOP busy
15	VCOP idle and done
16	VCOP wait for ARP32
17	VCOP ARP32 awaits
18	VCOP overhead
19	VCOP ld_stall_by_st
20	VCOP op_stall_by_ldst
21	VCOP op_stall_by_dependency
22	VCOP rd_ibufl
23	VCOP rd_ibufh

Table 26-13. SCTM Events for EVE Subsystem (continued)

Input Index	Event Description
24	VCOP rd_wbuf
25	VCOP wr_ibufl
26	VCOP wr_ibufh
27	VCOP wr_wbuf
28	VCOP loop_start
29	VCOP done

26.8 Processor Trace

The device supports DSP (C66x) processor trace for each DSP subsystem.

26.8.1 DSP Processor Trace

Trace targets the debug of unstable code, performance analysis, and quality assurance. This infrastructure component use bus snoopers to collect and export trace data using hardware dedicated to the trace function. The use of dedicated hardware to both collect, buffer, transfer trace data to the host makes trace non-intrusive. DSP processor trace characteristics are:

- Program trace
- Data trace (address and value)
- Time stamp
- Two exclusive trace sinks:
  - CS_TPIU – trace exported to an external trace receiver
  - CT_TBR – trace history stored into on-chip buffer

The trace function can collect and export a record of the program flow and timing at the same rate generated by the CPU. Tracing data references must be restricted however as the export mechanism is generally limited to size of on-chip CT_TBR to sustain tracing of all memory references. In the case of data trace, the Advanced Event Triggering facilities provide a means to restrict the trace data exported to data of interest to maintain the non-intrusive aspect of trace. This reduces the export bandwidth needs and facilitates the successful collection of the data of interest. Error indications are embedded in the debug stream in the event the export logic is unable to keep up with the data rate generated by the collection logic. This notification allows the user to scale back the amount of requested data collection.

The user can optionally select the export of all specified trace data. In this case the CPU is stalled to avoid the loss of trace data, with trace becoming intrusive to the application if trace related stalls are generated. The user is notified that trace stalls.

26.8.2 Trace Export

The debug subsystem implements two exclusive trace sinks:

- CS_TPIU
- CT_TBR

26.8.2.1 Trace Exported to External Trace Receiver

Processor trace can be exported to an external trace receiver through the CS_TPIU module. To achieve this, the debugger or application software must ensure to program the DRM module properly (that is, according to Table 26-27).

The CS_TPIU has a configurable export width of maximum 16 data pins (TRACEDATA) plus a dedicated export clock (TRACECLK) and a control signal (TRACECTL).

26.8.2.2 Trace Captured Into On-Chip Trace Buffer

Processor trace flow can be redirected to the on-chip trace buffer (CT_TBR working in buffer mode). The CT_TBR provides on-chip storage of trace data using a 32-KiB RAM memory.

The CT_TBR receives trace flow data through its ATB port. The debugger can then access trace data through CS_DAP (APB port).

The CT_TBR supports two modes when in buffer mode configuration:

- Circular buffer mode for continuous capture. In this mode, the CT_TBR functions as a circular buffer where the oldest data is overwritten with the most recent entries.
- Stop-on-full mode for no-loss (single shot) capture. In this mode, the buffer fills linearly and then halts when the last entry is written.

26.9 System Instrumentation

The device supports the following system instrumentation features:

- Real-time software trace
  - SoC software instrumentation via CT_STM (STP2.0) (see Section 26.9.3.1, *SoC Software Instrumentation*)
- OCP watchpoint
  - OCP target traffic monitoring (see Section 26.9.4.1, *OCP Target Traffic Monitoring*)
  - SoC events trace (see Section 26.9.4.2, *Messages Triggered from System Events*)
  - DMA transfer profiling (see Section 26.9.4.3, *DMA Transfer Profiling*)
- Statistics collector
  - L3 Master Latency Monitoring (see Section 26.9.7.1, *L3 Master Latency Monitoring*)
- PM events trace (see Section 26.9.8, *PM Instrumentation* [PMI])
- CM events trace (see Section 26.9.9, *CM Instrumentation* [CMI])

26.9.1 MIPI STM (CT_STM)

CT_STM is a trace module that aids in software debugging. The main features of this module are:

- Implements MIPI STP protocol (rev 2.0) with the following characteristics:
  - Highly optimized for software-generated traces
  - Automatic timestamping of messages
  - Support for 8-, 16-, and 32-bit data types
- Collects the following information:
  - Software messages
  - Hardware instrumentation trace from hardware agents:
    - OCP_WP_NOC
    - PMI
    - CMI
    - EVE SMSET
    - ISS CTSET (Note: ISS is not supported on the DRA78x family of devices)
    - L3 NoC statistics collectors
- Supports the following trace export paths:
  - ATB export – to CS_TPIU or CT_TBR
- Timestamps:
  - Can use local relative timestamp if exported to the CS_TPIU/CT_TBR (through ATB)
- Dedicated 128 × 32-bit FIFO buffer
- Mechanism to generate repeat MASTER and C8 messages periodically even if not needed

A maximum of 255 different bus masters can be connected to the STM trace port through a bus arbiter. STP recognizes two distinct modes of tracing (software and hardware types), which use slightly different message combinations to output different types of data. The bus masters can be configured for either type to optimize the system for the different types of trace data.

26.9.2 System Trace Export

26.9.2.1 CT_STM ATB Export

If a trace receiver cannot be attached to the device, or relevant CT_STM trace port pins are unavailable for a particular reason, the user can configure the CT_STM module to redirect the STP trace stream as an ATB stream to the CS_TPIU or CT_TBR and enable local timestamp. This is accomplished by outputting a local timestamp granularity (LTSG) message, which is a TI addition to the MIPI standard messages.

NOTE: For CT_STM timestamps when output to the ATB interface, the timestamp (TS) field is no longer the CT_STM FIFO depth, but is the time from the previous message.

Granularity is a function of the instrumentation port clock frequency.

26.9.2.2 Trace Streams Interleaving

Two levels of interleaving system instrumentation flows and arbitrating between instrumentation masters are implemented:

- CORE L3 instrumentation interconnect interleaves data coming from the following bus masters:
  - OCP_WP_NOC
  - L3 NoC statistics collectors, or software instrumentation (interleaving at the L3 level)
  - EVE SMSET and ARP32 (software instrumentation)
- EMU L3 instrumentation interconnect interleaves data coming from the following bus masters:
  - CORE L3 instrumentation interconnect
  - CMI
  - PMI

26.9.3 Software Instrumentation

26.9.3.1 SoC Software Instrumentation

The CT_STM module embedded in the debug subsystem provides a flexible interface for trace instrumentation.

The device provides support for real-time software trace through user-defined message writes to specific memory mapped register (MMR) locations. Software masters can transmit trace data from the operating system (OS) processes or tasks on 256 different channels, with each channel being defined by the software protocol implemented. The different channels can be used to group different types of data logically so that it is easy to filter out the data irrelevant to the ongoing debugging task. The message structures in STP-2.0 are optimized to provide an efficient transport for software data through the CT_STM module.

The software masters are:

- DAP (for testing purpose)
- DSP subsystem
- IPU subsystem
- EVE subsystem
- EDMA TPTC WR1 and WR2 channels

Each software master has a master-ID assigned to it (see Section 26.9.10, *Master-ID Encoding*, for more information).

Software messages can be interleaved with other hardware messages.

Software messages are intrusive and use both processor cycles and memory.

26.9.4 OCP Watchpoint

26.9.4.1 OCP Target Traffic Monitoring

The L3_MAIN interconnect provides several functional probes embedded and attached to the following L3_MAIN targets:

- GPMC
- L4_PER1, L4_PER2, L4_PER3

- L4_CFG
- L4_WKUP
- OCMC_RAM
- EMIF

The probes output are muxed together and then sent to the L3_MAIN interconnect debug port. A component called OCP_WP_NOC is used to collect data from functional probes and then transmit captured data to the CT_STM module. The OCP_WP_NOC drives a probe-ID signal to the L3_MAIN interconnect for probe selection. The probe selection is exclusive, meaning that interleaving is not possible.

The OCP_WP_NOC provides the following main features:

- Monitoring the OCP traffic originated by all initiators that can access the selected target where the probe is attached
- Filtering OCP monitored bus traffic by:
  - Address range
  - Initiator-ID (see Table 26-15)
  - Transaction type
  - Transaction qualifier
- Generating a trigger upon watchpoint match
- Starting and stopping OCP traffic monitoring upon:
  - WP address match
  - External trigger
- Profiling DMA transfers
- Generating hardware message upon system event
- OCP_WP_NOC messages can be interleaved with software messages
- Programming from:
  - Debugger
  - Application

NOTE: The OCP_WP_NOC is restricted to monitor request flow only.

Table 26-14 summarizes the OCP targets that can be monitored by the OCP_WP_NOC and their respective probe-ID.

Table 26-14. L3_MAIN Interconnect Functional Probe Mapping

Probe-ID	L3 OCP Target
0000	Reserved
0001	GPMC
0010	L4_PER1
0011	L4_CFG
0100	EMIF
0101	L4_WKUP
0110	OCMC_RAM
0111	L4_PER2
1000	L4_PER3

The user can program the OCP_WP to extract the traffic from a specific set of initiators (maximum four master-IDs). Table 26-15 lists the master-ID reported by the L3_MAIN debug port for the device initiators.

Table 26-15. Master-ID Mapping (Debug View)

Initiator	Debug View (Hex) – 6-bit	Debug View (Hex) – 7-bit
CS_DAP	4	8
IEEE1500_2_OCP	5	A
DSP1 MDMA	8	10
DSP1 CFG	9	12
DSP1 DMA	A	14
DSP2 DMA	B	16
DSP2 CFG	C	18
DSP2 MDMA	D	1A
EVE P1	10	21
IPU	18	30
EDMA_TC1 WR	1C	38
EDMA_TC1 RD	1C	39
EDMA_TC2 WR	1D	3A
EDMA_TC2 RD	1D	3B
DSS	20	40
MMU	21	43
VIP1 P1	24	48
VIP1 P2	24	49
GMAC_SW	2B	56
ISS RT(1)	30	60
ISS NRT1(1)	31	62
ISS NRT2(1)	32	64
EVE P2	34	69

(1) ISS is not supported on the DRA78x family of devices.

26.9.4.2 Messages Triggered from System Events

The OCP_WP_NOC can be programmed to export a message through the CT_STM upon detection of a system event (interrupt, DMA request, etc.). A bus of 16 system events pre-selected at SoC level from a large number of observable events is routed to the OCP_WP_NOC. This can be useful to determine interrupts latencies:

- Interrupt request traced through OCP_WP_NOC system event detection
- ISR boundary tracked either through CT_STM software trace or OCP_WP_NOC address range detection

26.9.4.3 DMA Transfer Profiling

The OCP_WP_NOC can be configured to profile DMA transfers. This feature provides to the user visibility on:

- DMA logical channel interleaving
- DMA channel ID
- DMA transfer duration (time stamp)
- DMA burst size
- DMA reads
- DMA writes

When operating in this mode, the OCP_WP_NOC exports to CT_STM:

- DMA channel ID
- Burst size

- R/W

The CT_STM module encapsulates the information above in a compact STP message.

The address range filtering remains active but OCP address data are not encapsulated into the trace message to maximize CT_STM throughput.

26.9.5 EVE SMSET

The device takes advantage of the system trace infrastructure to provide visibility to the user regarding EVE VCOP micro-task sequencing. This is supported through a SMSET module instantiated in the EVE subsystem. The VCOP start and done events are handled as generic events and encapsulated in STP messages with an event-ID and local timestamp and exported through the CT_STM module.

The EVE instrumentation scheme allows the user to understand micro-task dependencies, and potential bottlenecks. DMA transfer boundaries are reported as EVE events. Software messages from ARP32 execution can be interleaved with EVE VCOP events.

26.9.6 ISS Hardware Instrumentation

NOTE: ISS is not supported on the DRA78x family of devices.

The device takes advantage of the system trace infrastructure to provide visibility to the user regarding ISS key system events. The events are encapsulated in STP messages with an Event-ID and Global Time Stamp and are exported through the CTSET module.

The ISS instrumentation setup allows programming a sampling window. All the events detected within the user-defined sampling window will be reported with a common time-stamp.

The ISS instrumentation can be exported to a trace receiver or stored into an on-chip buffer.

26.9.7 L3 NOC Statistics Collector

The L3_MAIN interconnect supports a built-in performance monitoring feature by implementing a statistics collector (NOC_SC) component, which computes traffic statistics within a user-defined window and periodically reports to the user through the CT_STM interface. The following NOC_SC instances are instantiated in the device:

- Four statistics collectors dedicated to L3 Master Latency Monitoring – STAT_COLL1 through STAT_COLL4 (see Section 26.9.7.1, *L3 Master Latency Monitoring* , for more information)

Statistics collectors can report:

- Average burst length in bytes/packet per sampling window
- Average throughput in bytes/cycle
- Percent link occupancy on the request link (for store transactions) during a sampling window
- Percent link occupancy on the response link (for load transactions) during a sampling window
- Percent arbitration conflict cycles on the request link
- Percent initiator busy cycles on the response link
- Histogram of payload length in bytes (for example, 0–16, 16–32, 32–128) each sampling window.

The performance metrics are interleaved with software instrumentation data at the L3_MAIN interconnect level.

The performance monitoring probes implement three main functions:

- Events detection
- Transactions filtering
- Aggregation

The probes can be configured to detect the NTTP and OCP link events summarized in Table 26-16.

Table 26-16. Performance Monitoring Events Detection

Link Event	Definition
NONE	No event selected
ANY	Any clock cycles
TRANSFER	Word has been accepted by the receiver.
WAIT	Transfer has been initiated but the transmitter currently has no data to send.
BUSY	Receiver applies flow control
PKT	Transfer of a new packet header
DATA	Transfer of a payload word
IDLES	No communication over the link
LATENCY	Debug bit detection

The probes can be configured to filter the traffic based on the criteria summarized in Table 26-17.

Table 26-17. Performance Filtering Options

Filters	Comment
Master address	Mask and match
Slave address	
UserInfo	
Read	Opcode is a load
Write	Opcode is a store
Error	Mask and match

Table 26-18 specifies the master address mapping (all statistics collectors).

Table 26-18. Statistics Collector Master Address Mapping

Initiator	Debug View (Hex) – 8-bit
CS_DAP	10
IEEE1500_2_OCP	14
DSP1 MDMA	20
DSP1 CFG	24
DSP1 DMA	28
DSP2 DMA	2C
DSP2 CFG	30
DSP2 MDMA	34
EVE P1	42
IPU	60
EDMA_TC1 WR	70
EDMA_TC1 RD	72
EDMA_TC2 WR	74
EDMA_TC2 RD	76
DSS	80
MMU	86
VIP1 P1	90
VIP1 P2	92
GMAC_SW	AC
ISS RT(1)	C0
ISS NRT1(1)	C4

(1) ISS is not supported on the DRA78x family of devices.

Table 26-18. Statistics Collector Master Address Mapping (continued)

Initiator	Debug View (Hex) – 8-bit
ISS NRT2[1]	C8
EVE P2	D2

Table 26-19 specifies the slave address mapping.

Table 26-19. Statistics Collector Slave Address Mapping

Slave	Address (hex)
EMIF P1	2
TESOC[1]	3
DSP1 SDMA	4
DSP2 SDMA	5
DSS	6
EVE	7
CRC[1]	8
GPMC	C
HOST_CLK1_1	E
HOST_CLK1_2	F
IPU	10
L4_CFG	14
L4_PER1 P1	15
L4_PER1 P2	16
L4_PER1 P3	17
L4_PER2 P1	18
L3_INSTR	19
L4_PER2 P3	1A
L4_PER3 P1	1B
L4_PER3 P2	1C
L4_PER3 P3	1D
L4_WKUP	1E
McASP1	1F
MMU	22
OCMC_RAM	24
OCMC_ROM	27
EDMA_TPCC	30
EDMA_TC1	31
EDMA_TC2	32
QSPI	39
HOST_CLK2_1	40
DEBUGSS CT_TBR	41
L4_PER2 P2	42

(1) TESOC and CRC are not supported on the DRA78x family of devices.

The probes implement a user-defined set of counters that aggregate the events sampled by the detector and filtered according to the user setup.

NOTE: Statistics collectors counter values are accessible by application software.

Table 26-20 summarizes the performance probe aggregation modes.

Table 26-20. Aggregation Modes

Aggregation Mode	Description
FILTER_HIT	The counter increments by 1 when the filter hits.
MIN_MAX_HIT	The counter increments by 1 when the filter hits and the selected event information is within range. – Payload length (bytes) – Pressure value – Request/response latency (clock cycles)
EVT_INFO	The selected event information is added to the counter value when the filter hits. Payload length (bytes) Pressure value Request/response latency (clock cycles)
AND_FILTER	The counter increments by 1 when all unit filters hit.
OR_FILTER	The counter increments by 1 when at least one unit filter hits.
SUM_REQ_EVT	The counter sums the events from any request port.
SUM_RSP_EVT	The counter sums the events from any response port.
SUM_ALL_EVT	The counter sums the events from any port.
EXT_EVT	The counter increments by 1 when selected external event input signal is sampled high.

26.9.7.1 L3 Master Latency Monitoring

The L3 interconnect implements performance monitoring probes on initiator (master) interfaces. The master latency statistics are computed within a user-defined window and periodically reported to the user through the CT_STM interface.

The probes can be configured to filter latencies in four classes and report to the user a latency distribution along execution.

Because the performance metrics and the software events are exported through a unified export channel, it is possible to correlate latency trends with ongoing execution and system context.

Because computing latency requires maintaining the state between request and response ports, the probe cannot compute latency statistics on 100 percent of the initiator traffic. Hence, latency histograms must be extracted on large execution windows to be accurate.

26.9.7.1.1 STAT_COLL1 Configuration

STAT_COLL1 supports the following main features:

- Six probe inputs:
  - Probe 0: EMIF_SYS
  - Probe 1: IPU
  - Probe 2: DSP1_MDMA
  - Probe 3: DSP2_MDMA
  - Probe 4: GMAC_SW
  - Probe 5: ISS_NRT1
- Four 32-bit counters shared concurrently:
  - Counter 0 with one filter

  - Counter 1 with one filter
  - Counter 2 with one filter
  - Counter 3 with one filter
- Filtering according to:
  - L3 target
  - Access priorities
- 10-bit counter for latency measurement
- 32-bit collecting window counter
- Identifier is 0x0 (tie-off value)
- Dumps frames at slave address 0x19 (L3_INSTR)

Table 26-21 shows the STAT_COLL1 port mapping.

Table 26-21. STAT_COLL1 Port Mapping

Probe	Description	Link	Port
0	EMIF_SYS	NTTP REQ	0
		NTTP RSP	1
1	IPU	NTTP REQ	2
		NTTP RSP	3
2	DSP1_MDMA	NTTP REQ	4
		NTTP RSP	5
3	DSP2_MDMA	NTTP REQ	6
		NTTP RSP	7
4	GMAC_SW	NTTP REQ	8
		NTTP RSP	9
5	ISS_NRT1(1)	NTTP REQ	10
		NTTP RSP	11

(1) ISS is not supported on the DRA78x family of devices.

26.9.7.1.2 STAT_COLL2 Configuration

STAT_COLL2 supports the following main features:

- Eight probe inputs:
  - Probe 0: DSS
  - Probe 1: MMU
  - Probe 2: EDMA_TC0_RD
  - Probe 3: EDMA_TC0_WR
  - Probe 4: DSP2_CFG
  - Probe 5: DSP2_EDMA
  - Probe 6: Reserved
  - Probe 7: Reserved
- Four 32-bit counters shared concurrently:
  - Counter 0 with one filter
  - Counter 1 with one filter
  - Counter 2 with one filter
  - Counter 3 with one filter
- Filtering according to:
  - L3 target

  - Access priorities
- 10-bit counter for latency measurement
- 32-bit collecting window counter
- Identifier is 0x1 (tie-off value)
- Dumps frames at slave address 0x19 (L3_INSTR)

Table 26-22 shows the STAT_COLL2 port mapping.

Table 26-22. STAT_COLL2 Port Mapping

Probe	Description	Link	Port
0	DSS	NTTP REQ	0
		NTTP RSP	1
1	MMU	NTTP REQ	2
		NTTP RSP	3
2	EDMA_TC0_RD	NTTP REQ	4
		NTTP RSP	5
3	EDMA_TC0_WR	NTTP REQ	6
		NTTP RSP	7
4	DSP2_CFG	NTTP REQ	8
		NTTP RSP	9
5	DSP2_EDMA	NTTP REQ	10
		NTTP RSP	11
6	Reserved	NTTP REQ	12
		NTTP RSP	13
7	Reserved	NTTP REQ	14
		NTTP RSP	15

26.9.7.1.3 STAT_COLL3 Configuration

STAT_COLL3 supports the following main features:

- Seven probe inputs:
  - Probe 0: VIP1_P1
  - Probe 1: VIP1_P2
  - Probe 2: ISS_RT
  - Probe 3: ISS_NRT2
  - Probe 4: OCMC_RAM
  - Probe 5: DSP1_CFG
  - Probe 6: DSP1_EDMA
- Four 32-bit counters shared concurrently:
  - Counter 0 with one filter
  - Counter 1 with one filter
  - Counter 2 with one filter
  - Counter 3 with one filter
- Filtering according to:
  - L3 target
  - Access priorities
- 10-bit counter for latency measurement
- 32-bit collecting window counter

- Identifier is 0x2 (tie-off value)
- Dumps frames at slave address 0x19 (L3_INSTR)

Table 26-22 shows the STAT_COLL3 port mapping.

Table 26-23. STAT_COLL3 Port Mapping

Probe	Description	Link	Port
0	VIP1_P1	NTTP REQ	0
		NTTP RSP	1
1	VIP1_P2	NTTP REQ	2
		NTTP RSP	3
2	ISS_RT(1)	NTTP REQ	4
		NTTP RSP	5
3	ISS_NRT2(1)	NTTP REQ	6
		NTTP RSP	7
4	OCMC_RAM	NTTP REQ	8
		NTTP RSP	9
5	DSP1_CFG	NTTP REQ	10
		NTTP RSP	11
6	DSP1_EDMA	NTTP REQ	12
		NTTP RSP	13

(1) ISS is not supported on the DRA78x family of devices.

26.9.7.1.4 STAT_COLL4 Configuration

STAT_COLL4 supports the following main features:

- Eight probe inputs:
  - Probe 0: EVE_P1
  - Probe 1: EVE_P2
  - Probe 2: Reserved
  - Probe 3: GPMC
  - Probe 4: McASP1
  - Probe 5: EDMA_TC1_RD
  - Probe 6: EDMA_TC1_WR
  - Probe 7: Reserved
- Four 32-bit counters shared concurrently:
  - Counter 0 with one filter
  - Counter 1 with one filter
  - Counter 2 with one filter
  - Counter 3 with one filter
- Filtering according to:
  - L3 target
  - Access priorities
- 10-bit counter for latency measurement
- 32-bit collecting window counter
- Identifier is 0x4 (tie-off value)
- Dumps frames at slave address 0x19 (L3_INSTR)

Table 26-22 shows the STAT_COLL4 port mapping.

Table 26-24. STAT_COLL4 Port Mapping

Probe	Description	Link	Port
0	EVE_P1	NTTP REQ	0
		NTTP RSP	1
1	EVE_P2	NTTP REQ	2
		NTTP RSP	3
2	Reserved	NTTP REQ	4
		NTTP RSP	5
3	GPMC	NTTP REQ	6
		NTTP RSP	7
4	McASP1	NTTP REQ	8
		NTTP RSP	9
5	EDMA_TC1_RD	NTTP REQ	10
		NTTP RSP	11
6	EDMA_TC1_WR	NTTP REQ	12
		NTTP RSP	13
7	Reserved	NTTP REQ	14
		NTTP RSP	15

26.9.7.1.5 Statistics Collector Alarm Mode

Statistic collectors can be used to provide the application software with information about the NoC or SDRAM reaching corner cases (for example, too much traffic for a given OPP) while in application mode where for an end product use case the debug subsystem (which is normally exporting the statistic frames) is off. An interrupt-based scheme using a dedicated signal (L3_MAIN_IRQ_STAT_ALARM) is implemented to avoid CPU polling periodically statistic registers. This interrupt alert is fired when a given metric is out of specified range (below the programmed MIN threshold or above the programmed MAX threshold). The L3_MAIN_IRQ_STAT_ALARM interrupt is connected to the IRQ_CROSSBAR_16 input and the user can map it to any device INTC via Control Module.

NOTE: NTTP statistic collectors (SC_LAT) support latency measurement. However, comparison cannot be done on the latency counter. In case of latency measurement the comparison has to be done on number of latencies in user defined range, not on the latency value itself.

26.9.7.1.6 Statistics Collector Suspend Mode

The statistics collector module implements a suspend input that is used to avoid statistics collector counters to be updated while the processor has entered the debug state. This avoids triggering false alerts upon execution resume. When the statistics collector is asserted to 1, it freezes the monitoring process. When it goes back to 0, the monitoring resumes. If a frame is being dumped, it will not be stopped by suspend.

NOTE: Each statistic collector has an 'ignore suspend' register, which can be used to disable the suspend feature.

26.9.8 PM Instrumentation

The device takes advantage of the system trace infrastructure to provide visibility to the user about the major power-management events. This is supported through a PMI module (PM profiler) instantiated in the PRM module. The PRM state changes are handled as generic events and encapsulated in STP hardware messages and exported through the CT_STM module. The nature of the PM events does not require accurate timestamping and thus, timestamping is handled at CT_STM or trace receiver level.

The PM events are organized by class. Any PM state change from a specific class refreshes the entire instrumentation frame associated with that class. The STP message structure includes a PM event-ID indicating the class of the PM events.

The PM event classes supported are:

- Logic voltage domain OPP change
- Memory voltage domain OPP change
- Logic power domain state change
- Memory power domain state change

The PMI has a unique hardware master-ID assigned to it (see Section 26.9.10, *Master-ID Encoding*).

The PMI supports the possibility to report on activity in different event classes in the same sampling window. The user can size the capture sampling window.

Software events from the PM routines instrumentation can be interleaved with the PM hardware events. The user can take advantage of that feature to understand latencies for a specific power-management scenario or strategy.

The PM module implements an instrumentation port that directly interfaces with the debug subsystem (used to export PM events to CT_STM).

26.9.9 CM Instrumentation

The device instantiates a CMI module (CM profiler). It has a unique hardware master-ID assigned to it (for more information, see Section 26.9.10, *Master-ID Encoding*).

NOTE: Only CM_CORE_AON instrumentation is supported in this device. CM_CORE instrumentation is not supported.

The CM events profiling is similar to the PMI. Two exclusive instrumentation modes are supported:

- Clock activity:
  - Exposes to the user a snapshot of the state of all the clock domains derived from the same DPLL when CM detects a state change in the clock domain
  - Exposes to the user a snapshot of the DPLL settings when the CM signals a DPLL programming
- Module activity:
  - Exposes to the user periodically the active cycles count of the target modules
  - Exposes to the user periodically the active cycles count of the initiator modules

It provides visibility to the user about the state of the major clock domains along the application code execution. The STP message reports the effective state of the clock domain and therefore can highlight scenarios where a particular dependency is preventing the clock domain from being switched off.

The CM event classes supported are:

- Events capture mode – four classes:
  - Clock domain state
  - DPLL settings update
  - Clock frequency divider ratio
  - Clock source selection update
- Module activity collection mode – two classes:
  - Target module activity
  - Initiator module activity

When in events capture mode, the CMI supports the possibility of reporting on activity in different event classes in the same sampling window. The user can size the capture sampling window.

26.9.10 Master-ID Encoding

A master-ID (MReqMstID[7:0]) field is used by the CT_STM to encode a MASTER type of message:

- MReqMstID[7] – Differentiates software versus hardware masters (0 for software masters; 1 for hardware masters)
- MReqMstID[6:2] – Master address exported by the L3_MAIN interconnect debug subsystem target
- MReqMstID[1:0] – Additional qualifier for multicore masters (IPU)

26.9.10.1 Software Masters

The CT_STM module allows:

- Enabling a maximum of four SoC software masters
- Masking MReqMstID[1:0]
- Differentiating multicore software masters through MReqMstID[1:0]

Table 26-25 summarizes the software initiators that can export trace messages through the CT_STM.

Table 26-25. STM Message Software Masters

Initiator	MReqMstID				Restriction/Comment
	ConnID	[7]	[6:2]	[1:0]	
CS_DAP	0x4	0	00100	–	STP link testing
DSP1 MDMA	0x8	0	01000	–	
DSP1 CFG	0x9	0	01001	–	
DSP1 DMA	0xA	0	01010	–	Data logging
DSP2 DMA	0xB	0	01011	–	Data logging
DSP2 CFG	0xC	0	01100	–	
DSP2 MDMA	0xD	0	01101	–	
EVE	0x10	0	10000	–	Software messages routed through the EVE instrumentation port
IPU	0x18	0	11000	00	IPU_C0
				01	IPU_C1
EDMA_TC1 WR	0x1C	0	11100	–	Data logging
EDMA_TC2 WR	0x1D	0	11101	–	Data logging

26.9.10.2 Hardware Masters

The CT_STM module allows enabling a subset of SoC hardware masters (maximum = 4).

Table 26-26 summarizes the hardware initiators that can export trace messages through the STM.

Table 26-26. STM Message Hardware Masters

Initiator	MReqMstID			
	ConnID	[7]	[6:2]	[1:0]
EVE instrumentation (SMSET)	0x34	1	10100	00
ISS instrumentation (CTSET)(1)	0x37	1	10111	00
Performance probes (statistics collectors)(2)	0x38	1	11000	00
OCP watchpoint (OCP_WP_NOC)	0x39	1	11001	00
DMA profiling (OCP_WP_NOC)	0x3A	1	11010	00
System events (OCP_WP_NOC)	0x3B	1	11011	00
PMI events profiling	0x3D	1	11101	00
CMI events profiling	0x3E	1	11110	00

(1) ISS is not supported on the DRA78x family of devices.
(2) Master-ID assigned by the L3_MAIN interconnect

26.10 Concurrent Debug Modes

The debugger or application software can program the DRM to route a specific debug function to each device debug port pin.

Because of the limited number of pins allocated to debug, debug and trace source signals are multiplexed.

Table 26-27 summarizes the trace port configuration.

Table 26-27. Trace Port Configuration

Pin Name	Internal Signal Name	I/O	Trigger	CS_TPIU (DSP, CS_STM, CT_STM)
emu19	EMU19	O		TRACEDATA[15]
emu18	EMU18	O		TRACEDATA[14]
emu17	EMU17	O		TRACEDATA[13]
emu16	EMU16	O		TRACEDATA[12]
emu15	EMU15	O		TRACEDATA[11]
emu14	EMU14	O		TRACEDATA[10]
emu13	EMU13	O		TRACEDATA[9]
emu12	EMU12	O		TRACEDATA[8]
emu11	EMU11	O		TRACEDATA[7]
emu10	EMU10	O		TRACEDATA[6]
emu9	EMU9	O		TRACEDATA[5]
emu8	EMU8	O		TRACEDATA[4]
emu7	EMU7	O		TRACEDATA[3]
emu6	EMU6	O		TRACEDATA[2]
emu5	EMU5	O		TRACEDATA[1]
emu4	EMU4	O		TRACEDATA[0]
emu3	EMU3	O		TRACECTL
emu2	EMU2	O		TRACECLK
emu1	EMU1	I/O	Trigger1	
emu0	EMU0	I/O	Trigger0	

NOTE: The configuration of the trace port must comply with Table 26-27; otherwise, it will be ignored by DRM hardware. For example, if Trigger0 is programmed on EMU3 and Trigger1 is programmed on EMU4, this configuration will be ignored.

Table 26-28 summarizes the concurrent debug and trace in the device.

Table 26-28. Concurrent Debug and Trace

Debug Use Case	Concurrent Debug Flows	Debug Pins	Trace Pins		
		Triggers	Data	Control	Clock
1	DSP trace		16	1	1
	CT_STM				
	Triggers	2			
2	DSP trace		16	1	1
	CT_STM		1	–	1
	Triggers				
3	DSP trace		8	1	1
	CT_STM		4	–	1
	Triggers	2			

Table 26-28. Concurrent Debug and Trace (continued)

Debug Use Case	Concurrent Debug Flows	Debug Pins	Trace Pins		
		Triggers	Data	Control	Clock
4	DSP trace		11	1	1
	CT_STM		4	–	1
	Triggers	2			

26.11 DRM Register Manual

26.11.1 DRM Instance Summary

Table 26-29. DRM Instance Summary

Module Name	Module Base Address	Size
DRM	0x5416 0000	588 Bytes

26.11.2 DRM Registers

26.11.2.1 DRM Register Summary

NOTE: See for DRM suspend control registers assignment to corresponding peripherals.

Table 26-30. DRM Registers Mapping Summary

Register Name	Type	Register Width (Bits)	Address Offset	DRM Base Address
RESERVED	R	32	0x0000 0000 – 0x0000 00CC	0x5416 0000 – 0x5416 00CC
DRM_SUSPEND_CTRL0	RW	32	0x0000 0200	0x5416 0200
DRM_SUSPEND_CTRL1	RW	32	0x0000 0204	0x5416 0204
DRM_SUSPEND_CTRL2	RW	32	0x0000 0208	0x5416 0208
DRM_SUSPEND_CTRL3	RW	32	0x0000 020C	0x5416 020C
DRM_SUSPEND_CTRL4	RW	32	0x0000 0210	0x5416 0210
DRM_SUSPEND_CTRL5	RW	32	0x0000 0214	0x5416 0214
DRM_SUSPEND_CTRL6	RW	32	0x0000 0218	0x5416 0218
DRM_SUSPEND_CTRL7	RW	32	0x0000 021C	0x5416 021C
DRM_SUSPEND_CTRL8	RW	32	0x0000 0220	0x5416 0220
DRM_SUSPEND_CTRL9	RW	32	0x0000 0224	0x5416 0224
DRM_SUSPEND_CTRL10	RW	32	0x0000 0228	0x5416 0228
DRM_SUSPEND_CTRL11	RW	32	0x0000 022C	0x5416 022C
DRM_SUSPEND_CTRL12	RW	32	0x0000 0230	0x5416 0230
DRM_SUSPEND_CTRL13	RW	32	0x0000 0234	0x5416 0234
DRM_SUSPEND_CTRL14	RW	32	0x0000 0238	0x5416 0238
DRM_SUSPEND_CTRL15	RW	32	0x0000 023C	0x5416 023C
DRM_SUSPEND_CTRL16	RW	32	0x0000 0240	0x5416 0240
DRM_SUSPEND_CTRL17	RW	32	0x0000 0244	0x5416 0244
DRM_SUSPEND_CTRL18	RW	32	0x0000 0248	0x5416 0248
DRM_SUSPEND_CTRL19	RW	32	0x0000 024C	0x5416 024C
DRM_SUSPEND_CTRL20	RW	32	0x0000 0250	0x5416 0250

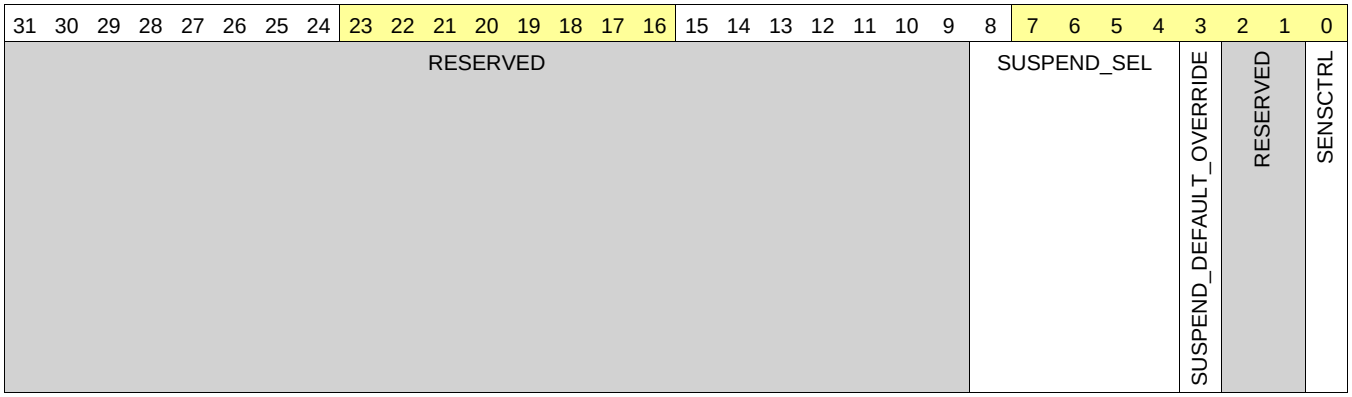
Table 26-30. DRM Registers Mapping Summary (continued)

Register Name	Type	Register Width (Bits)	Address Offset	DRM Base Address
DRM_SUSPEND_CTRL21	RW	32	0x0000 0254	0x5416 0254
DRM_SUSPEND_CTRL22	RW	32	0x0000 0258	0x5416 0258
DRM_SUSPEND_CTRL23	RW	32	0x0000 025C	0x5416 025C
DRM_SUSPEND_CTRL24	RW	32	0x0000 0260	0x5416 0260
DRM_SUSPEND_CTRL25	RW	32	0x0000 0264	0x5416 0264
DRM_SUSPEND_CTRL26	RW	32	0x0000 0268	0x5416 0268
DRM_SUSPEND_CTRL27	RW	32	0x0000 026C	0x5416 026C
DRM_SUSPEND_CTRL28	RW	32	0x0000 0270	0x5416 0270
DRM_SUSPEND_CTRL29	RW	32	0x0000 0274	0x5416 0274
DRM_SUSPEND_CTRL30	RW	32	0x0000 0278	0x5416 0278
DRM_SUSPEND_CTRL31	RW	32	0x0000 027C	0x5416 027C
DRM_SUSPEND_CTRL32	RW	32	0x0000 0280	0x5416 0280
RESERVED	R	32	0x0000 0400	0x5416 0400
RESERVED	R	32	0x0000 0500	0x5416 0500

26.11.2.2 DRM Register Description

Table 26-31. DRM_SUSPEND_CTRL0

Address Offset	0x0000 0200		
Physical Address	0x5416 0200	**Instance**	DRM
Description			
Type	RW		



Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reserved	R	0x0
8:4	SUSPEND_SEL	Suspend signal selection. Selects which suspend signal affects the peripheral. Only valid when SUSPEND_DEFAULT_OVERRIDE=0 and SENSCTRL=1. When read, these bits reflect the default suspend signal. 0000b: CPU suspend signal. All other values are reserved.	RW	0x0

Bits	Field Name	Description	Type	Reset
3	SUSPEND_DEFAULT_OVERRIDE	Enable or disable the override value in SUSPEND_SEL. 0: SUSPEND_SEL field will select which suspend signal reaches the peripheral. 1: SUSPEND_SEL field ignored. Default suspend signal will reach the peripheral.	R	0x0
2:1	RESERVED	Reserved	R	0x0
0	SENSCTRL	Sensitivity Control for suspend signals. When SUSPEND_DEFAULT_OVERRIDE=1, this bit is ignored and read as a 1. When SUSPEND_DEFAULT_OVERRIDE=0, 0: Suspend signal will not reach the peripheral. Peripheral will act as normal even during a debug halt. 1: Suspend signal will reach the peripheral. Peripheral will be suspended during debug halt.	RW	0x0

Table 26-32. Register Call Summary for Register DRM_SUSPEND_CTRL0

Primary Debug Support
- Debug Aware Peripherals and Host Processors: [0]

DRM Register Manual
- DRM Register Summary: [1]

Table 26-33. DRM_SUSPEND_CTRL1

Address Offset	0x0000 0204		
Physical Address	0x5416 0204	**Instance**	DRM
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								SUSPEND_SEL				SUSPEND_DEFAULT_OVERRIDE	RESERVED		SENSCTRL

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reserved	R	0x0
8:4	SUSPEND_SEL	Suspend signal selection. Selects which suspend signal affects the peripheral. Only valid when SUSPEND_DEFAULT_OVERRIDE=0 and SENSCTRL=1. When read, these bits reflect the default suspend signal. 0000b: CPU suspend signal. All other values are reserved.	RW	0x0
3	SUSPEND_DEFAULT_OVERRIDE	Enable or disable the override value in SUSPEND_SEL. 0: SUSPEND_SEL field will select which suspend signal reaches the peripheral. 1: SUSPEND_SEL field ignored. Default suspend signal will reach the peripheral.	R	0x0
2:1	RESERVED	Reserved	R	0x0

Bits	Field Name	Description	Type	Reset
0	SENSCTRL	Sensitivity Control for suspend signals. When SUSPEND_DEFAULT_OVERRIDE=1, this bit is ignored and read as a 1. When SUSPEND_DEFAULT_OVERRIDE=0, 0: Suspend signal will not reach the peripheral. Peripheral will act as normal even during a debug halt. 1: Suspend signal will reach the peripheral. Peripheral will be suspended during debug halt.	RW	0x0

Table 26-34. Register Call Summary for Register DRM_SUSPEND_CTRL1

Primary Debug Support • Debug Aware Peripherals and Host Processors: [0]
DRM Register Manual • DRM Register Summary: [1]

Table 26-35. DRM_SUSPEND_CTRL2

Address Offset	0x0000 0208		
Physical Address	0x5416 0208	**Instance**	DRM
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								SUSPEND_SEL				SUSPEND_DEFAULT_OVERRIDE	RESERVED		SENSCTRL

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reserved	R	0x0
8:4	SUSPEND_SEL	Suspend signal selection. Selects which suspend signal affects the peripheral. Only valid when SUSPEND_DEFAULT_OVERRIDE=0 and SENSCTRL=1. When read, these bits reflect the default suspend signal. 0000b: CPU suspend signal. All other values are reserved.	RW	0x0
3	SUSPEND_DEFAULT_OVERRIDE	Enable or disable the override value in SUSPEND_SEL. 0: SUSPEND_SEL field will select which suspend signal reaches the peripheral. 1: SUSPEND_SEL field ignored. Default suspend signal will reach the peripheral.	R	0x0
2:1	RESERVED	Reserved	R	0x0

Bits	Field Name	Description	Type	Reset
0	SENSCTRL	Sensitivity Control for suspend signals. When SUSPEND_DEFAULT_OVERRIDE=1, this bit is ignored and read as a 1. When SUSPEND_DEFAULT_OVERRIDE=0, 0: Suspend signal will not reach the peripheral. Peripheral will act as normal even during a debug halt. 1: Suspend signal will reach the peripheral. Peripheral will be suspended during debug halt.	RW	0x0

Table 26-36. Register Call Summary for Register DRM_SUSPEND_CTRL2

Primary Debug Support
- Debug Aware Peripherals and Host Processors: [0]

DRM Register Manual
- DRM Register Summary: [1]

Table 26-37. DRM_SUSPEND_CTRL3

Address Offset	0x0000 020C		
Physical Address	0x5416 020C	**Instance**	DRM
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								SUSPEND_SEL				SUSPEND_DEFAULT_OVERRIDE	RESERVED		SENSCTRL

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reserved	R	0x0
8:4	SUSPEND_SEL	Suspend signal selection. Selects which suspend signal affects the peripheral. Only valid when SUSPEND_DEFAULT_OVERRIDE=0 and SENSCTRL=1. When read, these bits reflect the default suspend signal. 0000b: CPU suspend signal. All other values are reserved.	RW	0x0
3	SUSPEND_DEFAULT_OVERRIDE	Enable or disable the override value in SUSPEND_SEL. 0: SUSPEND_SEL field will select which suspend signal reaches the peripheral. 1: SUSPEND_SEL field ignored. Default suspend signal will reach the peripheral.	R	0x0
2:1	RESERVED	Reserved	R	0x0

Bits	Field Name	Description	Type	Reset
0	SENSCTRL	Sensitivity Control for suspend signals. When SUSPEND_DEFAULT_OVERRIDE=1, this bit is ignored and read as a 1. When SUSPEND_DEFAULT_OVERRIDE=0, 0: Suspend signal will not reach the peripheral. Peripheral will act as normal even during a debug halt. 1: Suspend signal will reach the peripheral. Peripheral will be suspended during debug halt.	RW	0x0

Table 26-38. Register Call Summary for Register DRM_SUSPEND_CTRL3

Primary Debug Support
- Debug Aware Peripherals and Host Processors: [0]

DRM Register Manual
- DRM Register Summary: [1]

Table 26-39. DRM_SUSPEND_CTRL4

Address Offset	0x0000 0210		
Physical Address	0x5416 0210	**Instance**	DRM
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								SUSPEND_SEL				SUSPEND_DEFAULT_OVERRIDE	RESERVED		SENSCTRL

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reserved	R	0x0
8:4	SUSPEND_SEL	Suspend signal selection. Selects which suspend signal affects the peripheral. Only valid when SUSPEND_DEFAULT_OVERRIDE=0 and SENSCTRL=1. When read, these bits reflect the default suspend signal. 0000b: CPU suspend signal. All other values are reserved.	RW	0x0
3	SUSPEND_DEFAULT_OVERRIDE	Enable or disable the override value in SUSPEND_SEL. 0: SUSPEND_SEL field will select which suspend signal reaches the peripheral. 1: SUSPEND_SEL field ignored. Default suspend signal will reach the peripheral.	R	0x0
2:1	RESERVED	Reserved	R	0x0

Bits	Field Name	Description	Type	Reset
0	SENSCTRL	Sensitivity Control for suspend signals. When SUSPEND_DEFAULT_OVERRIDE=1, this bit is ignored and read as a 1. When SUSPEND_DEFAULT_OVERRIDE=0, 0: Suspend signal will not reach the peripheral. Peripheral will act as normal even during a debug halt. 1: Suspend signal will reach the peripheral. Peripheral will be suspended during debug halt.	RW	0x0

Table 26-40. Register Call Summary for Register DRM_SUSPEND_CTRL4

Primary Debug Support
- Debug Aware Peripherals and Host Processors: [0]

DRM Register Manual
- DRM Register Summary: [1]

Table 26-41. DRM_SUSPEND_CTRL5

Address Offset	0x0000 0214		
Physical Address	0x5416 0214	**Instance**	DRM
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								SUSPEND_SEL				SUSPEND_DEFAULT_OVERRIDE	RESERVED		SENSCTRL

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reserved	R	0x0
8:4	SUSPEND_SEL	Suspend signal selection. Selects which suspend signal affects the peripheral. Only valid when SUSPEND_DEFAULT_OVERRIDE=0 and SENSCTRL=1. When read, these bits reflect the default suspend signal. 0000b: CPU suspend signal. All other values are reserved.	RW	0x0
3	SUSPEND_DEFAULT_OVERRIDE	Enable or disable the override value in SUSPEND_SEL. 0: SUSPEND_SEL field will select which suspend signal reaches the peripheral. 1: SUSPEND_SEL field ignored. Default suspend signal will reach the peripheral.	R	0x0
2:1	RESERVED	Reserved	R	0x0

Bits	Field Name	Description	Type	Reset
0	SENSCTRL	Sensitivity Control for suspend signals. When SUSPEND_DEFAULT_OVERRIDE=1, this bit is ignored and read as a 1. When SUSPEND_DEFAULT_OVERRIDE=0, 0: Suspend signal will not reach the peripheral. Peripheral will act as normal even during a debug halt. 1: Suspend signal will reach the peripheral. Peripheral will be suspended during debug halt.	RW	0x0

Table 26-42. Register Call Summary for Register DRM_SUSPEND_CTRL5

Primary Debug Support • Debug Aware Peripherals and Host Processors: [0]
DRM Register Manual • DRM Register Summary: [1]

Table 26-43. DRM_SUSPEND_CTRL6

Address Offset	0x0000 0218		
Physical Address	0x5416 0218	**Instance**	DRM
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								SUSPEND_SEL				SUSPEND_DEFAULT_OVERRIDE	RESERVED		SENSCTRL

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reserved	R	0x0
8:4	SUSPEND_SEL	Suspend signal selection. Selects which suspend signal affects the peripheral. Only valid when SUSPEND_DEFAULT_OVERRIDE=0 and SENSCTRL=1. When read, these bits reflect the default suspend signal. 0000b: CPU suspend signal. All other values are reserved.	RW	0x0
3	SUSPEND_DEFAULT_OVERRIDE	Enable or disable the override value in SUSPEND_SEL. 0: SUSPEND_SEL field will select which suspend signal reaches the peripheral. 1: SUSPEND_SEL field ignored. Default suspend signal will reach the peripheral.	R	0x0
2:1	RESERVED	Reserved	R	0x0

Bits	Field Name	Description	Type	Reset
0	SENSCTRL	Sensitivity Control for suspend signals. When SUSPEND_DEFAULT_OVERRIDE=1, this bit is ignored and read as a 1. When SUSPEND_DEFAULT_OVERRIDE=0, 0: Suspend signal will not reach the peripheral. Peripheral will act as normal even during a debug halt. 1: Suspend signal will reach the peripheral. Peripheral will be suspended during debug halt.	RW	0x0

Table 26-44. Register Call Summary for Register DRM_SUSPEND_CTRL6

Primary Debug Support
- Debug Aware Peripherals and Host Processors: [0]

DRM Register Manual
- DRM Register Summary: [1]

Table 26-45. DRM_SUSPEND_CTRL7

Address Offset	0x0000 021C		
Physical Address	0x5416 021C	**Instance**	DRM
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								SUSPEND_SEL				SUSPEND_DEFAULT_OVERRIDE	RESERVED		SENSCTRL

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reserved	R	0x0
8:4	SUSPEND_SEL	Suspend signal selection. Selects which suspend signal affects the peripheral. Only valid when SUSPEND_DEFAULT_OVERRIDE=0 and SENSCTRL=1. When read, these bits reflect the default suspend signal. 0000b: CPU suspend signal. All other values are reserved.	RW	0x0
3	SUSPEND_DEFAULT_OVERRIDE	Enable or disable the override value in SUSPEND_SEL. 0: SUSPEND_SEL field will select which suspend signal reaches the peripheral. 1: SUSPEND_SEL field ignored. Default suspend signal will reach the peripheral.	R	0x0
2:1	RESERVED	Reserved	R	0x0

Bits	Field Name	Description	Type	Reset
0	SENSCTRL	Sensitivity Control for suspend signals. When SUSPEND_DEFAULT_OVERRIDE=1, this bit is ignored and read as a 1. When SUSPEND_DEFAULT_OVERRIDE=0, 0: Suspend signal will not reach the peripheral. Peripheral will act as normal even during a debug halt. 1: Suspend signal will reach the peripheral. Peripheral will be suspended during debug halt.	RW	0x0

Table 26-46. Register Call Summary for Register DRM_SUSPEND_CTRL7

Primary Debug Support • Debug Aware Peripherals and Host Processors: [0]
DRM Register Manual • DRM Register Summary: [1]

Table 26-47. DRM_SUSPEND_CTRL8

Address Offset	0x0000 0220		
Physical Address	0x5416 0220	**Instance**	DRM
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								SUSPEND_SEL				SUSPEND_DEFAULT_OVERRIDE	RESERVED		SENSCTRL

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reserved	R	0x0
8:4	SUSPEND_SEL	Suspend signal selection. Selects which suspend signal affects the peripheral. Only valid when SUSPEND_DEFAULT_OVERRIDE=0 and SENSCTRL=1. When read, these bits reflect the default suspend signal. 0000b: CPU suspend signal. All other values are reserved.	RW	0x0
3	SUSPEND_DEFAULT_OVERRIDE	Enable or disable the override value in SUSPEND_SEL. 0: SUSPEND_SEL field will select which suspend signal reaches the peripheral. 1: SUSPEND_SEL field ignored. Default suspend signal will reach the peripheral.	R	0x0
2:1	RESERVED	Reserved	R	0x0

Bits	Field Name	Description	Type	Reset
0	SENSCTRL	Sensitivity Control for suspend signals. When SUSPEND_DEFAULT_OVERRIDE=1, this bit is ignored and read as a 1. When SUSPEND_DEFAULT_OVERRIDE=0, 0: Suspend signal will not reach the peripheral. Peripheral will act as normal even during a debug halt. 1: Suspend signal will reach the peripheral. Peripheral will be suspended during debug halt.	RW	0x0

Table 26-48. Register Call Summary for Register DRM_SUSPEND_CTRL8

Primary Debug Support
- Debug Aware Peripherals and Host Processors: [0]

DRM Register Manual
- DRM Register Summary: [1]

Table 26-49. DRM_SUSPEND_CTRL9

Address Offset	0x0000 0224		
Physical Address	0x5416 0224	**Instance**	DRM
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								SUSPEND_SEL				SUSPEND_DEFAULT_OVERRIDE	RESERVED		SENSCTRL

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reserved	R	0x0
8:4	SUSPEND_SEL	Suspend signal selection. Selects which suspend signal affects the peripheral. Only valid when SUSPEND_DEFAULT_OVERRIDE=0 and SENSCTRL=1. When read, these bits reflect the default suspend signal. 0000b: CPU suspend signal. All other values are reserved.	RW	0x0
3	SUSPEND_DEFAULT_OVERRIDE	Enable or disable the override value in SUSPEND_SEL. 0: SUSPEND_SEL field will select which suspend signal reaches the peripheral. 1: SUSPEND_SEL field ignored. Default suspend signal will reach the peripheral.	R	0x0
2:1	RESERVED	Reserved	R	0x0

Bits	Field Name	Description	Type	Reset
0	SENSCTRL	Sensitivity Control for suspend signals. When SUSPEND_DEFAULT_OVERRIDE=1, this bit is ignored and read as a 1. When SUSPEND_DEFAULT_OVERRIDE=0, 0: Suspend signal will not reach the peripheral. Peripheral will act as normal even during a debug halt. 1: Suspend signal will reach the peripheral. Peripheral will be suspended during debug halt.	RW	0x0

Table 26-50. Register Call Summary for Register DRM_SUSPEND_CTRL9

Primary Debug Support • Debug Aware Peripherals and Host Processors: [0]
DRM Register Manual • DRM Register Summary: [1]

Table 26-51. DRM_SUSPEND_CTRL10

Address Offset	0x0000 0228		
Physical Address	0x5416 0228	**Instance**	DRM
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								SUSPEND_SEL				SUSPEND_DEFAULT_OVERRIDE	RESERVED		SENSCTRL

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reserved	R	0x0
8:4	SUSPEND_SEL	Suspend signal selection. Selects which suspend signal affects the peripheral. Only valid when SUSPEND_DEFAULT_OVERRIDE=0 and SENSCTRL=1. When read, these bits reflect the default suspend signal. 0000b: CPU suspend signal. All other values are reserved.	RW	0x0
3	SUSPEND_DEFAULT_OVERRIDE	Enable or disable the override value in SUSPEND_SEL. 0: SUSPEND_SEL field will select which suspend signal reaches the peripheral. 1: SUSPEND_SEL field ignored. Default suspend signal will reach the peripheral.	R	0x0
2:1	RESERVED	Reserved	R	0x0

Bits	Field Name	Description	Type	Reset
0	SENSCTRL	Sensitivity Control for suspend signals. When SUSPEND_DEFAULT_OVERRIDE=1, this bit is ignored and read as a 1. When SUSPEND_DEFAULT_OVERRIDE=0, 0: Suspend signal will not reach the peripheral. Peripheral will act as normal even during a debug halt. 1: Suspend signal will reach the peripheral. Peripheral will be suspended during debug halt.	RW	0x0

Table 26-52. Register Call Summary for Register DRM_SUSPEND_CTRL10

Primary Debug Support
- Debug Aware Peripherals and Host Processors: [0]

DRM Register Manual
- DRM Register Summary: [1]

Table 26-53. DRM_SUSPEND_CTRL11

Address Offset	0x0000 022C		
Physical Address	0x5416 022C	**Instance**	DRM
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								SUSPEND_SEL				SUSPEND_DEFAULT_OVERRIDE	RESERVED		SENSCTRL

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reserved	R	0x0
8:4	SUSPEND_SEL	Suspend signal selection. Selects which suspend signal affects the peripheral. Only valid when SUSPEND_DEFAULT_OVERRIDE=0 and SENSCTRL=1. When read, these bits reflect the default suspend signal. 0000b: CPU suspend signal. All other values are reserved.	RW	0x0
3	SUSPEND_DEFAULT_OVERRIDE	Enable or disable the override value in SUSPEND_SEL. 0: SUSPEND_SEL field will select which suspend signal reaches the peripheral. 1: SUSPEND_SEL field ignored. Default suspend signal will reach the peripheral.	R	0x0
2:1	RESERVED	Reserved	R	0x0

Bits	Field Name	Description	Type	Reset
0	SENSCTRL	Sensitivity Control for suspend signals. When SUSPEND_DEFAULT_OVERRIDE=1, this bit is ignored and read as a 1. When SUSPEND_DEFAULT_OVERRIDE=0, 0: Suspend signal will not reach the peripheral. Peripheral will act as normal even during a debug halt. 1: Suspend signal will reach the peripheral. Peripheral will be suspended during debug halt.	RW	0x0

Table 26-54. Register Call Summary for Register DRM_SUSPEND_CTRL11

Primary Debug Support
- Debug Aware Peripherals and Host Processors: [0]

DRM Register Manual
- DRM Register Summary: [1]

Table 26-55. DRM_SUSPEND_CTRL12

Address Offset	0x0000 0230		
Physical Address	0x5416 0230	**Instance**	DRM
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								SUSPEND_SEL				SUSPEND_DEFAULT_OVERRIDE	RESERVED		SENSCTRL

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reserved	R	0x0
8:4	SUSPEND_SEL	Suspend signal selection. Selects which suspend signal affects the peripheral. Only valid when SUSPEND_DEFAULT_OVERRIDE=0 and SENSCTRL=1. When read, these bits reflect the default suspend signal. 0000b: CPU suspend signal. All other values are reserved.	RW	0x0
3	SUSPEND_DEFAULT_OVERRIDE	Enable or disable the override value in SUSPEND_SEL. 0: SUSPEND_SEL field will select which suspend signal reaches the peripheral. 1: SUSPEND_SEL field ignored. Default suspend signal will reach the peripheral.	R	0x0
2:1	RESERVED	Reserved	R	0x0

Bits	Field Name	Description	Type	Reset
0	SENSCTRL	Sensitivity Control for suspend signals. When SUSPEND_DEFAULT_OVERRIDE=1, this bit is ignored and read as a 1. When SUSPEND_DEFAULT_OVERRIDE=0, 0: Suspend signal will not reach the peripheral. Peripheral will act as normal even during a debug halt. 1: Suspend signal will reach the peripheral. Peripheral will be suspended during debug halt.	RW	0x0

Table 26-56. Register Call Summary for Register DRM_SUSPEND_CTRL12

Primary Debug Support
- Debug Aware Peripherals and Host Processors: [0]

DRM Register Manual
- DRM Register Summary: [1]

Table 26-57. DRM_SUSPEND_CTRL13

Address Offset	0x0000 0234		
Physical Address	0x5416 0234	**Instance**	DRM
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								SUSPEND_SEL				SUSPEND_DEFAULT_OVERRIDE	RESERVED		SENSCTRL

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reserved	R	0x0
8:4	SUSPEND_SEL	Suspend signal selection. Selects which suspend signal affects the peripheral. Only valid when SUSPEND_DEFAULT_OVERRIDE=0 and SENSCTRL=1. When read, these bits reflect the default suspend signal. 0000b: CPU suspend signal. All other values are reserved.	RW	0x0
3	SUSPEND_DEFAULT_OVERRIDE	Enable or disable the override value in SUSPEND_SEL. 0: SUSPEND_SEL field will select which suspend signal reaches the peripheral. 1: SUSPEND_SEL field ignored. Default suspend signal will reach the peripheral.	R	0x0
2:1	RESERVED	Reserved	R	0x0

Bits	Field Name	Description	Type	Reset
0	SENSCTRL	Sensitivity Control for suspend signals. When SUSPEND_DEFAULT_OVERRIDE=1, this bit is ignored and read as a 1. When SUSPEND_DEFAULT_OVERRIDE=0, 0: Suspend signal will not reach the peripheral. Peripheral will act as normal even during a debug halt. 1: Suspend signal will reach the peripheral. Peripheral will be suspended during debug halt.	RW	0x0

Table 26-58. Register Call Summary for Register DRM_SUSPEND_CTRL13

Primary Debug Support • Debug Aware Peripherals and Host Processors: [0]
DRM Register Manual • DRM Register Summary: [1]

Table 26-59. DRM_SUSPEND_CTRL14

Address Offset	0x0000 0238		
Physical Address	0x5416 0238	**Instance**	DRM
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								SUSPEND_SEL				SUSPEND_DEFAULT_OVERRIDE	RESERVED		SENSCTRL

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reserved	R	0x0
8:4	SUSPEND_SEL	Suspend signal selection. Selects which suspend signal affects the peripheral. Only valid when SUSPEND_DEFAULT_OVERRIDE=0 and SENSCTRL=1. When read, these bits reflect the default suspend signal. 0000b: CPU suspend signal. All other values are reserved.	RW	0x0
3	SUSPEND_DEFAULT_OVERRIDE	Enable or disable the override value in SUSPEND_SEL. 0: SUSPEND_SEL field will select which suspend signal reaches the peripheral. 1: SUSPEND_SEL field ignored. Default suspend signal will reach the peripheral.	R	0x0
2:1	RESERVED	Reserved	R	0x0

Bits	Field Name	Description	Type	Reset
0	SENSCTRL	Sensitivity Control for suspend signals. When SUSPEND_DEFAULT_OVERRIDE=1, this bit is ignored and read as a 1. When SUSPEND_DEFAULT_OVERRIDE=0, 0: Suspend signal will not reach the peripheral. Peripheral will act as normal even during a debug halt. 1: Suspend signal will reach the peripheral. Peripheral will be suspended during debug halt.	RW	0x0

Table 26-60. Register Call Summary for Register DRM_SUSPEND_CTRL14

Primary Debug Support
- Debug Aware Peripherals and Host Processors: [0]

DRM Register Manual
- DRM Register Summary: [1]

Table 26-61. DRM_SUSPEND_CTRL15

Address Offset	0x0000 023C		
Physical Address	0x5416 023C	**Instance**	DRM
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								SUSPEND_SEL				SUSPEND_DEFAULT_OVERRIDE	RESERVED		SENSCTRL

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reserved	R	0x0
8:4	SUSPEND_SEL	Suspend signal selection. Selects which suspend signal affects the peripheral. Only valid when SUSPEND_DEFAULT_OVERRIDE=0 and SENSCTRL=1. When read, these bits reflect the default suspend signal. 0000b: CPU suspend signal. All other values are reserved.	RW	0x0
3	SUSPEND_DEFAULT_OVERRIDE	Enable or disable the override value in SUSPEND_SEL. 0: SUSPEND_SEL field will select which suspend signal reaches the peripheral. 1: SUSPEND_SEL field ignored. Default suspend signal will reach the peripheral.	R	0x0
2:1	RESERVED	Reserved	R	0x0

Bits	Field Name	Description	Type	Reset
0	SENSCTRL	Sensitivity Control for suspend signals. When SUSPEND_DEFAULT_OVERRIDE=1, this bit is ignored and read as a 1. When SUSPEND_DEFAULT_OVERRIDE=0, 0: Suspend signal will not reach the peripheral. Peripheral will act as normal even during a debug halt. 1: Suspend signal will reach the peripheral. Peripheral will be suspended during debug halt.	RW	0x0

Table 26-62. Register Call Summary for Register DRM_SUSPEND_CTRL15

Primary Debug Support • Debug Aware Peripherals and Host Processors: [0]
DRM Register Manual • DRM Register Summary: [1]

Table 26-63. DRM_SUSPEND_CTRL16

Address Offset	0x0000 0240		
Physical Address	0x5416 0240	**Instance**	DRM
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								SUSPEND_SEL				SUSPEND_DEFAULT_OVERRIDE	RESERVED		SENSCTRL

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reserved	R	0x0
8:4	SUSPEND_SEL	Suspend signal selection. Selects which suspend signal affects the peripheral. Only valid when SUSPEND_DEFAULT_OVERRIDE=0 and SENSCTRL=1. When read, these bits reflect the default suspend signal. 0000b: CPU suspend signal. All other values are reserved.	RW	0x0
3	SUSPEND_DEFAULT_OVERRIDE	Enable or disable the override value in SUSPEND_SEL. 0: SUSPEND_SEL field will select which suspend signal reaches the peripheral. 1: SUSPEND_SEL field ignored. Default suspend signal will reach the peripheral.	R	0x0
2:1	RESERVED	Reserved	R	0x0

Bits	Field Name	Description	Type	Reset
0	SENSCTRL	Sensitivity Control for suspend signals. When SUSPEND_DEFAULT_OVERRIDE=1, this bit is ignored and read as a 1. When SUSPEND_DEFAULT_OVERRIDE=0, 0: Suspend signal will not reach the peripheral. Peripheral will act as normal even during a debug halt. 1: Suspend signal will reach the peripheral. Peripheral will be suspended during debug halt.	RW	0x0

Table 26-64. Register Call Summary for Register DRM_SUSPEND_CTRL16

Primary Debug Support
- Debug Aware Peripherals and Host Processors: [0]

DRM Register Manual
- DRM Register Summary: [1]

Table 26-65. DRM_SUSPEND_CTRL17

Address Offset	0x0000 0244		
Physical Address	0x5416 0244	**Instance**	DRM
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								SUSPEND_SEL				SUSPEND_DEFAULT_OVERRIDE	RESERVED		SENSCTRL

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reserved	R	0x0
8:4	SUSPEND_SEL	Suspend signal selection. Selects which suspend signal affects the peripheral. Only valid when SUSPEND_DEFAULT_OVERRIDE=0 and SENSCTRL=1. When read, these bits reflect the default suspend signal. 0000b: CPU suspend signal. All other values are reserved.	RW	0x0
3	SUSPEND_DEFAULT_OVERRIDE	Enable or disable the override value in SUSPEND_SEL. 0: SUSPEND_SEL field will select which suspend signal reaches the peripheral. 1: SUSPEND_SEL field ignored. Default suspend signal will reach the peripheral.	R	0x0
2:1	RESERVED	Reserved	R	0x0

Bits	Field Name	Description	Type	Reset
0	SENSCTRL	Sensitivity Control for suspend signals. When SUSPEND_DEFAULT_OVERRIDE=1, this bit is ignored and read as a 1. When SUSPEND_DEFAULT_OVERRIDE=0, 0: Suspend signal will not reach the peripheral. Peripheral will act as normal even during a debug halt. 1: Suspend signal will reach the peripheral. Peripheral will be suspended during debug halt.	RW	0x0

Table 26-66. Register Call Summary for Register DRM_SUSPEND_CTRL17

Primary Debug Support
- Debug Aware Peripherals and Host Processors: [0]

DRM Register Manual
- DRM Register Summary: [1]

Table 26-67. DRM_SUSPEND_CTRL18

Address Offset	0x0000 0248		
Physical Address	0x5416 0248	**Instance**	DRM
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								SUSPEND_SEL				SUSPEND_DEFAULT_OVERRIDE	RESERVED		SENSCTRL

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reserved	R	0x0
8:4	SUSPEND_SEL	Suspend signal selection. Selects which suspend signal affects the peripheral. Only valid when SUSPEND_DEFAULT_OVERRIDE=0 and SENSCTRL=1. When read, these bits reflect the default suspend signal. 0000b: CPU suspend signal. All other values are reserved.	RW	0x0
3	SUSPEND_DEFAULT_OVERRIDE	Enable or disable the override value in SUSPEND_SEL. 0: SUSPEND_SEL field will select which suspend signal reaches the peripheral. 1: SUSPEND_SEL field ignored. Default suspend signal will reach the peripheral.	R	0x0
2:1	RESERVED	Reserved	R	0x0

Bits	Field Name	Description	Type	Reset
0	SENSCTRL	Sensitivity Control for suspend signals. When SUSPEND_DEFAULT_OVERRIDE=1, this bit is ignored and read as a 1. When SUSPEND_DEFAULT_OVERRIDE=0, 0: Suspend signal will not reach the peripheral. Peripheral will act as normal even during a debug halt. 1: Suspend signal will reach the peripheral. Peripheral will be suspended during debug halt.	RW	0x0

Table 26-68. Register Call Summary for Register DRM_SUSPEND_CTRL18

Primary Debug Support
- Debug Aware Peripherals and Host Processors: [0]

DRM Register Manual
- DRM Register Summary: [1]

Table 26-69. DRM_SUSPEND_CTRL19

Address Offset	0x0000 024C		
Physical Address	0x5416 024C	**Instance**	DRM
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								SUSPEND_SEL				SUSPEND_DEFAULT_OVERRIDE	RESERVED		SENSCTRL

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reserved	R	0x0
8:4	SUSPEND_SEL	Suspend signal selection. Selects which suspend signal affects the peripheral. Only valid when SUSPEND_DEFAULT_OVERRIDE=0 and SENSCTRL=1. When read, these bits reflect the default suspend signal. 0000b: CPU suspend signal. All other values are reserved.	RW	0x0
3	SUSPEND_DEFAULT_OVERRIDE	Enable or disable the override value in SUSPEND_SEL. 0: SUSPEND_SEL field will select which suspend signal reaches the peripheral. 1: SUSPEND_SEL field ignored. Default suspend signal will reach the peripheral.	R	0x0
2:1	RESERVED	Reserved	R	0x0

Bits	Field Name	Description	Type	Reset
0	SENSCTRL	Sensitivity Control for suspend signals. When SUSPEND_DEFAULT_OVERRIDE=1, this bit is ignored and read as a 1. When SUSPEND_DEFAULT_OVERRIDE=0, 0: Suspend signal will not reach the peripheral. Peripheral will act as normal even during a debug halt. 1: Suspend signal will reach the peripheral. Peripheral will be suspended during debug halt.	RW	0x0

Table 26-70. Register Call Summary for Register DRM_SUSPEND_CTRL19

Primary Debug Support
- Debug Aware Peripherals and Host Processors: [0]

DRM Register Manual
- DRM Register Summary: [1]

Table 26-71. DRM_SUSPEND_CTRL20

Address Offset	0x0000 0250		
Physical Address	0x5416 0250	**Instance**	DRM
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								SUSPEND_SEL				SUSPEND_DEFAULT_OVERRIDE	RESERVED		SENSCTRL

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reserved	R	0x0
8:4	SUSPEND_SEL	Suspend signal selection. Selects which suspend signal affects the peripheral. Only valid when SUSPEND_DEFAULT_OVERRIDE=0 and SENSCTRL=1. When read, these bits reflect the default suspend signal. 0000b: CPU suspend signal. All other values are reserved.	RW	0x0
3	SUSPEND_DEFAULT_OVERRIDE	Enable or disable the override value in SUSPEND_SEL. 0: SUSPEND_SEL field will select which suspend signal reaches the peripheral. 1: SUSPEND_SEL field ignored. Default suspend signal will reach the peripheral.	R	0x0
2:1	RESERVED	Reserved	R	0x0

Bits	Field Name	Description	Type	Reset
0	SENSCTRL	Sensitivity Control for suspend signals. When SUSPEND_DEFAULT_OVERRIDE=1, this bit is ignored and read as a 1. When SUSPEND_DEFAULT_OVERRIDE=0, 0: Suspend signal will not reach the peripheral. Peripheral will act as normal even during a debug halt. 1: Suspend signal will reach the peripheral. Peripheral will be suspended during debug halt.	RW	0x0

Table 26-72. Register Call Summary for Register DRM_SUSPEND_CTRL20

Primary Debug Support
- Debug Aware Peripherals and Host Processors: [0]

DRM Register Manual
- DRM Register Summary: [1]

Table 26-73. DRM_SUSPEND_CTRL21

Address Offset	0x0000 0254		
Physical Address	0x5416 0254	**Instance**	DRM
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								SUSPEND_SEL				SUSPEND_DEFAULT_OVERRIDE	RESERVED		SENSCTRL

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reserved	R	0x0
8:4	SUSPEND_SEL	Suspend signal selection. Selects which suspend signal affects the peripheral. Only valid when SUSPEND_DEFAULT_OVERRIDE=0 and SENSCTRL=1. When read, these bits reflect the default suspend signal. 0000b: CPU suspend signal. All other values are reserved.	RW	0x0
3	SUSPEND_DEFAULT_OVERRIDE	Enable or disable the override value in SUSPEND_SEL. 0: SUSPEND_SEL field will select which suspend signal reaches the peripheral. 1: SUSPEND_SEL field ignored. Default suspend signal will reach the peripheral.	R	0x0
2:1	RESERVED	Reserved	R	0x0

Bits	Field Name	Description	Type	Reset
0	SENSCTRL	Sensitivity Control for suspend signals. When SUSPEND_DEFAULT_OVERRIDE=1, this bit is ignored and read as a 1. When SUSPEND_DEFAULT_OVERRIDE=0, 0: Suspend signal will not reach the peripheral. Peripheral will act as normal even during a debug halt. 1: Suspend signal will reach the peripheral. Peripheral will be suspended during debug halt.	RW	0x0

Table 26-74. Register Call Summary for Register DRM_SUSPEND_CTRL21

Primary Debug Support
- Debug Aware Peripherals and Host Processors: [0]

DRM Register Manual
- DRM Register Summary: [1]

Table 26-75. DRM_SUSPEND_CTRL22

Address Offset	0x0000 0258		
Physical Address	0x5416 0258	**Instance**	DRM
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								SUSPEND_SEL				SUSPEND_DEFAULT_OVERRIDE	RESERVED		SENSCTRL

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reserved	R	0x0
8:4	SUSPEND_SEL	Suspend signal selection. Selects which suspend signal affects the peripheral. Only valid when SUSPEND_DEFAULT_OVERRIDE=0 and SENSCTRL=1. When read, these bits reflect the default suspend signal. 0000b: CPU suspend signal. All other values are reserved.	RW	0x0
3	SUSPEND_DEFAULT_OVERRIDE	Enable or disable the override value in SUSPEND_SEL. 0: SUSPEND_SEL field will select which suspend signal reaches the peripheral. 1: SUSPEND_SEL field ignored. Default suspend signal will reach the peripheral.	R	0x0
2:1	RESERVED	Reserved	R	0x0

Bits	Field Name	Description	Type	Reset
0	SENSCTRL	Sensitivity Control for suspend signals. When SUSPEND_DEFAULT_OVERRIDE=1, this bit is ignored and read as a 1. When SUSPEND_DEFAULT_OVERRIDE=0, 0: Suspend signal will not reach the peripheral. Peripheral will act as normal even during a debug halt. 1: Suspend signal will reach the peripheral. Peripheral will be suspended during debug halt.	RW	0x0

Table 26-76. Register Call Summary for Register DRM_SUSPEND_CTRL22

Primary Debug Support
- Debug Aware Peripherals and Host Processors: [0]

DRM Register Manual
- DRM Register Summary: [1]

Table 26-77. DRM_SUSPEND_CTRL23

Address Offset	0x0000 025C		
Physical Address	0x5416 025C	**Instance**	DRM
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								SUSPEND_SEL				SUSPEND_DEFAULT_OVERRIDE	RESERVED		SENSCTRL

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reserved	R	0x0
8:4	SUSPEND_SEL	Suspend signal selection. Selects which suspend signal affects the peripheral. Only valid when SUSPEND_DEFAULT_OVERRIDE=0 and SENSCTRL=1. When read, these bits reflect the default suspend signal. 0000b: CPU suspend signal. All other values are reserved.	RW	0x0
3	SUSPEND_DEFAULT_OVERRIDE	Enable or disable the override value in SUSPEND_SEL. 0: SUSPEND_SEL field will select which suspend signal reaches the peripheral. 1: SUSPEND_SEL field ignored. Default suspend signal will reach the peripheral.	R	0x0
2:1	RESERVED	Reserved	R	0x0

Bits	Field Name	Description	Type	Reset
0	SENSCTRL	Sensitivity Control for suspend signals. When SUSPEND_DEFAULT_OVERRIDE=1, this bit is ignored and read as a 1. When SUSPEND_DEFAULT_OVERRIDE=0, 0: Suspend signal will not reach the peripheral. Peripheral will act as normal even during a debug halt. 1: Suspend signal will reach the peripheral. Peripheral will be suspended during debug halt.	RW	0x0

Table 26-78. Register Call Summary for Register DRM_SUSPEND_CTRL23

Primary Debug Support • Debug Aware Peripherals and Host Processors: [0]
DRM Register Manual • DRM Register Summary: [1]

Table 26-79. DRM_SUSPEND_CTRL24

Address Offset	0x0000 0260		
Physical Address	0x5416 0260	**Instance**	DRM
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								SUSPEND_SEL				SUSPEND_DEFAULT_OVERRIDE	RESERVED		SENSCTRL

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reserved	R	0x0
8:4	SUSPEND_SEL	Suspend signal selection. Selects which suspend signal affects the peripheral. Only valid when SUSPEND_DEFAULT_OVERRIDE=0 and SENSCTRL=1. When read, these bits reflect the default suspend signal. 0000b: CPU suspend signal. All other values are reserved.	RW	0x0
3	SUSPEND_DEFAULT_OVERRIDE	Enable or disable the override value in SUSPEND_SEL. 0: SUSPEND_SEL field will select which suspend signal reaches the peripheral. 1: SUSPEND_SEL field ignored. Default suspend signal will reach the peripheral.	R	0x0
2:1	RESERVED	Reserved	R	0x0

Bits	Field Name	Description	Type	Reset
0	SENSCTRL	Sensitivity Control for suspend signals. When SUSPEND_DEFAULT_OVERRIDE=1, this bit is ignored and read as a 1. When SUSPEND_DEFAULT_OVERRIDE=0, 0: Suspend signal will not reach the peripheral. Peripheral will act as normal even during a debug halt. 1: Suspend signal will reach the peripheral. Peripheral will be suspended during debug halt.	RW	0x0

Table 26-80. Register Call Summary for Register DRM_SUSPEND_CTRL24

Primary Debug Support
- Debug Aware Peripherals and Host Processors: [0]

DRM Register Manual
- DRM Register Summary: [1]

Table 26-81. DRM_SUSPEND_CTRL25

Address Offset	0x0000 0264		
Physical Address	0x5416 0264	**Instance**	DRM
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								SUSPEND_SEL				SUSPEND_DEFAULT_OVERRIDE	RESERVED		SENSCTRL

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reserved	R	0x0
8:4	SUSPEND_SEL	Suspend signal selection. Selects which suspend signal affects the peripheral. Only valid when SUSPEND_DEFAULT_OVERRIDE=0 and SENSCTRL=1. When read, these bits reflect the default suspend signal. 0000b: CPU suspend signal. All other values are reserved.	RW	0x0
3	SUSPEND_DEFAULT_OVERRIDE	Enable or disable the override value in SUSPEND_SEL. 0: SUSPEND_SEL field will select which suspend signal reaches the peripheral. 1: SUSPEND_SEL field ignored. Default suspend signal will reach the peripheral.	R	0x0
2:1	RESERVED	Reserved	R	0x0

Bits	Field Name	Description	Type	Reset
0	SENSCTRL	Sensitivity Control for suspend signals. When SUSPEND_DEFAULT_OVERRIDE=1, this bit is ignored and read as a 1. When SUSPEND_DEFAULT_OVERRIDE=0, 0: Suspend signal will not reach the peripheral. Peripheral will act as normal even during a debug halt. 1: Suspend signal will reach the peripheral. Peripheral will be suspended during debug halt.	RW	0x0

Table 26-82. Register Call Summary for Register DRM_SUSPEND_CTRL25

Primary Debug Support
- Debug Aware Peripherals and Host Processors: [0]

DRM Register Manual
- DRM Register Summary: [1]

Table 26-83. DRM_SUSPEND_CTRL26

Address Offset	0x0000 0268		
Physical Address	0x5416 0268	**Instance**	DRM
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								SUSPEND_SEL				SUSPEND_DEFAULT_OVERRIDE	RESERVED		SENSCTRL

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reserved	R	0x0
8:4	SUSPEND_SEL	Suspend signal selection. Selects which suspend signal affects the peripheral. Only valid when SUSPEND_DEFAULT_OVERRIDE=0 and SENSCTRL=1. When read, these bits reflect the default suspend signal. 0000b: CPU suspend signal. All other values are reserved.	RW	0x0
3	SUSPEND_DEFAULT_OVERRIDE	Enable or disable the override value in SUSPEND_SEL. 0: SUSPEND_SEL field will select which suspend signal reaches the peripheral. 1: SUSPEND_SEL field ignored. Default suspend signal will reach the peripheral.	R	0x0
2:1	RESERVED	Reserved	R	0x0

Bits	Field Name	Description	Type	Reset
0	SENSCTRL	Sensitivity Control for suspend signals. When SUSPEND_DEFAULT_OVERRIDE=1, this bit is ignored and read as a 1. When SUSPEND_DEFAULT_OVERRIDE=0, 0: Suspend signal will not reach the peripheral. Peripheral will act as normal even during a debug halt. 1: Suspend signal will reach the peripheral. Peripheral will be suspended during debug halt.	RW	0x0

Table 26-84. Register Call Summary for Register DRM_SUSPEND_CTRL26

Primary Debug Support
- Debug Aware Peripherals and Host Processors: [0]

DRM Register Manual
- DRM Register Summary: [1]

Table 26-85. DRM_SUSPEND_CTRL27

Address Offset	0x0000 026C		
Physical Address	0x5416 026C	**Instance**	DRM
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								SUSPEND_SEL				SUSPEND_DEFAULT_OVERRIDE	RESERVED		SENSCTRL

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reserved	R	0x0
8:4	SUSPEND_SEL	Suspend signal selection. Selects which suspend signal affects the peripheral. Only valid when SUSPEND_DEFAULT_OVERRIDE=0 and SENSCTRL=1. When read, these bits reflect the default suspend signal. 0000b: CPU suspend signal. All other values are reserved.	RW	0x0
3	SUSPEND_DEFAULT_OVERRIDE	Enable or disable the override value in SUSPEND_SEL. 0: SUSPEND_SEL field will select which suspend signal reaches the peripheral. 1: SUSPEND_SEL field ignored. Default suspend signal will reach the peripheral.	R	0x0
2:1	RESERVED	Reserved	R	0x0

Bits	Field Name	Description	Type	Reset
0	SENSCTRL	Sensitivity Control for suspend signals. When SUSPEND_DEFAULT_OVERRIDE=1, this bit is ignored and read as a 1. When SUSPEND_DEFAULT_OVERRIDE=0, 0: Suspend signal will not reach the peripheral. Peripheral will act as normal even during a debug halt. 1: Suspend signal will reach the peripheral. Peripheral will be suspended during debug halt.	RW	0x0

Table 26-86. Register Call Summary for Register DRM_SUSPEND_CTRL27

Primary Debug Support
- Debug Aware Peripherals and Host Processors: [0]

DRM Register Manual
- DRM Register Summary: [1]

Table 26-87. DRM_SUSPEND_CTRL28

Address Offset	0x0000 0270		
Physical Address	0x5416 0270	**Instance**	DRM
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								SUSPEND_SEL				SUSPEND_DEFAULT_OVERRIDE	RESERVED		SENSCTRL

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reserved	R	0x0
8:4	SUSPEND_SEL	Suspend signal selection. Selects which suspend signal affects the peripheral. Only valid when SUSPEND_DEFAULT_OVERRIDE=0 and SENSCTRL=1. When read, these bits reflect the default suspend signal. 0000b: CPU suspend signal. All other values are reserved.	RW	0x0
3	SUSPEND_DEFAULT_OVERRIDE	Enable or disable the override value in SUSPEND_SEL. 0: SUSPEND_SEL field will select which suspend signal reaches the peripheral. 1: SUSPEND_SEL field ignored. Default suspend signal will reach the peripheral.	R	0x0
2:1	RESERVED	Reserved	R	0x0

Bits	Field Name	Description	Type	Reset
0	SENSCTRL	Sensitivity Control for suspend signals. When SUSPEND_DEFAULT_OVERRIDE=1, this bit is ignored and read as a 1. When SUSPEND_DEFAULT_OVERRIDE=0, 0: Suspend signal will not reach the peripheral. Peripheral will act as normal even during a debug halt. 1: Suspend signal will reach the peripheral. Peripheral will be suspended during debug halt.	RW	0x0

Table 26-88. Register Call Summary for Register DRM_SUSPEND_CTRL28

Primary Debug Support
- Debug Aware Peripherals and Host Processors: [0]

DRM Register Manual
- DRM Register Summary: [1]

Table 26-89. DRM_SUSPEND_CTRL29

Address Offset	0x0000 0274		
Physical Address	0x5416 0274	**Instance**	DRM
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								SUSPEND_SEL				SUSPEND_DEFAULT_OVERRIDE	RESERVED		SENSCTRL

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reserved	R	0x0
8:4	SUSPEND_SEL	Suspend signal selection. Selects which suspend signal affects the peripheral. Only valid when SUSPEND_DEFAULT_OVERRIDE=0 and SENSCTRL=1. When read, these bits reflect the default suspend signal. 0000b: CPU suspend signal. All other values are reserved.	RW	0x0
3	SUSPEND_DEFAULT_OVERRIDE	Enable or disable the override value in SUSPEND_SEL. 0: SUSPEND_SEL field will select which suspend signal reaches the peripheral. 1: SUSPEND_SEL field ignored. Default suspend signal will reach the peripheral.	R	0x0
2:1	RESERVED	Reserved	R	0x0

Bits	Field Name	Description	Type	Reset
0	SENSCTRL	Sensitivity Control for suspend signals. When SUSPEND_DEFAULT_OVERRIDE=1, this bit is ignored and read as a 1. When SUSPEND_DEFAULT_OVERRIDE=0, 0: Suspend signal will not reach the peripheral. Peripheral will act as normal even during a debug halt. 1: Suspend signal will reach the peripheral. Peripheral will be suspended during debug halt.	RW	0x0

Table 26-90. Register Call Summary for Register DRM_SUSPEND_CTRL29

Primary Debug Support • Debug Aware Peripherals and Host Processors: [0]
DRM Register Manual • DRM Register Summary: [1]

Table 26-91. DRM_SUSPEND_CTRL30

Address Offset	0x0000 0278		
Physical Address	0x5416 0278	**Instance**	DRM
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								SUSPEND_SEL				SUSPEND_DEFAULT_OVERRIDE	RESERVED		SENSCTRL

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reserved	R	0x0
8:4	SUSPEND_SEL	Suspend signal selection. Selects which suspend signal affects the peripheral. Only valid when SUSPEND_DEFAULT_OVERRIDE=0 and SENSCTRL=1. When read, these bits reflect the default suspend signal. 0000b: CPU suspend signal. All other values are reserved.	RW	0x0
3	SUSPEND_DEFAULT_OVERRIDE	Enable or disable the override value in SUSPEND_SEL. 0: SUSPEND_SEL field will select which suspend signal reaches the peripheral. 1: SUSPEND_SEL field ignored. Default suspend signal will reach the peripheral.	R	0x0
2:1	RESERVED	Reserved	R	0x0

Bits	Field Name	Description	Type	Reset
0	SENSCTRL	Sensitivity Control for suspend signals. When SUSPEND_DEFAULT_OVERRIDE=1, this bit is ignored and read as a 1. When SUSPEND_DEFAULT_OVERRIDE=0, 0: Suspend signal will not reach the peripheral. Peripheral will act as normal even during a debug halt. 1: Suspend signal will reach the peripheral. Peripheral will be suspended during debug halt.	RW	0x0

Table 26-92. Register Call Summary for Register DRM_SUSPEND_CTRL30

Primary Debug Support
- Debug Aware Peripherals and Host Processors: [0]

DRM Register Manual
- DRM Register Summary: [1]

Table 26-93. DRM_SUSPEND_CTRL31

Address Offset	0x0000 027C		
Physical Address	0x5416 027C	**Instance**	DRM
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								SUSPEND_SEL				SUSPEND_DEFAULT_OVERRIDE	RESERVED		SENSCTRL

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reserved	R	0x0
8:4	SUSPEND_SEL	Suspend signal selection. Selects which suspend signal affects the peripheral. Only valid when SUSPEND_DEFAULT_OVERRIDE=0 and SENSCTRL=1. When read, these bits reflect the default suspend signal. 0000b: CPU suspend signal. All other values are reserved.	RW	0x0
3	SUSPEND_DEFAULT_OVERRIDE	Enable or disable the override value in SUSPEND_SEL. 0: SUSPEND_SEL field will select which suspend signal reaches the peripheral. 1: SUSPEND_SEL field ignored. Default suspend signal will reach the peripheral.	R	0x0
2:1	RESERVED	Reserved	R	0x0

Bits	Field Name	Description	Type	Reset
0	SENSCTRL	Sensitivity Control for suspend signals. When SUSPEND_DEFAULT_OVERRIDE=1, this bit is ignored and read as a 1. When SUSPEND_DEFAULT_OVERRIDE=0, 0: Suspend signal will not reach the peripheral. Peripheral will act as normal even during a debug halt. 1: Suspend signal will reach the peripheral. Peripheral will be suspended during debug halt.	RW	0x0

Table 26-94. Register Call Summary for Register DRM_SUSPEND_CTRL31

Primary Debug Support • Debug Aware Peripherals and Host Processors: [0]
DRM Register Manual • DRM Register Summary: [1]

Table 26-95. DRM_SUSPEND_CTRL32

Address Offset	0x0000 0280		
Physical Address	0x5416 0280	**Instance**	DRM
Description			
Type	RW		

31	30	29	28	27	26	25	24	23	22	21	20	19	18	17	16	15	14	13	12	11	10	9	8	7	6	5	4	3	2	1	0
RESERVED																								SUSPEND_SEL				SUSPEND_DEFAULT_OVERRIDE	RESERVED		SENSCTRL

Bits	Field Name	Description	Type	Reset
31:9	RESERVED	Reserved	R	0x0
8:4	SUSPEND_SEL	Suspend signal selection. Selects which suspend signal affects the peripheral. Only valid when SUSPEND_DEFAULT_OVERRIDE=0 and SENSCTRL=1. When read, these bits reflect the default suspend signal. 0000b: CPU suspend signal. All other values are reserved.	RW	0x0
3	SUSPEND_DEFAULT_OVERRIDE	Enable or disable the override value in SUSPEND_SEL. 0: SUSPEND_SEL field will select which suspend signal reaches the peripheral. 1: SUSPEND_SEL field ignored. Default suspend signal will reach the peripheral.	R	0x0
2:1	RESERVED	Reserved	R	0x0

Bits	Field Name	Description	Type	Reset
0	SENSCTRL	Sensitivity Control for suspend signals. When SUSPEND_DEFAULT_OVERRIDE=1, this bit is ignored and read as a 1. When SUSPEND_DEFAULT_OVERRIDE=0, 0: Suspend signal will not reach the peripheral. Peripheral will act as normal even during a debug halt. 1: Suspend signal will reach the peripheral. Peripheral will be suspended during debug halt.	RW	0x0

Table 26-96. Register Call Summary for Register DRM_SUSPEND_CTRL32

Primary Debug Support • Debug Aware Peripherals and Host Processors: [0]
DRM Register Manual • DRM Register Summary: [1]

Glossary

2

2D— Two Dimensional

3

3D— Three Dimensional

3G— 3rd Generation of mobile communication systems

A

A/D— Analog to Digital Converter

ABB— Adaptive Body Bias

ABE— Audio Back End

ABS— Absolute Value

ACE— ASIC Compiler Environment.

ACK— Acknowledge

ADC— Analog–to–Digital Converter/Conversion.

ADMA— Advanced DMA

AET— Advanced Event Triggering - this capability can be used to debug complex problems as well as understand performance characteristics of user applications

AFE— Analog Front End

AFSX— Audio transmit frame synchronization

AHB— Advanced High-performance Bus

AHB-AP— Advanced High-performance Bus - Access Port

ALE— Address Latch Enable

AMBA— Advanced Micro-Controller Bus Architecture

AMMU— Attribute Memory Management Unit

AP— Address Protection

APB— Advanced Peripheral Bus

APB-AP— Advanced Peripheral Bus - Access Port

APWM— Asymmetrical Pulse Width Modulation (eCAP)

AR— Automatic Reload

ARGB— Alpha Red Green Blue

ARM— Advanced RISC Machine

ASCII— American Standard Code for Information Interchange

ASIC— Application–Specific Integrated Circuit

ATL— Audio Tracking Logic

ATB— AMBA Advanced Trace Bus

AVB— Audio Video Bridging

AVS— Adaptive Voltage Scaling

AWD— Analog Watchdog

B

B— Byte, 8 bits

BB— Bus Busy

BCD— Binary–Coded Decimal.

BCH— Bose-Chaudhuri-Hocquenghem (CRC Code)

BE— Big Endian.

BGA— Ball Grid Array

BGT— Block Guard Time

BIOS— Basic Input/Output System.

BOF— Beginning of Frame

BOL— Beginning of Line

BSP— Buffered Serial Port.

BTA— Bus Turn Around

BW— Band Width

BWM— Band Width Manager (functional entity in the TMS320C66x DSP CorePac)

C

CAN— Controller Area Network

CAS— Column Address Strobe

CBC— Cipher Block Chaining

CBUF— Circular Buffer (Associated with OCMC)

CC— Channel Controller

CCCR— Card Common Control Registers (SDIO)

CE— Chip Enable

CFI— Common Flash Interface device

CH— Configuration Header. To use different settings than ROM Code defaults, i.e. clock frequencies, SDRAM settings, GPMC settings if the customer wants.

CIF— Common Intermediate Format.

CIR— Consumer Infra Red

CL— Command List (SATA)

CLE— Command Latch Enable

CLK— Clock

CLUT— Color Look-Up Table

CM— Clock Management

CME— Credit Management Engine

CMOS— Complimentary Metal Oxide Semiconductor

COS— Class of Service

CPPI— Communications Port Programming Interface

CPR— Color Phase Rotation

CPU— Central Processing Unit

CRC— Cyclic Redundancy Check

CS— Chip–Select

CSWR— Closed switch retention

CTM— Counter-Timer Module

CTRL— Control

CTS— Clear to Send

CTT— Clock Tree Tool. Interactive PC tool from TI to assist PRCM software user.

CVBS— Composite Video Broadcast Signal

ConnID— Connection Identifier.An Initiator Module Identifier. A ConnID is transmitted in-band with the request and is used for security and error logging mechanism.

D

DAC— Digital to Analog Converter

DAP— Debug Access Port

DC— Direct Current

DCAN— CAN Controller Module

DCC— Duty Cycle Correction Circuit

DCD— Data Carrier Detect

DCO— Digitally Controlled Oscillator

DDR— Double Data Rate

DE— Data Enable

DED— Double Error Detection

DFF— Data Flip-Flop

DFT— Design For Test

DIR— Digital audio Interface Receiver

DISPC— Display Controller

DLB— Data Loopback.

DLL— Delay-Locked Loop

DMA— Direct Memory Access.

DMC— Digital Motor Control

DMEM— Data Memory

DMT— Display Monitor Timing

DPC— Defect Pixel Correction

DPI— Display Parallel Interface

DPLL— Digital Phase–Locked Loop. Digital implementation of PLL.

DPS— Digital Power Switching

DRDY— Data Ready

DRM— Digital Rights Management

DSI— Display Serial Interface

DSP— Digital Signal Processor.

DSR— Data Set Ready

DSS— Display Sub-System

DTC— Debug and Trace Controller

DTCM— Data Tightly Coupled Memory

DTR— Data Transmit Ready

DVFS— Dynamic Voltage and Frequency Scaling

DWD— Digital Watchdog

E

EAV— End of Active Video

ECC— Error Checking and Correction. Also Error Correction Code.

ED— Endpoint Descriptor

EDC— Error Detection Code

EDI— Edge Directed Interpolation

EDMA— Enhanced DMA

EDMA_TPCC— Enhanced DMA Third Party Channel Controller

EDMA_TPTC— Enhanced DMA Third Party Transfer Controller

EEE— Energy Efficient Ethernet

EMC— External Memory Controller

EMIF— External Memory Interface

EOB— End of Block

EOF— End of Frame

EOL— End of Line

EOP— End of Packet

EOT— End of Transfer

EPM— Emulation Pin Manager

ESM— Error Signaling Module

ESR— Event Set Register

ETB— Embdedded Trace Bus

ETM— Embedded Trace Macrocell

EVE— Embedded Vision Engine

F

FC— Frame Counter

FCLK— Functional Clock

FE— Framing Error

FF— Flip-Flop

FIFO— First In First Out

FIQ— Fast Interrupt Request. See ISR.

FIR— Fast Infrared

FM— Frequency Modulation

FRC— Free Running Counter

FROM— eFuse ROM

FS— Frame Synchronisation

FSC— Frame Start Code. Also Frame Start Count.

FSM— Finite State Machine.

FW— Firewall

G

GFX— Graphics

GMII— Gigabit Media Independent Interface

GP— General-purpose

GPIO— General Purpose Input/Output

GPMC— General Purpose Memory Controller

H

H/W— Hardware

HAL— Hardware Abstraction Layer

HBP— Horizontal Back Porch

HC— Host Controller

HD— High Definition

HDMI— High-Definition Multimedia Interface

HDTV— High-Definition Television

HFP— Horizontal Front Porch

HLOS— High-Level Operating System

HPF— High-Pass Filter

HRPWM— High Resolution Pulse Width Modulator (eHRPWM only)

HS— High-Speed

HSSCLL— High-Speed Serial Control Channel

HSW— Horizontal Synchronization Pulse Width

HSYNC— Horizontal Synchronization.

HW— Hardware

HWOBS— Hardware Observability

I

I/F— Interface

I/O— Input/Output

I2C— Inter–Integrated Circuit

I2S— Inter–IC Sound

IA— Identifier Address

ICE— In–Circuit Emulation

ICEPICK— Generic TAP for emulation control

ICLK— Interface Clock

ID— Identification

IDE— Integrated Development Environment. A programming environmentintegrated into an application.

INT— Interrupt .

INTC— Interrupt Controller

IP— Intellectual Property

IPC— Interprocessor Communication. (also referred to as “mailbox” on occasion)

IPU— Image-processing unit

IR— Infrared

IRQ— Interrupt Request.

ISA— Instruction Set Architecture

ISO— Isochronous.

ISR— Interrupt Service Routine.

IST— Interrupt Service Thread.

ITCCHEN— Intermediate transfer completion chaining enable.

ITCINTEN— Intermediate transfer completion interrupt enable.

ITCM— Instruction Tightly Coupled Memory

IVA— Image and Video Accelerator

J

JEDEC— Joint Electronic Devices Engineering Council

JTAG— Joint Test Action Group.

K

KB— Kilobyte, 1024 B

Kbps— Kilobits per second

L

L1— Level 1 cache/memory

L2— Level 2 cache/memory

L3— First level of interconnect

L4— Second level of interconnect

LA— Logical Address

LAN— Local Area Network

LBIST— Logical Built–In Self–Test

LCD— Liquid Crystal Display.

LD— Lens Distortion

LDO— Low Dropout

LE— Little Endian.

LED— Light Emitting Diode.

LF— Loop Filter

LH— Local Host

LINK— Link Layer Device

LLP— Low-Level Protocol

LPP— Lines Per Panel

LRU— Least Recently Used

LSB— Least Significant Bit

LSR— Linear Shift Register

LTC— Low Time Counter

LVCMOS— Low Voltage Complementary Metal Oxide Semiconductor

M

MAC— Multiply Accumulate. Also Medium Access Control

MBIST— Memory Built–In Self–Test

MCAN— Modular CAN Controller Module

McSPI— Multichannel Serial Peripheral Interface

MCU— Microcontroller Unit

MDIO— Management Data Input/Output

MDMA— Modem DMA. Second meaning is : Master DMA port of the DSPSS TMS320C66x CorePac.

MIF— Memory InterFace

MII— Media Independent Interface

MIPI— Mobile Industry Processor Interface

MIR— Medium Infrared

MMC— Multimedia Card

MMR— Memory Mapped Register

MMU— Memory Management Unit.

MPU— Microprocessor Unit.

MSB— Most Significant Bit

MUX— Multiplex/Multiplexer

Mbps— Mega bits per second

Modem— Modulator Demodulator

MuxMode— Refers to top-level pin multiplexing

N

N/A— Not Applicable

NAND— NAND Flash memory.

NIU— Network Interface Unit

NMI— Non-Maskable Interrupt

NOP— No OPeration (DSP/CPU instruction)

NOR— A type of flash memory

NRT— Nonreal–Time

NRZ— Non-Return-to-Zero

NTSC— National Television System Committee. Television broadcast system.

NVIC— Nested Vectored Interrupt Controller

O

OCM— On-chip Memory

OCMC— On-chip Memory Controller

OCP— Open–Core Protocol

OE— Output Enable

OMAP— Open Multimedia Application Platform

OPP— Operating Performance Point

OS— Operating System

P

PC— Program Counter

PCB— Printed Circuit Board

PCLK— Pixel Clock

PCM— Pulse Code Modulation.

PDC— Power-down Controller

PE— Parity Error

PFPW— Prefetch and Prewrite posting engine

PHY— Physical Layer Device

PID— Proportional-Integral-Derivative

PLL— Phase–Locked Loop

PM— Programming Model

PMEM— Program Memory

PMFW— Power Management FrameWork

PMIC— Power management Integrated Circuit

POR— Power–On Reset

PPL— Pixels per Line

PPM— Pulse-Position Modulation

PRCM— Power, Reset, Clock Management module

PRM— Power and Reset manager

PSA— Parallel Signature Analyzer

PSC— Prescaler Counter

PTI— Parallel Trace Interface

PWM— Pulse Width Modulation

PaRAM— Programmable RAM that stores 32-byte channel transfer definition that EDMA channels, QDMA channels, and linking uses.

Q

QDMA— Quick DMA

QOS— Quality of Service

QSPI— Quad Serial Peripheral Interface

R

R/W— Read/Write. Also RW.

RAM— Random Access Memory

RE— Read Enable

REQ— Request

RFBI— Remote Frame Buffer Interface

RFF— Retention Flip-Flop

RFU— Receive Format Unit

RGB— Red Green Blue

RGBA— Red Green Blue Alpha

RGMII— Reduced Gigabit Media Independent Interface

RI— Ring Indicator

RISC— Reduced Instruction Set Computing. A CPU whose instruction set and related decode mechanism are much simpler than those of microprogrammed complex instruction set computers. The result is a higher instruction throughput and a faster real–time interrupt service response from a smaller, cost–effective chip.

RMII— Reduced Media Independent Interface

RO— Read Only

ROM— Read Only Memory. A semiconductor storage element containing permanent data that cannot be changed.

RST— Reset

RT— Real–Time

RTA— Retention Till Access

RTC— Real–Time Clock. A clock that keeps track of the time even when the device is turned off.

RTI— Real Time Interrupt

RTOS— Real–Time Operating System

RTS— Request to Send

RX— Receive/Receiver

RXD— Receive Data

S

S/PDIF— Sony/Philips Digital Interface

S/W— Software

SAF— Standby Associated Function

SAR— Save and Restore. Hardware context saving for power saving.

SAV— Start of Active Video

SCL— Serial Clock. Programmable serial clock used in the I2C interface. Also SCLK.

SCR— SDIO Configuration Registers

SCTM— System Counter Timer Module

SD— Sigma-Delta. Also Secure Digital card (a non-volatile memory card).

SDA— Serial Data. Serial data bus in the I2C interface.

SDIO— Secure Digital Input/Output

SDMA— System Direct Memory Access module, Second meaning is: The Slave DMA port of the DSPSS TMS320C66x CorePac

SDP— Software Development Platform

SDR— Single Data Rate

SDRAM— Synchronous Dynamic Random Access Memory

SDRC— SDRAM Controller.

SDTV— Standard Digital Television

SEC— Single Error Correction

SECDED— Single-Error Correction Dual-Error Detection

SFI— Serial Flash Interface

SIF— Source Input Format

SIMD— Single Instruction-Stream, Multiple Data-Stream

SIR— Slow Infrared

SL2— Shared Level 2 (memory/interface)

SLM— Static Leakage Management

SMPS— Switch Mode Power Supply

SMSET— Software Message and System Event Trace module

SOC— System-On-a-Chip

SOF— Start Of Frame

SP— Serial Port or Small Page or Stack Pointer

SPI— Serial Peripheral Interface

SRAM— Static Random Access Memory

SRC— Sample Rate Conversion

SS— Subsystem

SSC— Spread Spectrum Clocking

STC— Store from Coprocessor (to memory) or System Time Clock, which is the master clock in an MPEG2 encoder or decoder system.

STM— Synchronous Transfer Mode or Store Multiple.

STN— Super–Twist Nematic. A technique for improving LCD display screens by twisting light rays.

SVC— Supervisor Call

SW— Software

SWI— Software Interrupt

SWJ-DP— CoreSightTM Serial Wire JTAG Debug Port

SYNCDIM— Transfer synchronization dimension.

SmartReflex— Dynamic voltage sensing module that generates the voltage error signal proportional to the difference in desired voltage and the current voltage

T

TA— Target Agent

TAP— Test Access Port

TBCLK— Time Base Clock (PWMSS ePWM / eHRPWM related)

TBCNT— Time Base Clock - driven Counter (PWMSS ePWM / eHRPWM related)

TBPRD— Period Value of the Time Base Counter (PWMSS ePWM / eHRPWM related)

TC— Traffic Controller. Allows asynchronous operation among the external memory interface, the MPU, and the DSP. Also, in PCI Express context - this means Traffic Class, mapped to a virtual channel

TCCHEN— Transfer complete chaining enable.

TCINTEN— Transfer complete interrupt enable.

TCK— Test Clock

TCM— Tightly Coupled Memory

TDI— Test Data Input

TDM— Time Division Multiplex/Multiplexing

TDO— Test Data Output

TESOC— Tester On-chip

TFT— Thin Film Transistor. A type of LCD flat panel display screen in which each pixel is controlled by one to four transistors.

TI— Texas Instruments

TILER— Tiling Isometric Lightweight Engine for Rotation

TLB— Translation Lookaside Buffer. A cache that contains entries for virtual–to–physical address translation and access permission checking.

TM— Target Module. A target module cannot generate read/write requests to the chip interconnects, but respond to these requests. However it may generate interrupts or DMA request to the system (typically: peripherals, memory controllers).

TMS— Test Mode Select

TOC— Table of Contents

TPIU— Trace Port Interface Unit

TR— Tranfer Request

TRM— Technical Reference Manual

TRST— Test Reset

TS— Transmission Start or Time-Stamp

TSHUT— Temperature Shutdown.

TTL— Transistor Transistor Logic or Time To Live (networking)

TWL— Table Walking Logic

TX— Transmit//Transmitter

TXC— Bidirectional Serial Transmit Clock

TXD— Transmit Data

U

UART— Universal Asynchronous Receiver/Transmitter. Another name for the asynchronous serial port.

UHS— Definition of SD cards with higher frequency

V

VA— Virtual Address

VBP— Vertical Back Porch

VBUF— Virtual Frame Buffer

VC-1— Video Codec Standard

VCOP— Vector Coprocessor

VENC— Video Encoder

VESA— Video Electronics Standards Association

VFP— Vertical Front Porch

VID— VLAN Identifier

VIP— Video Input Port

VLAN— Virtual Local Area Network

VLD— Variable Length Coder

VLIW— Very Long Instruction Word

VP— Video Port

VS— Vertical Synchronization

VSW— Vertical Synchronization Pulse Width

VSYNC— Vertical Synchronization. A bidirectional vertical timing signal occurring once per frame with a pulse–width defined as an integral number of lines (half–lines for interlaced mode). Also VS.

W

WB— Write Buffer

WC— Word Count

WD— Watchdog

WDT— Watchdog Timer. See WD.

WE— Write Enable

WFI— Wait For Interrupt

WIR— Wait In Reset

WNP— Write Non-Posted

WP— Write Protect

WSS— Wide-Screen Signaling

WT— Write Through

WUGEN— Wake-Up Generator

WWD— Windowed Watchdog

Word16— 16 bits word

X

X-LOADER— A user–defined pre–operating system bootstrap code that resides at the beginning of the external flash.

XIP— eXecution In Place

XMC— Extended Memory Controller (a functional entity in the TMS320C66X DSP CorePac)

Y

YUV— Luminance-Bandwidth-Chrominance

e

eCAP— Enhanced Capture Module

eHRPWM— Enhanced PWM module with an integrated High Resolution PWM feature

ePWM— Enhanced Pulse Width Modulation Logic (excluding high resolution PWM feature)

eQEP— Enhanced Quadrature Encoder Pulse Module

IMPORTANT NOTICE AND DISCLAIMER

TI PROVIDES TECHNICAL AND RELIABILITY DATA (INCLUDING DATASHEETS), DESIGN RESOURCES (INCLUDING REFERENCE DESIGNS), APPLICATION OR OTHER DESIGN ADVICE, WEB TOOLS, SAFETY INFORMATION, AND OTHER RESOURCES “AS IS” AND WITH ALL FAULTS, AND DISCLAIMS ALL WARRANTIES, EXPRESS AND IMPLIED, INCLUDING WITHOUT LIMITATION ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE OR NON-INFRINGEMENT OF THIRD PARTY INTELLECTUAL PROPERTY RIGHTS.

These resources are intended for skilled developers designing with TI products. You are solely responsible for (1) selecting the appropriate TI products for your application, (2) designing, validating and testing your application, and (3) ensuring your application meets applicable standards, and any other safety, security, or other requirements. These resources are subject to change without notice. TI grants you permission to use these resources only for development of an application that uses the TI products described in the resource. Other reproduction and display of these resources is prohibited. No license is granted to any other TI intellectual property right or to any third party intellectual property right. TI disclaims responsibility for, and you will fully indemnify TI and its representatives against, any claims, damages, costs, losses, and liabilities arising out of your use of these resources.

TI’s products are provided subject to TI’s Terms of Sale (www.ti.com/legal/termsofsale.html) or other applicable terms available either on ti.com or provided in conjunction with such TI products. TI’s provision of these resources does not expand or otherwise alter TI’s applicable warranties or warranty disclaimers for TI products.

Mailing Address: Texas Instruments, Post Office Box 655303, Dallas, Texas 75265